FUJITSU

Fujitsu Enterprise Postgres 17 for Kubernetes

Manual set

November 2024

FUJITSU

Fujitsu
Enterprise Postgres 17 for Kubernetes

Release Notes

Linux

J2UL-RN17-01ENZ0(00)
November 2024

Preface

Purpose of this document

This document provides release information for Fujitsu Enterprise Postgres for Kubernetes.

Structure of this document

This document is structured as follows:

Chapter 1 New Features and Improvements

Explains the new features and improvements in this version.

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Fujitsu Enterprise Postgres	FEP
Custom Resource	CR
Universal Base Image	UBI
OpenShift Container Platform	OCP
Mutual TLS	MTLS

Abbreviations of manual titles

The following abbreviations are used in this manual as manual titles:

Full Manual Title	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Release Notes	Release Notes
Fujitsu Enterprise Postgres for Kubernetes Overview	Overview
Fujitsu Enterprise Postgres for Kubernetes User's Guide	User's Guide
Fujitsu Enterprise Postgres for Kubernetes Reference	Reference

Trademarks

- Linux is a registered trademark or trademark of Mr. Linus Torvalds in the U.S. and other countries.
- Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.
- S/390 is a registered trademark of International Business Machines Corporation in the United States or other countries or both.
- Power and Power Systems are registered trademarks of International Business Machines Corporation in the United States or other countries or both.

Other product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

```
Edition 1.0: November  2024
```

Copyright

Copyright 2021-2024 Fujitsu Limited

Contents

Chapter 1 New Features and Improvements....1
1.1 Features Added FEP17 Operator in v7.1.0....1
1.1.1 OSS....1
1.1.1.1 PostgreSQL Rebase....1
1.1.1.2 OSS Updates Provided....1
1.1.2 Platform Enhancement....1
1.1.2.1 Additional OCP Support....1
1.1.2.2 Additional Kubernetes Support....2
1.1.3 Performance....2
1.1.3.1 Scheduling of an aggressive freeze for tuples (VACUUM FREEZE)....2
1.1.4 Operation....2
1.1.4.1 Vector-enabled database....2
1.1.5 Monitoring....2
1.1.5.1 Operational database monitoring with cloud services....2

Chapter 1 New Features and Improvements

This chapter explains Fujitsu Enterprise Postgres for Kubernetes new features and improvements added in this version.

Table 1.1 New features and improvements

Version and level	Classification	Feature
FEP 17 Operator image tag:v7.1.0 Container image tag:ubi9-17-1.0	OSS	PostgreSQL Rebase
		OSS Updates Provided
	Platform Enhancement	Additional OCP Support
		Additional Kubernetes Support
	Performance	Scheduling of an aggressive freeze for tuples (VACUUM FREEZE)
	Operation	Vector-enabled database
	Monitoring	Operational database monitoring with cloud services

1.1 Features Added FEP17 Operator in v7.1.0

This section explains the improvement in Fujitsu Enterprise Postgres for Kubernetes v7.1.0.

1.1.1 OSS

This section explains the new feature related to OSS:

- PostgreSQL Rebase
- OSS Updates Provided

1.1.1.1 PostgreSQL Rebase

The PostgreSQL version that Fujitsu Enterprise Postgres is based on is 17.0.

1.1.1.2 OSS Updates Provided

The OSS provided by Fujitsu Enterprise Postgres and Fujitsu Enterprise Postgres for Kubernetes has been updated.

See

Refer to "OSS Supported by Fujitsu Enterprise Postgres for Kubernetes" in the Overview.

1.1.2 Platform Enhancement

This section explains the new feature related to platform enhancement:

- Additional OCP Support
- Additional Kubernetes Support

1.1.2.1 Additional OCP Support

The following additional OCP is supported:

- OCP 4.16

See

Refer to "Supported Platform" in the User's Guide for details.

1.1.2.2 Additional Kubernetes Support

The following additional Kubernetes is supported:

- Kubernetes 1.29

See

Refer to "Supported Platform" in the User's Guide for details.

1.1.3 Performance

This section describes new features related to Performance.

- Scheduling of an aggressive freeze for tuples (VACUUM FREEZE)

1.1.3.1 Scheduling of an aggressive freeze for tuples (VACUUM FREEZE)

Provide features for monitoring transaction ID usage and periodically performing an aggressive freeze for tuples (VACUUM FREEZE), to avoid work stoppages when autovacuum does not perform freezing of transaction IDs in time.

See

Refer to "Scheduling of an aggressive freeze for tuples (VACUUM FREEZE)" in the User's Guide.

1.1.4 Operation

This section describes new features related to Operation:

- Vector-enabled database

1.1.4.1 Vector-enabled database

It captures the peripheral OSS pgvector, allowing vector storage and similarity searching to work.

See

Refer to "OSS Supported by Fujitsu Enterprise Postgres for Kubernetes" in the Overview.

1.1.5 Monitoring

This section describes new features related to Monitoring.

- Operational database monitoring with cloud services

1.1.5.1 Operational database monitoring with cloud services

You can use monitoring tools such as Amazon CloudWatch provided by Amazon Web Services, to collect metrics and logs about your databases, and use the collected information to monitor database operations.

See

Refer to "Collaboration with Amazon CloudWatch" and "FEP Logging" in the User's Guide.

FUJITSU

Fujitsu
Enterprise Postgres 17 for Kubernetes

Overview

Linux

J2UL-OV17-01ENZ0(00)
November 2024

Preface

Purpose of this document

This document explains the Fujitsu Enterprise Postgres for Kubernetes concepts to those who are to operate databases using it.

This document explains the features of Fujitsu Enterprise Postgres for Kubernetes.

Intended readers

This document is intended for people who are:

- Considering installing Fujitsu Enterprise Postgres for Kubernetes
- Using Fujitsu Enterprise Postgres for Kubernetes for the first time
- Wanting to learn about the concept of Fujitsu Enterprise Postgres for Kubernetes
- Wanting to see a functional overview of Fujitsu Enterprise Postgres for Kubernetes

Readers of this document are also assumed to have general knowledge of:

- Linux
- Kubernetes
- Containers
- Operators

Structure of this document

This document is structured as follows:

Chapter 1 Know about the Product

Explains the features of Fujitsu Enterprise Postgres for Kubernetes.

Chapter 2 Know What it does

Explains what you need to do.

Appendix A OSS Supported by Fujitsu Enterprise Postgres for Kubernetes

Explains the OSS supported by Fujitsu Enterprise Postgres for Kubernetes.

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Fujitsu Enterprise Postgres	FEP
Custom Resource	CR
Custom Resource Definition	CRD
Grafana, Alert Manager, Prometheus	GAP
Persistent Volume	PV
Persistent Volume Claim	PVC

Abbreviations of manual titles

The following abbreviations are used in this manual as manual titles:

Full Manual Title	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes User's Guide	User's Guide

Trademarks

- Linux is a registered trademark or trademark of Mr. Linus Torvalds in the U.S. and other countries.
- Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.
- S/390 is a registered trademark of International Business Machines Corporation ("IBM") in the U.S. and other countries.

Other product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

Edition 1.0: November 2024

Copyright

Copyright 2021-2024 Fujitsu Limited

Contents

Chapter 1 Know about the Product 1
1.1 What is Fujitsu Enterprise Postgres for Kubernetes? 1
1.2 Operator Features 2
1.2.1 Cluster Deployment 2
1.2.1.1 Creating a FEPCluster 2
1.2.1.2 Creating a FEP Pgpool2 Container 3
1.2.2 Highly Available Feature 3
1.2.2.1 Automatic Failover 3
1.2.2.2 Automatic Recovery 3
1.2.2.3 Manual Switchover 3
1.2.3 Backup Recovery 4
1.2.3.1 Automatic Backup 4
1.2.3.2 Point-in-time Recovery 4
1.2.4 Configuration Change 4
1.2.4.1 Parameter Change 4
1.2.4.2 Resource Change 4
1.2.4.3 Disk Expansion 4
1.2.5 Minor Version Upgrade 5
1.2.5.1 Minor Version Upgrade 5
1.2.6 FEP Features 5
1.2.6.1 Scope of FEP Feature Support 5
1.2.7 Automating Audit Log Operations 5
1.2.8 Monitoring & Alert 6
1.2.8.1 Monitoring 6
1.2.8.2 Alert and Event 6
1.2.9 Collaboration with Amazon CloudWatch 6
1.2.10 Scaling Replicas 7
1.2.10.1 Automatic Scale out 7
1.2.10.2 Manual Scale in/out 7
1.2.11 Disaster Recovery 7
1.2.12 Transparent Data Encryption Using Key Management System 8
1.2.13 Fixed Statistics 8
1.3 Operator System Configuration 8

Chapter 2 Know What it does 10
2.1 Deployment 10
2.2 High Availability (Automatic failover and recovery) 10
2.3 Configuration Change 10
2.4 Upgrade 11
2.4.1 Minor Version Upgrade 11
2.4.2 Major Version Upgrade 11
2.5 Configurable Volume Per Cluster 11
2.6 Deploying Pgpool-II and Connect to FEPCluster from Operator 11
2.7 Backup 12
2.7.1 Scheduling Backup from Operator 12
2.7.2 On-Demand Backup 12
2.8 Perform PITR and Latest Backup Restore from Operator 12
2.9 Monitoring & Alert 13
2.10 Server Log Monitoring 14
2.10.1 pgBadger Log Monitoring 14
2.10.2 Prometheus and Elasticsearch Log Monitoring 15
2.11 Scaling Replicas 15
2.11.1 Using the automatic scale out function 16
2.12 Disaster Recovery 17
2.13 Ability to Customize the FEP Server Container Image 17

2.14 Cloud Secret Store 17

Appendix A OSS Supported by Fujitsu Enterprise Postgres for Kubernetes 19

Chapter 1 Know about the Product

This chapter explains the features of Fujitsu Enterprise Postgres for Kubernetes.

1.1 What is Fujitsu Enterprise Postgres for Kubernetes?

Fujitsu Enterprise Postgres for Kubernetes provides automated operations for installing and managing your Fujitsu Enterprise Postgres 17 on OpenShift Container Platform or Kubernetes.

There are multiple components in the solution.

FEP operator: Manages the lifecycle of FEP server container, including deployment, configuration update, backup and recovery of FEP database.

FEP server container: Contains the FEP server software to run the Postgres engine.

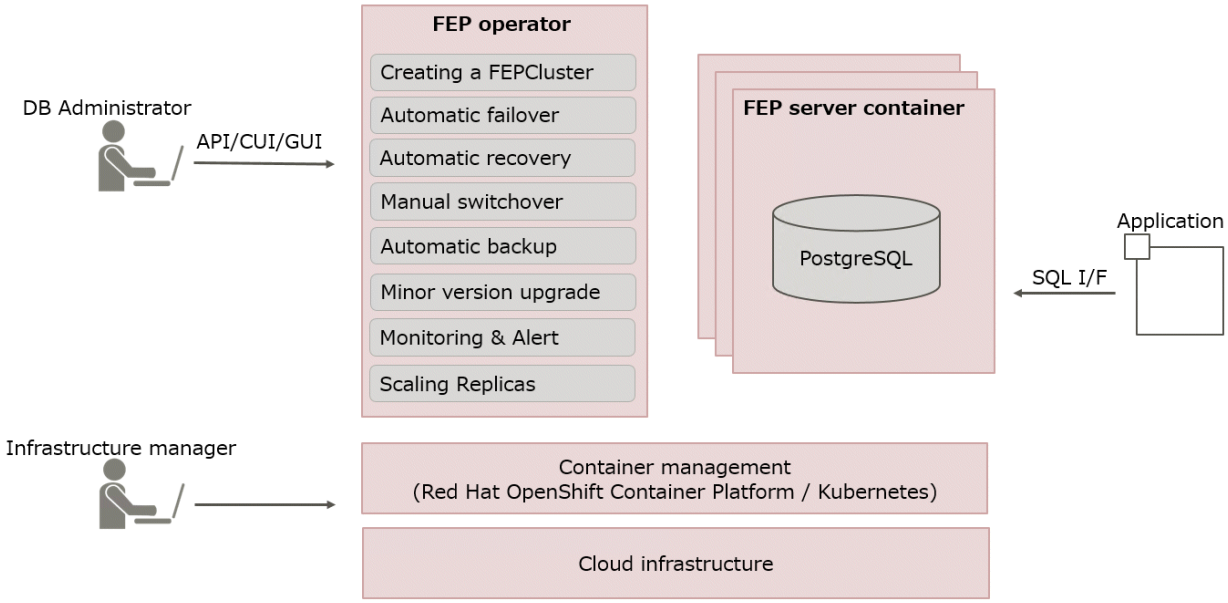
FEP backup container: Contains the FEP server software to perform scheduled backup operations.

FEP restore container: Contains the FEP server software to perform the restore operation.

FEP pgpool2 container: Contains the FEP server software to use Pgpool-II to provide load balancing and connection pooling.

FEP exporter container: expose various health metrics to Prometheus for monitoring

Up and running in minutes, the operator provides the features required to maximise the benefits of this enterprise PostgreSQL solution.



This operator will deploy a standalone as well as highly available Fujitsu Enterprise Postgres for Kubernetes provides automated operations for installing and managing your Fujitsu Enterprise Postgres 17 on OpenShift Container Platform or Kubernetes.

Enterprise Postgres cluster with pre-defined configuration to get started with small workload. User can adjust the configuration parameters at the time of deployment and after to make the instance suitable for the workload.

As the name implies, the FEP server container is intended to incorporate the Fujitsu Enterprise Postgres for Kubernetes provides automated operations for installing and managing your Fujitsu Enterprise Postgres 17 on OpenShift Container Platform or Kubernetes.

Enterprise Postgres server component.

In principle, a running FEP server container is considered as equivalent to a Fujitsu Enterprise Postgres for Kubernetes provides automated operations for installing and managing your Fujitsu Enterprise Postgres 17 on OpenShift Container Platform or Kubernetes.

Enterprise Postgres Server instance.

1.2 Operator Features

This product provides operator services to automate the construction and operation of databases on the customer's container management infrastructure. The features of the operator are as follows:

- Cluster Deployment
  - Creating a FEPCluster
  - Creating a FEP Pgpool2 Container
- Highly Available Feature
  - Automatic Failover
  - Automatic Recovery
  - Manual Switchover
- Backup Recovery
  - Automatic Backup
  - Point-in-time Recovery
- Configuration Change
  - Parameter Change
  - Resource Change
  - Disk Expansion
- Minor Version Upgrade
- FEP Features
- Automating Audit Log Operations
- Monitoring & Alert
- Collaboration with Amazon CloudWatch
- Scaling Replicas
- Disaster Recovery
- Transparent Data Encryption Using Key Management System
- Fixed Statistics

1.2.1 Cluster Deployment

1.2.1.1 Creating a FEPCluster

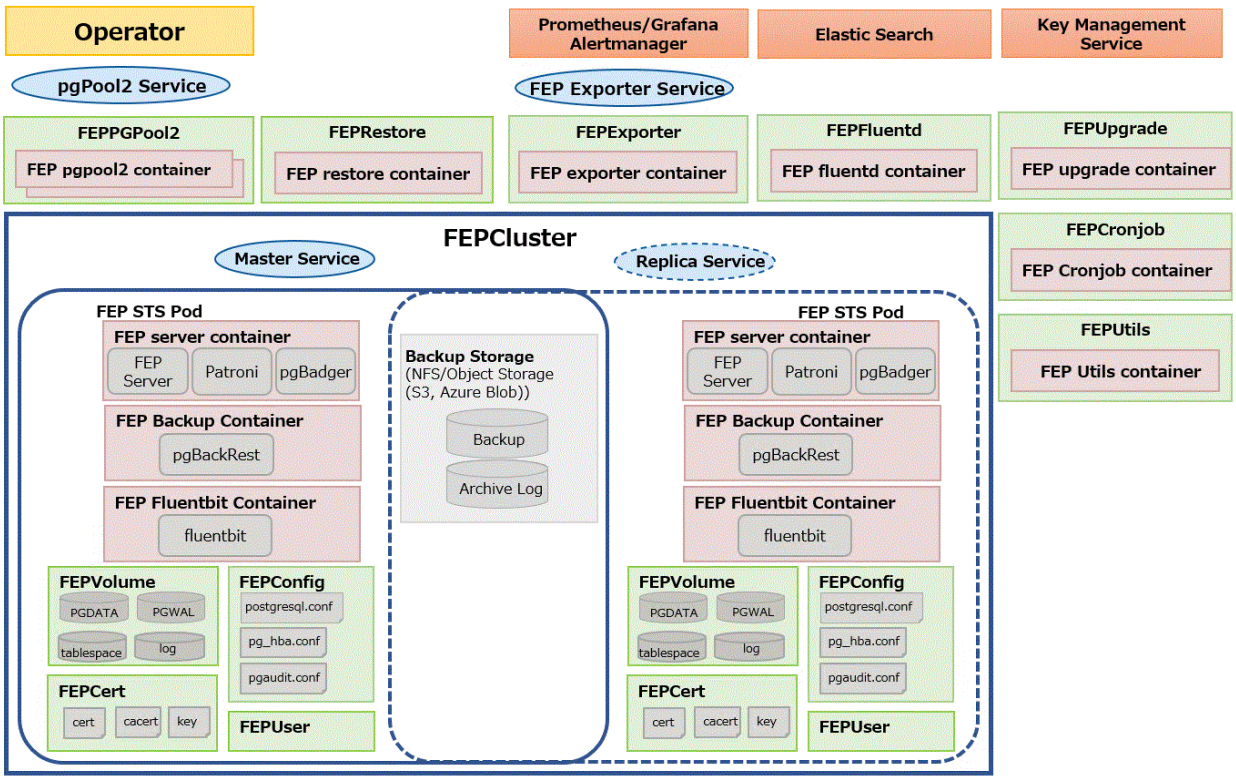
Users can instruct the operator to build a system that includes the provisioning of containers and volumes with FEP installed, and network resources. The resulting system is called a FEPCluster. The FEPCluster can be created a single master server or multi-servers with one master and two replicas. You can choose between synchronous and asynchronous replication replica servers. The default is asynchronous replication.

FEPCluster is composed of the following components:

- FEP server container
  - FEP server
  - Patroni
- FEP backup container
- CR FEPVolume for volumes

- CR FEPUser for database users
- CR FEPConfig for Postgres configuration
- CR FEPCert for secrets such as TLS certificate, keystore passphrase

The Below diagram depicts a FEPCluster with one Master and one Replica POD.



1.2.1.2 Creating a FEP Pgpool2 Container

Users can deploy Pgpool-II for load balancing and connection pooling with FEP pgpool2 container.

Users can deploy multiple FEP pgpool2 Pods in a single deployment to increase availability.

1.2.2 Highly Available Feature

1.2.2.1 Automatic Failover

When an error is detected in the container or POD of the master server, the cluster will perform an automatic failover by promoting one of the replicas to become the new master, and the connection destination of the database is switched. The database connection is broken, but you can reconnect by establishing a connection from the application again.

1.2.2.2 Automatic Recovery

If an error occurs on the master server and an automatic failover occurs, the POD or container of the failed old master server is automatically restarted and reincorporated into the cluster as a replica server.

If a replica server fails, it automatically restarts and rejoins the cluster as a replica server.

1.2.2.3 Manual Switchover

You can manually switch any replica server to the master server. In this case, the original master server becomes the replica server.

1.2.3 Backup Recovery

1.2.3.1 Automatic Backup

By taking regular backups, you can be prepared for full database downtime or data corruption due to application errors. Users can set an arbitrary schedule for automatic backup. The backup type can be a full backup or an incremental backup. You can back up the database to shared storage such as NFS persistent volume or AWS S3 compatible storage. Backups can be automatically deleted by setting a retention period of your choice.

1.2.3.2 Point-in-time Recovery

Point-in-time recovery can be used to recover data at specific times due to business failures or to replicate a cluster for migration to production environment. Allows point-in-time recovery from automated backup data to restore the cluster. You can choose between restoring data to an existing cluster and a new cluster. You can also choose to restore to the most recent data or to any time you specify.

1.2.4 Configuration Change

1.2.4.1 Parameter Change

You can change the parameters that make up the FEP. PostgreSQL provides two types of parameters: those that take effect immediately, and those that take effect after restarting FEP server process.

- postgresql.conf
- pg_hba.conf
- pgaudit.conf

Note

For parameters that take effect immediately, operator will apply the change to all FEP Pods and reload the FEP server process automatically. There is no outage on the cluster.

For parameters that take effect after restarting FEP server process, operator will update the configuration files on all FEP Pods. However, users have to initiate a manual restart of FEP process on all the FEP Pods using the FEPAction CR. There is a momentary outage on the cluster and the users should perform this action at a time that has least disruption to the service.

1.2.4.2 Resource Change

You can change the amount of CPU and memory resources allocated to FEP server containers, FEP backup containers, or FEP pgpool2 containers by changing the FEPCluster CR. The operator will apply the change to the Statefulset. However, the users have to perform a restart of all the Pods for the new resource allocation to take effect.

Note

Changing resource allocation will not take effect immediately. The users have to restart all the Pods for new resource allocation to take effect. There is a momentary outage on the cluster and the users should perform this action at a time that has least disruption to the service.

1.2.4.3 Disk Expansion

You can extend the disk size mounted on the FEP server container by modifying the FEPCluster custom resource.

Disk expansion can also work with Prometheus to automatically expand based on disk usage.

Disk extensions must take advantage of disks that support the Kubernetes PVC extension.

1.2.5 Minor Version Upgrade

1.2.5.1 Minor Version Upgrade

New and patched FEP releases are made available as new container image. When the latest container image is provided, the user can perform a minor version upgrade by changing the FEPCluster CR. The operator will perform a rolling update to enable the minor version upgrade with minimal system disruption.

Note

The minor version upgrade will take effect immediately. There is a momentary outage on the cluster and the users should perform this action at a time that has least disruption to the service.

1.2.6 FEP Features

1.2.6.1 Scope of FEP Feature Support

Note

These features also require the FEP Client.

The FEPCluster that is created supports the following features in addition to the PostgreSQL features of OSS. Enhances security and performance with transparent data encryption to prevent data loss in the event of database storage theft and in-memory capabilities with column-type index and data memory resident features to speed aggregation. Details of each feature can be found in the FEP documentation.

For transparent data encryption using a key management system, see "1.2.12 Transparent Data Encryption Using Key Management System".

Category	Feature
Operation	pgAdmin
	Global Meta Cache
Security	Transparent Data Encryption
	Audit Log
	Data Masking
High Performance	In-memory feature
	High-speed data load
Application Interface	Java Integration
	ODBC Integration
	.NET Framework Integration
	Embedded SQL Integration (C language)
	Embedded SQL Integration (COBOL)

1.2.7 Automating Audit Log Operations

As a Database Administrator you want to be able to perform audit schema changes or as a Security Officer, you want to be able to audit login/logout events, successful or failure, it is as part of company policy, you need to store auditlogs on external system for period of time.

As a Database Administrator, specify the following

- When the pgaudit function is enabled with the enable parameter in FEPCluster, the following is automatically set by the operator.
  - pgaudit is added in ‘shared_preload_libraries’
  - pgaudit log directory is configured
  - pgaudit file name is configured
  - pgaudit extension is created
- Specify an external ConfigMap for pgaudit configuration.
  - This ConfigMap contains full content of pgaudit.conf
  - Both session audit and object audit can be configured
  - The configurations in this ConfigMap overwrites the pgaudit configuration specified in FEPCluster CR
- Specify a destination so that the pgaudit log files can be uploaded periodically.
  - web server
  - Azure Blob
  - AWS S3 storage

Note

Azure blob is not supported on s390x platform.

1.2.8 Monitoring & Alert

1.2.8.1 Monitoring

Infrastructure administrator can start monitoring database almost simultaneously with database construction with standard monitoring tools.

Evaluation indicator data from a database point of view is provided in a format that can be displayed in Prometheus and Grafana.

The monitoring items are as follows:

- Database health
- OS performance information
- Disk usage
- Backup status
- Client connection information
- Server Log
- Audit Log

1.2.8.2 Alert and Event

Alerts enable infrastructure administrator to immediately understand and address anomalies. Define anomalous conditions from Monitoring's Matrix and set notifications in Prometheus. It is possible to integrate alerts with other services like emails, slack, sms or back-office systems for communication and action.

Perform recovery processing at the application layer after failover, synchronize with database backup, perform application backup, etc.

1.2.9 Collaboration with Amazon CloudWatch

You can work with CloudWatch, the Amazon Web Services(Hereinafter, AWS is used as an abbreviation.) monitoring service, to monitor FEP metrics and collect logs. The operator feature can automatically forward FEP metrics and logs to CloudWatch. The FEP and services

provided by AWS, such as EC2 and S3 object storage built into business systems, can be centrally monitored, reducing the workload of monitoring tasks.

1.2.10 Scaling Replicas

You can dynamically expand a read replica depending on the load on the read replica.

1.2.10.1 Automatic Scale out

With automatic scale out, the operator automatically extends the read replica according to the policy you specify.

The available policies are controlled by the CPU load or number of connections of read replica instance to automatically extend beyond a specified threshold.

Automatic scale out eliminates the need to use unnecessary resources for maximum potential load. It also reduces manual operations as the load increases.

- The automatic scale out feature adds replicas one at a time. In addition, additional replicas take time to service, depending on the environment and the amount of data stored. As a result, replica growth may not be able to keep up with the increased load.
- Even if the automatic scale out feature increases the number of replicas, incoming requests are not given priority to those replicas. As a result, existing FEP instances may continue to be temporarily overloaded after the number of replicas increases.
- The automatic scale out feature increases the number of replica requests that can be handled only by reference requests to the database. Requests with updates continue to be processed on the primary FEP instance. Therefore, the autoscale out feature may not reduce the load on the primary FEP instance.
- Currently, the automatic scale out feature does not delete replicas (reduce the number of replicas). If the load decreases after the number of replicas increases due to a temporary increase in load, the number of replicas remains increased. If necessary, manually change the number of replicas.

1.2.10.2 Manual Scale in/out

You can scale out or scale in the read replica at any time. This can be done by manipulating the CR of the FEPCluster.

1.2.11 Disaster Recovery

By using OSS (pgBackRest) functionality to store backup data in object storage, data can be migrated to a database cluster in a different OCP environment.

Even if it is difficult to operate in an OCP environment with a database cluster due to a disaster, it is possible to continue operating in a different OCP environment.

There are three disaster recovery methods available:

- Backup/Restore method

  Store backups of production environment data in object storage and restore data from object storage after a disaster.

- Continuous recovery method

  Create a container environment for production and another one for disaster recovery. Store the production environment data in object storage, and continuously restore to the disaster recovery environment.

- Streaming replication method

  Similarly to the continuous recovery method, create a container environment for production and another one for disaster recovery. Use streaming replication methods to synchronize data to the disaster recovery environment.

1.2.12 Transparent Data Encryption Using Key Management System

Key management system can be used to manage the master encryption key used in the transparent data encryption function. By using a key management system, you can centrally manage encryption keys according to your business security requirements.

The available key management systems for operators are:

- key management server using the KMIP protocol
- AWS key management service
- Azure key management service

Note

AWS and Azure key management services are only available in environments with a x86 CPU architecture.

1.2.13 Fixed Statistics

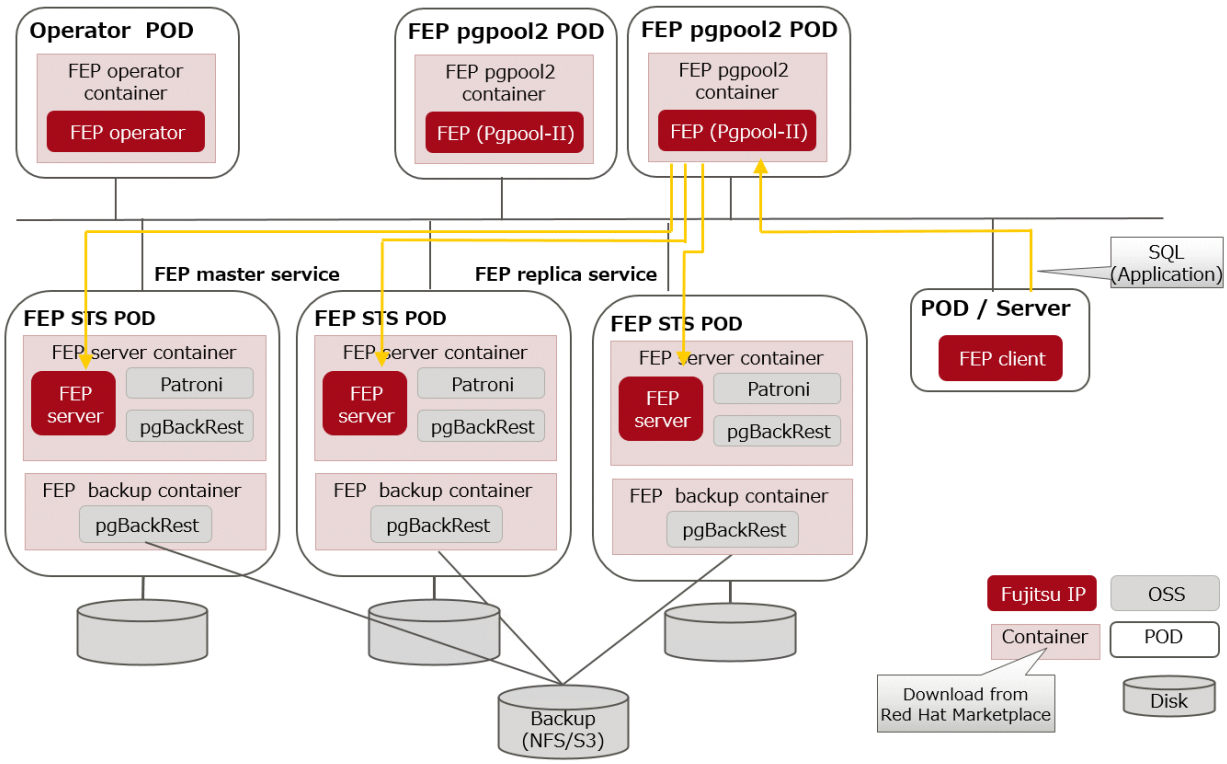
To prevent performance instability due to variations in execution plans, statistics can be fixed. By scheduling the fixation of statistics, regular statistics can be fixed without any burden on infrastructure administrators.

1.3 Operator System Configuration

The basic relationships among POD, containers and services are as follows.

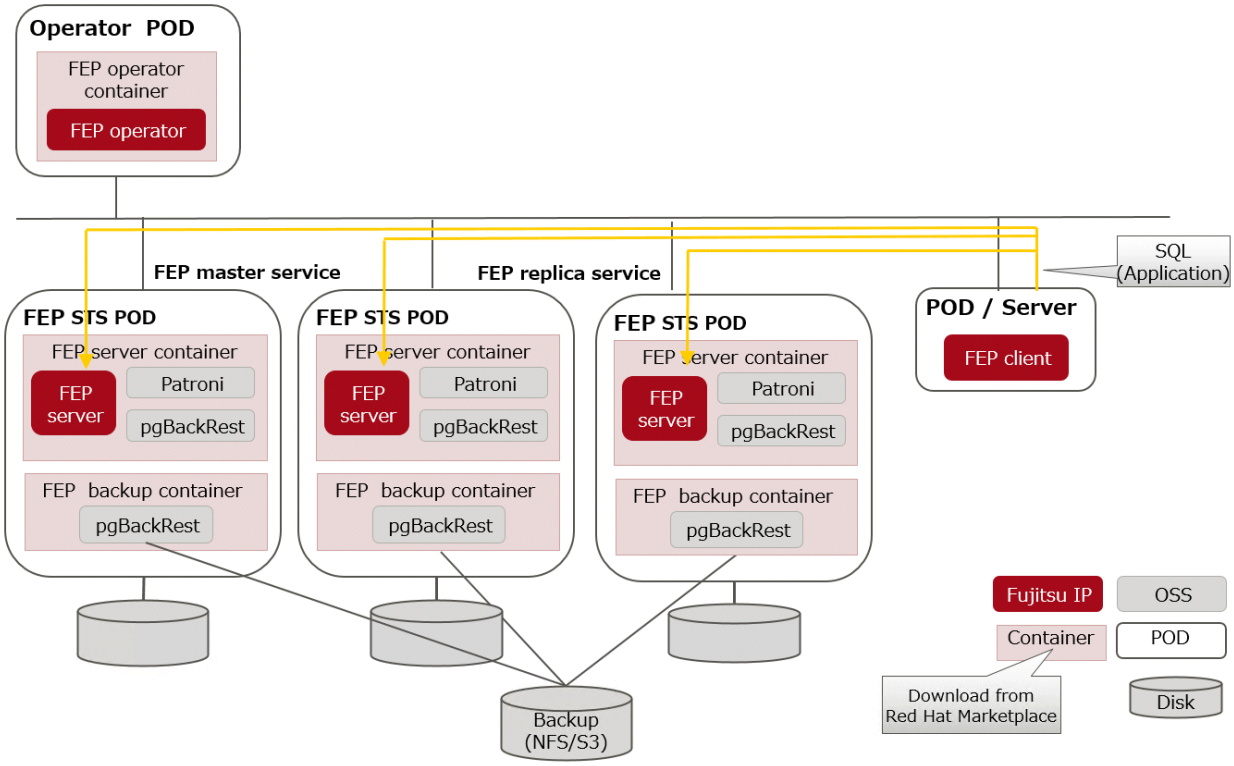
Example) Deployment with Pgpool-II

In this deployment scenario, Pgpool-II is used to provide connection pooling and load balancing. End user application will point its connection to Pgpool service. Depending on the transaction type, Pgpool will forward the connection to either the Master Pod or the Replica Pod. If a failover/switchover occurs, the FEP pgpool2 will direct traffic to the new FEP master Pod. This is transparent to the end user application.



Example) Deployment without Pgpool-II

Users can also run applications such as SQL directly against the FEPCluster without configuring Pgpool-II. In this deployment scenario, end user application will point its connection to the FEP master service. If a failover/switchover occurs, the FEP master service will point to the new FEP master Pod automatically. The end user application will experience a disconnection. When it re-establishes the connection, it will be connected to the new FEP master Pod. There is no need to reconfigure the application connection string.



Chapter 2 Know What it does

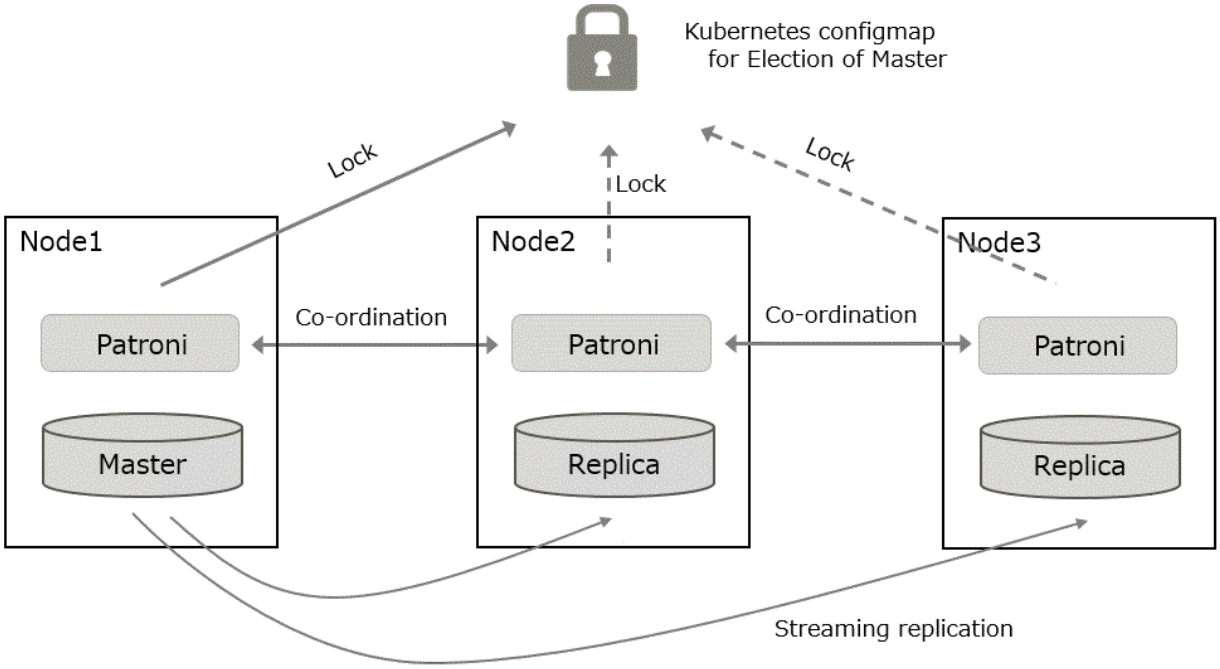
This chapter explains what you need to do.

2.1 Deployment

FEP operator is responsible for the lifecycle of FEPCluster. The operator will deploy a HA FEPCluster, together with all the associated containers such as backup container.

2.2 High Availability (Automatic failover and recovery)

The high availability and failover management of FEP is provided by Patroni. Both Patroni and FEP will be installed on the same container image. Patroni will then initialize and start an FEP instance. Patroni will then acquire a lock on a shared resource. In our case, it is a Kubernetes configmap. Whichever POD that can acquire the lock will become the Master. When subsequent FEP server container starts, Patroni will initialize that POD as a Replica with streaming replication.



If Patroni detects a failure in the cluster, either because the Postgres process crashed or the container where Postgres is running dies, Patroni will initiate a failover automatically.

2.3 Configuration Change

Traditionally, changing FEP configurations such as postgresql.conf, pg_hba.conf, TLS certificates and keystore passphrase will require a redeployment of FEP server container. That causes an outage in a Highly Available environment.

A new CRD FEPConfig is defined to encapsulate those configurations. The operator will monitor the CR with this CRD definition and perform action accordingly to minimize outages. For example, operator will reload FEP daemon, instead of redeploying the FEP server container when a reloadable postgresql.conf parameter is changed. If a parameter change requires restart of FEP (e.g. max_connections), the operator will update the configuration file but defer the restart. End user can follow a defined procedure to restart the cluster manually at a scheduled maintenance time.

2.4 Upgrade

2.4.1 Minor Version Upgrade

FEP version Minor upgrade is done by updating the Custom Resource with a new FEP image name. The POD will be redeployed with new image in a controlled manner. First, replica servers are upgraded, restarted and waited to be ready, one server at a time. When all replicas are upgraded, a controlled switchover is performed to pick a new master. Once that is done, the old master is upgraded as well.

2.4.2 Major Version Upgrade

You can upgrade a major version of FEP by specifying the upgrade parameters when you create the latest FEP cluster. Automates the process of dumping data from an older FEP cluster and restoring data to the new, latest cluster.

Application outages are required during major version upgrades.

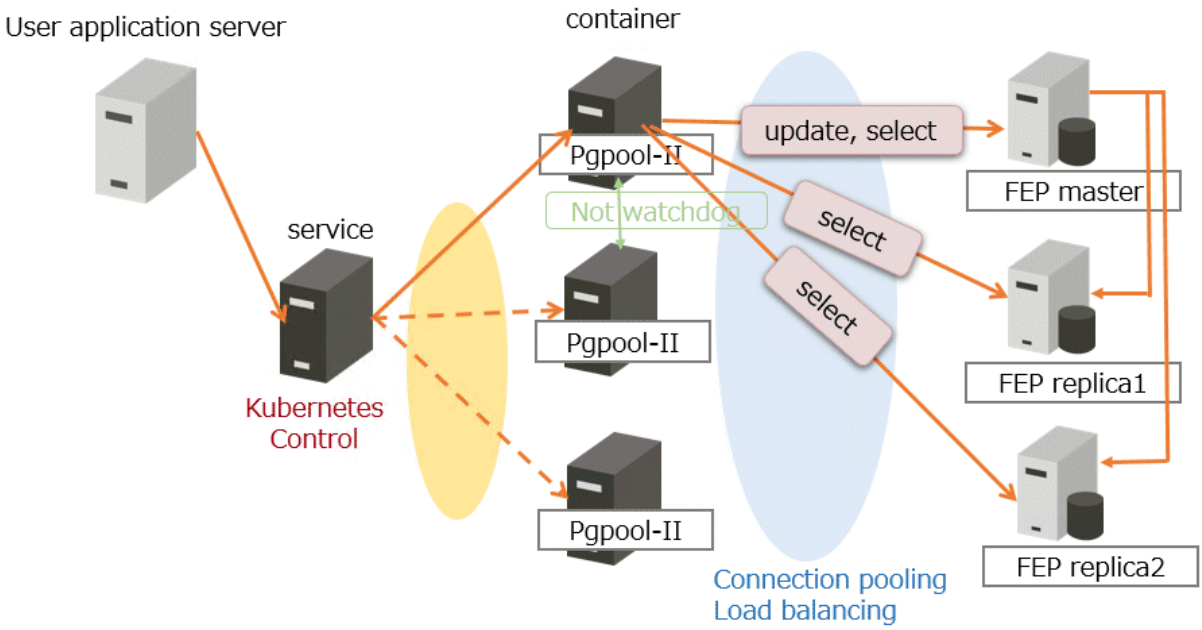
2.5 Configurable Volume Per Cluster

To improve performance, may want to separate the volume storing database files and WAL files. Similarly, one may want to use a dedicated volume for a new tablespace. The operator gives the end user the flexibility to create a FEPCluster with multiple PVs and select a suitable storage class for the PV. For example, one can create a FEPCluster with data volume, wal volume on a storage class backed up by SSD and a log volume on a storage class backed up by HDD.

2.6 Deploying Pgpool-II and Connect to FEPCluster from Operator

Users can deploy the FEP pgpool2 container and access the database via Pgpool-II to use load-balancing and connection pooling features.

Multiple FEP pgpool2 containers can be deployed for load-share and high availability. Users can request a Kubernetes service to distribute their work across multiple FEP pgpool2 containers.

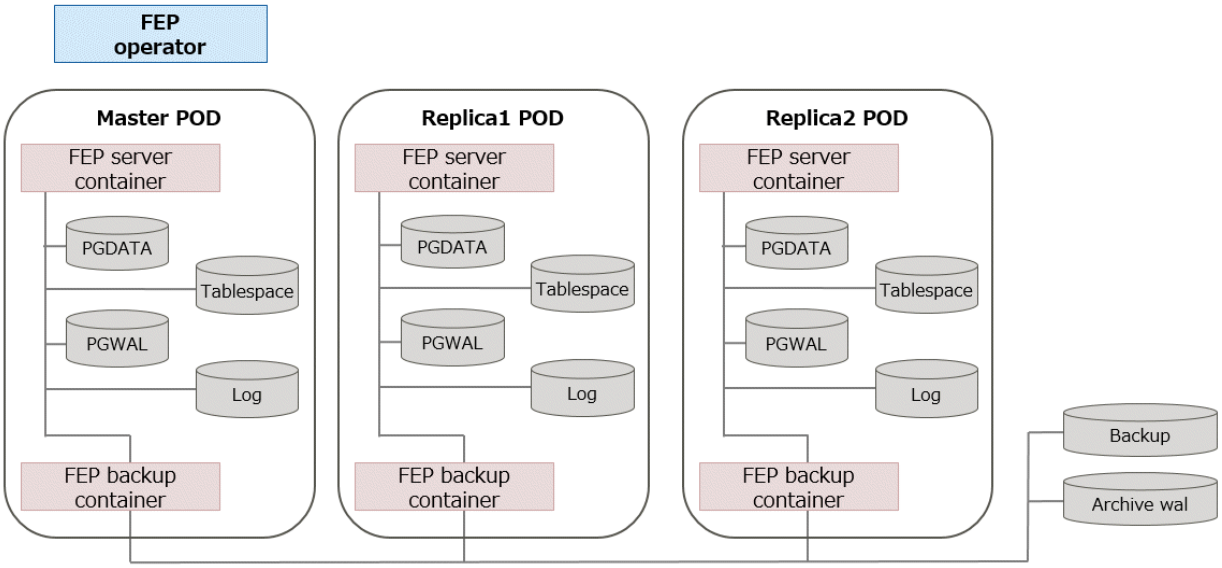


2.7 Backup

2.7.1 Scheduling Backup from Operator

The FEP backup container is deployed as a sidecar to each FEP server POD. The backup is performed at scheduled time set by the user (like crontab). The FEP backup container determines if the FEP server in the POD is a master or replica, and will perform the backup process only on the master POD. The volume storing backup and archived WAL files must be on a shared storage such as NFS or AWS S3.

Backup and WAL archiving is accomplished with pgBackRest.



2.7.2 On-Demand Backup

On-demand backups can also be taken at any time other than a predetermined schedule, such as before planned maintenance or after configuration changes.

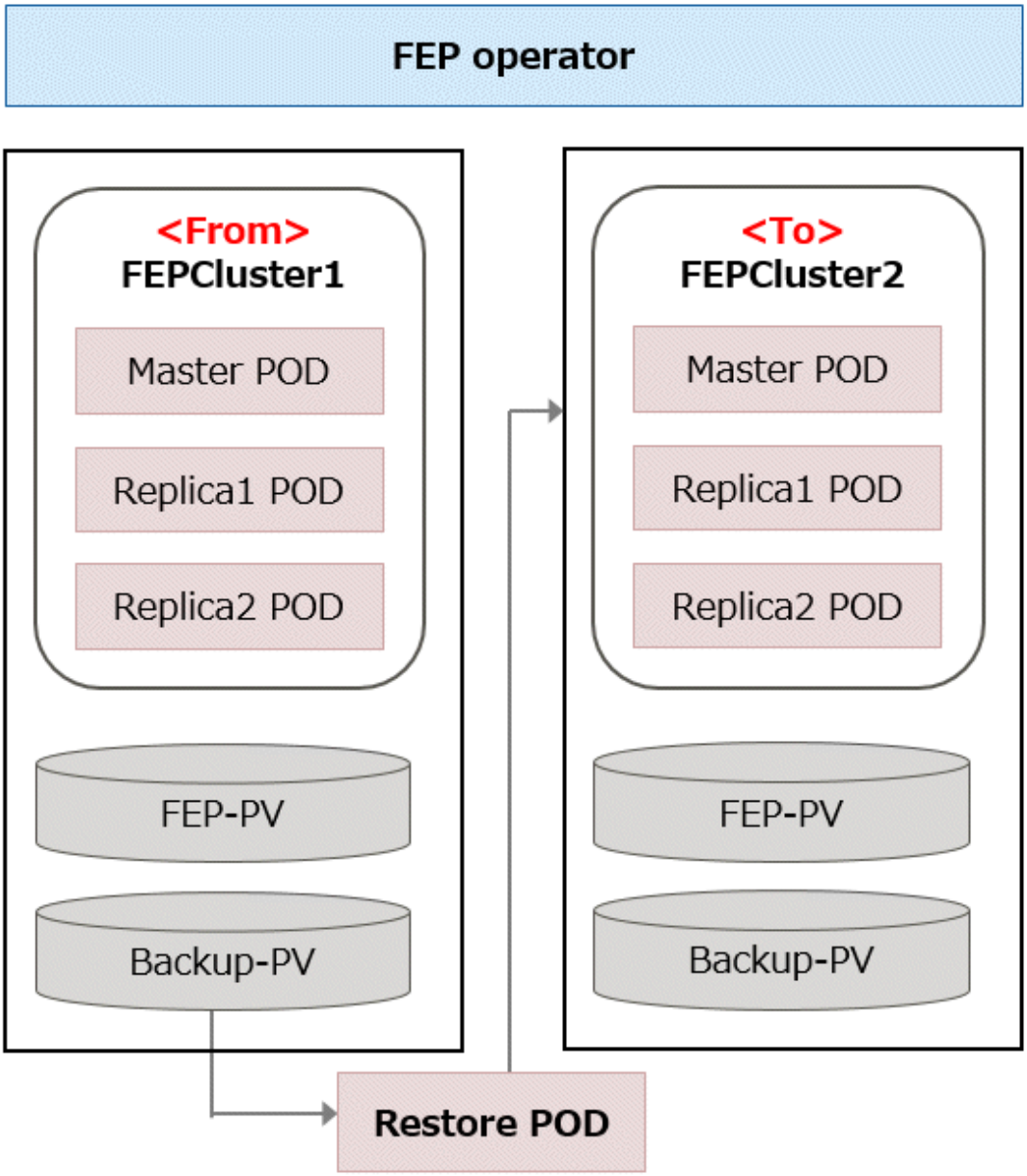
The backup storage location and retention period settings are the same as those for scheduled backups.

2.8 Perform PITR and Latest Backup Restore from Operator

There are two types of restore: one is to restore backup data to an existing FEP cluster, and the other is to create a new FEP cluster and restore backup data.

The former retains the attributes of the FEP cluster, such as IP address and name, while the latter is created from scratch.

The restore process deploys a restore container. The restore container performs the pgBackRest restore operation from the backup data to be restored to the master server of the FEP cluster. After the data is restored to the master server, the FEP cluster is created by synchronizing the data to two replica servers.



2.9 Monitoring & Alert

Monitoring and alerts system leverages standard GAP stack (Grafana, Alert manager, Prometheus) deployed on OCP(OpenShift Container Platform) and Kubernetes. GAP stack must be there before FEP operator & FEPCluster can be deployed.

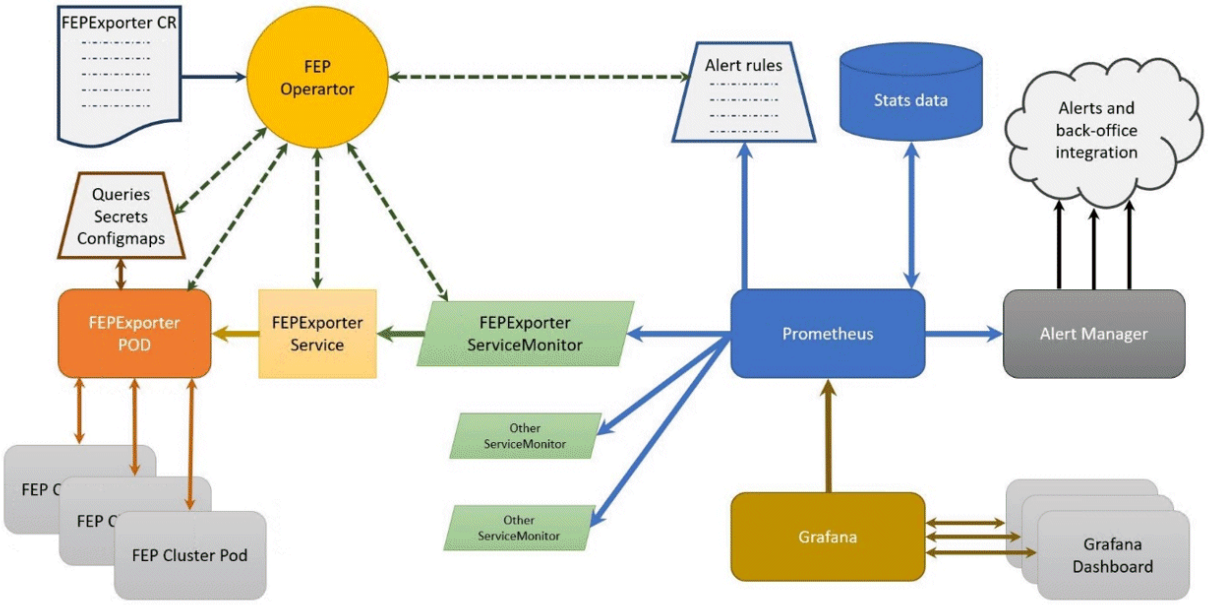
Prometheus is a condensed way to store time-series metrics. Grafana provides a flexible and visually pleasing interface to view graphs and guages of FEP metrics stored in Prometheus.

Together they let store large amounts of metrics that user can slice and break down to see how the FEP database is behaving. They also have a strong community around them to help deal with any usage and setup issues.

The Prometheus acts as storage and a polling consumer for the time-series data of FEP container. Grafana queries Prometheus to displaying informative and very pretty graphs.

If Prometheus rules are defined, it also evaluates rules periodically to fire alerts to Alert manager if conditions are met. Further Alert manager can be integrated with external systems like email, slack, SMS or back-office to take action on alerts raised.

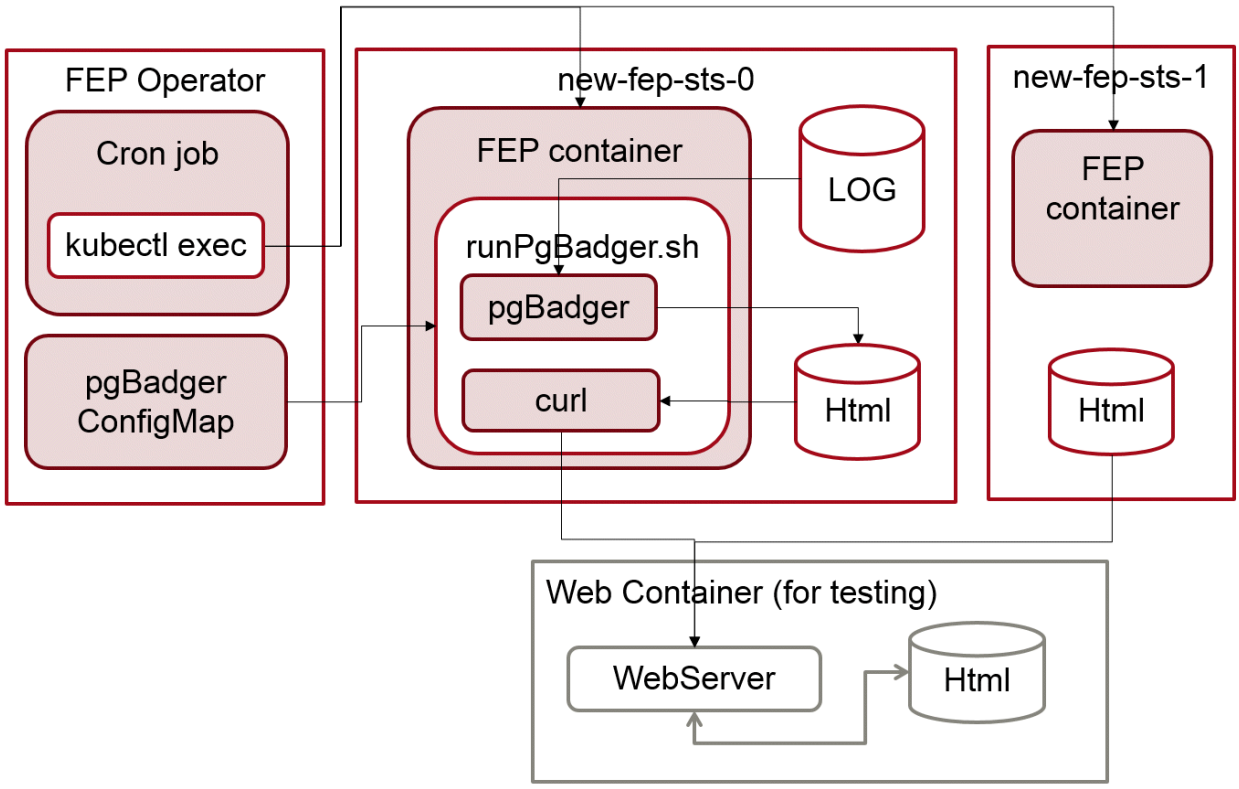
Metrics from FEP Cluster(s) is collected by Prometheus through optional components deployed using FEP Exporter with default set of metrices and corresponding Prometheus rules to raise alerts. User may extend or overwrite metrics by defining their custom metrics queries and define their custom Prometheus rules for alerting.



2.10 Server Log Monitoring

2.10.1 pgBadger Log Monitoring

pgBadger can parse PostgreSQL log files to produce daily and weekly statistical reports from many points of view.

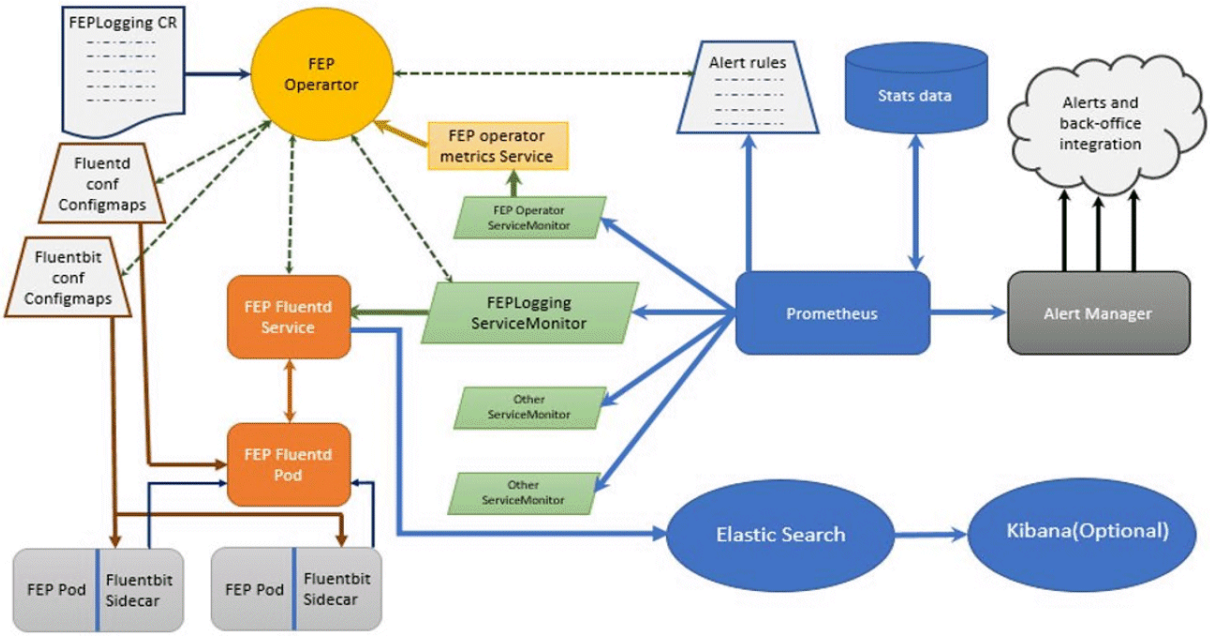


2.10.2 Prometheus and Elasticsearch Log Monitoring

Fluentbit is deployed as a sidecar in the FEPCluster along with patroni and collects postgres database logs. Fluentd is installed with the fluent-plugin-prometheus plugin required for integration with Prometheus and fluent-plugin-elasticsearch plugin required for integration with Elasticsearch.

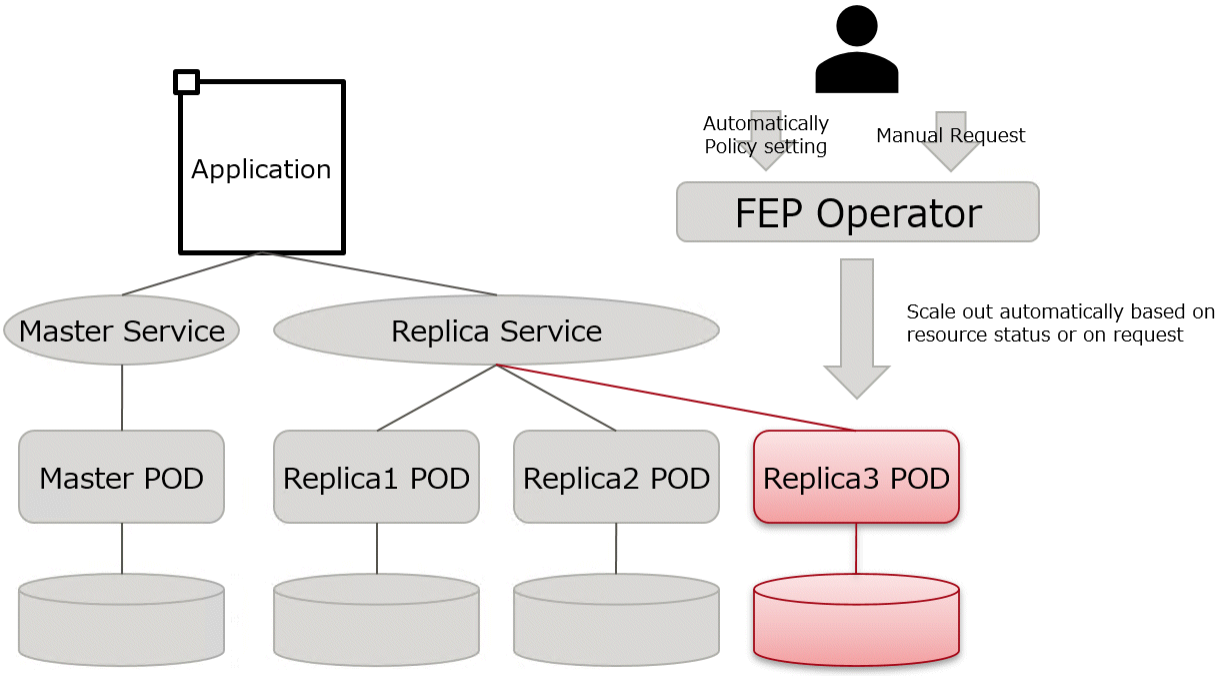
Fluentd counts the occurrence of different severity levels (PANIC, FATAL, ERROR, WARNING, DEBUG etc) by filtering the incoming log records and the counts are passed onto Prometheus along with the logfile name. The log file name makes it easier to investigate the reason of severity/issue.

If elastic search is enabled and configured, then fluentd sends logs to elastic search which can be viewed in kibana.



2.11 Scaling Replicas

The scaling feature creates a replica of the reference replica either automatically or manually by the customer. By querying the reference replica service, the customer will be able to direct the query to the automatically added replica instance.



2.11.1 Using the automatic scale out function

The automatic scale out feature works based on the performance metrics (metrics) of objects in a Kubernetes cluster. Metrics are provided by a service that implements the metrics API, called a metrics server.

There are three types of metrics servers (metrics APIs) in OpenShift/Kubernetes.

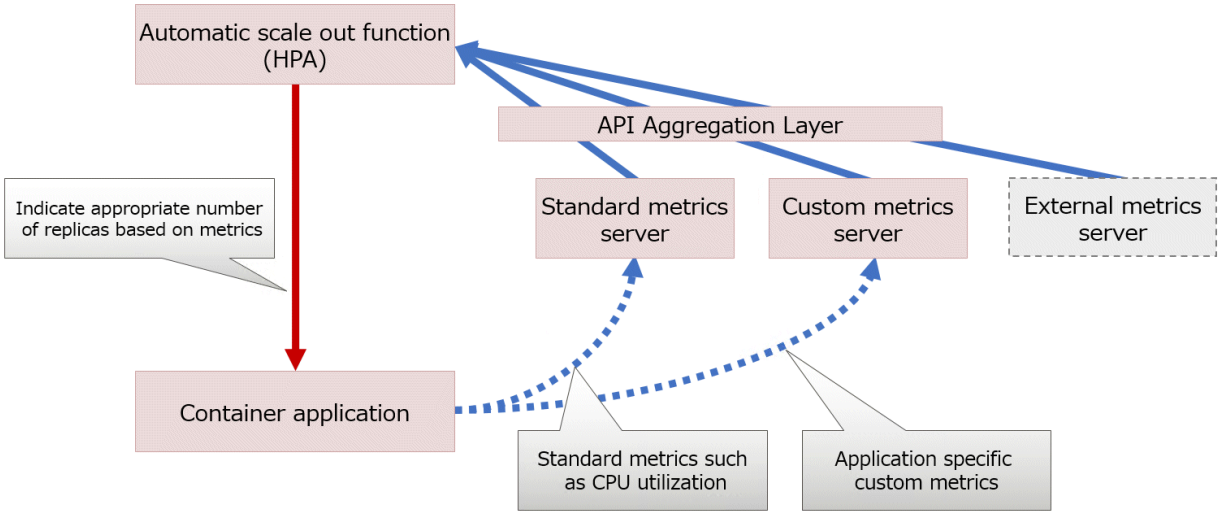
- Standard metrics server
- Custom metrics server
- External metrics server

Automatic scale out based on CPU utilization works with metrics obtained from standard metrics server.
Automatic scale out based on the number of connections works with metrics obtained from custom metrics server.

The custom metrics server looks at the metrics information for the FEP cluster collected by the monitoring function in Prometheus and publishes it in the form of a metrics API.

Custom metrics server are resources shared by OpenShift/Kubernetes clusters, so they are not built and configured for an extended FEP for Kubernetes installation. To take advantage of automatic scale out based on the number of connections, you must have a custom metrics server.

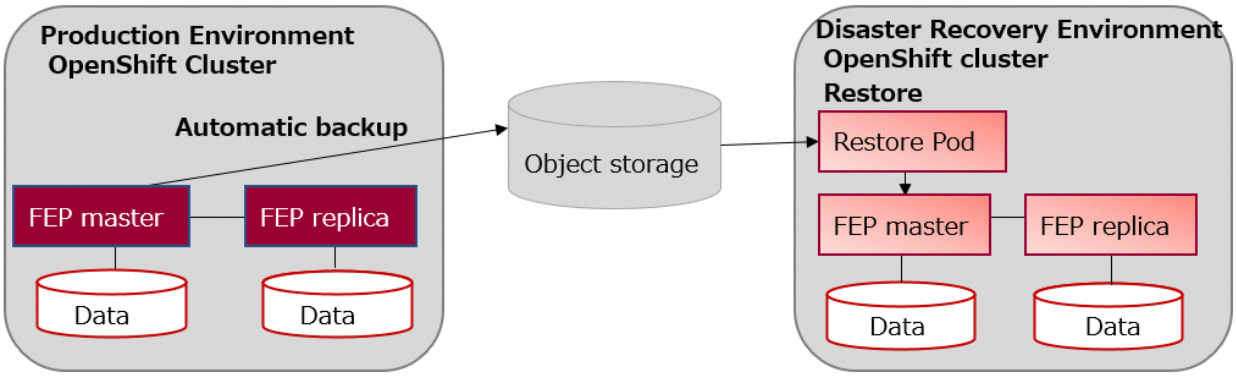
The custom metrics server must also reference Prometheus, which collects custom metrics for FEP clusters, and must be configured to publish custom metrics for FEP clusters.



2.12 Disaster Recovery

Retrieve an automated backup of the production environment to object storage.

Restore the FEP cluster from a backup on object storage on the disaster recovery environment OpenShift cluster.



2.13 Ability to Customize the FEP Server Container Image

Fujitsu Enterprise Postgres for Kubernetes bundles a number of OSS modules to extend the PostgreSQL capabilities and is able to handle most demanding workload. There are occasions where a customer would like to include additional modules to further extend PostgreSQL capability. The general idea is to use the FEP Server container image as a base, compile and add the extra modules to the base image to create a customized image hosted on a private container registry.

2.14 Cloud Secret Store

Fujitsu Enterprise Postgres for Kubernetes enhances security of PostgreSQL by providing some unique features. One such feature is the integration with Cloud Secret Store. Customers can opt to store the database secrets, such as database user password or certificates, in an external secret store such as:

- Azure Key Vault
- AWS Secrets Manager
- GCP Secret Manager and
- HashiCorp Vault

Cloud Secret Store leverages the Secret Store CSI Driver, https://secrets-store-csi-driver.sigs.k8s.io/ and respective provider drivers by Azure, AWS, GCP and HashiCorp to integrate with Fujitsu Enterprise Postgres for Kubernetes. With this integration, one can:

- Manage Postgres username/password on Cloud Secret Store
- Manage SSL Certificate on Cloud Secret Store

Benefits of this integration:

- Passwords and certificates are stored in a centralised Cloud Secret Store instead of locally on each Kubernetes cluster
- Allow automatic password and certificate rotation
- Separation of duties; person who maintain the FEP cluster can be a different person who maintain the passwords and certificates in Cloud Secret Store
- Access to secrets in Cloud Secret Store is controlled by authentication and authorization on the Cloud provider

Appendix A OSS Supported by Fujitsu Enterprise Postgres for Kubernetes

The OSS supported by Fujitsu Enterprise Postgres for Kubernetes is listed below.

OSS name	Version and level	Description	Reference
PostgreSQL	17.0	Database management system	PostgreSQL Documentation
orafce	4.13.4	Oracle-compatible SQL features	"Compatibility with Oracle Databases" in the Fujitsu Enterprise Postgres Application Development Guide
Pgpool-II	4.5.4	Failover, connection pooling, load balancing, etc.	"Pgpool-II" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pg_statsinfo	16.0	Collection and accumulation of statistics	"pg_statsinfo" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pg_hint_plan	17.1.7.0	Tuning (statistics management, query tuning)	- "pg_hint_plan" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server - "Enhanced Query Plan Stability" in the Fujitsu Enterprise Postgres Operation Guide
pg_dbms_stats	14.0		- "pg_dbms_stats" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server - "Enhanced Query Plan Stability" in the Fujitsu Enterprise Postgres Operation Guide
pg_repack	1.5.1	Table reorganization	"pg_repack" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pg_rman	1.3.16	Backup and restore management	"pg_rman" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pgBadger	12.4	Log analysis	"pgBadger" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pg_bigm	1.2-20240606	Full-text search (multibyte)	"pg_bigm" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
ldap2pg	6.1	Mnaging user	- "LDAP Authentication File Settings" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server - "ldap2pg" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Client
PostgreSQL JDBC driver	42.7.4	JDBC driver	"JDBC Driver" in the Fujitsu Enterprise Postgres Application Development Guide
psqlODBC	17.00.0001	ODBC driver	"ODBC Driver" in the Fujitsu Enterprise Postgres Application Development Guide
pgvector	0.7.4	Vector data storage, operation, and search (*1)	"pgvector" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server
pgBackRest	2.53.1	Backup and restore management	"Scheduling Backup from Operator" in the User's Guide

OSS name	Version and level	Description	Reference
patroni	4.0.1	Postgres cluster management	"High Availability" in the User's Guide
postgres_exporter	0.15.0	Postgresql metrics monitoring capabilities for Prometheus with Fujitsu updated queries	"Monitoring" in the User's Guide

*1: It does not provide the ability to generate vector data. If you want to identify relationships and similarities between text, images, audio, and other data, use software or services that generate vector data and store the generated vector data in a database.

FUJITSU

Fujitsu
Enterprise Postgres 17 for Kubernetes

User's Guide

Linux

J2UL-UG17-01ENZ0(00)
November 2024

Preface

Purpose of this document

This document describes system configuration, design, installation, setup, and operational procedures of the Fujitsu Enterprise Postgres for Kubernetes.

Intended readers

This document is intended for people who are:

- Considering installing Fujitsu Enterprise Postgres for Kubernetes
- Using Fujitsu Enterprise Postgres for Kubernetes for the first time
- Wanting to learn about the concept of Fujitsu Enterprise Postgres for Kubernetes
- Wanting to see a functional overview of Fujitsu Enterprise Postgres for Kubernetes

Readers of this document are also assumed to have general knowledge of:

- Linux
- Kubernetes
- Containers
- Operators

Structure of this document

This document is structured as follows:

Chapter 1 System Requirements

Describes the system requirements.

Chapter 2 Overview of Operator Design

Describes an overview of the operator design.

Chapter 3 Operator Installation

Describes the installation of the FEP operator.

Chapter 4 Deployment Container

Describes container deployment.

Chapter 5 Post-Deployment Operations

Describes the operation after deploying the container.

Chapter 6 Maintenance Operations

Describes the maintenance operation after deploying the container.

Chapter 7 Abnormality

Describes the actions to take when an error occurs in the database or an application.

Appendix A Quantitative Values and Limitations

Describes the quantitative values and limitations.

Appendix B Adding Custom Annotations to FEPCluster Pods using Operator

Describes instructions for adding custom annotations to a FEPCluster pod.

Appendix C Utilize Shared Storage

Describes how to build a FEPCluster when using shared storage.

Appendix C Utilize Shared Storage

Describes how to build a FEPCluster when using shared storage.

Appendix D Key Management System Available for Transparent Data Encryption

Describes the key management system available for transparent data encryption.

Appendix E Fluent Bit Integration Using Custom Secret

Describes the Fluent Bit integration using custom secret.

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Fujitsu Enterprise Postgres	FEP
Vertical Clustered Index	VCI
Transparent Data Encryption	TDE
Point in time recovery	PITR
Custom Resource	CR
Custom Resource Definition	CRD
Persistent Volume	PV
Persistent Volume Claim	PVC
Universal Base Image	UBI
OpenShift Container Platform	OCP
Mutual TLS	MTLS

Abbreviations of manual titles

The following abbreviations are used in this manual as manual titles:

Full Manual Title	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Release Notes	Release Notes
Fujitsu Enterprise Postgres for Kubernetes Overview	Overview
Fujitsu Enterprise Postgres for Kubernetes User's Guide	User's Guide
Fujitsu Enterprise Postgres for Kubernetes Reference	Reference

Trademarks

- Linux is a registered trademark or trademark of Mr. Linus Torvalds in the U.S. and other countries.
- Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.
- S/390 is a registered trademark of International Business Machines Corporation in the United States or other countries or both.

Other product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

```
Edition 1.0: November 2024
```

Copyright

Copyright 2021-2024 Fujitsu Limited

Contents

Chapter 1 System Requirements 1
1.1 Components Embedded 1
1.2 CPU 1
1.3 Supported Platform 1
1.4 Collaboration Tool 2

Chapter 2 Overview of Operator Design 3
2.1 Design Task 3
2.2 System Configuration Design 3
2.2.1 Server Configuration 3
2.2.2 User Account 5
2.2.3 Basic Information of the Container 5
2.3 Design Perspective for Each Feature 10
2.3.1 Deployment 11
2.3.2 High Availability 11
2.3.3 Configurable Volume per Cluster 11
2.3.3.1 Disk Space Management 13
2.3.3.1.1 Increasing Disk Space 13
2.3.3.1.2 Reducing Disk Usage 14
2.3.3.2 Configuring PVC Auto Expansion 14
2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator 15
2.3.5 Scheduling Backup from Operator 15
2.3.5.1 Important Setting Items 16
2.3.5.2 Parameters that cannot be Set 16
2.3.5.3 Restricted Parameters 19
2.3.5.4 About Sections in the Config File 19
2.3.6 Perform PITR and Latest Backup Restore from Operator 19
2.3.7 FEP Unique Feature Enabled by Default 20
2.3.8 Monitoring & Alert (FEPExporter) 20
2.3.8.1 FEPExporter Custom Resource 21
2.3.8.2 Change to FEPCluster CR - metrics user 21
2.3.8.3 FEPExporter CR auto-create for FEPCluster 21
2.3.9 Collaboration with Amazon CloudWatch 21
2.3.10 Scaling Replicas 21
2.3.10.1 Change to FEPCluster CR - auto scale out 22
2.3.11 Disaster Recovery 23
2.3.12 Transparent Data Encryption Using a Key Management System 23
2.3.13 Database Role Management 23
2.3.13.1 Creating Roles Related to Database Operation 24
2.3.13.1.1 Quarantine SUPERUSER 24
2.3.13.1.2 Database Administrator Role 24
2.3.13.1.3 Confidential Administrator Role 25
2.3.13.2 Expiration Management of Database Roles with Login Privileges 25
2.3.13.2.1 Policy-based Password Operation 25
2.3.13.2.2 Password Operation with the VALID UNTIL Clause 26
2.3.14 User Synchronization Using ldap2pg 27
2.3.14.1 How to Update FEPCluster CR for ldap2pg 28
2.3.15 Fixed Statistics 33

Chapter 3 Operator Installation 34
3.1 Using the OperatorHub 34
3.1.1 Pre-requisite 34
3.1.2 Deploying Operator 35
3.1.3 Upgrading Operators 37
3.2 Using the Helm Chart 37
3.2.1 Deploying Operator 37

3.2.2 Upgrading Operators....37
3.3 Using the Rancher UI....38
3.3.1 Pre-requisite....38
3.3.2 Register Helm Chart Repository....39
3.3.3 Deploying Operator....41
3.3.4 Upgrading Operators....42
3.4 Implement Collaborative Monitoring Tools....43
3.4.1 Implement GAP Stack....43
3.4.2 Implement Elastic Cloud on Kubernetes....43
3.4.2.1 Deploy ECK Operator....43
3.4.2.2 Deploy Elasticsearch Cluster....45
3.4.2.3 Deploy Enterprise Search....46
3.4.2.4 Deploy Kibana....46
3.4.2.5 Expose Kibana using OpenShift Route....47
3.4.2.6 Login to Kibana....49
3.5 Implement Client....50

Chapter 4 Deployment Container....51
4.1 Deploying FEPCluster using Operator....51
4.2 Deploy a Highly Available FEPCluster....56
4.3 Deploying FEPExporter....57
4.4 FEPExporter in Standalone Mode....59
4.5 How to Collaborate with CloudWatch....61
4.5.1 Preparation....61
4.5.2 FEPCluster Configuration....62
4.5.3 Forwarding Custom Metrics....62
4.6 Deploying FEPClusters with Cloud-based Secret Management....63
4.6.1 Installing Secret Store CSI Driver Using Helm Charts....63
4.6.2 Installing and Configuring Azure Provider for Secret Store CSI Driver....64
4.6.2.1 Install Azure Provider drivers using helm chart....64
4.6.2.2 Create Secret to Access Azure Key vault....64
4.6.2.3 Store Secret in Azure Key Vault....64
4.6.2.4 Store Certificate in Azure Key Vault....65
4.6.3 Installing and Configuring AWS Provider for Secret Store CSI Driver....66
4.6.3.1 Install AWS Provider drivers using helm chart....66
4.6.3.2 Setup EKS cluster along with service account with necessary IAM roles and permission to access Secret Manager....66
4.6.3.3 Store Secret in AWS Secrets Manager....67
4.6.3.4 Store Cert in AWS Secrets Manager....67
4.6.4 Installing GCP Provider for Secret Store CSI Driver....67
4.6.4.1 Install GCP Provider drivers using Kubernetes....67
4.6.4.2 Configure GCP secret manager and IAM....67
4.6.4.3 Create Secret to access GCP Secret manager....68
4.6.4.4 Store secret in GCP Secret manager....68
4.6.4.5 Store Cert in GCP Secret manager....68
4.6.5 Installing HashiCorp Vault Provider for Secret Store CSI Driver....68
4.6.5.1 Install HashiCorp Provider drivers using helm chart....68
4.6.5.2 Configure Kubernetes Authentication for HashiCorp Vault....68
4.6.5.3 Store Secret in HashiCorp Vault....68
4.6.5.4 Store Cert in HashiCorp Vault....69
4.6.5.5 Create policy and role to access the secrets from HashiCorp Vault....69
4.6.6 Configuring FEPCluster to use Provider for Secret Store Driver....69
4.6.6.1 Azure Provider for Secret Store CSI Driver....69
4.6.6.2 AWS Provider for Secret Store CSI Driver....70
4.6.6.3 GCP Provider for Secret Store CSI Driver....71
4.6.6.4 HashiCorp Vault Provider for Secret Store CSI Driver....72
4.7 Deploying a customized FEP server container image....73
4.7.1 Requirements....73

4.7.2 Build custom FEP image with extension....73
4.7.3 Adding SQLite Foreign Data Wrapper to FEP Server Container....73
4.7.4 Create FEP Cluster with custom image....74
4.8 Configuration FEP to Perform MTLS....75
4.8.1 When Using an Automatically Generated Certificate....75
4.8.1.1 How to Create a Certificate....75
4.8.1.2 How to Create a Client Certificate....75
4.8.2 When Using Your Own Certificate....76
4.8.2.1 Manual Certificate Management....76
4.8.2.2 Automatic Certificate Management....80
4.8.2.3 Deploy FEPCluster with MTLS support....85
4.8.2.4 Configurable Parameters....91
4.9 Replication Slots....94
4.9.1 Setting Up Logical Replication using MTLS....94
4.10 FEP Logging....97
4.10.1 FEPLogging Configuration....97
4.10.1.1 FEPLogging Custom Resources - spec....97
4.10.1.1.1 Define fepLogging image....99
4.10.1.1.2 Define fepLogging mcSpec....99
4.10.1.1.3 Define fepLogging restartRequired....100
4.10.1.1.4 Define fepLogging scrapeInterval and scrapeTimeout....100
4.10.1.1.5 Define fepLogging elastic....100
4.10.1.1.6 Define authSecret for elastic....101
4.10.1.1.7 Define fepLogging TLS....101
4.10.1.1.8 Define Prometheus TLS....101
4.10.2 Configuring FEPCluster Remote Logging....102
4.10.2.1 FEP Custom Resources - spec.fep.remoteLogging....102
4.10.2.1.1 Define remoteLogging enable and fluentdName....103
4.10.2.1.2 Define remoteLogging tls....104
4.10.2.1.3 Define remoteLogging image....104
4.10.2.1.4 Define remoteLogging fluentbitConfigSecretRef....104
4.10.2.1.5 Define remoteLogging awsCredentialSecretRef....105
4.10.3 FEPLogging Operations....105
4.10.3.1 Log Forwarding to Elasticsearch....105
4.10.3.2 Log severity based Alarms/Metrics....105
4.10.3.3 Forwarding auditlog to Elasticsearch....106
4.10.4 Limitations....106
4.11 Configuring pgBadger....107
4.11.1 FEP Custom Resources - spec.fep.pgBadger....107
4.11.2 Define pgBadger Schedules....107
4.11.3 Define pgBadger Options....107
4.11.4 Define Endpoint for Uploading Report....107
4.11.5 Uploaded File on Web Server....110
4.12 Automating Audit Log Operations....110
4.12.1 Simplifies Parameter Setting....111
4.12.2 Alerting....111
4.12.3 Store in Cloud Storage....112
4.13 Transparent Data Encryption Using a Key Management System....112
4.13.1 Registration of Authentication Information....113
4.13.1.1 When Using a KMIP Server....113
4.13.1.2 When Using AWS Key Management Service....113
4.13.1.3 When using Azure Key Management Service....113
4.13.2 Configuring FEPCluster Custom Resources....114
4.13.2.1 Define spec.fepChildCrVal.customPgParams....114
4.13.2.2 Define spec.fepChildCrVal.sysTde....114
4.14 Disaster Recovery in Hot Standby Configuration....116
4.14.1 Continuous Recovery Method....116

4.14.2 Streaming Replication Method ... 116
4.14.3 Defining a Hot Standby Configuration ... 117
4.14.3.1 Defining a Continuous Recovery Method ... 117
4.14.3.2 Defining a Streaming Replication Method ... 117
4.14.3.3 Defining FEPCluster Custom Resources ... 118
4.15 Enabling Client Authentication with scram-sha-256 Authentication ... 118
4.15.1 Enabling Client Authentication Using scram-sha-256 Authentication in the FEP Server Container ... 118
4.15.1.1 Define spec.fepChildCrVal.customPgParams ... 119
4.15.1.2 Define spec.fepChildCrVal.customPgHba ... 119
4.15.2 Enabling Client Authentication Using scram-sha-256 Authentication in the FEPpgpool2 Container ... 119
4.15.2.1 Creating the Resources Required to Enable scram-sha-256 Authentication ... 119
4.15.2.2 Editing FEPPgpool2 Custom Resources ... 120
4.15.3 Enabling Client Authentication Using scram-sha-256 Authentication on Existing FEP Server and FEPpgpool2 Containers 121
4.16 Backing Up Statistics ... 121

Chapter 5 Post-Deployment Operations ... 123
5.1 How to Connect to a FEP Cluster ... 123
5.2 Configuration Change ... 124
5.3 FEPCluster Resource Change ... 125
5.3.1 Changing CPU and Memory Allocation Resources ... 125
5.3.2 Resizing PVCs ... 125
5.4 FEPPGPool2 Configuration Change ... 125
5.5 Scheduling Backup from Operator ... 128
5.6 Configure MTLS Setting ... 129
5.6.1 Certification Rotation ... 129
5.7 Monitoring ... 129
5.7.1 Monitoring FEP Operator and Operands ... 130
5.7.2 Monitoring FEP Server ... 130
5.7.2.1 Architecture ... 130
5.7.2.2 Default Server Metrics Monitoring ... 131
5.7.2.3 Default Alerts ... 133
5.7.2.4 Graphical user interface ... 134
5.7.2.5 Metrics Collected by CloudWatch ... 135
5.7.3 Monitoring FEP Backup ... 135
5.7.3.1 pgbackrest_info_backup view ... 136
5.7.4 Monitoring FEP PGPool2 ... 136
5.7.4.1 pgpool2_stat_load_balance view ... 136
5.7.4.2 pgpool2_stat_conn_pool view ... 137
5.7.4.3 pgpool2_stat_sql_command view ... 137
5.8 Event Notification ... 138
5.8.1 Events raised ... 138
5.8.2 Events that Occur when Custom Resources are Updated ... 139
5.8.3 Viewing the Custom Events ... 139
5.9 Scaling Replicas ... 139
5.9.1 Automatic Scale Out ... 139
5.9.2 Manual Scale In/Out ... 140
5.10 Backing Up to Object Storage ... 140
5.10.1 Pre-creation of Resources ... 140
5.10.1.1 Storing CA Files (Root Certificates) ... 140
5.10.1.2 Storing Repository Key ... 140
5.10.2 Defining a FEPCluster Custom Resource ... 140
5.11 Disaster Recovery ... 142
5.11.1 Disaster Recovery by Backup/Restore Method ... 142
5.11.1.1 Disaster Recovery Prerequisites ... 142
5.11.1.2 Performing Disaster Recovery ... 142
5.11.1.2.1 Pre-creation of Resources ... 142
5.11.1.2.2 Defining a FEPCluster Custom Resource ... 142

5.11.2 Disaster Recovery with Continuous Recovery Method....144
5.11.2.1 Disaster Recovery Prerequisites.... 144
5.11.2.2 Performing Disaster Recovery....144
5.11.3 Disaster Recovery Using Velero....145
5.11.3.1 Disaster Recovery Prerequisites.... 145
5.11.3.2 Performing Disaster Recovery....145
5.11.3.2.1 Configuring FEPCluster Custom Resources....145
5.11.3.2.2 Velero Backup.... 147
5.11.3.2.3 Database Backup....148
5.11.3.2.4 Velero Restore.... 148
5.11.3.2.5 Return from Disaster Recovery Environment to Production Environment....148
5.11.4 Disaster Recovery with Streaming Replication Method....148
5.11.4.1 Disaster Recovery Prerequisites.... 148
5.11.4.2 Performing Disaster Recovery....148
5.11.5 Parameter Change in Disaster Recovery Environment....148
5.12 Operation of Transparent Data Encryption Using Key Management System....149
5.12.1 Updating Custom Resource Parameters.... 149
5.12.2 Update Credentials....149
5.12.3 Encrypting a Tablespace....149
5.12.4 Backup/Restore....150
5.12.5 Changing Key Management System Definitions....150
5.13 Confidentiality Management Feature.... 151
5.13.1 Enabling Confidentiality Management Feature....151
5.13.2 Monitoring Confidentiality Management Feature.... 151
5.14 Operation of Fixed Statistics....151
5.14.1 Preparing for Object Storage Connections.... 151
5.14.1.1 Using S3....151
5.14.1.2 Using AzureBlob.... 152
5.14.1.3 Using Google Cloud Storage.... 152
5.14.2 Storing validation environment statistics in object storage.... 152
5.14.3 Schedule for Fixed Statistics....152
5.14.4 Using Local Storage....153
5.14.4.1 Exporting Statistics Files to FEPCluster in a Validation Environment....153
5.14.4.2 Deploying Statistics Files to FEPCluster in a Production Environment.... 153
5.14.4.3 Specify File Names for FEPCluster Custom Resources....154
5.15 Scheduling of an Aggressive Freeze for Tuples (VACUUM FREEZE).... 154
5.16 pgvector Setup.... 155

Chapter 6 Maintenance Operations....156
6.1 Minor Version Upgrade.... 156
6.2 Cluster Master Switchover....156
6.3 Perform PITR and the Latest Backup Restore from Operator....156
6.3.1 Setting Item....157
6.3.2 After Restore....157
6.4 Major Version Upgrade.... 157
6.4.1 Pre-work on the Data Source FEP Cluster....157
6.4.2 Operator Upgrade.... 158
6.4.2.1 Uninstalling the Old Operator....158
6.4.2.2 Installing a New Version of the Operator....158
6.4.3 Major Version Upgrade of FEP.... 158
6.4.3.1 Creating a New FEPCluster CR....158
6.4.3.2 Verifying FEP Major Upgrade Complete....161
6.4.4 Updating Each Custom Resource.... 161
6.4.4.1 Removing a FEPClusterCR for a Data Source.... 161
6.4.4.2 FEPPgpool2.... 161
6.4.4.3 FEPExporter Built in Standalone Mode.... 162
6.5 Assigned Resources for Operator Containers....162

6.5.1 How to Change Assigned Resources....162
6.5.1.1 When installing using OperatorHub....162
6.5.1.2 When installing using Helm Chart or RancherUI....163
6.6 Using SUPERUSER Privilege....163
6.6.1 CREATE EXTENSION....163
6.6.2 Change Password of SUPERUSER....164
6.6.3 Using SUPERUSER....164

Chapter 7 Abnormality....165
7.1 Handling of Data Abnormalities....165
7.2 Handling when the Capacity of the Data Storage Destination or Transaction Log Storage Destination is Insufficient....165
7.3 What to do when the Capacity of the Backup Data Storage Area is Insufficient....165
7.4 Handling Access Abnormalities When Instance Shutdown Fails....165
7.5 Collection of Failure Investigation Information....165
7.6 Log Collection Tool....166
7.6.1 Logs Collected....167
7.6.2 Location and Dependencies....167
7.6.3 User Interface....168
7.6.3.1 Command-Line Syntax....168
7.6.4 Log Collection Directory....169
7.6.4.1 Command-Line Specification....169
7.6.4.2 Directory Structure....169

Appendix A Quantitative Values and Limitations....170
A.1 Quantitative Values....170
A.2 Limitations....170

Appendix B Adding Custom Annotations to FEPCluster Pods using Operator....171

Appendix C Utilize Shared Storage....173
C.1 Creating a StorageClass....173
C.2 Creating a PersistentVolume....173
C.3 Creating FEPCluster....174

Appendix D Key Management System Available for Transparent Data Encryption....175
D.1 KMIP Server....175
D.2 AWS Key Management Service....175
D.2.1 Available Services....175
D.2.2 Available AWS KMS Keys....175
D.2.3 Required Privileges....175
D.2.4 Key ID....175
D.3 Azure Key Management Service....175
D.3.1 Available Services....175
D.3.2 Available Keys....176
D.3.3 Available Algorithms....176
D.3.4 Key Operation....176
D.3.5 Key ID....176
D.3.6 Sign In....176

Appendix E Fluent Bit Integration Using Custom Secret....177
E.1 Create Custom Secret....177
E.1.1 Sample fluent-bit.yaml template....177
E.1.2 Sample parsers.conf template....178
E.1.3 Encode the fluent-bit.yaml content....179
E.1.4 Encode the parsers.conf content....179
E.1.5 Create the custom secret....179
E.2 Reference to the secret in FEPClusterCR....179
E.2.1 Reload Fluent Bit Configuration....180
E.2.2 Confirm Log Ingestion in Azure Blob....180

E.2.3 Confirm Log Ingestion in Elasticsearch....180
E.3 Fluent Bit configuration for Prometheus exporter....181
E.3.1 Create a Service....181
E.3.2 Create a ServiceMonitor....181
E.3.3 Confirm Log Ingestion in Prometheus....182
E.4 Fluent Bit configuration for AWS CloudWatch....182
E.4.1 Create AWS credentials Secret....182
E.4.2 Encode the config file content....183
E.4.3 Encode the credentials content....183
E.4.4 Create a secret using base64 encoded content....183
E.4.5 Reference the Secrets in FEPClusterCR....184

Chapter 1 System Requirements

This chapter describes the system requirements.

1.1 Components Embedded

The FEP Server container embeds following components. However it is understood that these components are bound to be upgraded in the maintenance phase.

No	Component	Version	Description
1	Red Hat UBI minimal	9	Meant to provide base OS image for the container
2	Fujitsu Enterprise Postgres Server	17.0	To provide server capabilities
3	Patroni	4.0.1	To provide HA capabilities and other management to the Cluster

1.2 CPU

It should be noted that it provides supports to both the following CPU Architectures to meet the scope of work.

No	CPU architecture
1	x86
2	s390x
3	ppc64le

1.3 Supported Platform

It supports running on the following platforms.

No	Platform		Version
1	OpenShift Container Platform		4.14, 4.15, 4.16
2	Rancher Kubernetes Engine (on Linux hosts) (*1)		1.4.0+
3	Vmware Tanzu Kubernetes Grid (*1)		1.6+
4	Full Managed Kubernetes Service	- Azure Kubernetes Service - Amazon Elastic Kubernetes Service - Alibaba Cloud Container Service for Kubernetes - Google Kubernetes Engine - IBM Cloud Kubernetes Service	1.27, 1.28, 1.29

*1: Kubernetes 1.27 - 1.29

Supports storage supported by OpenShift or Kubernetes (AKS, EKS, RKE, ACK, GKE, IKS and TKG).

However, you need shared storage, like NFS, or object storage for backup and archive WAL volumes. Object storage supports Amazon Simple Storage Service, Azure Blob Storage, and Google Cloud Storage.

1.4 Collaboration Tool

Supports integration with the following tools.

No	Tool	Version	How to obtain
1	Prometheus	- OpenShift The version installed OpenShift - Kubernetes - Prometheus v2.36.2 and later - AlertManager v0.24.0 and later - Rancher The version provided by Rancher Monitoring Chart	- OpenShift Preinstalled with OpenShift - Kubernetes prometheus-operator (v0.61.1 and later) https://github.com/prometheus-operator/prometheus-operator/prometheus-operator - Rancher Using the Rancher Monitoring Chart
2	AlertManager		
3	Grafana	- OpenShift and Kubernetes Grafana v7.5.17 and later - Rancher The version provided by Rancher Monitoring Chart	- OpenShift Provided by OperatorHub - Kubernetes grafana-operator (v4.7.1 and later) https://github.com/grafana-operator/grafana-operator - Rancher Using the Rancher Monitoring Chart
4	Helm	3.10.0 and later	- Kubernetes only Helm Web Site https://helm.sh/docs/intro/install/
5	Rancher	v2.7 and later	Rancher Web Site https://rancher.com/
6	Prometheus Adapter	- OpenShift and Kubernetes Confirmed the operation with v0.9.1 and later - Rancher The version provided by Rancher Monitoring Chart	- OpenShift and Kubernetes Prometheus Adapter https://github.com/kubernetes-sigs/prometheus-adapter - Rancher Using the Rancher Monitoring Chart
7	Elastic Search	8.5.2 and later	- OpenShift Provided by OperatorHub - Kubernetes https://github.com/elastic/helm-charts/tree/main/elasticsearch
8	Velero	v1.11 and later	Refer to the documentation published below. https://velero.io/docs/main/basic-install/

Chapter 2 Overview of Operator Design

This chapter describes an overview of the operator design.

2.1 Design Task

Installation/operation using an operator and necessity of design are shown below.

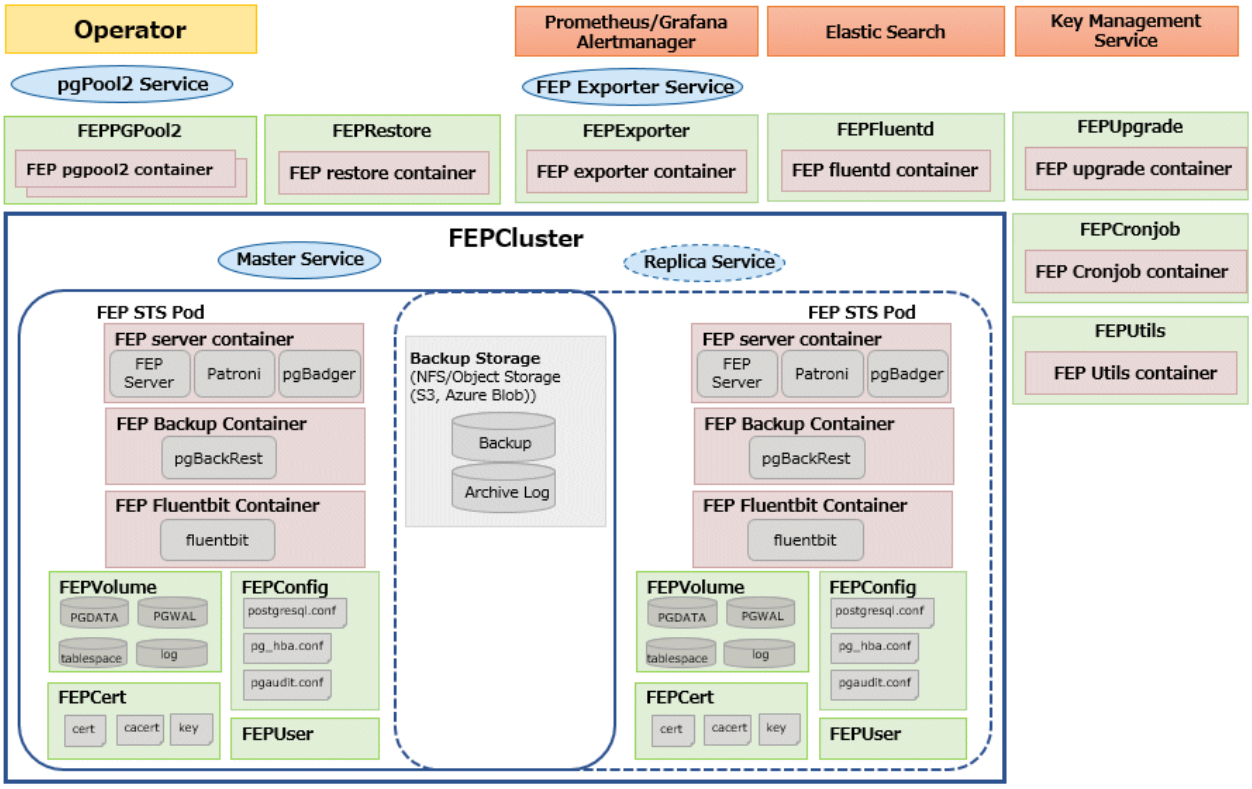
Task	Design required to operate FEP	Where to find
FEP setup	Required.	2.3.1 Deployment
High availability configuration	Recommended. (When checking or changing the behavior of high availability. However, even by default, constant high availability operation is possible.)	2.3.2 High Availability
Volume settings	Recommended. (When setting the volume. However, even by default, allocate a fixed volume.)	2.3.3 Configurable Volume per Cluster
Pgpool-II setup	Recommended. (When using Pgpool-II.)	2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator
Backup/restore settings	Recommended. (When using a backup and restore.)	2.3.5 Scheduling Backup from Operator 2.3.6 Perform PITR and Latest Backup Restore from Operator
Monitoring & Alert(FEPExporter)	Recommended. (When using Monitoring and Alert)	2.3.8 Monitoring & Alert (FEPExporter)
Scaling Replicas	Recommended (When using scaling feature)	2.3.10 Scaling Replicas
Key management system	Recommended. (When the key management system manages the master encryption key for transparent data encryption)	2.3.12 Transparent Data Encryption Using a Key Management System

2.2 System Configuration Design

This section describes the system configuration.

2.2.1 Server Configuration

The following is an overview diagram of the server configuration:



System component

Describes various system resources.

Configuration server type	Description
FEP operator	A container that accepts user requests and is responsible for automating database construction and operational operations.
FEP server container	A container for the FEP server.
FEP backup container	A container that performs scheduled backup operations. Created on the same Pod as the FEP server container.
FEP Fluentbit container	A container that collect FEP database CSV log and forward to fluentd container for processing.
FEP pgpool2 container	A container that uses Pgpool-II to provide load balancing and connection pooling. If you do not use it, you do not need to create it.
FEP restore container	A container that performs the restore operation. Temporarily created during a restore operation.
FEP Exporter container	A container that exposes http/https endpoint for monitoring stats scraping.
FEP Fluentd container	A container that summarise FEP log severity as metrics for Prometheus to consume. Optionally, forward log entries to Elasticsearch for detailed log analysis.
FEP upgrade container	A container that executes the major version upgrade process of the server container. A container created temporarily during the upgrade process.
FEP Cronjob container	A container that is started when the regular processing of each feature of the operator is executed.
FEP Utils container	A container for cloud-based management.
Backup storage	Storage where backup data is stored. If you do not need to obtain a backup, you do not need to create one.

Configuration server type	Description
FEPCluster	Parent CR for FEP Cluster definition and configuration.
FEPBackup	Child CR for backup configuration.
FEPVolume	Child CR for volumes.
FEPConfig	Child CR for FEP configurations.
FEPCert	Child CR for system certificates.
FEPUser	Child CR for database users.
FEPAction	CR for performing actions.
FEPExporter	CR for monitoring configuration.
FEPUpgrade	CR for major upgrade.
Master service	A service to connect to the master FEP server.
Replica service	A service to connect to the replica FEP server.
Pgpool2 service	A service for connecting to Pgpool-II.
Fepexporter service	A service to scrape metrices from all FEPCluster nodes.

2.2.2 User Account

The user accounts used by this product are as follows.

It is possible to improve security by clearly separating and managing the accounts that operate the operators for each role by the infrastructure administrator. It is also possible to manage multiple tenants on a single container management platform.

User type	User name	Description
Infrastructure administrator	Mandatory	A system administrator (superuser) who manages the container management platform that installs operators.
Database administrator	Mandatory	An administrator who performs setup, regular operation, and maintenance operations.
Confidential administrator	Mandatory	An administrator who sets appropriate privileges for each database resource for database users.
Application developer	Mandatory	Develops and executes database applications.

2.2.3 Basic Information of the Container

This section describes the basic information of the container.

FEP server container

The naming convention for the FEP server container is as below.

fujitsu-enterprise-postgres-17-server:*OS*-*FEPBaseVersion*-*MajorVersion*.*MinorVersion*-*ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images).

- fujitsu-enterprise-postgres-17-server:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-server:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-server:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-server:ubi9-17-1.0-ppc64le

FEP backup container

Use the same naming convention for FEP backup containers as for FEP server containers.

fujitsu-enterprise-postgres-17-backup:*OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-17-backup:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-backup:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-backup:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-backup:ubi9-17-1.0-ppc64le

FEP restore container

Use the same naming convention for FEP restore containers as for FEP server containers.

fujitsu-enterprise-postgres-17-restore:*OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-17-restore:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-restore:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-restore:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-restore:ubi9-17-1.0-ppc64le

FEP pgpool2 container

Use the same naming convention for FEP pgpool2 containers as for FEP server containers.

fujitsu-enterprise-postgres-17-pgpool2: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-17-pgpool2:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-pgpool2:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-pgpool2:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-pgpool2:ubi9-17-1.0-ppc64le

FEP Exporter container

FEP Exporter container as for FEP server containers.

fujitsu-enterprise-postgres-exporter: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-exporter:ubi9-17-1.0
  - fujitsu-enterprise-postgres-exporter:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-exporter:ubi9-17-1.0-s390x

- fujitsu-enterprise-postgres-exporter:ubi9-17-1.0-ppc64le

FEP Fluentd container

FEP Fluentd container as for FEP server containers.

fujitsu-enterprise-postgres-fluentd: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-fluentd:ubi9-17-1.0
  - fujitsu-enterprise-postgres-fluentd:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-fluentd:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-fluentd:ubi9-17-1.0-ppc64le

FEP Fluentbit container

FEP Fluentbit container as for FEP server containers.

fujitsu-enterprise-postgres-fluentbit: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0
  - fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0-ppc64le

FEP Cronjob container

FEP Cronjob container as for FEP server containers.

fujitsu-enterprise-postgres-cronjob: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-cronjob:ubi9-17-1.0
  - fujitsu-enterprise-postgres-cronjob:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-cronjob:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-cronjob:ubi9-17-1.0-ppc64le

FEP upgrade container

FEP upgrade container as for FEP server containers.

fujitsu-enterprise-postgres-17-upgrade: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-17-upgrade:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-upgrade:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-upgrade:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-upgrade:ubi9-17-1.0-ppc64le

FEP Utils container

FEP Utils container as for FEP server containers.

fujitsu-enterprise-postgres-17-utils: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi9	
FEPBaseVersion	17	
MajorVersion	1,2, ...	To be used when major change in image, including server patch application
MinorVersion	0,1,2 ...	To be used when minor changes in image, e.g bug fix in container script

The first publishing will expect following names / tagging (Manifest and Child images)

- fujitsu-enterprise-postgres-17-utils:ubi9-17-1.0
  - fujitsu-enterprise-postgres-17-utils:ubi9-17-1.0-amd64
  - fujitsu-enterprise-postgres-17-utils:ubi9-17-1.0-s390x
  - fujitsu-enterprise-postgres-17-utils:ubi9-17-1.0-ppc64le

2.3 Design Perspective for Each Feature

This section describes the design of each feature.

postgresql-cfg format

A postgresql-cfg represent ConfigMap for containing postgresql parameters. The file is used to contain the parameters which need to be reflected in postgresql.conf of the instance. Since patroni ignores all parameters which are not known by OSS postgresql.conf, an approach is defined to treat FEP Parameters in a special way.

The content of the ConfigMap is defined by key=value format. The following table shows the detail:

Spec	Example	Comment
The content may have multiple key/ value pairs	foo=bar foo1=bar1	-
The value cannot have space unless quoted.	foo=bar bar2	Invalid
The quoted value cannot have another value after	foo='bar bar2' something	Invalid
The key value pair must have a '=' sign	-	-
White spaces are allowed before/ after/between the key value pair	foo = bar	-

Any content after '#' will be ignored	# this is a comment
	foo=bar #this is a comment
		-

The value may be quoted by single quotes	foo='bar bar2'	-
Single quote can be escaped by two single quotes	foo='It''s ok'	Note: single quotes are not supported by Patroni edit-config command

Spec	Example	Comment
Backslash '\' will be replaced by '\\' when invoking patronictl edit-config command	-	To avoid command line escape
When a key value pair is invalid, it will be ignored. the update continue to process next pair	foobar foo2=bar2	The 'foobar' will be ignored
The container script does not validate the key and value as long as they are in correct format.	-	-

It is recommended to use the psql's show command to verify parameter is setting correctly.

2.3.1 Deployment

Information for the FEPCluster

Equivalent Kubernetes command: kubectl apply -f FEPClusterCR.yaml

This operation will create a FEPCluster with supplied information in FEPClusterCR.yaml.

Refer to "FEPCluster parameter" in the Reference for details.

2.3.2 High Availability

Describes the settings for using the highly available features.

Arbitration

Patroni is used to control and monitor FEP instance startup, shutdown, status and trigger failover should the master instance fails. It plays a significant role in the solution. If the Patroni process dies, especially on master POD, without notice, the Pod will not update the Patroni cluster lock. This may trigger an unwanted failover to one of the Replica, without corresponding corrective action on the running master. This can create a split brain issue. It is important to monitor Patroni's status to make sure it is running. This is done using liveliness probe. Important to note that this is not expected to be configured by end user.

```
livenessProbe:
  httpGet:
    scheme: HTTP
    path: /liveness
    port: 25001
  initialDelaySeconds: 30
  periodSeconds: 6
  timeoutSeconds: 5
  successThreshold: 1
  failureThreshold: 3
```

2.3.3 Configurable Volume per Cluster

Cluster node (Pod) volumes are created according to the values set in the storage section of fepChildCrVal in the FEPCluster custom resource.

Note

- After you create the FEPCluster for the first time, you cannot add new volumes later or modify the storageClass or accessModes.
- You can resize the initially created volume only if the underlying storageClass supports dynamic resizing.

The following is the schema for the storage section of the FEPCluster customer resource:

Field	Mandatory	Sub-Field	Default	Description
archivewalVol	No	size	1Gi	Volume size of the archive log. Refer to "Estimating Database Disk Space Requirements" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server to help you design the size.
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start
backupVol	No	size	2Gi	Volume size of the backup. Estimate based on the following formula: (full backup generations + incr backup generations + 1) * dataVol size
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start
dataVol	Yes	size	2Gi	Volume size of the data. Refer to "Estimating Database Disk Space Requirements" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server and base the design on table/index size.
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start
logVol	No	size	1Gi	Volume size of the log. If you change the log output level (default: WARNING) or enable the audit log feature, measure the actual amount of log output in a test environment.
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start
tablespaceVol	No	size	512Mi	Volume size of the tablespace. When using tablespaces, as with dataVol, you should refer to "Estimating Database Disk Space Requirements" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server for information on sizing.

Field	Mandatory	Sub-Field	Default	Description
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start
walVol	Yes	size	1200Mi	Volume size of the transaction log. Refer to "Estimating Database Disk Space Requirements" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server to help you design the size. Note that the default value for max_wal_size is 1 GB.
		storageClass	Defaults to platform default if omitted	SC is only set at start
		accessModes	Defaults to ReadWriteOnce if omitted	Access mode is only set at start

The 'accessMode' is been incorporated for the inclusion of pgBadger layer later. Giving it a shared volume capability will allow pgBadger Container to read logs from multiple server instance (master / replica) and expose it via a WebServer.

2.3.3.1 Disk Space Management

Due to a sudden increase in queries, etc., the amount of data and WAL will increase, and the disk capacity will be compressed, which may cause the database operation to stop. If the disk usage exceeds the threshold, or if database operation has stopped due to insufficient disk space, use the following methods to resolve the insufficient disk space.

- Increasing disk space
- Reducing disk usage

2.3.3.1.1 Increasing Disk Space

There are two ways to increase disk space:

- Expanding disk capacity

  If you are using a volume that can use the PVC extension function of Kubernetes, expand the disk capacity and solve the lack of space.

- Migrating to a database cluster with a large disk capacity

  If you are using a volume that cannot be used with the PVC extension function of Kubernetes, or if the volume cannot be expanded further due to the upper limit, etc., consider migrating the database cluster. Migrating data to a cluster that uses large-capacity disks solves the lack of capacity.

Expanding disk capacity

Expand your disk capacity with the Kubernetes PVC extension. Only disks that support the PVC expansion function can be expanded. Check the specifications of the CSI driver you are using to see if the disk supports PVC extensions.

Disk capacity expansion can be performed manually by the user at any time, or automatically by the operator when the usage exceeds the threshold in cooperation with the monitoring function.

Manual expansion can expand a PVC by changing the storage definition of the FEPCluster custom resource. Rewrite the custom resource and expand the PVC when AlertManager gives you a notification that the disk usage exceeds the threshold or the database stops due to lack of disk space.Refer to "5.3.2 Resizing PVCs" for more information on manual expansion. Also, refer to "Default Alert Rule" in the Reference for an example definition of AlertManager's alert rule.

Automatic expansion does not require database administrator monitoring or manual maintenance work (volume capacity expansion) until the expansion limit is reached.

For more information on auto expansion, refer to "2.3.3.2 Configuring PVC Auto Expansion".

If you are using a disk that does not support the PVC expansion function or if the disk capacity cannot be expanded, Refer to "Migrating to a database cluster with a large disk capacity" and "2.3.3.1.2 Reducing Disk Usage".

Migrating to a database cluster with a large disk capacity

Use the backup/restore function to construct a new database cluster on another disk and migrate the data. Use this method when:

- When the alert manager issues a notification that the disk usage exceeds the threshold
- When the database stops due to lack of disk space

Insufficient disk space can be resolved by changing to a disk with a larger capacity.

When migrating to a new database cluster, you can build a new FEPCluster and restore data by setting spec.changeParams of the FEPRestore custom resource and changing the definition from the restore source. For details, refer to "FEPRestore Custom Resource Parameters" in the Reference.

2.3.3.1.2 Reducing Disk Usage

Execute the REINDEX statement on the data storage destinations (dataVol, walVol, tablespaceVol) as preventive maintenance for insufficient disk space. For details, refer to "Reorganizing Indexes" in the Fujitsu Enterprise Postgres Operation Guide.

Consider reducing the amount of disk usage as preventive maintenance for insufficient capacity of the transaction log storage destination and backup data storage destination. If the capacity of the transaction log storage destination is insufficient, review the log file rotation settings and output level, and consider changing them. If the backup data or transaction log archive storage space is insufficient, consider reducing the number of backup generations saved.

You can reduce the number of backup generations by specifying "backup_expire" for spec.fepAction.type of the FEPAction custom resource. For details, refer to "FEPAction Custom Resource Parameters" in the Reference.

2.3.3.2 Configuring PVC Auto Expansion

By setting the FEPCluster custom resource spec.fepChildCrVal.storage.autoresize.enable to true, you can enable the PVC auto-grow feature that automatically expands disk space when disk usage exceeds a threshold.

The following two conditions must be met to enable the PVC auto-expansion function.

- Specify a volume that supports the PVC expansion function in StorageClass
- Specify true in the allowVolumeExpansion field in StorageClass

Check the specifications of the CSI driver you are using to see if the specified volume supports the PVC expansion function.

We also need Prometheus to monitor storage usage. Scrape the metrics captured by the kubelet below with Prometheus.

- kubelet_volume_stats_used_bytes (Volume used capacity (bytes))
- kubelet_volume_stats_capacity_bytes (Volume capacity (bytes))

Make sure that the scrape_config section of your Prometheus config file references /metrics, which the kubelet serves over https, for each node. Check the Prometheus documentation for more details.

The PVC auto-expansion feature can be enabled/disabled even after building the FEPCluster.

Enabling PVC auto-expansion builds a fep-tuning pod containing a pvc-auto-resize container. The pvc-auto-resize container periodically retrieves metrics from Prometheus for each defined PVC. If the PVC volume usage rate exceeds the defined threshold, the definition of the FEPCluster custom resource is automatically rewritten. The target PVC is automatically extended by rewriting the custom resource.

If the FEPCluster is configured with multiple units, the PVC will be expanded if the volume usage rate of even one unit exceeds the threshold.

The following parameters are used in the PVC auto expansion feature. For details of each parameter, refer to the Reference.

- spec.fep.autoTuning section: Prometheus connection information for retrieving metrics

- spec.fepChildCrVal.storage.autoresize section: Storage common extended settings
- spec.fepChildCrVal.storage.xxxVol section: definition and individual extension settings for each storage

In the expansion settings, it is possible to define the volume utilization threshold, amount of size expansion, upper limit of size that can be expanded, and so on.

Below is an example of defining a FEPCluster custom resource when enabling the PVC auto expansion feature.

```
spec:
  fep:
    autoTuning:
      prometheus:
        prometheusUrl: http://prometheus-prometheus-oper-prometheus.prometheus.svc:9090
  fepChildCrVal:
    storage:
      autoresize:
        enable: True
        threshold: 20
        increase: 20
      dataVol:         # Use the data volume as is defined under autoresize
        size: 10Gi
        storageClass: resizable-storage
      walVol:           # wal volume changes threshold and expansion limit
        size: 2Gi
        storageClass: resizable-storage
        threshold: 50
        storageLimit: 10
      backupVol:     # backup volume does not expand
        size: 20Gi
        storageClass: share-storage
        accessModes: ReadWriteMany
        storageLimit: 0
```

Concept of combination with monitoring feature

The PVC auto-expansion feature allows you to limit the amount of storage that can be expanded. However, there is a possibility that more data than expected may occur and the amount of data may exceed the set upper limit. To avoid this, we recommend using the monitoring function in conjunction with the PVC auto-expansion function.

In the alert rule created by default when using the FEPExporter function, an alert will be sent when the volume usage rate exceeds 90%. The default threshold for PVC autogrowth is 80%. As a result, even if the volume usage increases when the disk expansion limit is reached and automatic expansion is not performed, an alert will be sent from AlertManager, so you can be aware of the lack of disk space.

By setting the autogrowth threshold to a value lower than the alert rule threshold, you can keep more safe disk space.

2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator

Equivalent Kubernetes command: kubectl create FEPpgpool2

This operation will create a FEP pgpool2 container with supplied information.

Refer to "FEPPgpool2 Custom Resource Parameters" in the "Reference" for more information.

2.3.5 Scheduling Backup from Operator

When creating a FEPCluster, users can obtain scheduled backups by setting up backup definitions. Users can also modify the backup schedule by modifying the Backup custom resource that was created.

A backup definition includes the following:

- Acquisition time (Specify in crontab format)
- Backup type (Full or incremental backups)

Backup is taken on master Pod only.

Backup processing is performed by pgBackRest.

Parameter can be set to pgbackrestParams in CR definition.

The maximum number of backup schedules is 5.

See the pgBackRest User's Guide for details on the parameters.

However, some parameters are limited. Details are given below.

- Importantsetting items
- Parameters that cannot be set
- Restricted Parameters
- About sections in the config file

2.3.5.1 Important Setting Items

Here are the important parameters for setting pgBackRest. This parameter sets the retention period of backup information. If automatic backup is set and this parameter is not set, the risk of overflowing the backup area increases.

Parameter	Overview of parameters	Setting value
Full Retention Option (repo retention -full)	Specify number of full backups to keep No default (should be set according to user backup policy)	natural number
Full Retention Type Option (repo retention-full-type)	spec.retention -full Specifies whether the setting is a number of retention days (time) or a number of retention generations (count) No default (should be set according to user backup policy)	time/count

The following is a sample CR example of changing the backup retention period (How long the PITR is valid) to 30 days after the backup is taken.

```
apiVersion: fep.fujitsu.io/v1
kind: FEPBackup
metadata:
  name: fepcluster-backup
spec:
  pgBackrestParams: |
    # define custom pgbackrest.conf parameters below to override defaults.
    [global]
    repo-retention-full = 30
    repo-retention-full-type = time
...
```

2.3.5.2 Parameters that cannot be Set

The following parameters in the pgBackRest Configuration Reference are not configurable.

Parameter	Overview of parameters	Reason
Copy Archive Option (--archive -copy)	Copy the WAL segments needed for consistency to the backup	To use internal fixed values
Check Archive Mode Option (--archive-mode-check)	Check the PostgreSQL archive_mode setting.	Limited to backup from master

Parameter	Overview of parameters	Reason
Backup from Standby Option (--backup-standby)	Back up from the standby cluster	Limited to backup from master
Stop Auto Option (--stop-auto)	Stops a previously failed backup on a new backup.	Because they are 9.6 not supported in
pgBackRest Command Option (--cmd)	pgBackRest command	To use internal fixed values
SSH client command Option (--cmd-ssh)	Path to ssh client executable	Not using ssh
Compress Option (--compress)	Use File Compression	For obsolete options (Use compresse-type option instead)
Config Option (--config)	pgBackRest configuration file.	To use internal fixed values
Config Include Path Option (--config-include-path)	Path to additional pgBackRest configuration files.	To use internal fixed values
Config Path Option (--config-path)	Base path of pgBackRest configuration files.	To use internal fixed values
Delta Option (--delta)	Restore or Backup with Checksum	For new restores only
Dry Run Option (--dry-run)	Execute a dry-run for the command.	command-line only option
Lock Path Option (--lock-path)	Path where the lock file is stored	To use internal fixed values
Keep Alive Option (--sck -keep-alive)	Enable keep-alive messages on socket connections	To use internal fixed values
Spool Path Option (--spool-path)	Path to store temporary data for asynchronous archive-push and archive-get commands	For automatic determination from FEPCluster CR values
Stanza Option (--stanza)	Defines the stanza.	To use internal fixed values
Console Log Level Option (--log-level-console)	Console Log Level	It is not expected to operate on Pod.
Std Error Log Level Option (--log-level-stderr)	Stderr log level	It is not expected to operate on Pod.
Log Path Option (--log-path)	Log File Destination	For automatic determination from FEPCluster CR values
Repository Host Option (--repo-host)	Repository host for remote operations via SSH	Repository Host is not used
Repository Host Command Option (--repo-host-cmd)	Path of pgBackRest on Repository Host	
Repository Host Configuration Option (--repo-host-config)	Repository Host Configuration File Path	
Repository Host Configuration Include Path Option (--repo-host-config-include-path)	Repository hosts configuring include path	
Repository Host Configuration Path Option (--repo-host-config-path)	Repository Host Configuration Path	
Repository Host Port Option (--repo-host-port)	Repository host port when "repo-host" is configured	

Parameter	Overview of parameters	Reason
Repository Host User Option (--repo-host-user)	Repository host user when "repo-host" is configured	
Repository Path Option (--repo-path)	Path where backups and archives are stored	For automatic determination from FEPCluster CR values
Archive Retention Option (--repo-retention-archive)	The number of consecutive WAL backups to keep.	This option is not recommended, and WAL retention is controlled by the Full Retention Option and Full Retention Type Option.
Archive Retention Type Option (--repo-retention-archive-type)	Backup Type for WAL Retention	It is recommended not to change from the default.
Differential Retention Option (--repo-retention-diff)	Number of incremental backups to keep	No incremental backups
Archive Mode Option (--archive-mode)	Retains or disables the archive for the restored cluster.	To use internal fixed values
Exclude Database Option (--db-exclude)	Restore excluding the specified databases.	To restore the entire FEP cluster, including all databases
Include Database Option (--db-include)	Restore only the specified database	To restore the entire FEP cluster, including all databases
Link All Option (--link-all)	Restore all symbolic links.	To use internal fixed values
Link Map Option (--link-map)	Changes the destination of a symbolic link.	To use internal fixed values
Recovery Option Option (--recovery-option)	Setting options in postgreSQL recovery.conf	To use internal fixed values
Tablespace Map Option (--tablespace-map)	Restoring tablespace to a specified directory	For automatic determination from FEPCluster CR values
Map All Tablespaces Option (--tablespace-map-all)	Restores all tablespaces to the specified directory	No tablespace required because there is only one tablespace per FEPCluster
TLS Server Address Option (--tls-server-address)	TLS server address.	TLS server not used
TLS Server Authorized Clients Option (--tls-server-auth)	TLS server authorized clients.	
TLS Server Certificate Authorities Option (--tls-server-ca-file)	TLS server certificate authorities.	
TLS Server Certificate Option (--tls-server-cert-file)	TLS server certificate file.	
TLS Server Key Option (--tls-server-key-file)	TLS server key file.	
TLS Server Port Option (--tls-server-port)	TLS server port.	
PostgreSQL Database Option (--pg-database)	PostgreSQL database.	To use internal fixed values
PostgreSQL Host Option (--pg-host)	PostgreSQL host for remote operations via SSH	No SSH connection required

Parameter	Overview of parameters	Reason
PostgreSQL Host Command Option (--pg-host-cmd)	Path of pgBackRest exe on the PostgreSQL host	To use internal fixed values
PostgreSQL Host Configuration Option (--pg-host-config)	Path of the pgBackRest configuration file	To use internal fixed values
PostgreSQL Host Configuration Include Path Option (--pg-host-config-include-path)	Setting pgBackRest on PostgreSQL host include path	To use internal fixed values
PostgreSQL Host Configuration Path Option (--pg-host-config-path)	Path to configure pgBackRest on the PostgreSQL host	To use internal fixed values
PostgreSQL Host Port Option (--pg-host-port)	SSH Port Specification	No SSH connection required
PostgreSQL Host User Option (--pg-host-user)	The logon user when hosting PostgreSQL, if pg-host is set.	No SSH connection required
PostgreSQL Path Option (--pg-path)	PostgreSQL data directory.	For automatic determination from FEPCluster CR values
PostgreSQL Port Option (--pg-port)	PostgreSQL Ports	For automatic determination from FEPCluster CR values
PostgreSQL Socket Path Option (--pg-socket-path)	PostgreSQL Unix socket path	For automatic determination from FEPCluster CR values
PostgreSQL Database User Option (--pg-user)	PostgreSQL database user	To use internal fixed values

2.3.5.3 Restricted Parameters

Of the parameters in the pgBackRest Configuration Reference, the following parameters limit the configurable values.

Parameter	Overview of parameters	Possible Values
repoX-gcs-key-type	The type of key file you specify when using Google Cloud Storage	service

2.3.5.4 About Sections in the Config File

In FEPBackup CR, you can write the contents of pgbackrest.conf, but the setting for stanza (Backup space for pgBackRest) is specified internally.

The following sections are not allowed;

[stanza: command] , [stanza]

2.3.6 Perform PITR and Latest Backup Restore from Operator

There are two types of restore: one is to restore backup data to an existing FEPCluster, and the other is to create a new FEPCluster and restore backup data.

The former retains the attributes of the FEPCluster, such as IP address and name, while the latter is created from scratch.

The restore process deploys a FEP restore container. The FEP restore container performs the pgBackRest restore operation from the backup data to be restored to the master server of the FEPCluster. After the data is restored to the master server, the FEPCluster is created by synchronizing the data to two replica servers.

If user create a new FEPCluster, the newly created FEPCluster will inherit the settings of the source cluster, unless otherwise specified

User can also create a cluster with different settings from the source cluster by including the settings in FEPRestore CR.

Switching connections to the new cluster

The restore creates a new FEPCluster. If necessary, you need to set up Pgpool-II and change the access point of the application to the new cluster or the new Pgpool-II.

About recovering a failed FEPCluster

Even if the existing FEPCluster fails and the FEP is not running, if the volume of the backup area is safe, it is possible to restore from the backup data.

2.3.7 FEP Unique Feature Enabled by Default

Enable the following FEP features:

- Data masking
- Transparent Data Encryption (TDE)
- Fixed statistics

Data masking

The Data masking is enabled by default in the example FEPClster CR (in OpenShift UI). The postgresql.conf in container contains the following parameters:

```
shared_preload_libraries = 'pgx_datamasking,pg_prewarm'
session_preload_libraries = 'pg_prewarm'
max_worker_processes= 20
```

The user can overwrite these values in config map.

TDE

TDE is enabled by default. Select one of the following as the keystore to store the master encryption key used for transparent data encryption.

- File-based keystore
- External key management service

If you use a key management service as your keystore, you can change the keystore to another key management service even after you deploy the FEP cluster. You cannot change from a file-based keystore to a key management service, or from a key management service to a file-based keystore.

Refer to "2.3.12 Transparent Data Encryption Using a Key Management System" for the design perspective when using a key management system.

Fixed statistics

Fixed statistics(pg_dbms_stats) is enabled by default.

A dbms_stats schema is created in each database.

The list of statistics backups and the list of currently fixed objects can be referenced with pguser defined in the FEPClsuter custom resource.

It also takes a backup of statistics by default.

During a major upgrade, the backup statistics maintained in the database are no longer available. The immobilization status will also be released. Ttake a backup and fix it again after executing the major upgrade.

2.3.8 Monitoring & Alert (FEPExporter)

As the operator is level 5 certified, the system expose various metrics about its operand i.e. FEP containers.

FEP generates lot of useful database statistics via various views. The default statistics can be further augmented by using extensions like pg_stat_statements.

FEPExporter container by default is configured to extract useful database statistics and make the metrcs available to Prometheus on the platform. External components and utilities can be used to visualise, analyse, trigger alerts and take operational decision based on exposed metrics.

FEPExporter also sets defalt alert rules based on Prometheus metrics which are useful for active monitoring of FEP cluster.

2.3.8.1 FEPExporter Custom Resource

Refer to "FEPExporter Custom Resource" in the Reference for FEPExporter Custom Resource parameters.

- Custom queries to scrape metrics can be added in CR in optional section.
- Custom Prometheus alert rules are created by user manually.

2.3.8.2 Change to FEPCluster CR - metrics user

User may define pgMetricsUser, pgMetricsPassword and pgMetricsUserTls in target FEPCluster. If it is defined, FEPExporter will use metrics user details to connect to FEP cluster machines. All metrics user fields are optional and can be omitted in FEPCluster.

Refer to "FEPCluster Parameter" in the Reference for FEPCluster parameters.

2.3.8.3 FEPExporter CR auto-create for FEPCluster

User may define enableMonitoring flag as part of FEPCluster CR to monitor FEPCluster. It will automatically create FEPCluster specific FEPExporter so matrics scraping for FEPCluster will work.

Refer to "FEPCluster Parameter" in the Reference for FEPCluster parameters.

- FEPExporter will be named as <cluster-name>-fepexporter.
- Once FEPExporter created automatically, user can modify it manually from FEPExporter CR.
- If FEPCluster will be deleted, it will delete dependent FEPExporter as well.
- MTLS for FEPExporter will only supported when tls configuration defined for both Prometheus & FEPExporter specs.

2.3.9 Collaboration with Amazon CloudWatch

You can use CloudWatch, a monitoring service for Amazon Web Services(AWS), to monitor FEP metrics and collect logs. Typically, CloudWatch requires you to forward metrics and logs to CloudWatch for database metrics monitoring and log collection, but you can use an operator to automatically forward FEP metrics and logs to CloudWatch. You can centrally monitor the FEP and services provided by AWS, such as EC2 and S3 object storage built into business systems, thereby reducing the workload of monitoring tasks.

This feature is only supported on x86.

For information on how to collaborate with CloudWatch, refer to "4.5 How to Collaborate with CloudWatch". For information on the metrics to be collected, refer to "5.7.2.5 Metrics Collected by CloudWatch".

2.3.10 Scaling Replicas

Auto scale out occurs when the average database CPU utilization or number of connections exceeds the threshold. Select whether the criteria for auto scale out is CPU usage or the number of connections, depending on the resource that is the bottleneck of the database.

The maximum number of replica containers, excluding the master container, is 15.

Scale out based on CPU utilization

Performs a scale out if the average CPU utilization of all pods (primary pods and all replica pods) in the FEPCluster exceeds the threshold for a period of time.

CPU utilization is calculated with the value specified in spec.fep.mcSpec.requests.cpu specified for the FEPCluster custom resource as the denominator.

Scale out based on the number of connections

Performs a scale out if the average number of connections for all pods (primary pods and all replica pods) in the FEPCluster exceeds the threshold for a period of time.

Specify the threshold for the number of connections to perform automatic scale-out with a value less than or equal to the max_connections parameter of the FEP server.

The prerequisites for using the scale out feature based on the number of connections are as follows.

- The monitoring feature (see "2.3.8 Monitoring & Alert (FEPExporter)") is enabled.
- Metrics for the number of FEP server connections are collected by the monitoring feature.
- A custom metrics server is installed in the OCP/Kubernetes cluster.
- The custom metrics server publishes the average number of connections collected by the monitoring feature.

When using the scale out feature based on the number of connections, the auto scale out feature requests the custom metrics server for metrics associated with the following Kubernetes resources.

- kind: FEPCluster
- apiVersion: fep.fujitsu.io/v2
- name: Name of FEP Cluster
- namespace: The name of the namespace in which FEP Cluster is deployed

The name of the requested metric is the name specified in the metricName parameter.

This metric should represent the average number of connections for each pod in the specified FEPCluster.

Limitations

- If you want to use the scale out feature based on the number of connections, deploy FEPExporter according to the procedure of "4.3 Deploying FEPExporter".
- If FEPCluster metrics are collected by FEPExporter in standalone mode (see "4.4 FEPExporter in Standalone Mode"), the scale out feature based on the number of connections is not available.

When using the auto scale out feature, the FEPCluster sync mode should be "off".

Precautions when designing auto scale out

- The auto scale out feature adds replicas one at a time. In addition, additional replicas take time to service, depending on the environment and the amount of data stored. As a result, replica growth may not be able to keep up with the increased load.
- Even if the auto scale out feature increases the number of replicas, incoming requests are not given priority to those replicas. As a result, existing FEP instances may continue to be temporarily overloaded after the number of replicas increases.
- The auto scale out feature increases the number of replica requests that can be handled only by reference requests to the database. Requests with updates continue to be processed on the primary FEP instance. Therefore, the auto scale out feature may not reduce the load on the primary FEP instance.
- Currently, the auto scale out feature does not delete replicas (reduce the number of replicas). If the load decreases after the number of replicas increases due to a temporary increase in load, the number of replicas remains increased. If necessary, manually change the number of replicas.

2.3.10.1 Change to FEPCluster CR - auto scale out

If you want to use Auto Scale Out, set the parameter to FEPClusterCR.

Refer to "FEPCluster Parameter" in the Reference for FEPCluster parameters.

2.3.11 Disaster Recovery

By using object storage, data can be migrated to database clusters in different container environments. Even if it is difficult to operate in a container environment with a database cluster deployed due to a disaster, etc., it is possible to continue operation in a different container environment.

Available disaster recovery methods include the backup/restore method and the hot standby method.

Backup/restore method

Build a new container environment after a disaster (cold standby) and restore data from object storage. Compared to the method described later, this method does not require the construction of two container environments, so it is possible to keep costs down. It takes recovery time to execute.

Hot standby method

Before a disaster occurs, start the container environment of the production environment and the container environment of the disaster recovery environment. By implementing disaster recovery in a hot standby configuration, business systems can be restored more quickly in the event of a disaster.

You can also use Velero to back up a system in a Kubernetes environment to object storage for disaster recovery that restores the system to a different Kubernetes.

2.3.12 Transparent Data Encryption Using a Key Management System

Fujitsu Enterprise Postgres provides unique features that enhance the security of PostgreSQL. These security features help users keep their data safe from unauthorized access. One such security feature is Transparent Data Encryption (TDE), which encrypts data at rest, i.e. data stored on disk/persistent volume.

In contrast, TDE's default format stores the master encryption key in a password-protected file. A key management system allows you to store your master encryption key (MEK) in a cloud-based keystore, taking your security to the next level.

The key management system that can be used with transparent data encryption is one of the following:

- Key management server using KMIP protocol
- AWS key management service (x86 only)
- Azure key management service (x86 only)

Refer to "Appendix D Key Management System Available for Transparent Data Encryption" for detailed key management system requirements.

Transparent data encryption using a key management system can only be configured when the FEPCluster is first created. Users cannot configure an existing FEPCluster for transparent data encryption using a key management system.

If the master encryption key on the key management system is lost, the encrypted/backup data cannot be decrypted. As long as the data encrypted with the master encryption key remains valid, the master encryption key must also be available and maintained on a key management system.

If you have encrypted backups with old encryption key, you must keep the old encryption key available after the master encryption key is rotated. Otherwise, you will not be able to open the database restored from backup.

In addition, the key custodian must retain the referenced master encryption key for as long as the data encrypted under the old master encryption key remains valid.

2.3.13 Database Role Management

In order to manage data access control, you can easily implement database role privilege and expiration management.

Operators can easily create roles related to database operations, assign privileges, and manage the expiration dates of database roles with login privileges in order to manage data access control.

Databases contain important data such as personal information, and data protection is important.

Data protection is defined in security protocols and is an important aspect of operations.

In order to protect data from being viewed by a third party, it is necessary to properly set the access control of database roles.

In this feature, it is recommended to divide into the following database roles.

- Database administrator: Construction/operation of database system
- Confidential administrator: Set appropriate privileges for each database resource
- General users: End users of the database

By preparing multiple database operators/administrators and assigning privileges to each, it is possible to distribute privileges. This makes it possible to prevent data from being referenced or tampered with by users with strong privileges.

This section describes the roles of database roles created by this feature.

Database administrators can perform operations related to database operations, such as referencing system tables and canceling back-end queries.

Confidential administrators grant appropriate privileges to tables and roles to prevent third parties from viewing data. With this feature, it is possible to grant the confidential administrator the privilege to use the confidentiality management feature, and to grant the appropriate privilege to each database resource.

In addition, it is not recommended to use roles with SUPERUSER or BYPASSRLS privilege that can see all data for data protection. Therefore, in this feature, the SUPERUSER (postgres) password is isolated by hiding it, and the SUPERUSER and BYPASSRLS privileges are not granted to the created database role.

2.3.13.1 Creating Roles Related to Database Operation

2.3.13.1.1 Quarantine SUPERUSER

Create a database role "postgres" with SUPERUSER privileges for the operator when building the database.

By omitting "spec.fepChildCrVal.sysUsers.pgAdminPassword" in the FEPCluster custom resource, the postgres role password is created with a random value, making it impossible for general users to use SUPERUSER privileges. However, a separate method is provided to use the "postgres" role when the administrator needs SUPERUSER privileges for database operations. Therefore, monitor for unexpected usage using the audit feature of pgAudit.

2.3.13.1.2 Database Administrator Role

Database role for database management. Defining this role is mandatory.

The user name and password are defined in "pguser" and "pgpassword" under spec.fepChildCrVal.sysUsers in the FEPCluster custom resource. Has CREATE DATABASE privilege and can see system tables/cancel backend queries.

The database administrator role has the following privileges.

- NOSUPERUSER
- NOREPLICATION
- NOBYPASSRLS
- CREATEDB
- INHERIT
- LOGIN
- CREATEROLE

However, NOCREATEROLE privileges are granted when the confidential administrator role is created.

I also belong to the following roles:

- pg_monitor
- pg_signal_backend

2.3.13.1.3 Confidential Administrator Role

A database role that uses the confidentiality management feature to set appropriate privileges for each database resource for database users. Creating this role is optional.

User name and password are defined in "pgSsecurityUser" and "pgSsecurityPassword" under "spec.fepChildCrVal.sysUsers" of FEPCluster custom resource.

Confidential administrator roles can be defined after building FEPCluster. However, you cannot change the role name or delete the role after defining this role.

This role holds the following privileges.

- LOGIN
- CREATEROLE
- NOSUPERUSER
- NOREPLICATION
- NOBYPASSRLS
- NOCREATEDB
- NOINHERIT

The Confidential administrator role has ALL privileges for the database defined in the FEPCluster custom resource "spec.fepChildCrVal.sysUsers.pgdb", and can create database objects such as tables in the target database.

Confidential administrator roles are assigned the following privileges necessary to operate the confidentiality management feature of Fujitsu Enterprise Postgres, so the confidentiality management feature can be used immediately after the role is created.

- CREATEROLE privilege
- SELECT privilege, INSERT privilege, UPDATE privilege, and DELETE privilege to all tables included in the extension of confidentiality management feature

Grant ownership to the confidential administrator role for the database objects managed by the confidentiality management feature.

In addition, by granting the privileges required for the confidential administrator role to operate the confidentiality management feature to other database roles, the number of users who perform confidential management can be increased and the privileges can be distributed.

2.3.13.2 Expiration Management of Database Roles with Login Privileges

You can control who can connect to the database by managing role expiration.

This section describes two methods for managing role expiration:

- Policy-based password operation
- Password operation with the VALID UNTIL clause

In policy-based password operation, you define a policy as a profile and assign the profile to a role. It is possible to define policies other than expiration exactly as follows:. By assigning the same profile to a role, you can apply the same policy at once.

- Set a password life time
- Restrict password reuse
- Lock accounts that have failed to log in continuously

For password operations with the VALID UNTIL clause, only an expiration date can be set. An operator can force a VALID UNTIL clause to be specified within a specified interval of time when a CREATE or ALTER ROLE command is executed.

2.3.13.2.1 Policy-based Password Operation

For information about how to create and apply profiles that define policies, refer to "Policy-based Login Security" in the Fujitsu Enterprise Postgres Operation Guide.

When FEPCluster is deployed, users for streaming replication (User specified in spec.fepChildCrVal.sysUsers.pgrepluser of the FEPCluster custom resource) are automatically assigned membership in the pgx_update_profile_status group role. This ensures that the user lock status and password status detected on the standby server are applied to the entire cluster immediately after the FEPCluster is built.

Operators can use the monitoring feature to monitor metrics such as expiration.

When you deploy FEPExporter, the following metrics collection and alert rules are enabled:

To deploy the FEPExporter, refer to "4.3 Deploying FEPExporter" or "4.4 FEPExporter in Standalone Mode".

For definitions of metric queries and alert rules, refer to "Default Metric Queries" and "Default Alert Rules" in the Reference.

Metrics Collected by Default

- Number of all roles
- Number of enabled roles
- Number of roles in grace period
- Number of roles that have expired
- Number of roles locked

You can also add metrics queries to monitor, for example, how many roles expire within a specified date.

Default Alert Rules

- When Expired Role is 1 or More
- When the Grace Period Expired Role is 1 or More
- When the locked role is 1 or more

When you are notified of an alert, you can verify the database role in question from the system catalog.

For more information about the system catalogs for managing profile information, refer to "System Catalogs" in the Fujitsu Enterprise Postgres Operation Guide.

2.3.13.2.2 Password Operation with the VALID UNTIL Clause

You can manage password expiration for database roles with login privileges.

When defining passwords for database roles with login privileges in the CREATE ROLE or ALTER ROLE statements, it is possible to force them to expire within a specified period.

Specify the following parameters in the FEPCluster custom resource to enable this feature.

- Specify "fsep_operator_security" in shared_preload_libraries of spec.fepChildCrVal.customPgParams
- "spec.fepChildCrVal.sysUsers.passwordValid.days" specifies the number of days that can be specified from the time the password is changed to the expiration date

FEPCluster custom resource definition example

```
fepChildCrVal:
  customPgParams: |
    shared_preload_libraries='pgx_datamasking,pg_prewarm,pg_stat_statements,fsep_operator_security'
    …
 sysUsers:
    passwordValid:
      days: 30
    pgdb: mydb
    pgpassword: mydbpassword
    pguser: mydbuser
```

When this feature is enabled, the password expiration for pgpassword and pgSecurityPassword, as defined in spec.fepChildCrVal.sysUsers in the FEPCluster custom resource, is defined after the length of time specified in spec.fepChildCrVal.sysUsers.passwordValid.days since the password was changed.

However, passwords defined in pgAdminPassword, pgreplpassword, pgRewindUserPassword, and pgMetricsUserPassword are database role passwords required for database operation management, so their expiration dates are not managed.

In addition, if the CREATE ROLE or ALTER ROLE statement defines/changes the password for a database role that has login privileges, and the password for the database role does not expire or is longer than the length of time specified by pec.fepChildCrVal.sysUsers.passwordValid.days, the executed SQL will fail.

spec.fepChildCrVal.sysUsers.passwordValid.days can be defined or changed after building the FEPCluster. Changing this parameter updates the password expiration period for all managing database roles that have not expired.

Removing spec.fepChildCrVal.sysUsers.passwordValid.days or setting it to 0 will stop password expiration management.

By using the FEPExporter custom resource feature, it is possible to monitor the password expiration date of database roles and send an alert using AlertManager when there is a database role that is about to expire or has passed.

2.3.14 User Synchronization Using ldap2pg

ldap2pg manages database roles in an LDAP-configured system by applying management information such as users on the LDAP server to the database server. ldap2pg allows automatic synchronization of LDAP directories with FEP user accounts. The synchronisation is done by ldap2pg on a regular basis. The frequency of synchronisation can be defined as a cronjob like entry. Connectivity from ldap2pg to LDAP directory and FEP to LDAP directory must support LDAP, LDAPS and LDAP over TLS.

Built-in Operator users such as postgres, repluser, rewind_user, mydbuser should not be created/deleted/modified by ldap2pg as it may interfere with normal operation of FEP Operator and FEP Cluster. The default ldap2pg.yml supplied by the FEP Operator excludes these users.

To use ldap2pg, you need the following:

LDAP connection info

Details about LDAP directory, include ldap uri, basedn, binddn must be defined and made available to ldap2pg. The ldap.conf is used to store such information. It will be available to any software that needs connectivity information to LDAP directory, including ldap2pg and FEP.

Details of ldap.conf will be stored in a secret and mounted to fep-patroni under /etc/openldap/ldap.conf. End user should pre-create this secret before enabling ldap2pg synchronisation. If this secret is not present when ldap2pg is enabled, Operator will create a sample ldap.conf with connectivity info commented out. End user can update the secret with corresponding LDAP connectivity info. This updated information will be reflected in fep-patroni when kubelet performs syncPod function.

The Secret can be named as user-preferred, e.g., <fep-cluster>-ldapconf. Its key must be "ldap.conf" and its value should contain LDAP configuration.

Trusted CA bundle for LDAP directory

If the LDAP directory is using certificates signed by private CA to secure connections, the chain of certificates up to the root CA must be provided for ldap2pg and FEP as trusted certificates for a secure connection, be it ldaps or ldap over TLS. This chain of certificates will be stored in a configmap and mounted to fep-patroni under /tls/ldap/ca.crt. The use of such trusted certificates is optional but is recommended to authenticate the authenticity of the LDAP servers.

The ConfigMap can be named as user-preferred, e.g., <fep-cluster>-cacrt. Its key must be "ca.crt" and its value should contain LDAP server certificates which are provided by end-user.

ldap2pg.yml

Configuration file for ldap2pg to interact with LDAP directory and FEP.

Details of ldap2pg.yml will be stored in a configmap and mounted to fep-patroni under /tmp/.config/ldap2pg.yml. End user should pre-create this confimap before enabling ldap2pg synchronisation. If this configmap is not present when ldap2pg is enabled, Operator will create a sample ldap2pg.yml. End user can update the confimap. The updated information will be reflected in fep-patroni when kubelet performs syncPod function.

The ConfigMap can be named as user-preferred, e.g., <fep-cluster>-ldap2pgyml. Its key must be "ldap2pg.yml" and its value should contain ldap2pg configuration.

2.3.14.1 How to Update FEPCluster CR for ldap2pg

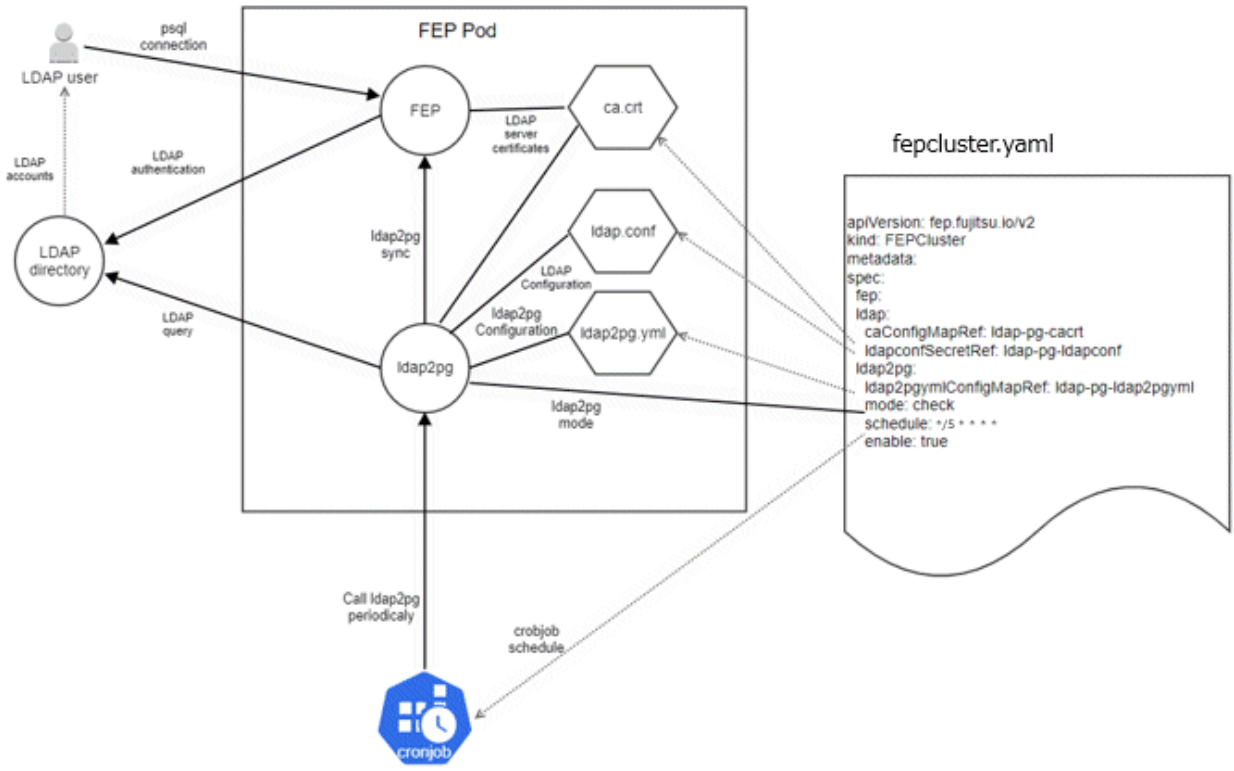
This section describes instructions for users to update FEPCluster CR to support ldap2pg.

First, it will need to create resources below for both LDAP configuration and ldap2pg configuration.

- Secret for LDAP Configuration (ldap.conf)
- ConfigMap for LDAP Server Certificates (ca.crt)
- ConfigMap for ldap2pg Configuration (ldap2pg.yml)

Second, it will need to add spec.ldap and spec.ldap2pg entries to a FEPCluster CR below to support ldap2pg (refer to figure below).

- FEPCluster with ldap2pg enabled (fepcluster.yaml)



Follow the steps below to update the FEPCluster CR for ldap2pg.

Each step includes the approach by using OpenShift console and command lines.

1. Create a Secret for LDAP configuration (ldap.conf)

   LDAP configuration includes LDAP connection info.

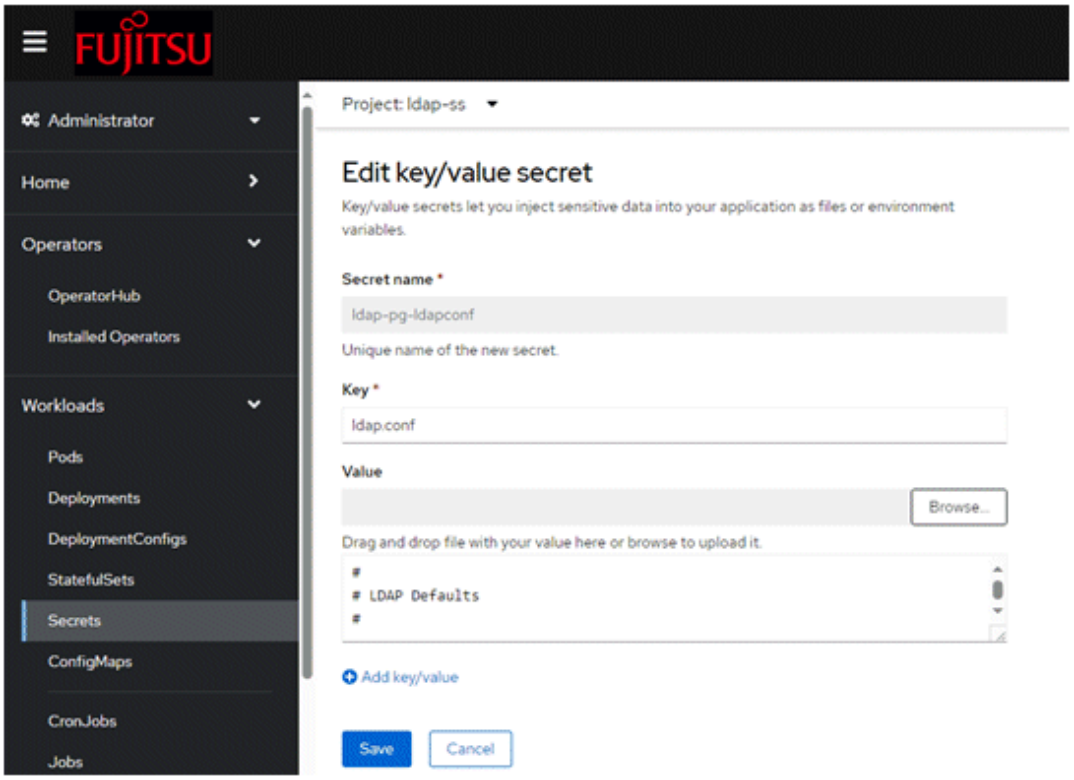
   In this example, we assume that the LDAP server is running at ldap-server.testprj.svc, listening for LDAPS connection on port 636. The base DN is dc=example,dc=org. The LDAP user cn=admin,dc=example,dc=org is used to connect to LDAP server and the password is "mypassword".

   Sample LDAP configuration (ldap.conf)

```
$ cat ldap.conf

#
# LDAP Defaults
#

#Refer to https://www.openldap.org/software/man.cgi?query=ldap.conf for full configuration
```

```
details
# This file should be world readable but not world writable.

#BINDDN: This is the user to bind to LDAP server
BINDDN  cn=admin,dc=example,dc=org

#BASE: Set it to the base DN where search will begin
BASE   dc=example,dc=org

#PASSWORD: Password of user as defined in BINDDN
PASSWORD mypassword

#URI: Specify the URI of LDAP server
URI   ldaps://ldap-server.testprj.svc:636

#SIZELIMIT    12
#TIMELIMIT    15
#DEREF        never
TIMEOUT       15

# TLS certificates (needed for GnuTLS)
# TLS_CACERT      /tls/ldap/ca.crt

# Specifies what checks to perform on server certificates in a TLS session. We are
# being relax here even if server certificate cannot be verified.
TLS_REQCERT allow
```

- Create a Secret by OpenShift.



- Create a Secret by command-lines

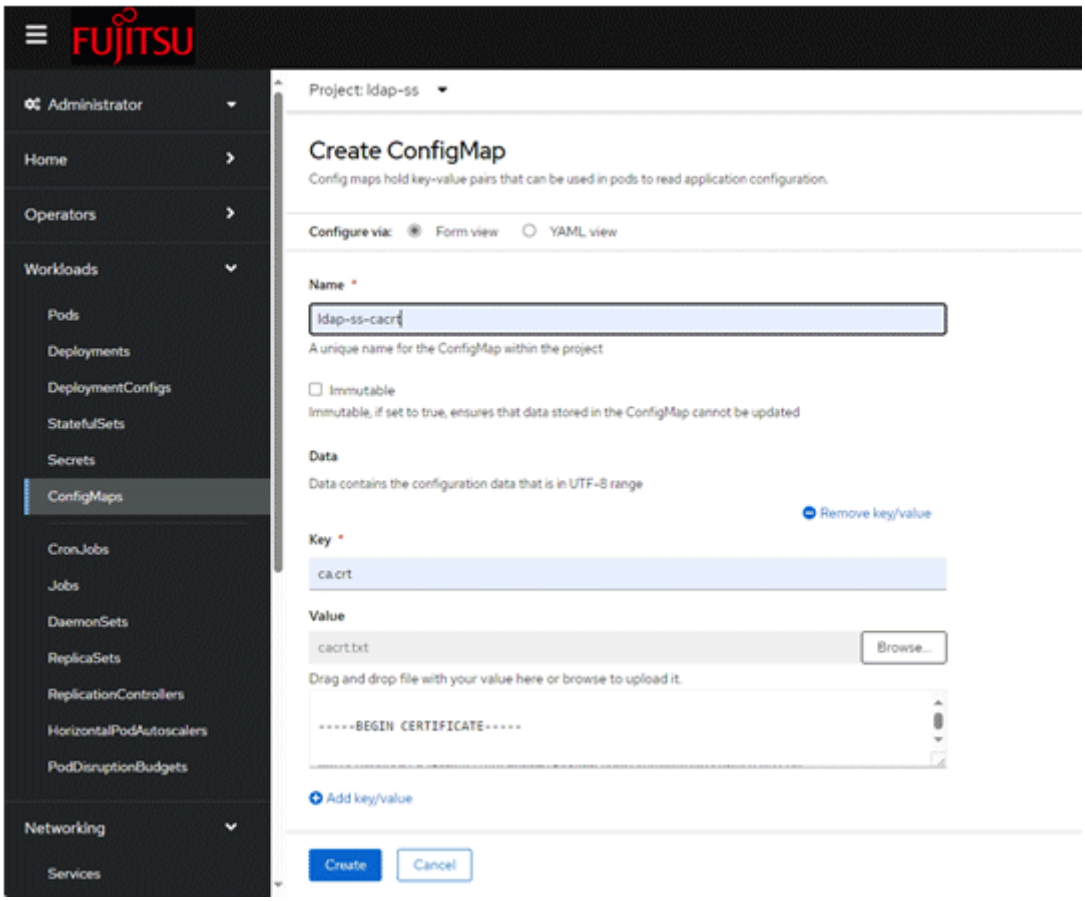
```
$ kubectl -n ldap-ss create secret generic ldap-ss-ldapconf --from-file=ldap.conf
secret/mysecret created
```

2. Create a ConfigMap with chain of certificates (ca.crt) that sign the LDAP server certificate

Sample LDAP server certificates (ca.crt)

```
$ cat ca.crt

-----BEGIN CERTIFICATE-----
xxx root cert xxx
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
yyy intermediate cert yyy
-----END CERTIFICATE-----
```

- Create a ConfigMap by OpenShift.



- Create a ConfigMap by command-lines

```
$ kubectl -n ldap-ss create configmap ldap-ss-cacrt --from-file=ca.crt
configmap/ldap-ss-cacrt created
```

3. Create a ConfigMap for ldap2pg.yml file

The following users are managed by operators, so specify them in roles_blacklist_query in the postgres section to remove them from ldap2pg management.

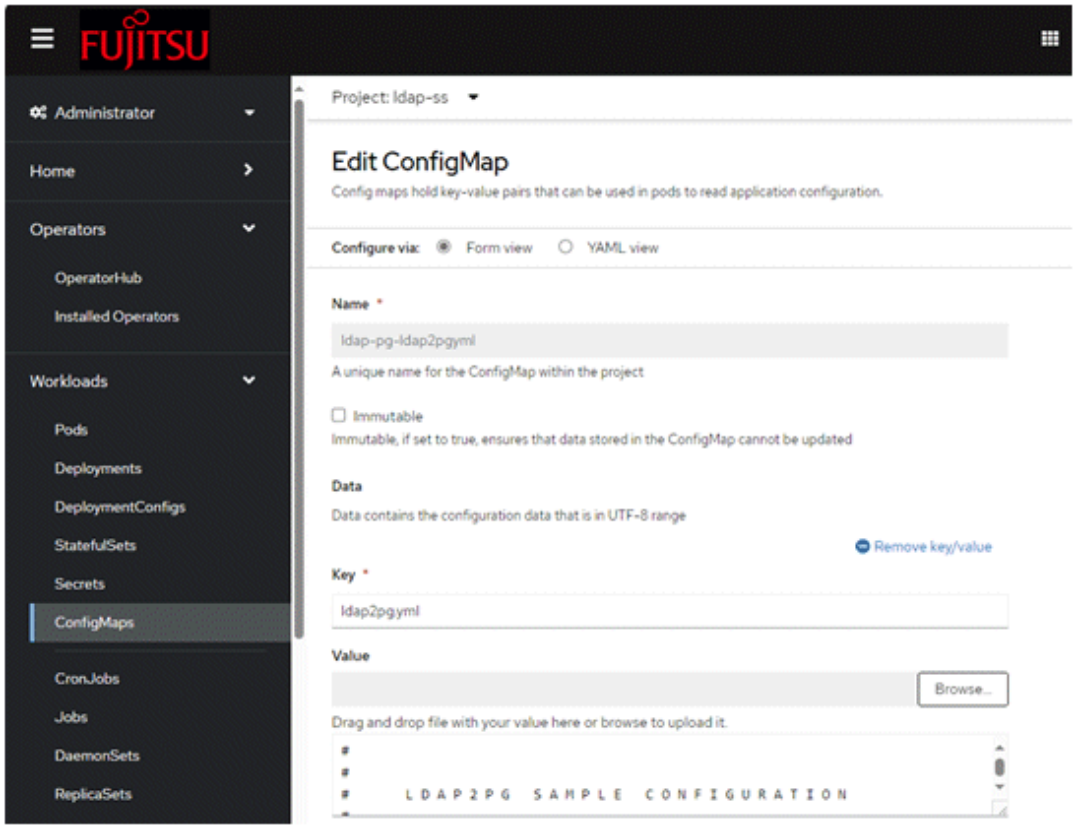
- postgres user
- Username specified in spec.fepChildCrVal.sysUsers of FEPCluster custom resource

For other ldap2pg.yml settings, refer to "ldap2pg" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Client.

Sample ldap2pg configuration (postgres section of ldap2pg.yml)

```
postgres:
  roles_blacklist_query: [postgres, repluser, rewind_user, pwsyncuser, pg_*, pgx_*, ]
  databases_query: [postgres]
```

- Create a ConfigMap by OpenShift.



- Create a ConfigMap by command-lines

```
$ kubectl -n ldap-ss create configmap ldap-ss-ldap2pgyml --from-file=ldap2pg.yml
configmap/ldap-ss-ldap2pgyml created
```

4. Add spec.ldap and spec.ldap2pg entries to FEPCluster CR (fepcluster.yaml)

   It will need to add the entries below to FEPCluster CR during the creation and updating of FEPCluster.

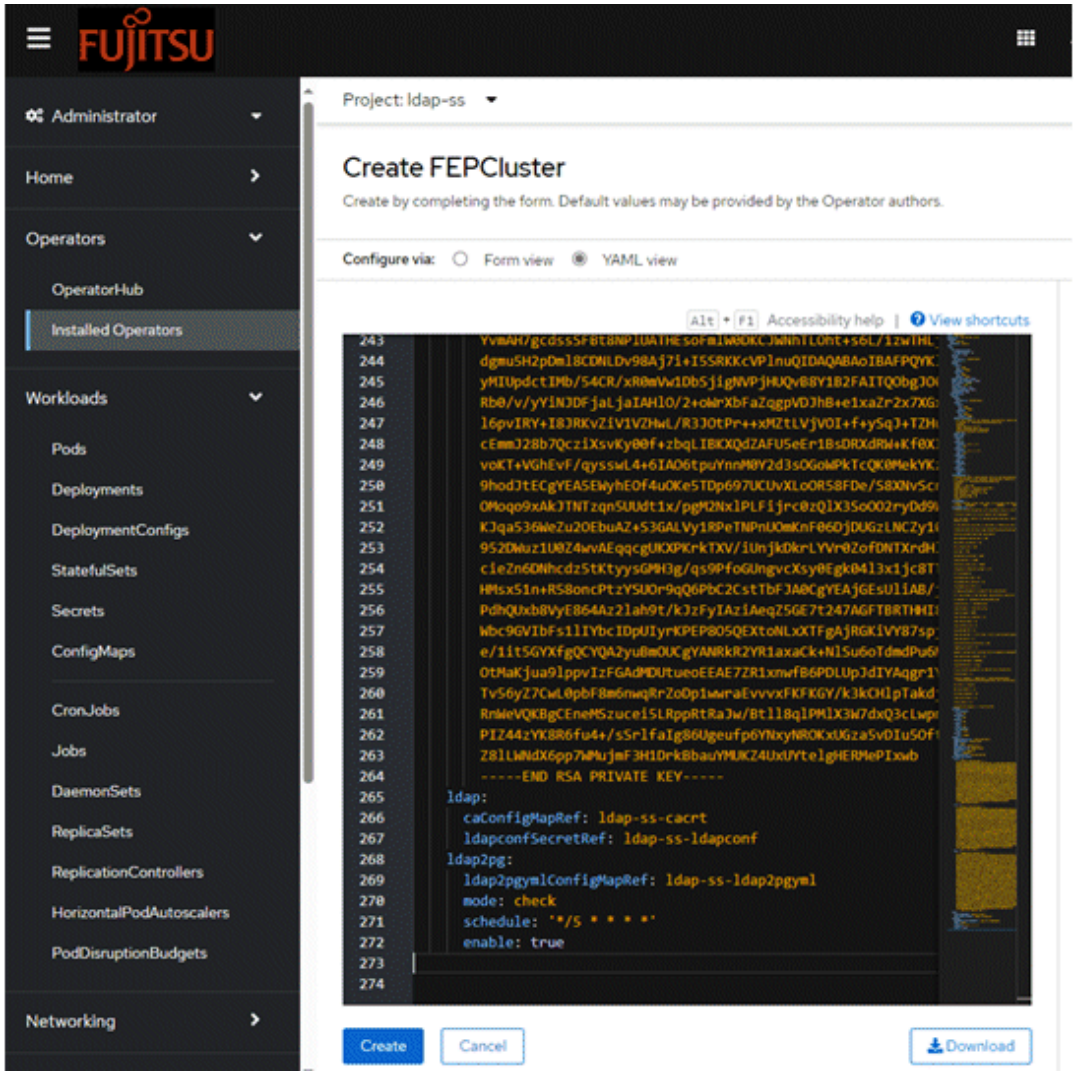
Key	Value	Description
spec.ldap.caConfigMapRef	ldap-ss-cacrt	Refer to step 1
spec.ldap.ldapconfSecretRef	ldap-ss-ldapconf	Refer to step 2
spec.ldap2pg.ldap2pgymlConfigMapRef	ldap-ss-ldap2pgyml	Refer to step 3
spec.ldap2pg.mode		Default value is "check"
spec.ldap2pg.schedule		Default value is '*/5 * * * *'
spec.ldap2pg.enable		Default value is true

   You can disable ldap2pg feature by setting the spec.ldap2pg.enable to false.

Sample FEPCluster CR with ldap2pg changes (ldap2pg is enabled) (fepcluster.yaml)

```
$ cat fepcluster.yaml

apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: ldap-ss
  namespace: ldap-ss
spec:
  ldap:
    caConfigMapRef: ldap-ss-cacrt
    ldapconfSecretRef: ldap-ss-ldapconf
  ldap2pg:
    ldap2pgymlConfigMapRef: ldap-ss-ldap2pgyml
    mode: check
    schedule: '*/5 * * * *'
    enable: true
```

- Update FEPCluster CR by OpenShift.



- Update FEPCluster CR by command-lines

```
$ kubectl apply -f fepcluster.yaml
```

2.3.15 Fixed Statistics

Fixed statistics means tuning the intended statistics so that they are always used to prevent performance degradation due to changes in query plans. This is achieved by the pg_dbms_stats extension. Operator can schedule statistics to be fixed. The user simply defines a schedule in the FEPCluster custom resource, and the operator automatically fixes the statistics that the user intended.

The following figure shows the procedure for executing the schedule. For more information, refer to "5.14 Operation of Fixed Statistics".

STEP1: Verify the performance of the statistics in the FEPCluster validation environment

STEP2: Store statistics from FEPCluster in validation environment to object storage

STEP3: Define a schedule for fixing statistics to FEPCluster custom resources in a production environment

STEP4: Deploy FEPCluster custom resources for production and schedule statistics consolidation
The binary file download of statistics from object storage occurs as needed.

You can also back up statistics on a regular basis. In the event of performance degradation due to changes in query plan, the backed up statistics can be used to achieve performance stabilization.

pg_dbms_stats allows you to do the following:

- Backup
- Restore
- Lock
- Unlock
- Purge
- Cleanup
- Import

Use FEPAction to perform various operations. For more information, refer to "FEPAction Specific Operation Details" in the Reference.

Chapter 3 Operator Installation

This chapter describes how to install FEP operator.

Refer to "6.5 Assigned Resources for Operator Containers" for more information about the resources assigned to installed operator containers and how to change them.

"<*x.y.z*>" in the screen example indicates the version level of the operator. Also, "<*X*>" indicates the product version of Fujitsu Enterprise Postgres.

Note

If you are using anti-virus software, exclude the directory for resource allocation from virus scanning.

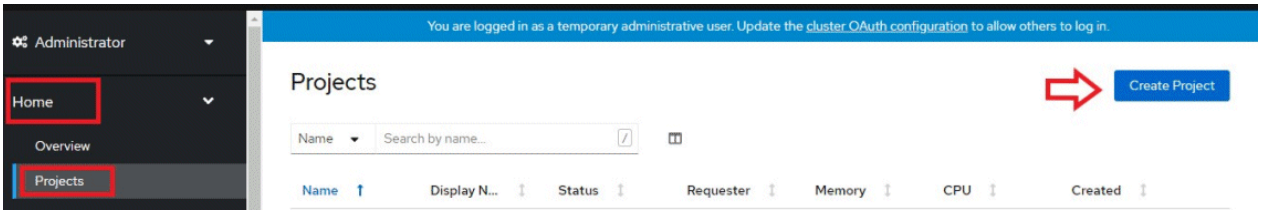
3.1 Using the OperatorHub

Describes how to use OperatorHub to install FEP operators into a new namespace on Openshift.

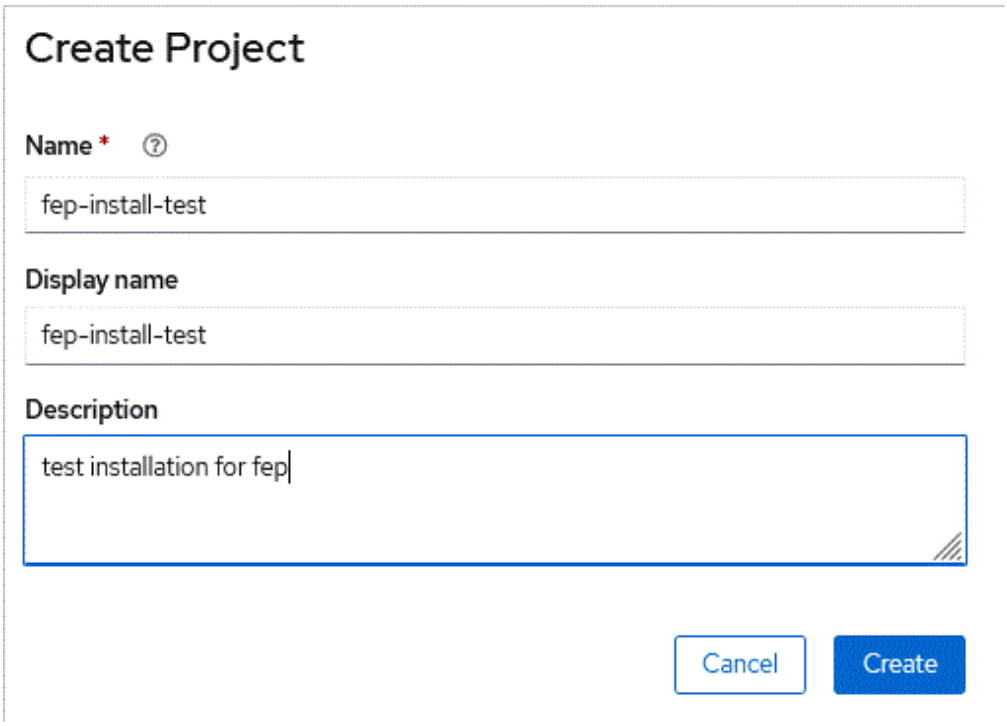
3.1.1 Pre-requisite

A project on openshift is essentially a namespace. It is a good practice to install FEP in a separate name space. On the RedHat OpenShift platform, click "Home" under "Projects" main menu and hence click on "Create Project".

(Screen Shot 1 and 2 - Create Project on OCP - ***for ref.***)



In the dialog box, specify a unique name for your namespace and an optional display name and description.



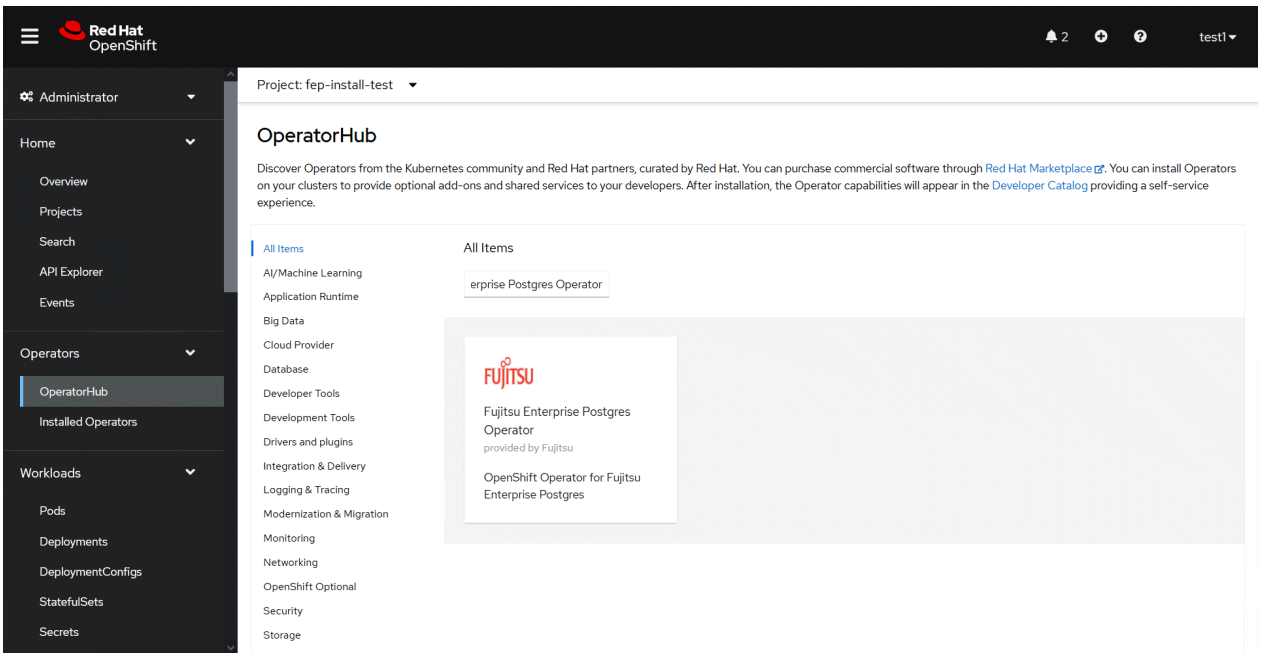
Note

Operator installation needs Prometheus to be pre-installed in the Openshift cluster.

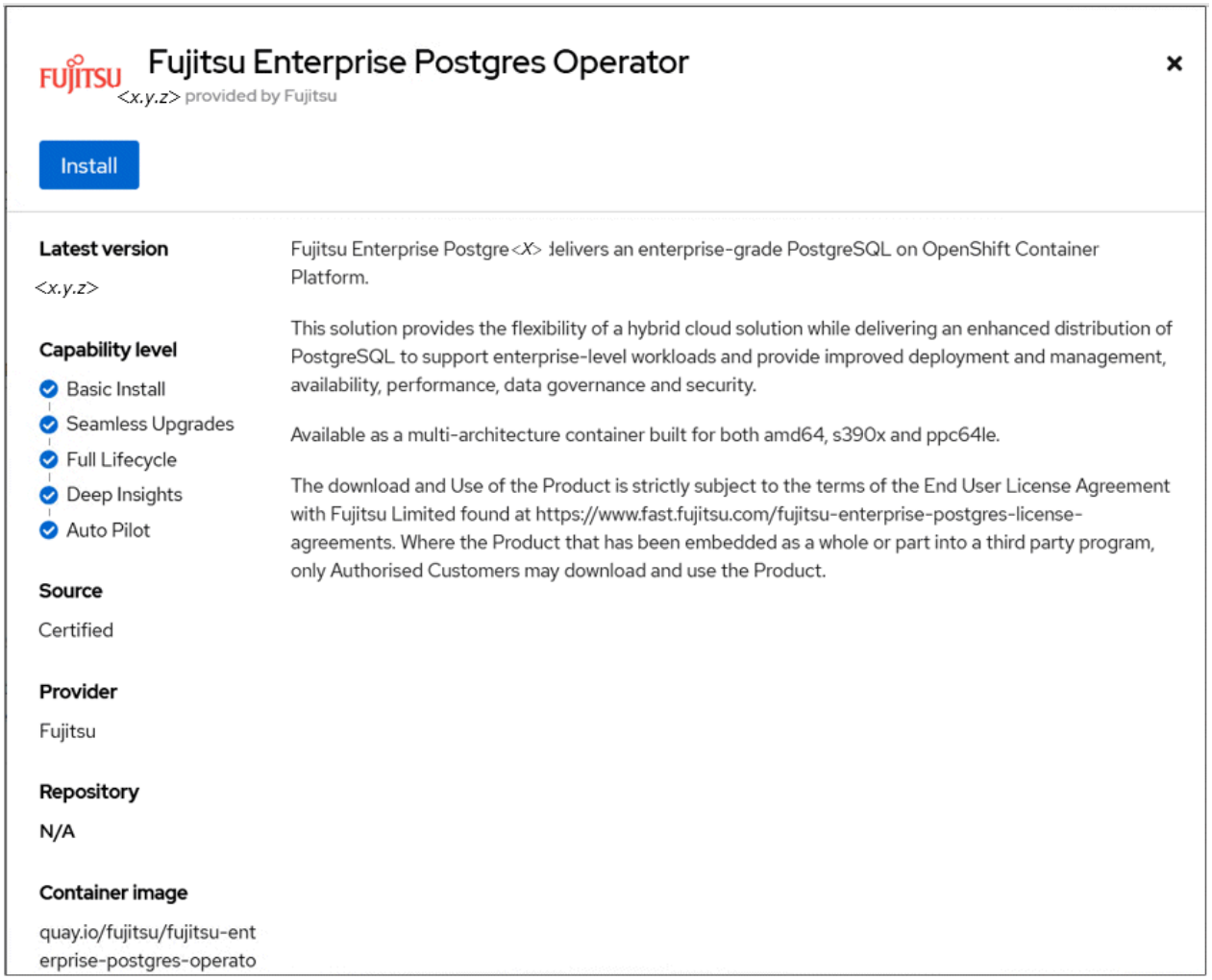
3.1.2 Deploying Operator

Once operator is certified by RedHat, it is made available on OperatorHub on all RedHat OpenShift container plaform.

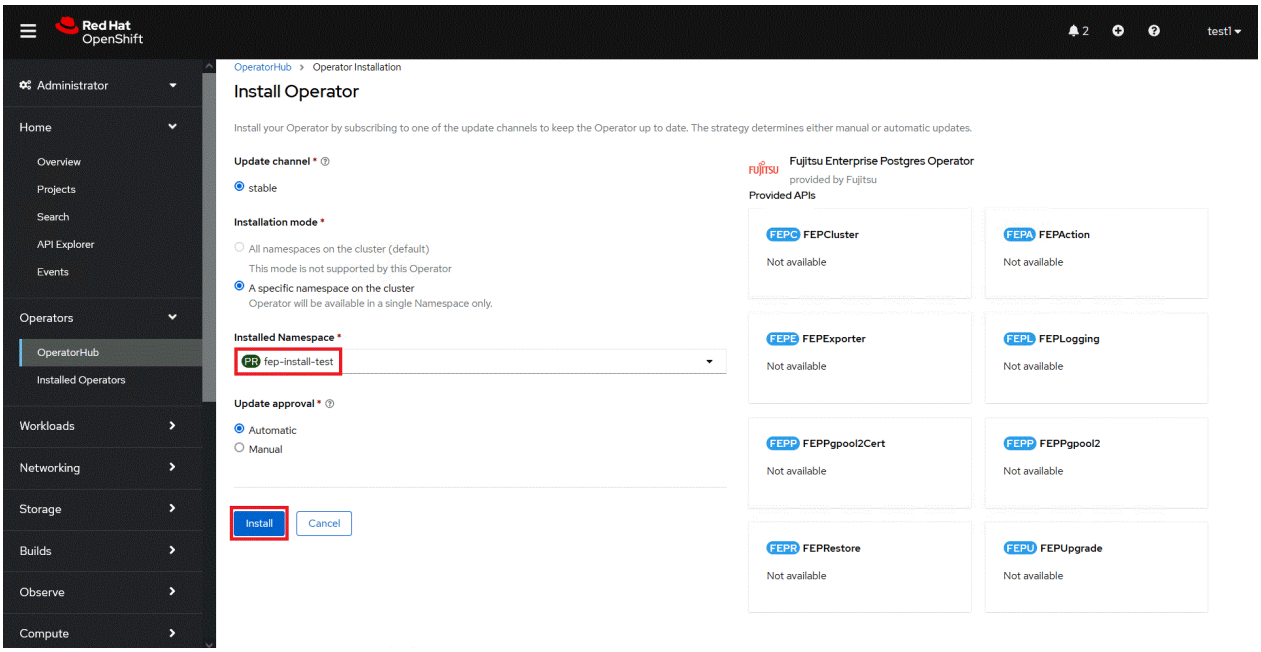
On OpenShift platform, logon with credentials that has privileges to install operator. Click on OperatorHub on menu item under Operators and type filter keyword "Fujitsu Enterprise Postgres Operator" to find Fujitsu Enterprise Postgres Operator.



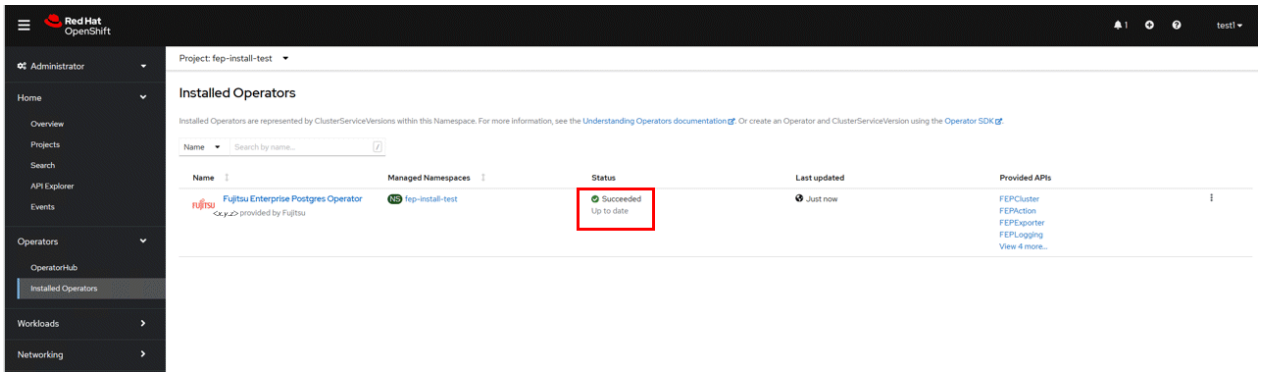
Click on Fujitsu Enterprise Postgres Operator to install operator. It will bring up details page with install button as below.



Click on "Install" button, to bring up following screen to choose namespace and approval strategy. Select "A specific namespace on the cluster" and choose desired namespace.Leave everything else to default and click install.



Wait still installation is complete and status changes to "Succeeded".



3.1.3 Upgrading Operators

If you select Automatic for "Update approval" on the "Install Operator" screen under "3.1.2 Deploying Operator", the operator will be automatically upgraded when a new version is published.

If Manual is selected for "Update approval", perform the following steps to upgrade manually.

1. Select the installed "Fujitsu Enterprise Postgres Operator" from [Operators] - [Installed Operators].
2. Follow the on-screen instructions to upgrade your operator from [Subscription details] in the [Subscription] tab

3.2 Using the Helm Chart

Describes how to install FEP operators into a new namespace on Kubernetes using the Helm feature.

3.2.1 Deploying Operator

1. Add a Helm Chart repository for the operator.

```
helm repo add fep-repo https://fujitsu.github.io/fep-operator-helm/v1
```

2. Create a namespace to install the operator.

```
kubectl create namespace fep-operator
```

Note

Operator installation needs Prometheus to be installed in the Kubernetes cluster in advance.

3. Run the helm command to install the operator.

```
helm install fep-operator-release fep-repo/fujitsu-enterprise-postgres-operator --namespace fep-operator
```

3.2.2 Upgrading Operators

1. Refresh Helm Chart repository information.

```
helm repo update
```

2. Check the Helm Chart version of the latest operator.

```
helm search repo fujitsu-enterprise-postgres-operator
```

3. Run the helm command to upgrade the operator.

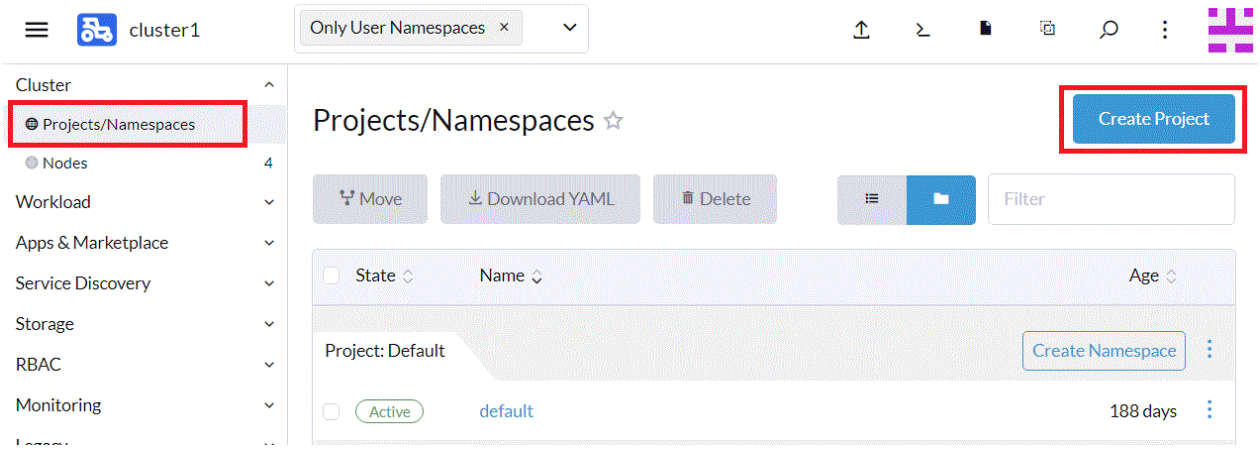
```
helm upgrade fep-operator-release fep-repo/fujitsu-enterprise-postgres-operator --namespace fep-operator
```

3.3 Using the Rancher UI

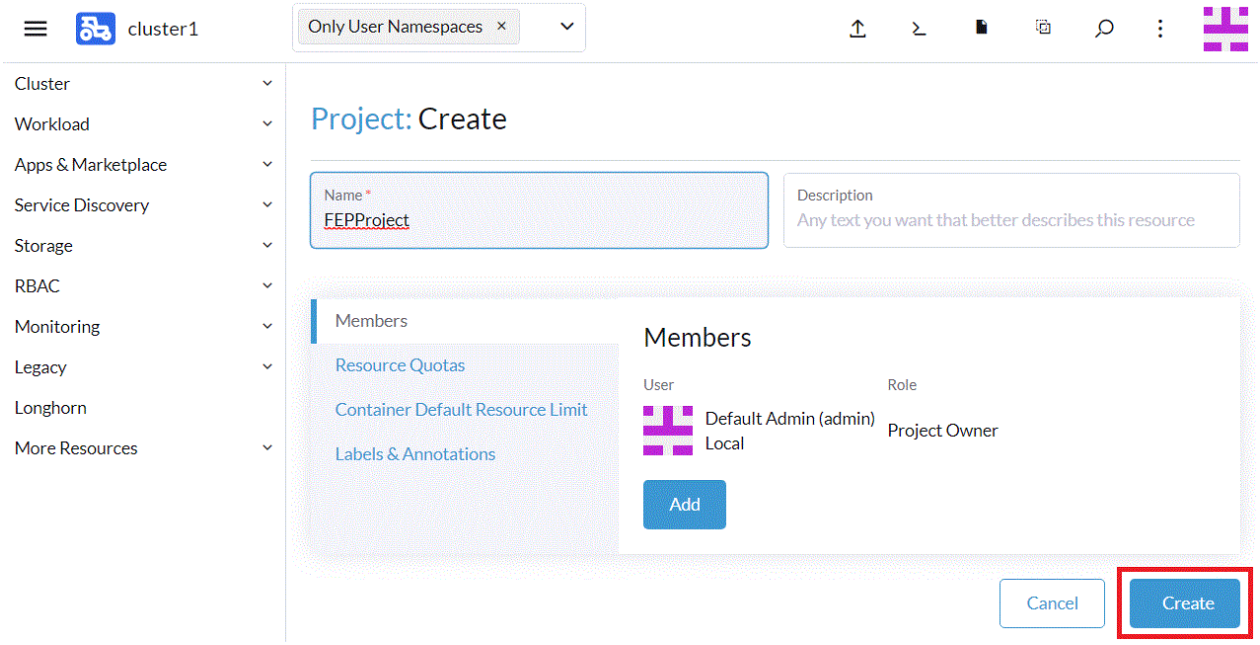
Describes how to install FEP operators into a new namespace on Rancher UI.

3.3.1 Pre-requisite

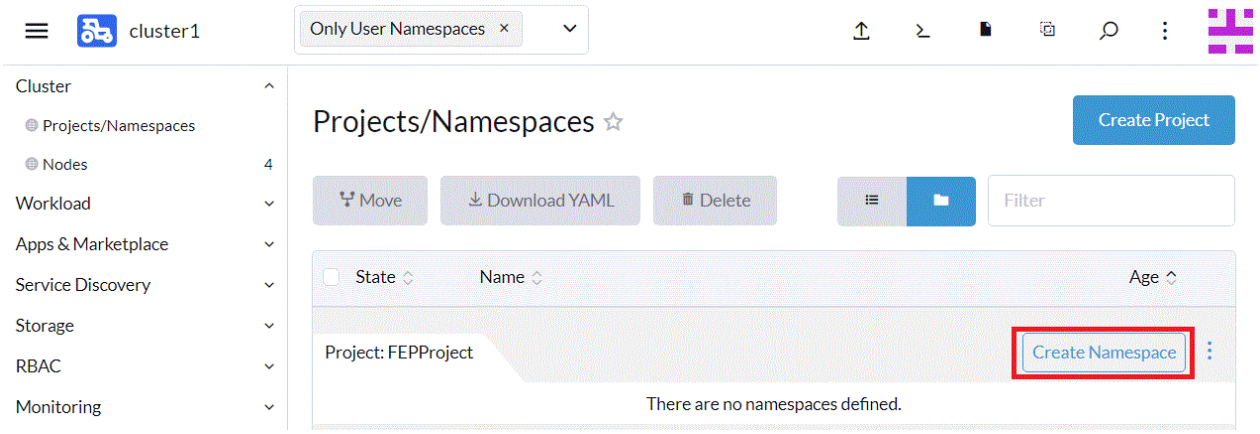
Create a project and its associated namespace on the Rancher UI. We recommend that you install FEP in a different namespace. In the Rancher UI, click [Projects/Namespaces], then click [Create Project] that appears.



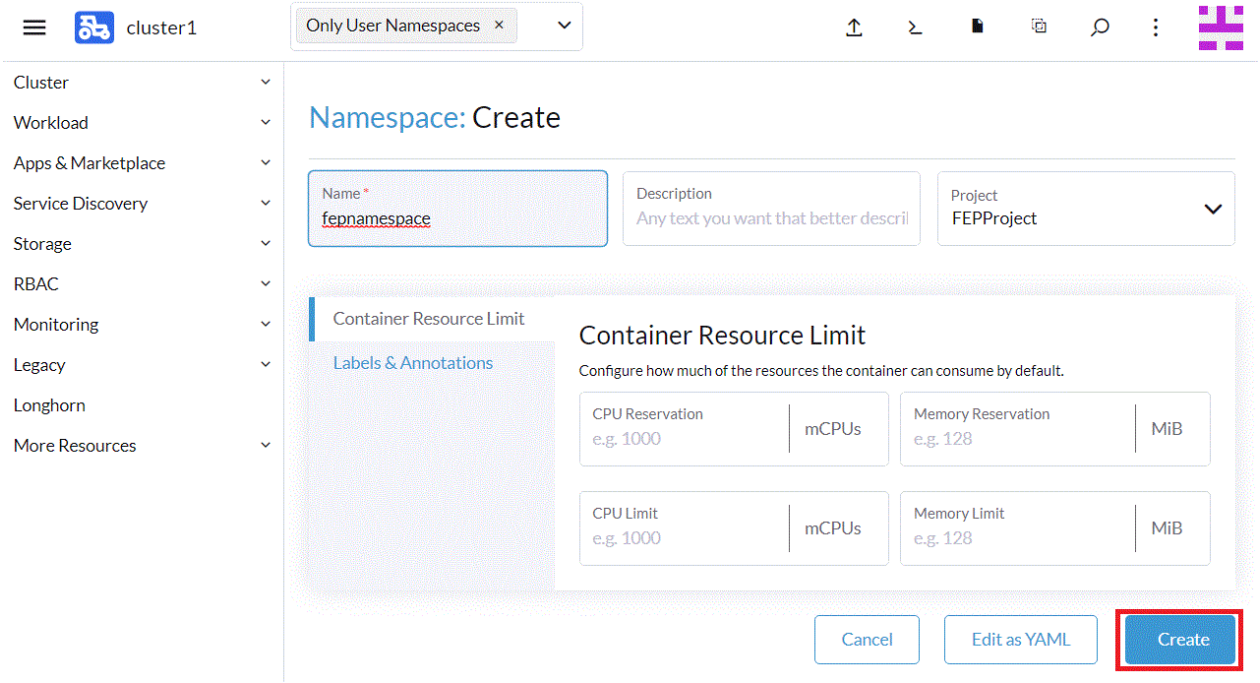
Specify a unique name for the project and click [Create].



Click [Create Namespace] displayed on the specified project.



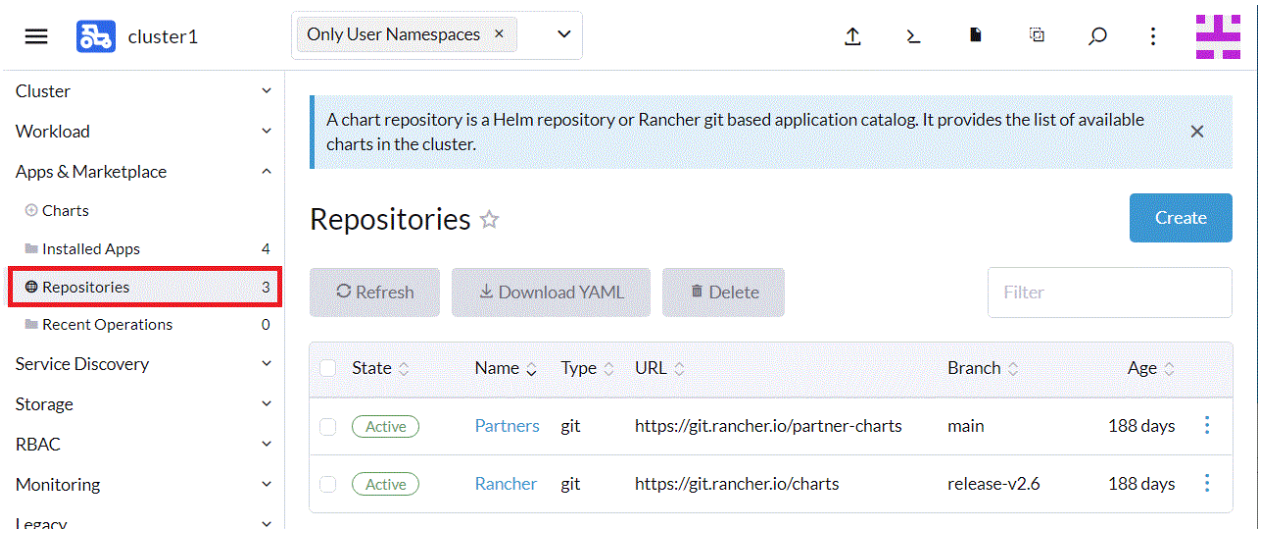
Specify a unique name in the namespace and click [Create].



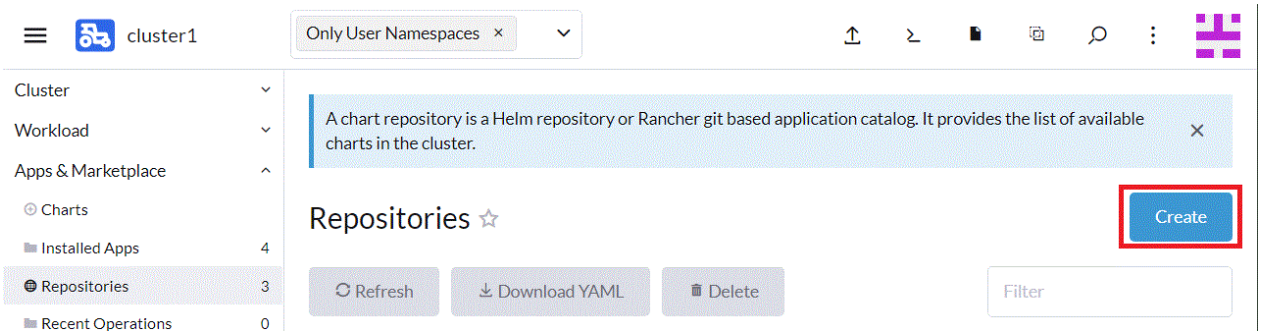
3.3.2 Register Helm Chart Repository

Register the Helm Chart repository of the operator feature on the Rancher UI.

In the Rancher UI, click [Apps & Marketplace], then click [Repositories] that appears.

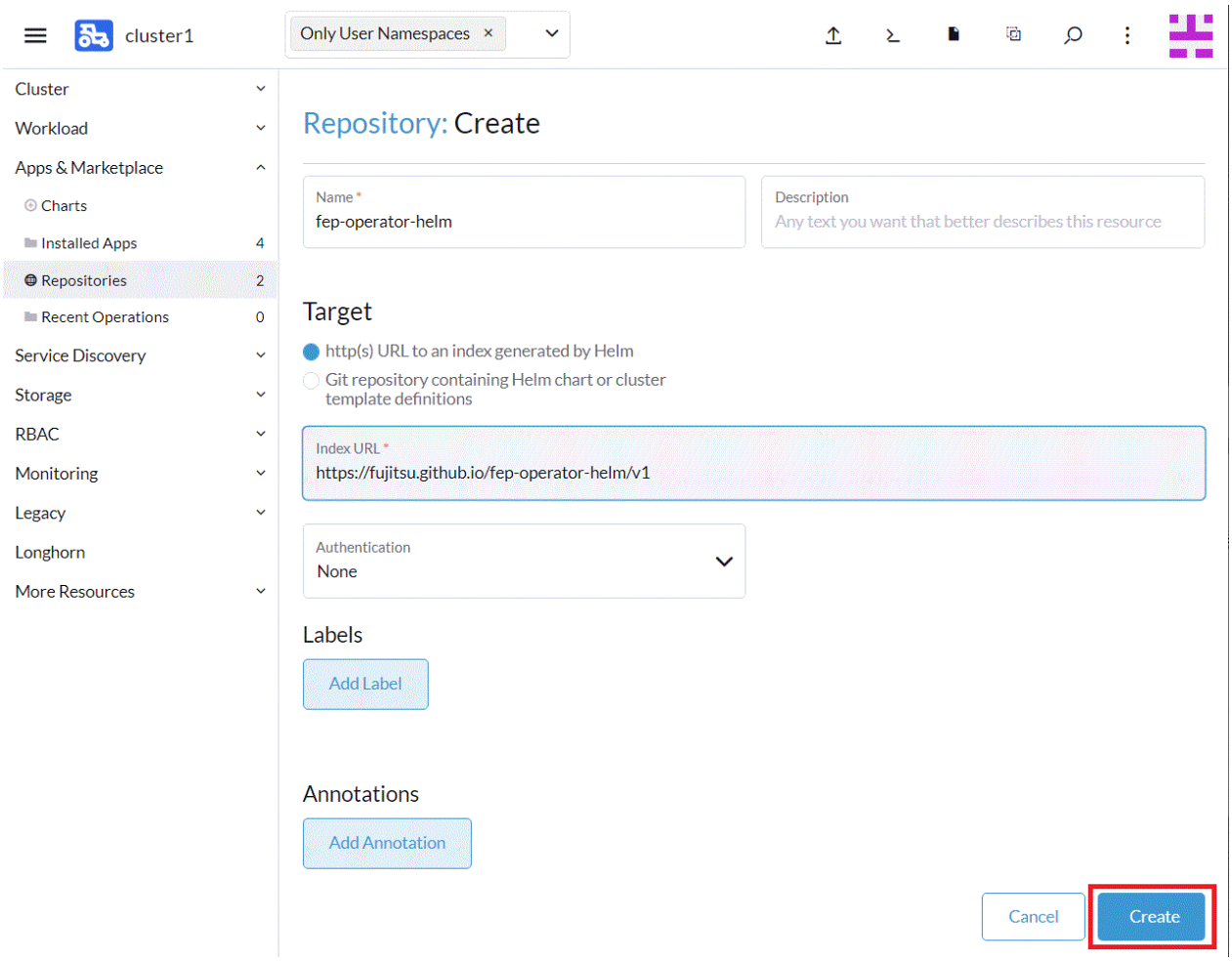


Click [Create] to create the Helm Chart repository.



Enter the unique name of the catalog and the URL of the catalog below, and click [Create].

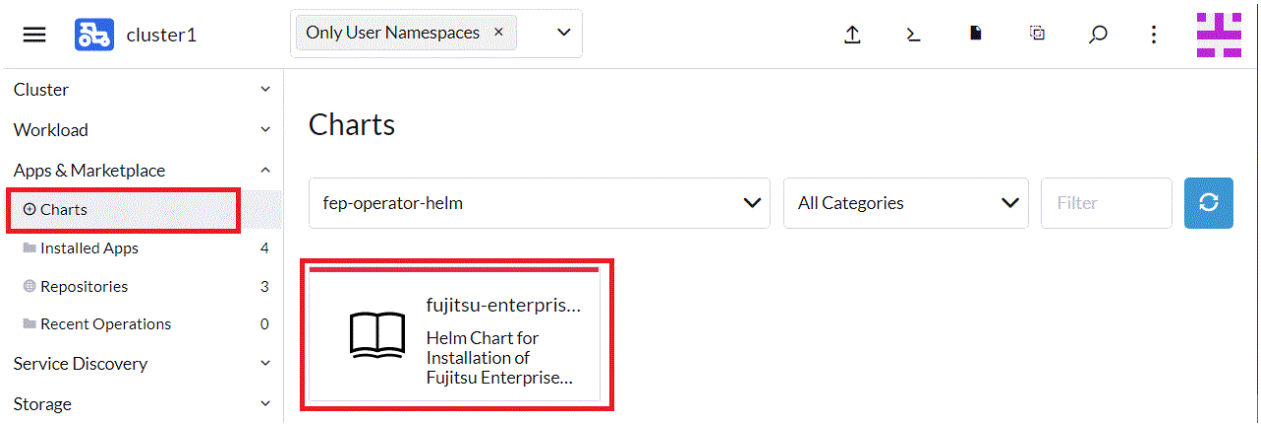
```
https://fujitsu.github.io/fep-operator-helm/v1
```



3.3.3 Deploying Operator

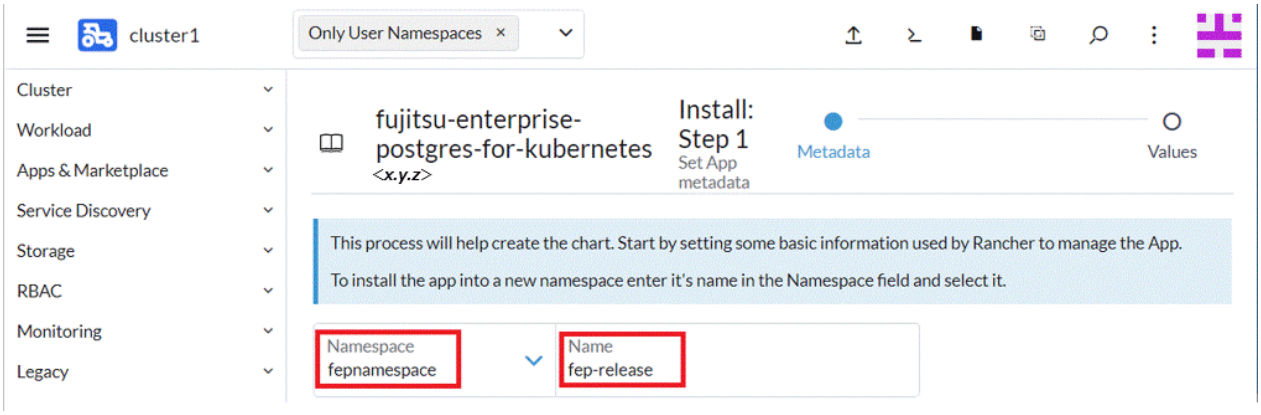
On the Rancher UI, apply the operator function Helm Chart to the project / namespace created in "3.3.1 Pre-requisite" and install the operator.

From the leftmost tab, click [Charts], then click [fujitsu-enterprise-postgres-operator].

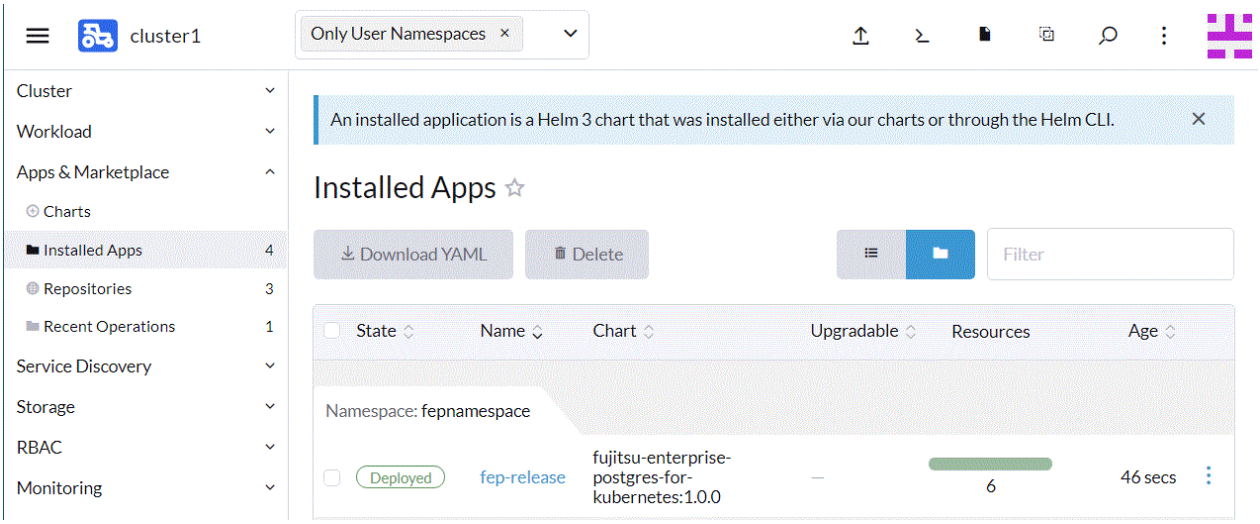


Click Install on the screen that appears.

Change the [Namespace] item to the name created in "3.3.1 Pre-requisite", enter the release name in the [Name] item, click [Next], and then click [Install] on the next screen.



The operator is deployed on the target namespace.



3.3.4 Upgrading Operators

1. Update the Helm Chart repository.

   In the Rancher UI, click [Apps & Marketplace] and then click [Repositories] that appears.

   Click Refresh to update the repository.

2. Upgrade the operator.

   In the Rancher UI, click Apps & Marketplace, then click Installed Apps.

   Select the operator you installed and follow the on-screen instructions to upgrade the operator.

3.4 Implement Collaborative Monitoring Tools

3.4.1 Implement GAP Stack

There is a pre-requisitie for running FEPExporter.

- GAP(Grafana, AlertManager, Prometheus) stack is installed on host OpenShift or Kubernetes cluster
- FEPCluster that needs to be scraped is deployed and running properly
- FEPCluster has following setting postgresql.conf:
  - pg_stats_statements library pre-loaded
  - track_activities and track_counts are turned on

For Prometheus and AlertManager, use the monitoring stack preinstalled on Openshift. Please refer to the following for deployment information.

(https://docs.openshift.com/container-platform/4.11/monitoring/monitoring-overview.html#understanding-the-monitoring-stack_monitoring-overview)

For Grafana, install and use the GrafanaOperator provided by OperatorHub. Grafana is not exposed by OperatorHub in s390x and ppc64le, so use Helm to build Grafana. Detailed instructions are available at the following site for your reference.

(https://www.postgresql.fastware.com/knowledge-base/how-to/setting-up-grafana-on-ibm-linuxone)

Grafana comes pre-installed on OpenShift, but it is recommended to use Grafana published in OperatorHub to customize the dashboard and monitor FEP performance information.

You can also use the sample dashboards published below.

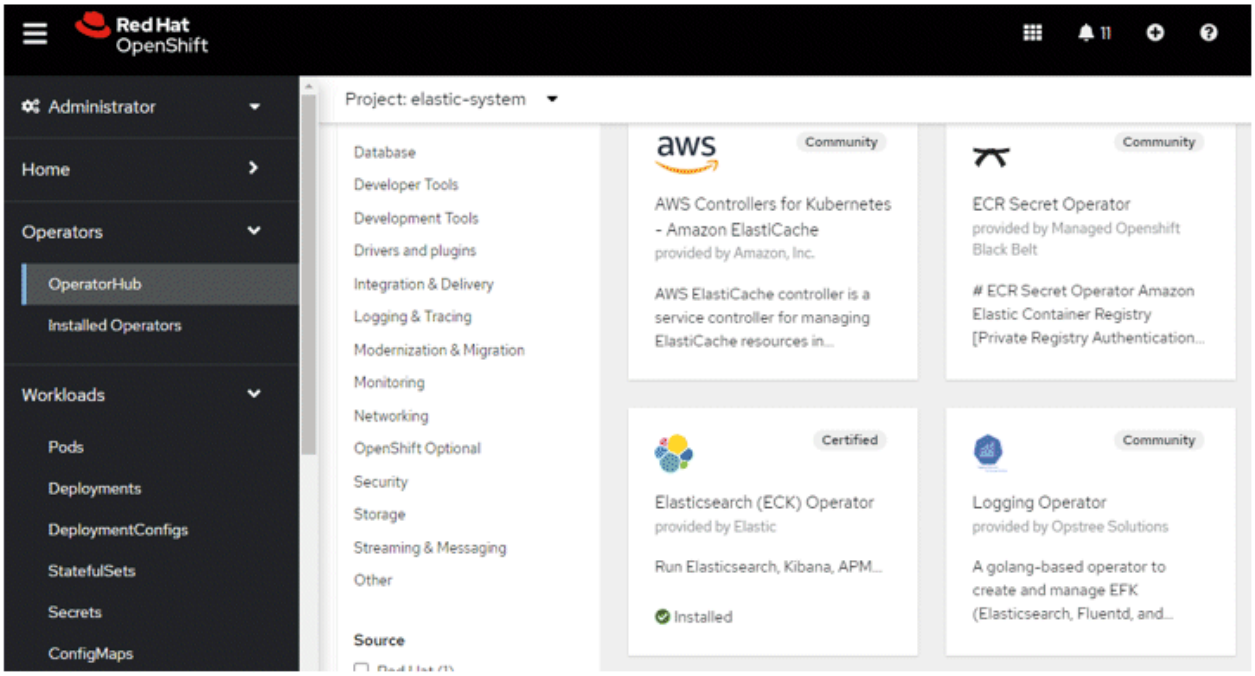
https://github.com/fujitsu/fep-operator-examples/blob/v4/Monitoring/dashboard/fep-dashboard.json

3.4.2 Implement Elastic Cloud on Kubernetes

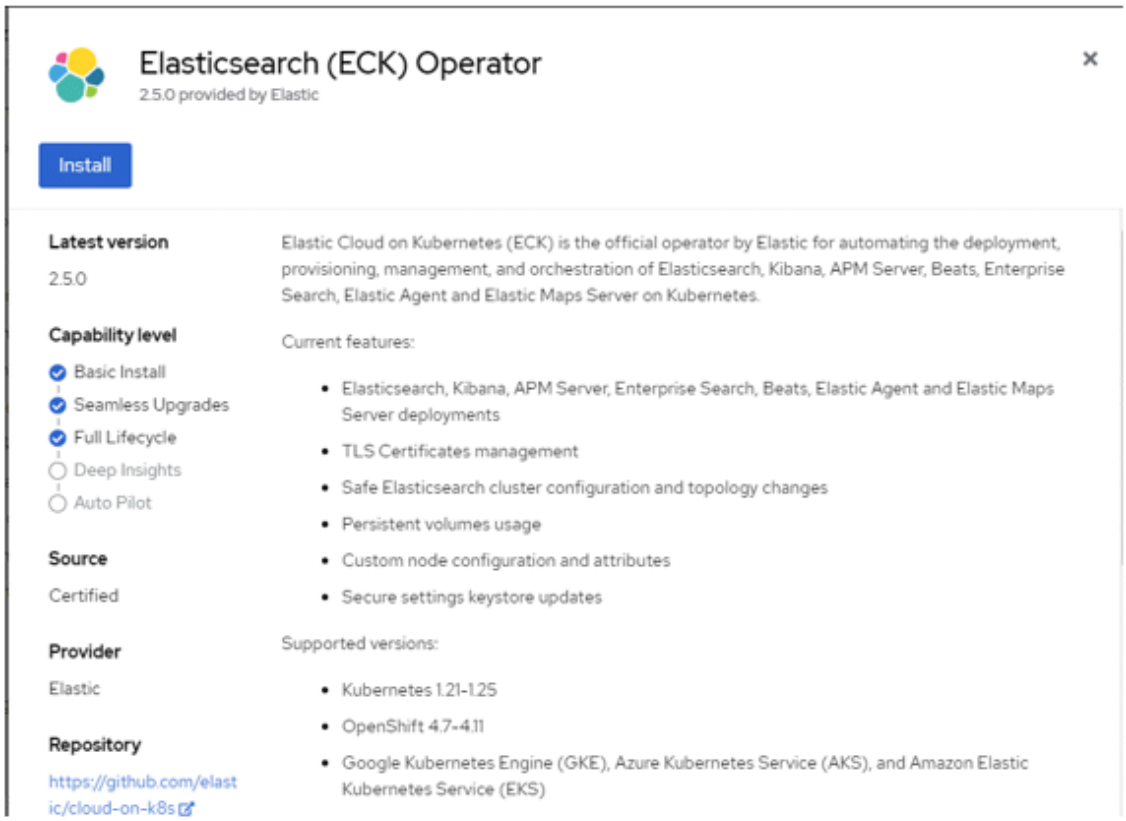
3.4.2.1 Deploy ECK Operator

1. Create namespace(project) elastic-system.

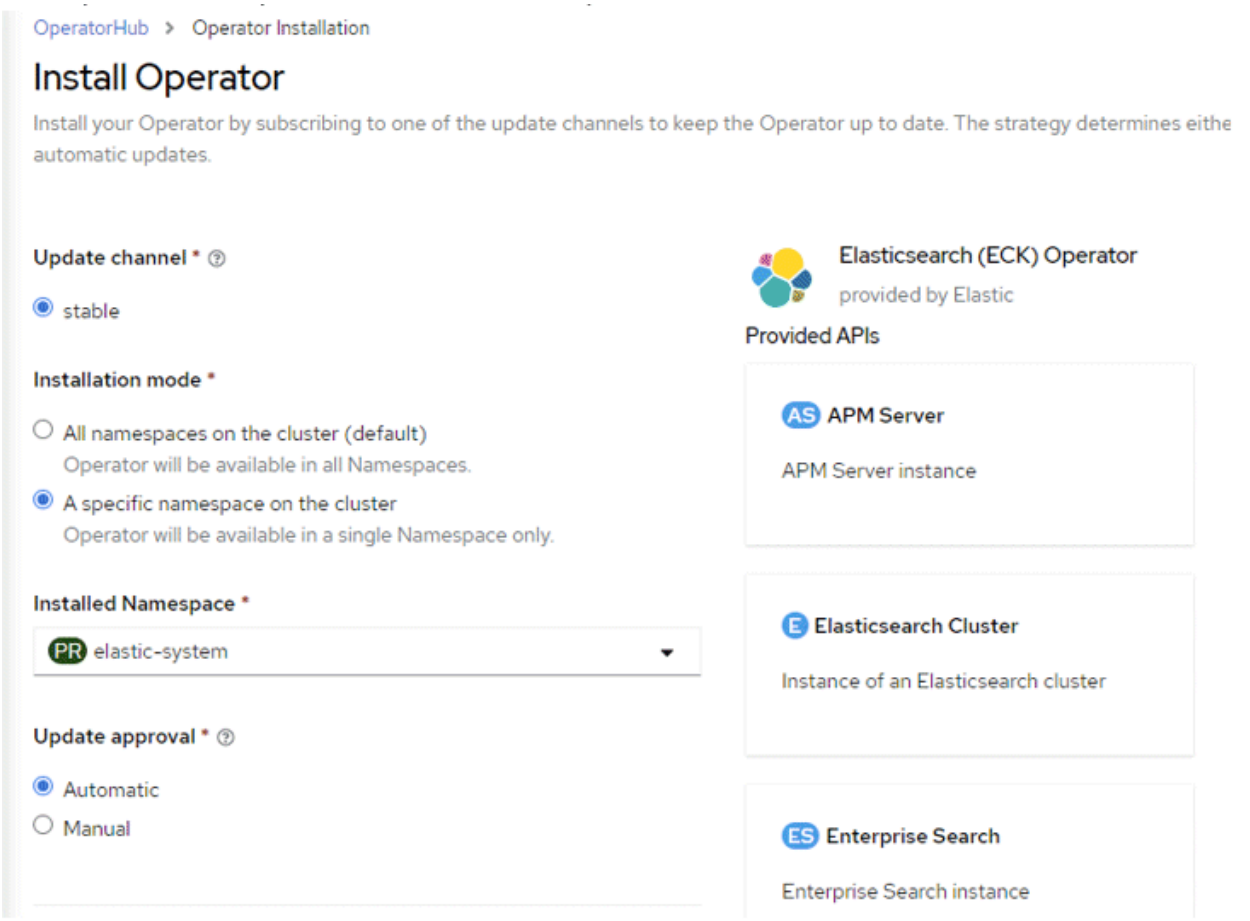
2. In OperatorHub, install Elasticsearch (ECK) Operator provided by Elastic.



3. Click Install to start proceed.

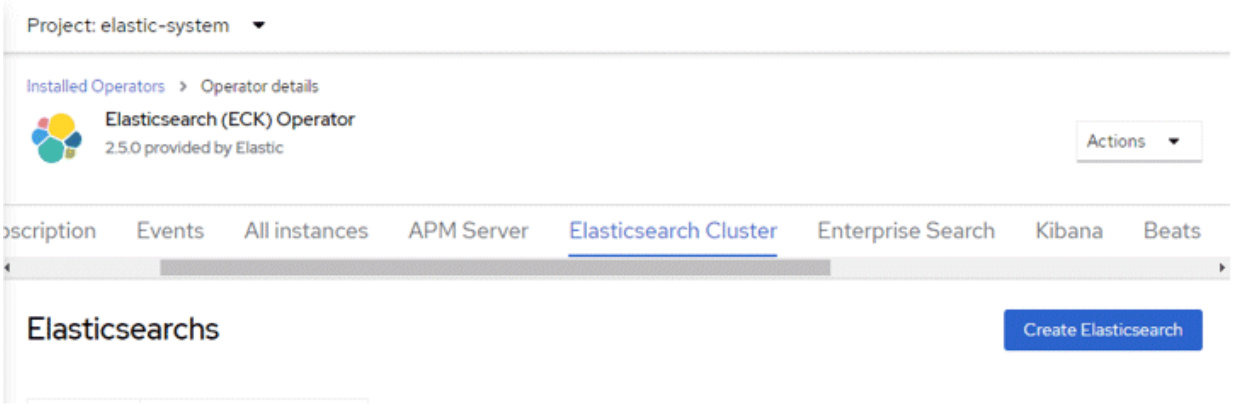


4. Change the Installation mode to "A specific namespace on the cluster" and select namespace "elastic-system". Click Install to complete the installation.



3.4.2.2 Deploy Elasticsearch Cluster

1. In Installed Operators, select "Elasticsearch (ECK) Operator".
2. Select "Elasticsearch Cluster" and "Create Elasticsearch".



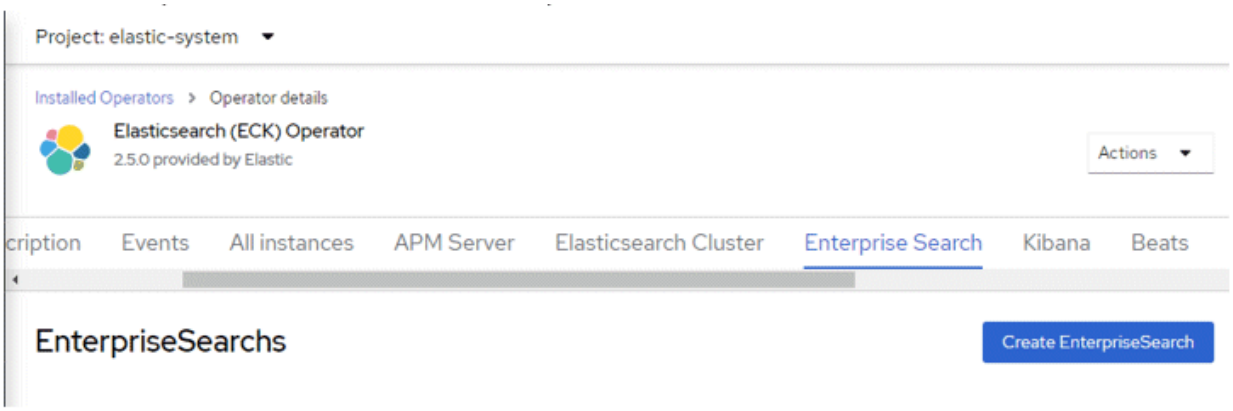
3. In YAML view, enter the following details and click "Create".

```
apiVersion: elasticsearch.k8s.elastic.co/v1
kind: Elasticsearch
metadata:
  name: quickstart
```

```
spec:
  version: 8.5.2
  nodeSets:
  - name: default
    count: 1
    config:
      node.store.allow_mmap: false
```

3.4.2.3 Deploy Enterprise Search

1. In Installed Operators, select "Elasticsearch (ECK) Operator".
2. Select "Enterprise Search" and "Create EnterpriseSearch".

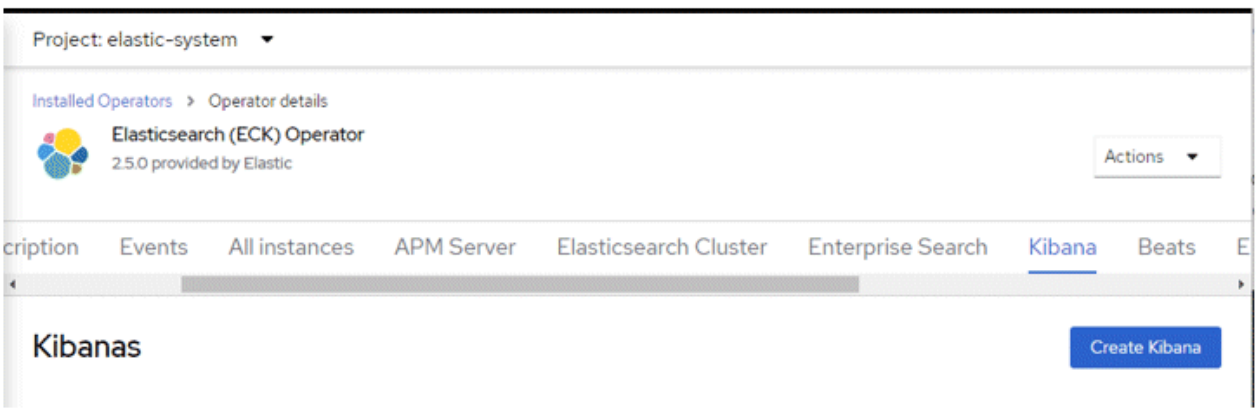


3. In YAML view, enter the following details and click "Create".

```
apiVersion: enterprisesearch.k8s.elastic.co/v1
kind: EnterpriseSearch
metadata:
  name: enterprise-search-quickstart
spec:
  version: 8.5.2
  count: 1
  elasticsearchRef:
    name: quickstart
```

3.4.2.4 Deploy Kibana

1. In Installed Operators, select "Elasticsearch (ECK) Operator".
2. Select "Kibana" and "Create Kibana".



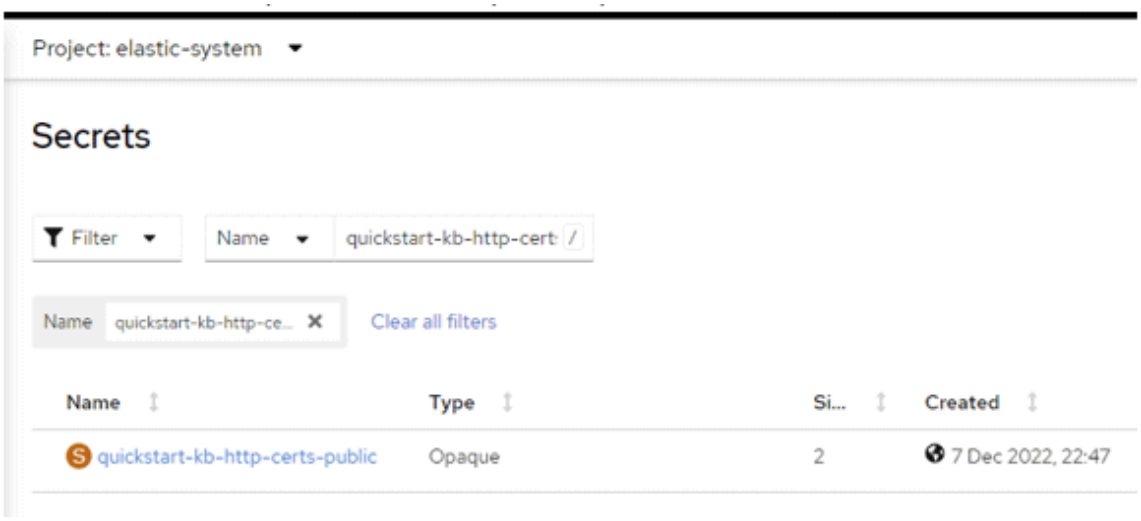
3. In YAML view, enter the following details and click "Create".

```
piVersion: kibana.k8s.elastic.co/v1
kind: Kibana
metadata:
  name: quickstart
spec:
  count: 1
  elasticsearchRef:
    name: quickstart
  enterpriseSearchRef:
    name: enterprise-search-quickstart
  version: 8.5.2
```

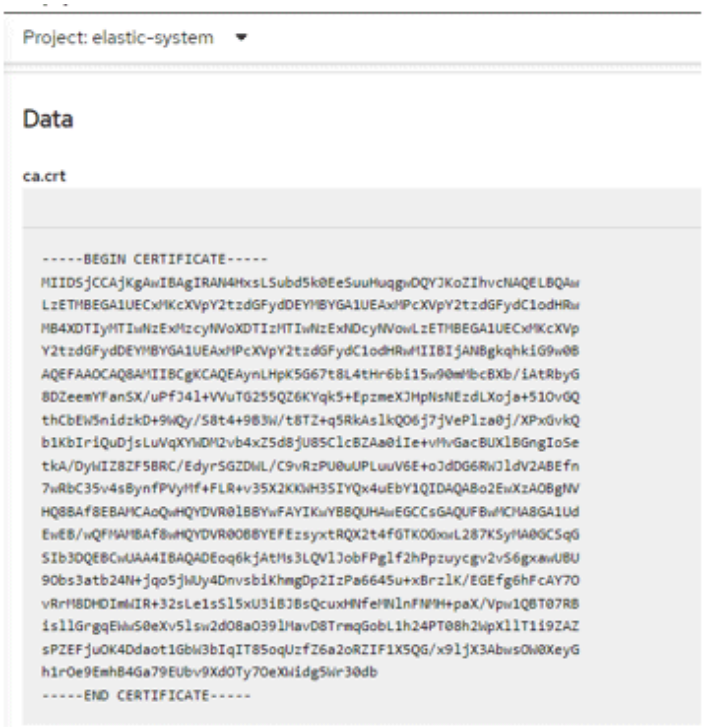
3.4.2.5 Expose Kibana using OpenShift Route

1. Obtain CA certificate that signs Kibana certificate.

   Locate the secret quickstart-kb-http-certs-public.



Copy the content of ca.crt.



2. Create route for Internet access.

   Navigate to Networking -> Route and select "Create Route".

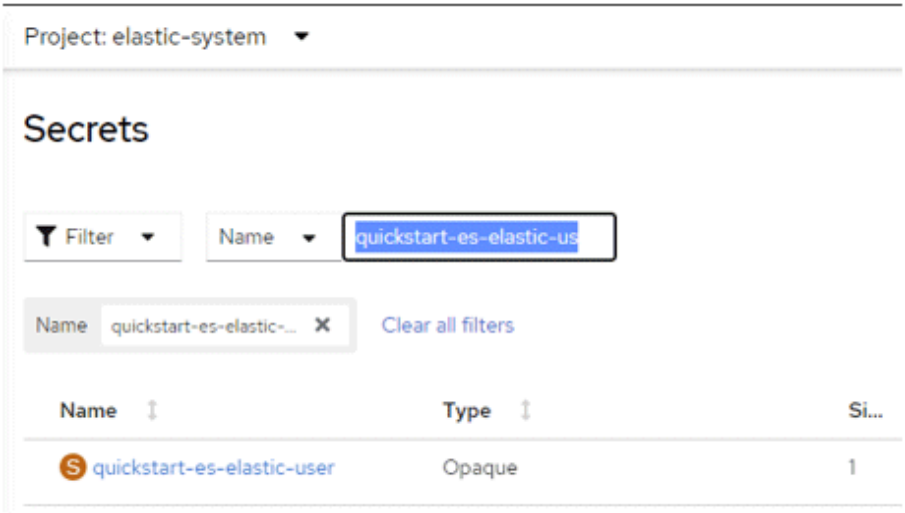
   Fill in the details

Key	Value
Name	kibana
Hostname	Leave empty
Path	/
Service	quickstart-kb-http
Target port	5601 -> 5601 (TCP)
Secure Route	selected
TLS termination	Re-encrypt
Insecure traffic	Redirect
Destination CA certificate	Content of ca.crt in previous step

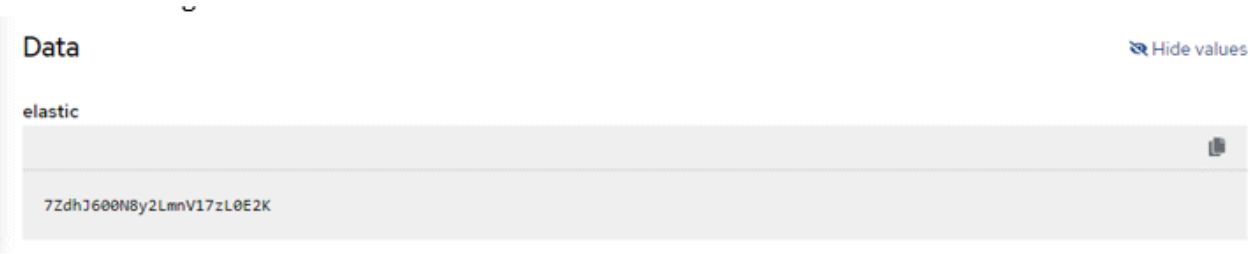
3.4.2.6 Login to Kibana

1. Obtain the Elasticsearch/Kibana login details

   Locate the secret quickstart-es-elastic-user

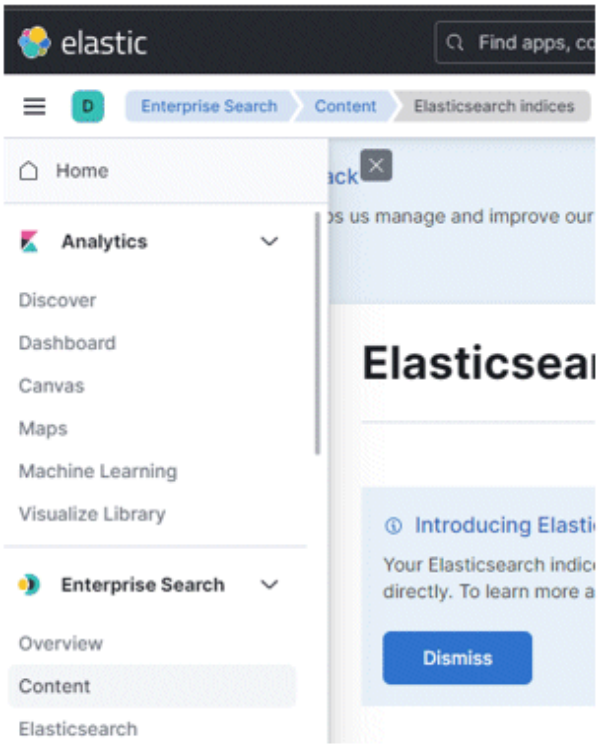


2. Observe the login details from the secret



3. Visit the URL as created in the Route above and use the above credential to login

4. Select the collapsed menu icon on top left corner and select Enterprise Search -> Content



5. If fluentd is forwarding logs to this elastic cluster, you will find the indexes here.

3.5 Implement Client

To use the FEP client, use the media or download the rpm module from the following site.

https://www.postgresql.fastware.com/fujitsu-enterprise-postgres-client-download

Chapter 4 Deployment Container

This chapter describes container deployment.

CR templates (sample files) for operating databases using Fujitsu Enterprise Postgres Operator are published in the following repository on GitHub. By using templates, you can easily deploy containers.

https://github.com/fujitsu/fep-operator-examples

Note

- Each volume of a Pod created by a FEPCluster deployment is sized by default for the following operations:
  - Data size: 1 GB
  - Daily update: about 50 MB

  Refer to "2.3.3 Configurable Volume per Cluster" to design each volume size according to actual operation.
- If you are using anti-virus software, exclude all volumes that make up the FEPCluster from virus scanning.

4.1 Deploying FEPCluster using Operator

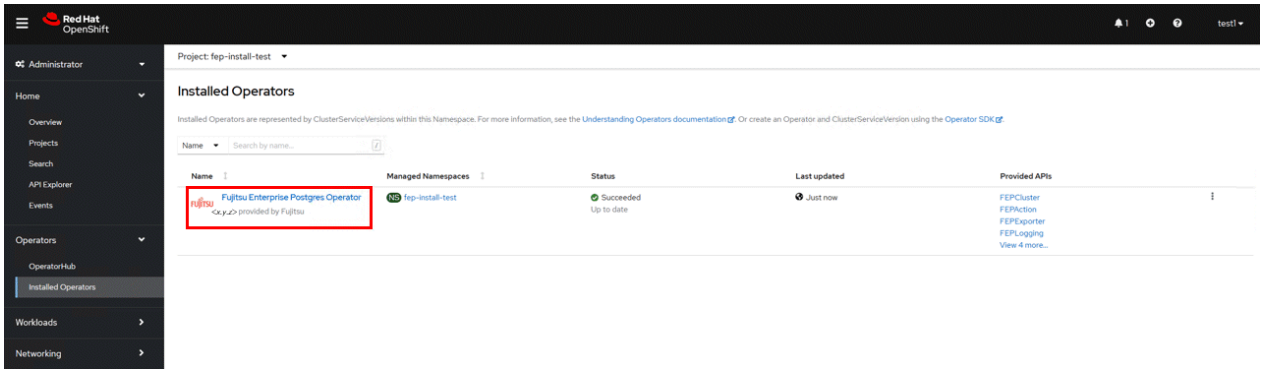
To deploy a FEPCluster in given namespace, follow these steps.

"<*x.y.z*>" in the screen example indicates the version level of the operator. Also, "<*X*>" indicates the product version of Fujitsu Enterprise Postgres.

Note

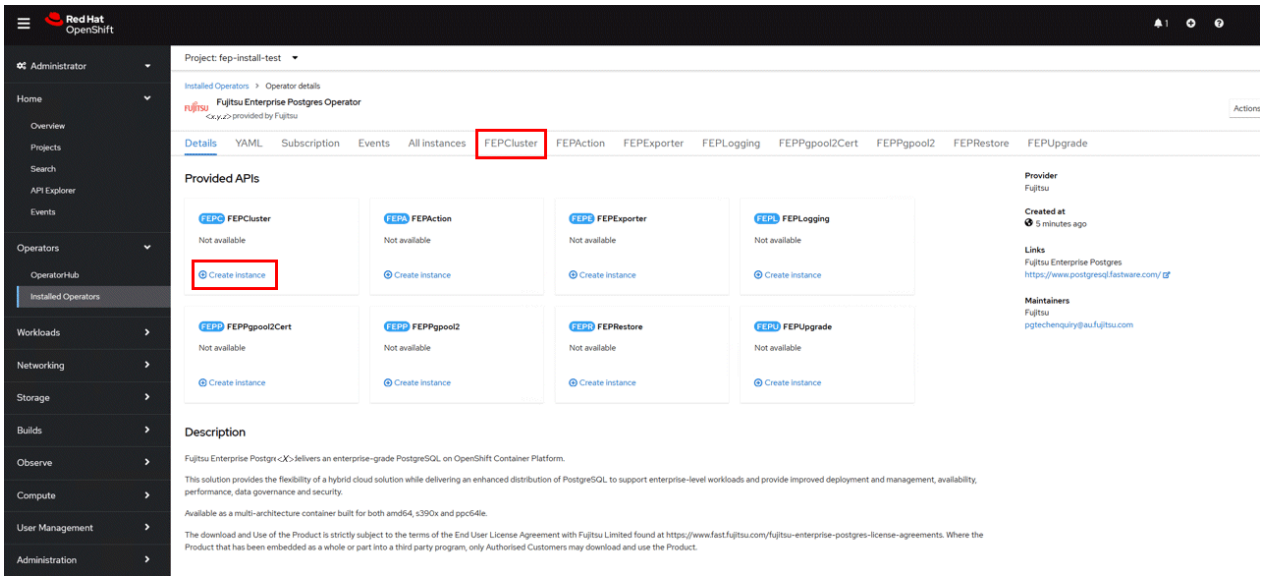
If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

1. Under "Operators" menu item, click on "**Installed Operators**". You would see the installed FEP operator deployed in "Chapter 3 Operator Installation". Click on the name of operator.

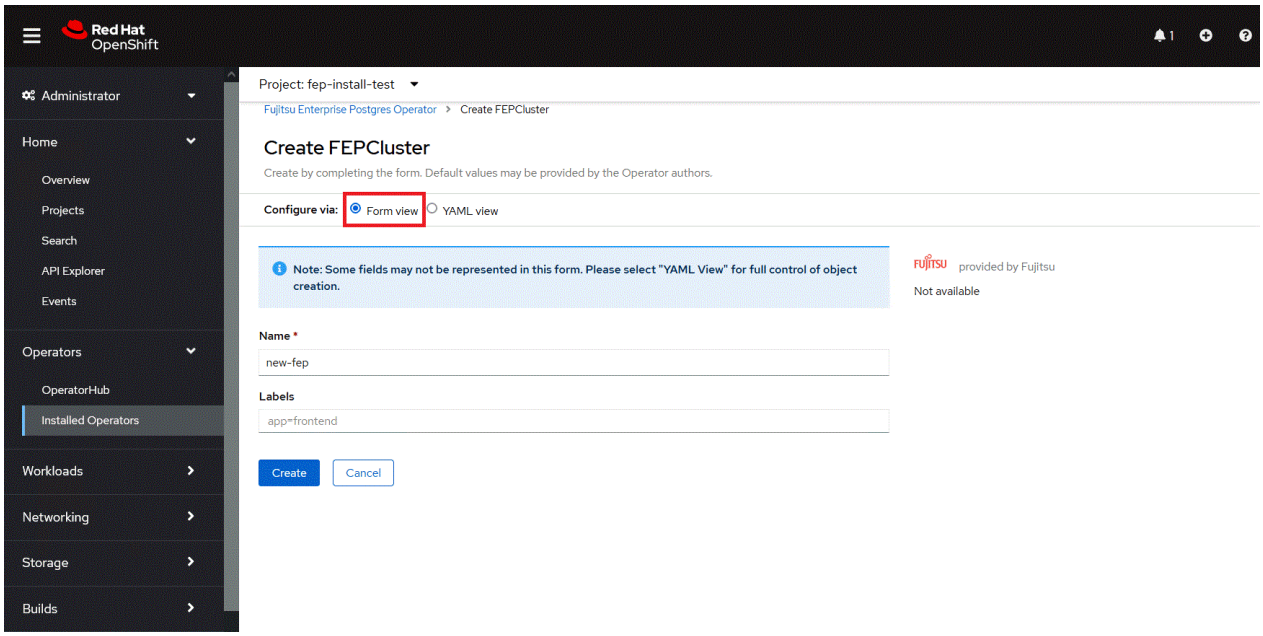


2. It will display a page with all CRs this operator supports. FEPCluster is the main CR and all others are child CR. We would create the main CR and all other CRs will be created automatically by Operator.
   To create Cluster CR, either
   (1) Click on "**Create Instance**" under FEPCluster.

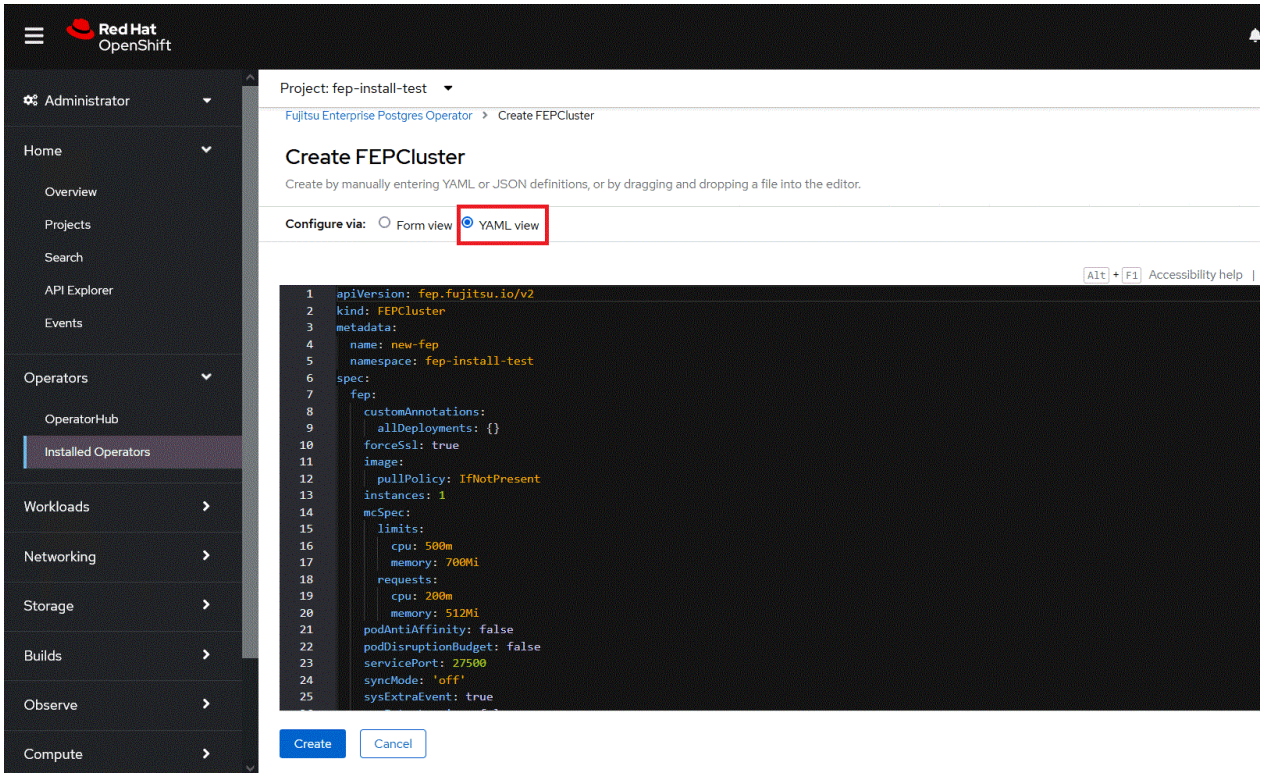
OR
(2) Click on "**FEPCluster**" on top and then click on "**Create FEPCluster**" on the next page.



3. This will bring to "Create FEPCluster" page. Here you have two options to configure. The first one is Form View. At the moment, in Form View , one can change only the name of cluster being deployed. The default name is "new-fep".
This name must be unique within a namespace.



4. In YAML View, starting value of CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.



The FEPCluster custom resource allows you to define the container's CPU, Memory, disk size, etc.

You can define each resource size individually, or you can use the following parameters to define the allocations for each resource in bulk.

Parameter	Description
spec.fep.databaseSize	Small, medium, and large define the following values for cpu/memory: small: 500m/700Mi medium: 2/4Gi large: 4/16Gi
spec.fepChildCrVal.storage.dataSize	Specifies the size of the data storage PV. Estimate and define the size of the backup storage area when the backup is enabled

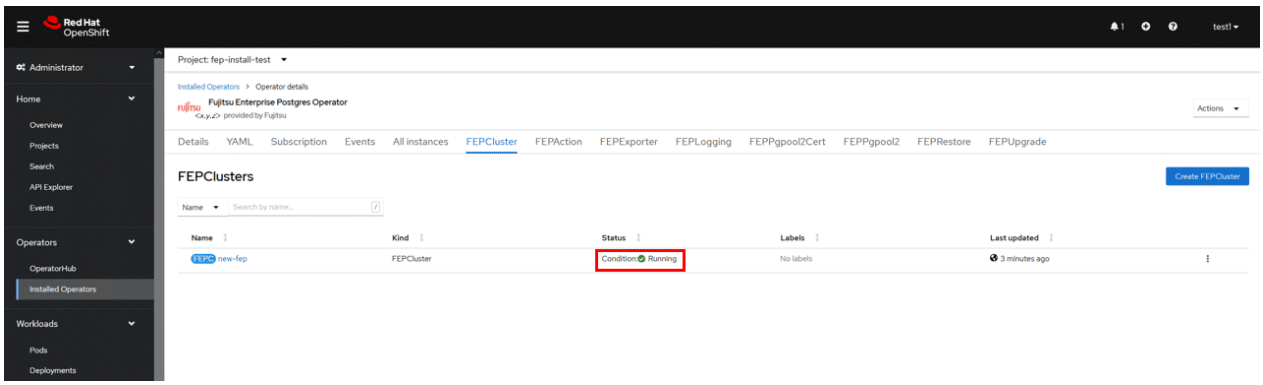
We recommend that you use transparent data encryption to store your data.
Tablespace PVs are mounted in/database/tablespaces/tbspace1.
After building the database cluster, create tablespaces and tables as follows:

```
# Creating an Encrypted Tablespace
CREATE TABLESPACE secure_tablespace LOCATION '/database/tablespaces/tbspace1' WITH
(tablespace_encryption_algorithm ='AES256');

# Create Table in Encrypted Tablespace
CREATE TABLE secure_table (id int, pref text, city text, data text) TABLESPACE secure_tablespace;
```

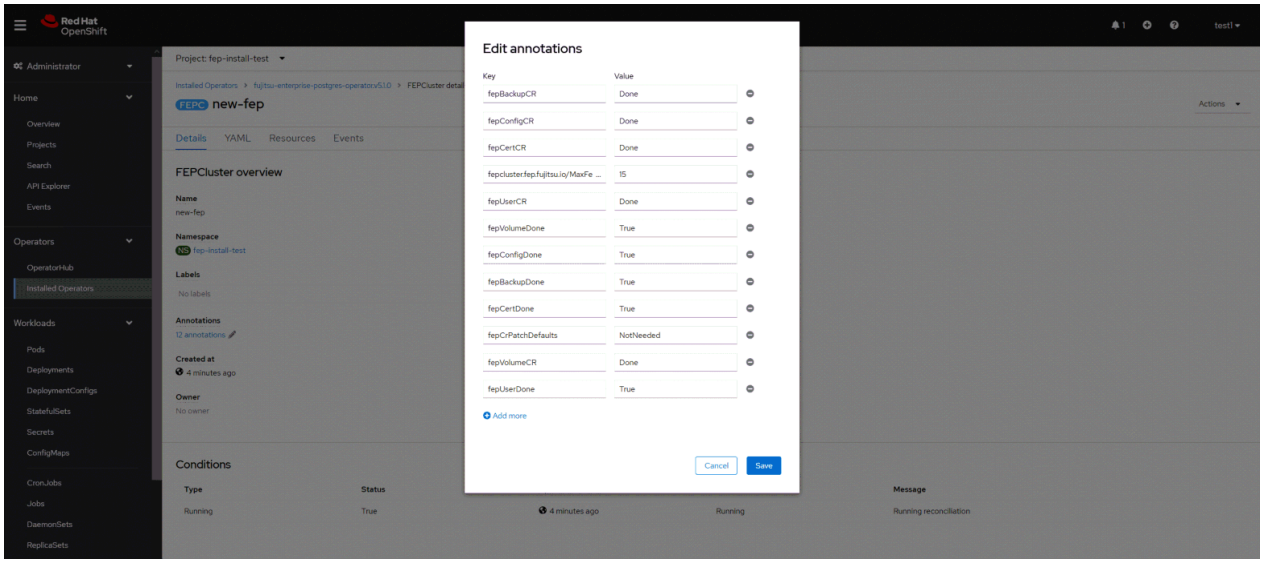
Operator provides features such as backup, audit logging, and monitoring.
These features can be enabled with the following parameters:

Parameter	Description
spec.fep.monitoring.enable	When true, monitoring feature is enabled.
spec.fep.pgAuditLog.enable	When true, audit log collection is enabled.
spec.fepChildCrVal.backup.type	local enables backup. The backup data is stored in the PV.
spec.fepChildCrVal.autoscale.scaleout.policy	cpu_utilization enables the ability to autoscale out replicas when CPU utilization exceeds a threshold.

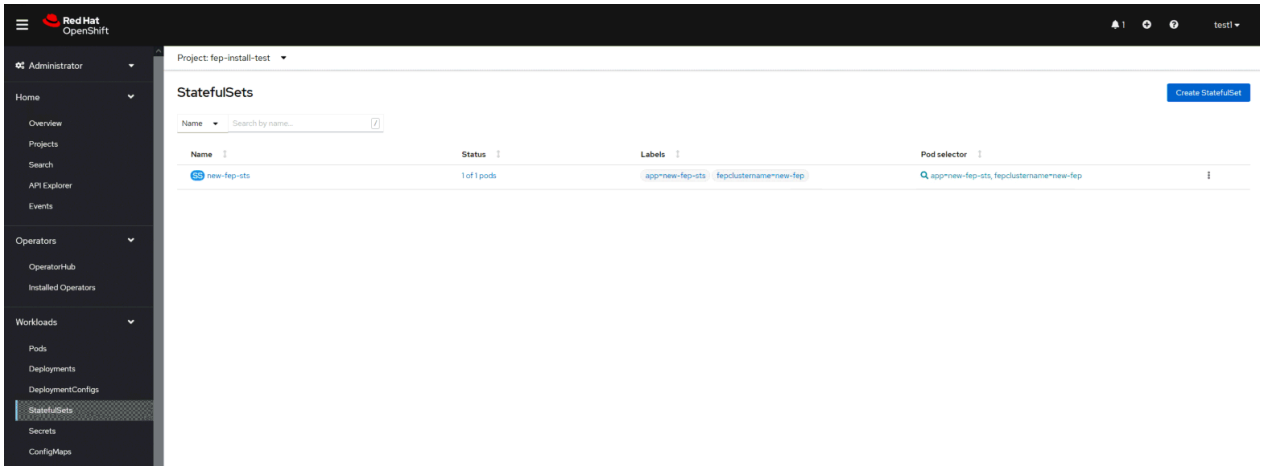
5. When "Create" is clicked on either of the two pages above, the operator creates FEPCluster CR, and there after one by one FEPBackup, FEPConfig, FEPVolume, FEPUser, and FEPCert child CRs are created automatically.
   The starting values for child CRs are taken from the "fepChildCrVal" section of the FEPCluster CR YAML file. Modifying value in FEPCluster "fepChildCrVal" section. Operator reflects changes from FEPCluster parent CR to respective child CRs. Only allowable changes are reflected in child CRs. Child CRs are marked internal objects and hence will not be visible on the OCP console. However, you can check child CRs using command-line tools.



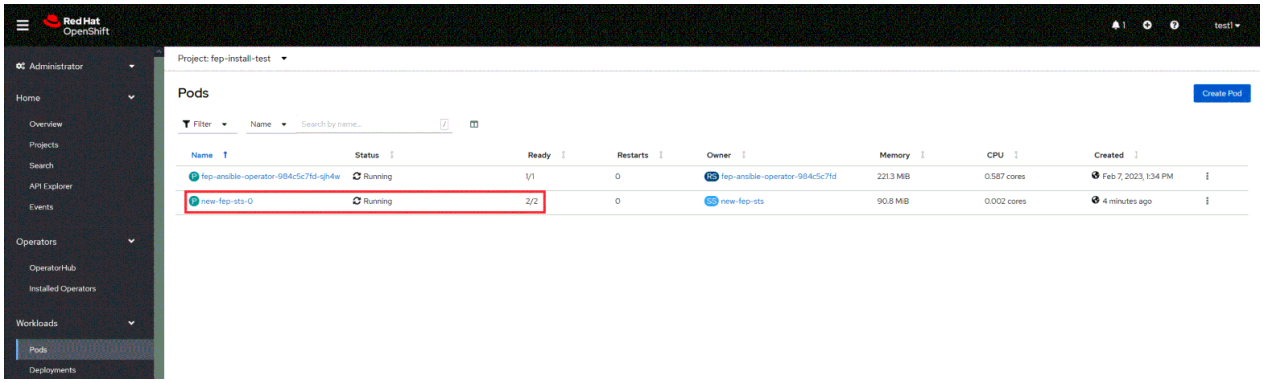
6. In FEPCluster CR, annotations are added to indicate that child CRs are created successfully and has initialised properly. It may take some time to complete.



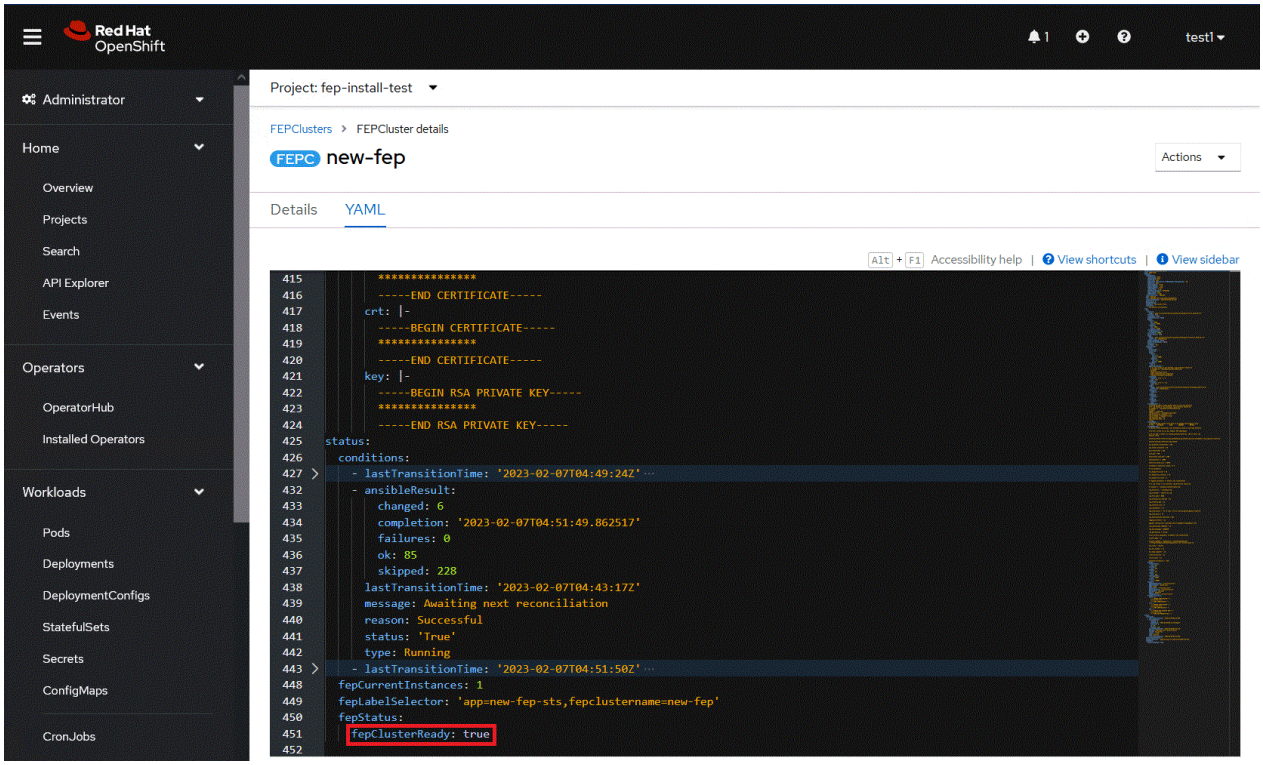
7. Once child CRs are marked done in annotations, operator creates StatefulSet for the cluster.



8. StatefulSet will start one FEP instance at one time and will wait it to be ready before starting next one.



9. Once all instances of FEP servers are started, the operator marks a flag "fepClusterReady" under "status.fepStatus" section of CR to be **true**, indicating that FEPCluster is ready for use. Looking at YAML of FEPCluster CR, it would look like as below:



10. Operator also masks the sensitive fields like passwords, passphrase, certificates and keys in FEPCluster fepChildCrVal and also in respective child CRs.

4.2 Deploy a Highly Available FEPCluster

In a highly available FEP cluster, load balancing is possible by distributing read queries to replica instances.

In addition, if the master instance fails, the user can switch to the replica instance immediately to localize the business interruption period.

In a highly available configuration, you can select the synchronization mode for the replica instance. Synchronous replication is recommended for systems that cannot tolerate data loss in the event of a master instance failure.

Because multiple instances are created in a highly available configuration, licenses are required for each.

To deploy a highly available FEPCluster in given namespace, follow these steps:

[Prerequisites]

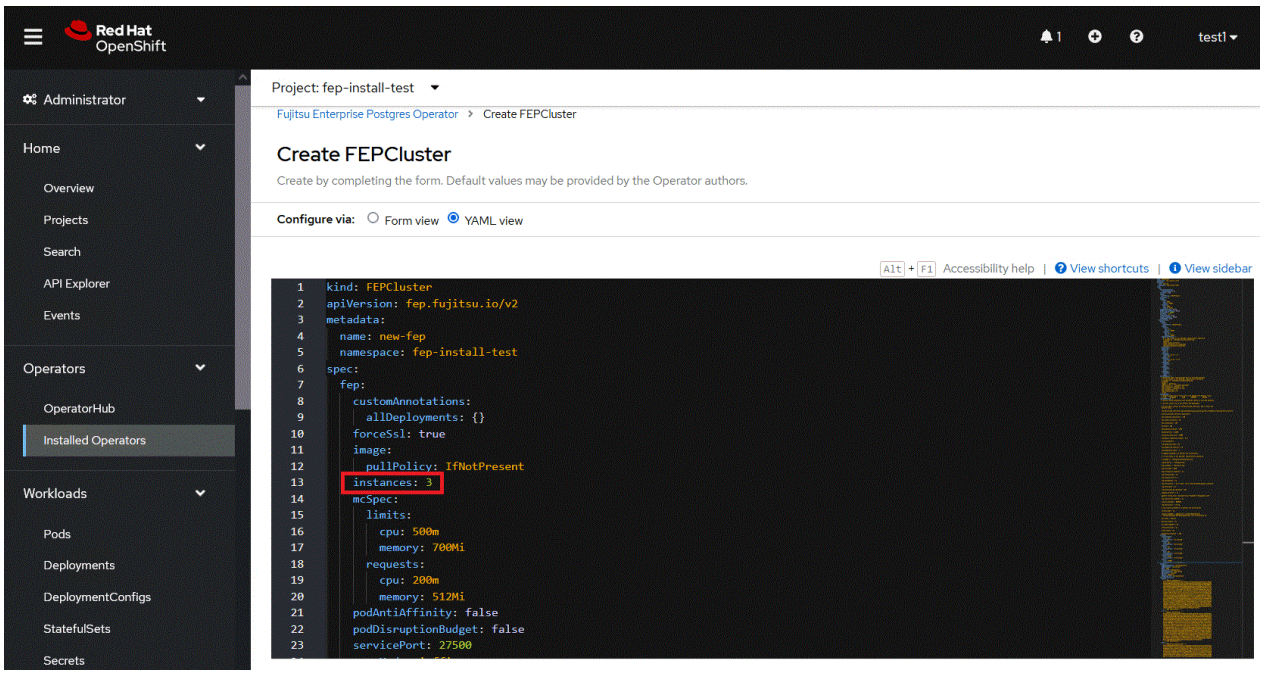
If the FEP cluster is running in HA mode, the backup and archive WAL volumes must be configured with shared storage (NFS, etc.) that supports ReadWriteMany. See the Openshift documentation for instructions on setting up shared storage. Also, the reference procedure is described in "Appendix C Utilize Shared Storage", so please check if necessary.

If you do not have shared storage, you can remove the backup section and the backup and archive volume sections to disable the backup feature and deploy the FEP cluster.

Note

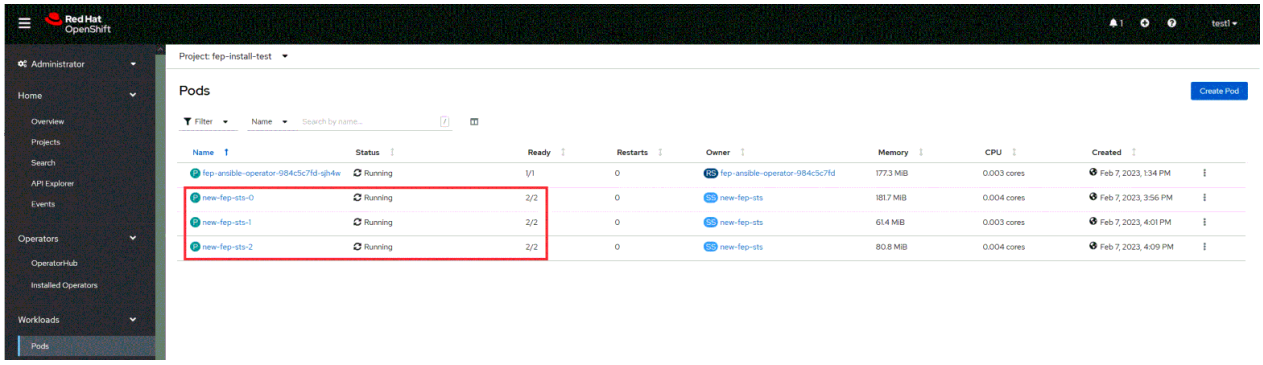
If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

1. It is the same as the procedure from step 1 to step 3 in "4.1 Deploying FEPCluster using Operator".
2. Instead of step 4 in "4.1 Deploying FEPCluster using Operator", change to the YAML view and specify '3' for the "instances" parameter of "fep" in "spec". Specify the storage class for the prepared shared storage for the backup and archive WAL volumes.



3. It is the same as the procedure from step 5 to step 10 in "4.1 Deploying FEPCluster using Operator".

4. Three pods deployed and ready for a highly available FEPCluster.



Information

You can determine whether the master or replica pod is the master or replica pod by issuing the following command:

```
$ oc get pod -L feprole
NAME                                     READY   STATUS    RESTARTS   AGE   FEPROLE
fep-ansible-operator-88f7fb4b-5jh85      1/1     Running   0          24m
new-fep-sts-0                            2/2     Running   0          17m   master
new-fep-sts-1                            2/2     Running   0          15m   replica
new-fep-sts-2                            2/2     Running   0          13m   replica
```

4.3 Deploying FEPExporter

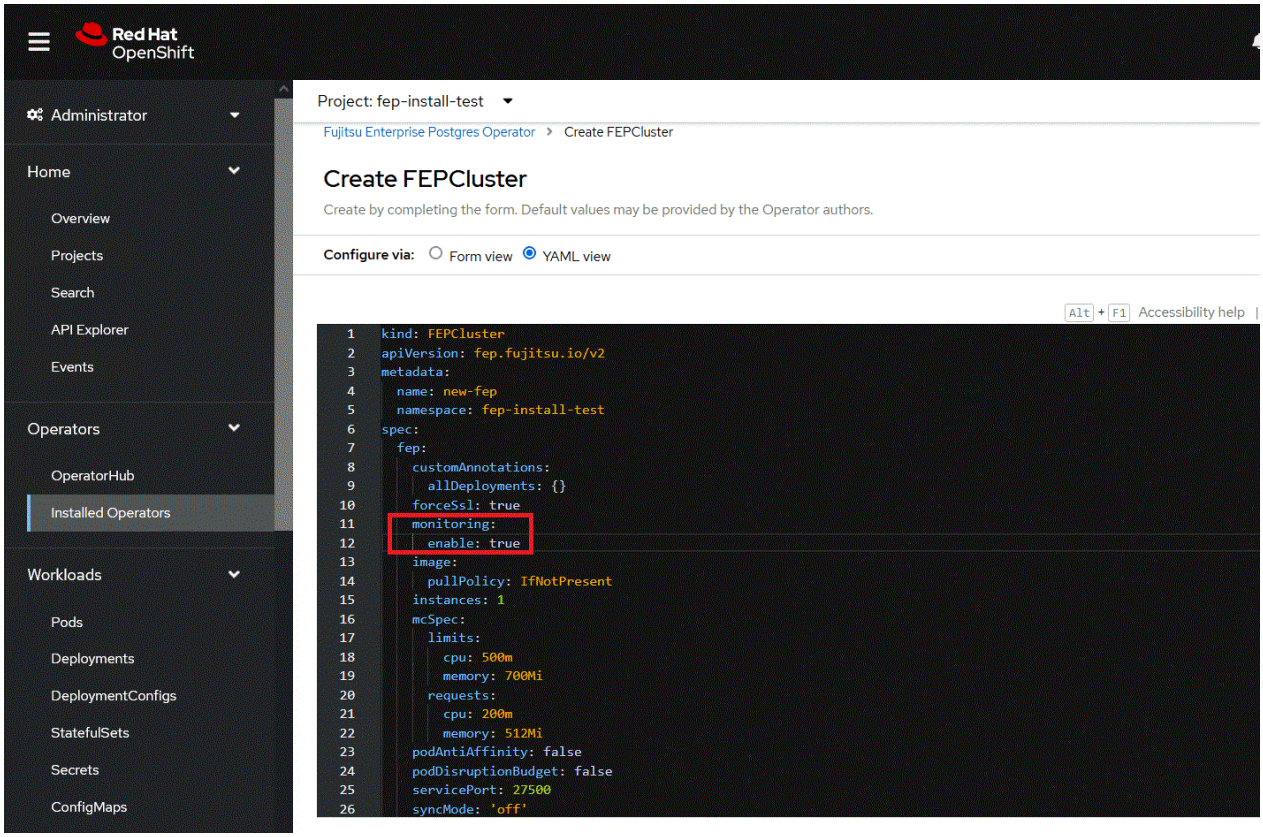
To deploy a FEPExporter, follow these steps.

"<*x.y.z*>" in the screen example indicates the version level of the operator.

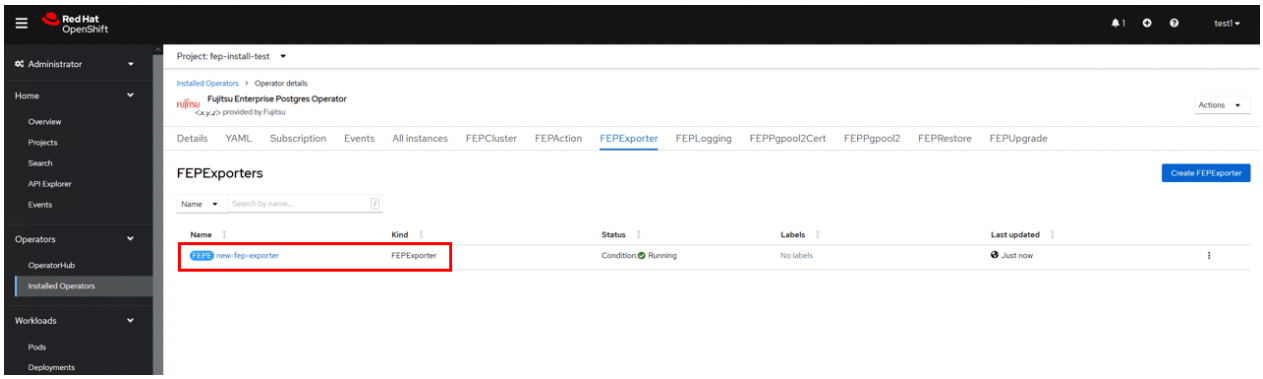
Note

If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

1. In order to deploy FEPExporter managed by Operator, it is as easy as setting fep.monitoring.enable to true in FEPCluster CR at the time of deployment.



2. FEPExporter will be created automatically under the name <cluster-name>-fepexporter. And it will list show all the database with statistics of specified FEPcluster.



3. FEPExporter spawned by FEP Operator in aforementioned way will scrape metrics by default from the Master and standby instances and make it available to Prometheus.

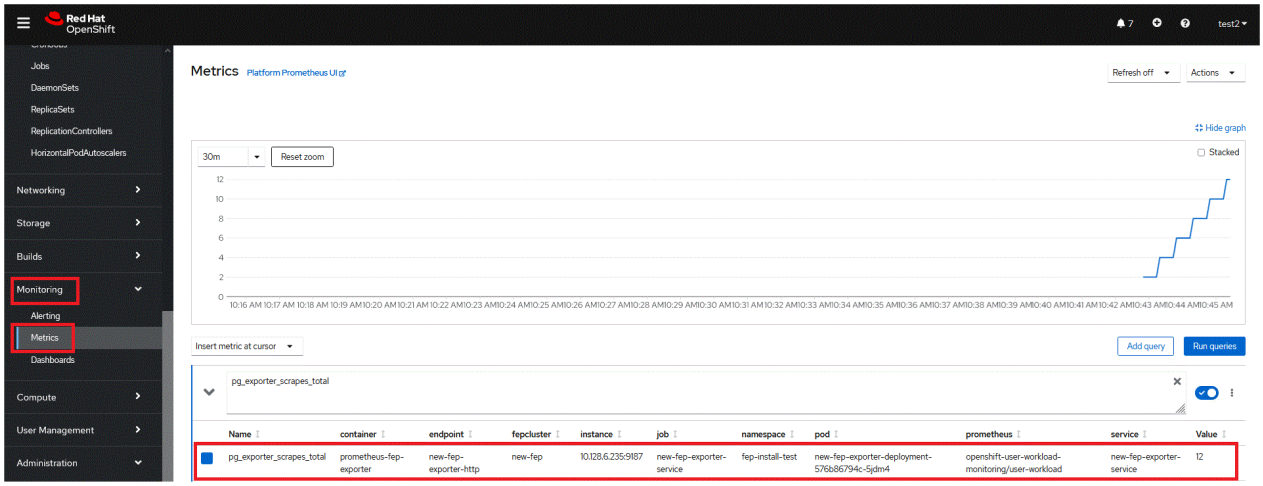
4. User can configure MTLS to be used for HTTP endpoint used by Prometheus for metrics scraping as well as connection from FEP Exporter to database.

    a. If pgMetricsUser, pgMetricsPassword and pgMetricsUserTls is defined in FEPcluster; FEPExporter will hence use these for securing connection to the postgres instances. In absence of these parameters, FEPExporter will use pgAdminUser (i.e. super user).

    b. User can configure Prometheus.tls and FEPExporter.tls to ensure that metrics end point (/metrics) by FEPExporter is also used with MTLS (Refer to "FEPExporter Custom Resource" in the Reference for details of fields)

5. User can also configure basic authentication by specifying a secret that contains username & password. (Refer to "FEPExporter Custom Resource" in the Reference for details of fields)

6. Now user can see scrape FEPExporter specific metrics on Openshift Platform in monitoring section area using PROMQL to specify a metrics of interest



Note

- User can set fep.monitoring.enable to true or false on an already instantiated cluster as well to achieve desired results
- pgMetricsUser can be defined later on a running FEPCluster with monitoring enabled and can force FEPExporter to use pgMetricsUser by mere restarting it (refer restartRequired). However, MTLS can not be configured in this case and user is expected to grant specific permission to pgMetricsUser for all the database objects which are expected to be use while scraping information.
- For MTLS to be forced, ensure usePodName and pg_hba.conf is been set appropriately.
- FEPExporter default metrics expects few following in postgresql.conf
  - pg_stats_statements library pre-loaded
  - **track_activities** and **track_counts** are turned on
  - Monitoring user needs permission on pg_stat_* views
- FEPExporter pod specification related to CPU memory can be changed. After changing resources specification, set **restartRequired** flag to true. FEPexporter will be restarted with new specifications
- FEP Monitoring is closely integrated with Prometheus available on platform. User should ensure that on openshift platform monitoring is enabled for user-defined projects (Refer: https://docs.openshift.com/container-platform/4.11/monitoring/enabling-monitoring-for-user-defined-projects.html). For platforms other than openshift, ensure Prometheus is installed before deployment of FEP operator

4.4 FEPExporter in Standalone Mode

FEPExporter is an independent CR; hence it does not necessarily depend on main FEPCluster CR. To deploy a FEPExporter in given namespace follow the below step.

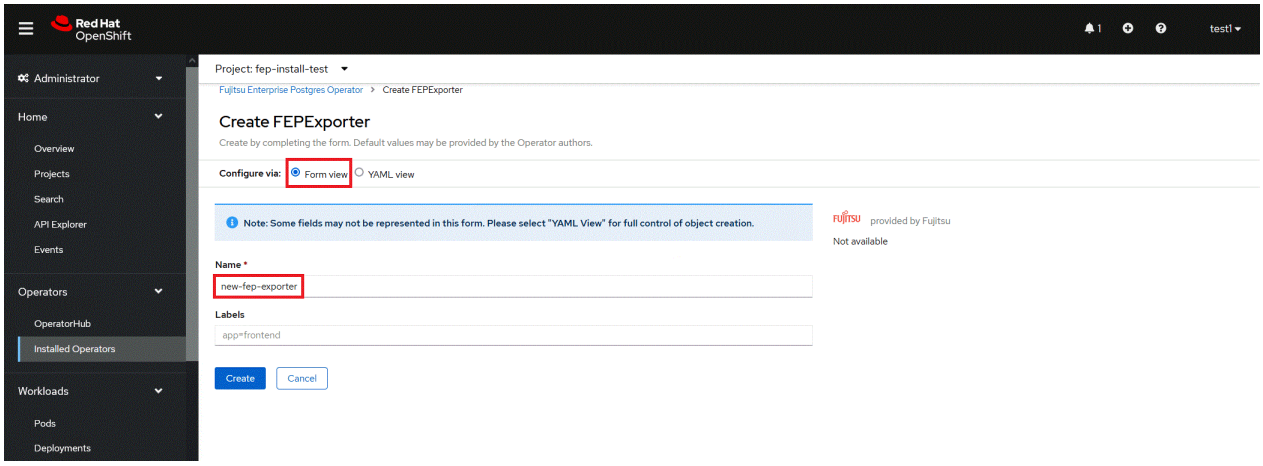
Note

If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

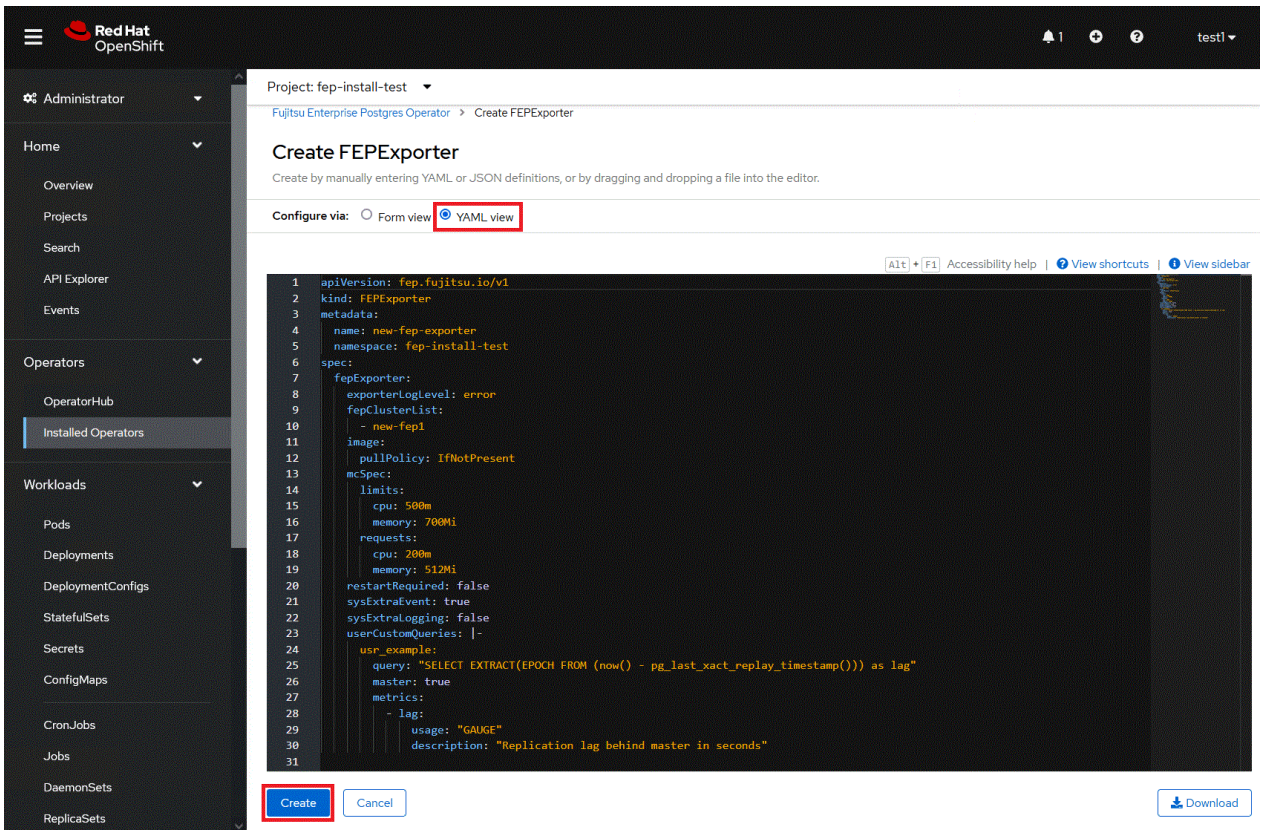
1. To create FEPExporter CR, either
   (1) Click on "**Create Instance**" under FEPExporter.
   OR
   (2) Click on "**FEPExporter**" on top and then click on "**Create FEPExporter**" on the next page.

2. In Form View, one can change only the name of cluster being deployed. The default name is "new-fep-exporter". This name must be unique within a namespace.

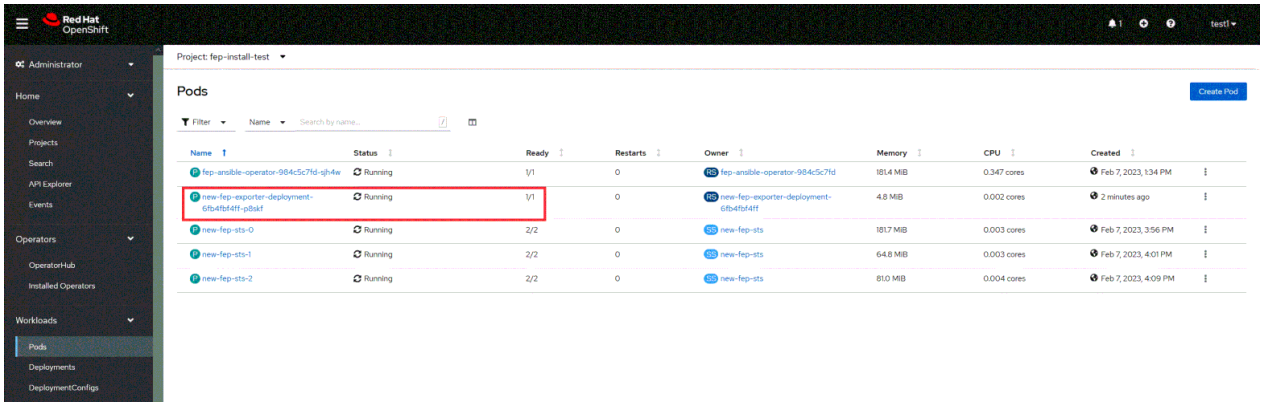
3. FEPExporter scrapes metrics for FEPCluster within same namespace.



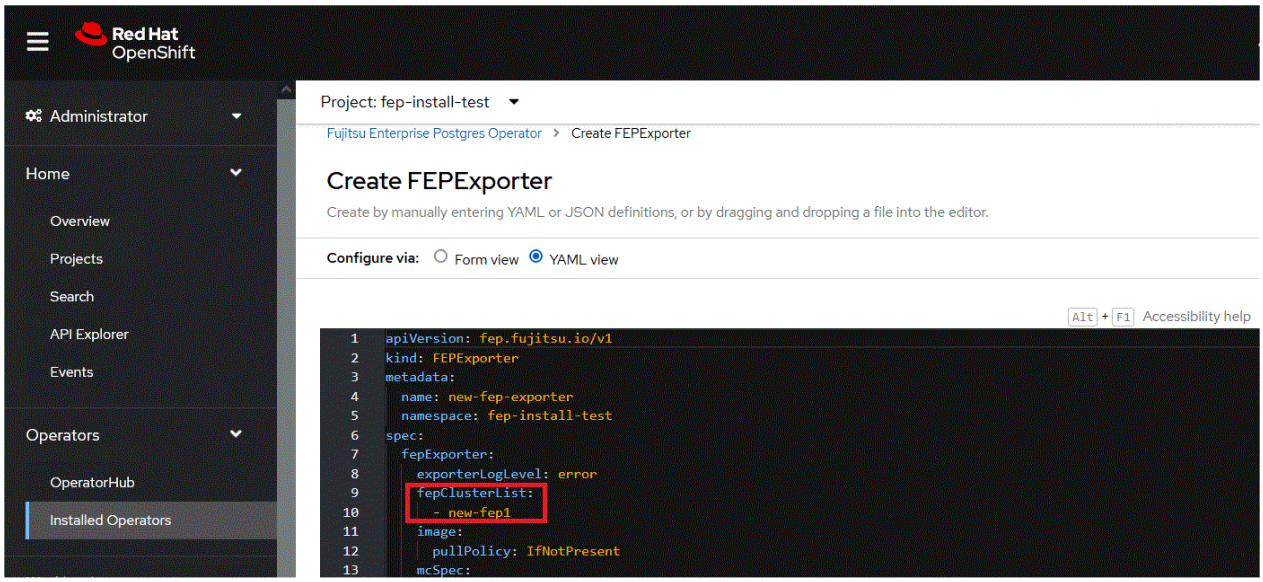
4. In YAML View, starting value of FEPExporter CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.



5. When clicked on the "Create" button. It will create FEPExporter pod with other resource like secret, service, configmap for data source queries.



6. Specify the name of the FEPCluster in spec.fepExporter.fepClusterList of FEPExporter. Before targeting cluster, Check the FEPCluster status and FEP StatefulSet are in running condition.



7. It will recreate FEPExporter pod with a new dataresource secret. It will list down all the database with statistics of specified FEPcluster in monitoring section.
8. If fepClusterList has more than one clusters listed, current exporter will collect metrics for all of those listed.
9. Multiple FEPExporters can be deployed within one namespace with their own cluster list to collect metrics from.

4.5 How to Collaborate with CloudWatch

Describe how to integrate with CloudWatch.

4.5.1 Preparation

Save credentials and other settings required for linking with CloudWatch in Secrets and ConfigMaps.

Prepare two files, credentials and config, which describe credentials and other settings according to the format specified by the AWS client interface. Specifying access_key_id and secret_access_key in the credentials file is mandatory.

An example of registering authentication information using the following configuration file is explained.

```
credentials   # credentials file
config        # config file
```

Create a ConfigMap to store config files. Specify config for the key name. The name of the ConfigMap is arbitrary (here my-aws-config).

```
$ oc create configmap my-aws-config --from-file=config=config -n my-namespace
```

Create a secret to save the credentials file. Specify credentials for the key name. The name of the Secret is arbitrary (here my-aws-credentials).

```
$ oc create secret generic my-aws-credentials --from-file=credentials=credentials -n my-namespace
```

See

Refer to below for AWS client interface configuration files.

https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html

4.5.2 FEPCluster Configuration

Add the following settings to the FEPClusterCR and deploy it.

- spec.fep.externalMonitoring.cloudWatch: Definition for forwarding metrics
- spec.fep.externalMonitoring.cloudWatch.authentication: Definition for authentication information to CloudWatch

Example of FEPClusterCR definition

```
spec:
  fep:
    externalMonitoring:
      cloudWatch:
        enable: true
        schedule: "0-59/10 * * * *"
        namespace:FEP_METRICS
        customMetrics:
          - config-map-query
        authentication:
          cloudWatchCredentials: my-aws-credentials
          cloudWatchConfig: my-aws-config
```

If spec.fep.externalMonitoring.cloudWatch.enable is true, FEP metrics are forwarded to CloudWatch as defined by spec.fep.externalMonitoring.cloudWatch.schedule.

This feature collects and transfers metrics from all FEPCluster Pods. When transferring metrics, the dimension name is specified as instance, and the value is specified as <kubernetesNamespace>-<FEP Pod name>. If you want to define additional dimensions, specify the dimensionName and dimensionValue under spec.fep.externalMonitoring.cloudWatch.

For information about the parameters under spec.fep.externalMonitoring, refer to "FEPCluster Parameter" in the Reference.

4.5.3 Forwarding Custom Metrics

If you want to forward metrics other than "5.7.2.5 Metrics Collected by CloudWatch" to CloudWatch, specify a file containing a query in spec.fep.externalMonitoring.cloudWatch.customMetrics. The query in the file can be read and executed, and the results can be forwarded to CloudWatch. Below is an example of a ConfigMap definition.

Example of ConfigMap "config-map-query**" definition**

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: config-map-query
data:
  metricsName: <Any name>
  dimensionColumns: |
    - <Columns that can be obtained with metricsQuery>
```

```
    - <Columns that can be obtained with metricsQuery>
  metricsColumns: |
    - <Columns that can be obtained with metricsQuery>: <Metrics unit>
    - <Columns that can be obtained with metricsQuery>: <Metrics unit>
  metricsQuery.sql: |
    SELECT <Columns specified in dimension_columns and metrics_columns> FROM …
```

Create a ConfigMap that defines custom metrics for each query.

Details of each parameter specified in the ConfigMap are shown below.

Item	Description
metricsName	Optional Specify any metrics name. Metrics will be forwarded to CloudWatch with the metrics name <metricsName>_<metricsColumns>.
dimensionColumns	Optional Specify the column to be used for the Dimension from the column obtained by the query specified in metricsQuery. The column name is used as the key, and the value obtained by the query is used as the value, and the dimension is registered in CloudWatch.
metricsColumns	Specify the columns to be used as metrics from the columns retrieved by the query specified in metricsQuery in the following format. - *Column name*: *Metrics unit* If a column name or metricsName is specified, <metricsName>_<column name> will be transferred to CloudWatch as the metric name.
metricsQuery.sql	Define the query to get metrics from PostgreSQL. Create the query so that the columns of the retrieved data include the columns specified in dimensionColumns and metricsColumns.

4.6 Deploying FEPClusters with Cloud-based Secret Management

Note

The cloud-based secret management feature cannot be used together with the following parameters.

- spec.fepChildCrVal.sysUsers.pgSecurityUser
- spec.fepChildCrVal.sysTde.tdek
- spec.fepChildCrVal.sysUsers.passwordValid.days

4.6.1 Installing Secret Store CSI Driver Using Helm Charts

Install Secret Store CSI Driver from Helm chart.

Add helm chart repository.

```
helm repo add secrets-store-csi-driver https://kubernetes-sigs.github.io/secrets-store-csi-driver/
charts
```

Install with helm command.

```
helm install csi-secrets-store secrets-store-csi-driver/secrets-store-csi-driver --namespace kube-system --set enableSecretRotation=true --set rotationPollInterval=30s
```

Information

- Setting enableSecretRotation=true enables auto rotation of secret. i.e if value of secret gets changed in one of the external secret store (Azure/AWS/GCP/HashiCorp vault) then the updated value will be reflected in the FEPCluster as well.
- Setting rotationPollInterval=30s enables rotation poll interval which checks how frequently the mounted secrets for all pods need to be resynced to the latest.
- For OpenShift cluster to allow CSI type volumes to be mounted in container,system Security Context Constraints needs to be patched. Patch the volumes section to include CSI for providers(nonroot,anyuid,hostmount-anyuid,machine-api-termination-handler,hostaccess,node-exporter,privileged,privileged-genevalogging,restricted).
- In scearios where existing OpenShift is upgraded kindly verify that CSI is included in system Security Context Constraints for the above mentioned providers.

4.6.2 Installing and Configuring Azure Provider for Secret Store CSI Driver

4.6.2.1 Install Azure Provider drivers using helm chart

```
helm repo add csi-secrets-store-provider-azure https://azure.github.io/secrets-store-csi-driver-provider-azure/charts
```

Note: By default when installing Azure Provider ; secret-store-csi-driver installation is set to true by default. If secret-store-csi-driver is already installed as per steps in "4.6.1 Installing Secret Store CSI Driver Using Helm Charts" execute below command.

```
helm install csi csi-secrets-store-provider-azure/csi-secrets-store-provider-azure --namespace kube-system --set secrets-store-csi-driver.install=false
```

Note: If secret-store-csi-driver is not installed as per step "4.6.1 Installing Secret Store CSI Driver Using Helm Charts". Execute below command to install azure provider along with secret-store-csi-driver.

```
helm install csi csi-secrets-store-provider-azure/csi-secrets-store-provider-azure --namespace kube-system --set secrets-store-csi-driver.enableSecretRotation=true --set secrets-store-csi-driver.rotationPollInterval=30s
```

4.6.2.2 Create Secret to Access Azure Key vault

```
kind: Secret
apiVersion: v1
metadata:
 name: <Secret Name>
 namespace: <WHERE FEP CLUSTER TO BE INSTALLED>
 labels:
  secrets-store.csi.k8s.io/used: 'true'
data:
 clientid: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXx
 clientsecret: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX=
type: Opaque
```

Clientid: clientid is SERVICE_PRINCIPAL_CLIENT_ID

Clientsecret: clientsecret is SERVICE_PRINCIPAL_CLIENT_SECRET

4.6.2.3 Store Secret in Azure Key Vault

```
az keyvault secret set --vault-name <Vault Name> --name <Secret Name> --value <Secret value>
```

4.6.2.4 Store Certificate in Azure Key Vault

Certificate should be in below format before uploading cert to Azure Key Vault i.e it should be one .pem file (key, crt and CA in one file)

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAx1rSsblocR8pROh5d2D3kuryTRRu6DA8axrSwrAaSDvdy1yU
KA7Q+Zg4IwaGwkt3cE2vK6oH4z3jwz+X0VjOxXo3hVh8tvfuXQ0uNpFEWCRRX1xt
3S8xc8OCzbnHRWQAKdxRGWhfmPSdWdlpPe7uNcVe865TVOWLMAjYzZbMOJnFHmK3
5EoxRkcLs3sGi74YhwDsGa1sNzBhZpdR+iIheEZKJUc65dl13jKx9oDhc1c8lcwR
ecrVgfRo6NfZ86bkR2ImL5xR0SWKnXP3KZqPOkL9DtCZK8iW2CgrfI8d2zcLbuUZ
UHEt4zzrwc9NV1yXe6nc8CrXbI6icwJYgVMZawIDAQABAoIBAF4kiN0/BpBtO8r7
0ejLVP7/jr9Rx/JEXTPjLeaczTyRcPNJW/nyzUMhXFlGCruUceoJ9ZA0Mpdgsb+R
t3s4aiUdyzxghjzNprYwtEM2pMTPGdJjzsomMD9P8+R9OBqP1/fswCu0e3i7A9fb
cPS7cajY9Tc0esvbvrhHZULpVLXhKl45SgDKgAWNaLJlm4u4gE56qpy+5kUKDzHg
yNOErpBSw2jlbtDE1Uta1hlR7BGWpK571UNvZ2AgLTbIgf1QFLq9IJdg9lll5pfm
DDn4AvcuFTHqJNj29DiMpsedvtPEnWceEkSScyzZnSvwJsADcdm2G8hyee0saQW+
/pVicfECgYEA7vADTIlWwOzcYH/CY+d0YAMaS0P08IPi5PXFj5FJ44q8BwZUDHGI
gUZylxJfipBvca2zYbrNSJ1ynF6mup30eeQDlVDSOdvcTg140CuSZuvl/mG+1sBK
G5QiXE15D6IJj3Ngu3wu+RFK3CCQuveERAaWD1kZizRlOFiacV7lJBkCgYEA1Zcz
1YNlLybKXJb0N3aFOhlz9RH1gNIx1PswJmDkM7qXlw5uxVpSPsvgngMsdAxMnSFQ
y5xxQY7fxUkv5ms6Po7c8BKyp2cLWRW2UH28ev8WT26yuml6OFXfv6XDhoF6CYeR
sGIlG9IUY2i4rkgajNYtyeE6r6O3LljoD7qNuiMCgYA55G94MOKTNhCjVPE9kYvx
426Qg/Op/tqPzTjD81jqx+eM8CyXIz8Gy5HiJrJ9eUd3TLXk3QT2Lifh2VEecD0W
93ciy4VUPYAgbUUzcwsy4r9EJly93bNXAUpeAOtvLTyRxEvQwWMEN/tiYIWQt34V
mV7scxMsVlKcF2O8SljMqQKBgBUgGV5a2p0pRwaVX55EuLSgY9mvZwrQv2EDXyXM
m4WKRQgJw2b9ofjYDWVThwgLV2CLNQSOep0zVmqa7IPrwx0A4FVWZBkule6/uQKJ
DSVVKY29syvA1vfPdovsB0S8daePoxdA/c6cnqueZfXG5+laHblD75wDo1CQNpOn
rfDlAoGBAJNI3q5XWGMciw8Rc00U2iWFSWWih9yHPpG3VGj2wUICDHd0oNvmYPik
PJMbemXI7fyUltthzx6TkY/8uvQpjNw1gLkKNSUQw/Fez8acA59jtvBnFy3ERDQD
+hsETWiHZ43QRo5fV0LjrUxurM9k/NTWzVBRov3yqc3XnVsgxujL
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIIFfjCCA2agAwIBAgIUCVqIwocAj7N/1NNCyLjporXLbE8wDQYJKoZIhvcNAQEL
BQAwVzEYMBYGA1UECgwPTXkgT3JnYW5pemF0aW9uMQswCQYDVQQLDAJDQTEuMCwG
A1UEAwwlTXkgT3JnYW5pemF0aW9uIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0y
MjEwMTMxMjE5MTBaFw0yMzEwMTMxMjE5MTBaMBMxETAPBgNVBAMMCHBvc3RncmVz
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAx1rSsblocR8pROh5d2D3
kuryTRRu6DA8axrSwrAaSDvdy1yUKA7Q+Zg4IwaGwkt3cE2vK6oH4z3jwz+X0VjO
xXo3hVh8tvfuXQ0uNpFEWCRRX1xt3S8xc8OCzbnHRWQAKdxRGWhfmPSdWdlpPe7u
NcVe865TVOWLMAjYzZbMOJnFHmK35EoxRkcLs3sGi74YhwDsGa1sNzBhZpdR+iIh
eEZKJUc65dl13jKx9oDhc1c8lcwRecrVgfRo6NfZ86bkR2ImL5xR0SWKnXP3KZqP
OkL9DtCZK8iW2CgrfI8d2zcLbuUZUHEt4zzrwc9NV1yXe6nc8CrXbI6icwJYgVMZ
awIDAQABo4IBhDCCAYAwggF8BgNVHREEggFzMIIBb4IKKi5ucy1hLnBvZIIYKi5u
cy1hLnBvZC5jbHVzdGVyLmxvY2FsghBuZjMzLXByaW1hcnktc3ZjghVuZjMzLXBy
aW1hcnktc3ZjLm5zLWGCGW5mMzMtcHJpbWFyeS1zdmMubnMtYS5zdmOCJ25mMzMt
cHJpbWFyeS1zdmMubnMtYS5zdmMuY2x1c3Rlci5sb2NhbIIQbmYzMy1yZXBsaWNh
LXN2Y4IVbmYzMy1yZXBsaWNhLXN2Yy5ucy1hghluZjMzLXJlcGxpY2Etc3ZjLm5z
LWEuc3ZjgiduZjMzLXJlcGxpY2Etc3ZjLm5zLWEuc3ZjLmNsdXN0ZXIubG9jYWyC
HG5mMzMtc3RzLTAubmYzMy1oZWFkbGVzcy1zdmOCPHB1Ymxpc2hlci1ob3N0LW5h
bWUubmFtZXNwYWNlLW9mLXB1Ymxpc2hlci5zdmMuY2x1c3Rlci5sb2NhbIIRbmYz
My1oZWFkbGVzcy1zdmMwDQYJKoZIhvcNAQELBQADggIBACBWl1DVvZj6kO5SSGpv
jXCCRu6jhWBaXH9jTH9Awg6DxXU6BzOATpCFMEcMP4Bv+1lG/2Gkz8p7PSfznsr9
LWK2ACuQ9FettgPZyQaHtV8e5AHCtCNK9WeSKoZ2XGIAKPJu3DZ7LZ0DP7lqinPC
T/cxY+4Qbtuga+gHoLKf0iATlM70sbRIpI5q4EosZtmp+dv8l1kHVZMLusDLhhV7
QYHhW1rJfpBEaUdrFaqUB+6Eo/MY3hbUzYMcGdae83KA1rW2/owL7E6pL8aJPhX9
igCT/XVwuIH3aaYkwDlOLZzU/ga8KOrs2cbEcHFB0tnNzs81hVebZmqV/GqmVTbD
ty8+IbU3miKa2/bDbmZBMWyvdVo52W1h62AZtGF93JvoaZVAAp53v3Gv6rs64lj2
7iP3CVLBs/OBFBG7y6q6/y0jlNEa4D9vOpPS3uBGSQDMpKG7mRIYksm0wULDBYI3
UjZpwVJjRuVY7N6ONgvZxOfC5HKb2Djb/u8RL8UrMmqqZlKNdh/060ZIZEX63esb
yHzQbiYSnop6LgpK5STtizJlaTpxkVcrJ2tzHuWp1PcCpShRTuKU+LFlOmOUMYk9
6Oi5h9GDTURDSO0O08RosiJd+locEBiKwZIA6dh98c+dd4eml9F+Pt3OlZA/wgcu
NwROKO5YLzFxBStiz2kiU0dZ
-----END CERTIFICATE-----
```

```
-----BEGIN CERTIFICATE-----
MIIFXzCCA0egAwIBAgIUR0l4D/Pjf9/VIxF+jYFV1MtKnpQwDQYJKoZIhvcNAQEL
BQAwVzEYMBYGA1UECgwPTXkgT3JnYW5pemF0aW9uMQswCQYDVQQLDAJDQTEuMCwG
A1UEAwwlTXkgT3JnYW5pemF0aW9uIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0y
MjEwMTMxMjE5MTBaFw0zMjEwMTAxMjE5MTBaMFcxGDAWBgNVBAoMD015IE9yZ2Fu
aXphdGlvbjELMAkGA1UECwwCQ0ExLjAsBgNVBAMMJU15IE9yZ2FuaXphdGlvbiBD
ZXJ0aWZpY2F0ZSBBdXRob3JpdHkwggIiMA0GCSqGSIb3DQEBAQUAA4ICDwAwggIK
AoICAQDOh1UQb6QLw/e1J2gar/eRf6W4PhkNpOKGmdS5Rm0J58sDEwb/BNaBRYzu
e05mLQ7R3YF3I83AZf19E0ss36tfi9puRaCr5toC/XaBqK1zLPSZmZVtlxadZSFG
9+3WB8IXrDuSQw1cZi9oos0Jeq962dPDqd56qicnEk7r8Vpd5ycYuadEclPDX7ne
zw6A6eHfIaAw9ETFOt1Ph88Yh3XhO+e937YOZOucpxJIXqxdGbK9yFgk4y4Pbjg7
yXWcFP1Cg2FKN/Odhr3k64WNDcqejpxbfJgxAtujg7lFjg/YuzbbMRjCzB1TZGPU
iM7TKPPw9PVoWKJ3siR5SoxJp5LgdkhvT83zx3zw87htjbcbnYPOy+F2PX88U5be
UpYzIcRjBPh59AYgfGJaBjTm5dy8ryWQ9diwAklxvnTwa7c443xG3IFHq5/Yt7ol
sbT1h5gp3hHfh/WvZxFagirX66Uz5TY2FDzWVsQHvoIGMHD8hcr7Reia8IPFnneW
zRE1lNPQNXhqgc0pflg/6u8FCMdEeR/QV1lsjavVEMXoJU0PEx+srhUg+4gVlzc1
7OPG/ThJ0dzXCeEEaI8Z6Yq5I3PJiEUvbWhEGOQ/S9pJeIlBwCsADGlVaAOXy+gy
5Hh8dTrWg+TwI8lpWQSWXJGIpY684/jLVFu16U5aawgacrmExwIDAQABoyMwITAP
BgNVHRMBAf8EBTADAQH/MA4GA1UdDwEB/wQEAwIBhjANBgkqhkiG9w0BAQsFAAOC
AgEAujV7RkipqwNopqW4kwbIbF4mn3JPBzbzKSjr8uraCFpk3ZTiRsiHm3D07/ox
N7KTqbk+DhSdbZ1NM+f1kZ7zDR6r4KGgBmKID51DOJ54jxNuCwRKndGUfePATuD0
yaLs0U1YAU02/S6cWKki1wEHv+t+p9zlJORd75M4GIKdQOyOtyEsiMPEbP3OqfJt
PJ7R+WBGvedt3TPEi3REubzUOMhgsDHuqeKKVBuRdh3zvcSI1q59DKYUir7wY60y
3fwJtEkrpyBD57Tp/Vsaf0Txv9KTtbyiCY0nwmiN3RqyFx4lIEipT1dhVc2oBUFq
YWvTkUPubFbG0aLxcbi5aySCOmjZHYZvUCNLSAekTL2wH649/RD8xSkQf+Qs2N6a
jJOE1nUrapYRrKlwFRXj+5aj+fhhoZ1uU43jPRakdwinEWmw7JPRk0gjRQwQE6a6
bhBvBfStOZKmuOULuoHrL75BCyQMK5JaOgljmcsAQMb0/ERpPxoNzkXAS825wOTx
E+lnRRuOKfmILIHMteOpn+ffozT2Djl3mFMJhbbbnYEL1NEYxwI2si2oL8GjE26i
A5ojkdJ06kmFgOp2boa49ja61lWVZToirWhbnR6G9AKHPy8aX0yH25xStxbdojjO
eTP+zKBUH3E15zT0YOnb7NnIplHNNhq1kwi/OCBXPs9FWow=
-----END CERTIFICATE-----
```

mycert.pem

```
az keyvault secret set --vault-name <Key Vault Name> --name <Secret Name> --file "mycert.pem"
```

Note

Only single key value for secret to be stored in key vault.

4.6.3 Installing and Configuring AWS Provider for Secret Store CSI Driver

4.6.3.1 Install AWS Provider drivers using helm chart

```
helm repo add aws-secrets-manager https://aws.github.io/secrets-store-csi-driver-provider-aws
```

```
helm install -n kube-system secrets-provider-aws aws-secrets-manager/secrets-store-csi-driver-
provider-aws -namespace kube-system
```

4.6.3.2 Setup EKS cluster along with service account with necessary IAM roles and permission to access Secret Manager

Follow below link to setup IAM roles and EKS for CSI.

https://github.com/aws/secrets-store-csi-driver-provider-aws

Create IAM role trust policy to access Secret Manager

```
Create IAM role trust policy to access Secret Manager
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Federated": "arn:aws:iam::123456789:oidc-provider/oidc.eks.ap-
southeast-3.amazonaws.com/id/ABCD1234567"
            },
            "Action": "sts:AssumeRoleWithWebIdentity",
            "Condition": {
                "StringEquals": {
                    " oidc.eks.ap-southeast-3.amazonaws.com /id/ ABCD1234567:sub":
"system:serviceaccount:myns:mysa",
                   " oidc.eks.ap-southeast-3.amazonaws.com /id/ ABCD1234567:aud": "sts.amazonaws.com"
                }
            }
        }
    ]
}
```

4.6.3.3 Store Secret in AWS Secrets Manager

```
aws secretsmanager create-secret --name <Secret Name> --secret-string <Secret Value>
```

4.6.3.4 Store Cert in AWS Secrets Manager

Certificate should be in below format before uploading cert to AWS Secrets Manager i.e it should be one .pem file (key, crt and CA in one file)

(Refer "mycert.pem" for sample certificate format)

```
aws secretsmanager create-secret --name <Secret Name> --secret-binary fileb://<File Name>
```

Note

Only single key value for secret to be stored in Secret Manager.

4.6.4 Installing GCP Provider for Secret Store CSI Driver

4.6.4.1 Install GCP Provider drivers using Kubernetes

```
wget https://raw.githubusercontent.com/GoogleCloudPlatform/secrets-store-csi-driver-provider-gcp/
main/deploy/provider-gcp-plugin.yaml
```

```
kubectl apply -f provider-gcp-plugin.yaml -namespace kube-system
```

4.6.4.2 Configure GCP secret manager and IAM

Create Service Account:

```
gcloud iam service-accounts create my-secret-acc;
```

Attach SecretManagerAdmin policy to the new service account

```
gcloud projects add-iam-policy-binding $PROJECT_ID \
--member="serviceAccount: my-secret-acc @$PROJECT_ID.iam.gserviceaccount.com" \
--role="roles/secretmanager.admin" \
--condition="None";
```

Generate a key for your new service account

```
gcloud iam service-accounts keys create iam-key.json \
--iam-account=" my-secret-acc @$PROJECT_ID.iam.gserviceaccount.com";
```

4.6.4.3 Create Secret to access GCP Secret manager

Use keys generated from "4.6.4.2 Configure GCP secret manager and IAM" (iam-key.json file)

```
kubectl create secret generic <secret-name> --from-file=<iam-key.json>
```

4.6.4.4 Store secret in GCP Secret manager

```
gcloud secrets create <secret name> --data-file="/path/to/file"
```

4.6.4.5 Store Cert in GCP Secret manager

Certificate should be in below format before uploading cert to GCP Secret Manager i.e it should be one .pem file (key, crt and CA in one file)

(Refer "mycert.pem" for sample certificate format)

```
gcloud secrets create <secret name> --data-file="/path/to/file"
```

Note

Only single key value for secret to be stored in Secret Manager.

4.6.5 Installing HashiCorp Vault Provider for Secret Store CSI Driver

4.6.5.1 Install HashiCorp Provider drivers using helm chart

```
helm repo add hashicorp https://helm.releases.hashicorp.com
```

```
helm install vault hashicorp/vault --set "server.enabled=false"  --set "injector.enabled=false" --set
"csi.enabled=true"
```

4.6.5.2 Configure Kubernetes Authentication for HashiCorp Vault

```
vault auth enable kubernetes
```

```
vault write auth/kubernetes/config \
token_reviewer_jwt="$(cat /var/run/secrets/kubernetes.io/serviceaccount/token)" \
kubernetes_host="https://$KUBERNETES_PORT_443_TCP_ADDR:443" \
kubernetes_ca_cert=@/var/run/secrets/kubernetes.io/serviceaccount/ca.crt
```

4.6.5.3 Store Secret in HashiCorp Vault

```
vault enable secret
```

```
vault kv put secret/<path> <secret name>=<secret value>
```

4.6.5.4 Store Cert in HashiCorp Vault

Certificate should be in below format before uploading cert to HashiCorp Vault i.e it should be one .pem file (key, crt and CA in one file)

(Refer "mycert.pem" for sample certificate format)

```
Vault kv put secret/<path> <secret name>=@<path to cert.pem>
```

4.6.5.5 Create policy and role to access the secrets from HashiCorp Vault

Policy:

```
vault policy write <policy name> - <<EOF
path "secret/database/credentials" {
capabilities = ["read", "write", "update","delete"]
}
EOF
```

Role:

```
vault write auth/kubernetes/role/<role name> \
bound_service_account_names=* \
bound_service_account_namespaces=*\
policies=<policy name> \
ttl=24h
```

Note: access can be restricted by assigning <fep-cluster>-sa service account to bound_service_account_names and also can be namespace restircted by assigning value to bound_service_account_namespaces

Only single key value for secret to be stored in HashiCorp vault.

4.6.6 Configuring FEPCluster to use Provider for Secret Store Driver

To enable use of Secret Store CSI driver, a new parameter "secretStore" under spec.fepChildCrVal section in the FEPClusterCR. Under secretStore.csi user should define the details to connect to external Seret store(Azure,AWS,GCP and HashiCorp Vault) and the list of secrets in that secret store. The definition of spec.fepChildCrVal.secretStore parameter will differ depending on the type of provider that is used.

4.6.6.1 Azure Provider for Secret Store CSI Driver

```
spec:
  ….
  fepChildCrVal:
    secretStore:
      method: csi
      csi:
        providerName: azure
        azureProvider:
          keyvaultname:
          tenantid:
          credentials:
          fepSecrets:
          - pgadminpassword: pgadminpassword
          - tdepassphrase: passphrase
          - systemCertificates: systemCerts
          - pguser: pgusername
          - pgpassword: pgpwd
```

```
          - pgdb: pgdbsecret
          - pgrepluser: pgrepluser
          - pgreplpassword: pgreplpassword
          - pgRewinduser: pgRewinduser
          - pgRewindpassword: pgRewindpassword
          - pgMetricsUser: metricsuser
          - pgMetricsPassword: metricspwd
          - patronitls: patronicrt
          - patronitlscacrt: patronica
          - postgrestls: postgrescrt
          - postgrestlscacrt: postgresca
          - pgAdminTls: admincrt
          - pgAdminTlscacrt: adminca
          - pgAdminTls_privateKeyPassword: adminpvtkey
          - pgRewindUserTls: rewindcrt
          - pgRewindUserTlscacrt: rewindca
          - pgRewindUserTls_privateKeyPassword: rwndpvtkey
          - pgrepluserTls: replcrt
          - pgrepluserTlscacrt: replca
          - pgrepluserTls_privateKeyPassword: replpvtkey
          - pgMetricsUserTls: metricscrt
          - pgMetricsUserTlscacrt: metricsca
          - pgMetricsUserTls_privateKeyPassword: adminpvtkey
          fepCustomCerts:
            - userName:user1
              userCrt: user1crt
              userCa: user1ca
            - userName: mydbuser
              userCrt: mydbusercrt
              userCa: mydbuserca
```

Note: The parameters which are in black in fepSecrets are mandatory.

4.6.6.2 AWS Provider for Secret Store CSI Driver

```
spec:
  ….
  fepChildCrVal:
    secretStore:
      method: csi
      csi:
        providerName: aws
        awsProvider:
          region:
          roleName:
          fepSecrets:
          - pgadminpassword: pgadminpassword
          - tdepassphrase: passphrase
          - systemCertificates: systemCerts
          - pguser: pgusername
          - pgpassword: pgpwd
          - pgdb: pgdbsecret
          - pgrepluser: pgrepluser
          - pgreplpassword: pgreplpassword
          - pgRewinduser: pgRewinduser
          - pgRewindpassword: pgRewindpassword
          - pgMetricsUser: metricsuser
          - pgMetricsPassword: metricspwd
          - patronitls: patronicrt
          - patronitlscacrt: patronica
          - postgrestls: postgrescrt
          - postgrestlscacrt: postgresca
          - pgAdminTls: admincrt
```

```
          - pgAdminTlscacrt: adminca
          - pgAdminTls_privateKeyPassword: adminpvtkey
          - pgRewindUserTls: rewindcrt
          - pgRewindUserTlscacrt: rewindca
          - pgRewindUserTls_privateKeyPassword: rwndpvtkey
          - pgrepluserTls: replcrt
          - pgrepluserTlscacrt: replca
          - pgrepluserTls_privateKeyPassword: replpvtkey
          - pgMetricsUserTls: metricscrt
          - pgMetricsUserTlscacrt: metricsca
          - pgMetricsUserTls_privateKeyPassword: adminpvtkey
          fepCustomCerts:
            - userName:user1
              userCrt: user1crt
              userCa: user1ca
            - userName: mydbuser
              userCrt: mydbusercrt
              userCa: mydbuserca
```

Note: The parameters which are in black in fepSecrets are mandatory.

4.6.6.3 GCP Provider for Secret Store CSI Driver

```
spec:
  ….
  fepChildCrVal:
    secretStore:
      method: csi
      csi:
        providerName: gcp
        gcpProvider:
          credentials:
          fepSecrets:
          - pgadminpassword: pgadminpassword
          - tdepassphrase: passphrase
          - systemCertificates: systemCerts
          - pguser: pgusername
          - pgpassword: pgpwd
          - pgdb: pgdbsecret
          - pgrepluser: pgrepluser
          - pgreplpassword: pgreplpassword
          - pgRewinduser: pgRewinduser
          - pgRewindpassword: pgRewindpassword
          - pgMetricsUser: metricsuser
          - pgMetricsPassword: metricspwd
          - patronitls: patronicrt
          - patronitlscacrt: patronica
          - postgrestls: postgrescrt
          - postgrestlscacrt: postgresca
          - pgAdminTls: admincrt
          - pgAdminTlscacrt: adminca
          - pgAdminTls_privateKeyPassword: adminpvtkey
          - pgRewindUserTls: rewindcrt
          - pgRewindUserTlscacrt: rewindca
          - pgRewindUserTls_privateKeyPassword: rwndpvtkey
          - pgrepluserTls: replcrt
          - pgrepluserTlscacrt: replca
          - pgrepluserTls_privateKeyPassword: replpvtkey
          - pgMetricsUserTls: metricscrt
          - pgMetricsUserTlscacrt: metricsca
          - pgMetricsUserTls_privateKeyPassword: adminpvtkey
          fepCustomCerts:
            - userName:user1
```

```
            userCrt: user1crt
            userCa: user1ca
          - userName: mydbuser
            userCrt: mydbusercrt
            userCa: mydbuserca
```

Note: The parameters which are in black in fepSecrets are mandatory.

4.6.6.4 HashiCorp Vault Provider for Secret Store CSI Driver

```
spec:
  ….
  fepChildCrVal:
    secretStore:
      method: csi
      csi:
        providerName: vault
        vaultProvider:
          roleName: "database"
          vaultAddress: "http://vault-url-addr:8765"
          fepSecrets:
          - pgadminpassword: pgadminpassword
          - tdepassphrase: passphrase
          - systemCertificates: systemCerts
          - pguser: pgusername
          - pgpassword: pgpwd
          - pgdb: pgdbsecret
          - pgrepluser: pgrepluser
          - pgreplpassword: pgreplpassword
          - pgRewinduser: pgRewinduser
          - pgRewindpassword: pgRewindpassword
          - pgMetricsUser: metricsuser
          - pgMetricsPassword: metricspwd
          - patronitls: patronicrt
          - patronitlscacrt: patronica
          - postgrestls: postgrescrt
          - postgrestlscacrt: postgresca
          - pgAdminTls: admincrt
          - pgAdminTlscacrt: adminca
          - pgAdminTls_privateKeyPassword: adminpvtkey
          - pgRewindUserTls: rewindcrt
          - pgRewindUserTlscacrt: rewindca
          - pgRewindUserTls_privateKeyPassword: rwndpvtkey
          - pgrepluserTls: replcrt
          - pgrepluserTlscacrt: replca
          - pgrepluserTls_privateKeyPassword: replpvtkey
          - pgMetricsUserTls: metricscrt
          - pgMetricsUserTlscacrt: metricsca
          - pgMetricsUserTls_privateKeyPassword: adminpvtkey
          fepCustomCerts:
            - userName:user1
              userCrt: user1crt
              userCa: user1ca
            - userName: mydbuser
              userCrt: mydbusercrt
              userCa: mydbuserca
```

Note: The parameters which are in black in fepSecrets are mandatory.

4.7 Deploying a customized FEP server container image

4.7.1 Requirements

The procedures documented below assume the use of docker command to build container image. Building container images using alternative tools such as podman is beyond the scope of this document.

4.7.2 Build custom FEP image with extension

Before building a new custom FEP Server container image, it is important to understand several build instructions specific to that image.

- FEP server container image is built on top of UBI8/UBI9 minimal image, ubi-minimal
- USER is default to 26

UBI8/UBI9 minimal image uses microdnf as package manager. Microdnf does not support installing RPM packages from remote URL or local file, only from a YUM repository. If you want to install RPM package that is not in YUM repository, first download the package and install it with rpm. However, rpm has the drawback that it does not resolve dependencies. The only way to resolve this problem is to install dnf first and use dnf to install packages from remote URL or local file.

As USER is default to 26, it does not have the permission to install RPM packages or write files to system directory such as /usr/bin, /usr/local/bin, etc. To workaround this issue, first set USER to root to continue the customization and set it back to 26.

```
FROM: quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0

USER root

RUN …(customization)

USER 26
```

4.7.3 Adding SQLite Foreign Data Wrapper to FEP Server Container

We will demonstrate adding the SQLite Foreign Data Wrapper module to FEP Server container.

1. Create Dockerfile

```
#use FEP 17 image as a base to compile sqlite_fdw
FROM quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0 as compile-sqlite_fdw

#change the user with root privilege
USER root

# install build tools
RUN microdnf -y install cmake gcc-c++ libtool clang which openssl-devel git llvm gettext

# Install sqlite_fdw build require
RUN microdnf install -y sqlite-devel

# Download sqlite_fdw source
RUN curl -sSL https://github.com/pgspider/sqlite_fdw/archive/refs/tags/v2.3.0.tar.gz | tar -zxf -

# Compile sqlite_fdw
RUN cd /sqlite_fdw-2.3.0 && \
    make install USE_PGXS=1

#Use base image is from FEPContainer to build the custom image
FROM quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0

#change the user with root privilege
USER root
```

```
#copy the prepared OSS extension binaries to FEP server lib folder
COPY --from=compile-sqlite_fdw /opt/fsepv17server64/lib/sqlite_fdw.so /opt/fsepv17server64/lib/
COPY --from=compile-sqlite_fdw /opt/fsepv17server64/lib/bitcode/sqlite_fdw /opt/fsepv17server64/
lib/bitcode/
COPY --from=compile-sqlite_fdw /opt/fsepv17server64/share/extension/sqlite_fdw* /opt/
fsepv17server64/share/extension/

# Install sqlite_fdw run time dependencies
RUN microdnf install -y sqlite-libs

#change the user to postgresql
USER 26
```

2. Build custom image

```
docker build -f Dockerfile -t my.registry/my-repo/fep-17-server-sqlite_fdw:ubi9-17-1.0
```

3. Push image to custom container registry

```
docker push my.registry/my-repo/fep-17-server-sqlite_fdw:ubi9-17-1.0
```

4.7.4 Create FEP Cluster with custom image

If the custom container registry requires authentication, create a pull secret with the name quay-pull-secret. FEP Operator will this this pull secret to download container image.

```
kind:Secret
apiVersion:v1
metadata:
  name:quay-pull-secret
  namespace:fep-container-ct
data:
  .dockerconfigjson:~>-
    xxxxxxbCI6ICIiCiAgICB9CiAgfQp9
type:kubernetes.io/dockerconfigjson
```

Create FEP Cluster CR

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: sqlite-fdw
  namespace: fep-container-ct
spec:
  fep:
    forceSsl: true
... ...
   image:
      image: 'my.registry/my-repo/fep-17-server-sqlite_fdw:ubi9-17-1.0'
... ...
   instances: 1
```

Deploy FEPCluster

```
oc apply -f sqlite_fdw.yaml
```

Create extension

```
postgres# CREATE EXTENSION sqlite_fdw;
CREATE EXTENSION
postgres=#
```

4.8 Configuration FEP to Perform MTLS

All three traffic can be secured by using TLS connection protected by certificates:

- Postgres traffic from Client Application to FEPCluster
- Patroni RESTAPI within FEPCluster
- Postgres traffic within FEPCluster (e.g. replication, rewind)

4.8.1 When Using an Automatically Generated Certificate

4.8.1.1 How to Create a Certificate

Operators can automatically generate certificates and use MTLS for communication between containers to improve safety.

If cert-manager is installed in the Kubernetes environment, use cert-manager; if cert-manager is not installed, use the openssl command in the container to generate the certificate.

If the certificate is automatically generated, the default connection method to PostgreSQL is only mutual authentication TLS communication.

If you want to change the authentication method for each user, define the connection method for each user in spec.fepChildCrVal.customPgHba.

To change the automatic certificate generation feature, create the ConfigMap "fepopr-cert-methood" below before installing the operator. cert-manager, openssl, and disable can be specified for data.fepopr-cert-method. To disable the automatic certificate generation feature, specify disable.

```
$ cat > /tmp/cert-method.yaml <<EOF
kind: ConfigMap
apiVersion: v1
metadata:
  name: fepopr-cert-method
immutable: true
data:
  fepopr-cert-method: disable
EOF

$ kubectl -n <namespace> apply -f /tmp/cert-method.yaml
```

If you want to use a manually created certificate, refer to "4.8.2 When Using Your Own Certificate". If you want to use your own certificate, disable the automatic certificate generation feature.

4.8.1.2 How to Create a Client Certificate

The automatic certificate generation feature automatically generates system user certificates for database management. If you want to create a certificate for a general user who connects to the database, refer to the following steps.

This section explains how to create a Secret resource containing a certificate to be mounted in the client container. The created Secret resource can mounte on the client's Pod and used to connect to FEPCluster using certificate authentication.

When using cert-manager to generate certificates

1. Verify that the Issuer resource "<namespace>-ca-issuer" exists.

2. Create a template for the Certificate resource.

```
$ cat mydbuser-cert.yaml
apiVersion:cert-manager.io/v1
kind:Certificate
metadata:
name: "<fepclustername>-<username>-cert"
namespace: "<namespace>"
spec:
commonName: "< username >"
issuerRef:
name: <namespace>-ca-issuer
secretName: "<fepclustername>-<username>-cert"
```

3. Apply the Certificate resource.

```
$ kubectl apply -f mydbuser-cert.yaml
```

When using openssl to generate certificates

1. Get the CA certificate stored in the Secret "fepopr-root-secret".

```
$ kubectl -n <namespace> get secret fepopr-root-secret-o jsonpath='{.data.tls\.key}' | base64
-d > <namespace>-ca.key

$ kubectl -n <namespace> get secret fepopr-root-secret-o jsonpath=’ {.data.tls\.crt}’  |
base64 -d > <namespace>-ca.crt
```

2. Create the certificate using the following command.

```
$ serial=$(openssl rand -hex 8)
$ openssl genrsa -out <fepclustername>-<username>.key 2048
$ openssl req -new -key <fepclustername>-<username>.key -out <fepclustername>-<username>.csr -
subj "/CN=<username>"
$ openssl x509 -req -in <fepclustername>-<username>.csr -CA <namespace>-ca.crt -CAkey
<namespace>-ca.key -set_serial "0x${serial}" -out <fepclustername>-<username>.crt -days 365
```

3. Create a secret to store the certificate.

```
$ kubectl -n <namespace> create secret tls <fepclustername>-<username>-cert --cert
<fepclustername>-<username>.crt -key=<fepclustername>-<username>.key
```

4.8.2 When Using Your Own Certificate

Here, we provide two methods to create certificates for securing the TLS connection and provide mutual authentication. The first method is to create and renew certificate manually. The second method is to use CertManager to create an automatically renew certificate.

Note

The following considerations apply to client connections to a database cluster in an MTLS configuration:.

- Distribute the Root certificate for server (validation) that you specified when you created the MTLS database cluster to the client machines.
- Create and use a new client certificate.
- If the server root certificate and the client root certificate are different, a server-side configuration update is required.

4.8.2.1 Manual Certificate Management

Overview of Procedures

The procedures to enable MTLS communication are listed below:

1. Create a self signed certificate as CA
2. Create Configmap to store CA certificate
3. Create a password for protecting FEP Server private key (optional)
4. Create FEP Server private key
5. Create FEP Server certificate signing request
6. Create FEP Server certificate signed by CA
7. Create TLS Secret to store FEP Server certificate and key
8. Create private key for Patroni
9. Create certificate signing request for Patroni
10. Create certificate signed by CA for Patroni
11. Create TLS secret to store Patroni certificate and key
12. Create private key for "postgres" user client certificate
13. Create certificate signing request for "postgres" user client certificate
14. Create client certificate for "postgres" user
15. Create TLS secret to store "postgres" certificate and key
16. Repeat step 12-15 for "repluser" and "rewinduser"

- The information in the manual is only an example, and in operation, use a certificate signed by a certificate authority (CA) that the user can trust.
- When working on a Kubernetes cluster, replace the oc command with the kubectl command.

Creating a CA Certificate

1. Create a self signed certificate as CA

```
openssl genrsa -aes256 -out myca.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
.......................++++
.....................................................++++
e is 65537 (0x010001)
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
Verifying - Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv

cat << EOF > ca.cnf
[req]
distinguished_name=req_distinguished_name
x509_extensions=v3_ca
[v3_ca]
basicConstraints = critical, CA:true
keyUsage=critical,keyCertSign,digitalSignature,cRLSign
[req_distinguished_name]
commonName=Common Name
EOF
```

```
openssl req -x509 -new -nodes -key myca.key -days 3650 -out myca.pem -subj "/O=My Organization/
OU=CA /CN=My Organization Certificate Authority" -config ca.cnf
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
```

2. Create Configmap to store CA certificate

```
oc create configmap cacert --from-file=ca.crt=myca.pem -n my-namespace
```

3. Create a password for protecting FEP Server private key (optional)

```
oc create secret generic mydb-fep-private-key-password --from-literal=keypassword=abcdefghijk -n
my-namespace
```

Creating a Server Certificate

4. Create FEP Server private key

```
openssl genrsa -aes256 -out fep.key 2048
Generating RSA private key, 2048 bit long modulus
.............................................+++
.......+++
e is 65537 (0x10001)
Enter pass phrase for fep.key: abcdefghijk
Verifying - Enter pass phrase for fep.key: abcdefghijk
```

5. Create FEP Server certificate signing request

```
cat << EOF > san.cnf
[SAN]
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.my-namespace.pod
DNS.2 =  *.my-namespace.pod.cluster.local
DNS.3 = mydb-primary-svc
DNS.4 = mydb-primary-svc.my-namespace
DNS.5 = mydb-primary-svc.my-namespace.svc
DNS.6 = mydb-primary-svc.my-namespace.svc.cluster.local
DNS.7 = mydb-replica-svc
DNS.8 = mydb-replica-svc.my-namespace
DNS.9 = mydb-replica-svc.my-namespace.svc
DNS.10 = mydb-replica-svc.my-namespace.svc.cluster.local
EOF

openssl req -new -key fep.key -out fep.csr -subj "/CN=mydb-headless-svc" -reqexts SAN -config
<(cat /etc/pki/tls/openssl.cnf <(cat san.cnf))
Enter pass phrase for fep.key: abcdefghijk
```

Note

The cluster name and namespace must be changed appropriately.

If you are connecting from outside the OCP cluster, you must also include the host name used for that connection.

6. Create FEP Server certificate signed by CA

```
openssl x509 -req -in fep.csr -CA myca.pem -CAkey myca.key -out fep.pem -days 365 -extfile
<(cat /etc/pki/tls/openssl.cnf <(cat san.cnf)) -extensions SAN -CAcreateserial  # all in one line
Signature ok
subject=/CN=mydb-headless-svc
Getting CA Private Key
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
```

7. Create TLS Secret to store FEP Server certificate and key

```
oc create secret generic mydb-fep-cert --from-file=tls.crt=fep.pem --from-file=tls.key=fep.key -n
my-namespace
```

8. Create private key for Patroni

At the moment, FEP container does not support password protected private key for Patroni.

```
openssl genrsa -out patroni.key 2048
Generating RSA private key, 2048 bit long modulus
.............................................+++
.......+++
e is 65537 (0x10001)
```

9. Create certificate signing request for Patroni

```
cat << EOF > san.cnf
[SAN]
subjectAltName = @alt_names
[alt_names]
DNS.1 = *.my-namespace.pod
DNS.2 = *.my-namespace.pod.cluster.local
DNS.3 = mydb-primary-svc
DNS.4 = mydb-primary-svc.my-namespace
DNS.5 = mydb-replica-svc
DNS.6 = mydb-replica-svc.my-namespace
DNS.7 = mydb-headless-svc
DNS.8 = mydb-headless-svc.my-namespace
EOF

openssl req -new -key patroni.key -out patroni.csr -subj "/CN=mydb-headless-svc" -reqexts SAN -
config <(cat /etc/pki/tls/openssl.cnf <(cat san.cnf))  # all in one line
```

Note

The cluster name and namespace must be changed appropriately.

If you are connecting from outside the OCP cluster, you must also include the host name used for that connection.

10. Create certificate signed by CA for Patroni

```
openssl x509 -req -in patroni.csr -CA myca.pem -CAkey myca.key -out patroni.pem -days 365 -extfile
<(cat /etc/pki/tls/openssl.cnf <(cat san.cnf)) -extensions SAN -CAcreateserial  # all in one line
Signature ok
subject=/CN=mydb-headless-svc
```

```
Getting CA Private Key
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
```

11. Create TLS secret to store Patroni certificate and key

```
oc create secret tls mydb-patroni-cert --cert=patroni.pem --key=patroni.key -n my-namespace
```

Creating a User Certificate

12. Create private key for "postgres" user client certificate

    At the moment, SQL client inside FEP server container does not support password protected certificate.

```
openssl genrsa -out postgres.key 2048
Generating RSA private key, 2048 bit long modulus
.............................................+++
.......+++
e is 65537 (0x10001)
```

13. Create certificate signing request for "postgres" user client certificate

```
openssl req -new -key postgres.key -out postgres.csr -subj "/CN=postgres"
```

14. Create client certificate for "postgres" user

```
openssl x509 -req -in postgres.csr -CA myca.pem -CAkey myca.key -out postgres.pem -days 365
Signature ok
subject=CN = postgres
Getting CA Private Key
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
```

15. Create TLS secret to store "postgres" certificate and key

```
oc create secret tls mydb-postgres-cert --cert=postgres.pem --key=postgres.key -n my-namespace
```

16. Repeat step 12-15 for "repluser" and "rewinduser"

4.8.2.2 Automatic Certificate Management

There are many Certificate Management tools available in the public. In this example, we will use cert-manager for the purpose.

- Note that certificates created in this example are not password protected.
- When working on a Kubernetes cluster, replace the oc command with the kubectl command.

Install cert-manager

```
oc create namespace cert-manager

oc apply -f https://github.com/jetstack/cert-manager/releases/download/v1.3.0/cert-manager.yaml
```

Create a Self Signed Issuer (This can be namespace specific or cluster wise)

This example creates an Issuer, that can create self signed certificate, in namespace my-namespace.

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: selfsigned-issuer
  namespace: my-namespace
spec:
  selfSigned: {}
EOF
```

Create a Self Signed CA certificate using selfsigned-issuer

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: cacert
  namespace: my-namespace
spec:
  subject:
    organizations:
      - My Organization
    organizationalUnits:
      - CA
  commonName: "My Organization Certificate Authority"
  duration: 87600h
  isCA: true
  secretName: cacert
  issuerRef:
    name: selfsigned-issuer
EOF
```

The above command will create a self signed Root certificate and private key stored in the Kubernetes secret "cacert" in namespace my-namespace.

Create a CA Issuer with above certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: ca-issuer
  namespace: my-namespace
spec:
  ca:
    secretName: cacert
EOF
```

Create FEP Server certificate using above CA Issuer

Assuming FEPCluster name is mydb in namespace my-namespace.

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: mydb-fep-cert
```

```
  namespace: my-namespace
spec:
  subject:
  commonName: "mydb-headless-svc"
  dnsNames:
  - "*.my-namespace.pod"
  - "*.my-namespace.pod.cluster.local"
  - "mydb-primary-svc"
  - "mydb-primary-svc.my-namespace"
  - "mydb-primary-svc.my-namespace.svc"
  - "mydb-primary-svc.my-namespace.svc.cluster.local"
  - "mydb-replica-svc"
  - "mydb-replica-svc.my-namespace"
  - "mydb-replica-svc.my-namespace.svc"
  - "mydb-replica-svc.my-namespace.svc.cluster.local"
  duration: 8760h
  usages:
  - server auth
  secretName: mydb-fep-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create Patroni certificate using above CA Issuer

Assuming FEPCluster name is mydb in namespace my-namespace.

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: mydb-patroni-cert
  namespace: my-namespace
spec:
  subject:
  commonName: "mydb-headless-svc"
  dnsNames:
  - "*.my-namespace.pod"
  - "*.my-namespace.pod.cluster.local"
  - "*.mydb-primary-svc"
  - "*.mydb-primary-svc.my-namespace"
  - "*.mydb-replica-svc "
  - "*.mydb-replica-svc.my-namespace"
  duration: 8760h
  usages:
  - server auth
  secretName: mydb-patroni-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create postgres user client certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: mydb-postgres-cert
  namespace: my-namespace
spec:
  subject:
```

```
  commonName: "postgres"
  duration: 8760h
  usages:
  - client auth
  secretName: mydb-postgres-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create repluser user client certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: mydb-repluser-cert
  namespace: my-namespace
spec:
  subject:
  commonName: "repluser"
  duration: 8760h
  usages:
  - client auth
  secretName: mydb-repluser-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create FEPLogging(Fluentd) server certificate using above CA Issuer

Assuming FEPLogging name is **nfl** in namespace **feplogging-dev.**

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: fluentd-cert
  namespace: feplogging-dev
spec:
  subject:
  commonName: "nfl-fluentd-headless-service"
  dnsNames:
  - 'nfl-fluentd-headless-service'
  - 'nfl-fluentd-headless-service.feplogging-dev'
  - 'nfl-fluentd-headless-service.feplogging-dev.svc'
  - 'nfl-fluentd-headless-service.feplogging-dev.svc.cluster.local'
  duration: 8760h
  usages:
  - server auth
  secretName: fluentd-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create FEPLogging client(prometheus) certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
```

```
  name: prometheus-cert
  namespace: feplogging-dev
spec:
  subject:
  commonName: "prometheus"
  duration: 8760h
  usages:
  - client auth
  secretName: prometheus-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create FEPLogging client(fluentbit) certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: fluentbit-cert
  namespace: feplogging-dev
spec:
  subject:
  commonName: "fluentbit"
  duration: 8760h
  usages:
  - client auth
  secretName: fluentbit-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create FEPExporter certificate using above CA Issuer

Assuming FEP Exporter name is **exp1** in namespace **my-namespace.**

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: fepexporter-cert
  namespace: my-namespace
spec:
  subject:
  commonName: "exp1-service"
  dnsNames:
  - 'exp1-service'
  - 'exp1-service.fepexporter-dev'
  - 'exp1-service.fepexporter-dev.svc'
  - 'exp1-service.fepexporter-dev.svc.cluster.local'
  duration: 8760h
  usages:
  - server auth
  secretName: fepexporter-cert
  issuerRef:
    name: ca-issuer
EOF
```

Create FEPExporter user client(prometheus) certificate

```
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: prometheus-cert
  namespace: my-namespace
spec:
  subject:
  commonName: "prometheus"
  duration: 8760h
  usages:
  - client auth
  secretName: prometheus-cert
  issuerRef:
    name: ca-issuer
EOF
```

4.8.2.3 Deploy FEPCluster with MTLS support

Deploy FEPCluster with manual certificate management

Use the following yaml as an example to deploy a FEPCluster with Manual Certificate Management. MTLS related parameters are highlighted in Red.

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: mydb
  namespace: my-namespace
spec:
  fep:
    usePodName: true
    patroni:
      tls:
        certificateName: mydb-patroni-cert
        caName: cacert
    postgres:
      tls:
        certificateName: mydb-fep-cert
        caName: cacert
        privateKeyPassword: mydb-fep-private-key-password
    forceSsl: true
    podAntiAffinity: false
    mcSpec:
      limits:
        cpu: 500m
        memory: 700Mi
      requests:
        cpu: 200m
        memory: 512Mi
    customAnnotations:
      allDeployments: {}
    servicePort: 27500
    image:
      image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0'
      pullPolicy: IfNotPresent
    sysExtraLogging: false
    podDisruptionBudget: false
    instances: 3
    syncMode: 'on'
  fepChildCrVal:
```

```
  customPgAudit: |
    # define pg audit custom params here to override defaults.
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'
    [output]
    logger = 'auditlog'
    log_directory = '/database/log/audit'
    [rule]
  customPgHba: |
    # define pg_hba custom rules here to be merged with default rules.
    # TYPE     DATABASE        USER        ADDRESS        METHOD
    hostssl    all             all         0.0.0.0/0      cert
    hostssl    replication     all         0.0.0.0/0      cert
  customPgParams: >+
    # define custom postgresql.conf parameters below to override defaults.
    # Current values are as per default FEP deployment
    shared_preload_libraries='pgx_datamasking,pgaudit,pg_prewarm'
    session_preload_libraries='pg_prewarm'
    max_prepared_transactions = 100
    max_worker_processes = 30
    max_connections = 100
    work_mem = 1MB
    maintenance_work_mem = 12MB
    shared_buffers = 128MB
    effective_cache_size = 384MB
    checkpoint_completion_target = 0.8

    # tcp parameters
    tcp_keepalives_idle = 30
    tcp_keepalives_interval = 10
    tcp_keepalives_count = 3

    # logging parameters in default fep installation
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'
    log_directory = '/database/log'
    log_filename = 'logfile-%a.log'
    log_file_mode = 0600
    log_truncate_on_rotation = on
    log_rotation_age = 1d
    log_rotation_size = 0
    log_checkpoints = on
    log_line_prefix = '%e %t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h'
    log_lock_waits = on
    log_autovacuum_min_duration = 60s
    logging_collector = on
    pgaudit.config_file='/opt/app-root/src/pgaudit-cfg/pgaudit.conf'
    log_replication_commands = on
    log_min_messages = WARNING
    log_destination = stderr

    # wal_archive parameters in default fep installation
    archive_mode = on
    archive_command = '/bin/true'
    wal_level = replica
    max_wal_senders = 12
    wal_keep_segments = 64

  storage:
    dataVol:
      size: 2Gi
      storageClass: nfs-client
    walVol:
```

```
      size: 1200Mi
      storageClass: nfs-client
    logVol:
      size: 1Gi
      storageClass: nfs-client
  sysUsers:
    pgAdminPassword: admin-password
    pgdb: mydb
    pgpassword: mydbpassword
    pguser: mydbuser
    pgrepluser: repluser
    pgreplpassword: repluserpwd
    pgRewindUser: rewinduser
    pgRewindPassword: rewinduserpwd
    pgAdminTls:
      certificateName: mydb-postgres-cert
      caName: cacert
      sslMode: prefer

    pgrepluserTls:
      certificateName: mydb-repluser-cert
      caName: cacert
      sslMode: prefer

    pgRewindUserTls:
      certificateName: mydb-rewinduser-cert
      caName: cacert
      sslMode: prefer

    tdepassphrase: tde-passphrase
  systemCertificates:
    key: |-
      -----BEGIN RSA PRIVATE KEY-----
      MIIEowIBAAKCAQEA0DFkImha8CIJiVcwXbBPlL+/DmS9/ipRhQQHxfO5x7jSOnse
      IHdFd6+Qx2GX8KAiAhVykf6kfacwBYTATU1xDgwWTm82KVRPh+kZDIj2wPcJr14m
      mTP6I6a2mavUgDhezHc9F8/dchYj3cw81X0kU6xamqrKQYlxQH48NkI0qcwhO6sK
      AHF4eWfCr8Ot44xADIA1JcU2CS1RKSZEtURZ+30Py+j907Enjp1YR33ZKUHw30pU
      9dpIneyfXBN/pT6cX3MetYwtgmpV/pHqY8pbxqGfoyRhgQDsSRCl4dtlecaZeZ4j
      uTOotcPkZELHP6eu8gaLtycG9lpbAMQl5w0r8QIDAQABAoIBACq213qPuoimExrQ
      fqXaNJmqNYK4fJqXCB6oUwf0F1u4ubkx5V532hLSPHwLs+a0lAWlbNozSoBVOu8G
      64VwrA9bv3/cJVqZZ6/UzUTbHPU+Ogh24qhwF5QU8kXZEUI1To3YsPoftalgjX9G
      Ff0fLcLVC8nL3K9RiaDXxXbEYpWrYu39M3FCpAXAzV2PrNxsP9PKyNWHnBPc08z5
      tFj45/bHn+j31AVVvgWtqz0pLks57hc4Q7yW/2RoRYq2md1KI709OLNwtkWEOVqb
      qnraorh2TwGnNaOB5oX5/lJvKtlq778fw96jGqykBr0+DKozj9rlr1OGgYOKDwlD
      nsZJPAECgYEA+Oqf/fxtPdsNGiaL2Z/heewvtaxjw/WoEVBFECb6/y4Ro7aux9nB
      16FcVi79Cwfp0UTJ7cnZvYSmBk5GWEObEIAeo61lvm/QeltM5+usAPd5/TcHXLYe
      92OnXmq7h3F4UXEkMayak8Lpu/TdmR5uOaL+m4aEu+XMY5tlxqDCnyECgYEA1h4X
      jCPi7Ja5CHK7a2Ud4TL2DNpIBE6GSK9iQ+0xFL6TsiK2Sfu6n8mx2sh+Jm0KHTiE
      /gWHdHQZSSWiuULfHoYEq3Rq8S6Av3GsGtRSpO03j7BE8C20Vpt0FnNTjZmdzf2/
      YZxc5KuYLh9qeY7Y7ceOsWA8JckDgMHPYzyLAtECgYBALD0TPgDr8Y1vMIDdmlqH
      FF04eTk/TBYIYKltgJ81KqthibeFzp4q+W7UyUhzj5a4XQOySlfYhFpJReTc3JEd
      r+o2SH3ymuEkqmUpZZjyptrMbWN4g3t4TDjaHqo6QQbD+GdcZyNy9M1Np9N5pl7E
      fUEm14dg6d3H0Ehs7QVAAQKBgQDRUx3mLXc9oKRINBIyDerGLJILQqLBQxtYl81T
      ZuFizGWL8w+PCIAMkpxDrVpWqqcGpiiuRi2ElbPapOaOg2epaY/LJscd/j5z6uc8
      W3JoNljpKoRa4fO578Pv5tM6TYHOzlF5Veoiy/a8sI3hRNuiqkM/+TsUHY5FJDRh
      aeDk4QKBgCOHIevvR+MWuwakzD6lNCbb8H6fvZ3WRAT8BYYz3wW9YfnV4J4uh/Bl
      moWYgIK2UpkrhA8scMUC790FoybQeParQ35x7Jl91bmTKkCqsX63fyqqYhx3SXRl
      JSktmH4E2cGmosZisjB7COKHR32w0J5JCgaGInQxjldbGrwhZQpn
      -----END RSA PRIVATE KEY-----
    crt: |-
      -----BEGIN CERTIFICATE-----
      MIID2DCCAsCgAwIBAgIQDfFYteD4kZj4Sko2iy1IJTANBgkqhkiG9w0BAQsFADBX
      MRgwFgYDVQQKEw9NeSBPcmdhbml6YXRpb24xCzAJBgNVBAsTAkNBMS4wLAYDVQQD
```

```
    EyVNeSBPcmdhbml6YXRpb24gQ2VydGlmaWNhdGUgQXV0aG9yaXR5MB4XDTIxMDQy
    MDAwMDQ1OVoXDTIxMDQyMDAxMDQ1OVowGDEWMBQGA1UEAwwNKi5jaGctcHRjLnBv
    ZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANAxZCJoWvAiCYlXMF2w
    T5S/vw5kvf4qUYUEB8Xzuce40jp7HiB3RXevkMdhl/CgIgIVcpH+pH2nMAWEwE1N
    cQ4MFk5vNilUT4fpGQyI9sD3Ca9eJpkz+iOmtpmr1IA4Xsx3PRfP3XIWI93MPNV9
    JFOsWpqqykGJcUB+PDZCNKnMITurCgBxeHlnwq/DreOMQAyANSXFNgktUSkmRLVE
    Wft9D8vo/dOxJ46dWEd92SlB8N9KVPXaSJ3sn1wTf6U+nF9zHrWMLYJqVf6R6mPK
    W8ahn6MkYYEA7EkQpeHbZXnGmXmeI7kzqLXD5GRCxz+nrvIGi7cnBvZaWwDEJecN
    K/ECAwEAAaOB3jCB2zATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNVHRMBAf8EAjAA
    MIG1BgNVHREEga0wgaqCCWxvY2FsaG9zdIIbKi5jaGctcHRjLnBvZC5jbHVzdGVy
    LmxvY2FsghMqLm15ZGItaGVhZGxlc3Mtc3ZjghsqLm15ZGItaGVhZGxlc3Mtc3Zj
    LmNoZy1wdGOCHyoubXlkYi1oZWFkbGVzcy1zdmMuY2hnLXB0Yy5zdmOCLSoubXlk
    Yi1oZWFkbGVzcy1zdmMuY2hnLXB0Yy5zdmMuY2x1c3Rlci5sb2NhbDANBgkqhkiG
    9w0BAQsFAAOCAQEALnhliDflu+BHp5conq4dXBwD/Ti2YR5TWQixM/0a6OD4KecZ
    MmaLl0T+OJJvA/j2IufZpc7dzEx5mZDkR2CRmoq10qZXqCRTrBZSXm6ARQWoYpeg
    9c0l4f8roxrkMGUKVPTKUwAvbnNYhD2l6PlBPwMpkMUfqFaSEXMaPyQKhrtQxdpH
    WjuS54OP0lm0peYu/yiaD98LtrTXnb6jch84SKf6Vii4HAVQyMeJaW+dpkqcI2+V
    Q4fkwWYSJy8BNcmXCwvHDLDy+s4EXWvHafhusuUhcp4HyMb1A6hd5hJhgFSnEvLy
    kLA0L9LaScxee6V756Vt9TN1NGjwmwyQDOhnQQ==
    -----END CERTIFICATE-----
  cacrt: |-
    -----BEGIN CERTIFICATE-----
    MIIDXDCCAkSgAwIBAgIRAMPzF3BNFxT9HWE+NXlFQjQwDQYJKoZIhvcNAQELBQAw
    VzEYMBYGA1UEChMPTXkgT3JnYW5pemF0aW9uMQswCQYDVQQLEwJDQTEuMCwGA1UE
    AxMlTXkgT3JnYW5pemF0aW9uIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMTA0
    MTkwNDQ0MjNaFw0zMTA0MTcwNDQ0MjNaMFcxGDAWBgNVBAoTD015IE9yZ2FuaXph
    dGlvbjELMAkGA1UECxMCQ0ExLjAsBgNVBAMTJU15IE9yZ2FuaXphdGlvbiBDZXJ0
    aWZpY2F0ZSBBdXRob3JpdHkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
    AQC5t6CS23Glk65YMw5e4i4xHldyxkCZS67w/6LWqeIlYKmFAaEl83WWy8MHUpOb
    4mahtUafEzDEOX6+URf72J8m0voldQ5FYr1AyUOyX8U90wGFqhbEgKRqt7vZEwIe
    2961fwqHh6917zI4xmt5W6ZJ5dBQVtkhzB+Pf7O6KBYjHoCnBBkfNVzsfZQ/1hnR
    0UzimfAc7Ze+UNwhXJhinFRJ3YuR+xiOTpPkl1GXPhLgFSQhekz4KepcbQEQKejb
    jg0dum1oBYIXZTSSbiO9rNmfUVLB5DcV0vZbSrGxLjWLBt5U8N2xf2d1bvkQW+bw
    Kklf9OG26bAi27tujurzN3r3AgMBAAGjIzAhMA4GA1UdDwEB/wQEAwICpDAPBgNV
    HRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAM0CN3n5C/KOT4uZ4ewwKK
    rHmANBPVM9u6MJBO8U62HcqLeoCuDFeU8zmUjLHjsQaPX64mJZlR7T5y52gEKO5A
    0qsBz3pg/vJ5DJTv0698+1Q1hB9k3smQdksAim19FZqysB7J4zK/+8aJ/q2kIFvs
    Jk3ekwQdQ3xfggklBQVuf76gr1v0uYlPtfPffPlfcGZ06Im6mqbajenXoR1PxPB0
    +zyCS8DkgPtDulplruwvXCFMYw9TPbzXKlt7tlsqRXogYLnxWJDzM1nOYCnD+rDm
    qxenV9Ir8RqZ0XSYuUyzRka5N4dhIhrzTAiNdeU5gzynXOz67u/Iefz1iK9ZcdE3
    -----END CERTIFICATE-----
```

Deploy FEPCluster with automatic certificate management

Use the following yaml as an example to deploy a FEPCluster with Automatic Certificate Management. MTLS related parameters are highlighted in Red.

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: mydb
  namespace: my-namespace
spec:
  fep:
    usePodName: true
    patroni:
      tls:
        certificateName: mydb-patroni-cert
    postgres:
      tls:
        certificateName: mydb-fep-cert
    forceSsl: true
    podAntiAffinity: false
```

```
  mcSpec:
    limits:
      cpu: 500m
      memory: 700Mi
    requests:
      cpu: 200m
      memory: 512Mi
  customAnnotations:
    allDeployments: {}
  servicePort: 27500
  image:
    image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0'
    pullPolicy: IfNotPresent
  sysExtraLogging: false
  podDisruptionBudget: false
  instances: '3'
  syncMode: 'on'
fepChildCrVal:
  customPgAudit: |
    # define pg audit custom params here to override defaults.
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'
    [output]
    logger = 'auditlog'
    log_directory = '/database/log/audit'
    [rule]
  customPgHba: |
    # define pg_hba custom rules here to be merged with default rules.
    # TYPE     DATABASE        USER        ADDRESS        METHOD
    hostssl    all             all         0.0.0.0/0      cert
    hostssl    replication     all         0.0.0.0/0      cert
  customPgParams: >+
    # define custom postgresql.conf parameters below to override defaults.
    # Current values are as per default FEP deployment
    shared_preload_libraries='pgx_datamasking,pgaudit,pg_prewarm'
    session_preload_libraries='pg_prewarm'
    max_prepared_transactions = 100
    max_worker_processes = 30
    max_connections = 100
    work_mem = 1MB
    maintenance_work_mem = 12MB
    shared_buffers = 128MB
    effective_cache_size = 384MB
    checkpoint_completion_target = 0.8

    # tcp parameters
    tcp_keepalives_idle = 30
    tcp_keepalives_interval = 10
    tcp_keepalives_count = 3

    # logging parameters in default fep installation
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'

    log_directory = '/database/log'
    log_filename = 'logfile-%a.log'
    log_file_mode = 0600
    log_truncate_on_rotation = on
    log_rotation_age = 1d
    log_rotation_size = 0
    log_checkpoints = on
    log_line_prefix = '%e %t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h'
    log_lock_waits = on
```

```
    log_autovacuum_min_duration = 60s
    logging_collector = on
    pgaudit.config_file='/opt/app-root/src/pgaudit-cfg/pgaudit.conf'
    log_replication_commands = on
    log_min_messages = WARNING
    log_destination = stderr

    # wal_archive parameters in default fep installation
    archive_mode = on
    archive_command = '/bin/true'
    wal_level = replica
    max_wal_senders = 12
    wal_keep_segments = 64

  storage:
    dataVol:
      size: 2Gi
      storageClass: nfs-client
    walVol:
      size: 1200Mi
      storageClass: nfs-client
    logVol:
      size: 1Gi
      storageClass: nfs-client
  sysUsers:
    pgAdminPassword: admin-password
    pgdb: mydb
    pgpassword: mydbpassword
    pguser: mydbuser
    pgrepluser: repluser
    pgreplpassword: repluserpwd
    pgRewindUser: rewinduser
    pgRewindPassword: rewinduserpwd
    pgAdminTls:
      certificateName: mydb-postgres-cert
      sslMode: verify-full

    pgrepluserTls:
      certificateName: mydb-repluser-cert
      sslMode: verify-full

    pgRewindUserTls:
      certificateName: mydb-rewinduser-cert
      sslMode: verify-full

    tdepassphrase: tde-passphrase
  systemCertificates:
    key: |-
      -----BEGIN RSA PRIVATE KEY-----
      MIIEowIBAAKCAQEA0DFkImha8CIJiVcwXbBPlL+/DmS9/ipRhQQHxfO5x7jSOnse
      IHdFd6+Qx2GX8KAiAhVykf6kfacwBYTATU1xDgwWTm82KVRPh+kZDIj2wPcJr14m
      mTP6I6a2mavUgDhezHc9F8/dchYj3cw81X0kU6xamqrKQYlxQH48NkI0qcwhO6sK
      AHF4eWfCr8Ot44xADIA1JcU2CS1RKSZEtURZ+30Py+j907Enjp1YR33ZKUHw30pU
      9dpIneyfXBN/pT6cX3MetYwtgmpV/pHqY8pbxqGfoyRhgQDsSRCl4dtlecaZeZ4j
      uTOotcPkZELHP6eu8gaLtycG9lpbAMQl5w0r8QIDAQABAoIBACq213qPuoimExrQ
      fqXaNJmqNYK4fJqXCB6oUwf0F1u4ubkx5V532hLSPHwLs+a0lAWlbNozSoBVOu8G
      64VwrA9bv3/cJVqZZ6/UzUTbHPU+Ogh24qhwF5QU8kXZEUI1To3YsPoftalgjX9G
      Ff0fLcLVC8nL3K9RiaDXxXbEYpWrYu39M3FCpAXAzV2PrNxsP9PKyNWHnBPc08z5
      tFj45/bHn+j31AVVvgWtqz0pLks57hc4Q7yW/2RoRYq2md1KI709OLNwtkWEOVqb
      qnraorh2TwGnNaOB5oX5/lJvKtlq778fw96jGqykBr0+DKozj9rlr1OGgYOKDwlD
      nsZJPAECgYEA+Oqf/fxtPdsNGiaL2Z/heewvtaxjw/WoEVBFECb6/y4Ro7aux9nB
      16FcVi79Cwfp0UTJ7cnZvYSmBk5GWEObEIAeo61lvm/QeltM5+usAPd5/TcHXLYe
      92OnXmq7h3F4UXEkMayak8Lpu/TdmR5uOaL+m4aEu+XMY5tlxqDCnyECgYEA1h4X
```

```
    jCPi7Ja5CHK7a2Ud4TL2DNpIBE6GSK9iQ+0xFL6TsiK2Sfu6n8mx2sh+Jm0KHTiE
    /gWHdHQZSSWiuULfHoYEq3Rq8S6Av3GsGtRSpO03j7BE8C20Vpt0FnNTjZmdzf2/
    YZxc5KuYLh9qeY7Y7ceOsWA8JckDgMHPYzyLAtECgYBALD0TPgDr8Y1vMIDdmlqH
    FF04eTk/TBYIYKltgJ81KqthibeFzp4q+W7UyUhzj5a4XQOySlfYhFpJReTc3JEd
    r+o2SH3ymuEkqmUpZZjyptrMbWN4g3t4TDjaHqo6QQbD+GdcZyNy9M1Np9N5pl7E
    fUEm14dg6d3H0Ehs7QVAAQKBgQDRUx3mLXc9oKRINBIyDerGLJILQqLBQxtYl81T
    ZuFizGWL8w+PCIAMkpxDrVpWqqcGpiiuRi2ElbPapOaOg2epaY/LJscd/j5z6uc8
    W3JoNljpKoRa4fO578Pv5tM6TYHOzlF5Veoiy/a8sI3hRNuiqkM/+TsUHY5FJDRh
    aeDk4QKBgCOHIevvR+MWuwakzD6lNCbb8H6fvZ3WRAT8BYYz3wW9YfnV4J4uh/Bl
    moWYgIK2UpkrhA8scMUC790FoybQeParQ35x7Jl91bmTKkCqsX63fyqqYhx3SXRl
    JSktmH4E2cGmosZisjB7COKHR32w0J5JCgaGInQxjldbGrwhZQpn
    -----END RSA PRIVATE KEY-----
  crt: |-
    -----BEGIN CERTIFICATE-----
    MIID2DCCAsCgAwIBAgIQDfFYteD4kZj4Sko2iy1IJTANBgkqhkiG9w0BAQsFADBX
    MRgwFgYDVQQKEw9NeSBPcmdhbml6YXRpb24xCzAJBgNVBAsTAkNBMS4wLAYDVQQD
    EyVNeSBPcmdhbml6YXRpb24gQ2VydGlmaWNhdGUgQXV0aG9yaXR5MB4XDTIxMDQy
    MDAwMDQ1OVoXDTIxMDQyMDAxMDQ1OVowGDEWMBQGA1UEAwwNKi5jaGctcHRjLnBv
    ZDCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBANAxZCJoWvAiCYlXMF2w
    T5S/vw5kvf4qUYUEB8Xzuce40jp7HiB3RXevkMdhl/CgIgIVcpH+pH2nMAWEwE1N
    cQ4MFk5vNilUT4fpGQyI9sD3Ca9eJpkz+iOmtpmr1IA4Xsx3PRfP3XIWI93MPNV9
    JFOsWpqqykGJcUB+PDZCNKnMITurCgBxeHlnwq/DreOMQAyANSXFNgktUSkmRLVE
    Wft9D8vo/dOxJ46dWEd92SlB8N9KVPXaSJ3sn1wTf6U+nF9zHrWMLYJqVf6R6mPK
    W8ahn6MkYYEA7EkQpeHbZXnGmXmeI7kzqLXD5GRCxz+nrvIGi7cnBvZaWwDEJecN
    K/ECAwEAAaOB3jCB2zATBgNVHSUEDDAKBggrBgEFBQcDATAMBgNVHRMBAf8EAjAA
    MIG1BgNVHREEga0wgaqCCWxvY2FsaG9zdIIbKi5jaGctcHRjLnBvZC5jbHVzdGVy
    LmxvY2FsghMqLm15ZGItaGVhZGxlc3Mtc3Zjghsq Lm15ZGItaGVhZGxlc3Mtc3Zj
    LmNoZy1wdGOCHyoubXlkYi1oZWFkbGVzcy1zdmMuY2hnLXB0Yy5zdmOCLSoubXlk
    Yi1oZWFkbGVzcy1zdmMuY2hnLXB0Yy5zdmMuY2x1c3Rlci5sb2NhbDANBgkqhkiG
    9w0BAQsFAAOCAQEALnhliDflu+BHp5conq4dXBwD/Ti2YR5TWQixM/0a6OD4KecZ
    MmaLl0T+OJJvA/j2IufZpc7dzEx5mZDkR2CRmoq10qZXqCRTrBZSXm6ARQWoYpeg
    9c0l4f8roxrkMGUKVPTKUwAvbnNYhD2l6PlBPwMpkMUfqFaSEXMaPyQKhrtQxdpH
    WjuS54OP0lm0peYu/yiaD98LtrTXnb6jch84SKf6Vii4HAVQyMeJaW+dpkqcI2+V
    Q4fkwWYSJy8BNcmXCwvHDLDy+s4EXWvHafhusuUhcp4HyMb1A6hd5hJhgFSnEvLy
    kLA0L9LaScxee6V756Vt9TN1NGjwmwyQDOhnQQ==
    -----END CERTIFICATE-----
  cacrt: |-
    -----BEGIN CERTIFICATE-----
    MIIDXDCCAkSgAwIBAgIRAMPzF3BNFxT9HWE+NXlFQjQwDQYJKoZIhvcNAQELBQAw
    VzEYMBYGA1UEChMPTXkgT3JnYW5pemF0aW9uMQswCQYDVQQLEwJDQTEuMCwGA1UE
    AxMlTXkgT3JnYW5pemF0aW9uIENlcnRpZmljYXRlIEF1dGhvcml0eTAeFw0yMTA0
    MTkwNDQ0MjNaFw0zMTA0MTcwNDQ0MjNaMFcxGDAWBgNVBAoTD015IE9yZ2FuaXph
    dGlvbjELMAkGA1UECxMCQ0ExLjAsBgNVBAMTJU15IE9yZ2FuaXphdGlvbiBDZXJ0
    aWZpY2F0ZSBBdXRob3JpdHkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
    AQC5t6CS23Glk65YMw5e4i4xHldyxkCZS67w/6LWqeIlYKmFAaEl83WWy8MHUpOb
    4mahtUafEzDEOX6+URf72J8m0voldQ5FYr1AyUOyX8U90wGFqhbEgKRqt7vZEwIe
    2961fwqHh6917zI4xmt5W6ZJ5dBQVtkhzB+Pf7O6KBYjHoCnBBkfNVzsfZQ/1hnR
    0UzimfAc7Ze+UNwhXJhinFRJ3YuR+xiOTpPkl1GXPhLgFSQhekz4KepcbQEQKejb
    jg0dum1oBYIXZTSSbiO9rNmfUVLB5DcV0vZbSrGxLjWLBt5U8N2xf2d1bvkQW+bw
    Kklf9OG26bAi27tujurzN3r3AgMBAAGjIzAhMA4GA1UdDwEB/wQEAwICpDAPBgNV
    HRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAM0CN3n5C/KOT4uZ4ewwKK
    rHmANBPVM9u6MJBO8U62HcqLeoCuDFeU8zmUjLHjsQaPX64mJZlR7T5y52gEKO5A
    0qsBz3pg/vJ5DJTv0698+1Q1hB9k3smQdksAim19FZqysB7J4zK/+8aJ/q2kIFvs
    Jk3ekwQdQ3xfggklBQVuf76gr1v0uYlPtfPffPlfcGZ06Im6mqbajenXoR1PxPB0
    +zyCS8DkgPtDulplruwvXCFMYw9TPbzXKlt7tlsqRXogYLnxWJDzM1nOYCnD+rDm
    qxenV9Ir8RqZ0XSYuUyzRka5N4dhIhrzTAiNdeU5gzynXOz67u/Iefz1iK9ZcdE3
    -----END CERTIFICATE-----
```

4.8.2.4 Configurable Parameters

To enable MTLS, make changes to the following parameters.

Key	Value	Details
spec.fep.usePodName	True	For MTLS, this key must be defined and set to true. For TLS connection without MTLS, it can be omitted. However, it is recommended to set this to true as well.
spec.fep.patroni.tls.certificateName	<secret-name>	Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for Patroni REST API. For MTLS Patroni REST API communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.
spec.fep.patroni.tls.caName	<configmap-name>	Name of Kubernetes configmap that contains the CA bundle. If using cert-manager, the ca.crt is already included in the secret above. In this situation, this key can be omitted.
spec.fep.postgres.tls.certificateName	<secret-name>	Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for Postgres server. For MTLS Postgres communication, this key must be defined. The private key can be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.
spec.fep.postgres.tls.caName	<configmap-name>	Name of Kubernetes configmap that contains the CA bundle. If using cert-manager, the ca.crt is already included in the secret above. In this situation, this key can be omitted.
spec.fep.postgres.tls.privateKeyPassword	<secret-name>	Name of Kubernetes secret that contains the password for the private key for Postgres Server.
spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName	<secret-name>	Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for "postgres" user. For MTLS Postgres communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.
spec.fepChildCrVal.sysUsers.pgAdminTls.caName	<configmap-name>	Name of Kubernetes configmap that contains the CA bundle. If using cert-manager, the ca.crt is already included in the secret

Key	Value	Details
		above. In this situation, this key can be omitted.
spec.fepChildCrVal.sysUsers.pgAdminTls.sslMode	verify-full	For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.
spec.fepChildCrVal.sysUsers.pgrepluserTls.certificateName	<secret-name>	Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for "repluser" user. For MTLS Postgres communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.
spec.fepChildCrVal.sysUsers.pgrepluserTls.caName	<configmap-name>	Name of Kubernetes configmap that contains the CA bundle. If using cert-manager, the ca.crt is already included in the secret above. In this situation, this key can be omitted.
spec.fepChildCrVal.sysUsers.pgrepluserTls.sslMode	verify-full	For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.
spec.fepChildCrVal.sysUsers.pgRewindUserTls.certificateName	<secret-name>	Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for "rewinduser" user. For MTLS Postgres communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.
spec.fepChildCrVal.sysUsers.pgRewindUserTls.caName	<configmap-name>	Name of Kubernetes configmap that contains the CA bundle. If using cert-manager, the ca.crt is already included in the secret above. In this situation, this key can be omitted.
spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode	verify-full	For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.

It is also required to customize pg_hba.conf to perform MTLS. Below are two possible settings.

spec.fep.customPgHba	hostssl all all 0.0.0.0/0 cert hostssl replication all 0.0.0.0/0 cert

The above setting will force FEP server to perform certification authentication. At the same time verify the authenticity of client certificate.

spec.fep.customPgHba	hostssl all all 0.0.0.0/0 md5 clientcert=verify-full hostssl replication repluser 0.0.0.0/0 md5 clientcert=verify-full

The above setting will foce FEP server to perform md5 authentication as well as verifying the authenticity of client certificate.

4.9 Replication Slots

4.9.1 Setting Up Logical Replication using MTLS

This section describes setup of logical replication.

To setup logical replication using MTLS, follow these steps:

1. Create two FEPClusters - to act as Publisher and Subscriber) and ensure that they can communicate with each other. You can see the creation of FEPCluster in the "4.1 Deploying FEPCluster using Operator".
2. To setup Publisher, make following changes to the FEPCluster yaml of the cluster that you want to use as publisher:
   a. Add section replicationSlots under spec.fep to create replication slots.

      The "**database**" should be the name of the database for which we are setting up logical replication.

   b. Add section postgres under spec.fep as shown below.

      caName = enter the name of configmap created for the CA

      certificateName = secret created by the end user that contains server certificate

c. Change the value of wal_level parameter under spec.fepChildCrVal.customPgParams from replica to logical.

```
301
302     archive_mode = on
303
304     archive_command = 'pgbackrest --stanza=backupstanza
305     --config=/database/userdata/pgbackrest.conf archive-push %p'
306
307     wal_level = logical
308
309     max_wal_senders = 12
310
311     wal_keep_size = 401
```

d. Add entry under spec.fepChildCrVal.customPgHba as shown below.

This requires the client to present a certificate and only certificate authentication is allowed.

Replace "SubClusterName" and "SubNamespace" with the appropriate values as per the Subscriber FEPCluster.

```
  [rule]
customPgHba: |
  # define pg_hba custom rules here to be merged with default rules.
  # TYPE      DATABASE        USER        ADDRESS         METHOD
  hostssl all all .<SubClusterName>-primary-svc.<SubNamespace>.svc.cluster.local cert
customPgParams: >
```

3. To setup Subscriber, make following changes to the FEPCluster yaml of the cluster that you want to use as subscriber:

a. Add customCertificates under spec.fepChildCrVal as shown below.

caName = enter the name of configmap created for the CA (i.e. The CA certificate which is used to sign/authenticate the server/ client certificates is mounted as a configMap called 'cacert')

certificateName = secret created by end user that contains a client certificate which can be verified by the server

username = name of the role created on publisher cluster for logical replication

```
74    fepChildCrVal:
75      customCertificates:
76        - caName: cacert
77          certificateName: my-logicalrepl-cert
78          userName: logicalrepluser
79      customPgAudit: |
80        # define pg audit custom params here to override defaults.
81        # if log volume is not defined, log_directory should be
```

4. Connect to the pod terminal of the Publisher FEPCluster and then connect to the postgres database as shown below.

```
sh-4.4$ psql -h /tmp -p 27500 -U postgres
Password for user postgres:
psql (13.1)
Type "help" for help.

postgres=#
```

5. Next, on the publisher side, connect to the database that contains the tables you want to replicate and create a role e.g., logicalrepluser and give the required permissions to this role.

   Consider the below image as example only, the privileges to grant may differ as per the requirements.

```
db1=# CREATE ROLE logicalrepluser WITH REPLICATION LOGIN PASSWORD 'my_password';
CREATE ROLE
db1=# GRANT ALL PRIVILEGES ON DATABASE db1 TO logicalrepluser;
GRANT
db1=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO logicalrepluser;
GRANT
db1=#
```

6. At the Publisher side, create a publication and alter the publication to add the tables that need to be replicated.

```
db1=# create publication my_publication;
CREATE PUBLICATION
db1=# alter publication my_publication add table my_table;
ALTER PUBLICATION
db1=#
```

7. At the subscriber side, the custom certificates added in the above step 3.a will be mounted at the path /tmp/custom_certs/ as shown:

```
sh-4.4$ ls -rlt /tmp/custom_certs
total 0
drwxr-xr-t. 3 1001190000 root 103 Aug 10 10:08 logicalrepluser
sh-4.4$ ls -rlt /tmp/custom_certs/logicalrepluser
total 0
lrwxrwxrwx. 1 1001190000 root 14 Aug 10 10:08 tls.key -> ..data/tls.key
lrwxrwxrwx. 1 1001190000 root 14 Aug 10 10:08 tls.crt -> ..data/tls.crt
lrwxrwxrwx. 1 1001190000 root 13 Aug 10 10:08 ca.crt -> ..data/ca.crt
sh-4.4$
```

8. The structure of the table to be replicated should be present in the subscriber cluster since logical replication only replicates the data and not the table structure.

   Create a subscription as shown below:

```
db1=# CREATE SUBSCRIPTION my_subscription CONNECTION 'host=fepcluster-publisher-primary-svc.ns-a.svc.cluster.local port=27500 sslcert=/tmp/custom_certs/logicalrepluser/tls.crt sslkey=/tmp/custom_certs/logicalrepluser/tls.key sslrootcert=/tmp/custom_certs/logicalrepluser/ca.crt sslmode=verify-full dbname=db1 user=logicalrepluser' PUBLICATION my_publication WITH (slot_name=myslot1, create_slot=false);
CREATE SUBSCRIPTION
```

   The command in the above example is :

```
CREATE SUBSCRIPTION my_subscription CONNECTION 'host=fepcluster-publisher-primary-svc.ns-
a.svc.cluster.local port=27500 sslcert=/tmp/custom_certs/logicalrepluser/tls.crt sslkey=/tmp/
custom_certs/logicalrepluser/tls.key sslrootcert=/tmp/custom_certs/logicalrepluser/ca.crt
sslmode=verify-full password=my_password user=logicalrepluser dbname=db1' PUBLICATION
my_publication WITH (slot_name=myslot1, create_slot=false);

Host = primary service of the publisher FEP Cluster
sslcert, sslkey, sslrootcert = path to certificates mounted on the Subscriber FEP Cluster
user= Role created on the Publisher side
```

```
password= password for the role
dbname= database which contains the tables to be replicated
```

Where

Host = primary service of the publisher FEP Cluster
sslcert, sslkey, sslrootcert = path to certificates mounted on the Subscriber FEP Cluster
user= Role created on the Publisher side and used to establish logical replication connection fromSubscriber to Publisher
dbname= database which contains the tables to be replicated

4.10 FEP Logging

FEPCluster generates log files and auditlog files, if configured, over the lifetime of execuation. These log files can be useful for understanding cluster healthness and debugging purpose. By default, the log files are stored on persistent volume of the container. User can enable log monitoring feature by forwarding those log files and auditlog files to a analytics platform such as Elasticsearch.

There are two steps to enable monitoring and forwarding.

1. FEPLogging Configuration - Creating FEP Logging instance
2. FEPCluster configuration - Enabling logging in FEPCluster

The FEP Logging instance is a standalone container running fluentd. It accepts log forwarded from FEP Clusters and aggagrate data according to log entries severity and present that to Prometheus for monitoring and alerting purpose. It can optionally be configured to forward those logs to an Elasticsearch instance for detail analysis.

The Fujitsu Enterprise Postgres 17 operator has been extended to allow PostgreSQL logs to be forwarded to supported destinations via Fluent-Bit. For more flexibility in choosing where to forward your logs, refer to "4.10.2 Configuring FEPCluster Remote Logging".

When logging is enabled on FEPCluster, a sidecar, containing fluentbit, will be deployed alongside the FEP server container. This fluentbit sidecar will monitor the FEP server log files and auditlog files on persistent volume and forward to the FEP Logging instance.

Multiple FEPClusters can forward logs to single FEPLogging instance.

User can have two types of connection between FEPCluster & FEPLogging

- Insecure connection: Without TLS/MTLS certificates
- Secure connection: With TLS/MTLS certificates

For the secure connections between the components, User have two options:

- User can use their own certificates
- User can generate self signed certificates (see "4.8.2.2 Automatic Certificate Management")

The FEP Logging instance can run standalone without additional component. For detail log analysis, the user can configure the FEP Logging instance to forward logs to Elastic Stack or Elastic Cloud. Please consult the Elastic Document on how to deploy a Elastic Stack or sign up to Elastic Cloud.

4.10.1 FEPLogging Configuration

This section describes how to deploy and configure FEP Logging instance via the FEPLogging custom resource. FEPLogging is a separate CR which will accept logs sent from FEPCluster and forwards them to Elasticsearch or Prometheus for raising alarm. User must create FEPLogging CR before enabling FEPCluster logging feature.

4.10.1.1 FEPLogging Custom Resources - spec

The fepLogging section needs to be added under spec to define required parameters for FEPLogging configuration.

Following is a sample template :

```
spec:
  fepLogging:
    elastic:
      authSecret:
```

```
      secretName: elastic-auth
      passwordKey: password
      userKey: username
    host: elastic-passthrough.apps.openshift.com
    logstashPrefix: postgres
    port: 443
    scheme: https
    sslVerify: true
    tls:
      certificateName: elastic-cert
      caName: elastic-cacert
  image:
    pullPolicy: IfNotPresent
  mcSpec:
    limits:
      cpu: 500m
      memory: 700Mi
    requests:
      cpu: 200m
      memory: 512Mi
  restartRequired: false
  sysExtraLogging: false
  scrapeInterval: 30s
  scrapeTimeout: 30s
  tls:
    certificateName: fluentd-cert
    caName: cacert
prometheus:
  …
```

Below is the list of all parameters defined in the fepLogging section, along with their brief description

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
spec.fepLogging.image.image	Optional	Fluentd Image of FEPLogging	Yes
spec.fepLogging.image.pullPolicy	Required	Fluentd Image pull policy of FEPLogging	Yes
spec.fepLogging.mcSpec.limits.cpu	Required	Max CPU allocated to fluentd container	Yes
spec.fepLogging.mcSpec.limits.memory	Required	Max memory allocated to fluentd container	Yes
spec.fepLogging.mcSpec.requests.cpu	Required	CPU allocation at start for fluentd container	Yes
spec.fepLogging.mcSpec.requests.memory	Required	Memory allocation at start for fluentd container	Yes
spec.fepLogging.sysExtraLogging	Required	To turn on extra debugging messages for operator, set value to true. It can be turned on/off at any time	Yes
spec.fepLogging.restartRequired	Required	To restart FEPLogging instance for applying any new configuration for example after certificate rotation	Yes
spec.fepLogging.scrapeInterval	Optional	Scrape interval for Prometheus to fetch metrics from FEPLogging instance	Yes
spec.fepLogging.scrapeTimeout	Optional	Scrape Timeout for Prometheus to fetch metrics from FEPLogging instance	Yes
spec.fepLogging.elastic.host	Optional	Target Elasticsearch host name	Yes
spec.fepLogging.elastic.port	Optional	Target Elasticsearch port number	Yes

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
spec.fepLogging.elastic.authSecret.secretName	Optional	Secret name which contains Elasticsearch authentication username & password	Yes
spec.fepLogging.elastic.authSecret.userKey	Optional	Username key specified in Elasticsearch authentication secret	Yes
spec.fepLogging.elastic.authSecret.passwordKey	Optional	Password key specified in Elasticsearch authentication secret	Yes
spec.fepLogging.elastic.logstashPrefix	Optional	Logstash prefix to differentiate index pattern in elastic search. Default value is postgres	Yes
spec.fepLogging.elastic.auditLogstashPrefix	Optional	Logstash prefix to differentiate index pattern in elastic search for auditlog. If not specified, it will default to the same value as ‘logstashPrefix’.	Yes
spec.fepLogging.elastic.scheme	Optional	Connection scheme between FEPLogging & Elasticsearch. Possible options http & https	Yes
spec.fepLogging.elastic.sslVerify	Optional	Set to true if you want to verify ssl certificate. If set to false then will not consider TLS crtificate	Yes
spec.fepLogging.elastic.tls.certificateName	Optional	Kubernetes secret name which holds fluentd certificate	Yes
spec.fepLogging.elastic.tls.caName	Optional	Kubernetes configmap which holds cacert of Elasticsearch to verify Elasticsearch TLS connection	Yes
spec.fepLogging.tls.certificateName	Optional	Kubernetes secret name which holds Fluentd certificate	Yes
spec.fepLogging.tls.caName	Optional	Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging & Prometheus	Yes
spec.prometheus.tls.certificateName	Optional	Kubernetes secret name which holds Prometheus certificate	Yes
spec.prometheus.tls.caName	Optional	Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging & Prometheus	Yes

4.10.1.1.1 Define fepLogging image

The image property is used to specify other than default Fluentd image and it’s pullPolicy from FEPLogging CR.

If not specified it will use default image provided by Operator.

Example)

```
spec:
  fepLogging:
    image:
      image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0'
      pullPolicy: IfNotPresent
```

4.10.1.1.2 Define fepLogging mcSpec

FEPLogging container Memory & CPU configuration can be provided by mcSpec properties.

Example)

```
spec:
  fepLogging:
    mcSpec:
      limits:
        cpu: 500m
        memory: 700Mi
      requests:
        cpu: 200m
        memory: 512Mi
```

4.10.1.1.3 Define fepLogging restartRequired

If FEPLogging required to be restarted to apply any new change, for example, after certificate rotation, FEPLogging container can be restarted by setting restartRequired flag as true. Default value of this flag is False. This flag will change back to false once the pod is restarted

Example)

```
spec:
  fepLogging:
    restartRequired: true
```

4.10.1.1.4 Define fepLogging scrapeInterval and scrapeTimeout

scrapeInterval and scrapeTimeout properties of FEPLogging are optional. These properties are used by Prometheus Servicemonitor to configure metrics fetching interval(scrapeInterval) and timeout of request.

Example)

```
spec:
  fepLogging:
    scrapeInterval: 30s
    scrapeTimeout: 30s
```

4.10.1.1.5 Define fepLogging elastic

To forward logs from FEPLogging(Fluentd) to Elasticsearch, need to configure elastic property. This is optional property. Elasticsearch server and certificates will be configured by user.

To configure log forwarding to Elasticsearch, the following properties are required.

- authSecret
- host
- port
- logstashPrefix
- auditLogstashPrefix
- scheme
- sslVerify
- tls(if sslVerify set to true)

Configure Elasticsearch server and use it's host name and port.

Here tls property is optional and works with sslVerify flag. To enable secure connection and tls verification set sslVerify true and provide valid certificateName & caName.

Elasticsearch caName is mandatory which holds CA cert of elastic search server.

Example)

```
  spec:
    fepLogging:
      elastic:
        authSecret:
          passwordKey: password
          secretName: elastic-auth
          userKey: username
        host: elastic-passthrough.apps.openshift.com
        logstashPrefix: postgres
        auditLogstashPrefix: postgres
        port: 443
        scheme: https
        sslVerify: false
        tls:
          certificateName: fluentd-cert
          caName: elastic-cacert
```

4.10.1.1.6 Define authSecret for elastic

authSecret is the secret which contains username & password in base64 format for elastic search authentication

Example)

```
kind: Secret
apiVersion: v1
metadata:
  name: elastic-auth
  namespace: my-namespace
data:
  password: OFBobzlyRUJWOGg1Mk0xcXdaMUQ5bzQ0
  username: ZWxhc3RpYw==
type: Opaque
```

4.10.1.1.7 Define fepLogging TLS

FEPLogging has optional TLS property. If user wants to forward logs from FEPCluster to FEPLogging instance over a secure connection, the TLS configuration for FEPCluster(remoteLogging section) and the TLS configuration for FEPLogging and Prometheus are mandatory. Configuring TLS configuration on just fepLogging or Prometheus will not work.

When a self signed certificate is used, caName can be skipped.

Example)

```
spec:
    fepLogging:
      tls:
        certificateName: fluentd-cert
        caName: cacert
```

4.10.1.1.8 Define Prometheus TLS

If secured connection between FEPLogging and FEPCluster is required, then TLS configuration for FEPLogging and Prometheus are mandatory. Configuring TLS on just fepLogging or Prometheus will not work.

When a self signed certificate is used, caName can be skipped.

Example)

```
spec:
  fepLogging:
   …
  prometheus:
     tls:
       certificateName: prometheus-cert
       caName: cacert
```

4.10.2 Configuring FEPCluster Remote Logging

This section describes how to enable logging in FEPCluster. FEP cluster provides a feature to forward logs to remote Fluentd(FEPLogging) and FEPLogging instance will forward the same logs to Elasticsearch(Optional) & Prometheus.

The following destinations can be used with remote logging:

- Fluentd
- Elasticsearch
- Azure BLOB
- Amazon CloudWatch

For more information about log forwarding with the remote logging feature, refer to "Appendix E Fluent Bit Integration Using Custom Secret".

4.10.2.1 FEP Custom Resources - spec.fep.remoteLogging

The remoteLogging section needs to be added under fep to define required parameters for remoteLogging configuration.

Following is a sample template:

```
spec:
  fep
    …
   remoteLogging:
      enable: true
      fluentdName: new-fep-logging
      tls:
        certificateName: fluentbit-cert
        caName: cacert
    …
```

Below is the list of all parameters defined in the remoteLogging section, along with their brief description:

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
remoteLogging.enable	Required	The 'enable' is set to true for enabling Logging feature	No
remoteLogging.fluentdName	Optional	The 'fluentdName' is the name of the FEPLogging CR where logs will be forwarded Specify this when transferring logs using the FEP log feature	Yes
remoteLogging.tls.certificateName	Optional	Secret name which contains MTLS certs of fluentbit	No
remoteLogging.tls.caName	Optional	Cacert of Fluentd for ssl verification	No
remoteLogging.image	Optional	Fluentbit image for remoteLogging	Yes
remoteLogging.pullPolicy	Optional	Fluentbit image pull policy	Yes

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
remoteLogging.mcSpec.limits.cpu	Optional	CPU allocation limit for fluentbit	Yes
remoteLogging.mcSpec.limits.memory	Optional	Memory allocation limit for fluentbit	Yes
remoteLogging.mcSpec.requests.cpu	Optional	CPU allocation request for fluentbit	Yes
remoteLogging.mcSpec.requests.memory	Optional	Memory allocation request for fluentbit	Yes
remoteLogging.fluentbitParams.memBufLimit	Optional	Defines the Mem_Buf_Limit in Fluentbit. This will affect all sections that use this parameter	Yes
remoteLogging.fluentbitConfigSecretRef	Optional	Specify the name of the Secret that contains fluent-bit.yaml when using the log forwarding feature with remote logging. If fluentbitConfigSecretRef is not defined or is defined but the referenced secret does not exist, the FEP operator creates a default Secret <fep-cluster>-fluent-bit-conf and updates this parameter with <fep-cluster>-fluent-bit-conf. If the referenced secret exists, the named secret is mounted to fep-logging-fluent-bit under /fluent-bit/etc.	Yes
remoteLogging.awsCredentialSecretRef	Optional	The 'awsCredentialSecretRef' is the name of a Secret that contains credentials to the AWS service. The credentials are stored in a configuration file and a credentials file. The configuration file must be named "config" and the credentials file must be named "credentials". If the referenced secret exists, the named secret will be mounted to fep-logging-fluent-bit under /fluent-bit/aws.	Yes

4.10.2.1.1 Define remoteLogging enable and fluentdName

Specify this if you want to forward logs to a Fluentd container created by the FEP logging feature.

The enable flag is used to describe that FEPCluster will enable logging feature if set as true.

If enable flag set as true then fluentdName is the mandatory field. It will describe the FEPLogging CR name to which FEPCluster will forwards the logs.

If the enable flag is set as false, the FEPCluster will not enable logging feature.

Example)

```
fep:
  remoteLogging:
    enable: true
    fluentdName: new-fep-logging
```

If user wants to update existing FEPCluster with logging feature then FEPCluster log_destination configuration must be set as **csvlogs.** For new cluster it will be already set.

Example)

```
fep:
  …
 remoteLogging:
```

```
      enable: true
      fluentdName: new-fep-logging
    ...

  fepChildCrVal:
    customPgParams:
       ...
      log_destination = csvlog
       ...
```

4.10.2.1.2 Define remoteLogging tls

When FEPCluster uses secure connection for remoteLogging, then TLS section is mandatory.

In the TLS section, provide the secret name that contains certificate and private key that is used for ssl verification.

For MTLS connection caName is required to mutually validate certificate.

Example)

```
  fep:
    remoteLogging:
      enable: true
      fluentdName: new-fep-logging
      tls:
        certificateName: fluentbit-cert-secret
        caName: ca-cert
```

Note

The Elasticsearch server is configured by user and it is NOT part of FEPLogging deployment by operator.

4.10.2.1.3 Define remoteLogging image

The image property is used to specify other than default Fluentbit image and it's pullPolicy.

If not specified it will use default image provided by Operator.

Example)

```
spec:
  fep:
    remoteLogging:
      image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-fluentbit:ubi9-17-1.0'
      pullPolicy: IfNotPresent
```

4.10.2.1.4 Define remoteLogging fluentbitConfigSecretRef

Specify this if you want to use the log forwarding function via the remote logging feature. 'fluentbitConfigSecretRef' is the name of the secret that contains fluentbit.yaml. Refer to "Appendix E Fluent Bit Integration Using Custom Secret".

If not specified, the default secret created by the operator, <fep-cluster>-fluent-bit-conf, is used.

Example)

```
spec:
  fep:
    remoteLogging:
      fluentbitConfigSecretRef: ' custom-secret'
```

4.10.2.1.5 Define remoteLogging awsCredentialSecretRef

'awsCredentialSecretRef' is the name of the secret that contains the credentials to the AWS service.

Example)

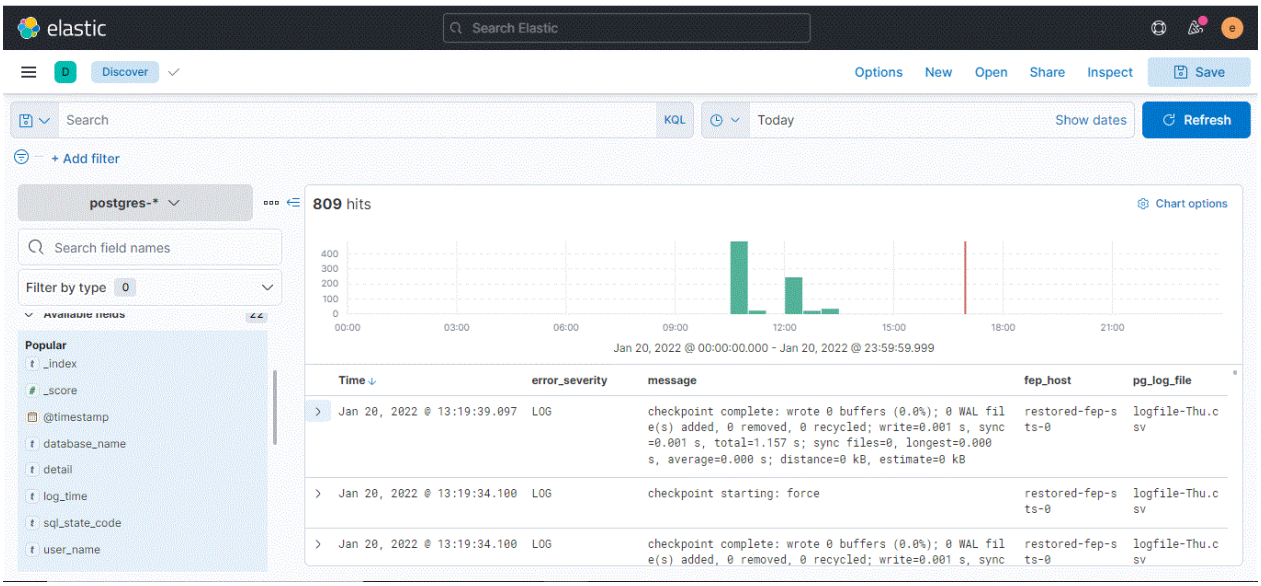
```
spec:
  fep:
    remoteLogging:
      awsCredentialSecretRef: 'aws-credential'
```

4.10.3 FEPLogging Operations

4.10.3.1 Log Forwarding to Elasticsearch

If the user has provided Elasticsearch configuration in the FEPLogging CR, and FEPCluster is configured to send server log files and auditlog files to that FEPLogging instance, those logs will be visible on Elasticsearch stack or Elastic Cloud. Assuming Elasticsearch has been configured with Kibana then logs will be visible in Kibana Dashborad. User can use fep log csv fields to create various Dashbord in Kiabana as well. LogstashPrefix and auditLogstashPrefix will be used to filter logs of specific FEPLogging instance.

User can verify if FEPLogging feature is configured properly or not by checking real time FEP logs are populating to the destination.

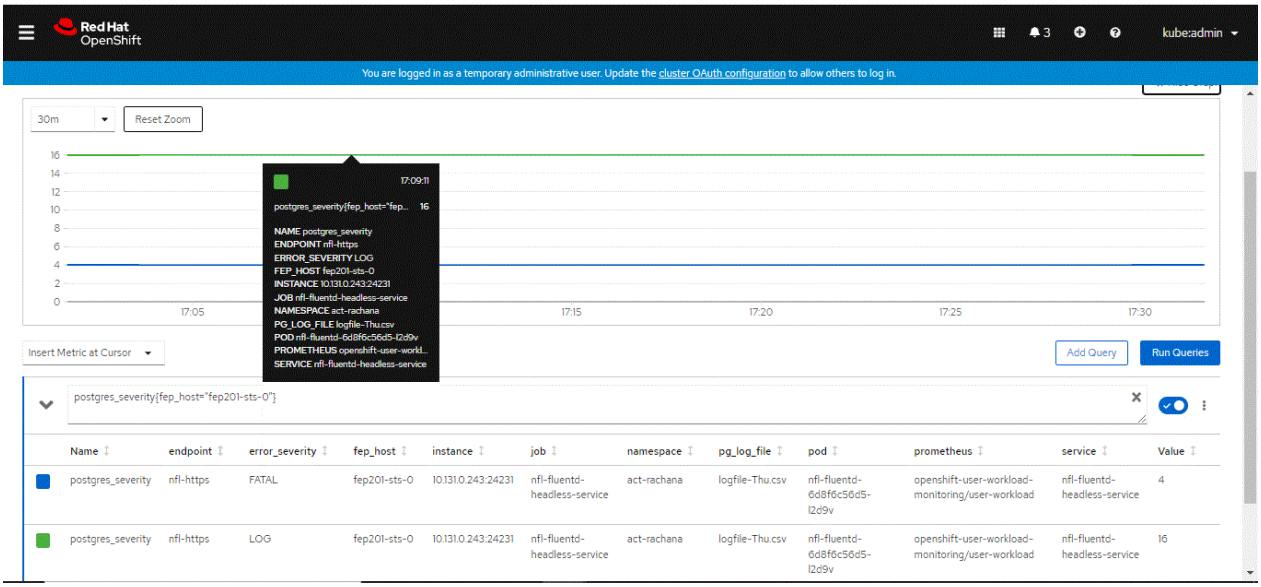


4.10.3.2 Log severity based Alarms/Metrics

FEPLogging feature is used for raising alarm/alert based on postgres severity counts as well. While user creates FEPLogging CR, Operator will forward real time counts of various postgres seviriry metrics to Openshift managed Prometheus. Openshift managed Alertmanager can access this metrics counters and user can use them to create alerts/alarms. There are 4 default alert rules already created as part of FEPLogging implementation as listed below:

- FEPLogErrorMessage
- FEPLogFatalMessage
- FEPLogPanicMessage
- FEPLogWarningMessage

Prometheus will scrape postgres_severity counter at every 30s as default scrape interval is 30s. User can modify this scrape interval from FEPLogging CR. After each scrape interval, if any change/increment found in postgres_severity counter then alert rule will be fired. User can check counts of postgres_severity metrics anytime from Prometheus dashboard as well.



4.10.3.3 Forwarding auditlog to Elasticsearch

In order to forward auditlog to Elasticsearch, update the FEPCluster to enable creating auditlog.

Example)

```
spec:
  fep:
    fepChildCrVal:
      customPgAudit: |
        [output]
        logger = 'auditlog'
        log_directory = '/database/log/audit'
      customPgParams: |
        shared_preload_libraries='...,pgaudit'
        session_preload_libraries='...,pgaudit'
```

Information

Refer to "4.12 Automating Audit Log Operations" for information on enabling audit logs. You can also enable audit logging by configuring the spec.fep.pgAuditLog.enable parameter.

4.10.4 Limitations

- Only postgres_severity including ERROR, PANIC, FATAL and WARNING are monitored.
- External fluentd can not be used for log monitoring and log forwarding.
- External Elasticsearch is required for log forwarding.
- User must decide at deployment time whether secured connection between FEPCluster and FEPLogging is required or not. After deployment, one can switch connection from insecure to secure but can not switch from secure to insecure connection.
- User must configure FEPLogging CR first then only FEPCluster can forward logs to particular FEPLogging otherwise Logging feature will not work.
- User must set log_destination in FEPCluster CR.

4.11 Configuring pgBadger

This section describes how to configure pgBadger. FEP cluster provides a feature to create pgbadger report on defined schedule and upload the report to a web server outside.

4.11.1 FEP Custom Resources - spec.fep.pgBadger

Custom Resource spec	Change Effect
pgBadger.schedules.create	The 'create' schedule to create report and upload it to endpoint
pgBadger.schedules.cleanup	The 'cleanup' schedule to delete the report left in container
pgBadger.options.incremental	Default: false; When set to True: create incremental report in pgbadger
pgBadger.endpoint.authentication	a secret to contain authentication info to access endpoint support basic auth only
pgBadger.endpoint.customCertificateName	Client certitificate reference in customCertificate CR
pgBadger.endpoint.fileUploadParameter	The file upload parameter defined by the web server Default: 'file'
pgBadger.endpoint.insecure	equivalent to curl -insecure option, default to false
pgBadger.endpoint.url	Web server url to upload the report file

4.11.2 Define pgBadger Schedules

The schedules are used to create and run a job periodically, written in Cron format.

If the schedule format is invalid, the cronjob will not be created, so no pgBadger report will be created and uploaded.

Example)

```
pgBadger:
  schedules:
    cleanup: '10 * * * *'
    create: '50 * * * *'
```

4.11.3 Define pgBadger Options

When the incremental option is set to false, pgBadger will create normal html report and upload the html file to the web server.

When the incremental option is set to true, pgBadger will create incremental report and upload a zip file to the web server.

Example)

```
pgBadger:
  options:
    incremental: true
```

4.11.4 Define Endpoint for Uploading Report

Web server url

Both http and https are supported.

Example)

```
pgBadger:
  endpoint:
    url: 'https://webserver-svc:4443/cgi-bin/upload.php'
```

Web Server authentication

Only basic auth is supported

To configure web server authentication:

Create a base64 encoded text from username:password

Example)

```
$ echo -ne "myuser:mypass" | base64

amFzb253Omphc29udw==
```

Wrap the output with base64 for creating a secret

Example)

```
$ echo -ne "amFzb253Omphc29udw==" | base64

YW1GemIyNTNPbXBoYzI5dWR3PT0=
```

Ceate a secret by using the wrapped text. The key must be 'basic_auth'.

Example)

```
kind: Secret
apiVersion: v1
metadata:
  name: pgbadger-endpoint-auth
  namespace: fep-container-ct
data:
  basic_auth: YW1GemIyNTNPbXBoYzI5dWR3PT0=
type: Opaque
```

Add the secret name in the endpoint definition.

Example)

```
    pgBadger:
      endpoint:
        authentication: pgbadger-endpoint-auth
```

Web Server certificates

When certificate files are required by the web server, FEP cluster provides customCertificate CR to mount the certificates files in container.

To use certificates for web server.

Create a secret based on the cert and key files.

Example)

```
oc create secret tls  webserver-cert --cert=webserver.pem --key=webserver.key
```

The webserver.pem and webserver.key are certificate files for accessing web server

Create a configmap based on the CA cert.

Example)

```
oc create configmap webserver-cacert --from-file=ca.crt=webca.pem
```

The webca.pem is the CA certificate file for accessing web server.

Define custom certificates in FEPCluster CR.

Example)

```
spec:
  fepChildCrVal:
    customCertificates:
      - userName: pgbadger-custom
       certificateName: webserver-cert
       caName: webserver-cacert
```

The userName is a reference in the pgBadger enpoint.

The certificateName is the secret created above.

The caName is the configmap created above.

Refer the custom certificate name in pgbadger endpoint.

Example)

```
    pgBadger:
      endpoint:
        customCertificateName: pgbadger-custom
```

Insecure access to web server

The pgbadger CR provides an option to the web server endpoint when secure connection is not required:

Example)

```
    pgBadger:
      endpoint:
        insecure: true
```

File upload parameter

This parameter specify the request parameter for uploading a file to a web server. The value of this parameter is depended on the web server implementation.

Example)

```
    pgBadger:
      endpoint:
        fileUploadParameter: uploadfile
```

curl command and parameters

FEP cluster uses curl command to upload the generated report to a web server endpoint. The CR in enpoint section will be converted to curl command parameters. The following table shows the mapping:

curl command parameter	User configuration
[URL]	Endpoint url
--cert	webserver.pem included in the secret referred in customCertificateName
--key	webserver.key included in the secret referred in customCertificateName

curl command parameter	User configuration
--cacert	webca.pem included in the configmap referred in customCertificateName
--form "uploadfile=@/path/to/report"	Endpoint fileUploadParameter
--header "Authorization: Basic passxxx"	Endpoint authentication configmap
--insecure	When endpoint.insecure is set to true

4.11.5 Uploaded File on Web Server

The FEP cluster uploads the pgbadger report according to the incremental mode:

incremental mode	Uploaded file name	Example
True	[fep cluster name]-sts-[pod index].zip	pgbadger-test3-sts-0.zip pgbadger-test3-sts-1.zip
False	[fep cluster name]-sts-[pod index].html	pgbadger-test3-sts-0.html pgbadger-test3-sts-1.html

The zip file contains a folder of pgbadger incremental report.

Example)

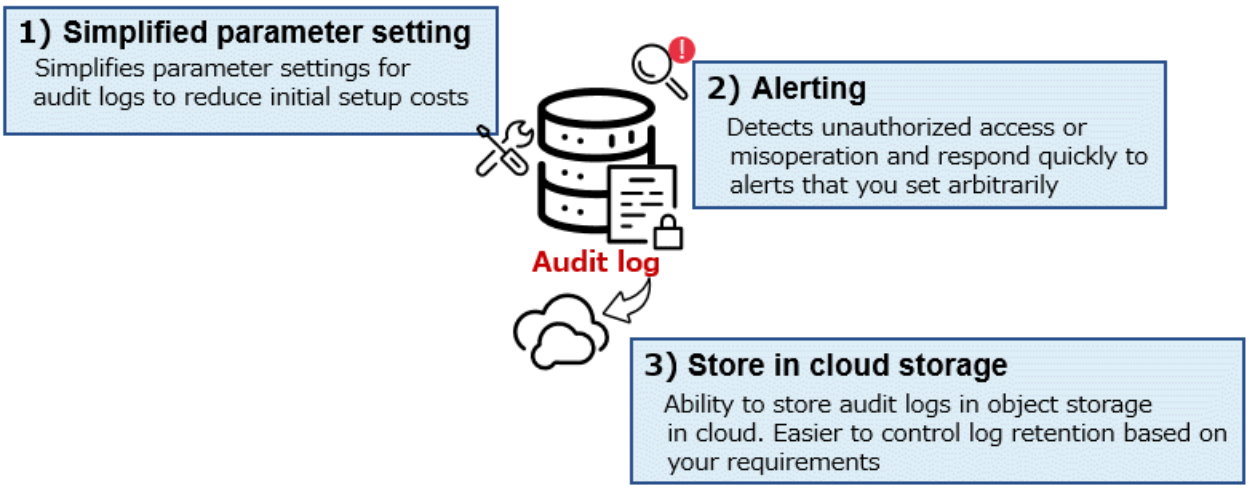
```
\database
  \log
    \pgbadger-report
      \[years]
        \[months]
        \[weeks]
```

Note

- The web server is NOT included in the FEP cluster solution.
- The web server is responsible to the uploaded files according to the customer's business logic.

4.12 Automating Audit Log Operations

Simplifies the operation of your audit logs to implement operations that meet security requirements such as audits.



4.12.1 Simplifies Parameter Setting

Simplify audit log parameter settings and reduce initial setup costs. The only setting required is enabling the enable parameter. The pgaduit module is loaded when the FEP server starts and audit logs are stored in the logs directory.

By customizing the audit log configuration file as necessary, it is also possible to make settings according to operational requirements.

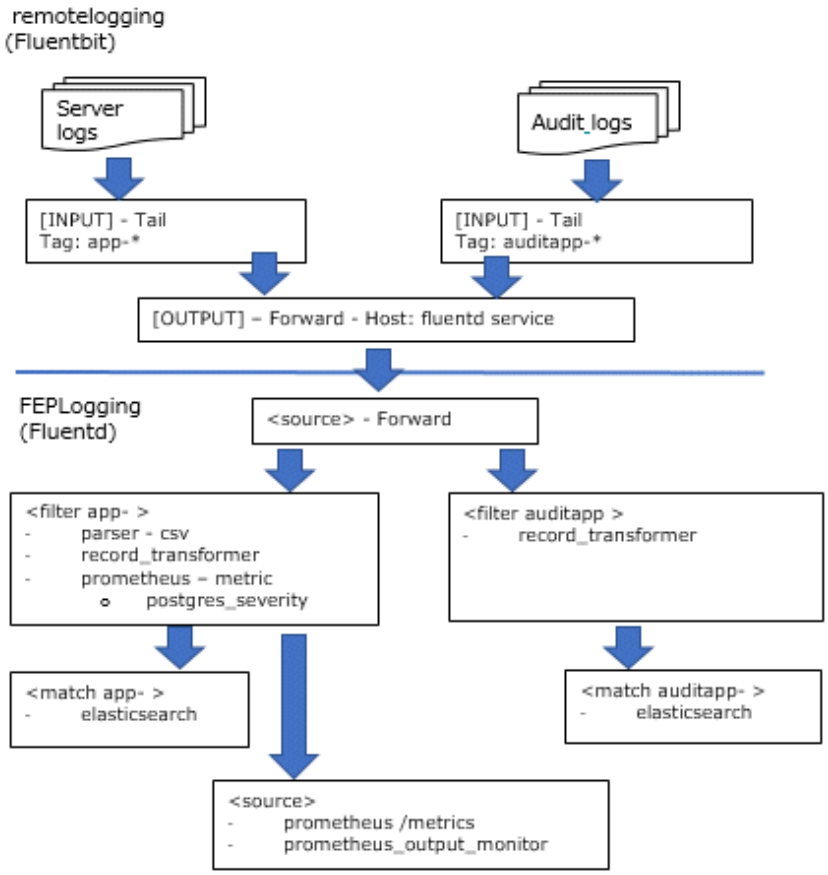
```
spec:
  fep:
    pgAuditLog:
      enable: true
```

4.12.2 Alerting

According to preset alert conditions, unauthorized access and erroneous operations can be detected at an early stage, enabling rapid response.

Audit logs are sent to Fluentd using the remotelogging function, and audit logs are monitored by setting alerts in Prometheus according to sqlstate conditions.

The alert is triggered when the 1 minute average of sqlstate(28P01) (invalid password) exceeds 50.



4.12.3 Store in Cloud Storage

Long-term retention of audit logs may be required in accordance with system or industry standard security policy requirements. However, long-term storage of logs requires the continuation of complicated operations such as disk management and rotation management.

Therefore, with this function, by saving audit logs to cloud object storage, you can easily control the saving of logs based on your requirements.

```
spec:
  fep:
    pgAuditLog:
      enable: true
      endpoint:
        protocol: s3
        url: s3://pgaudit/cluster1
        authentication: s3-secret
      schedules:
        upload: '30 * * * *'
```

4.13 Transparent Data Encryption Using a Key Management System

Describes how to configure transparent data encryption using a key management system.

Transparent data encryption using a key management system can only be configured when the FEPCluster is first created. Users cannot configure an existing FEPCluster for transparent data encryption using a key management system.

4.13.1 Registration of Authentication Information

4.13.1.1 When Using a KMIP Server

Save the certificate used for TLS communication between KMIP server in Secret or ConfigMap.

The Secret or ConfigMap you created gives the FEPCluster custom resource a resource name and mounts it in the FEP container.

Create a Secret to store the client certificate and private key for connecting to KMIP server.

Also, optionally create a ConfigMap to store the root certificate.

An example of registering credentials using the credentials file below is explained.

```
kmip.pem   # Client certificate for connecting to KMIP server
kmip.key   # Private key
myca.pem   # Root certificate
```

Create a Secret to store the client certificate and private key.

Specify tls.crt and tls.key as file names when mounting the client certificate and private key, respectively.

```
$ oc create secret generic kmip-cert --from-file=tls.crt=kmip.pem --from-file=tls.key=kmip.key -n
kmip-demo
```

Optionally create a ConfigMap to store your root certificates.

Specify ca.crt as the file name to be mounted.

```
$ oc create configmap kmip-cacert --from-file=ca.crt=myca.pem -n my-namespace
```

4.13.1.2 When Using AWS Key Management Service

Save credentials and other settings required to connect to AWS key management services in Secrets and ConfigMaps.

Prepare two files, credentials and config, which describe credentials and other settings according to the format specified by the AWS client interface. Specifying access_key_id and secret_access_key in the credentials file is mandatory.

An example of registering authentication information using the following configuration file is explained.

```
credentials   # credentials file
config         # config file
```

Create a ConfigMap to store config files. Specify config for the key name. The name of the ConfigMap is arbitrary (here aws-kms-config).

```
$ oc create configmap aws-kms-config --from-file=config=config -n my-namespace
```

Create a secret to save the credentials file. Specify credentials for the key name. The name of the Secret is arbitrary (here aws-kms-credentials).

```
$ oc create secret generic aws-kms-credentials --from-file=credentials=credentials -n my-namespace
```

See

Refer to below for AWS client interface configuration files.

https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-files.html

4.13.1.3 When using Azure Key Management Service

Save the credentials required to connect to Azure's key management service in Secret.

The available authentication methods are either authentication using passwords or authentication using client certificates.

For password-based authentication, create a YAML format file that defines a secret like the one below. The secret name is arbitrary (here azure-key-vault-passphrase). data.clientsecret contains a base64-encoded password.

```
kind: Secret
apiVersion: v1
metadata:
  name: azure-key-vault-passphrase
  namespace: my-namespace
data:
  clientsecret: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX=
type: Opaque
```

Create a secret based on the created YAML file. Here we are using a YAML file named azure-client-secret.yaml.

```
$ kubectl apply -f azure-client-secret.yaml -n my-namespace
```

For authentication using a client certificate, store the client certificate file and private key in Secret.

Here is an example of creating a Secret using the certificate file below.

```
azuremycert.pem    # PEM file containing client certificate and private key
```

Create a secret to store the client certificate. Specify azure-key-vault.crt for the key name. The secret name is arbitrary (here azure-key-vault-secret).

```
$ oc create secret generic azure-key-vault-secret --from-file=azure-key-vault.crt=azuremycert.pem -n
my-namespace
```

4.13.2 Configuring FEPCluster Custom Resources

To enable TDE using a key management system, you need to set "spec.fepChildCrVal.customPgParams" and "spec.fepChildCrVal.sysTde".

4.13.2.1 Define spec.fepChildCrVal.customPgParams

The fepChildCrVal.customPgParams section must define the following parameters:

shared_preload_libraries

Add the 'tde_kms' library to the list of libraries in shared_preload_libraries.

Example)

```
spec:
  fep:
   …
   fepChildCrVal:
      …
     customPgParams:
       shared_preload_libraries='pgx_datamasking,pg_prewarm,pg_stat_statements,tde_kms'
```

Do not remove 'tde_kms' library from 'shared_preload_libraries' list after cluster creation.

4.13.2.2 Define spec.fepChildCrVal.sysTde

Add a sysTde section under spec.fepChildCrVal to define the parameters required to connect to your key management system. Under sysTde there are two parameters defined:

- tdeType
- tdek

Define spec.fepChildCrVal.sysTde.tdeType

sysTde itself is an optional parameter (if sysTde is not defined, use a file-based keystore). However, if sysTde is defined by the user, sysTde.tdeType must also be defined.

If configuring TDE with a key management system, set sysTde.tdeType to "tdek".

Example)

```
    sysTde:
      tdeType: tdek
```

Define spec.fepChildCrVal.sysTde.tdek.kmsDefinition

If you set sysTde.tdeType to "tdek", you must also define sysTde.tdek.

Define the connection information of the key management system in sysTde.tdek.kmsDefinition. Based on the information defined here, the operator creates the key management system connection information file used by Fujitsu Enterprise Postgres.

Information for multiple key management systems can be defined in kmsDefinition. For type, specify the type of key management system (either kmip, awskms, or azurekeyvault).

Example)

```
    sysTde:
      tdeType: tdek
      tdek:
        targetKmsName: kms_conninfo1
        kmsDefinition:
          - name: kms_conninfo1
            type: kmip
...
```

Refer to the Reference for details of each parameter.

Specify the name of the Secret or ConfigMap created in "4.13.1 Registration of Authentication Information" in the corresponding parameter under kmsDefinition. If type is awskms, profile specifies the name of the profile to use from the profile in the AWS client interface configuration file.

Example)

```
  spec:
    fep:
     …
     fepChildCrVal:
        …
       sysTde:
        tdeType: tdek
        tdek:
          targetKmsName: kms_conninfo1
          targetKeyId: xxxyyyzzz
          kmsDefinition:
            -  name: kms_conninfo1
               type: kmip
               address: xxx.xxx.xxx.xxx
               port: 100
               authMethod: cert
               sslpassphrase: ssl-password
               cert:
                 certificateName: kmip-cert
```

```
                caName: kmip-cacert
                sslcrlName: kmip-crl
```

Define spec.fepChildCrVal.sysTde.tdek.targetKeyId, spec.fepChildCrVal.sysTde.tdek.targetKmsName

Specify one of the key management system names defined in kmsDefinition in sysTde.tdek.targetKmsName as the name of the key management system to use as the keystore. sysTde.tdek.targetKeyId specifies the key ID of the encryption key within that key management system to use as the master encryption key.

4.14 Disaster Recovery in Hot Standby Configuration

By implementing disaster recovery in a hot standby configuration, business systems can be restored more quickly in the event of a disaster. This function has the following two methods.

Continuous recovery method

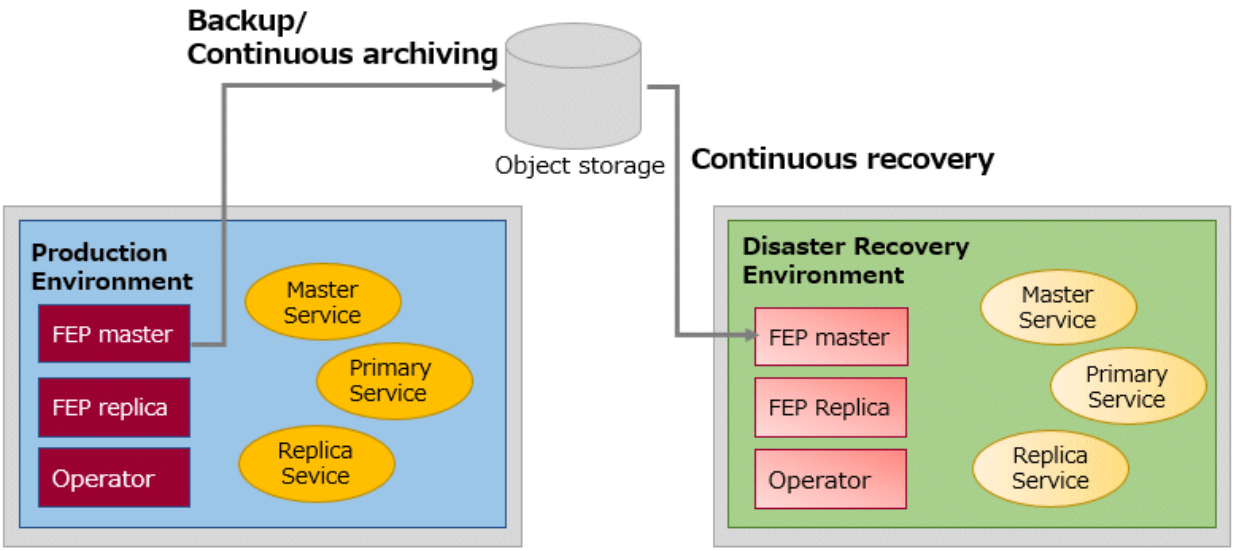
Create a container environment for production and another one for disaster recovery. Store the production environment data in object storage, and continuously restore to the disaster recovery environment. This method enables quick recovery compared to the backup/restore method, but RPO (Recovery Point Objective) increases depending on the timing of regular backups.

Streaming replication method

Similarly to the continuous recovery method, create a container environment for production and another one for disaster recovery. Use streaming replication methods to synchronize data to the disaster recovery environment. This method enables faster recovery compared to the backup/restore method, lower RPO compared to the continuous recovery method, and real-time data synchronization. However, because network settings for the streaming replication method are required, the management cost is high, and there is a slight impact on the performance of the database in the production environment.

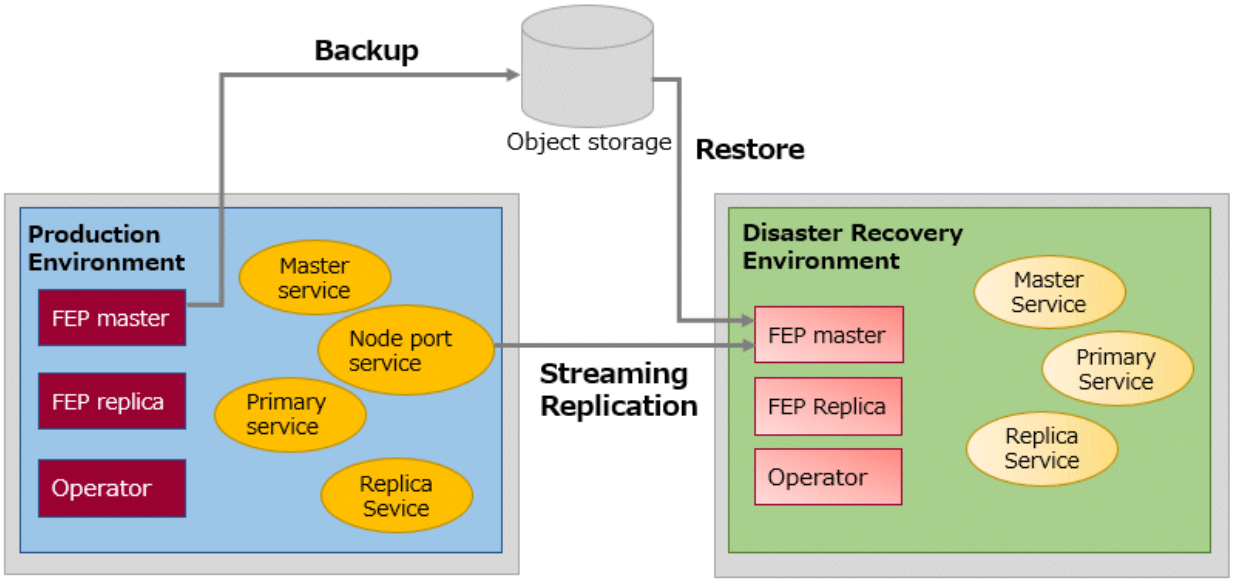
4.14.1 Continuous Recovery Method

The continuous recovery method uses object storage to synchronize production and disaster recovery environments. Specify object storage that is located in a region that you consider safe for the range of possible disasters.



4.14.2 Streaming Replication Method

The streaming replication method achieves direct data synchronization between the database in the production environment and the database in the disaster recovery environment.



4.14.3 Defining a Hot Standby Configuration

This section describes the deployment procedures for the continuous recovery method and the streaming replication method of the hot standby configuration.

4.14.3.1 Defining a Continuous Recovery Method

Custom resource definitions for FEPCluster in the production environment do not have parameters that are only used in hot standby configurations. When using the continuous recovery method, define the FEPCluster custom resource in the disaster recovery environment as follows.

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
 …
spec:
  fep:
    standby:
      enable: true
      method: archive-recovery
      pgBackrestConf: |
        [global]
        log-path=/database/log/backup
        repo1-type=azure
        repo1-path=< Backup path of primary/ cluster from which data is to be restored>
        repo1-azure-account=<my storage account>
        repo1-azure-container=fepbackups
        repo1-azure-key=<my storage account key >
…
```

4.14.3.2 Defining a Streaming Replication Method

When using the streaming replication method, define as follows.

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
...
spec:
  fep:
```

```
    standby:
      enable: true
      method: streaming
      streaming:
        host: <LoadBalancer IP>
        port: 27500
      pgBackrestConf: |
        [global]
        log-path=/database/log/backup
        repo1-type=azure
        repo1-path=< Backup path of primary/ cluster from which data is to be restored>
        repo1-azure-account=<my storage account>
        repo1-azure-container=fepbackups
        repo1-azure-key=<my storage account key >
...
```

For streaming replication, FEPClusterCR is defined as above, but a separate LoadBalancer must be deployed.

```
kind: Service
apiVersion: v1
metadata:
  name: my-fep-internal-svc
  namespace: sample-namespace
  annotations:
    service.beta.kubernetes.io/azure-load-balancer-internal: 'true'
spec:
  ports:
    - protocol: TCP
      port: 27500
  type: LoadBalancer
  selector:
    app: <my-fep-cluster>-sts
    feprole: master
```

4.14.3.3 Defining FEPCluster Custom Resources

The parameters of FEPClusterCR in the disaster recovery environment that are required to realize a hot standby configuration are shown below. For parameter details, refer to "FEPCluster Parameters" in the Reference.

- spec.fep.standby.enable
- spec.fep.standby.method
- spec.fep.standby.pgBackrestConf
- spec.fep.standby.streaming.host
- spec.fep.standby.streaming.port

4.15 Enabling Client Authentication with scram-sha-256 Authentication

This section describes step-by-step instructions on how to enable client authentication using scram-sha-256 authentication.

4.15.1 Enabling Client Authentication Using scram-sha-256 Authentication in the FEP Server Container

To enable scram-sha-256 authentication on a newly built FEP server container, deploy a FEPCluster custom resource with the following settings.

4.15.1.1 Define spec.fepChildCrVal.customPgParams

Specify scram-sha-256 for password_encryption.

```
password_encryption = 'scram-sha-256'
```

You must restart the database in the FEP server container for the configuration to take effect.

4.15.1.2 Define spec.fepChildCrVal.customPgHba

Add an entry to allow scram-sha-256 authentication.

To allow scram-sha-256 authentication, specify scram-sha-256 in the METHOD.

You must restart the database in the FEP server container for the configuration to take effect.

4.15.2 Enabling Client Authentication Using scram-sha-256 Authentication in the FEPpgpool2 Container

This section describes how to enable scram-sha-256 authentication on a newly built FEPpgpool2 container.

4.15.2.1 Creating the Resources Required to Enable scram-sha-256 Authentication

Create the resources required to enable scram-sha-256 authentication.

1. Create secret for encryption key

   Create an encryption key secret (pgpoolkeySecret) containing the password to use for scram-sha-256 authentication.
   The key is pgpoolkey and the value is a base64 encrypted version of the password to be used for encryption.

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: scrampgpoolkey-secret
   type: Opaque
   data:
     pgpoolkey: cGdwb29sa2V5cGFzc3dvcmQ=
   ```

   After creating the secret, write the name of the secret you created in the FEPPgpool2 custom resource. This mounts the created secret in the FEPpgpool2 container as a file in the volume and passes the password used for encryption to the FEPpgpool2 container.

   If you use scram-sha-256 authentication and do not create a secret, the operator automatically creates the following secret based on the information in the FEPPgpool2 custom resource.

   ```
   apiVersion: v1
   kind: Secret
   metadata:
     name: "{{spec.name}}-feppgpool2-pgpoolkey"
   type: Opaque
   data:
     pgpoolkey:
   K1kxOVZxKzRrdWluT3A2UHNQMzcwcUJuOUZ2UUoxUklNMms2cktIY1NkekFOemZBYkhjZDFadG5VR3ZtTVR6Uw==
   ```

   Value is set to the value of a password randomly generated by the operator and encrypted with base64. The secret name is "{{spec.name}} -feppgpool2-pgpoolkey".

   Information

   scram-sha-256 encryption uses an encryption key. This encryption key is used by the FEPpgpool2 container to decrypt the AES-encrypted password if it was stored in the pool_passwd file. Therefore, the encryption and decryption keys must be the same.

2. Create a secret for database user information

   The user/password used for client authentication in the FEPpgpool2 container must be the same as the database user used in the database in the FEP server container.

   Create a database user information secret (userinfoSecret) in the FEPpgpool2 container to inform the database user username and password information in the FEP server container.
   The user name and password should describe the secret in the following format.

```
apiVersion: v1
kind: Secret
metadata:
  name: scramuserinfo-secret #Specify any name
type: Opaque
data:
  user1: dXNlcjFwYXNzd2Q=
  user2: dXNlcjJwYXNzd2Q=
```

   For database users that Operator creates automatically, you do not need to create a secret because Operator retrieves the database username and password information from the FEPCluster custom resource and the FEP server container.

   If you use scram-sha-256 authentication and do not create a secret, the operator automatically creates the following secret:

```
apiVersion: v1
kind: Secret
metadata:
  name: "{{spec.name}}-feppgpool2-userinfo"
type: Opaque
data:
```

   The secret name is "{{spec.name}}-feppgpool2-userinfo".

4.15.2.2 Editing FEPPgpool2 Custom Resources

To enable scram-sha-256 authentication, edit and deploy a FEPPgpool2 custom resource with the following settings:

1. Set secret information

   Specify scram for spec.clientAuthMethod in the FEPPgppol2 custom resource.

   In addition, in the secret for the encryption key (spec.scram.pgpoolkeySecret), specify the secret name that you created in step 1 of "4.15.2.1 Creating the Resources Required to Enable scram-sha-256 Authentication" Similarly, in the database user information secret (spec.scram.userinfoSecret), specify the secret name that you created in step 2 of "4.15.2.1 Creating the Resources Required to Enable scram-sha-256 Authentication".

```
spec:
  clientAuthMethod: scram
  scram:
    pgpoolkeySecret: scrampgpoolkey-secret
    userinfoSecret: scramuserinfo-secret
```

   This mounts the created secret in the FEPpgpool2 container as a file in the volume and notifies the FEPpgpool2 container of the username and password information.

2. Edit spec.customhba

   Edit the spec.customhba field in the FEPPgpool2 custom resource to add an entry for scram-sha-256 authentication.

3. Edit pgpool.conf

   You must edit the spec.customparams field in the FEPPgpool2 custom resource to edit the parameters related to the authentication settings.

```
enable_pool_hba=true
```

4.15.3 Enabling Client Authentication Using scram-sha-256 Authentication on Existing FEP Server and FEPpgpool2 Containers

This section describes how to enable client authentication using scram-sha-256 authentication on existing FEP server and FEPpgpool2 containers

For information about FEP server containers, refer to "4.15.1 Enabling Client Authentication Using scram-sha-256 Authentication in the FEP Server Container". At this time, if the database user created by the user already has a password encrypted with md5, change the password to one encrypted with scram-sha-256. Therefore, reset the password.

For information about the FEPpgpool2 container, refer to "4.15.2 Enabling Client Authentication Using scram-sha-256 Authentication in the FEPpgpool2 Container". If an existing FEPpgpool2 custom resource is deployed, delete the FEPpgpool2 custom resource. You do not need to delete the FEPpgpool2 custom resource if you want to change the contents of the secret specified in the FEPpgpool2 custom resource or the secret created automatically by the operator.

4.16 Backing Up Statistics

In case the query plan becomes unstable, you can periodically backup the statistics using pg_dbms_stats. Set spec.fep.backupStats.enable to true and set the backup to spec.fep.backupStats.scheduleN (N is an integer) in the FEPClsuter custom resource. Note that spec.fep.backupStats.enable defaults to true.

The following are the configurable items.

- Schedule
- Whether to execute the schedule
- What to back up
- Retention
- Comment

The example below is a setting to take a backup of mydb every day at 20:00, and at the same time delete the backup from 5 days ago.

Example) Definition example of FEPCluster custom resource

```
spec:
  fep:
    backupStats:
      enable: true
      schedule1:
        enable: true
        backupSchedule: "0 20 * * *"
        targetDb: mydb
        retention: 5
        comment: "mydbBackup"
```

Additionally, if there is no backup definition in spec.fep.backupStats.schedule1 when building FEPCluster for the first time, the backup definition will be defined in spec.fep.backupStats.schedule1 of the FEPCluster custom resource with the following settings by default.

Schedule: 23:30 every day

Target: All databases

Retention: Statistics backed up in that database older than 7 days will be deleted.

Comment: "FepDefaultBackup"

```
spec:
  fep:
    backupStats:
      enable: true
      schedule1:
        enable: true
        backupSchedule: "30 23 * * *"
```

```
      retention: 7
      comment: "FepDefaultBackup"
```

If you do not want to perform scheduled backups, set spec.fep.backupStats.scheduleN.enable to false. Also, if you do not want to perform all backups, set spec.fep.backupStats.enable to false.

Chapter 5 Post-Deployment Operations

This chapter describes the operation after deploying the container.

5.1 How to Connect to a FEP Cluster

When connecting from within the same project of the OpenShift system

Service resources are used to connect to FEPCluster and FEPPgpool2 from within the same project.

A service resource provides a single endpoint for communicating with containers.

Service resources are created with the following naming conventions.

FEPCluster service

- <FEPCluster name>-primary-svc
- <FEPCluster name>-replica-svc
- <FEPCluster name>-headless-svc

FEPPGPool2 service

- <FEPPgpool2 name>-feppgpool2-svc

Example of checking service resources of FEPCluster container and FEPpgpool2 container

```
$ oc get all
```

Check where the resource type is Service (Begin with "svc/").

You can also check with the oc get svc command. The following is an example.

```
$ oc get svc
NAME                              TYPE         CLUSTER-IP      EXTERNAL-IP  PORT(S)            AGE
<FEPCluster name>-headless-svc   ClusterIP   None              <none>   27500/TCP,25001/TCP   24h
<FEPCluster name>-primary-svc    ClusterIP   xxx.xxx.xxx.xxx  <none>   27500/TCP,25001/TCP    24h
<FEPCluster name>-replica-svc    ClusterIP   yyy.yyy.yyy.yyy  <none>   27500/TCP,25001/TCP    24h
<FEPPgpool2 name>-feppgpool2-svc NodePort    zzz.zzz.zzz.zzz  <none>   9999:31707/TCP,9998:31906/TCP
24h
```

Example of accessing FEPpgpool2 container

```
$ psql -h <FEPPgpool2 name>-feppgpool2-svc  -p 9999 -c "select version();"
```

When connecting from outside the OpenShift system

Automatically creating a service with ClusterIP to connect to the deployed container. You can connect to FEP or FEP pgpool2 services from the OpenShift system's internal network. To access from outside the OpenShift system, you need to know the address of the OpenShift node.

For example, "Access the FEP pgpool2 container from an application server that is running outside the OpenShift system but is part of the Internal network".

An example of how to check the node IP in OpenShift.

```
$ oc get nodes
NAME                                    STATUS   ROLES    AGE    VERSION
openshiftcluster1-cmfv8-master-0        Ready    master   370d   v1.19.0+4c3480d
openshiftcluster1-cmfv8-master-1        Ready    master   370d   v1.19.0+4c3480d
openshiftcluster1-cmfv8-master-2        Ready    master   370d   v1.19.0+4c3480d
$ oc describe nodes openshiftcluster1-cmfv8-master-0 | grep IP
 InternalIP: 10.0.2.8
```

An example of verifying the service resource for the FEP pgpool2 container.

```
$ oc get all
```

Check where the resource type is Service (Begin with "svc/").

You can also see this with the oc get svc command. The following is an example.

```
$ oc get svc
NAME                       TYPE     CLUSTER-IP    EXTERNAL-IP PORT(S)                          AGE
svc-feppgpool2-feppgpool2 NodePort 172.30.248.12 <none>      9999: 30537/TCP, 9998: 30489/TCP 2m5s
```

This is an example of accessing the FEP pgpool2 container.

```
$psql -h 10.0.2.8 -p 30537 -c "show pool_nodes"
```

5.2 Configuration Change

This section describes changes to the FEPCluster configuration.

List FEPCluster

Equivalent Kubernetes command: kubectl get FEPClusters (-A)

This operation will list all FEPClusters in a namespace, or if the -A option is specified, will list all FEPClusters in all namespace.

Default output format:

Field	Value	Details
NAME	.metadata.name	Name of Cluster
AGE	Elapsed time	Indicates the amount of time that has elapsed since the cluster was created

Example)

```
# kubectl get fepclusters -A

NAMESPACE       NAME            AGE
namespace1      ns1fep1         21h
namespace2      ns2fep2         22h
```

Update FEPCluster

Equivalent Kubernetes command: kubectl apply -f <new_spec>

Operations that can be performed here.

Custom Resource spec	Change effect
.spec.fep.instances: *n*	Increase the number of nodes in the cluster to *n*.
.spec.fep.image.image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.1'	Minor upgrade of FEP image to ubi9-17-1.1.
spec.fepChildCrVal.backup.image.image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-17-backup:ubi9-17-1.1'	Minor upgrade of Backup image to ubi9-17-1.1.

This will impact behaviour for values in fep section only.
All parameters can be updated from the FEPCluster custom resource.

Delete FEPCluster

Equivalent Kubernetes command: kubectl delete FEPCluster <cluster_name>

This operation will remove the FEPCluster by the cluster_name and all Child CRs (FEPVolume, FEPConfig, FEPCert & FEPUser) & resources associated with it.

Deleting a FEPCluster will delete all PV associated with the cluster, including backup and archived WAL volumes (except when using pre-made PV or AWS S3). This is an unrecoverable action.

5.3 FEPCluster Resource Change

5.3.1 Changing CPU and Memory Allocation Resources

Describes how to change the CPU and memory resources assigned to a pod created by a FEPCluster.

This allows you to scale the pod vertically through custom resources.

To modify CPU and memory resources, modify the spec.fep.mcSpec section(*1) of the FEPCluster custom resource and apply your changes.

When the changes are applied, restart the replica server with the new resource settings. If there are multiple replica servers, restart them one at a time. When all replica servers are restarted, one of them is promoted to the new master server due to a switchover. Then restart the container image on the original master server. This allows you to change resource settings for all servers with minimal disruption.

*1) Modifying this section scales up the FEP server container. For information about other container resource sections, refer to "FEPCluster Parameters" in the Reference.

5.3.2 Resizing PVCs

Describes how to resize a PVC assigned to a pod created by a FEPCluster.

This allows you to increase the size of the volume allocated to the pod through custom resources.

To change the PVC size, modify the size of each volume in the spec.fepChildCrVal.storage section of the FEPCluster custom resource and apply the change. These changes apply to all PVCs assigned to the pod created by the FEPCluster.

- PVC resizing is extensible only.
- You can resize a PVC only if the StorageClass supports dynamic resizing.
- If the StorageClass does not support resizing PVCs, use the FEPRestore custom resource to create a new FEPCluster to resize the PVC. For more information, refer to "FEPRestore Custom Resource Parameters" in the Reference.

5.4 FEPPGPool2 Configuration Change

This section describes changes to the FEPPGPool2 configuration.

List FEPPGPool2

Equivalent Kubernetes command: kubectl get FEPPGPool2 (-A)

This operation will list all FEPPGPool2 in a namespace, or if the -A option is specified, will list all FEPPGPool2 in all namespace.

Default output format:

Field	Value	Details
Name	.metadata.name	Name of pgpool2

Example)

```
# kubectl get feppgpool2 -A

NAMESPACE       NAME
namespace1      fep1-pgpool2
namespace2      fep2-pgpool2
```

Delete FEPPGPool2

Equivalent Kubernetes command: kubectl delete FEPPGPool2 <pgpool2_name>

This operation will remove the FEPPGPool2 by the pgpool2_name.

Update FEPPGPool2

Equivalent Kubernetes command: kubectl apply -f <new_spec>

Refer to "FEPPgpool2 Custom Resource Parameters" in the Reference and specify the parameters to be updated. Only the following parameters can be specified.

Custom Resource spec	Change Effect
.spec.count: n	Increase the number of nodes in the cluster to n.
.spec.serviceport	Change the TCP port for connecting to the Pgpool-II.
.spec.statusport	Change the TCP port for connecting to the PCP process.
.spec.limits.cpu	Change limits of cpus.
.spec.limits.memory	Change limits of memory.
.spec.requests.cpu	Change requests of cpus.
.spec.requests.memory	Change requests of memory.
.spec.fepclustername	Change fepcluster to connect.
.spec.customhba	Change pool_hba.conf file.
.spec.customparams	Change pgpool2 parameters
.spec.custompcp	Change pcp.conf file.
.spec.customsslkey	Change key content
.spec.customsslcert	Change the contents of the public x 509 certificate.
.spec.customsslcacert	Change the contents of the CA root certificate in PEM format.

Some of the customparams parameters, customhba and custompcp, require a restart of pgpool2.

Equivalent Kubernetes command: Kubectl apply -f <new_spec>

"pgpool2_restart" action type expects users to specify the name of the pgpool2 that they want to restart from.

Specify the metadata.Name of the FEPPGPool2 CR in the targetPgpool2Name section of the FEPAction CR, as below:

```
spec:
    targetPgpool2Name: fep1-pgpool2
```

```
  fepAction:
      type: pgpool2_restart
```

Note

When updating FEPPGPool2, the Pod of FEPPGPool2 is restarted. If configured with more than one FEPPGpool2, they are rebooted sequentially. The application should be designed to reconnect the connection because the connection being connected is broken.

Update each resource used for client authentication using scram-sha-256 authentication

Set password for new database user

When you add a new user to the database, you must notify the FEPpgpool2 container of the user and password.
In this case, in addition to the information of the existing user, add the information of the new user to the secret as follows.

```
apiVersion: v1
kind: Secret
metadata:
  name: test-secret
type: Opaque
data:
  user_name_1:cGFzc3dvcmRfMQ==
  user_name_2:cGFzc3dvcmRfMg==
  user_name_new:cGFzc3dvcmRfbmV3
```

After updating the secret, the FEPpgpool2 container automatically updates the contents of the password file pool_passwd.
If you want to edit the operator-created secret to add new user information, delete the postgres key entry. If there are still postgres key entries, the new user information will not be reflected.

Update passwords for existing database users

The FEPCluster container may update the passwords of database users for reasons such as password expiration.
In this case, keep the information of the existing user, and update the user information secret for the updated password as follows.

```
apiVersion: v1
kind: Secret
metadata:
  name: test-secret
type: Opaque
data:
  user_name_1:cGFzc3dvcmRfMQ==
  user_name_2:cGFzc3dvcmRfMl9uZXc=
```

After updating the secret, the FEPpgpool2 container automatically updates the contents of the password file pool_passwd.

Update encryption key file pgpoolkey

If you want to update the encryption key file pgpoolkey, update the contents of the secret defined in the encryption key secret pgpoolkeySecret.

```
apiVersion: v1
kind: Secret
metadata:
  name: pgpoolkey-secret
type: Opaque
data:
  pgpoolkey:bmV3LXBncG9vbGtleXBhc3N3b3Jk
```

If you want to update the encryption key file pgpoolkey, you must use the updated pgpoolkey and encrypt it again with scram-sha-256.
You will also need to restart pgpool2 after updating the encryption key secret.

After updating the secret, the FEPpgpool2 container automatically updates the contents of the password file pool_passwd with the updated pgpoolkey.

5.5 Scheduling Backup from Operator

Operational status confirm

Information about the backup can be found by running the command in the FEP backup container, as shown in the example below.

```
$ oc exec pod/fepserver-XXXXX -c FEPbackup -- pgbackrest info
stanza: fepbackup
    status: ok
    cipher: none

    db (current)
        wal archive min/max (12-1): 000000010000000000000001/000000010000000000000005

        full backup: 20201125-025043F
            timestamp start/stop: 2020-11-25 02:50:43 / 2020-11-25 02:50:52
            wal start/stop: 000000010000000000000003 / 000000010000000000000003
            database size: 31.7MB, backup size: 31.7MB
            repository size: 3.9MB, repository backup size: 3.9MB

        incr backup: 20201125-025043F_20201125-025600I
            timestamp start/stop: 2020-11-25 02:56:00 / 2020-11-25 02:56:02
            wal start/stop: 000000010000000000000005 / 000000010000000000000005
            database size: 31.7MB, backup size: 24.3KB
            repository size: 3.9MB, repository backup size: 619B
            backup reference list: 20201125-025043F
```

Update FEPBackup

Equivalent Kubernetes command: kubectl apply -f <new_spec>

Refer to "FEPBackup Child Custom Resource Parameters" in the Reference and specify the parameters to be updated. Only the following parameters can be specified.

Custom Resource spec	Change Effect
spec.schedule.num	Change the Number of Registered Backup Schedules
spec.scheduleN.schedule	Change the scheduled backup time
spec.scheduleN.type	Change the scheduled backup type
spec.pgBackrestParams	Change pgBackRest parameters
spec.scheduleN.repo	If you specified more than one repository for spec.pgBackrestParams, select the repository in which to store the backup data. The default is 1.

- Changes made during the backup are reflected from the next backup.
- Changes to the backup schedule do not affect the application.
- If you perform any of the following update operations, be sure to obtain a backup after the update.
  - When the master encryption key is updated with pgx_set_master_key
  - When the encryption passphrase for transparent data encryption is updated (can be updated by the tdeppassphrase parameter of FEPCluster CR)

5.6 Configure MTLS Setting

5.6.1 Certification Rotation

All certificates are bounded by the time limit. At certain time, it needs to be renewed. We recommend to renew the certificate when it reaches 3/4 of its life cycle or as soon as possible if it is compromised. When a certificate is renewed, we need to rotate it inside the FEP server container. At the moment, FEP server container does not support automatic certificate rotation. Depending on which certificate has renewed, there are different procedures to handle that.

Patroni Certificate Rotation

When Patroni certificate is renewed, we have to re-deploy each and every Pod for FEP server container to pick up the new certificate. There is a down time on FEPCluster.

FEP Server Certificate Rotation

When FEP Server certificate is renewed, we can use FEPAction CR to trigger a reload of the database and FEP server will pick up the new certificate with no interruption to service.

Client certification Rotation

When any of the client certificate is renewed, FEP server container internally will use the new certificate next time it establishes a connection to FEP server. However, to avoid any unexpected interruption to service, it is recommended to re-deploy each and every Pod as soon as possible.

5.7 Monitoring

Monitoring is collecting historic data points that you then use to generate alerts (for any anomalies), to optimize databases and lastly to be proactive in case something goes wrong (for example, a failing database).

There are five key reasons to monitor FEP database.

1. Availability

   It is a very simple equation that if you do not have a database in running, your application will not work. If the application is critical, it directly effects on users and the organization.

2. System Optimization

   Monitoring helps to identify the system bottlenecks and according to the user can make changes to your system to see if it resolves the problem or not. To put this into perspective, there may be a situation where users see a very high load on the system. And figured out that there is a host parameter that can be set to a better value.

3. Identify Performance Problems

   Proactive monitoring can help you to identify future performance problems. From the database side, it could be related to bloating, slow running queries, table and index statistics, or the vacuum being unable to catch up.

4. Business Process Improvement

   Every database user has a different need and priority. Knowing the system (load, user activity, etc.) helps you to prioritize customer tasks, reporting, or downtime. Monitoring helps to make business process improvement.

5. Capacity Planning

   More user or application growth means more system resources. It leads to key questions: Do you need more disk space? Do you need a new read replica? Do you need to scale your database system vertically? Monitoring helps you to understand your current system utilization—and if you have data, points spread over a few weeks or months, it helps to forecast system scaling needs.

This article describes monitoring and alerting operations using OpenShift's standard Pod alive monitoring, resource monitoring and database statistics provided by the FEP Exporter.

5.7.1 Monitoring FEP Operator and Operands

The monitoring of FEP operators and operands are achieved by Prometheus' standard alive and resource monitoring.

Metrics name	Details
Alive monitoring	Can monitor Pod status
Resouce monitoring	You can monitor the following resource status - CPU Usage - CPU Quota - Memory Usage - Memory Quota - Current Network Usage - Receive Bandwidth - Transmit Bandwidth - Rate of Received Packets - Rate of Transmitted Packets - Rate of Received Packets Dropped - Rate of Transmitted Packets Dropped

By setting alert rules based on these monitoring items, operators and operands can be monitored. For the setting method, refer to the appendix in the Reference.

If an error is detected by monitoring the operator's alive, it can be dealt with by recreating the Pod.

If resource monitoring detects an error, consider allocating more resources to the Operator or Operands.

Check the Operator Hub or Red Hat Operator Catlog page to see which version you are currently using, which can be updated, and to check for security vulnerabilities.

5.7.2 Monitoring FEP Server

Monitoring and alerts system leverages standard GAP stack (Grafana, Alert manager, Prometheus) deployed on OCP and Kubernetes. GAP stack must be there before FEP operator & FEPCluster can be deployed.

Prometheus is a condensed way to store time-series metrics. Grafana provides a flexible and visually pleasing interface to view graphs of FEP metrics stored in Prometheus.

Together they let store large amounts of metrics that user can slice and break down to see how the FEP database is behaving. They also have a strong community around them to help deal with any usage and setup issues.

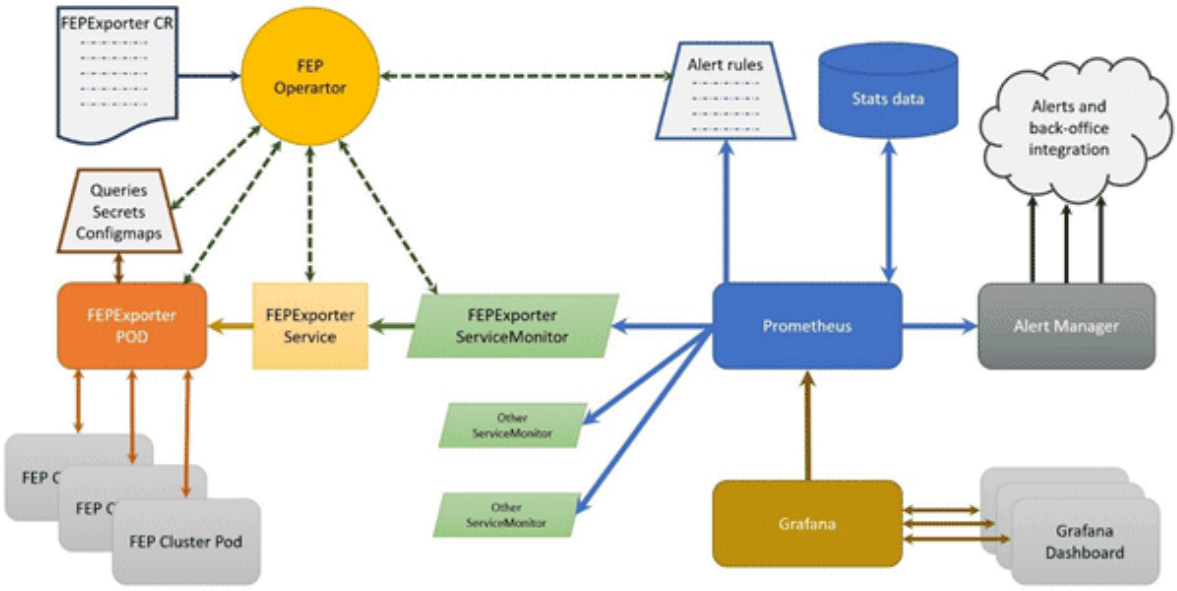
The Prometheus acts as storage and a polling consumer for the time-series data of FEP container. Grafana queries Prometheus to displaying informative and very pretty graphs.

If Prometheus rules are defined, it also evaluates rules periodically to fire alerts to Alert manager if conditions are met. Further Alert manager can be integrated with external systems like email, slack, SMS or back-office to take action on alerts raised.

Metrics from FEP Cluster(s) is collected by Prometheus through optional components deployed using FEP Exporter with default set of metrices and corresponding Prometheus rules to raise alerts. User may extend or overwrite metrics by defining their custom metrics queries and define their custom Prometheus rules for alerting.

5.7.2.1 Architecture

Block diagram of monitoring FEP server is as follows.



- FEPExporter CR is managed by FEP Operator
- When FEPExporter CR is created, FEP operator creates following kubernetes objects:
  - ConfigMap that contains default and custom queries to collect metrics from database cluster from each node
  - Secret containing JDBC URL for all FEPCluster nodes to connect and request metrics. This string contains authentication details as well to make JDBC connection.
  - Prometheus rules corresponding to default alert rules
  - ServiceMonitor for Prometheus to discover FEPExporter service
  - FEPExporter container using FEPExporter image to scrape metrices from all FEPCluster nodes

Note

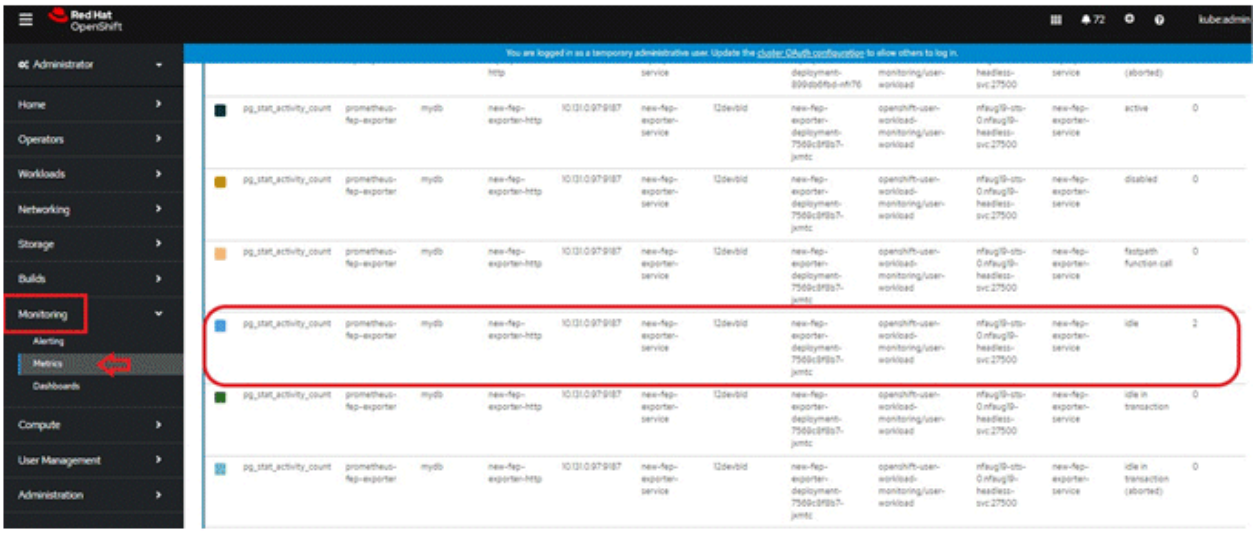
- Alert Manager integration to back-office to send mail / message / raising ticket is done by user based on their environment
- Grafana installation and integration is done by user. Use the Grafana Operator provided by OperatorHub.
- Grafana dashboard is created by user based on their requirements and design.

5.7.2.2 Default Server Metrics Monitoring

By default FEPExporter scrapes some useful metrics for server.

Once FEPExporter is running, user can check the collected metrics under Openshift->Monitoring->Metrics submenu.

Refer an example below.



There are 2 types of default server metrics defined by FEP Exporter.

Type	Details
Default mandatory	Are collected by FEP Exporter. These are kept enabled by default by FEP Exporter and can not be disabled by end user.
Default useful	Useful focused metrics for health and performance metrics. Can be disabled by end user.

Default mandatory metrics

These metrics are either from basic statistics view of the database or FEP Exporter own metrics;

Various metrics under this category are

Metrics name	Details
pg_stat_bgwriter_*	Maps to view in Statistic Collector
pg_stat_database_*	Maps to view in Statistic Collector
pg_stat_database_conflicts_*	Maps to view in Statistic Collector
pg_stat_archiver_*	Maps to view in Statistic Collector
pg_stat_activity_*	Maps to view in Statistic Collector
pg_stat_replication_*	Maps to view in Statistic Collector
pg_replication_slots_*	Maps to System Catalog pg_replication_slots
pg_settings_*	Maps to System Catalog pg_settings
pg_locks_*	Maps to System Catalog pg_locks
pg_exporter_*	Exposes exporter metrics: - last_scrape_duration_seconds (Duration of the last scrape of metrics from PostgresSQL) - scrapes_total (Total number of times PostgresSQL was scraped for metrics) last_scrape_error (*Whether the last scrape of metrics from PostgreSQL resulted in an error; 1 for error & 0 for success*)
pg_*	Exposes exporter metrics - pg_up (set to 1 if the connection to service is success, 0 otherwise)

Metrics name	Details
	- pg_static (can be used to fetch label short_version / version containing postgres server version information)

Default useful metrics

There are certain useful queries which are additionally added to evaluate the health of the Database system.

Metrics name	Details
pg_capacity_connection_*	Metrics on connections e.g. txns running for 1 hour
pg_capacity_schema_*	Metrics on disk space of schema
pg_capacity_tblspace_*	Metrics on disk space of tablespace
pg_capacity_tblvacuum_*	Metrics on tables without vacuum for days
pg_capacity_longtx_*	Number of transactions running longer than 5 minutes Review the information and consider SQL tuning and resource enhancements.
pg_performance_locking_detail_*	Details of processes in blocked state
pg_performance_locking_*	Number of processes in blocked state
pg_replication_*	Replication lag behind master in seconds Provides the ability to check for the most current data in a reference replica To solve the problem, it is necessary to consider measures such as increasing network resources and reducing the load
pg_postmaster_*	Time at which postmaster started
pg_stat_user_tables_*	Important statistics from pg_stat_user_tables
pg_statio_user_tables_*	Important statistics from pg_statio_user_tables
pg_database_*	Database size If the database runs out of space, database restore is required
pg_stat_statements_*	Statistics of SQL statements executed by server
pg_capacity_tblbloat_*	Fetched bloat in tables
pg_tde_encrypted_*	Presence or absence of transparent data encryption in the tablespace and the number of tables and indexes stored
pg_password_valid_*	Database Role Password Validity Period
pg_not_set_password_valid_*	Number of database roles with no password expiration
pg_user_profile_*	Number of database roles in each status for policy-based password operation
pg_txid_*	Transaction ID usage

Note

You can tune the intervals and thresholds at which information is gathered by changing the values specified in the information gathering query. For more information, refer to the queries in the appendix of the Reference Guide, and make your own settings.

5.7.2.3 Default Alerts

There are few basic alert rules which are setup by the FEP Operator as below

Alert rule	Alert Level	Condition persistence	Description
ContainerHighCPUUsage	Warning	5 mins	FEP server container/Pod CPU usage is exceeding 80% of the resource limits
ContainerHighRAMUsage	Warning	30 mins	FEP server container/Pod memory usage is exceeding 80% of the resource limits
PVCLowDiskSpace	Warning	5 mins	A FEP PVC (volume) has less that 10% disk available
ContainerDisappeared	Warning	60 seconds	FEP server container/Pod has disappeared since last 60 seconds
PostgresqlDown	Error	-	FEP server apparently went down or not accessible
PostgresqlTooManyConnections	Warning	-	FEP server container/Pod connection usage is beyond 90% of its available capacity
PostgresqlRolePasswordCloseExpierd	Warning	-	A Postgresql role exists with a password expiration of less than 7 days
PostgresqlRolePasswordExpired	Warning	-	A Postgresql role exists with an expired password
PasswordIsGraceTimeByUserProfile	Warning	-	There is a Postgresql role that is in the grace period due to policy-based password operation
PasswordExpiredByUserProfile	Warning	-	There is a Postgresql role whose password has expired due to policy-based password operation
PasswordLockedByUserProfile	Warning	-	There is a Postgresql role that is locked by policy-based password operation
PostgresqlTooManyTxidUsage	Warning	24 hours	The amount of transaction ID usage has exceeded the value of the autovacuum_freeze_max_age parameter in postgresql.conf for more than 24 consecutive hours.

You can configure any alert by adding alert rules to other monitoring items.

Alerts are based on statistics/metrics. Incorrect platform statistics can cause false alarms. For example, when using NFS storage, the system may raise false alarms for PVCLowDiskSpace when the storage driver is not showing the correct metrics for PV byte usage.

5.7.2.4 Graphical user interface

User can build their custom dashboard using default and custom metrics.

An example Grafana dashboard screenshot is shown below

5.7.2.5 Metrics Collected by CloudWatch

The metrics collected by CloudWatch are listed below.

Metrics name	Description
pg_stat_bgwriter_*	Maps displayed in statistics collection
pg_stat_database_*	Maps displayed in statistics collection
pg_stat_database_conflicts_*	Maps displayed in statistics collection
pg_stat_archiver_*	Maps displayed in statistics collection
pg_stat_activity_*	Maps displayed in statistics collection
pg_stat_replication_*	Maps displayed in statistics collection
pg_replication_slots_*	Mapping to the pg_replication_slots system catalog
pg_locks_*	Mapping to the pg_locks system catalog
pg_capacity_connection_*	Connection metrics (e.g. txns running for 1 hour)
pg_capacity_schema_*	Schema disk space metrics
pg_capacity_tblspace_*	Tablespace disk space metrics
pg_capacity_tblvacuum_*	Metrics for tables that haven't been vacuumed in days
pg_capacity_longtx_*	Number of transactions running for more than 5 minutes
pg_performance_locking_detail_*	Blocked process details
pg_performance_locking_*	Number of blocked processes
pg_stat_user_tables_*	Vital statistics from pg_stat_user_tables
pg_statio_user_tables_*	Vital statistics from pg_statio_user_tables
pg_stat_statements_*	Statistics for SQL statements executed by the server
pg_capacity_tblbloat_*	Table fetch bloat

5.7.3 Monitoring FEP Backup

You can view information about the backed-up data and the status of the backup process in the FEP server tables and system views.

Backup information is updated when the automatic backup process completes or when backup data is deleted as specified by retention.

The following tables and views are added. The tables and views to be added are created under the fep_exporter schema in the postgres database on the FEP server.

Table/View name	Details
pgbackrest_info_backup	Backup Processing Status

5.7.3.1 pgbackrest_info_backup view

Contains one line per backup for information about the state of the backup.

Column	Type	Description
label	text	Information identifying the backup
type	text	full: full backup, incr: incremental backup
prior	text	Label of the backup that should be applied first (For incremental backups only)
database_size	bigint	Database size
database_size_comp	bigint	Database size (After Compression)
backup_size	bigint	Backup size
backup_size_comp	bigint	Backup size (After Compression)
archive_start	text	Range of WALs required for restore (Start)
archive_stop	text	Range of WALs required for restore (End)
backup_start	timestamp with timezon	Backup Start Time
backup_stop	timestamp with timezone	Backup End Time
backup_exec_time	interval	The duration of the backup

5.7.4 Monitoring FEP PGPool2

Information about pgpool2 activity and replication status can be found in the FEP server table and in the system view.

The pgpool2 statistics are updated according to the schedule specified in the parameter.

The tables and views that have been added are described below. The tables and views to be added are created under the fep _ exporter schema in the postgres database on the FEP server.

Table/View name	Details
pgpool2_stat_load_balance	Load Balance Information in pgpool2
pgcluster_stat_replication	Replication State
pgpool2_stat_conn_pool	Connection Pool State for pgpool2
pgpool2_stat_sql_command	SQL Command Statistics

5.7.4.1 pgpool2_stat_load_balance view

Contains one row for MasterService and one row for ReplicaService.

Column	Type	Description
node_id	integer	database node id (0 or 1)
status	text	status (up or down)
lb_weight	double precision	load-balancing weight

Column	Type	Description
role	text	role (primary or standby)
last_status_change	timestamp with time zone	last status change time

5.7.4.2 pgpool2_stat_conn_pool view

Indicates the state of the connection pool. Contains connection pool information for each pcpool2 instance.

Column	Type	Description
pgpool2_node_id	integer	pgpool2 node id（0 - the number of pgpool2 instance -1）
pool_pid	integer	The PID of the displayed Pgpool-II process
start_time	timestamp with timezone	The timestamp of when this process was launched
pool_id	integer	The pool identifier (should be between 0 and max_pool - 1)
backend_id	integer	The backend identifier (should be between 0 and the number of configured backends minus one)
role	text	role (primary or standby)
database	text	The database name for this process's pool id connection
username	text	The user name for this process's pool id connection
create_time	timestamp with timezo	The creation time and date of the connection
majorversion	integer	The protocol version numbers used in this connection
minorversion	integer	The protocol version numbers used in this connection
pool_counter	integer	Counts the number of times this pool of connections (process) has been used by clients
pool_connected	boolean	True (1) if a frontend is currently using this backend

5.7.4.3 pgpool2_stat_sql_command view

Represents SQL command statistics.

Column	Type	Description
node_id	integer	The backend identifier (should be between 0 and the number of configured backends minus one)
role	text	role (primary or standby)
select_cnt	integer	The numbers of SQL command: SELECT
insert_cnt	integer	The numbers of SQL command: INSERT
update_cnt	integer	The numbers of SQL command: UPDATE
delete_cnt	integer	The numbers of SQL command: DELETE
ddl_cnt	integer	The numbers of SQL command: DDL
other_cnt	integer	The numbers of SQL command: others
panic_cnt	integer	The numbers of failed commands
fatal_cnt	integer	The numbers of failed commands
error_cnt	integer	The numbers of failed commands

5.8 Event Notification

The eventing mechanism introduced, is to enable operator to raise customized Kubernetes events. The custom events will be raised during the creation of custom resources. Currently following events are raised.

5.8.1 Events raised

- fepcluster - During FEPCluster CR creation
  - Event is raised when FEPVolume CR creation is initiated and when FEPVolume CR creation initiation fails.
  - Event is raised when FEPConfig CR creation is initiated and when FEPConfig CR creation initiation fails.
  - Event is raised when FEPUser CR creation is initiated and when FEPUser CR creation initiation fails.
  - Event is raised when FEPCert CR creation is initiated.and when FEPCert CR creation initiation fails.
  - Event is raised when Statefulset creation is successful and Statefulset creation fails.
  - Event is raised when PDB creation is successful and when PDB creation fails.
  - Event is raised when FEPBackup CR creation is initiated and when FEPBackup CR creation initiation fails.
  - Event is raised when a FEPCluster custom resource with Velero enabled specifies the same object storage path for production and disaster recovery environments.

Please note the following child CR events are raised as part of Create FEP Cluster

  - fepcert - During FEPCert CR creation
    - Event is raised when FEPCert CR creation is successful, when FEPCert CR fails annotating FEPCluster and when FEPCert CR creation fails.
  - fepconfig - During FEPConfig CR creation
    - Event is raised when FEPConfig CR creation is successful, when FEPConfig CR fails annotating FEPCluster and when FEPConfig CR creation fails.
  - fepvolume - During FEPVolume CR creation
    - Event is raised when FEPVolume CR creation is successful, when FEPVolume CR fails annotating FEPCluster and when FEPVolume CR creation fails.
  - fepbackup - During FEPBackup CR creation
    - Event is raised when FEPBackup cronjob1 creation is successful and when FEPBackup cronjob1 creation fails.
    - Event is raised when FEPBackup cronjob2 creation is successful and when FEPBackup cronjob2 creation fails.
    - Event is raised when FEPBackup cronjob3 creation is successful and when FEPBackup cronjob3 creation fails.
    - Event is raised when FEPBackup cronjob4 creation is successful and when FEPBackup cronjob4 creation fails.
    - Event is raised when FEPBackup cronjob5 creation is successful and when FEPBackup cronjob5 creation fails.
- feppgpool2- During FEPPgPool2 CR creation
  - Event is raised when FEPPgPool2 CR creation is successful and when FEPPgPool2 CR creation fails.
  - Event is raised when FEPPgPool2Cert CR creation is initiated and when FEPPgPool2Cert CR creation initiation fails.

Please note the following child CR event are raised as part of Create FEP PgPool2

  - feppgpool2cert- During FEPPgPool2Cert CR creation
    - Event is raised when FEPPgPool2Cert CR creation is successful, when FEPPgPool2Cert CR fails annotating FEPPgPool2 and when FEPPgPool2Cert CR creation fails

- feprestore - During FEPRestore CR creation
    - Event is raised when FEPRestore CR creation is successful and when FEPRestore CR creation fails.

5.8.2 Events that Occur when Custom Resources are Updated

If the "sysExtraEvent" parameter is specified in the custom resource, an event will be generated when changes to FEPCluster, FEPLogging, FEPExporter are detected, or when the changes are applied successfully/failed.

Refer to "Operator operation event notification" in "Reference" for information about events that occur.

5.8.3 Viewing the Custom Events

The custom events can be viewed on CLI as well as the Openshift console

On cli

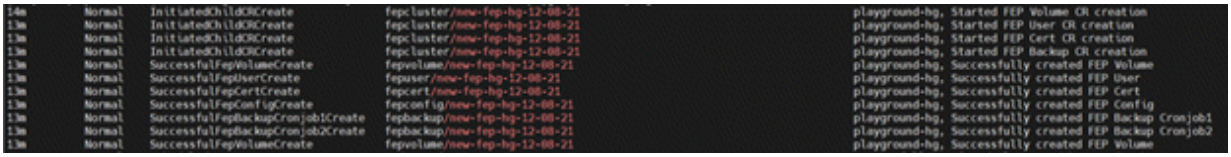
Executing the command

kubectl get events

OR

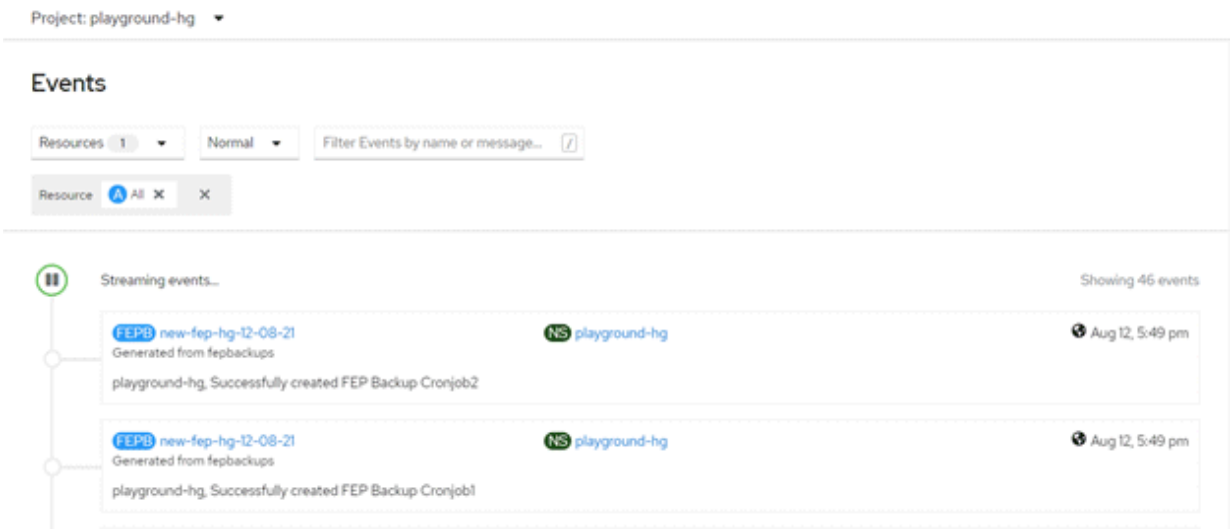
oc get events

Following is a snippet of the events output is ==shown when the above command is executed,



On openshift console

For the specific project/ namespace the custom events can be viewed along with Kubernetes events under the events as shown in the following screenshot.



5.9 Scaling Replicas

5.9.1 Automatic Scale Out

Automatic scale out occurs when the average CPU utilization or number of connections of the DB container exceeds the threshold.

The maximum number of replica containers, excluding the master container, is 15.

If the load decreases after the number of replicas increases due to a temporary increase in load, the number of replicas will remain increased. Perform manual scale in if necessary.

Specify spec.fepChildCrVal.autoscale.scaleout in FEPClusterCR when you want to perform Automatic scale out. Refer to "FEPCluster Parameters" in the Reference for information about the values to specify.

```
$ oc edit fepcluster <FEPClusterCR name>
```

5.9.2 Manual Scale In/Out

To manually scale in or out of a FEPCluster, edit the "spec.fep.instances" in FEPClusterCR.

The value must be between 1 and 16. (Number of instances with one master)

```
$ oc edit fepcluster <FEPClusterCR name>
```

Note

- Do not scale in from two to one replica instance when the syncMode is 'on'. Update SQL cannot be executed.
- Any database connections to the replica Pod that are deleted during a scale in will be forced to disconnect.

5.10 Backing Up to Object Storage

Describes how to store backup data in object storage.

5.10.1 Pre-creation of Resources

5.10.1.1 Storing CA Files (Root Certificates)

If you want to use a non-default root certificate for object storage connections, register it in ConfigMap.

```
$ oc create configmap storage-cacert --from-file=ca.crt=storage-ca.pem -n my-namespace
```

5.10.1.2 Storing Repository Key

When using the parameter (repo-gcs-key) of pgBackRest, register the GCS repository key in Secret.

```
$ oc create secret generic storage-key-secret --from-file=key.json=storage-key.json -n my-namespace
```

5.10.2 Defining a FEPCluster Custom Resource

List the backup settings under spec.fepChildCrVal.backup in the FEPCluster custom resource.

Specify the object storage for the backup data in pgbackrestParams. Refer to "2.3.5 Scheduling Backup from Operator" for possible values for pgbackrestParams.

Specify the ConfigMap name created in "5.10.1.1 Storing CA Files (Root Certificates)" for caName.

FEPCluster Custom Resource Example: Only Object Storage Used for Backup Repository

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  …
spec:
  fepChildeCrVal:
    backup:
      pgbackrestParams: |
        repo1-type=s3
```

```
        repo1-path=/backup/cluster1
        repo1-s3-bucket= sample-bucket
        repo1-s3-endpoint=s3.ap-northeast-1.amazonaws.com
        repo1-s3-region=ap-northeast-1
        repo1-storage-ca-file=/pgbackrest/storage-certs/ca.crt
      pgbackrestKeyParams: |
        repo1-s3-key=SAMPLEKEY
        repo1-s3-key-secret=SAMPLESECRET
      caName:
       - storage-cacert
...
```

If the persistent volume and object storage specified in spec.fepChildeCrVal.storage.backupVol are to be used together in the backup repository, specify the object storage setting after "repo2".

If "repo1" is not defined, a permanent volume is automatically designated as the storage destination for the backup volume.

FEPCluster Custom Resource Example: When using object storage and PV

```
  ...
spec:
  fepChildeCrVal:
    backup:
      pgbackrestParams: |
        repo2-type=s3
        repo2-path=/backup/cluster1
        repo2-s3-bucket= sample-bucket
        repo2-s3-endpoint=s3.ap-northeast-1.amazonaws.com
        repo2-s3-region=ap-northeast-1
        repo2-storage-ca-file=/pgbackrest/storage-certs/ca.crt
      pgbackrestKeyParams: |
        repo2-s3-key=SAMPLEKEY
        repo2-s3-key-secret=SAMPLESECRET
      caName:
       - storage-cacert
...
```

When using object storage GCS as a backup repository, specify as follows.

For repoKeySecretName, specify the Secret created in "5.10.1.2 Storing Repository Key". Also, specify service for gcs-key-type.

FEPCluster Custom Resource Example: When using GCS as a backup repository

```
apiVersion: fep.fujitsu.io/v1
kind: FEPCluster
metadata:
  ...
spec:
  fepChildeCrVal:
    backup:
      pgbackrestParams: |
        repo1-type=gcs
        repo1-path=/backup-ct/test2
        repo1-gcs-bucket=dbaas-gcs
        repo1-gcs-endpoint=localhost
        repo1-storage-ca-file=/pgbackrest/storage-certs/ca.crt
        repo1-gcs-key=/pgbackrest/storage-keys/key.json
        repo1-gcs-key-type=service
      caName:
       - storage-cacert
      repoKeySecretName:
```

```
      - storage-key-secret
...
```

5.11 Disaster Recovery

Available disaster recovery methods include backup/restore method, continuous recovery method, and streaming replication method.

5.11.1 Disaster Recovery by Backup/Restore Method

5.11.1.1 Disaster Recovery Prerequisites

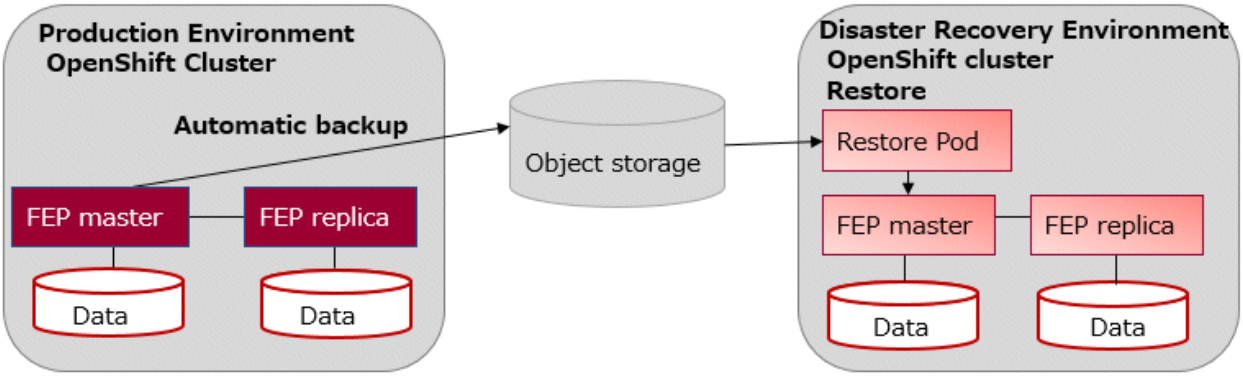
The configuration diagram of the Pod arrangement and backup repository, which are prerequisites for the backup function to perform disaster recovery using the backup/restore method, is shown below.

In FEPCluster to get a backup, specify the object storage as the backup data storage destination with spec.fepChildCrVal.backup.pgbackrestParams.

Specify object storage that is in an area that is considered safe for the scope of the expected disaster.

The definition of the FEPCluster custom resource is not inherited when performing disaster recovery.

We recommend that you save your production environment FEPCluster custom resource definitions in case of a disaster.



5.11.1.2 Performing Disaster Recovery

Describes the procedure for restoring to an OCP environment different from the restore source using the backup data stored in the object storage.

5.11.1.2.1 Pre-creation of Resources

Storing CA Files (Root Certificates)

If you want to use a non-default root certificate for object storage connections, register it in ConfigMap.

```
$ oc create configmap storage-cacert --from-file=ca.crt=storage-ca.pem -n my-namespace
```

Storing GCS Repository Key

When using the parameter (repo-gcs-key) of pgBackRest, register the GCS repository key in Secret.

```
$ oc create secret generic storage-key-secret --from-file=key.json=storage-key.json -n my-namespace
```

5.11.1.2.2 Defining a FEPCluster Custom Resource

In addition to the FEPCluster settings, specify the Restore settings below.

FEPCluster Custom Resource Example

```
apiVersion: fep.fujitsu.io/v1
kind: FEPCluster
metadata:
  …
spec:
  fepChildCrVal:
    restore:
      pgbackrestParams: |
        repo1-type=s3
        repo1-path=/backup/cluster1
        repo1-s3-bucket=sample-bucket
        repo1-s3-endpoint=s3.ap-northeast-1.amazonaws.com
        repo1-s3-region=ap-northeast-1
        repo1-storage-ca-file=/pgbackrest/storage-certs/ca.crt
      pgbackrestKeyParams: |
        repo1-s3-key=SAMPLEKEY
        repo1-s3-key-secret=SAMPLESECRET
      caName:
       - storage-cacert

…
```

When using object storage GCS as a backup repository, specify as follows.

For repoKeySecretName, specify the Secret created in "Storing GCS Repository Key". Also, specify service for gcs-key-type.

```
apiVersion: fep.fujitsu.io/v1
kind: FEPCluster
metadata:
  …
spec:
  fepChildeCrVal:
    backup:
      pgbackrestParams: |
        repo1-type=gcs
        repo1-path=/backup-ct/test2
        repo1-gcs-bucket=dbaas-gcs
        repo1-gcs-endpoint=localhost
        repo1-storage-ca-file=/pgbackrest/storage-certs/ca.crt
        repo1-gcs-key=/pgbackrest/storage-key/key.json
        repo1-gcs-key-type=service
      caName:
       - storage-cacert
      repoKeySecretName:
       - storage-key-secret
…
```

Setting value

Field	Default	Details
spec.fepChildCrVal.restore		Define when restoring by specifying the backup data stored in the object storage.
spec.fepChildCrVal.restore.pgbackrestParams		Optional "|" is fixed, and the following lines specify the parameters to set in pgbackrest.conf. Specify the object storage where the backup data is stored.

Field	Default	Details
		If you want to use a root certificate other than the default, specify the following: repo1-storage-ca-path=/pgbackrest/storage-certs/<*file name*> Register the CA file in ConfigMap and specify the ConfigMap name in spec.fepChildCrVal.restore.caName.
spec.fepChildCrVal.restore.pgbackrestKeyParams		Optional "|" is fixed, and the following lines specify the parameters to set in pgbackrest.conf. The value described by this parameter is masked with *****. Specify the parameter you want to mask, such as a password.
spec.fepChildCrVal.restore.caName		Optional Specify when you use a CA file other than the system default. Specify the name of the created ConfigMap in list format. The specified ConfigMap will be mounted in /pgbackrest/storage-certs.
spec.fepChildCrVal.restore.mcSpec.limits	cpu: 200m memory: 300Mi	Optional CPU and memory allocated to the container performing the restore.
spec.fepChildCrVal.restore.mcSpec.requests	cpu: 100m memory: 200Mi	Optional CPU and memory allocated to the container performing the restore.
spec.fepChildCrVal.restore.restoretype	latest	Optional Restore Type (latest or PITR)
spec.fepChildCrVal.restore.restoredate		Optional Specify the date to restore when spec.fepChildCrVal.restore.restoretype is "PITR".
spec.fepChildCrVal.restore.restoretime		Optional Specify the time to restore when spec.fepChildCrVal.restore.restoretype is "PITR".
spec.fepChildCrVal.restore.image		Optional Image of the container to perform the restore. It is omitted by default. In this case, the URL for image is obtained from the operator container environment.
spec.fepChildCrVal.restore.imagePullPolicy	IfNotPresent	Optional

5.11.2 Disaster Recovery with Continuous Recovery Method

5.11.2.1 Disaster Recovery Prerequisites

When performing disaster recovery using the continuous recovery method, the database cluster must be configured based on the "4.14.3.1 Defining a Continuous Recovery Method". At this time, it is necessary to periodically back up the database to object storage.

5.11.2.2 Performing Disaster Recovery

In the event of a disaster, the FEPCluster in the disaster recovery environment should be promoted.

Describes the procedure for promoting a FEPCluster in a disaster recovery environment deployed in a hot standby configuration. If the production environment becomes unusable when a disaster occurs, you can promote the disaster recovery environment to the production environment by executing FEPAction(promote_standby).

An example FEPAction definition is shown below.

```
...
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: new-fep-promote-standby-action
  namespace: my-namespace
spec:
  fepAction:
    type: promote_standby
  sysExtraEvent: true
  sysExtraLogging: true
  targetClusterName: my-fep
...
```

5.11.3 Disaster Recovery Using Velero

This section describes disaster recovery using Velero.

5.11.3.1 Disaster Recovery Prerequisites

Prepare for the following in production environment and disaster recovery environment. Before enabling this feature in a production environment, and before restoring using this feature in a disaster recovery environment, you must prepare the following:

- Installing VeleroCLI
- Installing Velero
- Installing FEPOperator
- StorageClass, namespace, CRD, and other resources needed to build the system

If you plan to use a static PV with FEPCluster, prepare the PV before restoring to a disaster recovery environment. If you use a storage class or certificate that is different from the production environment, provide a storage class or certificate with the same name as the production environment in your disaster recovery environment. However, if the object storage certificate or secret where the database backup data is stored differs between the production environment and the disaster response environment, set a different name. For more information, refer to "FEPCluster Parameters" in the Reference.

This feature only supports x86.

5.11.3.2 Performing Disaster Recovery

Provides step-by-step instructions on how to enable this feature to perform disaster recovery from a production environment to a disaster recovery environment. It also describes how to perform disaster recovery again from a disaster recovery environment to a production environment.

FEPCluster and FEPPool2 LOG volumes cannot be restored. Before using this feature, be sure to test and validate in both production and disaster recovery environments.

5.11.3.2.1 Configuring FEPCluster Custom Resources

To use this feature, in addition to the normal configuration of FEPCluster custom resources in your production environment, configure the following to deploy on Kubernetes. For more information, refer to "FEPCluster Parameters" in the Reference.

- fep.velero.enable
- fep.velero.labels
- fep.velero.backup
- fep.velero.restore
- fepChildCrVal.backup

Example) FEPCluster Custom Resource Definition Example

```
spec:
  fep:
    velero:
      enable: true
      labels:
        backup-dev: my-backup1
        backup-dep: my-backup2
      backup:
        pgbackrestParams: |
          [global]
           repo1-retention-full=5
           repo1-retention-full-type=count
           repo2-retention-full=5
           repo2-retention-full-type=count
           log-path=/database/log/backup
           log-level-file=debug
           repo2-path=/ velero-backup/ velero-backup-for-dr2
           repo2-s3-bucket=my-s3-backet
           repo2-s3-endpoint=s3.ap-northeast-1.amazonaws.com
           repo2-s3-region=ap-northeast-1
           repo2-type=s3
         pgbackrestKeyParams: |
           repo2-s3-key=XXXXXXXXXX
           repo2-s3-key-secret=YYYYYYYYYY
         caName: DR-objectstorage-cert
         repoKeySecretName: XXX

resotre:
         image:
           image: "XXX-amd64"
           pullPolicy: IfNotPresent
         mcSpec:
           limit:
             cpu: 200m
             memory: 300Mi
           request:
             cpu: 100m
             memory: 200Mi
         restoreTargetRepo: 2
  fepChildCrVal:
      backup:
        pgbackrestParams: |
        [global]
         repo1-retention-full=5
         repo1-retention-full-type=count
         repo2-retention-full=5
         repo2-retention-full-type=count
         log-path=/database/log/backup
         log-level-file=debug
         repo2-path=/ velero-backup/ velero-backup-for-dr1
         repo2-s3-bucket=my-s3-backet
         repo2-s3-endpoint=s3.ap-northeast-1.amazonaws.com
         repo2-s3-region=ap-northeast-1
         repo2-type=s3
        pgbackrestKeyParams: |
          repo2-s3-key=XXXXXXXXXX
          repo2-s3-key-secret=YYYYYYYYYY
        caName: objectstorage-cert
        repoKeySecretName: ZZZ

        schedule:
          num: 1
          schedule1:
```

```
      schedule: "0-59/3 * * * *" #schedule1.schedule
    type: "full" #schedule1.type
    repo: 2
```

5.11.3.2.2 Velero Backup

After building FEPCluster in this environment or modifying FEPCluster custom resources, use the following command to back up the resources on kubernetes, including FEPCluster custom resources, to object storage.

Example)

```
velero backup create <Bavkup name> --selector <Backup target label>
```

This feature supports only Velero backup commands that specify the labels described in fep.velero.labels for FEPCluster custom resources and back up only those resources to which the labels are attached. Other commands may back up and restore unnecessary resources, resulting in build and data recovery failures. When specifying labels, do not specify the following keys to avoid confusion with existing labels.

- app
- control-plane
- name
- pod-template-hash
- vendor
- app.kubernetes.io/component
- app.kubernetes.io/instance
- app.kubernetes.io/managed-by
- app.kubernetes.io/name
- app.kubernetes.io/part-of
- control-plane
- controller-revision-hash
- fepclustername
- feprole
- statefulset.kubernetes.io/pod-name
- controller-uid
- job-name
- pod-template-hash
- release

This feature labels the following resources:

- FEPCluster Custom Resources
- Secrets required to build FEPCluster

When backing up the following resources using Velero, assign the labels specified in spec.fep.velero.labels (If spec.fep.velero.labels is omitted, the default backup-group: fep-backup) to the ConfigMap and Secret specified for the various custom and custom resources.

- FEPPgpool2
- FEPExporter
- FEPLogging

You can also back up manually created resources (Application system, ConfigMap, Secret, etc.) by assigning them the labels specified in spec.fep.velero.labels (If spec.fep.velero.labels is omitted, the default backup-group: fep-backu) and making them eligible for Velero backups.

For more information about using the Velero command, refer to the official documentation.

Each time you modify a custom resource, such as FEPCluster, a Velero backup is required. The database is brought up to date with object storage. If a custom resource is in a pre-change state, it may be inconsistent with the database, causing data loss, performance degradation, security, and other issues.

5.11.3.2.3 Database Backup

Perform regular database backups.

5.11.3.2.4 Velero Restore

If business cannot continue in a production environment due to a disaster, use the following command to restore the resources on kubernetes, including the FEPClueter custom resources stored in object storage. The database data is restored to the latest archived WAL state stored in object storage.

Example)

```
velero restore create <Reatore name> --from-backup <Backup name>
```

For more information about using the Velero command, refer to the official documentation.

5.11.3.2.5 Return from Disaster Recovery Environment to Production Environment

When restoring from a disaster recovery environment to a production environment again using Velero, the FEPCluster custom resources deployed in the disaster recovery environment must be modified. The FEPClsuter custom resource after disaster recovery, fep.velero.backup, fep.velero.restore, defines pre-disaster recovery information. Update the information to the object storage used in the production environment. Then back up with Velero in a disaster recovery environment and restore with Velero in a production environment.

5.11.4 Disaster Recovery with Streaming Replication Method

5.11.4.1 Disaster Recovery Prerequisites

When performing disaster recovery using the streaming replication method, the database cluster must be configured based on the "4.14.3.2 Defining a Streaming Replication Method". After deployment, database data is synchronized between the production environment and the disaster recovery environment using the streaming replication method, so no operation is required during operation.

5.11.4.2 Performing Disaster Recovery

When a disaster occurs, it is necessary to promote the FEPCluster in the disaster recovery environment. For promotion of FEPCluster in a disaster recovery environment, refer to "5.11.2.2 Performing Disaster Recovery".

5.11.5 Parameter Change in Disaster Recovery Environment

If parameters (such as FEPClusterCR) are changed in the production environment, manually reflect the changes in the disaster recovery environment. However, if the database password is changed, the post-change value is automatically reflected in the disaster recovery environment, so manual reflection is not required.

5.12 Operation of Transparent Data Encryption Using Key Management System

5.12.1 Updating Custom Resource Parameters

When using a newly generated master encryption key in your key management system, update the FEPCluster custom resource fepChildCrVal.sysTde.tdek.targetKeyId to the ID of the new master encryption key. The operator will automatically re-enable TDE when this value is updated.

Also, if the credentials for connecting with the key management system are updated, update the corresponding values in the FEPCluster custom resource. The operator automatically performs a keystore open when the credentials are updated.

When re-enabling TDE or opening the keystore is completed, the following event will be notified.

```
# When re-enabling TDE
$ kubectl get event
LAST SEEN   TYPE    REASON                      OBJECT                       MESSAGE
164m       Normal  SuccessfulTdeSetMasterKey  fepconfig/<FEPClusterCR name> <namespace>, Successfully
set TDE masterKey

# When re-enabling TDE fails
$ kubectl get event
LAST SEEN   TYPE    REASON                      OBJECT                       MESSAGE
164m        Warning  FailedTdeSetMasterKey      fepconfig/<FEPClusterCR name> <namespace>, Error/
Failure set TDE masterKey
```

If the process fails, review the parameters defined in the FEPCluster custom resource and re-enter the correct values.

If only the contents of the Secret or ConfigMap that stores the credentials are updated and the custom resource is not modified, open the keystore using the FEPAction custom resource described in "5.12.2 Update Credentials".

5.12.2 Update Credentials

If the credentials in the key management system are updated, update the contents of the corresponding Secret or ConfigMap. If there are no changes to the values specified in the FEP cluster custom resource, apply the FEPAction custom resource to update the credentials used by FEP.

Example) Definition example of FEPAction custom resource

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: new-fep-action
spec:
  sysExtraLogging: false
  targetClusterName: nf-131851
  fepAction:
    type: open_tde_masterkey
```

5.12.3 Encrypting a Tablespace

If you create an encrypted tablespace, configure the encryption algorithm in runtime parameters. For example, to create a tablespace named secure_tablespace using AES with a 256-bit key length as the encryption algorithm, define:

```
-- Specify the encryption algorithm for the tablespace to be created below
SET tablespace_encryption_algorithm = 'AES256';
CREATE TABLESPACE secure_tablespace LOCATION /database/tablespaces/tbspace1;
-- Specify that the tablespace to be created below is not to be encrypted
SET tablespace_encryption_algorithm = 'none';
```

Or

```
CREATE TABLESPACE tbs_tst_new LOCATION '/database/tablespaces/tbspace1' WITH
(tablespace_encryption_algorithm = 'AES256' );
```

Checking for encrypted tablespaces

You can check which tablespaces are encrypted by executing the following SQL.

```
SELECT spcname, spcencalgo FROM pg_tablespace ts, pgx_tablespaces tsx WHERE ts.oid =
tsx.spctablespace;
```

5.12.4 Backup/Restore

In case the FEP cluster is damaged or lost, backups should be made at the following times:

- When the cluster is first created
- When the master encryption key is changed

When you use the FEPRestore custom resource to create a cluster restored from backup, the restored cluster is restored with the master encryption key at the time the backup was taken on the source cluster (where the backup was created from).

If a newer master encryption key is specified in sysTde.tdek.targetKeyId than when the source FEPCluster was backed up, the value will be carried over to the restore destination FEPCluster custom resource, and the operator automatically re-enables TDE with the new master encryption key after data recovery.

Also, update the authentication information to the key management system before executing the restore. If your credentials are not up-to-date, FEP will not be able to connect to the key management service and restore your data.

If you mistakenly update the information for connecting to the key management system under sysTde.tdek.kmsDefinition after building FEPCluster, FEP will not be able to refer to the key management system when restoring data. Before executing the restore process, confirm that the correct values are described in the FEPCluster custom resource.

5.12.5 Changing Key Management System Definitions

Modify the parameters under spec.fepChildCrVal.sysTde.tdek.kmsDefinition in the FEPCluster custom resource if you want to add or change the connection information to the key management system.

If you make any of the following changes, the replica server will be restarted with the new parameters:. If there are multiple replica servers, they are restarted one at a time. When all replica servers are restarted, one of them is promoted to the new master server due to a switchover. The original master server's container image is then restarted. This allows you to change the definition of the key management system for all servers with minimal disruption.

- Add a new key management system definition
- Delete an existing key management system definition
- Change the order of key management system definitions
- Add, Delete, or rename ConfigMap or Secret resources that you specify as credentials

If you make changes that require a restart, temporarily disable the automatic scale out feature for the database before making the changes. The automatic scale out feature can be disabled with the spec.fepChildCrVal.autoscale.scaleout.policy parameter of the FEPCluster custom resource.

You cannot rename the ConfigMap/Secret resource that you currently specify as the credential for the key management system you are using as the keystore.

5.13 Confidentiality Management Feature

5.13.1 Enabling Confidentiality Management Feature

When building FEPCluster, the extension "pgx_confidential_management_support" of the confidentiality management feature was installed and set up in the following database.

- template1
- postgres
- Database specified in spec.fepChildCrVal.sysUsers.pgdb

In addition, when creating a confidential administrator role (spec.fepChildCeVal.sysUsers.pgSecurityUser), this role is assigned the following functions necessary for executing confidentiality management feature.

- CREATE ROLE
- SELECT, INSERT, UPDATE, and DELETE privileges on all tables included in the extension

Therefore, immediately after FEPCluster is built, database objects can be managed by the confidentiality management feature in a database in which the extension "pgx_confidential_management_support" is installed or in a database created from template1.

Refer to "Confidentiality Management" in the Fujitsu Enterprise Postgres Security Operation Guide for details on how to operate the security management support function.

Refer to "Tables Used by Confidentiality Management Feature" in the Fujitsu Enterprise Postgres Security Operation Guide for tables included in the extension.

In addition, if a database role other than the confidential administrator role needs to operate the confidentiality management feature, such as by preparing a database role for each schema that manages database objects using the confidentiality management feature, the confidentiality management feature assign the following privileges to the datdabase role.

- CREATE ROLE
- SELECT, INSERT, UPDATE, and DELETE privileges on all tables included in the extension

When using the confidentiality management feature to manage database objects created by other users, it is necessary to grant ownership of the database objects to the database role that operates the confidentiality management feature.

Example) When giving ownership of the table "security_table" to the confidential administrator user "security_user"

```
ALTER TABLE security_table OWNER TO security_user;
```

The owner of the database object can be confirmed using the PostgreSQL meta-command "\d".

5.13.2 Monitoring Confidentiality Management Feature

You can forward pgAudit's audit logs to Elasticseach using the FEP logging feature.

Analyze the transferred audit log and monitor for changes in privileges on database objects by unintended users.

For the FEP log feature, please refer to "4.10 FEP Logging".

5.14 Operation of Fixed Statistics

Describes how to periodically fix statistics in the production environment using statistics whose performance is guaranteed in the validation environment.

5.14.1 Preparing for Object Storage Connections

Deploy a secret that contains the credentials required to operate object storage.

5.14.1.1 Using S3

Secret - Name: my-aws-s3-secret

```
data:
aws_access_key: cG9zdGdyZXM=
aws_access_secret: ZnNlcaAZnNlcA3A==
```

5.14.1.2 Using AzureBlob

Secret - Name: my-azure-blob-secret

```
data:
azure_storage_account_name: cG9zdGdyZXM=
azure_storage_account_key: ZnNcG9zdGads3cGzdGdyZXMyZXMlcA==
```

5.14.1.3 Using Google Cloud Storage

Deploy the json file required for authentication with the following command:

```
kubectl create secret generic <sercret> --from-file=key.json=<json file>
```

5.14.2 Storing validation environment statistics in object storage

Run FEPAction to export the validation environment statistics and store them in object storage. Define the connection information to object storage in the FEPCluster custom resource before executing FEPAction.

The following is an example definition of FEPAction that stores a binary file that exports the database "mydatabase" in object storage AWS s3.

Example) Definition example of FEPCluster custom resource

```
spec:
  fep:
    fixedStats:
      endpoint: #Required
        protocol: s3
        authentication: aws-fixedstats-credentials
```

Example) Definition example of FEPAction custom resource

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: fep-action-fixdStats
spec:
  fepAction:
    targetClusterName: new-fep #target cluster
    type: fixed_stats
    args:
      fixedStatsType: export
      targetDb: mydatabase
      url: 's3://export_stats/file'
      targetStats: effective
```

5.14.3 Schedule for Fixed Statistics

Define the object storage information and schedule where the statistics are stored in the FEPCluster custom resource in the production environment. You can also add, modify, or delete schedules after they are built.

The following is an example of the FEPCluster custom resource definition for each Friday from 20:00 to 23:00 when the database "mydatabase" is pinned with statistics stored in object storage AWS s3.

Example) Definition example of FEPCluster custom resource

```
spec:
  fep:
    fixedStats:
      schedule1:
        update: true
        fixSchedule: "0 20 * * 5"
        unfixSchedule: "0 23 * * 5"
        targetDb: mydatabase
        url: 's3://export_stats/file '
      endpoint:
        protocol: s3
        authentication: my-aws-s3-secret
```

Once the statistics are downloaded from object storage, they are backed up to the database. Also, spec.fep.fixedStats.schedleN.update will be from true to false.

If you want to update the statistics that you want to freeze, change spec.fep.fixedStats.schedleN.update to ture and run the statistics import from object storage again.

5.14.4 Using Local Storage

If object storage is not available, use local storage.

5.14.4.1 Exporting Statistics Files to FEPCluster in a Validation Environment

Execute FEPAction and export the statistics whose performance is guaranteed in the verification environment to the file system /database/userdata/action/local/ on the container.

Example) Definition example of FEPCluster custom resource

```
spec:
  fep:
    fixedStats:
      endpoint: #Required
        protocol: local
```

Example) Definition example of FEPAction custom resource

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: fep-action-fixdStats
spec:
  fepAction:
    targetClusterName: new-fep
    type: fixed_stats
    args:
      fixedStatsType: export
      targetDb: mydatabase
      file: file_name.dump
      targetStats: effective
```

5.14.4.2 Deploying Statistics Files to FEPCluster in a Production Environment

Deploy the statistics binary file under /database/userdata/fixedstats/action/local/ in the primary database of FEPCluster in the production environment.

```
kubectl cp <file-to-copy> \
<namespace>/<primary-pod-name>:/database/userdata/fixedstats/action/local/ -c fep-patroni
```

5.14.4.3 Specify File Names for FEPCluster Custom Resources

Specify the name of the deployed statistics binary file for the FEPCluster custom resource.

Example) Definition example of FEPCluster custom resource

```
spec:
  fep:
    fixedStats:
      schedule1: #Required, N is the identification number
        update: true
        fixSchedule: "0 20 * * 5"  #Required
        unfixSchedule: "0 23 * * 5" #Optional,  If not specified, do not unfreeze
        targetDb: mydatabase #Required
        file: file_name.dump
      endpoint:
        protocol: local
```

5.15 Scheduling of an Aggressive Freeze for Tuples (VACUUM FREEZE)

As an architectural limitation, PostgreSQL has a transaction wraparound problem that can cause the system to stop, and in order to avoid this, it is necessary to collect transaction IDs by freezing tuples.

Tuple freezing works by autovacuum or aggressive freeze for tuples (VACUUM FREEZE). However, with autovacuum alone, in a system where transactions are consumed at a high speed, you may encounter transaction ID wraparound problems. This is because autovacuum is slow to prevent collisions with applications and increases in system load. In such a system, it is necessary to perform aggressive freeze for tuples at an appropriate time.

To avoid such problems, it provides the ability to monitor transaction ID usage and to periodically perform an aggressive freeze for tuples. When an FEPCluster is built with monitoring enabled, monitoring of transaction ID usage is enabled by default.

If AlertManager reports Warning "PostgresqlTooManyTxidUsage", it is possible that autovacuum alone is not sufficient to freeze transaction IDs in time. In such cases, consider enabling the periodic feature of aggressive freeze for tuples.

To enable the periodic execution of aggressive freeze for tuples, set spec.fep.freezingTuples.enable in the FEPClsuter custom resource to true. The default for spec.fep.freezingTuples.enable is false.

The following items can be set.

- Schedule
- Execution period

The following example defines an aggressive freeze for tuples for two schedules: Sunday at 1:00 AM for three hours and Wednesday at 2:00 AM for one hour.

```
spec:
  fep:
    freezingTuples:
      enable: true
      schedule1:
        start: 0 1 * * 0
        executionTime: 10800
      schedule2:
        start: 0 2 * * 3
        executionTime: 3600
```

When scheduling an aggressive freeze for tuples, specify a time period with low business load when tuple freezing will have minimal impact.

Even after aggressive freeze for tuples is scheduled, continue to monitor the trend in transaction ID usage. If you determine that the freezing process is not keeping up with the schedule, you will need to reconsider the schedule. To collect the statistical information required for the review, enable the extended functionality for aggressive freeze for tuples using the following procedure.

1. Set the shared_preload_libraries parameter in postgresql.conf

```
shared_preload_libraries = 'pgx_stat_vacuum_freeze'
```

2. Execute the following SQL

```
SELECT datname FROM pg_database WHERE datallowconn = true;
```

   Execute the following CREATE EXTENSION for all databases to be output and define the extension.

```
CREATE EXTENSION pgx_stat_vacuum_freeze;
```

For information on tuning the allocation time for aggressive freeze for tuples, refer to the Fujitsu Enterprise Postgres Operation Guide.

5.16 pgvector Setup

With pgvector, vector storage and similarity searching become standard.

Execute CREATE EXTENSION on the database to use this feature, and then use the psql command to connect to the database to be set up.

```
mydatabase=# CREATE EXTENSION vector;
CREATE EXTENSION
```

To set up this feature with SUPERUSER, refer to "6.6.1 CREATE EXTENSION" and execute CREATE EXTENSION.

Note that OSS is named "pgvector", but the binaries and the extensions themselves are named "vector".

Chapter 6 Maintenance Operations

This chapter describes the maintenance operation after deploying the container.

6.1 Minor Version Upgrade

This section explains the steps for minor upgrade of a container image.

Before upgrading each container version, please refer to "Chapter 3 Operator Installation" to update your operators. Also, please refer to RedHat's ecosystem catalog for updated information on each container.

You can upgrade the container image by changing the parameters of the FEPCluster custom resource.

Parameter	Description
spec.fep.image.image	Update the FEP container image.
spec.fepChildCrVal.backup.image.image	Update the backup container image.
spec.fep.feputils.image	Update the container image responsible for cloud secret management feature.
spec.fep.remoteLogging.image	Update the container image that handles the Log Collection feature.

After changing the custom resource parameters, if you have multiple FEPPods, restart the Pods one by one from the replica server to upgrade the container image. Once all replica servers are upgraded, one of them will be promoted to master server. The old master server's Pods are then restarted and the container images are upgraded.

These rolling upgrades allow you to upgrade all container images with minimal disruption to your business.

Note

The upgrade process will cause an outage on the cluster for the duration to upgrade both Master and Sync Replica. If there is no Sync Replica in the cluster, the outage is limited to the length of time to upgrade the Master (or actually the failover time required to take another replica been promoted by patroni).

6.2 Cluster Master Switchover

You can switch a master instance to a replica instance in the event of a master instance performance failure or planned node maintenance.

Specify "switchover" for the action type of the FEPAction CR to update FEPAction CR.

Equivalent Kubernetes command: kubectl apply -f <new_spec>

The "switchover" action type requires the user to specify the name of the target cluster on which to perform the switchover. The args section is not needed for switchovers, as FEPAction internally identifies the pod to switch from and promotes a new master pod.

```
spec:
fepAction:
 type: switchover
 targetClusterName: new-fep
```

Refer to "FEPAction Custom Resource Parameters" in the Reference for more information on parameters.

6.3 Perform PITR and the Latest Backup Restore from Operator

It can be used to restore a database to a specific location due to an application failure or to prepare a duplicate database for production.

Restore process can restore data by creating a CR (FEPRestore CR) for the restore as follows:

oc create -f [Custom Resource Files]

Example)

```
$oc create -f config/samples/postgres_v1_restore.yaml
```

There are two methods of restoring: restoring data to an existing FEPCluster or restoring data to a new FEPCluster.

When restoring to an existing FEPCluster, information such as the FEPCluster name, IP address, and various settings remain the same.

If you restore to a new FEPCluster, the FEPCluster name is the one you specified in CR and the new IP address is also given. If the setting value is not specified, the new cluster will inherit the settings from the restore source cluster, but you can change the settings to create a new cluster by specifying them in CR.

6.3.1 Setting Item

Refer to "FEP Restore Custom Resource Parameters" in the Reference for the items to be set in a custom resource file.

6.3.2 After Restore

Switching connections to the new cluster

The restore creates a new FEPCluster. If necessary, you need to set up Pgpool-II and change the access point of the application to the new cluster or the new Pgpool-II.

Backup data of the destination cluster

PITR restores to the pre-restore time are not possible, because the backup of the destination cluster begins after the restore completes.

6.4 Major Version Upgrade

Describes the procedure for upgrading the major version of the operator and FEP container.

A major version upgrade of a FEP builds a new major version of the FEP in the same Namespace as the previous major version of the FEP. At this time, by defining the "spec.fepChildCrVal.upgrade" field in FEPClusterCR, the operator creates the upgrade execution container. The upgrade execution container uses the previous version of FEP Cluster specified in "spec.fepChildCrVal.upgrade.sourceCluster" as the data source FEPCluster and migrates the data to the newly created FEPCluster.

6.4.1 Pre-work on the Data Source FEP Cluster

Stop the running business application before executing the major version upgrade.

Next, edit "spec.fepChildCrVal.customPgHba" of the data source FEPCluster Custom Resource to allow the connection of the upgrade execution container.

The addresses that are allowed to connect are specified as follows:

```
<fep>-upgrade-pod.<fep>- upgrade-headless-svc.<namespace>.svc.cluster.local
```

<fep> specifies the name of the newly created FEPCluster Custom Resource.

The authentication method can be either trust/md5/cert.

Example of Editing a FEPCluster Custom Resource in a Data Source:

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: source-fep
  namespace: my-namespace
spec:
  fepChildCrVal:
    customPgHba: |
      host  all  all  destination-fep-upgrade-pod. destination-fep-upgrade-headless-svc. my-
```

```
namespace.svc.cluster.local  trust
...
```

6.4.2 Operator Upgrade

Describes the instructions for upgrading the operator.

Note

After an operator upgrade, any custom resource configuration changes you defined in the previous version are not reflected in the container.

6.4.2.1 Uninstalling the Old Operator

Uninstall the old operator.

Select "Uninstall Operator" from "Operators"> "Installed Operators"> "Fujitsu Enterprise Postgres <Old version> Operator"> Actions.

6.4.2.2 Installing a New Version of the Operator

Refer to "Chapter 3 Operator Installation" to install the new version of the operator.

6.4.3 Major Version Upgrade of FEP

6.4.3.1 Creating a New FEPCluster CR

Refer to the Reference to define a new major version of the FEPCluster custom resource. At this time, allow the running upgrade container to connect as you did in "6.4.1 Pre-work on the Data Source FEP Cluster".

In addition, a major version upgrade of FEP is performed by defining the "spec.fepChildCrVal.upgrade" field, as in the following example of defining a FEPCluster custom resource.

The upgrade execution container uses PV to store dump files retrieved from the FEPCluster of the data source.

If you have not enabled the automatic PV provisioning feature in your Kubernetes environment, create a PV for the upgrade in addition to the new PV for the FEPCluster before creating the FEPCluster custom resource.

Also, edit "spec.fepChildCrVal.customPgHba" to allow the connection of the upgrade execution container, as in "6.4.1 Pre-work on the Data Source FEP Cluster".

Example of Defining a FEPCluster Custom Resource to Perform an Upgrade:

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: destination-fep
  namespace: my-namespace
spec:
  fep:
    ...
  fepChildCrVal:
    upgrade
      sourceCluster: source-fep-cluster
      storage:
        size: 8Gi
    customPgHba: |
      host  all  all  destination-fep-upgrade-pod.destination-fep-upgrade-headless-svc.my-
namespace.svc.cluster.local  trust
    ...
```

FEPCluster Custom Resource Fields "spec.fepChildCrVal.upgrade"

Field	Default	Details
spec.fepChildCrVal.upgrade		Optional When this field is defined, a major version upgrade is performed. However, if spec.fepChildCrVal.restore is defined, the FEPCluster build stops.
spec.fepChildCrVal.upgrade.sourceCluster		Specify the FEPCluster CR name of the data migration source. Be sure to specify spec.fepChildCrVal.upgrade when defining it.
spec.fepChildCrVal.upgrade.mcSpec.limits	cpu: 200m memory: 300Mi	Optional Specify the maximum number of resources allocated to the upgrade execution container.
spec.fepChildCrVal.upgrade.mcSpec.requests	cpu: 100m memory: 200Mi	Optional Specify the lower limit of resources allocated to the upgrade execution container.
spec.fepChildCrVal.upgrade.image		Optional If omitted, the URL of the image is obtained from the operator container environment.
spec.fepChildCrVal.upgrade.imagePullPolicy	IfNotPresent	Optional Specify the pull policy for the container image. - Always - IfNotPresent - Never
spec.fepChildCrVal.upgrade.source.pgAdminTls.certificateName		Optional If the data source FEPCluster used "cert" as the authentication method for the Upgrade Execution Container, use the secret certificate that defines spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName for the data source FEPCluster. If the above parameter is not defined, it points to the Kubernetes TLS secret containing the certificate of the Postgres user "postgres" in the data source.

Field	Default	Details
		Refer to "4.8.2.1 Manual Certificate Management" for information about creating secrets.
spec.fepChildCrVal.upgrade.destination.pgAdminTls.certificateName		Optional If the newly created FEPCluster used the "cert" authentication method for the running upgrade container, use the secret certificate that defines the spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName of the newly created FEPCluster. If the above parameter is not defined, it points to the Kubernetes TLS secret containing the certificate of the newly created Postgres user "postgres". Refer to "4.8.2.1 Manual Certificate Management" for information about creating secrets.
spec.fepChildCrVal.upgrade.storage		Optional Defines storage for storing dump files.
spec.fepChildCrVal.upgrade.storage.storageClass		Optional If omitted, the default storage class of the operating environment will be used.
spec.fepChildCrVal.upgrade.storage.size	2Gi	Optional Specify the size of the storage to store the dump file.
spec.fepChildCrVal.upgrade.storage.accessModes	ReadWriteOnce	Optional Storage access mode for storing dump files As an array of access modes. e.g. [ReadWriteMany] If omitted, it is treated as [ReadWriteOnce].

Note

Connect to the database and run the following SQL to check the size of the database in advance:

```
$ SELECT pg_size_pretty(sum(pg_database_size(datname))) AS dbsize FROM pg_database;
```

Since the pg_dumpall command used in the upgrade execution container outputs the database data as an SQL command, the file actually created is as follows.

For example, the integer type 2147483647 is 4 bytes for database data.

However, this is 10 bytes because SQL commands output them as strings. Therefore, make sure that the storage (PV) for dump files has sufficient disk space.

6.4.3.2 Verifying FEP Major Upgrade Complete

If you migrate your data to the new FEPCluster and the FEP major version upgrade is successful, the following event will be output:

```
$ kubectl get event
LAST SEEN  TYPE    REASON               OBJECT                                          MESSAGE
164m       Normal  SuccessfulFepUpgrade fepupgrade/<Name of the new FEPClusterCR>  <namespace>,
Successfully FEP Upgrade
```

In addition, the following annotation will be added to YAML in FEPClusterCR:

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  annotations:
    FEPUpgradeDone: true
...
  name: destination-fep-cluster
  namespace: my-namespace
spec:
...
```

Note

When a major upgrade of FEP fails, an event similar to the following is output:

```
$ kubectl get event
LAST SEEN   TYPE     REASON            OBJECT                                          MESSAGE
164m        Warning  FailedFepUpgrade  fepupgrade/<Name of the new FEPClusterCR>  <namespace>, Error/
Failure in FEP Upgrade
```

Obtain the Kubernetes resource information listed in the OBJECT column, review the output messages, and then recreate the new FEPCluster custom resource.

```
$ kubectl describe fepupgrade/<Name of the new FEPClusterCR>
```

6.4.4 Updating Each Custom Resource

Describes the procedures for each custom resource used to operate the FEPCluster for the data source after the major FEP upgrade is complete.

After this process is complete, resume the suspended business applications.

6.4.4.1 Removing a FEPClusterCR for a Data Source

Delete the FEPCluster for the data source.

For the Openshift GUI console:

From "Operators" > "Installed Operators" > "Fujitsu Enterprise Postgres < New version > Operator" > "FEPCluster" > "FEPCluster name to delete" > Actions, select "Delete FEPCluster".

6.4.4.2 FEPPgpool2

Re-create FEPPgpool2 to match the version of the client with the version of the upgraded FEP.

6.4.4.3 FEPExporter Built in Standalone Mode

Edit the FEPExporter custom resource "spec.fepExporter.fepClusterList" to specify the new version of the FEPCluster custom resource.

Refer to "FEPExporter Custom Resource" in the Reference for more information about the parameters.

6.5 Assigned Resources for Operator Containers

The following resources are allocated by default to the operator containers provided by this product.

```
resources:
limits :
  cpu: 2
  memory: 1536Mi
requests :
  cpu: 500m
  memory: 768Mi
```

If there is only one FEPCluster custom resource managed by an operator, it can be operated with the resource assigned by default. However, when deploying and operating multiple FEPCluster custom resources, change the assigned resource of the operator container.

Note

If you have changed the resource, the resource value will revert to the default value after the operator version upgrade. Therefore, change the resource again after upgrading the operator.

6.5.1 How to Change Assigned Resources

Describes how to change the resources assigned to an operator container.

When updating resources assigned to an operator container, the operator container is recreated. At this time, the operation of already built containers such as FEPCluster will not stop.

How you change the allocated resources depends on how the operator was installed.

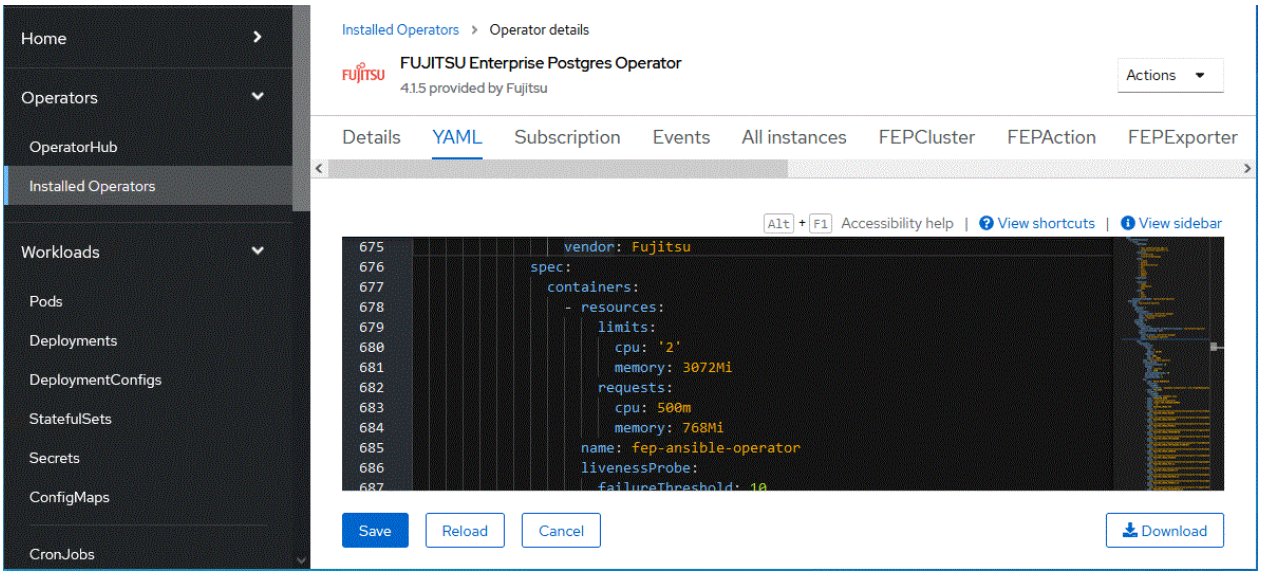
6.5.1.1 When installing using OperatorHub

If you are using an operator installed from OperatorHub To change the resources assigned to the operator container, edit the ClusterServiceVersion (CSV).

Editing the CSV "spec.install.spec.deployments[0].spec.template.spec.containers[0].resources" will recreate the operator container and apply the specified resources.

When editing CSV from the OCP GUI console

Click [Installed Operators] in the menu item under Operators and select the installed operator. On the [YAML] tab, edit the specified part of the allocation resource and click [Save].



When editing CSV from the CUI console using the OC client

Check the CSV name of the installed operator with the "oc get" command.

```
$ oc get csv
NAME                                          DISPLAY                                VERSION   REPLACES   PHASE
fujitsu-enterprise-postgres-operator.v4.1.5   Fujitsu Enterprise Postgres Operator
4.1.5                Succeeded
```

Edit the CSV with the "oc edit" command.

```
$ oc edit csv fujitsu-enterprise-postgres-operator.v4.1.5
```

6.5.1.2 When installing using Helm Chart or RancherUI

If the operator is installed using Helm Chart or RancherUI, edit the deployment of the operator container to change the resources assigned to the operator container.

Editing the Deployment's "spec.template.spec.containers[0].resources" will recreate the operator container and apply the specified resources.

Edit the Deployment "fep-ansible-operator" with the "kubectl edit" command.

```
$ kubectl get deployment fep-ansible-operator
NAME                   READY   UP-TO-DATE   AVAILABLE   AGE
fep-ansible-operator   1/1     1            1           2m10s

$ kubectl edit deployment fep-ansible-operator
```

6.6 Using SUPERUSER Privilege

6.6.1 CREATE EXTENSION

When executing the CREATE EXTENSION command to install external extensions for PostgreSQL, there are extensions that can only be installed by SUPERUSER. To install such extensions we make use of the FEPAction custom resource.

By specifying "create_extension" in spec.fepAction.type of FEPAction custom resource, CREATE EXTENSION can be executed for the specified FEPCluster container.

Please refer to the Reference for how to use.

6.6.2 Change Password of SUPERUSER

To update the password of SUPERUSER "postgres", specify update_admin_password for fepActionType of the FEPAction custom resource.

Recreate the password with a random value and update it.

Please refer to the Reference for how to use.

6.6.3 Using SUPERUSER

If SUPERUSER privileges are required for database operation, you can obtain the password for SUPERUSER "postgres" by following the steps below.

1. Get the base64-encoded password from the Secret with the same name as the FEPCluster custom resource name.

   Example) When the FEPCluster custom resource name is new-fep

```
$ kubectl get -o yaml secret new-fep | grep PG_ADMIN_PASSWORD
  PG_ADMIN_PASSWORD: YWRtaW4tcGFzc3dvcmQ=
```

2. Decode the obtained password.

```
$ echo YWRtaW4tcGFzc3dvcmQ= | base64 -d
admin-password
```

Note

In order to prevent SUPERUSER from being used by a third party, please set Kubernetes Role permissions to the Secret so that only the database administrator can refer it.

Chapter 7 Abnormality

This chapter describes the actions to take when an error occurs in the database or an application, while FEP is operating.

Depending on the type of error, recover from the backed-up material, reserve capacity, check the operator log, and check the FEP log.

7.1 Handling of Data Abnormalities

Recover the database cluster from the backup immediately prior to failure in any of the following cases:

- A hardware failure occurs on the data storage disk or the backup data storage disk.
- If the data on the disk is logically corrupted and the database does not work correctly
- Data corruption caused by user error

Refer to "6.3 Perform PITR and the Latest Backup Restore from Operator" for restore instructions.

7.2 Handling when the Capacity of the Data Storage Destination or Transaction Log Storage Destination is Insufficient

If you run out of space in the data storage location, first check if there are any unnecessary files on the disk, and then delete them so that you can continue working.

If deleting unnecessary files does not solve the problem, you may need to migrate the data to a larger disk.

Use a backup restore to migrate data.

You can use the FEPRestore custom resource to build a new FEPCluster and migrate data. Refer to "FEPRestore Custom Resource Parameters" in the Reference.

7.3 What to do when the Capacity of the Backup Data Storage Area is Insufficient

If you run out of space in the backup data destination, first check the disk for unnecessary files, and then delete the unnecessary files. Or reduce the backup retention generation.

By specifying backup_expire in spec.fepAction.type of the FEPAction custom resource, you can reduce the number of backup generations saved. Refer to "FEPAction Custom Resource Parameters" in the Reference for details.

7.4 Handling Access Abnormalities When Instance Shutdown Fails

If an instance fails to start or stop, refer to the Operator log and the FEP log to determine the cause.

For checking the operator log and the FEP log, refer to "7.5 Collection of Failure Investigation Information".

7.5 Collection of Failure Investigation Information

If the cause of the trouble that occurred during the construction or operation of the environment is not identified, information for the initial investigation is collected.

I will explain how to collect information for the initial investigation.

- Product log
- Operator log

Product log

FEP log

Get into the container and collect the log.

The log location is specified by log_directory in the custom resource FEP Clusterspec.startupValues.customPgParam parameter. The default is/database/log.

Pgpool-II log

Get into the container and collect the log.

The log location is/var/log/pgpool/pool.log.

Operator log

Check the operator log as follows.

Verification Example

```
$oc get po
NAME                                    READY   STATUS    RESTARTS   AGE
fep-ansible-operator-7dc5fd9bf7-4 smzk   1/1     Running   0          20m
```

How to check the log

```
$oc logs pod fep-ansible-operator-7dc5fd9bf7-4 smzk -c manager
```

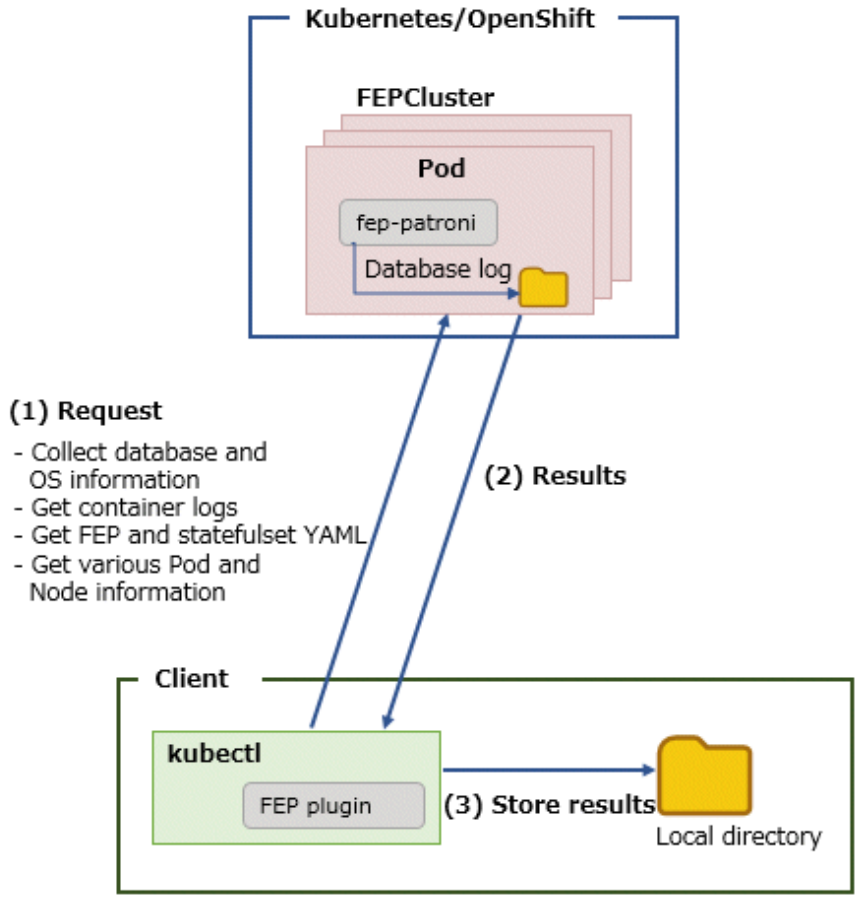
The log will be output to the console. Please check the file output by redirection.

7.6 Log Collection Tool

The Log Collection Tool is a client tool for the FEP container environment, that allows logs, and other useful environment and statistical information, to be retrieved from FEP containers running on a specified FEP cluster on Kubernetes/OpenShift. The collected information is stored locally in a filesystem directory on the client machine, for subsequent analysis.

The purpose of the tool is to obtain helpful diagnostic and troubleshooting information, in the case that an FEP containers is, for example, non-responsive, has terminated abnormally or exhibits abnormal performance.

The Log Collection Tool collects container-specific information as well as database and OS information.



7.6.1 Logs Collected

The following logs and information are collected by the Log Collection Tool:

Type of log(s)	Source
OS information and statistics, FEP catalog contents, database logs	fep-patroni (FEP server) container, in each pod
FEP yaml	FEP cluster
Statefulset yaml	FEP cluster
Container logs	Log from each FEP-related container running in the FEP cluster ("kubectl logs ⋯" output)
Detailed pod information, for the namespace	"kubectl get pods -n <namespace> -o wide" output
Detailed node information, for the Kubernetes cluster	"kubectl get nodes -o wide" output
Pod CPU and memory utilization, for the namespace	"kubectl top pod -n <namespace>" output
Node CPU, memory, and resource utilization, for the Kubernetes cluster	"kubectl top node" output

7.6.2 Location and Dependencies

The Log Collection Tool is the kubectl plugin "kubectl-fep". Add Log Collection Tool as subcommands to kubectl to extend the command line syntax.

Log Collection Tool are stored in the operator's Pod. Install it on the Kubernetes client machine using the command below. <destination-path> specifies the storage location of the plugin.

```
1.Check the OperatorPod name with the command below.
$ kubectl get pods --selector=name=fep-ansible-operator -o name
pod/fep-ansible-operator-5985969857-ldj2x

2. Copy the plugin.
# Linux Client
$ kubectl cp fep-ansible-operator-5985969857-ldj2x:/opt/fepopr-kubectl-plugin/kubectl-fep
<destination-path>/kubectl-fep

# Windows Client
$ kubectl cp fep-ansible-operator-5985969857-ldj2x:/opt/fepopr-kubectl-plugin/kubectl-fep.ps1
<destination-path>/kubectl-fep.ps1
```

The Log Collection Tool kubectl plugin is installed on client machines along with the kubectl/oc command for remote access to the target Kubernetes/Openshift environment. Specify the directory where both the kubectl plugin and the kubectl/oc command are stored in the PATH environment variable. Also, set appropriate execution permissions for the plugin.

On a client machine, the kubectl/oc command supports different methods of authentication/login to the Kubernetes Cluster (e.g., username + password, TLS client certificates, OAuth token) that allow different users to execute the command, provided they have suitable privileges to the Kubernetes Cluster.

The diagnostic logs collected by the Log Collection Tool are stored locally on the filesystem on the client.

7.6.3 User Interface

The Log Collection Tool is invoked through the kubectl plugin command-line interface.

7.6.3.1 Command-Line Syntax

The command-line syntax of the Log Collection Tool is shown below.

The Log Collection Tool corresponds to the "cluster logs" subcommand of the FEP kubectl plugin.

There is a companion "cluster list" subcommand which displays the list of FEP clusters in the specified namespace.

Usage:

```
kubectl fep cluster logs -n <NAMESPACE> [-o <OUTPUT-DIRECTORY>] [-d <DATABASE>] <FEP-CLUSTER>
kubectl fep cluster list -n <NAMESPACE>
```

Note that for the Openshift Container Platform, the Openshift "oc" command can be used as an alternative to "kubectl", as it has the same features as the kubectl binary, but it has been further extended to natively support the OCP features.

Windows considerations

The command-line syntax for invoking the FEP kubectl plugin (but not supplying its command parameters) may differ between Operating Systems, due to the type of executable that the plugin is implemented as.

For example, under Windows, since the FEP kubectl plugin is implemented as a Powershell script the plugin needs to be invoked directly, rather than indirectly from kubectl/oc.

From a Command Prompt:

Powershell 5.1:

```
powershell kubectl-fep …
```

Powershell 7+:

```
pwsh -command kubectl-fep …
```

From a Powershell Window:

```
kubectl-fep …
```

An additional consideration is the current Powershell "ExecutionPolicy" in effect, which may block Powershell scripts from being run for security reasons, if they are not signed. In this case, since the FEP kubectl plugin is a trusted script, the ExecutionPolicy can be bypassed for the script execution, using the following syntax:

Powershell 5.1:

```
powershell -executionpolicy bypass kubectl-fep …
```

Powershell 7+:

```
pwsh -executionpolicy bypass -command kubectl-fep …
```

7.6.4 Log Collection Directory

7.6.4.1 Command-Line Specification

The Log Collection Tool output directory may optionally be specified using the -o flag, and if the directory exists, it must be empty. If the output directory is not specified, the following default output directory is used:

Linux:

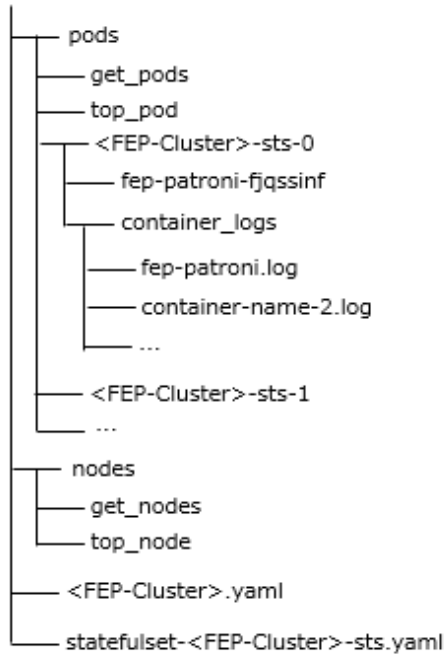
```
/tmp/fep_cluster_logs_<namespace>_<cluster>_<date>_<time>
```

Windows:

```
%TEMP%\fep_cluster_logs_<namespace>_<cluster>_<date>_<time>
```

7.6.4.2 Directory Structure

The structure of the output directory of the Log collection Tool is as follows.



Appendix A Quantitative Values and Limitations

A.1 Quantitative Values

Refer to the Fujitsu Enterprise Postgres Installation and Setup Guide for Server.

A.2 Limitations

Note

If you log in to a container and edit the configuration file directly, restarting the container may undo your changes.

If you want to change the settings, modify the custom resource files as described in "5.2 Configuration Change" and reapply. Depending on the parameters to be changed, the container may be redeployed. Refer to "5.2 Configuration Change" for details of the parameters.

Unavailable FEP features

Since FEP server container is based on other components (like UBI and Patroni), there are certain limitations that doesn't allow it to be 100% functionally capable to VM based server instance. The known limitations are as below.

No	Limitation	Reason for Limitation	Description
1	Crypto Express cards are not supported	IBM LinuxOne doesn't support CryptoExpress cards in Openshift container platform at this stage.	FEP TDEz extension cannot be used on LinuxOne Openshift environment. However, User can still use TDE on both LinuxOne Openshift environment as well as Azure (x86) Openshift environment.

Fixed parameter

Some parameters cannot be changed. Refer to "2.3.5.2 Parameters that cannot be Set".

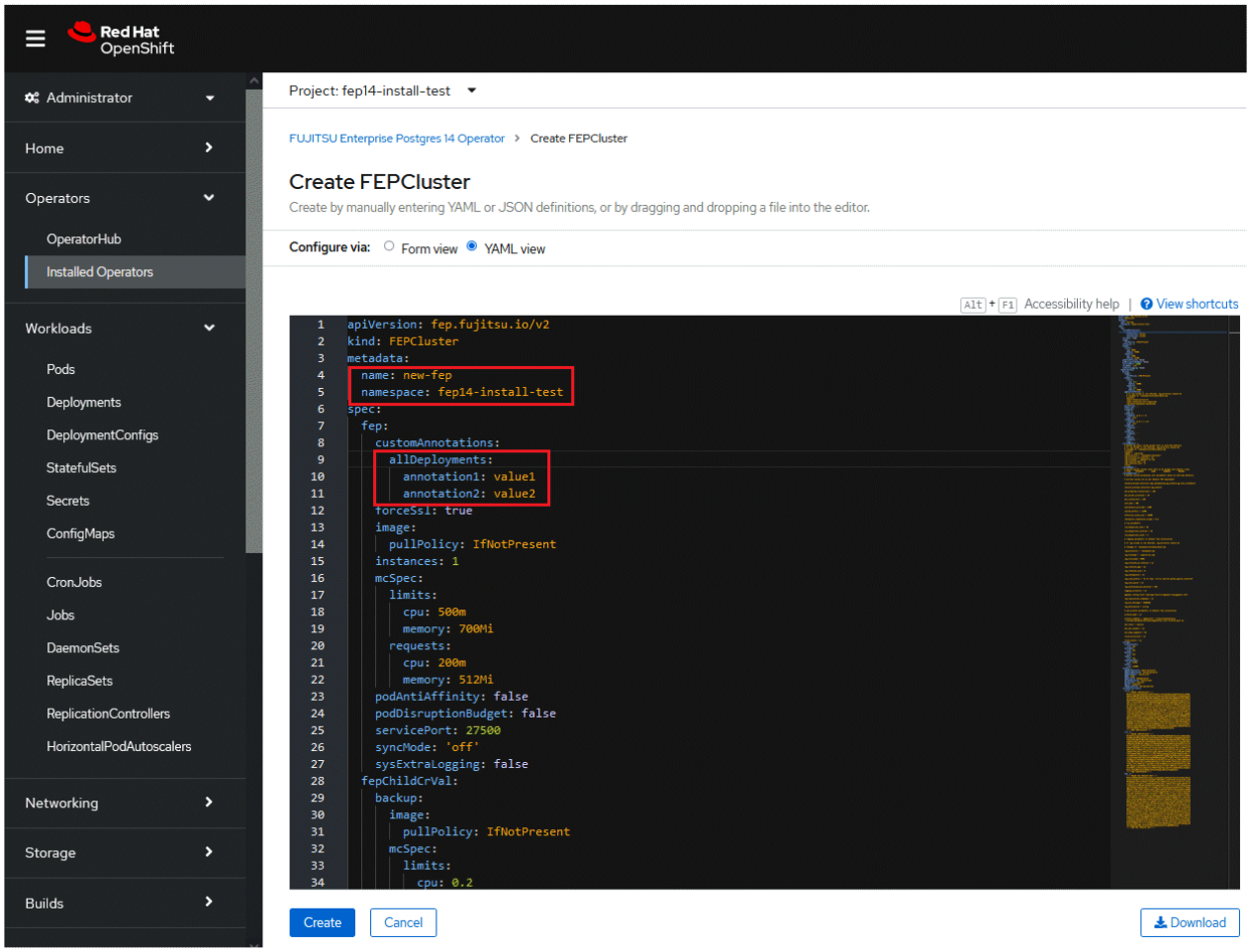
FEP features that needs to be set when using

Refer to "2.3.7 FEP Unique Feature Enabled by Default".

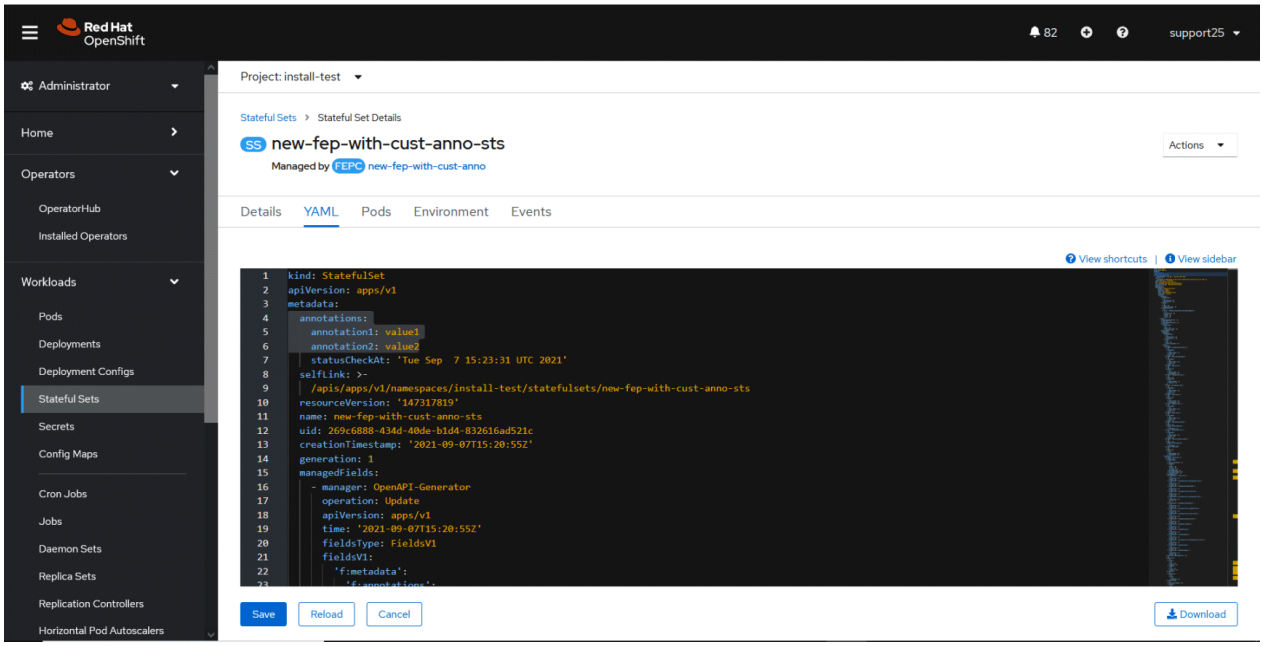
Appendix B Adding Custom Annotations to FEPCluster Pods using Operator

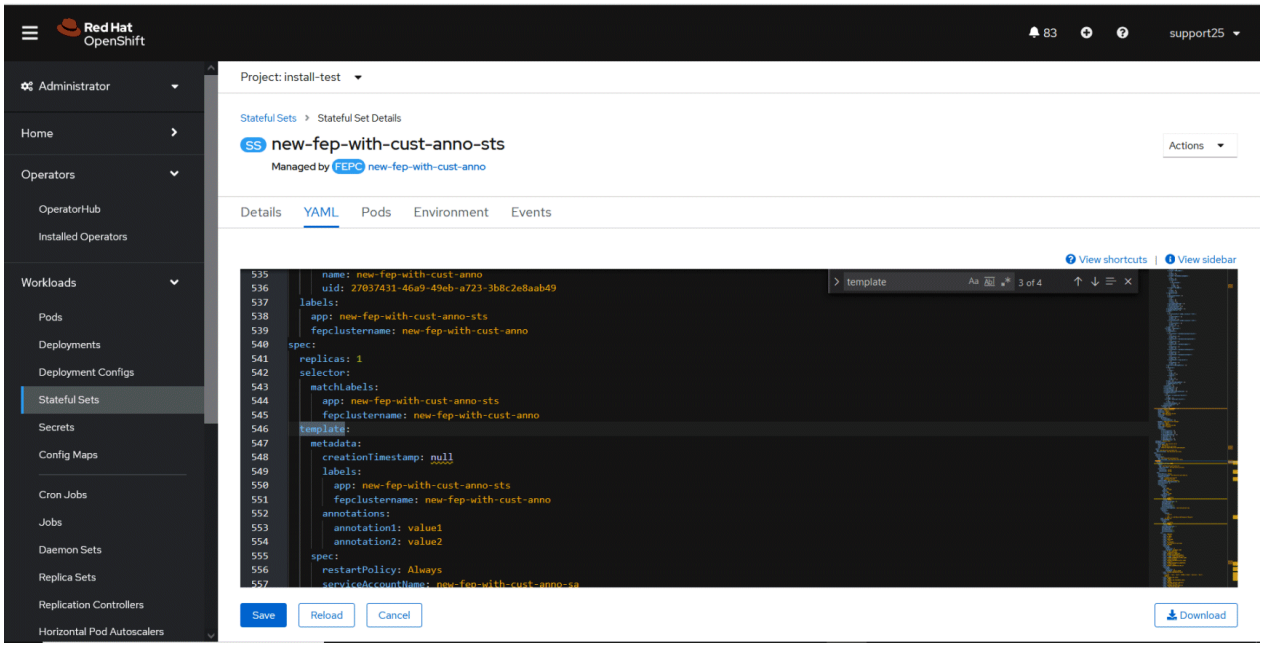
This section describes instructions for adding custom annotations to a FEPCluster pod.

1. In YAML view of the Create FEPCluster section, add custom annoatations as below and then click on Create.



2. Both the Statefulset and its resulting pods will be annotated with your prvided annotations: archivewalVol and backupVol must be ReadWriteMany.





Appendix C Utilize Shared Storage

Explains how to build a FEPCluster when using shared storage.

Use a disk where PV accessModes can specify ReadWriteMany.

This chapter shows an example of using NFS as PV in static provisioning.

C.1 Creating a StorageClass

Create a StorageClass.

In the OCP WebGUI screen, click "StorageClass" in the main menu "Storage", then press "Create Storage Class" > "Edit YAML" and edit YAML to create the StorageClass.

If you are using the CLI, create a yaml file and create a StorageClass with the following command:

```
$ oc create -f <file_name>.yaml
```

YAML definitions are created with reference to the following samples.

Example)

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: < StorageClass Name >
provisioner: kubernetes.io/no-provisioner
reclaimPolicy: Delete
volumeBindingMode: WaitForFirstConsumer
```

C.2 Creating a PersistentVolume

Create as many PersistentVolumes (PV) as you need.

On the Web GUI screen, click "PersistentVolumes" in the main menu "Storage", click "Create PersistentVolume", and edit YAML to create PV.

If you are using the CLI, create a yaml file and create a PV using the following command:

```
$ oc create -f <file_name>.yaml
```

YAML definitions are created with reference to the following samples.

The StorageClass name specifies the StorageClass created in "C.1 Creating a StorageClass".

Assign a different NFS directory for each PV.

In addition, accessModes is ReadWriteMany.

Example)

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: < PV name >
spec:
  capacity:
    storage: < Capacity Required ex.8Gi >
  accessModes:
  - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  mountOptions:
    - hard
  nfs:
```

```
    path: < NFS directory path (Assign a different directory for each PV) ex. /nfs/pv >
    server: < IP address of the NFS server ex. 192.168.1.10>
  storageClassName: < StorageClass name created in "C.1 Creating a StorageClass">
```

C.3 Creating FEPCluster

Specifies that ReadWriteMany PV is used in the YAML definition in step 4 of "4.1 Deploying FEPCluster using Operator".

In spec.fepChildCRVal.storage, specify the StorageClass and AccessModes of the PV created in "C.2 Creating a PersistentVolume".

The "spec.fepChildCRVal.storage.<Volume Type>.size" should be less than or equal to the PV allocated.

Example) Using PV created by archivewalVol and backupVol

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: t3-fep
spec:
  ~ Suppress ~
  fepChildCrVal:
    storage:
      archivewalVol:
        size: < Capacity Required ex. 8Gi >
        storageClass: <StorageClass name created in C.1 Creating a StorageClass" >
        accessModes:
        - "ReadWriteMany"
      backupVol:
        size: < Capacity Required ex. 8Gi >
        storageClass: <StorageClass name created in C.1 Creating a StorageClass" >
        accessModes:
        - "ReadWriteMany"
  ~ Suppress ~
```

Appendix D Key Management System Available for Transparent Data Encryption

Describes the key management system available for transparent data encryption.

D.1 KMIP Server

Refer to "To Connect to a key Management System Using the KMIP Protocol" in the Fujitsu Enterprise Postgres Installation and Setup Guide for Server for KMIP server requirements.

D.2 AWS Key Management Service

D.2.1 Available Services

By using the AWS KMS adapter, you can use encryption keys on the Key Management Service (hereafter referred to as AWS KMS) provided by AWS. There is no region restriction as long as it is a region supported by AWS KMS.

D.2.2 Available AWS KMS Keys

The KMS key's key spec must be "symmetric". "asymmetric" keys cannot be used. Also, the KMS key usage must be ENCRYPT_DECRYPT.

D.2.3 Required Privileges

For the KMS key to be used, the following operations must be permitted for the user accessing AWS KMS.

- Encrypt
- Decrypt
- DescribeKey

D.2.4 Key ID

The following can be specified as key IDs for the TDE key management system linkage feature.

- Key ARN

D.3 Azure Key Management Service

D.3.1 Available Services

Accessible via Azure's Key Vault API using the Azure KMS Adapter, and a key management service is available that allows you to use symmetric keys.

Refer to below for key management services for which symmetric keys are available.

https://learn.microsoft.com/en-us/azure/key-vault/keys/about-keys#key-types-and-protection-methods

Refer to below for Azure key management services.

https://learn.microsoft.com/en-us/azure/security/fundamentals/key-management#azure-key-management-services

D.3.2 Available Keys

A symmetric key is available.

D.3.3 Available Algorithms

The following algorithms are available during encryption/decryption operations.

- A256GCM

D.3.4 Key Operation

For the key to be used, the following operations must be permitted for the user who accesses Azure's key management service.

- encrypt
- decrypt
- get

D.3.5 Key ID

The following can be specified as key IDs for the TDE key management system linkage feature.

- Key object identifier

D.3.6 Sign In

Sign in to Azure using your service principal. You will need your application ID, tenant ID, and credentials to sign in.

See

Refer to below for service principals.

https://learn.microsoft.com/en-us/cli/azure/create-an-azure-service-principal-azure-cli#4-sign-in-using-a-service-principal

Appendix E Fluent Bit Integration Using Custom Secret

Describes the Fluent Bit integration using custom secret.

E.1 Create Custom Secret

E.1.1 Sample fluent-bit.yaml template

fluent-bit.yaml

```
---
env:
  flush_interval: 5                  # Interval in seconds for flushing logs to outputs
  tail_path: /databaselogs/*.csv      # File path pattern for application logs (CSV format)
  audit_tail_path: /databaselogs/audit/*.log   # File path pattern for audit logs
  tail_db_path: /databaselogs/postgreslog.db   # Path to DB tracking log read positions for app logs
  audit_tail_db_path: /databaselogs/auditlog.db  # Path to DB tracking log read positions for audit
logs
  skip_long_lines: on                 # Skip lines longer than the buffer limit
  refresh_interval: 60                # Interval in seconds to refresh and check log files
  read_from_head: true                # Start reading logs from the beginning of the file
  multiline: on                       # Enable processing of multiline log entries
  parser_firstline: firstline_app_parser    # Parser for the start of multiline entries in app logs
  parser_audit_firstline: firstline_audit_parser  # Parser for the start of multiline entries in audit
logs
  rotate_wait: 20                     # Time to wait before completing log file rotation
  storage_type: memory                # Type of storage used for logs, here using memory

service:
  flush: ${flush_interval}            # Sets the flush interval using the `flush_interval` variable
  daemon: off                         # Runs Fluent Bit in the foreground
  log_level: info                     # Sets log verbosity to 'info' level
  parsers_file: parsers.conf          # Specifies the file containing custom parsers
  http_server: On                     # Enables the built-in HTTP server for metrics and health checks
  http_listen: 0.0.0.0                # Configures the HTTP server to listen on all network interfaces
  http_port: 2020                     # Sets the port for the HTTP server
  hot_reload: On                      # Enables hot reload of the configuration without restarting

pipeline:
  inputs:
    - name: tail                      # Uses the `tail` plugin to monitor and read log files
      db: ${tail_db_path}             # Database file to track reading positions of log files
      tag: app-*                      # Tags logs with `app-*` for filtering/routing later
      path: ${tail_path}              # Path to the log files using `tail_path` variable
      Skip_Long_Lines: ${skip_long_lines}  # Skips lines longer than the buffer can handle
      Refresh_Interval: ${refresh_interval}  # Interval to refresh monitored files for new data
      Read_from_head: ${read_from_head}  # Reads logs from the start of the file
      Multiline: ${multiline}          # Enables support for multiline log entries
      Parser_Firstline: ${parser_firstline}  # Parser for identifying the start of multiline logs
      rotate_wait: ${rotate_wait}      # Time to wait before considering log file rotation complete

  filters:
    - name: parser                    # Applies a parser to the logs matching `app-*` tag
      match: app-*                    # Only processes logs tagged with `app-*`
      key_name: log                   # Specifies the key containing the log message to parse
      parser: app_parser              # Parser to use for the log messages
    - name: rewrite_tag               # Modifies log tags for re-emission
      match: app-*                    # Matches logs tagged with `app-*`
      rule: '$log_time "[^,]+"  appcsv-.$TAG[1].$TAG[2].$TAG[3].$TAG[4] false'  # Rule for renaming
tags
```

```
      emitter_name: re_emitted_appcsv # Name of the emitter for re-emitted logs
     emitter_storage.type: ${storage_type}  # Storage type for re-emitted logs, using `storage_type`
variable

  outputs:
    - name: azure_blob                # Azure Blob Storage Output plugin name
      match: "*"                      # Matches all logs for output to Azure Blob Storage
      account_name: <azure_storage_account>      # Azure Storage account name
      shared_key: <shared_key>     # Shared key for authentication with Azure
      path: < multiline_parser>         # Path within Blob Storage where logs are stored
      container_name: <fluentbit-aug28>  # Container name in Azure Blob Storage
      auto_create_container: on       # Automatically create the container if it doesn't exist
      tls: on                         # Enables TLS encryption for secure communication
      net.dns.resolver: legacy        # Uses legacy DNS resolver

    - name: es                         # Elasticsearch Output plugin name
     match: "*"                      # Matches all logs for this output plugin; the asterisk (*) means
all tags
     host: <Elasticsearch_Host>        # The hostname or IP address of the Elasticsearch server
     port: <Elasticsearch_Port>        # The port on which Elasticsearch is running (usually 9200)
     index: <fluentbit>              # The name of the index where logs will be stored in Elasticsearch
     type: _doc                   # The document type in Elasticsearch; _doc is the preferred type for
ES 7.x and  later
     http_user: <HTTP_User>           # Username for HTTP Basic Authentication to access Elasticsearch
     http_passwd: <HTTP_Password>      # Password for HTTP Basic Authentication
     tls: On                         # Enables TLS (Transport Layer Security) for secure communication
with Elasticsearch
     tls.verify: Off  # Disables verification of the Elasticsearch server's TLS certificate (not
recommended for production)
     Suppress_Type_Name: On  # Suppresses adding the '_type' field to documents (important for
Elasticsearch 7.x and later)

# Note:- for this integration please refer additional steps in section - "E.3 Fluent Bit configuration
for Prometheus exporter"
    - name : prometheus_exporter  #prometheus exporter Output plugin name
       match: "*"  # Applies to all logs
       host: 0.0.0.0  # Listens on all network interfaces
       port: 24231  # Port for receiving logs
       metrics_path: /metrics  # Endpoint for metrics exposure

    - name: stdout                    # Outputs logs to standard output (stdout)
      match: "*"                      # Matches all logs for output to stdout
```

See

Refer to below for Fluent Bit configuration..

https://docs.fluentbit.io/manual

E.1.2 Sample parsers.conf template

```
[PARSER]
    Name        firstline_app_parser
    Format      regex
    Regex       ^(?<log>[0-9]{4}-[0-9]{2}-[0-9]{2} [0-9]{2}:[0-9]{2}:[0-9]{2}\.[0-9]{3} [A-Z]+,(.*))
    # This regex captures both the timestamp and the rest of the line.
[PARSER]
    Name        firstline_audit_parser
    Format      regex
    Regex       ^(?<log>(?:AUDIT:\s(?:SESSION|OBJECT))(.*))
  # This regex captures both audit headers "AUDIT: (SESSION|OBJECT)" and the rest of the line.
```

E.1.3 Encode the fluent-bit.yaml content

```
cat fluent-bit.yaml | base64 -w 0
```

E.1.4 Encode the parsers.conf content

```
cat parsers.conf | base64 -w 0
```

E.1.5 Create the custom secret

The custom secret must contain the required key fluent-bit.yaml and optional included keys, e.g., parsers.conf as shown below.

Sample Secret YAML Definition:

```
apiVersion: v1
kind: Secret
metadata:
  name: fluent-bit-secret       # Name of the secret
  namespace: your-namespace     # Namespace where the secret will be created
type: Opaque
data:
  fluent-bit.yaml: <base64_encoded_content>
  parsers.conf: <base64_encoded_content>
```

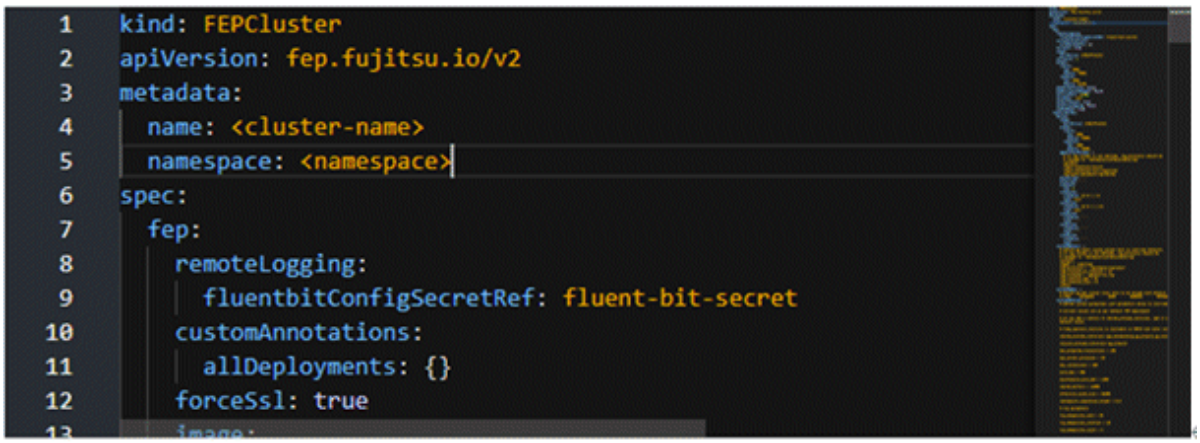
Example:

Assuming the base64 encoded content of the fluent-bit.yaml file is c29tZS1iYXNlNjQtZW5jb2RlZC1jb250ZW50, the secret would look like this:

```
apiVersion: v1
kind: Secret
metadata:
  name: fluent-bit-secret       # Name of the secret
  namespace: your-namespace     # Namespace where the secret will be created
type: Opaque
data:
  fluent-bit.yaml: c29tZS1iYXNlNjQtZW5jb2RlZC1jb250ZW50  # Replace with your actual base64 encoded
content
parsers.conf: a29tZS1iYXNlNjQtZW5jb2RlZC1jb250ZW51  # Replace with your actual base64 encoded
content
```

E.2 Reference to the secret in FEPClusterCR

spec.fep.remoteLogging.fluentbitConfigSecretRef as shown below:



E.2.1 Reload Fluent Bit Configuration

After saving the secret in FEPClusterCR, reload Fluent Bit container to apply the changes.

Apply the changes in one of the following ways:

- Create a FEPAction with type fluentbit_reload as shown below:

```yaml
kind: FEPAction
apiVersion: fep.fujitsu.io/v1
metadata:
  name: <fep-action-name>
  namespace: <namespace>
spec:
  fepAction:
    args:
    type: fluentbit_reload
  sysExtraEvent: true
  sysExtraLogging: false
  targetClusterName: <target-cluster-name>
```

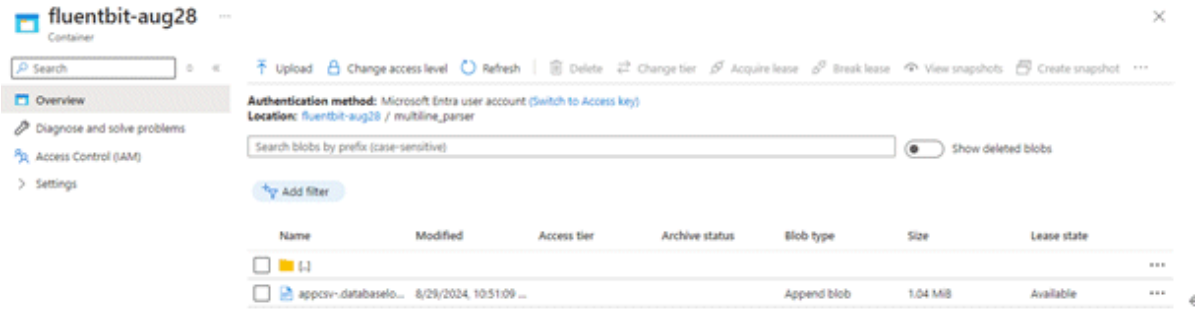
- Use the below command and execute in fluent-bit container if you wish to reload the Fluent Bit's configuration manually:

```
curl -s -X POST -d '{}' localhost:2020/api/v2/reload
```

E.2.2 Confirm Log Ingestion in Azure Blob

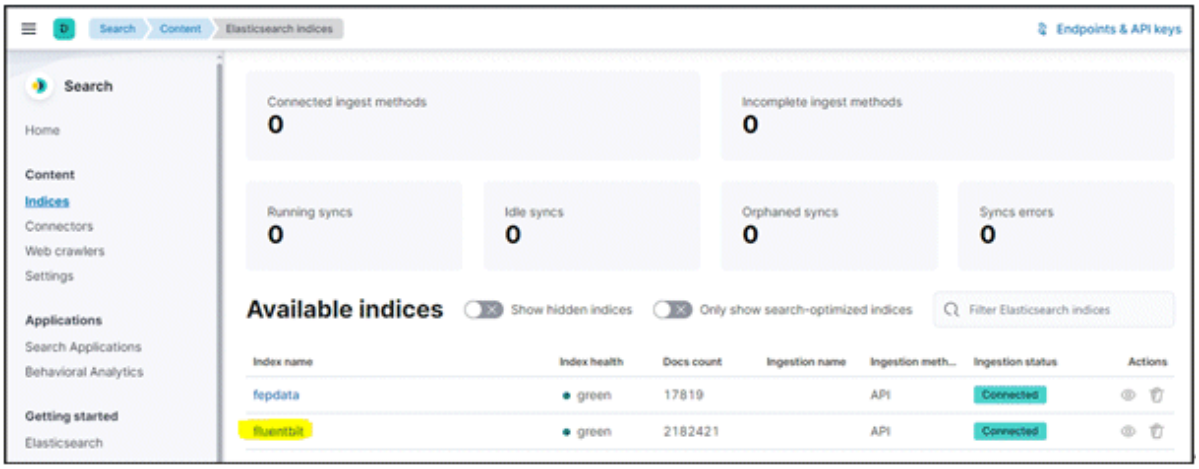
After reloading the configuration, check your Azure Blob container to confirm that logs are being ingested properly. You can use the Azure portal or Azure Storage Explorer for this purpose.

Based on the Output Plugin's specified path, the logs were stored in the respective directory.



E.2.3 Confirm Log Ingestion in Elasticsearch

Once Fluent Bit has been reloaded, verify that data is being correctly ingested into your Elasticsearch index.



E.3 Fluent Bit configuration for Prometheus exporter

To expose Fluent Bit metrics to Prometheus, you'll need to add the prometheus_exporter Output plugin and the appropriate Filters to the Fluent-bit.yaml configuration file and Create a custom secret. For more information, refer to the Fluent Bit Manual.

E.3.1 Create a Service

To allow Prometheus to scrape metrics from Fluent Bit, you need to create a Service in OpenShift.

Sample YAML file for create a Service:

```
apiVersion: v1
kind: Service
metadata:
  name: <service-name>
  namespace: <namespace>
  labels:
    app: <fluent-bit-pod-name>
    fepclustername: <fepcluster-name>
spec:
  ports:
    - name: prometheus-metrics
      port: 80
      targetPort: 24231
      protocol: TCP
  selector:
    app: <fluent-bit-pod-name>
```

- name: The name of the Service.
- namespace: The namespace in which this Service is deployed.
- ports: Maps the Service port (`80`) to the target port (`24231`) used by Fluent Bit's Prometheus Exporter.

E.3.2 Create a ServiceMonitor

To automatically configure Prometheus to scrape metrics from the Fluent Bit service, you'll need to create a `ServiceMonitor` resource. This is particularly useful in Kubernetes environments that use the Prometheus Operator.

Sample YAML file for Create a ServiceMonitor:

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: <serviceMonitor-name>
  namespace: <namespace>
```

```
    labels:
      release: prometheus
  spec:
    selector:
      matchLabels:
        app: <fluent-bit-pod-name>
        fepclustername: <fepcluster-name>
    endpoints:
      - port: prometheus-metrics
        path: /metrics
        interval: 15s
    namespaceSelector:
      matchNames: <namespace>
```

- name: The name of the ServiceMonitor.
- namespace: The namespace in which the ServiceMonitor is deployed.
- selector: The labels used to select the Service to be monitored, matching the labels used in the Fluent Bit service.
- endpoints: Specifies the port and path for the Prometheus metrics, along with the scrape interval.
- namespaceSelector: Specifies the namespace(s) in which the ServiceMonitor should look for Services to monitor.

E.3.3 Confirm Log Ingestion in Prometheus

Once Prometheus starts scraping Fluent Bit metrics, you can view them in OpenShift's Observe-Metrics tab.

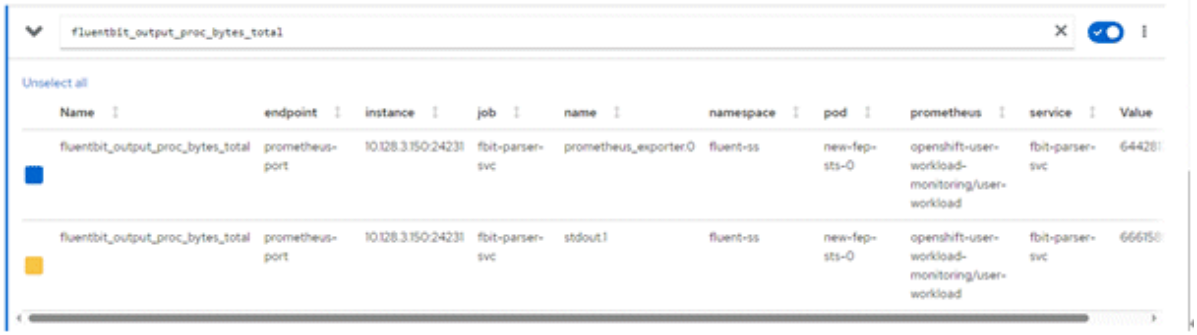
1. Access OpenShift Console

   Navigate to the OpenShift Console and go to the "Observe" section.

2. Explore Fluent Bit Metrics

   Use Prometheus queries to explore Fluent Bit metrics, for example:

   ```
   fluentbit_output_proc_bytes_total
   ```



E.4 Fluent Bit configuration for AWS CloudWatch

Integrating Fluent Bit with AWS CloudWatch allows you to efficiently collect, process, and forward logs from your FEP cluster to CloudWatch Logs. Please follow the detailed integration process below.

E.4.1 Create AWS credentials Secret

To allow Prometheus to scrape metrics from Fluent Bit, you need to create a Service in OpenShift.

Create a secret which include the config and credentials keys to export the environment variables as local variables for authentication to fep-logging-fluent-bit container, such as AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY.

Name of secret that contains the credential to AWS service. The credential is stored in Shared Configuration File and Credentials File.

The name of the Shared Configuration File must be "config" and the name of the Credentials File must be "credentials"

Sample template to create a config file

```
[default]  # Section for default AWS CLI settings that apply globally
region = <aws_region>  # The AWS region to be used for requests
output = json  # Format for the output of AWS CLI commands (e.g., json, text, table)
max_attempts = 1  # Maximum number of retry attempts for AWS CLI commands in case of failure

[profile svc-fep-tdedev0]  # Section for specific profile settings for svc-fep-tdedev0
region = ap-northeast-1  # The AWS region to be used for this specific profile
output = json  # Output format for commands executed under this profile (e.g., json)
```

Sample template to create a credentials file

```
[default]  # Section for the default profile
aws_access_key_id =  < aws_access_key_id > # Your AWS Access Key ID (to be filled in)
aws_secret_access_key = < aws_secret_access_key > # Your AWS Secret Access Key (to be filled in)

[svc-fep-tdedev0]  # Section for the svc-fep-tdedev0 profile
aws_access_key_id = < aws_access_key_id >  # Your AWS Access Key ID for this profile (to be filled
in)
```

E.4.2 Encode the config file content

```
cat config| base64 -w 0
```

E.4.3 Encode the credentials content

```
cat credentials | base64 -w 0
```

E.4.4 Create a secret using base64 encoded content

Sample aws-credentials secret YAML template

```
kind: Secret
apiVersion: v1
metadata:
  name: aws-credentials
  namespace: <namespace>
data:
  config: <base64_encoded_content>
  credentials: <base64_encoded_content>
type: Opaque
```

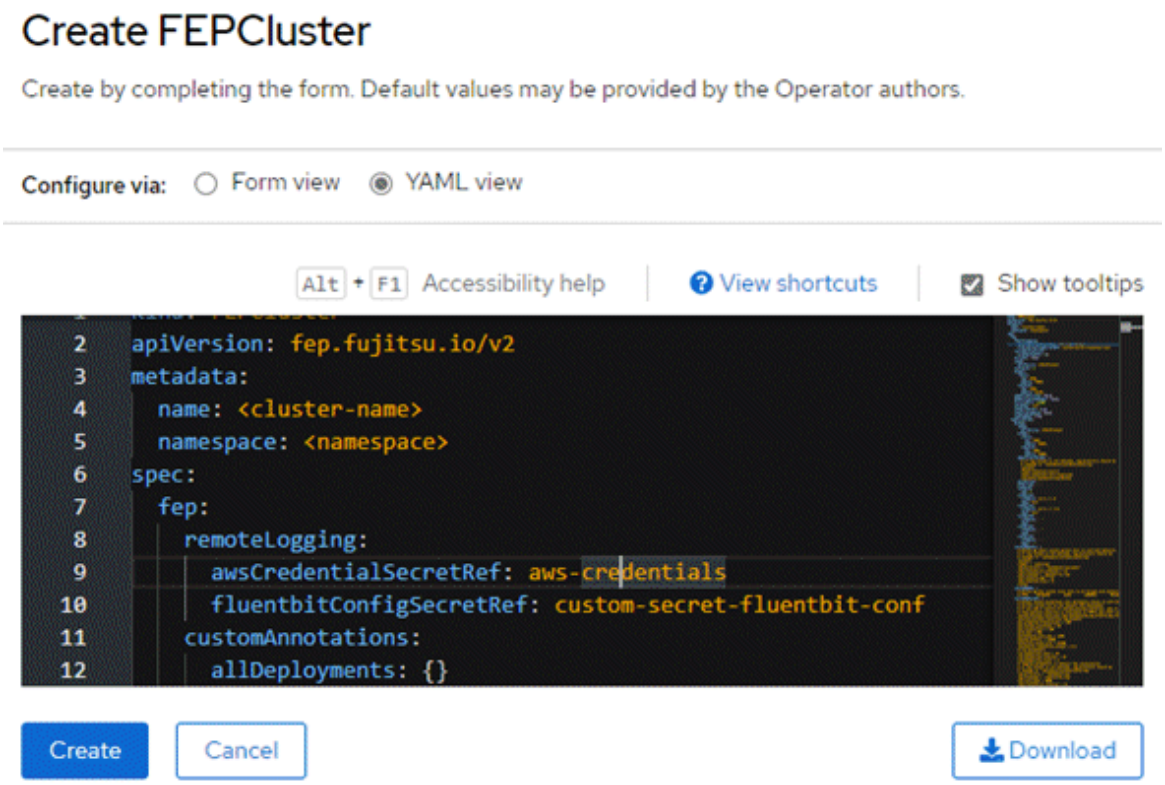
Create a Fluent Bit custom secret using the cloudwatch_logs Output plugin for Fluent Bit integration. For more information, refer to the Fluent Bit Manual.

In your main configuration file fluent-bit.yaml append the following Output section:

```
OUTPUT:
  Name: cloudwatch_logs  # Specifies the output plugin name
  Match: '*'             # Matches all log streams
  region: us-east-1      # The AWS region for CloudWatch Logs
  log_group_name: fluent-bit-cloudwatch  # Name of the CloudWatch log group
  log_stream_prefix: from-fluent-bit-     # Prefix for log stream names
  auto_create_group: On  # Automatically create the log group if it doesn't exist
  profile: <cloudwatch_logs_profile_default>  # The AWS profile to use for authentication
  auto_retry_requests: false  # Disables automatic retry for failed requests
  log_format: json    # Format of the logs to be sent
```

E.4.5 Reference the Secrets in FEPClusterCR

Reference the aws-credentials secret and the fluent-bit config secret in the FEPCluster CR using the appropriate parameters under remoteLogging, awsCredentialSecretRef and fluentbitConfigSecretRef as shown in the screenshot and create a cluster.



Once the FEPCluster is up and running, verify the fep-logging-fluent-bit container logs to ensure the integration is successful and Confirm Log Ingestion in Amazon CloudWatch Logs.

FUJITSU

Fujitsu
Enterprise Postgres 17 for Kubernetes

Reference

Linux

J2UL-RF17-01ENZ0(00)
November 2024

Preface

Purpose of this document

This document is a reference, and explains parameter.

Intended readers

This document is aimed at people who manage and operate.

Readers of this document are also assumed to have general knowledge of:

- Linux
- Kubernetes
- Containers
- Operators

Structure of this document

This document is structured as follows:

Chapter 1 Custom Resource Parameters

Explains the parameter.

Appendix A Default Metrics Queries

Explains the Default Metrics Queries

Appendix B Default Alert Rules

Explains the Default Alert Rules

Appendix C Operator Operation Event Notification

Explains the Operator Operation Event Notification

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes Fujitsu Enterprise Postgres	FEP
Transparent Data Encryption	TDE
Custom Resource	CR
Custom Resource Definition	CRD
Persistent Volume	PV

Abbreviations of manual titles

The following abbreviations are used in this manual as manual titles:

Full Manual Title	Abbreviations
Fujitsu Enterprise Postgres for Kubernetes User's Guide	User's Guide

Trademarks

- Linux is a registered trademark or trademark of Mr. Linus Torvalds in the U.S. and other countries.

- Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.
- S/390 is a registered trademark of International Business Machines Corporation ("IBM") in the U.S. and other countries.

Other product and company names mentioned in this manual are the trademarks or registered trademarks of their respective owners.

Export restrictions

If this document is to be exported or provided overseas, confirm legal requirements for the Foreign Exchange and Foreign Trade Act as well as other laws and regulations, including U.S. Export Administration Regulations, and follow the required procedures.

Issue date and version

Edition 1.0: November 2024

Copyright

Copyright 2021-2024 Fujitsu Limited

Contents

Chapter 1 Custom Resource Parameters 1
1.1 FEPCluster Parameter 1
1.2 Custom Resource Parameters 40
1.2.1 FEPCluster Custom Resource Parameters 40
1.2.2 FEP Cluster Configuration 41
1.2.3 FEPConfig Child Custom Resource Parameters 41
1.2.4 FEPUser Child Custom Resource Parameters 43
1.2.5 FEPVolume Child Custom Resource Parameters 45
1.2.5.1 Create Volumes 45
1.2.5.2 Delete Volumes 46
1.2.6 FEPCert Child Custom Resource Parameters 47
1.2.6.1 Create/ Update Certificates 47
1.2.6.2 Delete Certificates 47
1.2.7 FEPBackup Child Custom Resource Parameters 48
1.2.8 FEPRestore Custom Resource Parameters 49
1.2.9 FEPPgpool2 Custom Resource Parameters 53
1.2.10 FEPAction Custom Resource Parameters 57
1.2.10.1 FEPAction Specific Operation Details 59
1.2.11 FEPExporter Custom Resource 65
1.2.12 FEPAutoscale Custom Resource 67
1.2.13 FEPUpgrade Custom Resource 67
1.2.14 FEPLogging Custom Resources 69
1.2.15 FEP Custom Resources - spec.fep.pgBadger 70
1.2.16 FEP Custom Resources - spec.fep.pgAuditLog 71
1.2.16.1 Details of pgAuditLog.endpoint.authentication 71
1.2.16.2 CR example for customized pgaudit ConfigMap 72
1.2.16.3 CR example when uploading logs to Azure Blob 72
1.2.16.4 CR example for uploading logs to S3 73

Appendix A Default Metrics Queries 74

Appendix B Default Alert Rules 85

Appendix C Operator Operation Event Notification 88
C.1 FEPCluster Event Notification on Custom Resource Changes 88
C.2 FEPExporter Event Notification on Custom Resource Changes 94
C.3 Event Notification When FEPLogging Custom Resource Changes 95

Chapter 1 Custom Resource Parameters

This chapter explains the parameter.

1.1 FEPCluster Parameter

Equivalent Kubernetes command: kubectl apply -f FEPClusterCR.yaml

This operation will create a FEPCluster with supplied information in FEPClusterCR.yaml.

Initial configuration and subsequent changes to FEP Cluster are done through FEP Cluster CR.

Field	Default	Details
metadata.name	new-fep	Name for the Cluster. FEP server container will use this value for Patroni scope. e.g. new-fep
spec.fep.autoPodRestart	<omitted>	Optional This parameter affects the behaviour when value(s) of CPU, memory and/or image for FEP and/or optional Backup container are updated in FEPCluster CR. If it is NOT defined and set to true, operator will automatically create an action CR to make values effective by restarting all pods in an orderly fashion to minmise outage. If is set to false, automatic restart of PoDs will NOT happen. To make the changes effective, user must restart pods by creating action CR with type ‘pod_restart’ and arguments ‘ALL’
spec.fep.fepVersion	<omitted>	Optional When deploying a new FEP cluster, this parameter controls which FEP major version will be used for the deployment. If not specified, Operator will use latest FEP version supported by the Operator. When fepVersion is defined but not spec.fep.image.image, Operator will deploy the specific version of FEP. When both fepVersion and image are defined, Operator will use the image and discard the value of fepVersion. Current support value: 13, 14, 15, 16, 17 Note: Changing fepVersion from one version to another version is not supported after deployment.

Field	Default	Details
spec.fep.customAnnotation.allDeployments	{} (*)	Contents under this are optional. User can remove {} and add multiple key-value pairs. All of these pair will be added to annotations of FEP statefulSet and FEP Pods. If left at default, no annotation is added to Pods and statefulSets
spec.fep.image.image	<omitted>	FEP server container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-17-server:ubi9-17-1.0 It is optional Image line is omitted by default. This key has a higher precedence than fepVersion. If both fepVersion and image are omitted, Operator will use the latest FEP version that it supports. If both fepVersion and image are specified, Operator will use the specified image and ignore the value in fepVersion.
spec.fep.image.pullPolicy	IfNotPresent	
spec.fep.mcSpec.limits	cpu: 500m memory: 700Mi (If spec.fep.databaseSize is medium) cpu: 2 memory: 4Gi (If spec.fep.databaseSize is large) cpu: 4 memory: 16Gi	
spec.fep.mcSpec.requests	cpu: 200m memory: 512Mi (If spec.fep.databaseSize is medium) cpu: 1 memory: 2Gi (If spec.fep.databaseSize is large) cpu: 2 memory: 8Gi	
spec.fep.databaseSize		Specifiable values: small, medium, large The operator defines the values for cpu, memory, and postgresql.conf, matching the specified values. If the target parameter is defined, it is not overwritten. Can be set only when creating

Field	Default	Details
		FEPCluster custom resource. After you create a FEPCluster custom resource, you customize it by editing each parameter.
spec.fep.sysExtraLogging	false	To turn extra debugging on, set value to true It can be turned on/off at any time
spec.fep.sysExtraEvent	false	Options To turn on event notification for custom resource changes, set the value to true. You can turn it on or off at any time.
spec.fep.instances	1	Number of nodes in the cluster, including both Master and Replicas. In Example CR, it is kept at 1 for certification. However, user can change it to 3 for 1 master and 2 replicas.
spec.fep.servicePort	27500	TCP port for FEP master service
spec.fep.syncMode	off	Replication Mode: off - async replication on - sync replication
spec.fep.standby.enable	false	This parameter enables the hot standby configuration. Enabled at true.
spec.fep.standby.method		Specifies the method for achieving a hot standby configuration. archive-recovery - Uses continuous recovery. streaming - Uses streaming replication.
spec.fep.standby.pgBackrestConf		Required for both continuous recovery and streaming replication methods. You must specify the backup storage on which the production environment is backed up. AWS S3 and Azure Blob Storage are available.
spec.fep.standby.streaming.host		Specify this option to use the streaming replication method. Specify the external IP of the LoadBalancer you created in "Defining a Streaming Replication Method" in the User's Guide.
spec.fep.standby.streaming.port		Specify this option to use the streaming replication method. Specify the port defined in the LoadBalancer you created in "Defining a Streaming Replication Method" in the User's Guide.
spec.fep.forceSsl	true	Controls the use of SSL only for communication between FEPCluster

Field	Default	Details
		containers. The changes are reflected in pg_hba.conf. Changing this parameter is not reflected in pg_hba.conf if the automatic certificate generation feature is enabled.
spec.fep.locale	<omitted> (*)	Optional Can only be specified when creating a FEPCluster. Database Cluster Locale Settings: ja_JP - Japanese locale Default - C
spec.fep.monitoring		This is an Optional section. This defines whether monitoring enabled(true) or disabled(false) , MTLS enabled or disabled & Basic authentication enabled or not
spec.fep.monitoring.enable	false	If set true, the operator will create FEPExporter with given spec
spec.fep.monitoring.fepExporter		This is Optional section. Exporter spec section applied only if enable: true
spec.fep.monitoring.fepExporter.authSecret		This is Optional section. Base Authentication secret to provide username & encrypted password of user
spec.fep.monitoring.fepExporter.authSecret.secretName	(created by user)	Mandatory Name of secret that contains username and password
spec.fep.monitoring.fepExporter.authSecret.userKey	(created by user)	Mandatory Key of username in specified secret
spec.fep.monitoring.fepExporter.authSecret.passwordKey	(created by user)	Mandatory Key of password in specified secret
spec.fep.monitoring.fepExporter.tls		This is optional section. FEPExporter MTLS specs. Mandatory if tls specs defined for Prometheus specs
spec.fep.monitoring.fepExporter.tls.certificateName	(created by user)	Mandatory.This points to Kubernetes TLS secret that contains the certificate of FepExporter. Prometheus will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.fep.monitoring.fepExporter.tls.caName	(created by user)	Mandatory This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt.

Field	Default	Details
spec.fep.monitoring.fepExporter.customLabel		Optional List of key value pair to be added to Prometheus ServiceMonitor label. The following label will always be added to ServiceMonitor, regardless if a value is specified here or not. fepsmgrp: sm-fep-exporter
spec.fep.monitoring.prometheus		This is Optional section. Prometheus specs are mandatory if tls specs defined for FEPExporter
spec.fep.monitoring.prometheus.tls		Prometheus MTLS specs
spec.fep.monitoring.prometheus.tls.certificateName	(created by user)	This is an Optional parameter. These points to Kubernetes TLS secret that contains the certificate of Prometheus. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.fep.monitoring.prometheus.tls.caName	(created by user)	This is an Optional parameter. This point to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt.
spec.fep.externalMonitoring.cloudWatch		This is an Optional section. Define this option when linking with CloudWatch.
spec.fep.externalMonitoring.cloudWatch.enable	true	Optional A value of true forwards Fujitsu Enterprise Postgres metrics to CloudWatch. Specify false to cancel log transfer.
spec.fep.externalMonitoring.cloudWatch.shedule	"0-59/10 * * * *"	You can specify how often to transfer. Specified as a Cron value.
spec.fep.externalMonitoring.cloudWatch.namespace		Specifies the CloudWatch Namespace to which metrics are forwarded. Required if spec.fep.externalMonitoring.cloudWatch is specified
spec.fep.externalMonitoring.cloudWatch.defaultMetrics	true	Specify true to capture and forward the metrics described in "Metrics Collected by CloudWatch" in the User's Guide. Specify false to forward only custom metrics to CloudWatch.
spec.fep.externalMonitoring.cloudWatch.customMetrics		Optional If you want to forward custom metrics, specify the ConfigMap name where you defined the custom metrics.
spec.fep.externalMonitoring.cloudWatch.dimensionName		Optional

Field	Default	Details
		If adding a Dimension, specify a name of your choice. If dimensionValue is omitted, this value is ignored.
spec.fep.externalMonitoring.cloudWatch.dimensionValue		Optional If adding a Dimension, specify any value. If dimensionName is omitted, this value is ignored.
spec.fep.externalMonitoring.cloudWatch.authentication.cloudWatchCredentials		Specify a secret name that defines a credentials for a role that has permission to forward metrics to CloudWatch. Required if spec.fep.externalMonitoring.cloudWatch is specified
spec.fep.externalMonitoring.cloudWatch.authentication.cloudWatchConfig		Specify a ConfigMap name that defines the config information for a role that has permission to forward metrics to CloudWatch. Required if spec.fep.externalMonitoring.cloudWatch is specified
spec.fep.externalMonitoring.cloudWatch.databases	postgres	Specifies the databases from which to collect metrics, in list format. The default is to collect metrics from the postgres database.
spec.fep.podAntiAffinity	false	Defines that all the pods should not run on same worker node
spec.fep.podDisruptionBudget	false	Allows to maintain minimum number of pods of an application even when some nodes are voluntarily drained for say, maintenance
spec.fep.replicationSlots		List of Patroni permanent replication slots.
spec.fep.replicationSlots.demo_subscription1		The 'demo_subscription1' is the slot name. This name cannot be same as any pod name (e.g., new-fep-sts-01) in the cluster. Otherwise, the slot will not be created.
spec.fep.replicationSlots.type	logical	Must be 'logical' for logical replication
spec.fep.replicationSlots.database	postgres	Specify the database name for logical replication
spec.fep.replicationSlots.plugin	pgoutput	FEP supports 'pgoutput' by default.
spec.fep.usePodName		Optional Setting this key to true will make internal POD communication, both Patroni and Postgres to use hostname,

Field	Default	Details
		instead of IP address. This is important for TLS as the hostname of the POD is predictable and can be used to create Server Certificate, whereas IP address is unpredictable and cannot be used to create Certificate. There is no negative effect setting this key to true even if TLS (i.e. Server Certificate) is not used.
spec.fep.patroni.tls.certificateName	(created by user)	Optional This point to Kubernetes TLS secret that contains the certificate for Patroni. The certificate itself is stored in the key tls.crt. This field is optional. When this key is set, the Operator will ignore the value in systemCertificates
spec.fep.patroni.tls.caName	(created by user)	Optional This points to Kubernetes configmap that contains additional CA for Patroni to verify client. The CA is stored in the key ca.crt. This field is optional.
spec.fep.postgres.tls.certificateName	(created by user)	Optional This points to Kubernetes TLS secret that contains the certificate for Postgres. The certificate itself is stored in the key tls.crt. This field is optional. When this key is set, Operator will ignore the value in systemCertificates
spec.fep.postgres.tls.caName	(created by user)	Optional This point to Kubernetes configmap that contains additional CA for Postgres to verify client. The CA is stored in the key ca.crt. This field is optional.
spec.fep.postgres.tls.privateKeyPassword	(created by user)	Optional This points to Kubernetes secret that contains the password for the above private key. This field is optional.
spec.fep.pgAuditLog.auditLogPath		Use this value for log_directory in pgaudit.conf If pgAuditLog.auditLogPath is not defined: use '/database/log/audit' or '/database/userdata/data/log' when log volume is not defined .
spec.fep.pgAuditLog.schedules		Schedule to upload auditlog
spec.fep.pgAuditLog.schedules.upload		Upload schedule in crontab format
spec.fep.pgAuditLog.endpoint.protocol	http	Optional Default: http Supported values:

Field	Default	Details
		- ‘http’ - ‘s3’ - ‘blob’
spec.fep.pgAuditLog.endpoint.url		Webserver URL to upload the auditlog files
spec.fep.pgAuditLog.endpoint.customCertificateName		Optional Secret that contains the certificate to setup communication with Web server
spec.fep.pgAuditLog.endpoint.insecure	false	Optional equivalent to curl -insecure option
spec.fep.pgAuditLog.endpoint.authentication		Optional This item is the secret name for endpoint authentication. The end user needs to provide this secret to use upload feature. This secret is used for authentication of each protocol accordingly. Refer to "1.2.16.1 Details of pgAuditLog.endpoint.authentication" for details. If this is not specified, a default secret <cluster-name>-pgauditlog-auth will be created.
spec.fep.pgAuditLog.endpoint.fileUploadParameter	file	Optional The file upload parameter defined by the web server
spec.fep.pgAuditLog.endpoint.azureBlobName		Only take effect when protocol is ‘blob’ Optional The blob name of pgaudit log file. Default: [cluster name]-sts-[pod index]-pgauditlog.zip
spec.fep.pgAuditLog.endpoint.azureContainerName		Required with protocol is ‘blob’ This item is the container name of the Azure Storage account
spec.fep.pgAuditLog.config		Optional Default: none This item requires a ConfigMap with this name to exist in the same namespace of the FEPCluster. The ConfigMap will be used as pgAudit config file. The ConfigMap need to have a key ‘pgaudit.conf’.
spec.fep.pgAuditLog.enable		Optional Default: false When set to ‘true’, the pgaudit extension is enabled automatically.
spec.fep.pgBadger.schedules.create		The 'create' schedule to create report and upload it to endpoint

Field	Default	Details
spec.fep.pgBadger.schedules.cleanup		The 'cleanup' schedule to delete the report left in container
spec.fep.pgBadger.options.incremental	false	Default: false; When set to true: create incremental report in pgbadger
spec.fep.pgBadger.endpoint.authentication		a secret to contain authentication info to access endpoint support basic auth only
spec.fep.pgBadger.endpoint.customCertificateName		Client certitificate reference in customCertificate CR
spec.fep.pgBadger.endpoint.fileUploadParameter	file	The file upload parameter defined by the web server
spec.fep.pgBadger.endpoint.insecure	false	equivalent to curl -insecure option
spec.fep.pgBadger.endpoint.url		Web server url to upload the report file
spec.ldap2pg.enable	true	Setting this to "true" will enable ldap2pg to execute periodically according to schedule defined. Setting this to "false" will remove the cronjob that execute the ldap2pg.
spec.ldap.caConfigMapRef		If LDAP server certificate is signed by a private CA, this key should point to a configmap that has the chain of certificates that ldap2pg and FEP should trust. Operator expects the key name in the configmap be ca.crt.
spec.ldap.ldapconfSecretRef		Name of secret that contains the ldap.conf When spec.ldap is defined but spec.ldap.ldapconfSecretRef is not defined, operator will create a default secret <fep-cluster>-ldapconf. Operator expects the key name in the secret be ldap.conf.
spec.ldap2pg.ldap2pgymlConfigMapRef		Name of configmap that contains the ldap2pg.yml When spec.ldap2pg is defined but spec.ldap2pg.ldap2pgymlConfigMapRef is not defined, Operator will create a default configmap <fep-cluster>-ldap2pgyml. FEP Operator expects the key name in the configmap be ldap2pg.yml.
spec.ldap2pg.mode	check	Whether ldap2pg should run in "check" mode or "real" mode. If not defined, ldap2pg will run in check mode.

Field	Default	Details
spec.ldap2pg.schedule		Schedule to execute ldap2pg in a crontab format. If defined, Operator will create a cronjob using fep-cronjob container and remotely execute ldap2pg on fep-patroni container on a regular basis. If the schedule is not provided, Operator will set the schedule to 5/* * * *.
spec.ldap2pg.skipPrivileges	false	Options Configure synchronization of role attributes and permissions between ldap2pg and FEPCluster. Specify false if you want to synchronize role attributes with permissions, or true if you do not. If true, the behavior is the same as the --skip-privileges (-P) option of the ldap2pg command.
spec.fep.feputils.image	<omitted>	FEPUtils container image to use, quay.io/fujitsu/fujitsu-enterprise-postgres-17-utils:ubi9-17-1.0 Optional. Omitted by default. In this case, the image URL is obtained from the operator container environment. If you specify an image, the operator will use that image to deploy the Utils container. When fepChildCrVal.storage.autoresize.enable is true, use this image to expand the pvc-auto-resize container of the fep-tuning Pod.
spec.fep.fepcronjob.image	<omitted>	FEPCronjob container image to use, quay.io/fujitsu/fujitsu-enterprise-postgres-cronjob:ubi9-17-1.0 Optional. Omitted by default. In this case, the image URL is obtained from the operator container environment. If you specify an image, the operator will use that image to deploy the Cronjob container.
spec.fep.autoTuning.prometheus.prometheusUrl		Required if fepChildCrVal.storage.autoresize.enable is true. Specifies the URL of the Prometheus for which you want to retrieve metrics.

Field	Default	Details
spec.fep.autoTuning.prometheus.authSecret		Optional Basic authentication secret that provides the user name and encrypted password
spec.fep.autoTuning.prometheus.authSecret.secretName		Username and password, or the name of the secret that contains the token
spec.fep.autoTuning.prometheus.authSecret.userKey		Key of the Secret given the user name
spec.fep.autoTuning.prometheus.authSecret.passwordKey		Key of the Secret with the password specified
spec.fep.autoTuning.prometheus.authSecret.tokenKey		Key of the Secret given the token
spec.fep.autoTuning.prometheus.authSecret.proxyKey		Key of the Secret specified by the proxy
spec.fep.autoTuning.prometheus.tls		
spec.fep.autoTuning.prometheus.tls.certificateName		Refers to the Kubernetes TLS secret that contains the certificate and private key. Prometheus uses this for certificate authentication. The certificate and private key itself are stored in the tls.crt and tls.key keys.
spec.fep.autoTuning.prometheus.tls.caName		Refers to the Kubernetes ConfigMap containing the additional CA that the client uses to verify the server certificate. The CA is stored in the ca.crt key.
spec.fep.autoTuning.prometheus.maxRetry		Specifies the maximum number of retries when a query to Prometheus fails. If not specified, a maximum of 60 retries are attempted.
spec.fep.velero.enable	false	Optional Specifies whether the Velero DR feature is used (true) or not (false). This is omitted by default. In this case, the Velero DR feature is not available.
spec.fep.velero.labels		Optional If the Velero DR feature is used, specify the label to be given to the resource to be backed up by Velero. You can specify multiple labels. If omitted, backup-group: fep-backup is given.
spec.fep.velero.backup		Specifies the object storage information that stores the backup data and archive wal for FEPCluster to be built in a DR environment.

Field	Default	Details
		Otherwise, FEPCluster built in a DR environment will fail to back up to object storage.
spec.fep.velero.backup.pgbackrestParams		"|" When nothing is specified, and the parameter set in pgbackrest.conf is described from the line below. (Descriptions vary depending on the provider used) If you use the same object storage as in production, specify a different object storage path (repo*-path) than fepChildCrVal.backup.pgbackrestParams. If you specify the same object storage path, you will receive an event notification.
spec.fep.velero.backup.pgbackrestKeyParams		Optional "|" is fixed, and the following line describes the parameters to be set in pgbackrest.conf. The value described by this parameter is masked with *****.
spec.fep.velero.backup.caName		Optional Set to use a CA file other than the system default. Specifies the name of the Configmap you created. If you use a different CA file than the production environment, give the CA file a different name and set it here. It must also be deployed in the DR environment.
spec.fep.velero.backup.repoKeySecretName		Optional Specifies the name of the Kubernetes Secret generated from the object storage key file. Specify in array format. If you use a different secret from the production environment, give the secret a different name and set it here. It must also be deployed in the DR environment.
spec.fep.velero.restore.image.image		Optional Image of the container to perform the restore It is omitted by default. In this case, the URL for image is obtained from the operator container environment.
spec.fep.velero.restore.image.pullPolicy	IfNotPresent	Specifies the pull policy for the image. - Always - IfNotPresent

Field	Default	Details
spec.fep.velero.restore.mcSpec.limit	cpu: 200m memory: 300Mi	Specifies the maximum number of resources to allocate to the restore execution container.
spec.fep.velero.restore.mcSpec.request	cpu: 100m memory: 200Mi	Specifies the lower number of resources to allocate to the restore execution container.
spec.fep.velero.resotre.restoreTargetRepo	1	Specifies the backup data used to restore FEPCluster to the DR environment and the object storage information where the archive wal is stored. This is the number of the repo in fepChildCrVal.backup.pgbackrestParams.
spec.fep.fixedStats.scheduleN		Schedule a locked statistics. Specify an integer for N.
spec.fep.fixedStats.scheduleN.fixSchedule		Time to start locked statistics. schedule in cron format The date and time are in UTC time.
spec.fep.fixedStats.scheduleN.unfixSchedule		Options Time to unpin locked statistics. Returns to regular statistics. If not specified, it is not cleared. schedule in cron format The date and time are in UTC time.
spec.fep.fixedStats.scheduleN.targetDb		Specify the database for which you want to locked statistics.
spec.fep.fixedStats.scheduleN.fixedObject		Options If the scope to be locked is smaller than the database, specify that object. Specify schema.table.column. The minimum range is the column and the maximum range is the schema.
spec.fep.fixedStats.endpoint.protocol		Specify the object storage vendor where statistics are stored. s3, blob, gcs can be specified. Also, if you want to directly import a file on the container, specify local.
spec.fep.fixedStats.endpoint.authentication		Specify this option if the protocol is s3, blob, gcs. Authentication for accessing object storage Specify a secret file that contains confidential information.
spec.fep.fixedStats.scheduleN.url		Specify this option if the protocol is s3, gcs.

Field	Default	Details
		Specify the URL from which to download the statistics binary file.
spec.fep.fixedStats.scheduleN.azureBlobName		Specify this if the protocol is a blob. Name of the blob containing the statistics binary file.
spec.fep.fixedStats.scheduleN.azureContainerName		Specify this if the protocol is a blob. This item is the container name of the Azure storage account.
spec.fep.fixedStats.scheduleN.file		Specify this if the protocol is a local. Specify the name of the file deployed on the fep-patropni container.
spec.fep.fixedStats.scheduleN.update	false	Set this setting to true if you want to download statistics to be locked from object storage.
spec.fep.fixedStats.image		The CronJob image to use. If not specified, the operator uses the latest version supported by the operator.
spec.fep.fixedStats.pullPolicy	IfNotPresent	
spec.fep.fixedStats.scheduleN.enable		Options You can specify whether scheduled statistics are to be locked or released. Executed if omitted or true, not if false.
spec.fep.freezingTuples.enable	false	Options When true is specified, enables periodic execution of freezing operations.
spec.fep.freezingTuples.scheduleN		Options Specifies the schedule for the freeze operation. You can specify multiple names in dictionary format. Specify an integer for N.
spec.fep.freezingTuples.scheduleN.start	0 1 * * *	Specifies the date and time for starting processing in cron format. If omitted, the default values are applied.
spec.fep.freezingTuples.scheduleN.executionTime	3600	Options Specified value: string Units: s, m, h,d Specifies the duration of the processing. If no unit is specified, s is assumed. If omitted, the default values are applied.
spec.fep.backupStats.enable	false	You can set statistics to be backed up. If set to false, no backup is performed. If set to true, backup is performed.

Field	Default	Details
		If spec.fep.backupStats.enable is not defined as false when FEPCluster is first built, it is set to true. If spec.fep.backupStats.schedule1 is not defined when FEPCluster is first built, a backup with default settings is defined in the FEPCluster custom resource.
spec.fep.backupStats.image		The CronJob image to use. If not specified, the operator uses the latest version supported by the operator.
spec.fep.backupStats.pullPolicy	IfNotPresent	
spec.fep.backupStats.scheduleN		Schedule a backup of the statistics. Specify an integer for N.
spec.fep.backupStats.scheduleN.backupSchedule		Time to start taking statistics backups. schedule in cron format The date and time are in UTC time.
spec.fep.backupStats.scheduleN.targetDb		Optional Specify the database to be backed up. If omitted, runs for all databases.
spec.fep.backupStats.scheduleN.fixedObject		Optional If the backup target is less than the database, specify the object. Specify schema.table.column. The minimum range is the column and the maximum range is the schema.
spec.fep.backupStats.scheduleN.comment		Optional Comments that can be defined when backing up statistics. If omitted, FepFixedStatsBackup: scheduleN is set. Do not use the following phrases in comments. FepFixedStats
spec.fep.backupStats.scheduleN.retention		Options At the same time as a scheduled backup, you can delete backups that are stored in the target database for a specified number of days or earlier. Specify an integer. If omitted, no deletion is performed.
spec.fep.backupStats.scheduleN.enable		Options You can specify whether scheduled statistics are to be locked or released. Executed if omitted or true, not if false.

Field	Default	Details
spec.fepChildCrVal.customCertificates		Optional An array of elements for defining a certificate. It consists of the following parameters: - username - certificateName - caName Used to setup SSL connection between publisher and subscriber clusters for logical replication.
spec.fepChildCrVal.customCertificates.userName		Optional This should be the username of the publisher database. When this parameter is specified, an empty folder is created under FEP Server Container- /tmp/custom_certs/ <username>. The custom certificates are mounted in this empty folder. However, if this parameter is not specified, the section is ignored and folder is not created; hence the certificates are not mounted without it.
spec.fepChildCrVal.customCertificates.certificateName	(created by user)	Optional This points to Kubernetes TLS secret that contains the custom certificate. The certificate itself is stored in the key tls.crt.
spec.fepChildCrVal.customCertificates.caName	(created by user)	Optional This points to Kubernetes configmap that contains CA certificate to verify server. The CA is stored in the key ca.crt.
spec.fepChildCrVal.backup		Optional This section is defined to enable fepbackup sidecar for cluster backup feature.
spec.fepChildCrVal.backup.image.image	<omitted>	FEP backup container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-17-backup:ubi9-17-1.0 It is optional. Image line is omitted by default. In such a case, it will pick up URL of image from operator container environment. If you specify the image, Operator will take that image to deploy backup container
spec.fepChildCrVal.backup.image.pullPolicy	IfNotPresent	

Field	Default	Details
spec.fepChildCrVal.backup.mcSpec.limits	cpu: 0.2 memory: "300Mi"	
spec.fepChildCrVal.backup.mcSpec.requests	cpu: 0.1 memory: "200Mi"	
sepc.fepChildCrVal.backup.type		Optional Specifiable value: local Apply the settings for taking a backup to the PV to the custom resource. Can only be set the first time the FEPCluster custom resource is applied. You cannot add or change settings after applying.
spec.fepChildCrVal.backup.pgbackrestParams	(If sepc.fepChildCrVal.backup.type is local) [global] repo1-retention-full=7 repo1-retention-full-type=time log-path=/database/log/backup	Specifies the object storage information that stores the backup data and archive wal. "|" When nothing is specified, and the parameter set in pgbackrest.conf is described from the line below. The value described by this parameter is masked with *****. (Descriptions vary depending on the provider used)
spec.fepChildCrVal.backup.pgbackrestKeyParams		Optional "|" is fixed, and the following line describes the parameters to be set in pgbackrest.conf. The value described by this parameter is masked with *****.
spec.fepChildCrVal.backup.caName		Optional Set to use a CA file other than the system default. Specifies the name of the Configmap you created.
spec.fepChildCrVal.backup.repoKeySecretName		Optional Specifies the name of the Kubernetes Secret generated from the object storage key file. Specify in array format.
spec.fepChildCrVal.backup.schedule.num	0 (If sepc.fepChildCrVal.backup.type is local) 2	Number of schedules to set The maximum number of backup schedules is 5.
spec.fepChildCrVal.backup.scheduleN.schedule	(If sepc.fepChildCrVal.backup.type is local) schedule1: schedule: "15 0 * * 0"	Backup schedule in cron format. The date and time is UTC time.

Field	Default	Details
	schedule2: schedule: "15 0 * * 1-6"	
spec.fepChildCrVal.backup.scheduleN.type	(If sepc.fepChildCrVal.backup.type is local) schedule1: type: full schedule2: type: incr	full: Perform a full backup (Back up the contents of the database cluster). incr ー Perform an incremental backup (Back up only the database cluster files that were changed to the last backup migration).
spec.fepChildCrVal.backup.scheduleN.repo	1	Optional Gets a backup in the specified repository. The range is 1 to 256.
spec.fepChildCrVal.customPgAudit	[output] logger = 'auditlog' log_directory = '/database/log/audit' log_truncate_on_rotation = on log_filename = 'pgaudit-%a.log' log_rotation_age = 1d log_rotation_size = 0 [rule]	PgAudit file content
spec.fepChildCrVal.customPgHba	# define pg_hba custom rules here to be merged with default rules. # TYPE DATABASE USER ADDRESS METHOD	Entries to be inserted into pg_hba.conf

spec.fepChildCrVal.customPgParams	# define custom postgresql.conf parameters below to override defaults.
	# Current values are as per default FEP deployment
	shared_preload_libraries='pgx_datamasking,pg_prewarm,pg_stat_statements,fsep_operator_security'
	session_preload_libraries='pg_prewarm'
	max_prepared_transactions = 100
	max_worker_processes = 30
	max_connections = 100
	work_mem = 1MB
	maintenance_work_mem = 12MB
	shared_buffers = 128MB
	effective_cache_size = 384MB
	checkpoint_completion_target = 0.8
	# tcp parameters
	tcp_keepalives_idle = 30
	tcp_keepalives_interval = 10
		Postgres configuration in postgresql.conf
		If the FEP server container utilizes images with a FEPBaseVersion less than 15, exclude fsep_operator_security from the configuration.
		If spec.fep.databaseSize is defined, the default value will be changed as shown below.
		shared_buffers = 30% of spec.fep.mcSpec.limits.memory
		work_mem = 30% of spec.fep.mcSpec.limits.memory / max_connections / 2
		effective_cache_size = 75% of spec.fep.mcSpec.limits.memory
		maintenance_work_mem = 10% of spec.fep.mcSpec.limits.memory / (1 + autovacuum_max_workers)

Field	Default	Details

	tcp_keepalives_count = 3
	# logging parameters in default fep installation
	# if log volume is not defined, log_directory should be
	# changed to '/database/userdata/data/log'
	log_directory = '/database/log'
	log_filename = 'logfile-%a.log'
	log_file_mode = 0600
	log_truncate_on_rotation = on
	log_rotation_age = 1d
	log_rotation_size = 0
	log_checkpoints = on
	log_line_prefix = '%e %t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h'
	log_lock_waits = on
	log_autovacuum_min_duration = 60s
	logging_collector = on
	pgaudit.config_file='/opt/app-root/src/pgaudit-cfg/pgaudit.conf'
	log_replication_commands = on
	log_min_messages = WARNING
	log_destination = stderr
	# wal_archive parameters in default fep installation
	archive_mode = on
	archive_command = 'pgbackrest --stanza=backupstanza --config=/database/userdata/pgbackrest.conf archive-push %p'
	wal_level = replica
	max_wal_senders = 12
	wal_keep_segments = 64
	track_activities = on
	track_counts = on
	password_encryption = 'md5'	

spec.fepChildCrVal.storage.dataVol		Mandatory volume
spec.fepChildCrVal.storage.dataVol.size	2Gi (**)	Size of data volume. Data volume must be specified
spec.fepChildCrVal.storage.dataVol.storageClass	<omitted> (*)	StorageClass for data volume:

Field	Default	Details
		When this line is omitted, the PV created will use default storage class in the Kubernetes cluster
spec.fepChildCrVal.storage.dataVol.accessModes	<omitted> (*)	accessModes for data volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]
spec.fepChildCrVal.storage.walVol		Mandatory volume
spec.fepChildCrVal.storage.walVol.size	1200Mi (**) (If spec.fepChildCrVal.storage.dataSize is defined) 5Gi	Size of WAL volume. WAL volume must be specified
spec.fepChildCrVal.storage.walVol.storageClass	<omitted> (*)	StorageClass for WAL volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster
spec.fepChildCrVal.storage.walVol.accessModes	<omitted> (*)	accessModes for WAL volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]
spec.fepChildCrVal.storage.tablespaceVol		Optional volume
spec.fepChildCrVal.storage.tablespaceVol.size	512Mi (**) (If spec.fepChildCrVal.storage.dataSize is defined) The value specified in spec.fepChildCrVal.storage.dataSize.	Size of tablespace volume. This volume is optional and can be omitted
spec.fepChildCrVal.storage.tablespaceVol.storageClass	<omitted> (*)	StorageClass for tablespace volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster
spec.fepChildCrVal.storage.tablespaceVol.accessModes	<omitted> (*)	accessModes for tablespace volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]
spec.fepChildCrVal.storage.archivewalVol		Mandatory if backup section is defined. Optional otherwise

Field	Default	Details
spec.fepChildCrVal.storage.archivewalVol.size	1Gi (**) (If spec.fepChildCrVal.storage.dataSize is defined and sepc.fepChildCrVal.backup.type is local) (spec.fepChildCrVal.storage.dataSize/10)*14	Size of archivewal volume. This volume is optional and can be omitted
spec.fepChildCrVal.storage.archivewalVol.storageClass	<omitted> (*)	StorageClass for Archived WAL volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster When the number of instance is more than 1 and backup is not done on S3, both archivewalVol and backupVol must be hosted on Shared storage such as NFS with respective storageClass
spec.fepChildCrVal.storage.archivewalVol.accessModes	<omitted> (*)	accessModes for Archived WAL volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce] When the number of instance is more than 1 and backup is not done on S3, both archivewalVol and backupVol must be hosted on Shared storage such as NFS with accessMode set to [ReadWriteMany]
spec.fepChildCrVal.storage.logVol		Optional volume
spec.fepChildCrVal.storage.logVol.size	1Gi (**) (If spec.fepChildCrVal.storage.dataSize is defined) 5Gi	Size of log volume. This volume is optional and can be omitted
spec.fepChildCrVal.storage.logVol.storageClass	<omitted> (*)	StorageClass for log volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster
spec.fepChildCrVal.storage.logVol.accessModes	<omitted> (*)	accessModes for log volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]

Field	Default	Details
spec.fepChildCrVal.storage.backupVol		Mandatory if backup section is defined. Optional otherwise
spec.fepChildCrVal.storage.backupVol.size	2Gi (**) (If spec.fepChildCrVal.storage.dataSize is defined and sepc.fepChildCrVal.backup.type is local) spec.fepChildCrVal.storage.dataSize*14	Size of backup volume. This volume is optional and can be omitted
spec.fepChildCrVal.storage.backupVol.storageClass	<omitted> (*)	StorageClass for backup volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster When the number of instance is more than 1 and backup is not done on S3, both archivewalVol and backupVol must be hosted on Shared storage such as NFS with respective storageClass
spec.fepChildCrVal.storage.backupVol.accessModes	<omitted> (*)	accessModes for backup volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce] When the number of instance is more than 1 and backup is not done on S3, both archivewalVol and backupVol must be hosted on Shared storage such as NFS with accessMode set to [ReadWriteMany]
sepc.fepChildCrVal.storage.dataSize		Specify the amount of data at the data storage destination. The operator defines the size of dataVol, walVol, logVol, tablespaceVol, archivewalVol, and backupVol based on the specified value. If you specify individual volumes, specify the size of each volume definition.
sepc.fepChildCrVal.storage.accessModes		Specify the accessModes for each volume if you want to specify them in a batch. If you want to set them individually, specify the accessModes for each volume definition.
sepc.fepChildCrVal.storage.storageClass		Specify the storageClass for each volume if you want to specify them in a batch. If you want to set them individually,

Field	Default	Details
		specify the storageClass for each volume definition.
spec.fepChildCrVal.storage.autoresize		
spec.fepChildCrVal.storage.autoresize.enable	false	Optional Specified value: boolean true to enable auto-extension for PVCs.
spec.fepChildCrVal.storage.autoresize.mcSpec.limits	cpu: 50m memory: 60Mi	Optional Specifies the resource limit that can be allocated to pvc-auto-resize container.
spec.fepChildCrVal.storage.autoresize.mcSpec.requests	cpu: 10m memory: 5Mi	Optional Specifies the resources to assign that can be allocated to pvc-auto-resize container.
spec.fepChildCrVal.storage.autoresize.interval	30	Optional Units: s Specifies the interval between metric checks. If 0 or less is specified, the PVC is not extended.
spec.fepChildCrVal.storage.autoresize.threshold	80	Optional Specified value: integer Unit:% Specifies the storage utilization threshold. Extends the PVC when this value is exceeded. When 0 is specified, storage utilization is not checked. The xxxVol.threshold applies to all storage that is not defined.
spec.fepChildCrVal.storage.autoresize.increaseType	percent	Optional Specified value: percent, size Specifies how the PVC extension is estimated when the threshold is exceeded. When percent is specified Expands the PVC by the specified percentage of its original capacity. If size is specified Extends the PVC by the specified amount (Gi). Applies to all storage where xxxVol.increaseType is not defined.

Field	Default	Details
spec.fepChildCrVal.storage.autoresize.increase	25	Optional Specified value: integer Units:% or Gi Specifies the extension amount for the PVC. The units depend on the value specified for increaseType. If a value less than or equal to 0 is specified, no extension is performed. This applies to all storage where xxxVol.increase is not defined.
spec.fepChildCrVal.storage.autoresize.storageLimit		Optional Specified value: integer Units: Gi Specifies the maximum value by which the PVC can be extended. If not specified, the extension is unrestricted. If you do not specify this value, we recommend that you verify that the storage class being used has a namespace quota. Do not extend the PVC when less than or equal to disk space is specified. Applies to all storage where xxxVol.storageLimit is not defined.
spec.fepChildCrVal.storage.xxxVol		xxx is the contents of data, wal, log, tablespace, archivewal, backup
spec.fepChildCrVal.storage.xxxVol.threshold		Optional Specified value: integer Unit:% Specifies the storage utilization threshold. Extends the PVC when this value is exceeded. When 0 is specified, storage utilization is not checked. If not specified, it follows the value specified in autoresize.threshold.
spec.fepChildCrVal.storage.xxxVol.increaseType		Optional Specified value: percent, size Specifies how the PVC extension is estimated when the threshold is exceeded. When percent is specified

Field	Default	Details
		Expands the PVC by the specified percentage of its original capacity. If size is specified Extends the PVC by the specified amount (Gi). If not specified, the value specified by autoresize.increaseType.
spec.fepChildCrVal.storage.xxxVol.increase		Optional Specified value: integer Units:% or Gi Specifies the extension amount for the PVC. The units depend on the value specified for increaseType. If not specified, the value specified by autoresize.increase.
spec.fepChildCrVal.storage.xxxVol.storageLimit		Optional Specified value: integer Units: Gi Specifies the maximum capacity by which the PVC can be extended. Do not expand if the specification is less than or equal to the disk capacity. If not specified, it follows the value specified by autoresize.storageLimit.
spec.fepChildCrVal.sysUsers.pgAdminPassword	<omitted>	Password for user "postgres" Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $% ^ & * () - = < >,.? ; : |/+) If this parameter is omitted, the Operator automatically generates a password. If the FEP server container uses an image with a FEPBaseVersion less than 15, be sure to specify this parameter.
spec.fepChildCrVal.sysUsers.pgdb	mydb (*)	Database to be created during provisioning Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number. Upper case letters are treated as lower case letters.

Field	Default	Details
		Maximum string length 63 characters
spec.fepChildCrVal.sysUsers.pguser	mydbuser (*)	Database user to be created during provisioning Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number. Upper case letters are treated as lower case letters. Maximum string length 63 characters This database user is the owner of the database defined in "spec.fepChildCrVal.sysUsers.pgdb" and has the role of database administrator. This user has the following privileges:. NOSUPERUSER, NOREPLICATION, NOBYPASSRLS, CREATEDB, INHERIT, LOGIN, CREATEROLE (NOCREATEROLE when spec.fepChildCrVal.sysUsers.pgSecurityUser is defined) They also belong to the following roles:. pg_monitor, pg_signal_backend
spec.fepChildCrVal.sysUsers.pgpassword	mydbpassword	Password for database user pguser Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $% ^ & * () - = < >,.? ; : |/+)
spec.fepChildCrVal.sysUsers.pgrepluser	repluser (*)	Database user for replication Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number. Maximum string length 63 characters
spec.fepChildCrVal.sysUsers.pgreplpassword	repluserpwd	Alphanumeric characters
spec.fepChildCrVal.sysUsers.tdepassphrase	tde-passphrase	TDE keystore passphrase

Field	Default	Details
spec.fepChildCrVal.sysUsers.pgRewindUser	rewind_user	Database user for Rewind Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number. Maximum string length 63 characters
spec.fepChildCrVal.sysUsers.pgRewindUserPassword	rewind_password	Password for database user rewinduser Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $% ^ & * () - = < >,.? ; : |/+)
spec.fepChildCrVal.sysUsers.pgMetricsUser		Optional user for FEPExporter connection. Can be defined afterwards Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number. Upper case letters are treated as lower case letters. Maximum string length 63 characters
spec.fepChildCrVal.sysUsers.pgMetricsUserPassword		Optional Password for metrics user. Can be defined afterwards Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $% ^ & * () - = < >,.? ; : |/+)
spec.fepChildCrVal.sysUsers.pgSecurityUser		Options Username of the security administrator user. Can be defined later. This parameter is optional, but cannot be changed or deleted after it has been defined. Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_) However, you cannot start with a number.

Field	Default	Details
		Upper case letters are treated as lower case letters. Maximum string length 63 characters
spec.fepChildCrVal.sysUsers.pgSecurityPassword		Options Defines the password for the sensitive administrator user. This parameter is optional but required when "pgSsecurityUser" is defined. Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $% ^ & * () - = < >,.? ; : |/+)
spec.fepChildCrVal.sysUsers.passwordValid		Options Manage password expiration for database users. Sets the expiration date for database user passwords defined in the FEPCluster custom resource below. - pgpassword, pgSecurityPassword In addition, if shared_preload_libraries in customPgParams is set to "fsep_operator_security" and the "CREATE ROLE" or "ALTER ROLE" command is used to update the password of a database user with login privileges and the expiration time is not defined or is longer than the specified expiration time, the operation will fail. Updates the password expiration date for database users with login privileges that have not expired when the specified expiration date is updated.
spec.fepChildCrVal.sysUsers.passwordValid.days		Options Specifies the number of days the database role is valid. Specify an integer value greater than or equal to 0. If any other value is entered, it is treated as 0 (no expiration date is set). The 'days' option is not available when using the Cloud-based Secret Management feature. When you take advantage of the Cloud-based Secret Management feature, the database user password expiration can be managed by a rotation policy

Field	Default	Details
		provided by an external secret store service.
spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName		This points to Kubernetes TLS secret that contains the certificate of Postgres user "postgres". Patroni will use this for certificate authentication. The certificate itself is stored in the key tls.crt. This field is optional.
spec.fepChildCrVal.sysUsers.pgAdminTls.caName		This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt. This field is optional.
spec.fepChildCrVal.sysUsers.pgAdminTls.sslMode	prefer	Specify the type of TLS negotiation with the server. - disable - allow - prefer - require - verify-ca - verify-full
spec.fepChildCrVal.sysUsers.pgreplUserTls.certificateName		This points to Kubernetes TLS secret that contains the certificate of Postgres user "repluser". Patroni will use this for certificate authentication. The certificate itself is stored in the key tls.crt. This field is optional.
spec.fepChildCrVal.sysUsers.pgreplUserTls.caName		This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt. This field is optional.
spec.fepChildCrVal.sysUsers.pgreplUserTls.sslMode	prefer	Specify the type of TLS negotiation with the server. - disable - allow - prefer - require - verify-ca - verify-full
spec.fepChildCrVal.sysUsers.pgRewindUserTls.certificateName		This points to Kubernetes TLS secret that contains the certificate of Postgres user "rewinduser". Patroni will use this for certificate authentication. The certificate itself is stored in the key tls.crt. This field is optional.

Field	Default	Details
spec.fepChildCrVal.sysUsers.pgRewindUserTls.caName		This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt. This field is optional.
spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode	prefer	Specify the type of TLS negotiation with the server. - disable - allow - prefer - require - verify-ca - verify-full
spec.fepChildCrVal.sysUsers.pgMetricsUserTls.certificateName		Optional This points to Kubernetes TLS secret that contains the certificate of Postgres user defined by pgMetricsUser. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.fepChildCrVal.sysUsers.pgMetricsUserTls.caName		Optional This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt.
spec.fepChildCrVal.sysUsers.pgMetricsUserTls.sslMode	prefer	Optional Specify the type of TLS negotiation when FEPExporter connects to FEP server. - disable - allow - prefer - require - verify-ca - verify-full
spec.fepChildCrVal.sysTde	(*)	Optional If the user selects a file-based TDE, you do not need to define it. Required when implementing TDE with a key management system (KMS).
spec.fepChildCrVal.sysTde.tdeType	(*)	Optional The parameter itself is optional, but required when spec.fepChildCrVal.sysTde is defined. Specify tdek.
spec.fepChildCrVal.sysTde.tdek		Optional Defines the connection information to

Field	Default	Details
		the KMS. Required when tdek is specified for spec.fepChildCrVal.sysTde.tdeType.
spec.fepChildCrVal.sysTde.tdek.targetKmsName		Specify one of the key management system names defined in kmsDefinition[*].name as the name of the key management system to use as the keystore.
spec.fepChildCrVal.sysTde.tdek.targetKeyId		Specifies the key ID (Identifier attribute in KMIP) attached to the encryption key in KMS. When you update this parameter, the Operator automatically updates the master key.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition		Specifies KMS connection information. Specify in array format. You can specify connection information for multiple KMS.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].name	(*)	The name given to the KMS (key management system name) specified in spec.fepChildCrVal.sysTde.tdek.targetKmsName. The KMS name must be a string of no more than 63 characters beginning with a-z, consisting of a-z, numbers (0 -9), and underscores. Upper and lower case letters are the same.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].type	(*)	Specifies the type of KMS. You can specify either kmip, awskms, or azurekeyvault.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].address	(*)	Specifies the host name or IP address of the KMIP server.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].port	(*)	Specifies the port of KMIP server.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].authMethod	(*)	Specifies the authentication method in KMIP server. Currently, the only possible value is cert.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].sslpassphrase		Optional Specifies the passphrase of the client certificate private key file when connecting to KMIP server. This can be omitted if no passphrase is set in the private key file.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].cert		Optional Specifies the name of the Secret/ ConfigMap containing the certificate file, etc., when cert is specified as authMethod.

Field	Default	Details
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].cert.certificateName	(*)	Specifies the TLS Secret name that contains the client certificate and private key for TLS communication with KMIP server.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].cert.caName	(*)	Specifies the ConfigMap name that contains the file name of the SSL Certificate Authority certificate. Used to verify the server certificate of the connection destination.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].profile		Specify a profile that uses AWS KMS. For more information about profile, see the official AWS documentation.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].awsKmsCredentials		Specify a Secret that contains credentials (access key id and secret access key) to AWS KMS.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].awsKmsConfig		Specify a ConfigMap that contains configuration information for the AWS KMS CLI.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].appid		Enter the application ID when using Azure Key Vault. You can get this when you create a service principal.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].tenantid		Specify tenantid when using Azure Key Vault. You can get this when you create a service principal.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].encAlgorithm		Specifies when using Azure Key Vault. See the appendix for the algorithms you can select, refer to "Available Algorithms" in the User's Guide.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].azureKeyVaultClientPassphrase		Used to authenticate to Azure Key Vault. Specifies the secret that contains the client Secret (password).
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[*].azureKeyVaultClientCert		Used to authenticate to Azure Key Vault. Specifies the Secret that contains the client certificate.
spec.fepChildCrVal.systemCertificates.key		Use spec.fep.postgres.tls specification instead.
spec.fepChildCrVal.systemCertificates.crt		Use spec.fep.postgres.tls specification instead.
spec.fepChildCrVal.systemCertificates. cacrt		Use spec.fep.postgres.tls specification instead.
spec.fepChildCrVal.autoscale.scaleout.policy	off	Specifies whether to use the automatic scale out feature and the metric to base on. Specify one of the following: - cpu_utilization (if based on CPU utilization) - connection_number (if based on number of connections) - off (without automatic scale out)

Field	Default	Details
		If omitted, off is assumed.
spec.fepChildCrVal.autoscale.scaleout.threshold	40	Specifies an integer as the threshold for performing scale out. - When cpu_utilization is specified for policy Specifies the average CPU utilization as a percentage for the threshold. If this option is omitted, 40 (40%) is assumed. - When connection_number is specified for policy Specifies the average value of the number of connections as a threshold. If you omit this option, 40 is assumed.
spec.fepChildCrVal.autoscale.scaleout.metricName	pg_capacity_connection_average	Specify this parameter if policy is connection_number. Ignored if policy is cpu_utilization. The custom metrics server must publish the average number of connections in the FEP cluster under this name. If omitted, pg_capacity_connection_average is assumed.
spec.fepChildCrVal.autoscale.scaleout.stabilizationWindowSeconds	0	This parameter controls the stability of scaling (variation in the number of replicas). Scale out is not performed unless the metric exceeds the threshold for more than the number of seconds specified for this parameter. If omitted, 0 is assumed.
spec.fepChildCrVal.autoscale.limits.maxReplicas	2	Maximum number of replicas (0 to 15) (Value out of range) Do not perform auto scale out
spec.fepChildCrVal.restore		Optional Defines to restore specified backup data stored in object storage.
spec.fepChildCrVal.restore.pgbackrestParams		Optional "|" is fixed, and the following line describes the parameters to be set in pgbackrest.conf. Specifies the object storage where the backup data is stored. If you need to use a root certificate other than the default, specify the following: repo1-storage-ca-path =/pgbackrest/storage-certs/filename The CA file is registered in ConfigMap and the ConfigMap name is listed in spec.fepChildCrVal.restore.caName.

Field	Default	Details
spec.fepChildCrVal.restore.pgbackrestKeyParams		Optional "|" is fixed, and the following line describes the parameters to be set in pgbackrest.conf. The value described by this parameter is masked with *****. Specify the parameter you want to mask, such as a password.
spec.fepChildCrVal.restore.caName		Optional Set to use a CA file other than the system default. Specifies the name of the ConfigMap created, in list format. The ConfigMap specified is mounted in /pgbackrest/storage-certs.
spec.fepChildCrVal.restore.repoKeySecretName		Optional Specifies the name of the Kubernetes Secret generated from the object storage key file. Specify in array format. The specified Secret will be mounted in /pgbackrest/storage-key.
spec.fepChildCrVal.restore.mcSpec.limits	cpu: 200m memory: 300Mi	Optional CPU and memory allocated to the container performing the restore
spec.fepChildCrVal.restore.mcSpec.requests	cpu: 100m memory: 200Mi	Optional CPU and memory allocated to the container performing the restore
spec.fepChildCrVal.restore.restoretype	latest	Optional Select the type of restore (latest or PITR).
spec.fepChildCrVal.restore.restoredate		Optional Specifies the date to restore when spec.fepChildCrVal.restore.restoretype is "PITR".
spec.fepChildCrVal.restore.restoretime		Optional Specifies the time to restore when spec.fepChildCrVal.restore.restoretype is "PITR".
spec.fepChildCrVal.restore.image		Optional Image of the container to perform the restore It is omitted by default. In this case, the URL for image is obtained from the operator container environment.
spec.fepChildCrVal.restore.imagePullPolicy	IfNotPresent	Optional

Field	Default	Details
spec.fepChildCrVal.upgrade		Optional When this field is defined, a major version upgrade is performed. However, if spec.fepChildCrVal.restore is defined, the FEPCluster build stops.
spec.fepChildCrVal.upgrade.sourceCluster		Specifies the FEPClusterCR name from which to migrate data. Required if spec.fepChildCrVal.upgrade is defined.
spec.fepChildCrVal.upgrade.mcSpec.limits	cpu: 200m memory: 300Mi	Optional Specifies the maximum number of resources to allocate to the upgrade execution container.
spec.fepChildCrVal.upgrade.mcSpec.requests	cpu: 100m memory: 200Mi	Optional Specifies the lower limit of resources allocated to the upgrade execution container.
spec.fepChildCrVal.upgrade.image		Optional By default, the URL of image is obtained from the operator container environment.
spec.fepChildCrVal.upgrade.imagePullPolicy	IfNotPresent	Optional Specifies the pull policy for the container image. - Always - IfNotPresent - Never
spec.fepChildCrVal.upgrade.source.pgAdminTls.certificateName		Optional If you do not define spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName for the data source, it points to the Kubernetes TLS secret that contains the certificate for the Postgres user "postgres" in the data source. If the data source FEP has set the authentication method for the upgrade execution container to "cert", then the upgrade execution container uses the certificate defined as secret.
spec.fepChildCrVal.upgrade.destination.pgAdminTls.certificateName		Optional If you have not defined the spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName of the newly created FEPCluster, it points to the

Field	Default	Details
		Kubernetes TLS secret that contains the certificate of the Postgres user "postgres" in the data source. If you create a new FEP with the "cert" authentication method for the upgrade execution container, the upgrade execution container uses the certificate defined as secret.
spec.fepChildCrVal.upgrade.storage		Optional Defines the storage for storing dump files.
spec.fepChildCrVal.upgrade.storage.storageClass		Optional If omitted, the default storage class for your environment is used.
spec.fepChildCrVal.upgrade.storage.size	2Gi	Optional Specifies the size of the storage to store the dump file.
spec.fepChildCrVal.upgrade.storage.accessModes	ReadWriteOnce	Optional accessModes for store the dump file Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]
spec.fep.remoteLogging.enable		Set to true to forward logs from fluentbit to fluentd
spec.fep.remoteLogging.image		Optional Fluentbit image to be used. If not specified, Operator will use the latest version that is supported by the Operator.
spec.fep.remoteLogging.pullPolicy	IfNotPresent	Optional
spec.fep.remoteLogging.fluentdName		Optional The name of the FEPLogging CR to which logs are transferred. Specify this option to use the FEPLogging function to transfer logs.
spec.fep.remoteLogging.tls.certificateName		Optional Kubernetes secret name which holds fluentbit certificate. FEPLogging will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.fep.remoteLogging.tls.caName		Optional Kubernetes configmap which holds cacert of Fluentd to which fluentbit will use to perform MTLS.

Field	Default	Details
spec.fep.remoteLogging.mcSpec.limits.cpu	50m	Optional CPU allocation limit for fluentbit.
spec.fep.remoteLogging.mcSpec.limits.memory	60Mi	Optional Memory allocation limit for fluentbit.
spec.fep.remoteLogging.mcSpec.requests.cpu	10m	Optional CPU allocation request for fluentbit.
spec.fep.remoteLogging.mcSpec.requests.memory	5Mi	Optional Memory allocation request for fluentbit.
spec.fep.remoteLogging.fluentbitParams.memBufLimit	5MB	Optional Defines the Mem_Buf_Limit in Fluentbit. This will affect all sections that use this parameter.
spec.fep.remoteLogging.fluentbitConfigSecretRef		Optional Specifies the name of the secret containing fluent-bit.yaml when using the log transfer feature with remote logging. If fluentbitConfigSecretRef is not defined, or if fluentbitConfigSecretRef is defined but the referenced secret does not exist, the operator creates a default Secret <fep-cluster>-fluent-bit-conf and updates this parameter with <fep-cluster>-fluent-bit-conf. If the referenced secret exists, the named secret is mounted to fep-logging-fluent-bit under/fluent-bit/etc.
spec.fep.remoteLogging.awsCredentialSecretRef		Optional Specify the name of the Secret that contains authentication information to the AWS service. Authentication information stores configuration files and authentication information files. The configuration file must be named "config" and the credentials file must be named "credentials". If the referenced secret exists, the named secret is mounted to fep-logging-fluent-bit under/fluent-bit/aws.
spec.fepChildCrVal.secretStore.csi.providerName		Optional Provider name. Can be one of the following: Azure/AWS/GCP/Vault.

Field	Default	Details
		Must be "Azure" or "azure" in case of azure provider
spec.fepChildCrVal.secretStore.csi.azureProvider.credentials		Optional Secret created by User that contains the required credentials to connect to Azure keyvault
spec.fepChildCrVal.secretStore.csi.azureProvider.tenantid		Optional Tenant id where keyvault is created
spec.fepChildCrVal.secretStore.csi.azureProvider.keyvaultName		Optional Name of the keyvault where secrets are stored
spec.fepChildCrVal.secretStore.csi.azureProvider.fepSecrets		Optional List of the parameters and their corresponding secret created in the Vault Eg: <fep parameter name>: <secret in keyvault>
spec.fepChildCrVal.secretStore.csi.azureProvider.fepCustomCert		Optional Only defined when logical replication feature is enabled
spec.fepChildCrVal.secretStore.csi.awsProvider.region		Optional AWS Region where EKS cluster is created
spec.fepChildCrVal.secretStore.csi.awsProvider.roleName		Optional Role Name for the IAM trust policy
spec.fepChildCrVal.secretStore.csi.awsProvider.fepSecrets		Optional List of the parameters and their corresponding secret created in the Vault Eg: <fep parameter name>: <secret in keyvault>
spec.fepChildCrVal.secretStore.csi.awsProvider.fepCustomCert		Optional Only defined when logical replication feature is enabled
spec.fepChildCrVal.secretStore.csi.gcpProvider.credentials		Optional Secret created by User that contains the required credentials to connect to GCP Secret Manager
spec.fepChildCrVal.secretStore.csi.gcpProvider.fepSecrets		Optional List of the parameters and their corresponding secret created in the Vault

Field	Default	Details
		Eg: <fep parameter name>: <secret in keyvault>
spec.fepChildCrVal.secretStore.csi.gcpProvider.fepCustomCert		Optional Only defined when logical replication feature is enabled
spec.fepChildCrVal.secretStore		Optional Not required to be defined if user opts to store all secrets in kubernetes environment
spec.fepChildCrVal.secretStore.csi.vaultProvider.roleName		Optional roleName created by user in the Vault
spec.fepChildCrVal.secretStore.csi.vaultProvider.vaultAddress		Optional Address of the vault that is accessible from the FEP environment
spec.fepChildCrVal.secretStore.csi.vaultProvider.fepSecrets		Optional List of the parameters and their corresponding secret created in the Vault Eg: <fep parameter name> : </path/to/secret/secretName> in vault>
spec.fepChildCrVal.secretStore.csi.vaultProvider.fepCustomCert		Optional Only defined when logical replication feature is enabled

Note

- (*) - These parameters can be specified only at creation time and should not be changed. Any change to these parameters will be ignored and will not have any effect on FEP cluster functioning.
- (**) - The storage volumes size can be increased provided underlying storage supports the operation. Optional volumes can be specified only at initial FEP cluster creation. If an optional volume is added later, operator will ignore it and no action will be taken.
- User should do or remove unsupported CR changes manually.
- spec.fep.postgres.tls CR specification should be used instead of spec.fepChildCrVal.systemCertificates. The lateral spec can still be used, however spec.fep.postgres.tls gives better flexibility to control MTLS access of the cluster.
- Either spec.fep.postgres.tls specification (old specification) or spec.fepChildCrVal.systemCertificates should be used. They should not be used interchangeable.
- Server certificate specified under spec.fep.postgres.tls can be rotated by changing the secret and executing reload (e.g. using FEPAction); however for others specified in the CR, it is required to do restart of the PoDs
- When spec.ldap.caConfigMapRef is defined, and the referenced configmap exist, the named configmap will be mounted on fep-patroni under /tls/ldap.
- When the referenced secret spec.ldap.ldapconfSecretRef exists, the named secret will be mounted on fep-patroni under /etc/openldap. The environment variable LDAPCONF=/etc/openldap/ldap.conf will be exported to the fep-patroni container. If this key is not defined, Operator will create and mount default secret <fep-cluster>-ldapconf with the following content.

- When the referenced configmap spec.ldap2pg.ldap2pgymlConfigMapRef exists, the named configmap will be mounted on fep-patroni under /tmp/.config. If the secret does not exist, Operator will create that named secret with the following content.

While in running state - following value will dynamically appear in the FEPCluster to reflect the cluster status

Field name	Details
status.fepStatus.fepClusterReady	Will be true or false to reflect if the whole cluster is ready. Kubernetes cluster information is fetched to check number of instances 'READY' & 'RUNNING' is equal to number of Configured instances.

Note

"fepClusterReady" flag will be set at first FEPCluster creation time only. fepClusterReady flag does not participate in the next reconcillation loop)

1.2 Custom Resource Parameters

This section explains the Custom Resource Parameters.

1.2.1 FEPCluster Custom Resource Parameters

Category	Details
CRD Name	`FEPCluster`
`Definition`	`///`

Operations	`Create: kubectl create -f fepcluster.yaml`
	`Delete: kubectl delete fepcluster <clusername>`
	`Update: kubectl apply -f fepcluster.yaml`
	`List: kubectl get fepcluster`

FEPCluster CR Example

```
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: new-fep
  namespace: new-fep
spec:
  fep:
///
        wuC4
        -----END CERTIFICATE-----
```

It should also be noted that all the passwords / passphrase and certificates will be masked after the creation of the CR. This includes

- Also, inital pgAdminPassword: admin-password
- pgpassword: mydbpassword
- pgreplpassword: repluserpwd
- tdepassphrase: tde-passphrase
- pgRewindPassword: rewind_passoword (Optional - if defined)

- pgMetricsPassword: metrics_password (Optional - if defined)
- pgSecurityPassword (if defined)
- sslpassphrase under sysTde.tdek.kmsDefinition (if defined)
- certificate.key
- certificate.crt
- certificate.cacrt

Values of child CRs at the time of initial deployment of cluster, are stored in FEPCluster under fepChildCrVals, e.g. for Server certificates, Configuration of FEP, User details.

All fields for FEPCluster CR and its child CRs should be managed through FEPCluster CR only. Operator will reflect the changes to respective child CR to be processed. The fields that not allowed to change will not be reflected from parent to child CR and hence will not have any affect.

1.2.2 FEP Cluster Configuration

Configuration of all aspects of FEP Cluster is done through FEPCluster CR only.

All fields for FEPCluster CR and its child CRs should be managed through FEPCluster CR only. Operator will reflect the changes to respective child CR to be processed. The fields that not allowed to change will not be reflected from parent to child CR and hence will not have any affect. Refer to "1.1 FEPCluster Parameter" for details.

All child CRs are marked as internal objects in RedHat OCP and will not appear on console. However, it can be checked on command line using oc or kubectl commands.

Following table shows Child CRs of FEPCluster CR and respective sections in parent CR related to given child CR.

Configuration changes are made in these sections will update allowable fields only in corresponding child CR.

Child CR Name	Relevant sections in FEP Cluster CR
FEPBackup	spec.fepChildCrVal.backup
FEPCert	spec.fepChildCrVal.systemCertificates
FEPConfig	spec.fepChildCrVal.customPgAudit spec.fepChildCrVal.customPgHba spec.fepChildCrVal.customPgParams
FEPUser	spec.fepChildCrVal.sysUsers
FEPVolume	spec.fepChildCrVal.storage

1.2.3 FEPConfig Child Custom Resource Parameters

Field	Default	Details
metadata.name	<same-as-in-FEPCluster>	This value is inherited from parent FEPCluster CR
metadata.namespace	<same-as-in-FEPCluster>	This value is inherited from parent FEPCluster CR
spec.customPgAudit	All line specified in spec.fepChildCrVal.customPgAudit of FEPCluster CR	Audit rules can be updated in this section. Requires restart. Note: initial values inherited once only at start. Changes to FEPConfig directly
spec.customPgHba	All line specified in spec.fepChildCrVal.customPgHba of FEPCluster CR	pg_hba rules can be added in this section Note: Inherited once at start. Changes to FEPConfig directly

Field	Default	Details
spec.customPgParams	All line specified in spec.fepChildCrVal.customPgParams of FEPCluster CR	All postgres parameters are listed here to overwrite defaults. Note: Inherited once at start. Changes to FEPConfig directly
spec.replicationSlots		Optional: Details of replication slots if defined in FEPCluster

Example of FEPConfig CR created

```
apiVersion: fep.fujitsu.io/v1
kind: FEPConfig
metadata:
  name: new-fep-19ncfg
  namespace: cfg-expt
spec:
  sysExtraLogging: false
  customPgAudit: |
    # define pg audit custom params here to override defaults.
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'
    [output]
    logger = 'auditlog'
    log_directory = '/database/log/audit'
    log_truncate_on_rotation = on
    log_filename = 'pgaudit-%a.log'
    log_rotation_age = 1d
    log_rotation_size = 0
    [rule]

  customPgHba: |
    # define pg_hba custom rules here to be merged with default rules.
    # TYPE     DATABASE        USER        ADDRESS        METHOD
  customPgParams: |+
    # define custom postgresql.conf parameters below to override defaults.
    # Current values are as per default FEP deployment
    shared_preload_libraries='pgx_datamasking,pgaudit,pg_prewarm,pg_stat_statements'
    session_preload_libraries='pg_prewarm'
    max_prepared_transactions = 100
    max_worker_processes = 20
    max_connections = 100
    work_mem = 1MB
    maintenance_work_mem = 20MB
    shared_buffers = 128MB
    effective_cache_size = 384MB
    checkpoint_completion_target = 0.8
    pgx_global_metacache = 10MB
    temp_buffers = 10MB

    # tcp parameters
    tcp_keepalives_idle = 30
    tcp_keepalives_interval = 10
    tcp_keepalives_count = 3

    # logging parameters in default fep installation
    # if log volume is not defined, log_directory should be
    # changed to '/database/userdata/data/log'    log_directory = '/database/log'
    log_filename = 'logfile-%a.log'
    log_file_mode = 0600
```

```
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
log_checkpoints = on
log_line_prefix = '%e %t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h'
log_lock_waits = on
log_autovacuum_min_duration = 60s
logging_collector = on
pgaudit.config_file= '/opt/app-root/src/pgaudit-cfg/pgaudit.conf'
log_replication_commands = on
log_min_messages = WARNING
log_destination = stderr

# wal_archive parameters in default fep installation
archive_mode = on
wal_level = replica
max_wal_senders = 10
wal_keep_segments = 64
wal_sender_timeout = 60s
track_activities = on
track_counts = on
```

1.2.4 FEPUser Child Custom Resource Parameters

Field	Default	Details
metadata.name	<same-as-in-FEPCluster>	This value is inherited from parent FEPCluster CR
metadata.namespace	<same-as-in-FEPCluster>	This value is inherited from parent FEPCluster CR
spec.pgAdminPassword	spec.fepChildCrVal.users.pgAdminPassword of FEPCluster CR	postgres superuser password. Masked once secret is created/changed Note: initial values inherited once only at start. Changes to FEPUser directly
spec.pgdb	spec.fepChildCrVal.users.pgdb of FEPCluster CR	Name of a user database Note: Created once only at start. Cannot be changed
spec.pgpassword	spec.fepChildCrVal.users.pgpassword of FEPCluster CR	Password for superuser for user database pgdb. Masked once secret is created/changed Note: initial values inherited once only at start. Changes to FEPUser directly
spec.pguser	spec.fepChildCrVal.users.pguser of FEPCluster CR	Name of a user database Note: Created once only at start. Cannot be changed
spec.pgrepluser	spec.fepChildCrVal.users.pgrepluser of FEPCluster CR	Name of a database user for replication
spec.pgreplpassword	spec.fepChildCrVal.users.pgreplpassword of FEPCluster CR	Password for pgrepluser
spec.tdepassphrase	spec.fepChildCrVal.users.tdepassphrase of FEPCluster CR	Passphrase for encrypting/decrypting keystore file which contains the TDE encryption key
spec.pgRewindUser	rewind_user	Database user for Rewind

Field	Default	Details
spec.pgRewindUserPassword	rewind_password	Password for database user rewinduser
spec.pgMetricsUser	spec.fepChildCrVal.sysUsers.pgMetricsUser	Optional Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.
spec.pgMetricsPassword	spec.fepChildCrVal.sysUsers.pgMetricsPassword	Optional Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.
spec.pgAdminTls	spec.fepChildCrVal.sysUsers.pgAdminTls	Optional section Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.
spec.pgrepluserTls	spec.fepChildCrVal.sysUsers.pgrepluserTls	Optional section Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.
spec.pgRewindUserTls	spec.fepChildCrVal.sysUsers.pgRewindUserTls	Optional section Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.
spec.pgMetricsUserTls	spec.fepChildCrVal.sysUsers.pgMetricsUserTls	Optional section Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.

Example of FEPUser CR created

```
apiVersion: fep.fujitsu.io/v1
kind: FEPUser
metadata:
  name: new-fep-19n
  namespace: testswatiproject
spec:
  pgAdminPassword: '***************'
  pgdb: mydb
  pgpassword: '***************'
  pgreplpassword: '***************'
  pgrepluser: repluser
  pguser: mydbuser
  tdepassphrase: '***************'
  sysExtraLogging: false
  pgRewindUser: rewind_user
  pgRewindUserPassword: rewind_password
  pgAdminTls:
    certificateName: admin-client-certs-secret
    caName: admin-ssl-rootcert-configmap
    sslMode: prefer
  pgrepluserTls:
    certificateName: repluser-client-certs-secret
    caName: repluser-ca-name-configmap
    sslMode: prefer
  pgRewindUserTls:
    certificateName: rewinduser-client-certs-secret
    caName: rewinduser-ca-name-configmap
    sslMode: prefer
```

Note

- Password and Passphrase are masked in output from CR. The original values can still be found in the respective Kubernetes secrets and configmaps.
- TDE is enabled by default with given tdepassphrase and must have a value.
- TDE is enabled by using the key tdepassphrase with the desired passphrase.Do not remove this key once TDE is enabled. Otherwise, the database may go into a crash loop. If the Cluster is running on Async Replication and a failover/switchover occured during the crash loop, there could be data lost. The team is looking at preventing the deletion of this passphrase from Operator even if customer tries to remove it in customer resource.
- Database users and their passwords managed by the FEPUser CR should not be changed in the SQL interface. Inconsistencies with the information managed by the operator can cause problems with operator operation. If you make changes in the SQL interface, use the SQL interface again to restore the original state.

1.2.5 FEPVolume Child Custom Resource Parameters

1.2.5.1 Create Volumes

Volumes for the cluster nodes(pods) are initially created in accordance with the values set in fepChildCrVal’ storage section of the parent FEPCluster CR.

The parent FEPCluster CR creates a child FEPVolume CR with the respective startup values and the relevant controller(FEPColume Controller) takes care of creating the required volumes. After initial FEPCluster create, new volume cannot be added later and storageClass or accessModes can not be changed.
Only size of an initially created volume can be changed if and only if underlying storageClass supports dynamic change of size.

Below is the schema of the FEPVolume CR:

Field	Mandatory	Sub-Field	Default	Description
archivewalVol	No	size storageClass accessModes	1Gi Defaults to platform default if omitted Defaults to ReadWriteOnce if omitted	Size of the volume,expandable later SC is only set at start Access mode is only set at start Additional details in section 3.2
backupVol	No	size storageClass accessModes	2Gi Defaults to platform default if omitted Defaults to ReadWriteOnce if omitted	-do-
dataVol	Yes	size storageClass accessModes	2Gi Defaults to platform default if omitted Defaults to ReadWriteOnce if omitted	-do-
logVol	No	size storageClass accessModes	1Gi Defaults to platform default if omitted	-do-

Field	Mandatory	Sub-Field	Default	Description
			Defaults to ReadWriteOnce if omitted	
tablespaceVol	No	size storageClass accessModes	512Mi Defaults to platform default if omitted Defaults to ReadWriteOnce if omitted	-do-
walVol	Yes	Size storageClass accessModes	1200Mi Defaults to platform default if omitted Defaults to ReadWriteOnce if omitted	-do-

1.2.5.2 Delete Volumes

Equivalent Kubernetes command: kubectl delete FEPVolume <cr_name>

This operation will remove all the PVCs and possibly PVs depending on the default reclaimPolicy of the storageclass used per volume.

With right backup and restore integration by customer, they may not need volumes to be persisted.

Note

Do not delete this CR unless the Cluster has been removed.

Example of FEPVolume CR created

```
apiVersion: fep.fujitsu.io/v1
kind: FEPVolume
metadata:
  name: new-fep-19n
  namespace: testswatiproject
spec:
  archivewalVol:
    size: 1Gi
  backupVol:
    size: 2Gi
  dataVol:
    size: 2Gi
  logVol:
    size: 1Gi
  tablespaceVol:
    size: 512Mi
  walVol:
    size: 1Gi
  selectedVolList:
  - name: data
  - name: tablespace
  - name: wal
  - name: log
  sysExtraLogging: false
```

1.2.6 FEPCert Child Custom Resource Parameters

1.2.6.1 Create/ Update Certificates

Certificate secret for the FEP cluster is initially created in accordance with the values set in fepChildCrVal’ certs section of the parent FEPCluster CR.

Below is the schema of the FEPCert CR:

Field	Default	Description
cacrt	Defaults to dummy self signed crt from parent FEPCluster CR	Can be replaced with customer’s own CA cert
crt	Defaults to dummy self signed crt from parent FEPCluster CR	Can be replaced with customer’s own trusted cert
key	Defaults to dummy key from parent FEPCluster CR	Can be replaced with customer’s own key

By default, Operator will create Kubernetes secrets to store the CA Cert, Server Cert and Key file. These files are exposed under the mount point /fep-certs in the container. The default FEPCluster template will also set the following postgres parameters in postgresql.conf.

```
ssl = on
ssl_cert_file = '/fep-certs/fep.crt'
ssl_key_file = '/fep-certs/fep.key'
ssl_ca_file = '/fep-certs/ca.crt'
```

It should also be possible to change the certificates by end user, by changing ALL key, crt and cacrt. However, user will need to restart the cluster to let change take effect.

1.2.6.2 Delete Certificates

Equivalent Kubernetes command: kubectl delete FEPCert <cr_name>

This operation will remove the secret containing the TLS Certificates and keys for the cluster.

Below is an example CR for certificates to be used by FEP server container

```
apiVersion: fep.fujitsu.io/v1
kind: FEPCert
metadata:
  name: new-fep
  namespace: ansible-operator-poc
spec:
  key: |-
    -----BEGIN RSA PRIVATE KEY-----
    MIIEowIBAAKCAQEA4AI33yvHZws+jta6qpV6wzJqF8odIfTIpCfbrVcUUtLFKJ1I
    2e4SceTKi6O3C/I1XuvWlpng5IO65+fQQLO06z1/AuQT78YUn/Wlm9x1aHVsv4AN
    B5JWWqDOjrRT3o7nRPGXfilabP0rGE2mJJcVR9nExJ3IeaktgT3sb8YlXvtchyYp
    mjdbfxabTz07ig0+6/cwKoRRxOK8Uf7f5euE0cI/490J6r5Rs4lgD8sIQNCUFlTF
    YvmAH7gcdssSFBt8NPlUATHEsoFmlW0DKCJWNhTLOht+s6L/1zwTHLjPG2pdkG6W
    dgmu5H2pDml8CDNLDv98Aj7i+I5SRKKcVPlnuQIDAQABAoIBAFPQYKlOzw/+BA0b
    yMIUpdctIMb/54CR/xR0mVw1DbSjigNVPjHUQvB8Y1B2FAITQObgJO06bAv0QdWN
    Rb0/v/yYiNJDFjaLjaIAHlO/2+oWrXbFaZqgpVDJhB+e1xaZr2x7XGxm+p925k30
    l6pvIRY+I8JRKvZiV1VZHwL/R3JOtPr++xMZtLVjVOI+f+ySqJ+TZHuAjm49EKxj
    cEmmJ28b7QcziXsvKy00f+zbqLIBKXQdZAFU5eEr1BsDRXdRW+Kf0XIvftuy4BJZ
    voKT+VGhEvF/qysswL4+6IAO6tpuYnnM0Y2d3sOGoWPkTcQK0MekYKzL/WmtCjNs
    9hodJtECgYEA5EWyhEOf4uOKe5TDp697UCUvXLoOR58FDe/S8XNvScn29jjOkqIg
    OMoqo9xAkJTNTzqn5UUdt1x/pgM2NxlPLFijrc0zQlX3SoOO2ryDd9WNi7YKtN16
    KJqa536WeZu2OEbuAZ+S3GALVy1RPeTNPnUOmKnF06DjDUGzLNCZy10CgYEA+zfw
    952DWuz1U0Z4wvAEqqcgUKXPKrkTXV/iUnjkDkrLYVr0ZofDNTXrdHl+UedFmaOC
    cieZn6DNhcdz5tKtyysGMH3g/qs9PfoGUngvcXsy0Egk04l3x1jc8TTCLqXZXYaQ
```

```
    HMsx51n+R58oncPtzYSUOr9qQ6PbC2CstTbFJA0CgYEAjGEsUliAB/jknfEzjXjG
    PdhQUxb8VyE864Az2lah9t/kJzFyIAziAeqZ5GE7t247AGFTBRTHHI8e1Qoemi3P
    Wbc9GVIbFs1lIYbcIDpUIyrKPEP8O5QEXtoNLxXTFgAjRGKiVY87spjCAJ+W2ZhO
    e/1it5GYXfgQCYQA2yuBmOUCgYANRkR2YR1axaCk+NlSu6oTdmdPu6M5x7PNQE7O
    OtMaKjua9lppvIzFGAdMDUtueoEEAE7ZR1xnwfB6PDLUpJdIYAqgr1YfPt8qkjaZ
    Tv56yZ7CwL0pbF8m6nwqRrZoDp1wwraEvvvxFKFKGY/k3kCHlpTakdjEoDjn3gDi
    RnWeVQKBgCEneMSzucei5LRppRtRaJw/Btll8qlPMlX3W7dxQ3cLwpmLOn0m51Fp
    PIZ44zYK8R6fu4+/sSrlfaIg86Ugeufp6YNxyNROKxUGza5vDIu5OftwWtBeg+UK
    Z8lLWNdX6pp7WMujmF3H1DrkBbauYMUKZ4UxUYtelgHERMePIxwb
    -----END RSA PRIVATE KEY-----
crt: |-
    -----BEGIN CERTIFICATE-----
    MIIDUTCCAjmgAwIBAgIRAMocW3qMoHrD6qRvMPppMkMwDQYJKoZIhvcNAQELBQAw
    NzEQMA4GA1UECgwHRnVqaXRzdTEjMCEGA1UEAwwaRkVQIFJvb3QgQ0EgZm9yIEt1
    YmVybmV0ZXMwHhcNMjEwMjA2MDQzMjM2WhcNMjYwMjA1MDQzMjM2WjA/MRAwDgYD
    VQQKEwdGdWppdHN1MSswKQYDVQQDEyJGVUpJVFNVIEVudGVycHJpc2UgUG9zdGdy
    ZXMgU2VydmVyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4AI33yvH
    Zws+jta6qpV6wzJqF8odIfTIpCfbrVcUUtLFKJ1I2e4SceTKi6O3C/I1XuvWlpng
    5IO65+fQQLO06z1/AuQT78YUn/Wlm9x1aHVsv4ANB5JWWqDOjrRT3o7nRPGXfila
    bP0rGE2mJJcVR9nExJ3IeaktgT3sb8YlXvtchyYpmjdbfxabTz07ig0+6/cwKoRR
    xOK8Uf7f5euE0cI/490J6r5Rs4lgD8sIQNCUFlTFYvmAH7gcdssSFBt8NPlUATHE
    soFmlW0DKCJWNhTLOht+s6L/1zwTHLjPG2pdkG6Wdgmu5H2pDml8CDNLDv98Aj7i
    +I5SRKKcVPlnuQIDAQABo1AwTjAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUH
    AwIwDAYDVR0TAQH/BAIwADAfBgNVHSMEGDAWgBQcwrrUO0u+FhIUuVdrDRCQRsi6
    ZjANBgkqhkiG9w0BAQsFAAOCAQEAm5dxBoI9pScOCvRAchg4CprdRDSJb9K6yB3O
    nCAxnM47iHeXnY3WlnI388kHu8DU7O4ba1tJbGs3KY9KzioPk43pU12jWkO1onoF
    +mTDjx/Ef1cYWA9r5q/LtgTa6Q2sxV4O2x67QW82aAnaxO34dV5zWCPIvAoovZBV
    HRT+BgCg3r2vD1RGKK2nl1aYJtWhO1SZubam+VttdZ/vbM9oOJctxmImsEtBXjkY
    KteePdQtLL5o03JhyXWyRshCq+HMmKf2KgyY8gvydGcP4eLQdBWcW40LcnVq6UjT
    0kJycJEKngMVademq1ZWHGaiYB7hyT6GhgIcHUJ2cKrPgbEh1Q==
    -----END CERTIFICATE-----
cacrt: |-
    -----BEGIN CERTIFICATE-----
    MIIDTzCCAjegAwIBAgIUYssQ8I74US5g+1+Z7CHuaDgkZnEwDQYJKoZIhvcNAQEL
    BQAwNzEQMA4GA1UECgwHRnVqaXRzdTEjMCEGA1UEAwwaRkVQIFJvb3QgQ0EgZm9y
    IEt1YmVybmV0ZXMwHhcNMjEwMjA2MDM1MjI4WhcNMzEwMjA0MDM1MjI4WjA3MRAw
    DgYDVQQKDAdGdWppdHN1MSMwIQYDVQQDDBpGRVAgUm9vdCBDQSBmb3IgS3ViZXJu
    ZXRlczCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMs97gUF0xkUzCgL
    7MiiDju9ySr/ziwjvcYU7jA9ML+SLmftMs3HtcYbAmSntqI+MDBSR/FAJTOoytuT
    pV+mCFcGj2YAjDpliHPeNcUpbryy4YMChF3+MovkIwGCksxo5rhiWhGmoBYpA48P
    4Xe8SPlzqMzhFvNeKzyiUhvjutS2Y1Ss38lsTaurFPx64vQ2PaC54XzdwMptXtpb
    tYmWSzCpJWwxZ6lF3vitdA2w0tnBWNyctAd0+RIM/fvArxiIqseAux9t0uogm5to
    lRIhvekuxOpXBPEqtIYQ4j9XUW2JH8vUDnzPkPvjrq+A3Ug8OyyfGVrW7+VYXozu
    c4aP7P0CAwEAAaNTMFEwHQYDVR0OBBYEFBzCutQ7S74WEhS5V2sNEJBGyLpmMB8G
    A1UdIwQYMBaAFBzCutQ7S74WEhS5V2sNEJBGyLpmMA8GA1UdEwEB/wQFMAMBAf8w
    DQYJKoZIhvcNAQELBQADggEBAMDwD85RAaWEBptFgLzKw+9xEUy1vcZaonAuA1qc
    T342XTueyAugxkC11HwdCGgGS34VyctfMGqj4AW6pA2ez4tLrbOps4DmV4sw8uBL
    8pgRDgfly3ob9FEg2wa0hmrwX9jH5Bt4vySUE2785uPAqaspT2UNtTBxS85BUi1T
    sKId2Rtil6an281Z81wyWVI6Jm2D4MG0mbsiGcTPlCtdg/UljvDYymXlAvd4vNhl
    k9hDa13TgDqJKgKdTIcmZoNQdpEVgFcO0h9AEUy5AuLqxHq60dLfZ6ESGPlMI7Lm
    i4PzYbCnBmOe+7TnHcPSyrnehs66Ik+oifRd82eYS7vKjFw=
    -----END CERTIFICATE-----
```

Note

This approach of specifying FEPCerts is getting deprecated. Should follow Secrets as referred in section to configure Certs for Server, Patroni and Users.

1.2.7 FEPBackup Child Custom Resource Parameters

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Fixed
kind	FEPBackup	Fixed
metadata.name	<clustername>	Enter the CR name.
spec.pgbackrestParams	"|"	"|" It is fixed, and the parameter set in pgbackrest.conf is described from the line below.
spec.schedule.num	Integer	Number of schedules to set The maximum number of backup schedules is 5.
spec.scheduleN.schedule	-	Write the date and time of the Nth schedule in cron format. The date and time is UTC time.
spec.scheduleN.type	full/incr	full: Perform a full backup (Back up the contents of the database cluster). incr ー Perform an incremental backup (Back up only the database cluster files that were changed to the last backup migration).
spec.preScript	" "	This parameter must specify a default value.
spec.postScript	" "	This parameter must specify a default value.

Example of FEPBackup CR created

```
apiVersion: fep.fujitsu.io/v1
kind: FEPBackup
metadata:
  name: fepcluster-backup
spec:
  schedule:
    num : 2
  schedule1:
    schedule : "0 0 1 * *"
    type : "full"
  schedule2:
    schedule : "0 0 1-6 * *"
    type : "incr"
  preScript: " "
  postScript: " "
  pgbackrestParams: |
    # define custom pgbackrest.conf parameters below to override defaults.
    [global]
    repo1-retention-full = 30
    repo1-retention-full-type = time
...
```

1.2.8 FEPRestore Custom Resource Parameters

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Fixed
kind	FEPRestore	Fixed
metadata.name	-	Enter the CR name.
spec.fepVersion		Optional

Field	Default	Details
		To use FEPRestore image of given version. Possible values: 12, 13, 14, 15, 16 & 17
spec.image	<current-released-image>	FEP restore container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-17-restore:ubi9-17-1.0 It is optional. Image is left blank by default. In such a case, it will pick up URL of image from operator container environment. If you specify the image, Operator will take that image to deploy container
spec.imagePullPolicy	IfNotPresent	
spec.mcSpec.limits	cpu: 0.2 memory: "300Mi"	
spec.mcSpec.requests	cpu: 0.1 memory: "200Mi"	
spec.fromFEPcluster	<from_clustername>	The name of the FEPcluster from which to restore
spec.toFEPcluster	<to_clustername>	Specifies the name of the FEP cluster to restore to. When restoring to an existing cluster, do not specify the line of this parameter.
spec.restoretype	latest/PITR	latest - Restore Latest State PITR - Date-Time Restore
spec.restoredate	-	If spec.restoretype is PITR, specify the day of PITR (UTC) in YYYY-MM-DD format Be sure to use single quotes. Example) '2020-11-25'
spec.restoretime	-	If spec.restoretype is PITR, specifies the PITR time (UTC) in HH: MM: SS format Be sure to use single quotes. Example) '02:50:43'
spec.restoreTargetRepo		Optional If you are using multiple repositories, specify the repository from which to restore. If not specified, "1" is substituted.
spec.changeParams.fepChildCrVal.backup.pgbackrestParams		Optional Specify this to change the spec.fepChildCrVal.backup.pgbackrestParams setting in FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.backup.pgbackrestKeyParams		Optional Specify this to change the spec.fepChildCrVal.backup.pgbackrestKeyParams setting in FEPClusterCR when restoring to a new DB cluster.

Field	Default	Details
spec.changeParams.fepChildCrVal.backup.caName		Optional Specify if you want to change the spec.fepChildCrVal.backup.caName setting of FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.backup.repoKeySecretName		Optional Specify if you want to change the spec.fepChildCrVal.backup.repoKeySecretName setting of FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.backupVol		Optional Specify this to change the spec.fepChildCrVal.storage.backupVol setting in FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.archivewalVol		Optional Specify this option to change the spec.fepChildCrVal.storage.archivewalVol setting for FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.dataVol		Optional Specify this to change the spec.fepChildCrVal.storage.dataVol setting for FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.walVol		Optional Specify this to change the spec.fepChildCrVal.storage.walVol setting for FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.logVol		Optional Specify this to change the spec.fepChildCrVal.storage.logVol setting for FEPClusterCR when restoring to a new DB cluster.
spec.changeParams.fepChildCrVal.storage.tablespaceVol		Optional Specify this to change the spec.fepChildCrVal.storage.tablespaceVol setting for FEPClusterCR when restoring to a new DB cluster.

Example of FEPRestore CR created

```
apiVersion: fep.fujitsu.io/v1
kind: FEPRestore
metadata:
  name: feprestore
spec:
  mcSpec:
    limits:
      cpu: 200m
      memory: 300Mi
    requests:
      cpu: 100m
      memory: 200Mi
  fromFEPcluster: fepcluster1
  toFEPcluster: fepcluster2
```

```
  restoretype: latest
  imagePullPolicy: IfNotPresent
```

Example of Point-In-Time-Recovery using FEPRestore CR

```
apiVersion: fep.fujitsu.io/v1
kind: FEPRestore
metadata:
  name: feprestore
spec:
  mcSpec:
    limits:
      cpu: 300m
      memory: 700Mi
    requests:
      cpu: 200m
      memory: 512Mi
  fromFEPcluster: fepclusterA
  toFEPcluster: fepclusterB
  restoretype: PITR
  restoredate: 2020-11-25
  restoretime: 02:50:43
  imagePullPolicy: IfNotPresent
```

Note

Upon successful completion, custom resources in FEPRestore are automatically deleted.

You can specify spec.changeParams in the FEPRestore custom resource to modify the definition from the source to build a new FEPCluster and restore the data.

This allows you to mount another storage in the new cluster, for example, to expand the PVC size, even if you are using storage that does not support PVC extensions.

Example of a FEPRestore Custom Resource for Modifying the Storage Class and Storage Capacity of a FEPCluster "source-cluster"

```
apiVersion: fep.fujitsu.io/v1
kind: FEPRestore
metadata:
  name: feprestore
spec:
  mcSpec:
    limits:
      cpu: 300m
      memory: 700Mi
    requests:
      cpu: 200m
      memory: 512Mi
  fromFEPcluster: source-cluster
  toFEPcluster: new-cluster
  restoretype: latest
  changeParams:
    fepChildCrVal:
      storage:
        dataVol:
          size: 50 Gi
          storageClass: new-storage
```

1.2.9 FEPPgpool2 Custom Resource Parameters

Equivalent Kubernetes command: kubectl create FEPPgpool2

This operation will create a PGPool2 with supplied information.

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Fixed
kind	FEPPgpool2	Fixed
metadata.name	-	List the name of the FEP Pgpool2 container.
metadata.namespace	-	Specify the namespace of the environment where you want to deploy the operator.
spec.fepVersion		Optional To use FEPPgpool2 image of given version. Possible values: 12, 13, 14, 15, 16 & 17
spec.image	<current-released-image>	FEPPgpool2 container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-17-pgpool2:ubi9-17-1.0 It is optional. Image is left blank by default. In such a case, it will pick up URL of image from operator container environment. If you specify the image, Operator will take that image to deploy container.
spec.count	2	List the number of FEP Pgpool2 containers to create.
spec.serviceport	9999	Describes the TCP port for connecting to the FEP Pgpool2 container.
spec.statusport	9898	Identifies the TCP port for connecting to the PCP process.
spec.limits.cpu	400m	List the number of CPUs (restriction) to allocate to resources.limits.cpu.
spec.limits.memory	512Mi	Specifies the memory size (restriction) to allocate to resources.limits.memory.
spec.requests.cpu	200m	List the number of CPUs (request) to allocate to resources.requests.cpu.
spec.requests.memory	256Mi	Specifies the memory size (request) to allocate to resources.requests.memory
spec.fepclustername	new-fep	Enter the FEPCluster name to connect to.
spec.customhba		If you want to use pool _ hba.conf, describe what pool _ hba.conf should contain from the line below.
spec.customparams	listen_addresses = '*' pcp_listen_addresses = '*' num_init_children = 32 reserved_connections = 0 enable_pool_hba = off allow_clear_text_frontend_auth = off authentication_timeout = 80	"|" and the Pgpool-II parameters. Refer to "Pgpool-II parameters" for detail.

Field	Default	Details
	backend_weight0 = 1 backend_weight1 = 1 backend_flag0 = 'ALWAYS_PRIMARY' backend_flag1 = 'DISALLOW_TO_FAILOVER' connection_cache = on max_pool = 4 listen_backlog_multiplier = 2 serialize_accept = off child_life_time = 300 client_idle_limit = 0 child_max_connections = 0 connection_life_time = 0 reset_query_list = 'ABORT; DISCARD ALL' client_min_messages = info log_min_messages = debug1 log_statement = on log_per_node_statement = on log_client_messages = on log_hostname = on log_connections = on log_line_prefix = '%t: pid %p: ' load_balance_mode = on ignore_leading_white_space = on white_function_list = '' black_function_list = 'currval,lastval,nextval,setval' black_query_pattern_list = '' database_redirect_preference_list = '' app_name_redirect_preference_list = '' allow_sql_comments = off disable_load_balance_on_write = 'transaction' statement_level_load_balance = on sr_check_period = 0	

Field	Default	Details
	sr_check_user = 'postgres' delay_threshold = 0 log_standby_delay = 'none' ssl = on ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL' ssl_prefer_server_ciphers = off ssl_ecdh_curve = 'prime256v1' ssl_dh_params_file = '' relcache_expire = 0 relcache_size = 256 check_temp_table = catalog check_unlogged_table = on enable_shared_relcache = off relcache_query_target = primary wd_port0 = 9000 failover_on_backend_error = off	
spec.custompcp	"|"	If you use the pcp command, "|" and the contents of pcp.conf from the line below.
spec.customsslkey	"|"	If you want to do it, "|" and the Beethoven key content in the line below.
spec.customsslcert	"|"	If you want to do it, "|" and the contents of the public x 509 certificate from the line below.
spec.customsslcacert	"|"	If you want to do it, "|" and the following lines describe the contents of the CA root certificate in PEM format.
spec.customlogsize	100 Mi	Specifies the persistent volume size for log output.
spec.storageclassname		Specifies the storage class for log output. NFS storage is not available if you enable the following parameters: - enable_shared_relcache - memory_cache_enabled
spec.clientAuthMethod		Optional Specifies the method for client authentication. Define "scram" to use scram-sha-256 authentication. Otherwise, the client authentication method is still md5.
spec.scram.pgpoolkeySecret		Optional Describes the name of the secret that stores the key for use in encryption/decryption. Can only be specified at creation time.

Field	Default	Details
		If you change the contents of the specified secret, a restart is required.
spec.scram.userinfoSecret		Optional Write the name of the secret that contains the user name you added and the password that corresponds to that user. Can only be specified at creation time.

Pgpool-II parameters

The parameters that can be specified are shown in the table below. For details on the parameters, refer to the Pgpool-II manual.

Category	Parameter name (Specified format)	Restart required after change
Connection settings	listen_addresses (string)	Y
	pcp_listen_addresses (string)	Y
	num_init_children (integer)	Y
	reserved_connections (integer)	Y
Authentication settings	enable_pool_hba (boolean)	
	allow_clear_text_frontend_auth (boolean)	
	authentication_timeout (integer)	
Backend settings	backend_weight0 (floating point)	
	backend_weight1 (floating point)	
	backend_flag0	
	backend_flag1	
Connection pooling	connection_cache (boolean)	Y
	max_pool (integer)	Y
	listen_backlog_multiplier (integer)	Y
	serialize_accept (boolean)	Y
	child_life_time (integer)	Y
	client_idle_limit (integer)	
	child_max_connections (integer)	Y
	connection_life_time (integer)	Y
	reset_query_list (string)	
Error reporting and log acquisition	client_min_messages (enum)	
	log_min_messages (enum)	
	log_statement (boolean)	
	log_per_node_statement (boolean)	
	log_client_messages (boolean)	
	log_hostname (boolean)	
	log_connections (boolean)	
	log_error_verbosity (enum)	
	log_line_prefix (string)	

Category	Parameter name (Specified format)	Restart required after change
Load sharing settings	load_balance_mode (boolean)	Y
	ignore_leading_white_space (boolean)	
	white_function_list (string)	
	black_function_list (string)	
	black_query_pattern_list (string)	
	database_redirect_preference_list (string)	
	app_name_redirect_preference_list (string)	
	allow_sql_comments (boolean)	
	disable_load_balance_on_write (string)	Y
	statement_level_load_balance (boolean)	
Health check	connect_timeout (integer)	
Streaming replication check	sr_check_period (integer)	
	sr_check_user (string)	
	sr_check_password (string)	
	sr_check_database (string)	
	delay_threshold (integer)	
	log_standby_delay (string)	
Secure Sockect Layer (SSL)	ssl (boolean)	Y
	ssl_ciphers (string)	Y
	ssl_prefer_server_ciphers (boolean)	Y
	ssl_ecdh_curve (string)	Y
	ssl_dh_params_file (string)	Y
Other parameters	relcache_expire (integer)	Y
	relcache_size (integer)	Y
	enable_shared_relcache (boolean)	Y
	relcache_query_target (enum)	
	check_temp_table (enum)	
	check_unlogged_table (boolean)	

1.2.10 FEPAction Custom Resource Parameters

Specify parameters in the format described below.

Custom resource spec	Default	Change effect
.spec.targetClusterName		Must specify target FEP Cluster name within namespace mentioned in metadata.
.spec.targetPgpool2Name		Must specify target FEPPgpool2 name within namespace mentioned in metadata when using pgpool2_restart.
.spec.fepAction.type		Must specify action type. Supported action types are: restart

Custom resource spec	Default	Change effect
		pod_restart reload list switchover failover pgpool2_restart backup open_tde_masterkey create_extension update_admin_password backup_expire promote_standby fixed_stats
.spec.fepAction.args		Must specify arguments needed for given action. For details of args corresponding to each action refer to "1.2.10.1 FEPAction Specific Operation Details".
.spec.fepAction.backupType	full	Options If you specify backup for fepAction.type, the type of backup is used. full : Performs a full backup (backs up the contents of the database cluster). incr : Perform an incremental backup (Back up only the database cluster files that were changed during the last backup migration).
.spec.fepAction.backupRepo	1	Options Gets a backup in the specified repository. The range is 1 to 256.
.spec.sysExtraLogging		To turn extra debugging on, set value to true. It can be turned on/off at any time.

After execution of FEPAction CR, status is reflected in fepStatus field that is dynamically inserted in current FEPAction CR as needed.

fepStatus field used for FEPAction CR are described here

fepStatus (with possible values)	Remarks
fepActionStatus:	fepStatus is inserted at the top of FEPAction CR
fepActionCondition: Success | Failure	This flag is inserted in fepAction CR to reflect success or failure of requested action
fepActionResult: > "details"	The result contains verbose details corresponding to the specific action been executed. Should be noted that it is either plain text of HTTP output.
processedTimestamp: <time stamp>	Denotes time of action execution by the Operator

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
fepActionStatus:
 fepActionCondition: Success
metadata:
 name: new-fep-reload-action
 namespace: myns
```

```
spec:
 fepAction:
  args:
  - new-fep-sts-0
  - new-fep-sts-1
  type: reload
 sysExtraLogging: false
 targetClusterName: new-fep
```

Note

- Please do not use the FEPAction to perform a switchover or restart while executing backup. Failed to get the backup.
- You must create a new FEPAction custom resource for each operation.

1.2.10.1 FEPAction Specific Operation Details

Action type - reload

The reload action will manually reload the FEP database on the targeted FEPCluster.

"reload" action type expects users to specify the name of individual FEP pods that they want to run the database reload operation on. They specify that in the args section under the FEPAction CR spec as below :

```
spec:
fepAction:
 args:
 - nf-131851-sts-0
 - nf-131851-sts-1
 type: reload
 targetClusterName: nf-131851
```

Action type - restart

The restart action will manually restart the FEP database on the targeted FEPCluster.

"restart" action type expects users to specify the name of individual FEP pods that they want to run the database restart operation on. They specify that in the args section under the FEPAction CR spec as below:

```
spec:
fepAction:
 args:
 - nf-131851-sts-0
 - nf-131851-sts-1
 type: restart
 targetClusterName: nf-131851
```

Action type - pod_restart

The pod_restart action will restart specified list of POD for given target cluster. User can specify key word ‘ALL’ under ‘args’ section to restart all pods in target cluster. Alternatively, user can give the list of pods to be started in target cluster. User should either give ALL or the list of the pods.

This action restarts the replica pods first. Once all replicas have been restarted, it switches over the mastership to one of the replica before restarting old master pod. If it is a single node cluster, master will be restarted in its current state. This action is automatically created to restart pods when image or machine specs are changed for fep or backup container depending on autoPodRestart flag in FEPCluster CR (Refer to "1.2.1 FEPCluster Custom Resource Parameters" for details.):

```
spec:
fepAction:
args:
 - nf-131851-sts-0
```

```
 - nf-131851-sts-1
 type: pod_restart
 targetClusterName: nf-131851
```

Action type - list

The list action will return the status of the targeted FEPCluster.

"list" action type expects users to specify just the target cluster name to list the details of the same. Looks like below:

```
spec:
fepAction:
 type: list
 targetClusterName: nf-131851
```

Action type - switchover

The switchover action performs a manually switchover of the current leader/primary database from one pod to another pod of the targeted FEPCluster.

"switchover" action type expects users to specify the name of the target cluster that they want to perform switchover. args section is not required for switchover as FEPAction operator code will internally find it and promote new master. FEPAction CR spec as below:

```
spec:
fepAction:
 type: switchover
 targetClusterName: nf-131851
```

Action type - failover

The failover action performs a manually failover of the current primary database from one pod to another pod of the targeted FEPCluster. The difference between switchover and failover is that, switchover expects the primary database is running at the time whereas failover can force switchover of primary role from a non-responding pod to another pod. Note that failover is a disruptive action and may cause data lost.

"failover" action type expects users to specify the names of the candidate pods that they want to failover to. They specify that in the args section under the FEPAction CR spec as below:

```
spec:
fepAction:
 args:
 - nf-131851-sts-1
 - nf-131851-sts-2
 type: failover
 targetClusterName: nf-131851
```

Here, nf-131851-sts-1 and nf-131851-2 are the candidate pods to failover to. In this example, the current primary pod would be nf-131851-sts-0.

Action type - pgpool2_restart

"pgpool2_restart" action type expects users to specify the name of individual FEPPgpool2 resource that they want to restart operation on. They specify that in the targetPgpool2Name section under the FEPAction CR spec as below:

```
spec:
fepAction:
 type: pgpool2_restart
 targetPgpool2Name: nf-131851-pgpool2
```

Action type - backup

The "backup" action performs a backup on the target FEPCluster.

The "backup" action type requires you to specify the type of backup and the repository in which to store the data.

In the fepAction section of the FEPAction custom resource specification, specify the following:.

```
spec:
  targetClusterName: new-fep
  fepAction:
    type: backup
    backupType: full
    backupRepo: 1
```

Note

- Regardless of how the backup was performed (scheduled or FEPAction), if backups were performed at the same time by the same FEPCluster, subsequent backups will fail.
- If the backup repository Retention Option is specified in the FEPCluster custom resource spec.fepChildCrVal.backup.pgbackrestParams, the backup files obtained by the FEPAction are also deleted as specified by the option.

Action type - open_tde_masterkey

The open_tde_masterkey action opens a keystore for a TDE-enabled target cluster.

The "open_tde_masterkey" action type requires the user to specify the name of the target cluster on which the keystore will be opened. The args section is not required.

Specify the following:

```
spec:
  targetClusterName: nf-131851
  fepAction:
    type: open_tde_masterkey
```

Action type - create_extension

The create_extension action executes "CREATE EXTENSION" on the target FEPCluster and installs the extension.

In fepAction.args, specify the "extension name, version", "database", "schema", and "apply CASCADE options" to be installed.

Parameters specified by args	Description
extension	Required Specify the extension and version to be installed.
version	Optional Specifies the version of the extension to be specified for the VARSION option. If omitted, the VARSION option is omitted.
database	Option Specifies the database to install. If omitted, install in the "postgres" database.
schema	Option Specifies the schema to be installed, which is specified in the SCHEMA option. If omitted, the SCHEMA option is omitted.
cascade	Option true or false Enables or disables the CASCADE option for CREATE EXTENSION. If omitted, false.

An example specification is shown below.

```
spec:
  targetClusterName: new-fep
```

```
  fepAction:
    type: create_extension
    args:
      extension: "vci"
      version: "2.0"
      database: "mydb"
      schema: "public"
      cascade: "true"
```

Action type - update_admin_password

The update_admin_password action redefines the password for SUPERUSER "postgres" on the target FEPCluster with a random value.

This action will be executed when the FEPCluster custom resource spec.fepChildCrVal.sysUsers.pgAdminPassword is not defined.

An example specification is shown below.

```
spec:
fepAction:
type: update_admin_password
targetClusterName: new-fep
```

Action type - backup_expire

You can run the "pgbackrest expire" command on the FEPPod to remove expired backups.

The "pgbackrest expire" command is normally run automatically upon a successful backup, but it can be run by the user, for example, when the definition of the number of generations to retain for a backup is reduced, and the backup data can be deleted so that the number of retained generations conforms to the changed definition.

If you want to reduce the number of backup retention generations and free up disk space, apply "backup_expire" in the FEPAction after changing the retention setting for backup data under the FEPCluster custom resource fepChildCrVal.backup.pgbackrestParams.

You can specify the repository from which to remove the backup by specifying args.repo.

Parameters specified by args	Description
repo	Options Specified value: integer Specifies the number of the repository from which to remove the backup. If omitted, delete the backup for all backup repositories.

Note

The number of the backup repository must be N for repoN-type, as defined in the FEPCluster custom resource spec.fepChildCrVal.backup.pgbackrestParams.

The following is an example of changing the retention setting for backup data in a FEPCluster custom resource.

You want to reduce the number of backup generations stored in S3.

```
spec:
  fepChildCrVal:
    backup:
      pgbackrestParams:
        repo2-type=s3
        repo2-retention-full=5   # Change it to the number of generations you want to keep
        repo2-retention-full-type=time
```

The following is an example of a FEPAction custom resource that reduces the number of backup generations:

Since the backup repository for s3 is specified as repo2-type in pgbackrestParams, specify 2 for spec.fepActoin.repo.

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: backup-expire-action
spec:
  targetCluster: new-fep
  fepAction:
    type: backup_expire
    args:
      repo: 2
```

Action type - promote_standby

promote_standby promotes the FEP database in the disaster recovery environment from Standby DB to Primary DB. You must specify the DB cluster to be promoted.

The following shows a specification example.

```
spec:
  fepAction:
    type: promote_standby
  targetClusterName: my-fep
```

Action type - fixed_stats

The fixed_stats action performs a statistics operation (pg_dbms_stats) on the target FEPCluster.

The fixed_stats action must specify the operations that can be performed by pg_dbms_stats. Specify fixedStatsType: < operation type > in spec.fepAction.args. Depending on the operation type, there are additional required and optional arguments. Refer to the table below.

Table 1.1 Operation Types and Arguments

Operation Type	Required Arguments	Optional Argument
backup	None All databases are backed up with statistics. Also, if an identifying comment is omitted, "FEPAction" is assigned as the comment.	targetDb: Target database fixedObject: Name of the object after the schema Example)Target specific columns fixedObject: myschema.mytable.myclomun comment: "XXX"
restore	targetDb: Target database backupId: Backup ID or timestamp: timestamp The data type is timestamptz. The statistics are pinned using the specified backup ID or the most recent backup before the specified time.	Only when timestamp is defined can a post-schema object be specified. fixedObject: Name of the object after the schema Example)Target specific columns fiexdObject: myschema.mytable.myclomun
purge	backupId: Backup ID targetDb: Target data base Erases backups before the specified backup ID for the database specified in targetDb.	You can specify a delete flag. force: true force: false If omitted, it will be executed as false.
lock	targetDb: Target data base Fix statistics for the specified database.	fixedObject: Name of the object after the schema Example)Target specific columns fixedObject: myschema.mytable.myclomun

Operation Type	Required Arguments	Optional Argument
unlock	targetDb: Target data base Unpin statistics performed on the specified database.	fixedObject: Name of the object after the schema Example)Target specific columns fixedObject: myschema.mytable.myclomun
cleanup	targetDb: Target data base Performs a statistics cleanup on the specified database.	None
import	targetDb: Target data base Fixes statistics from a binary file for the database specified in targetDb. The file is imported from the fep.fixedStats.endpoint.protocol information defined in the FEPCluster custom resource. protocol changes the required arguments. See "Table 1.2 Required Arguments Per Protocol".	fixedObject: Name of the object after the schema
export	targetDb: Target data base targetStats: Valid and regular statistics Which to export (effective or plain) Exports the currently valid or canonical statistics for the database specified in targetDb. The file export destination uses the fep.fixedStats.endpoint. protocol information defined in the FEPCluster custom resource. protocol changes the required arguments. See "Table 1.2 Required Arguments Per Protocol".	fixedObject: Name of the object after the schema

Table 1.2 Required Arguments Per Protocol

Protocol	Required Arguments	Description
s3, gcs	url	URL of object storage
blob	azureBlobName	BLOB name of the statistics binary file
	azureContainerName	Container name of the Azure storage account
local	file	Specify the name of the file to which statistics will be exported/imported.

The following shows a specification example.

```
apiVersion: fep.fujitsu.io/v1
kind: FEPAction
metadata:
  name: fep-action-firxdStats
spec:
  fepAction:
    targetClusterName: new-fep
    type: fixed_stats
    args:
      fixedStatsType: backup
      targetDb: mydb
      fixedObject: schema.table.attaname
      comment: "xxyyzz"
```

1.2.11 FEPExporter Custom Resource

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Mandatory as it is
kind	FEPExporter	Mandatory as it is
metadata.name	fep-monitor	Name of FEPExporter CR - must be unique in namespace
metadata.namespace	fep-ns	Namespace - OCP populates it as current
spec.prometheus		Optional Prometheus MTLS spec section
spec.prometheus.tls		
spec.prometheus.tls.certificateName		Optional This points to Kubernetes TLS secret that contains the certificate of Prometheus ServiceMonitor. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.prometheus.tls.caName		Optional This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt.
spec.fep.remoteLogging.enable		Set to true to forward logs from fluentbit to fluentd
spec.fep.remoteLogging.image		Optional Fluentbit image to be used. If not specified, Operator will use the latest version that is supported by the Operator.
spec.fep.remoteLogging.pullPolicy	IfNotPresent	Optional
spec.fepExporter.		Exporter spec section
spec.fepExporter.authSecret		Optional Base Authentication secret to provide username & encrypted password of user
spec.fepExporter.authSecret.secretName		Secret name
spec.fepExporter.authSecret.userKey		Key of username in specified secret
spec.fepExporter.authSecret.passwordKey		Key of password in specified secret
spec.fepExporter.customLabel		Custom label to be added to Prometheus ServiceMonitor
spec.fepExporter.tls		FEPExporter MTLS specs
spec.fepExporter.tls.certificateName		Optional This point to Kubernetes TLS secret that contains the certificate of FepExporter. Prometheus will use this for certificate authentication. The certificate itself is stored in the key tls.crt.
spec.fepExporter.tls.caName		Optional This points to Kubernetes configmap that contains additional CA the client use to verify a server certificate. The CA is stored in the key ca.crt.

Field	Default	Details
spec.fepExporter.disableDefaultQueries	false	Optional Not defined or set to false => Create default queries Defined and set to true => Do not create default queries.
spec.fepExporter.disableDefaultAlertRules	false	Optional Not defined or set to false => Create default alert rules Defined and set to true => Do not create default alert rules. If Default queries are disabled => Do not create default alert rule.
spec.fepExporter.exporterLogLevel	error	Set logging level: one of debug, info, warn, error
spec.fepExporter.fepClusterList		Array of FEPCluster to monitor
spec.fepExporter.image.image		quay.io/fujitsu/fujitsu-enterprise-postgres-exporter:ubi9-17-1.0 Optional If not specified; image name is picked up from operator environment variable
spec.fepExporter.image.pullPolicy	IfNotPresent	Always or IfNotPresent
spec.fepExporter.mcSpec.limits	cpu: 500m memory: 700Mi	Max CPU allocated to exporter container Max memory allocated to exporter container
spec.fepExporter.mcSpec.requests	cpu: 200m memory: 512Mi	CPU allocation at start for exporter container memory allocation at start for exporter container
spec.fepExporter.scrapeInterval	30s	Optional This parameter may be specified to change statistics scraping frequency. If specified, Prometheus will poll FEPExporter at given interval. CHANGE THIS PARAMETER ONLY IF REALLY REQUIRED
spec.fepExporter.scrapeTimeout	30s	Optional This parameter may be specified to change statistics scraping timeout. If specified, Prometheus will wait for FEPExporter for maximum this given period to return statistics. CHANGE THIS PARAMETER ONLY IF REALLY REQUIRED
spec.fepExporter.sysExtraLogging	true	To turn on extra debugging messages for operator, set value to true *It can be turned on/off at any time*
spec.fepExporter.sysExtraEvent		Optional. To turn on event notification for custom resource changes, set the value to true. Can be turned on or off at any time.
spec.fepExporter.restartRequired	false	true: To restart FEPExporter, when there is any change found in CR or FEPCluster false: Will not restart FEPExporter
spec.fepExporter.userCustomQueries		Optional Section Example user's custom query to extract additional metrics.

```
    usr_example:
      query: "SELECT EXTRACT(EPOCH FROM (now() - pg_last_xact_replay_timestamp())) as lag"
      master: true
      metrics:
        - lag:
            usage: "GAUGE"
            description: "Replication lag behind master in seconds"
```

1.2.12 FEPAutoscale Custom Resource

When FEPClusterCR is defined, FEPAutoscaleCR is defined.

The parameters are as follows:

Configuration changes are made in FEPClusterCR.

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Fixed
kind	FEPAutoscale	Fixed
metadata.name	Same as FEPClusterCR	Fixed
metadata.namespace	Same as FEPClusterCR	Fixed
spec.scaleout.policy	off	[cpu_utilization/connection_number/off]
spec.scaleout.threshold	cpu_utilization: 40 connection_number: 40	Threshold
spec.scaleout.metricName	pg_capacity_connection_average	Specify this parameter if policy is connection_number. The custom metrics server must publish the average number of connections in the FEP cluster under this name.
spec.scaleout.stabilizationWindowSeconds	0	If the duration (seconds) threshold of this parameter has been exceeded continuously, a scale out is performed.
spec.limits.maxReplicas	2	Maximum number of replicas (0 to 15) If the value is out of range, no automatic scale out is performed.

1.2.13 FEPUpgrade Custom Resource

If "spec.fepChildCrVal.upgrade" is defined for the FEPCluster custom resource, the FEPUpgrade custom resource is defined.

The parameters are as follows:

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Fixed
kind	FEPUpgrade	Fixed
metadata.name	Same as FEPClusterCR	Fixed
metadata.namespace	Same as FEPClusterCR	Fixed
spec.upgrade		
spec.upgrade.sourceCluster		Specifies the FEPClusterCR name from which to migrate data. Required.
spec.upgrade.mcSpec.limits	cpu: 200m	Optional

Field	Default	Details
	memory: 300Mi	Specifies the maximum number of resources to allocate to the upgrade execution container.
spec.upgrade.mcSpec.requests	cpu: 100m memory: 200Mi	Optional Specifies the lower limit of resources allocated to the upgrade execution container.
spec.upgrade.image		Optional If omitted, the URL for image is obtained from the operator container environment.
spec.upgrade.imagePullPolicy	IfNotPresent	Optional Specifies the pull policy for the container image. - Always - IfNotPresent - Never
spec.upgrade.source.pgAdminTls.certificateName		Optional If you do not define spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName for the data source, it points to the Kubernetes TLS secret that contains the certificate for the Postgres user "postgres" in the data source. If the data source FEP has set the authentication method for the upgrade execution container to "cert", then the upgrade execution container uses the certificate defined as secret.
spec.upgrade.destination.pgAdminTls.certificateName		Optional If you have not defined the spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName of the newly created FEPCluster, it points to the Kubernetes TLS secret that contains the certificate of the Postgres user "postgres" in the data source. If you create a new FEP with the "cert" authentication method for the upgrade execution container, the upgrade execution container uses the certificate defined as secret.
spec.upgrade.storage		Optional Defines the storage for storing dump files.
spec.upgrade.storage.storageClass		Optional If omitted, the default storage class for your environment is used.
spec.upgrade.storage.size	2Gi	Optional Specifies the size of the storage to store the dump file.
spec.upgrade.storage.accessModes	ReadWriteOnce	Optional accessModes for store the dump file Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]

1.2.14 FEPLogging Custom Resources

The fepLogging section needs to be added under spec to define required parameters for FEPLogging configuration.

Following is a sample template :

```
spec:
  fepLogging:
    elastic:
      authSecret:
        secretName: elastic-auth
        passwordKey: password
        userKey: username
      host: elastic-passthrough.apps.openshift.com
      logstashPrefix: postgres
      port: 443
      scheme: https
      sslVerify: true
      tls:
        certificateName: elastic-cert
        caName: elastic-cacert
    image:
      pullPolicy: IfNotPresent
    mcSpec:
      limits:
        cpu: 500m
        memory: 700Mi
      requests:
        cpu: 200m
        memory: 512Mi
    restartRequired: false
    sysExtraLogging: false
    scrapeInterval: 30s
    scrapeTimeout: 30s
    tls:
      certificateName: fluentd-cert
      caName: cacert
  prometheus:
    ...
```

Below is the list of all parameters defined in the fepLogging section, along with their brief description

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
spec.fepLogging.image.image	Optional	Fluentd Image of FEPLogging	Yes
spec.fepLogging.image.pullPolicy	Required	Fluentd Image pull policy of FEPLogging	Yes
spec.fepLogging.mcSpec.limits.cpu	Required	Max CPU allocated to fluentd container	Yes
spec.fepLogging.mcSpec.limits.memory	Required	Max memory allocated to fluentd container	Yes
spec.fepLogging.mcSpec.requests.cpu	Required	CPU allocation at start for fluentd container	Yes
spec.fepLogging.mcSpec.requests.memory	Required	Memory allocation at start for fluentd container	Yes
spec.fepLogging.sysExtraLogging	Required	To turn on extra debugging messages for operator, set value to true. It can be turned on/off at any time	Yes

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
spec.fepLogging.sysExtraEvent	Optional	To turn on event notification for changes to custom resources, set the value to true. You can turn it on or off at any time.	Yes
spec.fepLogging.restartRequired	Required	To restart FEPLogging instance for applying any new configuration for example after certificate rotation	Yes
spec.fepLogging.scrapeInterval	Optional	Scrape interval for Prometheus to fetch metrics from FEPLogging instance	Yes
spec.fepLogging.scrapeTimeout	Optional	Scrape Timeout for Prometheus to fetch metrics from FEPLogging instance	Yes
spec.fepLogging.elastic.host	Optional	Target Elasticsearch host name	Yes
spec.fepLogging.elastic.port	Optional	Target Elasticsearch port number	Yes
spec.fepLogging.elastic.authSecret.secretName	Optional	Secret name which contains Elasticsearch authentication username & password	Yes
spec.fepLogging.elastic.authSecret.userKey	Optional	Username key specified in Elasticsearch authentication secret	Yes
spec.fepLogging.elastic.authSecret.passwordKey	Optional	Password key specified in Elasticsearch authentication secret	Yes
spec.fepLogging.elastic.logstashPrefix	Optional	Logstash prefix to differentiate index pattern in elastic search. Default value is postgres	Yes
spec.fepLogging.elastic.auditLogstashPrefix	Optional	Logstash prefix to differentiate index pattern in elastic search for auditlog. Default value is postgres	Yes
spec.fepLogging.elastic.scheme	Optional	Connection scheme between FEPLogging & Elasticsearch. Possible options http & https	Yes
spec.fepLogging.elastic.sslVerify	Optional	Set to true if you want to verify ssl certificate. If set to false then will not consider TLS crtificate	Yes
spec.fepLogging.elastic.tls.certificateName	Optional	Kubernetes secret name which holds fluentd certificate	Yes
spec.fepLogging.elastic.tls.caName	Optional	Kubernetes configmap which holds cacert of Elasticsearch to verify Elasticsearch TLS connection	Yes
spec.fepLogging.tls.certificateName	Optional	Kubernetes secret name which holds Fluentd certificate	Yes
spec.fepLogging.tls.caName	Optional	Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging & Prometheus	Yes
spec.prometheus.tls.certificateName	Optional	Kubernetes secret name which holds Prometheus certificate	Yes
spec.prometheus.tls.caName	Optional	Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging & Prometheus	Yes

1.2.15 FEP Custom Resources - spec.fep.pgBadger

Custom Resource spec	Change Effect
pgBadger.schedules.create	The 'create' schedule to create report and upload it to endpoint
pgBadger.schedules.cleanup	The 'cleanup' schedule to delete the report left in container
pgBadger.options.incremental	Default: false; When set to true: create incremental report in pgbadger
pgBadger.endpoint.authentication	a secret to contain authentication info to access endpoint support basic auth only
pgBadger.endpoint.customCertificateName	Client certitificate reference in customCertificate CR
pgBadger.endpoint.fileUploadParameter	The file upload parameter defined by the web server Default: 'file'
pgBadger.endpoint.insecure	equivalent to curl -insecure option, default to false
pgBadger.endpoint.url	Web server url to upload the report file

1.2.16 FEP Custom Resources - spec.fep.pgAuditLog

1.2.16.1 Details of pgAuditLog.endpoint.authentication

Protocol	Required key	Description
'http' or not defined	basic_auth	The basic authentication for http web server
's3'	aws_access_key	AWS access key
	aws_secret_key	AWS secret key
'blob'	azure_storage_account_name	Azure storage account name
	azure_storage_account_key	Azure storage account key

The Operator creates a default secret with keys for all the protocols with empty values when "pgAuditLog.endpoint.authentication" is not defined or empty.

The default secret is a template which the end user can update its proper values. The following is its content:

Default Authentication Secret

```
kind: Secret
apiVersion: v1
metadata:
  name: [FEPCluster name]-pgauditlog-auth
  namespace: [FEPCluster namespace]
type: Opaque
data:

    basic_auth:  ""
   aws_access_key:  ""
   aws_access_secret:  ""

   azure_storage_account_name:  ""
   azure_storage_account_key:  ""
```

When the default secret is created, the Operator also updates the created secret name in the FEPCluster CR:

FEPCluster

```
spec.fep
pgAuditLog:
 enable: 'true'
 endpoint:
   protocol: 's3'
   authentication:  '[FEPCluster name]-pgauditlog-auth'
...
```

The Operator uses the default secret but the upload feature will fail as the secret does not contain correct values. So the end user needs to update the values of the default secret to use upload feature properly.

Note

- The Operator does not own - user specified secret because it is created by the end user. Only the default secret created by operator is owned by the cluster.
- When the FEPCluster has been delete, this secret will remain.

1.2.16.2 CR example for customized pgaudit ConfigMap

- Enable pgAudit
  - The pgAudit extension will be enabled.
- Use custom pgAudit config file
  - The pgAudit log will be output based on custom configuration

FEPCluster

```
spec.fep
pgAuditLog:
 enable: 'true'
 config: my-pgaudit-conf
 endpoint: ... ...
# fepChildCrVal.customPgAudit will be ignored in this case
```

ConfigMap - Name: my-pgaudit-conf

```
data:
 pgaudit.conf: |
  [output]
  logger = 'auditlog'
  log_directory = '/database/log/audit'
  [rule]
  audit_role='jason'
  database='demo'
  class='READ, WRITE'
  [option]
```

1.2.16.3 CR example when uploading logs to Azure Blob

Use Azure blob as an endpoint to upload pgAudit file

FEPCluster (using Azue blob as endpoint)

```
spec.fep
pgAuditLog:
 enable: 'true'
```

FEPCluster (using Azue blob as endpoint)

```
 endpoint:
  protocol: 'blob'
  authentication: my-azure-blob-secret
  azureContainerName: cluster1
  azureBlobName: pgaudit-log-1
 schedules:
  upload: '30 * * * *'
```

Secret - Name: my-azure-blob-secret

```
data:
   azure_storage_account_name: cG9zdGdyZXM=
   azure_storage_account_key: ZnNcG9zdGads3cGzdGdyZXMyZXMlcA==
```

1.2.16.4 CR example for uploading logs to S3

Use AWS S3 as an endpoint to upload pgAudit file

- The pgAudit log will be uploaded to AWS s3 storage based on the provided schedule.

FEPCluster (using S3 as endpoint)

```
spec.fep
pgAuditLog:
 enable: 'true'
 endpoint:
  url: 's3://pgaudit1/cluster1'
  protocol: 's3'
  authentication: my-aws-s3-secret
 schedules:
  upload: '30 * * * *'
```

Secret - Name: my-aws-s3-secret

```
data:
   aws_access_key: cG9zdGdyZXM=
   aws_access_secret: ZnNlcaAZnNlcA3A==
```

Appendix A Default Metrics Queries

```
pg_capacity_connection:
    query: |
      select sys, idle, idleintx, idleintx10min, idleintx1hour, idleintx1day, idleintx1week,
(curr.idle + curr.idleintx + curr.active) total,  s.setting "max" from
      (
       select
       count(CASE WHEN a.state is null THEN 1 END) sys,
       count(CASE WHEN a.state='idle' THEN 1 END) idle,
       count(CASE WHEN a.state='idle in transaction' OR a.state='idle in transaction (aborted)' THEN
1 END) idleintx,
      count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND
age(now(), state_change) > interval '10 min'  THEN 1 END) idleintx10min,
      count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND
age(now(),state_change) > interval '1 hour'  THEN 1 END) idleintx1hour,
      count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND
age(now(),state_change) > interval '1 day'  THEN 1 END) idleintx1day,
      count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND
age(now(),state_change) > interval '1 week'  THEN 1 END) idleintx1week,
       count(CASE WHEN a.state='active' THEN 1 END) active
       from  pg_stat_activity a
      ) curr, pg_settings s where name = 'max_connections'
    master: true
    metrics:
      - sys:
          usage: 'GAUGE'
          description: 'Number of system connections.'
      - idle:
          usage: 'GAUGE'
          description: 'Number of idle connections.'
      - idleintx:
          usage: 'GAUGE'
          description: 'Number of idle in transaction connections.'
      - idleintx10min:
          usage: 'GAUGE'
          description: 'Number of idle in transaction connections running longer than 10 min.'
      - idleintx1hour:
          usage: 'GAUGE'
          description: 'Number of idle in transaction connections running longer than 1 hour.'
      - idleintx1day:
          usage: 'GAUGE'
          description: 'Number of idle in transaction connections running longer than 1 day.'
      - idleintx1week:
          usage: 'GAUGE'
          description: 'Number of idle in transaction connections running longer than 1 week.'
      - total:
          usage: 'GAUGE'
          description: 'Number of total connections.'
      - max:
          usage: 'GAUGE'
          description: 'Max number of connections.'

  pg_capacity_schema:
    query: |
      SELECT current_database() AS database_name, table_schema,
COALESCE(SUM(pg_total_relation_size('"'||table_schema||'"."'||table_name||'"')), 0) AS size
      FROM information_schema.tables GROUP BY table_schema
    master: true
    metrics:
      - database_name:
          usage: 'LABEL'
```

```
          description: 'Database name.'
      - table_schema:
          usage: 'LABEL'
          description: 'Table schema name.'
      - size:
          usage: 'GAUGE'
          description: 'Disk space of schema.'

  pg_capacity_tblspace:
    query: |
      SELECT pg_tablespace.spcname AS tablespace_name, pg_tablespace_size(pg_tablespace.spcname) AS
tablespace_size FROM pg_tablespace
    master: true
    metrics:
      - tablespace_name:
          usage: 'LABEL'
          description: 'Table space name.'
      - tablespace_size:
          usage: 'GAUGE'
          description: 'Disk space of table space.'

  pg_capacity_tblvacuum:
    query: |
      SELECT current_database() datname, t.table_schema,  count(t.table_name) table_count
      FROM information_schema.tables t
      INNER JOIN pg_catalog.pg_stat_user_tables tu on t.table_schema::text=tu.schemaname::text and
t.table_name::text=tu.relname::text
      and
      age(now(),greatest(COALESCE(last_vacuum, '1970-01-01Z'), COALESCE(last_autovacuum,
'1970-01-01Z'))) > interval '1 day'
      GROUP BY t.table_schema
    master: true
    metrics:
      - datname:
          usage: 'LABEL'
          description: 'Database name.'
      - table_schema:
          usage: 'LABEL'
          description: 'Table schema name.'
      - table_count:
          usage: 'GAUGE'
          description: 'Number of tables without vacuum for more than a day.'

  pg_capacity_longtx:
    query: |
      with xact_count as (
       SELECT COALESCE(datname, '') datname, count(1)
       FROM pg_stat_activity
       where backend_type='client backend' and age(now(), COALESCE(xact_start, '1970-01-01Z')) >
interval '5 minutes'
       group by datname
      )
      select d.datname, coalesce(xc.count, 0) as count from pg_database d left join xact_count xc on
d.datname=xc.datname
    master: true
    metrics:
      - datname:
          usage: 'LABEL'
          description: 'Database name.'
      - count:
          usage: 'GAUGE'
          description: 'Number of transactions running longer than 5 minutes.'
```

```
  pg_capacity_tblbloat:
    query: |
      SELECT DISTINCT
        current_database() as datname, schemaname, tablename as relname, /*reltuples::bigint,
relpages::bigint, otta,*/
        CASE WHEN relpages < otta THEN 0 ELSE bs*(sml.relpages-otta)::BIGINT END AS wastedbytes
      FROM (
        SELECT
          schemaname, tablename, cc.reltuples, cc.relpages, bs,
          CEIL((cc.reltuples*((datahdr+ma-
           (CASE WHEN datahdr%ma=0 THEN ma ELSE datahdr%ma END))+nullhdr2+4))/(bs-20::float)) AS otta,
         COALESCE(c2.relname,'?') AS iname, COALESCE(c2.reltuples,0) AS ituples, COALESCE(c2.relpages,
0) AS ipages,
          COALESCE(CEIL((c2.reltuples*(datahdr-12))/(bs-20::float)),0) AS iotta -- very rough
approximation, assumes all cols

        FROM (
          SELECT
            ma,bs,schemaname,tablename,
            (datawidth+(hdr+ma-(case when hdr%ma=0 THEN ma ELSE hdr%ma END)))::numeric AS datahdr,
           (maxfracsum*(nullhdr+ma-(case when nullhdr%ma=0 THEN ma ELSE nullhdr%ma END))) AS nullhdr2
          FROM (
            SELECT
              schemaname, tablename, hdr, ma, bs,
              SUM((1-null_frac)*avg_width) AS datawidth,
              MAX(null_frac) AS maxfracsum,
              hdr+(
                SELECT 1+count(*)/8
                FROM pg_stats s2
                WHERE null_frac<>0 AND s2.schemaname = s.schemaname AND s2.tablename = s.tablename
              ) AS nullhdr
            FROM pg_stats s, (
              SELECT
                (SELECT current_setting('block_size')::numeric) AS bs,
                CASE WHEN substring(v,12,3) IN ('8.0','8.1','8.2') THEN 27 ELSE 23 END AS hdr,
                CASE WHEN v ~ 'mingw32' THEN 8 ELSE 4 END AS ma
              FROM (SELECT version() AS v) AS foo
            ) AS constants
            GROUP BY 1,2,3,4,5
          ) AS foo
        ) AS rs
        JOIN pg_class cc ON cc.relname = rs.tablename
       JOIN pg_namespace nn ON cc.relnamespace = nn.oid AND nn.nspname = rs.schemaname AND nn.nspname
<> 'information_schema'
        LEFT JOIN pg_index i ON indrelid = cc.oid
        LEFT JOIN pg_class c2 ON c2.oid = i.indexrelid
      ) AS sml
      ORDER BY wastedbytes DESC
    master: true
    metrics:
      - datname:
          usage: 'LABEL'
          description: 'Database name.'
      - schemaname:
          usage: 'LABEL'
          description: 'Schema name.'
      - relname:
          usage: 'LABEL'
          description: 'Name of this table.'
      - wastedbytes:
          usage: 'GAUGE'
          description: 'Number of bytes wasted for table.'
```

```
pg_performance_locking_detail:
  query: |
    SELECT blocked_locks.pid AS blocked_pid,
    blocked_activity.usename AS blocked_user,
    blocking_locks.pid AS blocking_pid,
    blocking_activity.usename AS blocking_user,
    blocked_activity.query AS blocked_statement,
    1 locks
    FROM pg_catalog.pg_locks blocked_locks
    JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid = blocked_locks.pid
    JOIN pg_catalog.pg_locks blocking_locks
    ON blocking_locks.locktype = blocked_locks.locktype
    AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
    AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
    AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
    AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
    AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
    AND blocking_locks.transactionid IS NOT DISTINCT FROM blocked_locks.transactionid
    AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
    AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
    AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
    AND blocking_locks.pid != blocked_locks.pid
   JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid = blocking_locks.pid
    WHERE NOT blocked_locks.GRANTED
  master: true
  metrics:
    - blocked_pid:
        usage: 'LABEL'
        description: 'Blocked process id.'
    - blocked_user:
        usage: 'LABEL'
        description: 'Blocked user.'
    - blocking_pid:
        usage: 'LABEL'
        description: 'Blocking process id.'
    - blocking_user:
        usage: 'LABEL'
        description: 'Blocking user.'
    - blocked_statement:
        usage: 'LABEL'
        description: 'Blocked statement.'
    - locks:
        usage: 'GAUGE'
        description: 'Number of processes in blocked state.'

pg_performance_locking:
  query: |
    WITH
    locks as (
     SELECT blocked_locks.DATABASE, count(blocked_locks.pid) locks
     FROM pg_catalog.pg_locks blocked_locks
     JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid = blocked_locks.pid
     JOIN pg_catalog.pg_locks blocking_locks
     ON blocking_locks.locktype = blocked_locks.locktype
     AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
     AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
     AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
     AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
     AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
     AND blocking_locks.transactionid IS NOT DISTINCT FROM blocked_locks.transactionid
     AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
     AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
     AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
```

```
       AND blocking_locks.pid != blocked_locks.pid
       JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
       WHERE NOT blocked_locks.GRANTED group by blocked_locks.DATABASE
      ),
      dbs as (
       select * from pg_catalog.pg_database
      )
      select dbs.datname, coalesce(locks.locks, 0) locks from dbs left join locks on dbs.oid=DATABASE
    master: true
    metrics:
      - datname:
          usage: 'LABEL'
          description: 'Database name'
      - locks:
          usage: 'GAUGE'
          description: 'Number of processes in blocked state.'

  pg_replication:
    query: |
      SELECT CASE WHEN pg_last_wal_receive_lsn() = pg_last_wal_replay_lsn() THEN 0 ELSE GREATEST (0,
EXTRACT(EPOCH FROM (now() - pg_last_xact_replay_timestamp()))) END AS lag
    master: true
    metrics:
      - lag:
          usage: "GAUGE"
          description: "Replication lag behind master in seconds"

  pg_postmaster:
    query: |

      SELECT pg_postmaster_start_time as start_time_seconds from pg_postmaster_start_time()
    master: true
    metrics:
      - start_time_seconds:
          usage: "GAUGE"
          description: "Time at which postmaster started"

  pg_stat_user_tables:
    query: |
      SELECT
        current_database() datname,
        schemaname,
        relname,
        seq_scan,
        seq_tup_read,
        idx_scan,
        idx_tup_fetch,
        n_tup_ins,
        n_tup_upd,
        n_tup_del,
        n_tup_hot_upd,
        n_live_tup,
        n_dead_tup,
        n_mod_since_analyze,
        last_vacuum,
        last_autovacuum,
        last_analyze,
        last_autoanalyze,
        vacuum_count,
        autovacuum_count,
        analyze_count,
        autoanalyze_count
```

```
      FROM
        pg_stat_user_tables
    master: true
    metrics:
      - datname:
          usage: "LABEL"
          description: "Name of current database"
      - schemaname:
          usage: "LABEL"
          description: "Name of the schema that this table is in"
      - relname:
          usage: "LABEL"
          description: "Name of this table"
      - seq_scan:
          usage: "COUNTER"
          description: "Number of sequential scans initiated on this table"
      - seq_tup_read:
          usage: "COUNTER"
          description: "Number of live rows fetched by sequential scans"
      - idx_scan:
          usage: "COUNTER"
          description: "Number of index scans initiated on this table"
      - idx_tup_fetch:
          usage: "COUNTER"
          description: "Number of live rows fetched by index scans"
      - n_tup_ins:
          usage: "COUNTER"
          description: "Number of rows inserted"
      - n_tup_upd:
          usage: "COUNTER"
          description: "Number of rows updated"
      - n_tup_del:
          usage: "COUNTER"
          description: "Number of rows deleted"
      - n_tup_hot_upd:
          usage: "COUNTER"
          description: "Number of rows HOT updated (i.e., with no separate index update required)"
      - n_live_tup:
          usage: "GAUGE"
          description: "Estimated number of live rows"
      - n_dead_tup:
          usage: "GAUGE"
          description: "Estimated number of dead rows"
      - n_mod_since_analyze:
          usage: "GAUGE"
          description: "Estimated number of rows changed since last analyze"
      - last_vacuum:
          usage: "GAUGE"
         description: "Last time at which this table was manually vacuumed (not counting VACUUM FULL)"
      - last_autovacuum:
          usage: "GAUGE"
          description: "Last time at which this table was vacuumed by the autovacuum daemon"
      - last_analyze:
          usage: "GAUGE"
          description: "Last time at which this table was manually analyzed"
      - last_autoanalyze:
          usage: "GAUGE"
          description: "Last time at which this table was analyzed by the autovacuum daemon"
      - vacuum_count:
          usage: "COUNTER"
          description: "Number of times this table has been manually vacuumed (not counting VACUUM
FULL)"
      - autovacuum_count:
```

```
          usage: "COUNTER"
          description: "Number of times this table has been vacuumed by the autovacuum daemon"
      - analyze_count:
          usage: "COUNTER"
          description: "Number of times this table has been manually analyzed"
      - autoanalyze_count:
          usage: "COUNTER"
          description: "Number of times this table has been analyzed by the autovacuum daemon"

  pg_statio_user_tables:
    query: |
      SELECT current_database() datname, schemaname, relname, heap_blks_read, heap_blks_hit,
idx_blks_read, idx_blks_hit, toast_blks_read, toast_blks_hit, tidx_blks_read, tidx_blks_hit FROM
pg_statio_user_tables
    metrics:
      - datname:
          usage: "LABEL"
          description: "Name of current database"
      - schemaname:
          usage: "LABEL"
          description: "Name of the schema that this table is in"
      - relname:
          usage: "LABEL"
          description: "Name of this table"
      - heap_blks_read:
          usage: "COUNTER"
          description: "Number of disk blocks read from this table"
      - heap_blks_hit:
          usage: "COUNTER"
          description: "Number of buffer hits in this table"
      - idx_blks_read:
          usage: "COUNTER"
          description: "Number of disk blocks read from all indexes on this table"
      - idx_blks_hit:
          usage: "COUNTER"
          description: "Number of buffer hits in all indexes on this table"
      - toast_blks_read:
          usage: "COUNTER"
          description: "Number of disk blocks read from this table's TOAST table (if any)"
      - toast_blks_hit:
          usage: "COUNTER"
          description: "Number of buffer hits in this table's TOAST table (if any)"
      - tidx_blks_read:
          usage: "COUNTER"
          description: "Number of disk blocks read from this table's TOAST table indexes (if any)"
      - tidx_blks_hit:
          usage: "COUNTER"
          description: "Number of buffer hits in this table's TOAST table indexes (if any)"

  pg_database:
    query: |

      SELECT pg_database.datname, pg_database_size(pg_database.datname) as size_bytes FROM pg_database
    master: true
    cache_seconds: 30
    metrics:
      - datname:
          usage: "LABEL"
          description: "Name of the database"
      - size_bytes:
          usage: "GAUGE"
          description: "Disk space used by the database"
```

```
  pg_stat_statements:
    query: |
      SELECT t2.rolname, t3.datname, queryid, calls, total_plan_time / 1000 as
total_plan_time_seconds, total_exec_time / 1000 as total_exec_time_seconds, min_plan_time / 1000 as
min_plan_time_seconds, min_exec_time / 1000 as min_exec_time_seconds, max_plan_time / 1000 as
max_plan_time_seconds, max_exec_time / 1000 as max_exec_time_seconds, mean_plan_time / 1000 as
mean_plan_time_seconds, mean_exec_time / 1000 as mean_exec_time_seconds, stddev_plan_time / 1000 as
stddev_plan_time_seconds, stddev_exec_time / 1000 as stddev_exec_time_seconds, rows, shared_blks_hit,
shared_blks_read, shared_blks_dirtied, shared_blks_written, local_blks_hit, local_blks_read,
local_blks_dirtied, local_blks_written, temp_blks_read, temp_blks_written, blk_read_time / 1000 as
blk_read_time_seconds, blk_write_time / 1000 as blk_write_time_seconds FROM pg_stat_statements t1
JOIN pg_roles t2 ON (t1.userid=t2.oid) JOIN pg_database t3 ON (t1.dbid=t3.oid) WHERE t2.rolname !=
'rdsadmin'
    master: true
    metrics:
      - rolname:
          usage: "LABEL"
          description: "Name of user"
      - datname:
          usage: "LABEL"
          description: "Name of database"
      - queryid:
          usage: "LABEL"
          description: "Query ID"
      - calls:
          usage: "COUNTER"
          description: "Number of times executed"
      - total_plan_time_seconds:
          usage: "COUNTER"
          description: "Total plan time spent in the statement, in milliseconds"
      - total_exec_time_seconds:
          usage: "COUNTER"
          description: "Total exec time spent in the statement, in milliseconds"
      - min_plan_time_seconds:
          usage: "GAUGE"
          description: "Minimum plan time spent in the statement, in milliseconds"
      - min_exec_time_seconds:
          usage: "GAUGE"
          description: "Minimum exec time spent in the statement, in milliseconds"
      - max_plan_time_seconds:
          usage: "GAUGE"
          description: "Maximum plan time spent in the statement, in milliseconds"
      - max_exec_time_seconds:
          usage: "GAUGE"
          description: "Maximum exec time spent in the statement, in milliseconds"
      - mean_plan_time_seconds:
          usage: "GAUGE"
          description: "Mean plan time spent in the statement, in milliseconds"
      - mean_exec_time_seconds:
          usage: "GAUGE"
          description: "Mean exec time spent in the statement, in milliseconds"
      - stddev_plan_time_seconds:
          usage: "GAUGE"
          description: "Population standard deviation of plan time spent in the statement, in
milliseconds"
      - stddev_exec_time_seconds:
          usage: "GAUGE"
          description: "Population standard deviation of exec time spent in the statement, in
milliseconds"
      - rows:
          usage: "COUNTER"
          description: "Total number of rows retrieved or affected by the statement"
      - shared_blks_hit:
```

```
          usage: "COUNTER"
          description: "Total number of shared block cache hits by the statement"
      - shared_blks_read:
          usage: "COUNTER"
          description: "Total number of shared blocks read by the statement"
      - shared_blks_dirtied:
          usage: "COUNTER"
          description: "Total number of shared blocks dirtied by the statement"
      - shared_blks_written:
          usage: "COUNTER"
          description: "Total number of shared blocks written by the statement"
      - local_blks_hit:
          usage: "COUNTER"
          description: "Total number of local block cache hits by the statement"
      - local_blks_read:
          usage: "COUNTER"
          description: "Total number of local blocks read by the statement"
      - local_blks_dirtied:
          usage: "COUNTER"
          description: "Total number of local blocks dirtied by the statement"
      - local_blks_written:
          usage: "COUNTER"
          description: "Total number of local blocks written by the statement"
      - temp_blks_read:
          usage: "COUNTER"
          description: "Total number of temp blocks read by the statement"
      - temp_blks_written:
          usage: "COUNTER"
          description: "Total number of temp blocks written by the statement"
      - blk_read_time_seconds:
          usage: "COUNTER"
          description: "Total time the statement spent reading blocks, in milliseconds (if
track_io_timing is enabled, otherwise zero)"
      - blk_write_time_seconds:
          usage: "COUNTER"
          description: "Total time the statement spent writing blocks, in milliseconds (if
track_io_timing is enabled, otherwise zero)"

  pg_password_valid:
    query: |
      SELECT
        rolname,
        TRUNC (EXTRACT (EPOCH FROM (rolvaliduntil - now())) / (60*60*24)) AS days,
        EXTRACT (EPOCH FROM (rolvaliduntil - now())) AS seconds,
        cast(rolvaliduntil AS TEXT) AS date
      FROM
        pg_roles
      WHERE
        rolvaliduntil!='infinity' AND rolvaliduntil is not null
    master: true
    metrics:
      - rolname:
          usage: "LABEL"
          description: "Name of user"
      - date:
          usage: "LABEL"
          description: "Password Expiration Date"
      - days:
          usage: "GAUGE"
          description: "Number of days remaining before password expires."
      - seconds:
          usage: "GAUGE"
          description: "Number of seconds remaining before password expires."
```

```
  pg_not_set_password_valid:
    query: |
      SELECT
       COUNT(CASE WHEN a.rolvaliduntil is null AND a.rolcanlogin='t' THEN 1 END) null_count,
       COUNT(CASE WHEN a.rolvaliduntil='infinity' AND a.rolcanlogin='t' THEN 1 END) infinity_count,
       COUNT(CASE WHEN (a.rolvaliduntil is null OR a.rolvaliduntil='infinity') AND a.rolcanlogin='t'
THEN 1 END) all_count
      FROM pg_roles a
    master: true
    metrics:
      - null_count:
          usage: "GAUGE"
          description: "Number of days remaining before password valid is null."
      - infinity_count:
          usage: "GAUGE"
          description: "Number of days remaining before password valid is infinity."
      - all_count:
          usage: "GAUGE"
          description: "Number of days remaining before password valid is null or infinity."

  pg_tde_encrypted:
    query: |
      SELECT
         current_database() datname,
         ts.oid AS tablespace_oid,
         ts.spcname AS tablespace_name,
         tsx.spcencalgo AS encryption_algorithm,
         coalesce(t.count, 0) AS objs
      FROM
         pg_tablespace ts
         JOIN pgx_tablespaces tsx ON ts.oid = tsx.spctablespace
         LEFT OUTER JOIN (
            SELECT
               CASE WHEN c.reltablespace <> 0
                  THEN c.reltablespace
                  ELSE (select dattablespace from pg_database where datname = current_database())
               END AS reltablespaceid,
               count(*) AS count
            FROM pg_class c
               LEFT JOIN pg_namespace n ON n.oid = c.relnamespace
            WHERE c.relkind = ANY (ARRAY['r'::"char", 'm'::"char", 'p'::"char", 'i'::"char"])
               AND (n.nspname <> ALL (ARRAY['pg_toast'::name, 'pg_catalog'::name,
'information_schema'::name]))
               GROUP BY c.reltablespace
         ) t ON t.reltablespaceid = ts.oid
    metrics:
      - datname:
          usage: 'LABEL'
          description: "Database name."
      - tablespace_oid:
          usage: 'LABEL'
          description: "oid of the tablespace to check."
      - tablespace_name:
          usage: 'LABEL'
          description: "Name of the tablespace to check."
      - encryption_algorithm:
          usage: 'LABEL'
          description: "Algorithm used for encryption."
      - objs:
          usage: 'GAUGE'
          description: "Number of tables and indexes in the tablespace."
```

```
  pg_user_profile:
    query: |
      SELECT
        COUNT(*) AS total_roles,
        SUM(CASE WHEN userprfpasswordstatus = 'o' AND userprfaccountlock = 0 THEN 1 ELSE 0 END) AS
valid_roles,
        SUM(CASE WHEN userprfpasswordstatus = 'g' AND userprfaccountlock = 0 THEN 1 ELSE 0 END) AS
grace_time_roles,
        SUM(CASE WHEN userprfpasswordstatus = 'e' AND userprfaccountlock = 0 THEN 1 ELSE 0 END) AS
expired_roles,
       SUM(CASE WHEN userprfpasswordstatus = 'o' AND userprfaccountlock IN (1, 2) THEN 1 ELSE 0 END)
AS locked_roles
      FROM pgx_user_profile;
    metrics:
      - total_roles:
          usage: "GAUGE"
          description: "number of roles"
      - valid_roles:
          usage: "GAUGE"
          description: "number of valid roles"
      - grace_time_roles:
          usage: "GAUGE"
          description: "number of grace time roles"
      - expired_roles:
          usage: "GAUGE"
          description: "number of expired roles"
      - locked_roles:
          usage: "GAUGE"
          description: "number of locked roles"

  pg_txid:
    query: |
      SELECT max(age(datfrozenxid)) AS usage from pg_database where datallowconn = true
    metrics:
      - usage:
          usage: 'GAUGE'
          description: 'Cumulative number of transaction ID'
```

Appendix B Default Alert Rules

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: {{ ansible_operator_meta.name }}-{{ item.name }}-alertrules
  namespace: {{ ansible_operator_meta.namespace }}
  labels:
    app: prometheus-postgres-exporter-alertrules
    name: {{ ansible_operator_meta.name }}-{{ item.name }}-alertrules
spec:
  groups:
    - name: fep-container
      rules:
        - alert: ContainerDisappeared
          annotations:
            description: {{ 'Container {{$labels.container}}/{{$labels.pod}} from
{{$labels.namespace}} has been disappeared' }}
            summary: Container Pod disappeared.
          expr: time() -
            container_last_seen{ container="fep-patroni",
namespace="{{ ansible_operator_meta.namespace }}", pod=~"^{{ item.name }}-sts-.*" } >  60
          labels:
            severity: warning
        - alert: ContainerHighCPUUsage
          annotations:
            description: {{ 'Container {{$labels.container}}/{{$labels.pod}} from
{{$labels.namespace}} has been high on CPU usage(>80%) for 5 mins' }}
            summary: High Container CPU usage.
          expr:
(sum(node_namespace_pod_container:container_cpu_usage_seconds_total:sum_rate{pod=~"{{ item.name }}-
sts.*", namespace="{{ ansible_operator_meta.namespace }}", container="fep-patroni"}) by
(pod,namespace,container)/sum(kube_pod_container_resource_limits_cpu_cores) by
(pod,namespace,container))*100 > 80
          for: 5m
          labels:
            severity: warning
        - alert: ContainerHighRAMUsage
          annotations:
            description: {{ 'Container {{$labels.container}}/{{$labels.pod}} from
{{$labels.namespace}} has been high on RAM usage(>80%) since 30 mins' }}
            summary: High container memory usage.
          expr: sum(container_memory_working_set_bytes{pod=~"{{ item.name }}-sts.*",
namespace="{{ ansible_operator_meta.namespace }}", container="fep-patroni"}  /
container_spec_memory_limit_bytes * 100) by (pod, container, instance) > 80
          for: 30m
          labels:
            severity: warning
        - alert: PVCLowDiskSpace
          annotations:
            description: {{ 'Found low disk space on {{$labels.persistentvolumeclaim}} in
{{$labels.namespace}} namespace.' }}
            summary: {{ 'Found low disk space on {{$labels.persistentvolumeclaim}} in
{{$labels.namespace}} namespace.' }}
          expr:
kubelet_volume_stats_available_bytes{namespace="{{ ansible_operator_meta.namespace }}",
persistentvolumeclaim=~"fep.*{{ item.name }}.*"}/ (kubelet_volume_stats_capacity_bytes) * 100 < 10
          for: 5m
          labels:
            severity: warning
    - name: postgres
      rules:
```

```
        - alert: PostgresqlDown
          annotations:
            description:  "Postgresql one or more instances are down in FEPCluster {{ item.name }} in
{{ ansible_operator_meta.namespace }} namespace. Please check the FEP pods in this cluster"
            summary: "Postgresql FEPCluster {{ item.name }} in {{ ansible_operator_meta.namespace }}
namespace is degraded"
          expr: count(pg_static{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{item.name}}-sts.*" }) <
{{item.instances | length}}
          labels:
            severity: error
        - alert: PostgresqlTooManyConnections
          annotations:
            description: {{ 'PostgreSQL instance has too many connections on server
{{ $labels.server }} in {{ $labels.namespace }} namespace.' }}
            summary: {{ 'Postgresql too many connections (FEPCluster server {{ $labels.server }})' }}
          expr: pg_capacity_connection_total{namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*"}/
pg_settings_max_connections > 0.9
          labels:
            severity: warning
        - alert: PostgresqlRolePasswordCloseExpierd
          annotations:
            description:  "The Postgresql role's password expires in less than 7 days. Please update
the password."
            summary: "Postgresql Role Password expires in less than 7 days."
          expr: count(pg_password_valid_days{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*", rolname=~".*" }
< 8) > 0
          labels:
            severity: warning
        - alert: PostgresqlRolePasswordExpired
          annotations:
            description:  "The Postgresql role's password has already expired. Please update the
password."
            summary: "Postgresql Role Password has already expired. "
          expr: count(pg_password_valid_seconds{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*", rolname=~".*" }
< 0) > 0
          labels:
            severity: warning
        - alert: PasswordIsGraceTimeByUserProfile
          annotations:
          description: The password for the role in the grace time exists. Please change your password.
            summary: There is a password in the grace time
          expr: grace_time_roles{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*" } > 0
          labels:
            severity: warning
        - alert: PasswordExpiredByUserProfile
          annotations:
            description: Expired role password exists. Please change your password.
            summary: Expired role password exists
          expr: expired_roles{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*" } > 0
          labels:
            severity: warning
        - alert: PasswordLockedByUserProfile
          annotations:
            description: There is a role with a password lock. Please confirm the role.
            summary: Password locked role exists
          expr: locked_roles{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*" } > 0
```

```
          labels:
            severity: warning
        - alert: PostgresqlTooManyTxidUsage
          annotations:
           description: "Transaction ID usage has exceeded the value of autovacuum_freeze_max_age for
more than 24 hours. Consider periodic aggressive vacuuming."
            summary: "Transaction ID usage exceeds autovacuum_freeze_max_age"
          expr: pg_txid_usage{ namespace="{{ ansible_operator_meta.namespace }}",
service="{{ ansible_operator_meta.name }}-service", server=~"{{ item.name }}-sts.*" } >
pg_settings_autovacuum_freeze_max_age
  for: 24h
          labels:
            severity: warning
```

Appendix C Operator Operation Event Notification

C.1 FEPCluster Event Notification on Custom Resource Changes

When "spec.fep.sysExtraEvent" is true, event notification of operator actions occurs when you change the value of the following fields defined in the FEPCluster custom resource.

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.fep.image.image	Start Change	FEPCluster	Started patching fep-patroni spec.fep.image.image
	Change Successful	FEPCluster	Successfully patching fep-patroni spec.fep.image.image
	Change Failed	FEPCluster	Error/Failure in patching fep-patroni spec.fep.image.image
	FEPActrion Successfully Inherits Action to Custom Resource	FEPCluster	Successfully creaeting FEPActionCR for restart so check FEPAcrion result
	Fail to inherit processing to FEPAction custom resource	FEPCluster	Error/Failure in creating FEPActionCR for restart
	Start Reflection	FEPAction	Started restart Action for ALL Pods
	reflection success	FEPAction	Successfully Restart Action for ALL Pods
	Reflection failed	FEPAction	Error/Failure Restart Action for ALL Pods
spec.fep.mcSpec	Start Change	FEPCluster	Started patching fep-patroni spec.fep.mcSpec
	Change Successful	FEPCluster	Successfully patching fep-patroni spec.fep.mcSpec
	Change Failed	FEPCluster	Error/Failure in patching fep-patroni spec.fep.mcSpec
	FEPActrion Successfully Inherits Action to Custom Resource	FEPCluster	Successfully creaeting FEPActionCR for restart so check FEPAcrion result
	Fail to inherit processing to FEPAction custom resource	FEPCluster	Error/Failure in creating FEPActionCR for restart
	Start Reflection	FEPAction	Started restart Action for ALL Pods
	1.2.14reflection success	FEPAction	Successfully Restart Action for ALL Pods
	Reflection failed	FEPAction	Error/Failure Restart Action for ALL Pods
spec.fep.instances (Scale in)	Start Change	FEPCluster	Started scale in FEP Cluster
	Change Successful	FEPCluster	Successfully scale in FEP Cluster
	Change Failed	FEPCluster	Error/Failure in scale in FEP Cluster
spec.fep.instances (Scale out)	Start Change	FEPCluster	Started scale out FEP Cluster
	Change Successful	FEPCluster	Successfully scale out FEP Cluster

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
	Change Failed	FEPCluster	Error/Failure in scale out FEP Cluster
spec.fep.pgBadger	Start Change	FEPCluster	Started update FEPConfig CR
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values
	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fep.pgBadger
	reflection success	FEPConfig	Successfully patching spec.fep.pgAuditLog and spec.fep.pgBadger
	Reflection failed	FEPConfig	Error/Failure in patching spec.fep.pgAuditLog and spec.fep.pgBadger
spec.fep.pgBadger.schedule.create	reflection success	FEPConfig	Successfully updating spec.fep.pgBadger.schedules.create
	Reflection failed	FEPConfig	Error/Failure in updating spec.fep.pgBadger.schedules.create
spec.fep.pgBadger.schedule.cleanup	reflection success	FEPConfig	Successfully updating spec.fep.pgBadger.schedules.cleanup
	Reflection failed	FEPConfig	Error/Failure in updating spec.fep.pgBadger.schedules.cleanup
spec.fep.replicationSlots	Start Change	FEPCluster	Started update FEPConfig CR
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values
	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fepChildCrVal.replicationSlots
	reflection success	FEPConfig	Successfully patching spec.fepChildCrVal.replicationSlots
	Reflection failed	FEPConfig	Error/Failure in patching spec.fepChildCrVal.replicationSlots
spec.fep.pgAuditLog	Start Change	FEPCluster	Started update FEPConfig CR
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.fep.pgAuditLog	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fep.pgAuditLog
	reflection success	FEPConfig	Successfully patching spec.fep.pgAuditLog and spec.fep.pgBadger
	Reflection failed	FEPConfig	Error/Failure in patching spec.fep.pgAuditLog and spec.fep.pgBadger
spec.fep.pgAuditLog.auditLogPath	Start Reflection	FEPConfig	Started patching spec.fep.pgAuditLog
	reflection success	FEPConfig	Successfully patching spec.fep.pgAuditLog
	Reflection failed	FEPConfig	Error/Failure in patching spec.fep.pgAuditLog
spec.fepChildCrVal.customPgAudit	Start Change	FEPCluster	Started update FEPConfig CR
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values
	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fepChildCrVal.customPgAudit
	reflection success	FEPConfig	Successfully patching spec.fepChildCrVal.customPgAudit so restart DB
	Reflection failed	FEPConfig	Error/Failure in patching fepStatus to patch spec.fepChildCrVal.customPgAudit
spec.fepChildCrVal.customPgHba	Start Change	FEPCluster	Started update FEPConfig CR
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values
	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fepChildCrVal.customPgHba
	reflection success	FEPConfig	Successfully patching spec.fepChildCrVal.customPgHba
	Reflection failed	FEPConfig	Error/Failure in patching spec.fepChildCrVal.customPgHba
spec.fepChildCrVal.customPgParams	Start Change	FEPCluster	Started update FEPConfig CR

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
	Succeeded in inheriting processing to FEPConfig custom resource	FEPCluster	Successfully updateing FEPConfig CR with current values
	Fail to inherit processing to FEPConfig custom resource	FEPCluster	Error/Failure in updateing FEPConfig CR with current values
	Start Reflection	FEPConfig	Started patching spec.fepChildCrVal.customPgParams
	reflection success	FEPConfig	Successfully patching spec.fepChildCrVal.customPgParams
	Reflection failed	FEPConfig	Error/Failure in patching spec.fepChildCrVal.customPgParams
spec.fepChildCrVal.backup.image	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.image
	Change Successful	FEPCluster	Successfully patching fepbackup spec.fepChildCrVal.backup.image
	Change Failed	FEPCluster	Error/Failure in patching fepbackup spec.fepChildCrVal.backup.image
spec.fepChildCrVal.backup.mcSpec	Start Change	FEPCluster	Started patching fepbackup spec.fepChildCrVal.backup.mcSpec
	Change Successful	FEPCluster	Successfully patching fepbackup spec.fepChildCrVal.backup.mcSpec
	Change Failed	FEPCluster	Error/Failure in patching fepbackup spec.fepChildCrVal.backup.mcSpec
spec.fepChildCrVal.backup.schedule.num	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule.num
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup
spec.fepChildCrVal.backup.pgbackrestKeyParams	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.pgbackrestKeyParams
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup
spec.fepChildCrVal.backup.pgbackrestParams	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.pgbackrestParams
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup
spec.fepChildCrVal.backup.schedule1	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule1

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup.schedule1
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup.schedule1
spec.fepChildCrVal.backup.schedule2	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule2
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup.schedule2
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup.schedule2
spec.fepChildCrVal.backup.schedule3	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule3
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup.schedule3
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup.schedule3
spec.fepChildCrVal.backup.schedule4	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule4
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup.schedule4
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup.schedule4
spec.fepChildCrVal.backup.schedule5	Start Change	FEPCluster	Started patching spec.fepChildCrVal.backup.schedule5
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.backup.schedule5
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.backup.schedule5
spec.fepChildCrVal.autoscale	Start Change	FEPCluster	Started patching spec.fepChildCrVal.autoscale
	Change Successful	FEPCluster	Successfully patching spec.fepChildCrVal.autoscale
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.autoscale
spec.fepChildCrVal.storage	Start Change	FEPCluster	Started patching FEPVolume CR
	Change Successful	FEPCluster	Successfully patching FEPVolume CR with current values
	Change Failed	FEPCluster	Error/Failure in patching FEPVolume CR with current values
spec.fepChildCrVal.sysUsers	Start Change	FEPCluster	Started patching spec.fepChildCrVal.sysUsers passwords
	Change Successful	FEPCluster	uccessfully patching spec.fepChildCrVal.sysUsers in FEPCluster
	Change Failed	FEPCluster	Error/Failure in patching spec.fepChildCrVal.sysUsers in FEPCluster

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.fepChildCrVal.sysUsers.pgMetricsPassword	Start Change	FEPCluster	Started setting spec.fepChildCrVal.sysUsers.pgMetricsPassword to FEPUserCR where spec.fepChildCrVal.sysUsers.pgMetricsPassword is undefined
	Change Successful	FEPCluster	Successfully setting spec.fepChildCrVal.sysUsers.pgMetricsPassword
	Change Failed	FEPCluster	Error/Failure in Setting spec.fepChildCrVal.sysUsers.pgMetricsPassword
spec.fepChildCrVal.sysUsers.pgMetricsUserTls	Start Change	FEPCluster	Started setting spec.fepChildCrVal.sysUsers.pgMetricsUserTls to FEPUserCR where spec.fepChildCrVal.sysUsers.pgMetricsUserTls is undefined
	Change Successful	FEPCluster	Successfully setting spec.fepChildCrVal.sysUsers.pgMetricsUserTls
	Change Failed	FEPCluster	Error/Failure in setting spec.fepChildCrVal.sysUsers.pgMetricsUserTls
spec.fepChildCrVal.sysUsers.pgMetricsUser	Start Change	FEPCluster	Started delete spec.fepChildCrVal.sysUsers.pgMetricsUser
	Change Successful	FEPCluster	Successfully setting spec.fepChildCrVal.sysUsers.pgMetricsUser or spec.fepChildCrVal.sysUsers.pgMetricsUserTls
	Change Failed	FEPCluster	Error/Failure in setting spec.fepChildCrVal.sysUsers.pgMetricsUser or spec.fepChildCrVal.sysUsers.pgMetricsUserTls
spec.fepChildCrVal.sysUsers.pgMetricsUserTls	Start Change	FEPCluster	Started delete spec.fepChildCrVal.sysUsers.pgMetricsUserTls
spec.fepChildCrVal.sysUsers.pgMetricsUserTls	Start Change	FEPCluster	Successfully setting spec.fepChildCrVal.sysUsers.pgMetricsUser or spec.fepChildCrVal.sysUsers.pgMetricsUserTls
	Change Successful	FEPCluster	Error/Failure in setting spec.fepChildCrVal.sysUsers.pgMetricsUser or spec.fepChildCrVal.sysUsers.pgMetricsUserTls
spec.sysTde.tdek.targetKeyId	Change Failed	FEPCluster	Started patching spec.sysTde.tdek.targetKeyId
	Start Change	FEPCluster	Successfully patching spec.sysTde.tdek.targetKeyId
	Change Successful	FEPCluster	Error/Failure in patching spec.sysTde.tdek.targetKeyId
spec.sysTde.tdek.kmsDefinition.sslpassphrase	Change Failed	FEPCluster	Started patching spec.sysTde.tdek.kmsDefinition.sslpassphrase
	Start Change	FEPCluster	Successfully patching spec.sysTde.tdek.kmsDefinition.sslpassphrase
	Change Successful	FEPCluster	Error/Failure in patching spec.sysTde.tdek.kmsDefinition.sslpassphrase

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.remoteLogging.image	Change Failed	FEPCluster	Started patching fep-logging-fluent-bit spec.remoteLogging.image
	Start Change	FEPCluster	Successfully patching fep-logging-fluent-bit spec.remoteLogging.image
	Change Successful	FEPCluster	Error/Failure in patching patching fep-logging-fluent-bit spec.remoteLogging
spec.monitoring.fepExporter.authSecret (new)	Change Failed	FEPCluster	Started patching FEPExporter CR because spec.fepExporter.authSecret or spec.fepExporter.tls details are newly defined
	Start Change	FEPCluster	Successfully patching FEPExporter CR for spec.fepExporter.authSecret or spec.fepExporter.tls details
	Start Change	FEPCluster	Error/Failure in patching FEPExporter CR for spec.fepExporter.authSecret or spec.fepExporter.tls details
spec.monitoring.fepExporter.tls (new)	Change Successful	FEPCluster	Started patching FEPExporter CR because spec.fepExporter.authSecret or spec.fepExporter.tls details are newly defined
	Change Failed	FEPCluster	Successfully patching FEPExporter CR for spec.fepExporter.authSecret or spec.fepExporter.tls details
spec.monitoring.fepExporter.tls (new)	Change Failed	FEPCluster	Error/Failure in patching FEPExporter CR for spec.fepExporter.authSecret or spec.fepExporter.tls details
spec.monitoring.fepExporter.authSecre (removed)	Start Change	FEPCluster	Started patching FEPExporter CR because spec.fepExporter.authSecret details are deleted
	Change Successful	FEPCluster	Successfully patching spec.fepExporter.authSecret
	Change Failed	FEPCluster	Error/Failure in patching spec.fepExporter.authSecret
spec.monitoring.fepExporter.tls (removed)	Start Change	FEPCluster	Started patching FEPExporter CR because spec.fepExporter.tls details are deleted
	Change Successful	FEPCluster	Successfully patching spec.fepExporter.tls
	Change Failed	FEPCluster	Error/Failure in patching spec.fepExporter.tls
spec.monitoring.fepExporter	Start Change	FEPCluster	Started creating FEPExporter CR
	Change Successful	FEPCluster	Successfully creating FEPExporter CR
	Change Failed	FEPCluster	Error/Failure in Creating FEPExporter CR

C.2 FEPExporter Event Notification on Custom Resource Changes

When "spec.fepExporter.sysExtraEvent" is true, provides event notification of operator actions when the value of the following fields defined in the FEPExporter custom resource are changed.

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.fepExporter.restartRequired	Start Change	FEPExporter	Started patching spec.fepExporter.restartRequired
	Change Successful	FEPExporter	Successfully patching spec.fepExporter.restartRequired
	Change Failed	FEPExporter	Error/Failure in patching spec.fepExporter.restartRequired
spec.fepExporter.userCustomQueries	Start Change	FEPExporter	Started patching spec.fepExporter.userCustomQueries
	Change Successful	FEPExporter	Successfully patching spec.fepExporter.userCustomQueries
	Change Failed	FEPExporter	Error/Failure in patching spec.fepExporter.userCustomQueries

C.3 Event Notification When FEPLogging Custom Resource Changes

When "spec.fepLogging.sysExtraEvent" is true, provides event notification of operator actions when you change the value of the following fields defined in the FEPLogging custom resource.

Field Whose Value You Want to Change	Notification Timing	Notification Custom Resources	Notification Message
spec.fepLogging.restartRequired	Start Change	FEPLogging	Started patching spec.fepLogging.restartRequired
spec.fepLogging.restartRequired	Change Successful	FEPLogging	Successfully patching spec.fepLogging.restartRequired
spec.fepLogging.restartRequired	Change Failed	FEPLogging	Error/Failure in patching spec.fepLogging.restartRequired
spec.fepLogging.scrapeInterval spec.fepLogging.scrapeTimeout	Start Change	FEPLogging	Started patching spec.fepLogging.scrapeInterval and spec.fepLogging.scrapeTimeou
spec.fepLogging.scrapeInterval spec.fepLogging.scrapeTimeout	Change Successful	FEPLogging	Successfully patching spec.fepLogging.scrapeInterval and spec.fepLogging.scrapeTimeout
spec.fepLogging.scrapeInterval spec.fepLogging.scrapeTimeout	Change Failed	FEPLogging	Error/Failure in patching spec.fepLogging.scrapeInterval and spec.fepLogging.scrapeTimeout
spec.fepLogging.restartRequired	Start Change	FEPLogging	Started patching spec.fepLogging.restartRequired

Fujitsu Enterprise Postgres for Kubernetes

Quick Start Guide

J2UL-QK17-01ENZ0(00)

Copyright 2022-2024 Fujitsu Limited

1. Prerequisites

- Registered OpenShift cluster with Red Hat Marketplace

  cf. https://marketplace.redhat.com/en-us/documentation/clusters#register-openshift-cluster-with-red-hat-marketplace

- Buy or try the product 'Fujitsu Enterprise Postgres for Kubernetes ' from Red Hat Marketplace

  cf. https://marketplace.redhat.com/en-us/documentation/operators

2. System Requirments

2.1. CPU

The following CPU architectures are supported.

No	CPU architecture
1	x86
2	s390x
3	ppc64le

2.2. Supported Platform

The following platform is supported.

No	Platform	Version
1	OpenShift Container Platform	4.14, 4.15, 4.16

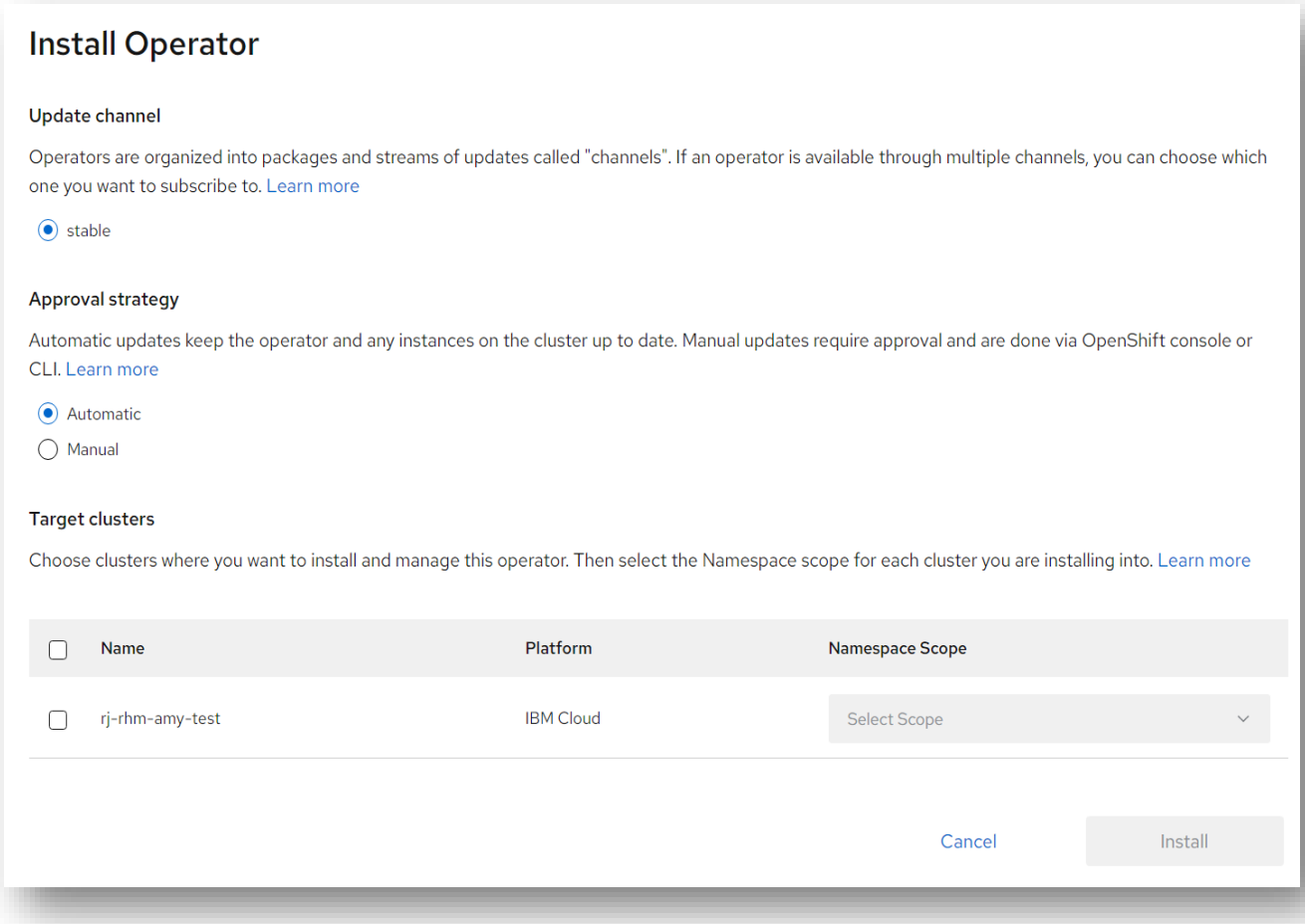
3. Operator installation from Red Hat Marketplace

1. For information on registering your cluster and creating a namespace, see Red Hat Marketplace Docs. This must be done prior to installing the operator.
2. On the main menu, click **Workspace**, click **Software,** click on the **product box** of 'Fujitsu Enterprise Postgres for Kubernetes ', and then click Install Operator.
3. On the *Update Channel section*, select an option.
4. On the *Approval Strategy* section, select either *Automatic or Manual*. The approval strategy corresponds to how you want to process operator upgrades.
5. On the *Target Cluster* section:

Copyright 2022-2024 Fujitsu Limited

- Click the checkbox next to the clusters where you want to install the Operator.
- For each cluster you selected, under **Namespace Scope**, on the **Select Scope** list, select an option.

6. Click **Install**. It may take several minutes for installation to complete.
7. Once installation is complete, the status will change from **installing** to **Up to date**.
8. For further information, see the [Red Hat Marketplace Operator documentation]

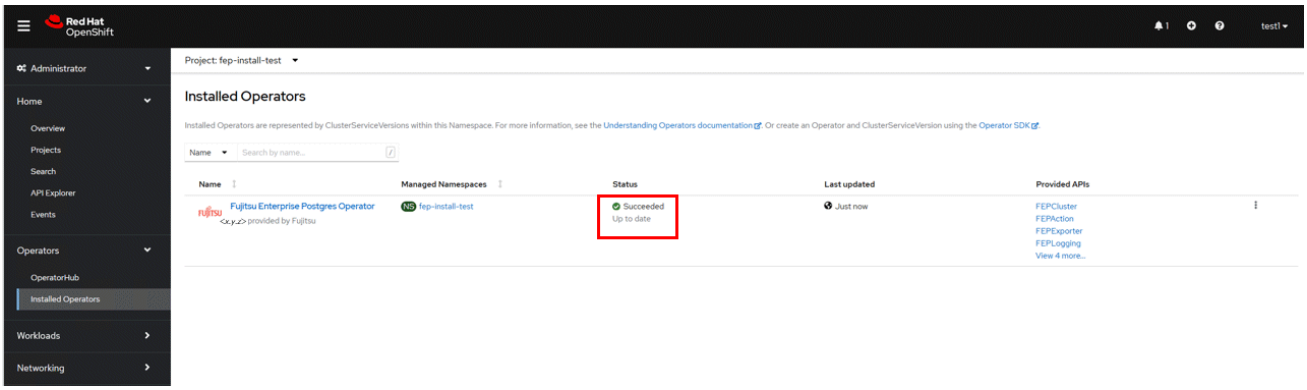


4. Verification of operator installation

1. Once status changes to Up to date, click the vertical ellipsis and select Cluster Console.
2. Open the cluster where you installed the product
3. Go to **Operators > Installed Operators**
4. Select the Namespace or Project you installed on
5. Verify status for product is **Succeeded**

Copyright 2022-2024 Fujitsu Limited

"<x.y.z>" in the screen example indicates the version level of the operator.



Installed operators status changes to “Succeeded”

5. Deploying FEPCluster using Operator

CR templates are published in the following repository on GitHub.

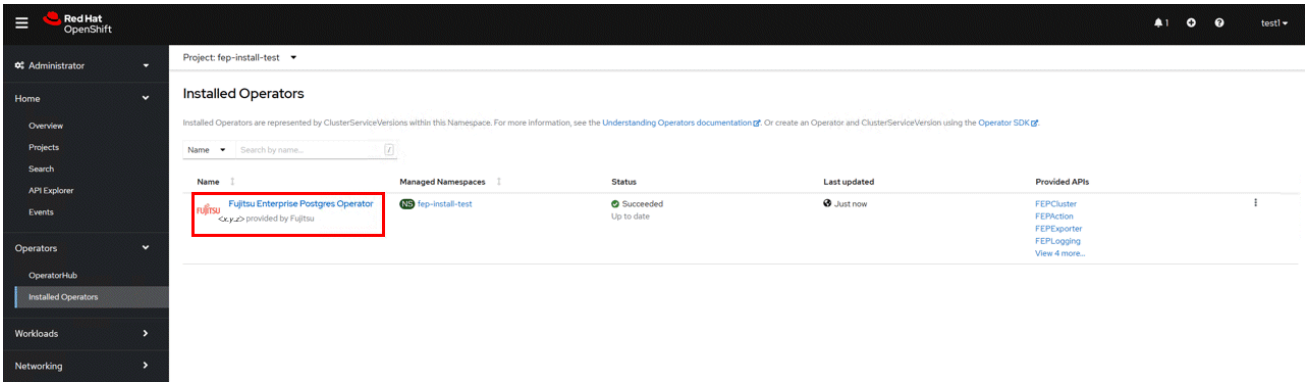
https://github.com/fujitsu/fep-operator-examples

This repository provides sample files that you can use to run Fujitsu Enterprise Postgres Operator.

To deploy a FEPCluster in a given namespace, follow these steps:

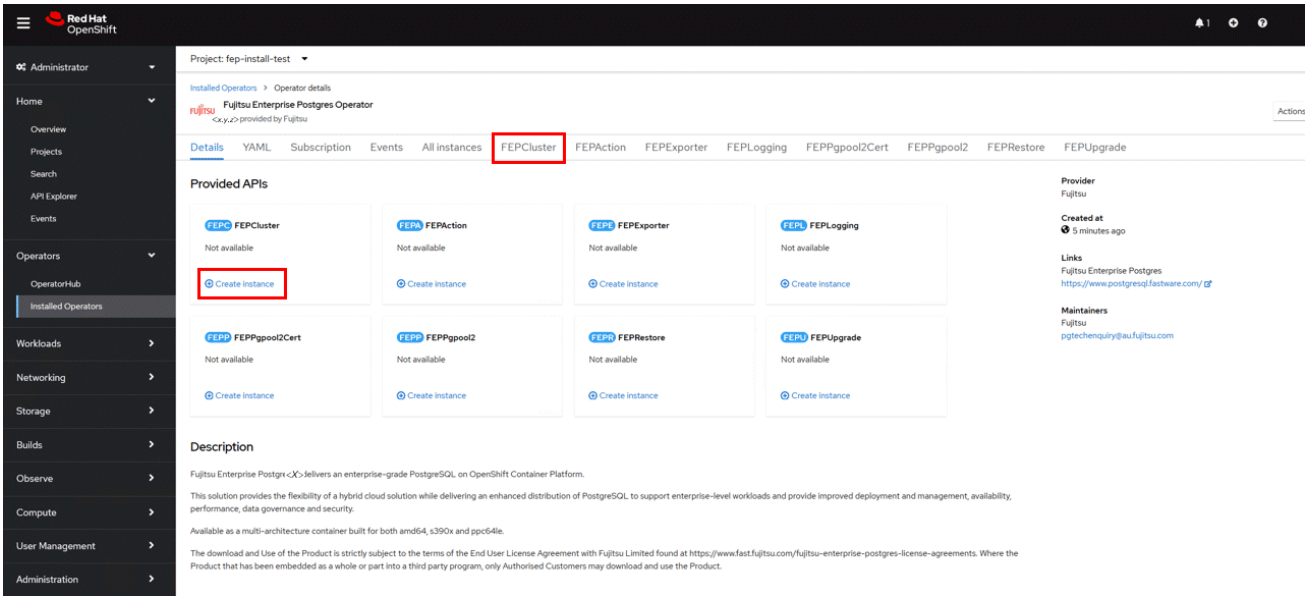
"<x.y.z>" in the screen example indicates the version level of the operator. "<X>" indicates the product version of Fujitsu Enterprise Postgres.

1. Under "Operators" menu item, click on "**Installed Operators**". You will see the installed FEP operator. Click on the name of operator.

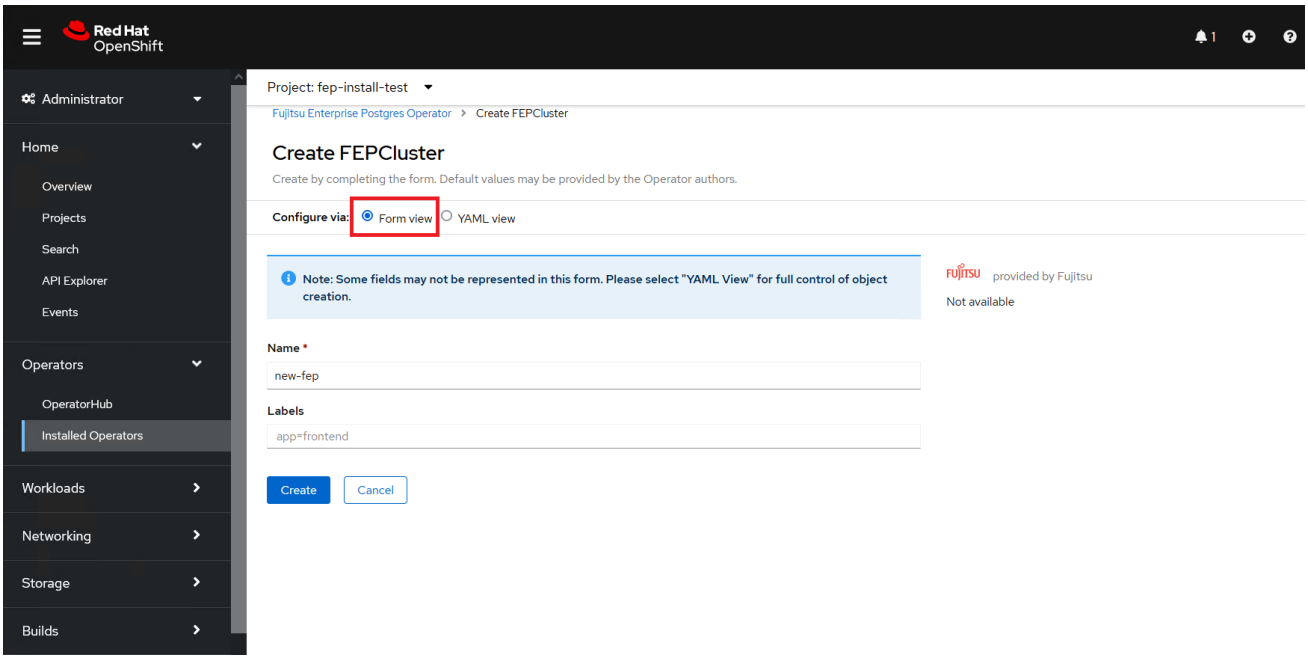


2. A page with all CRs that this operator supports will be displayed. FEPCluster is the main CR and all others are child CRs. We would create main CR and all other CRs will be created automatically by Operator.
   To create Cluster CR, either
   (1) Click on "**Create Instance**" under FEPCluster.
   OR
   (2) Click on "**FEPCluster**" on top and then click on "**Create FEPCluster**" on next page.

Copyright 2022-2024 Fujitsu Limited

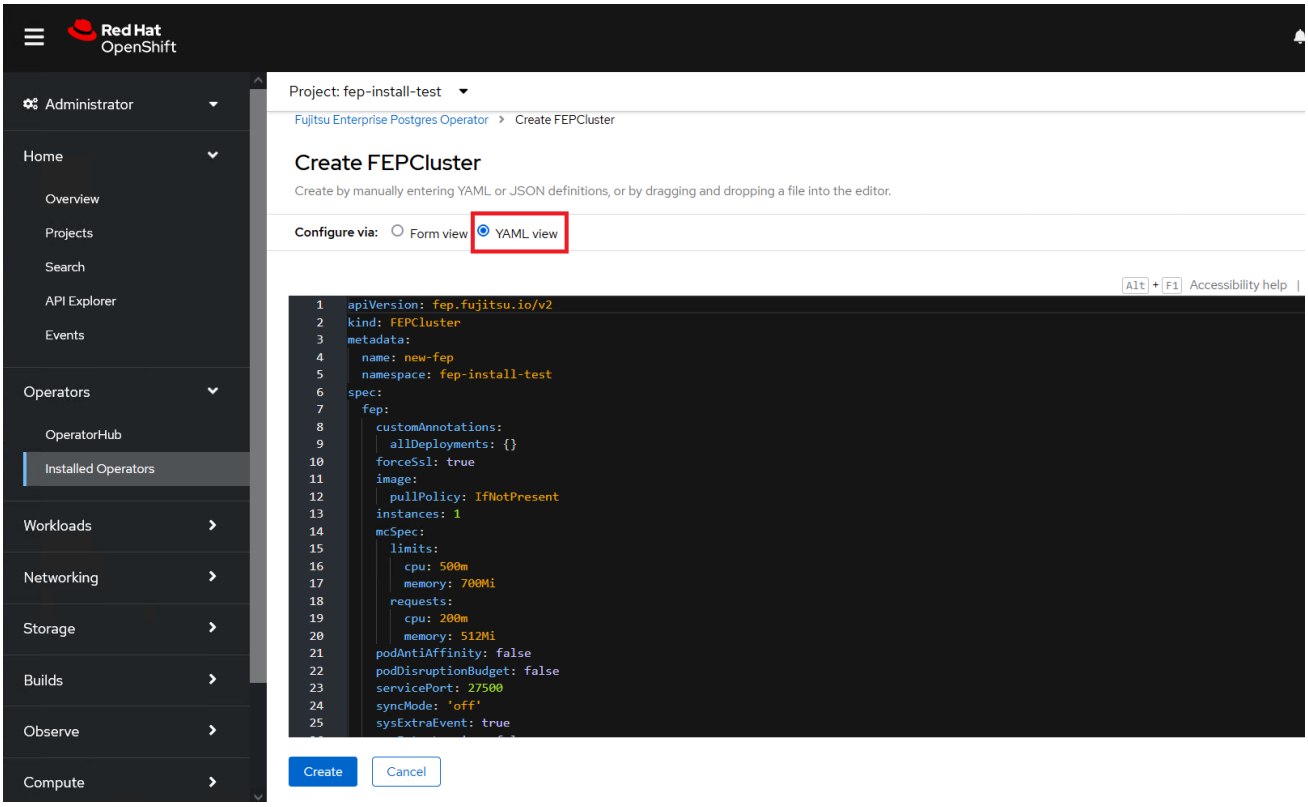


3. This will bring to "Create FEPCluster" page. Here you have two options to configure. The first one is Form View. At the moment, in Form View, one can change only the name of cluster being deployed. Default name is "new-fep".
   This name must be unique within a namespace.



4. In YAML View, starting value of CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.For example, add a configuration value for the customPgHba parameter according to your environment.

Copyright 2022-2024 Fujitsu Limited



The FEPCluster custom resource allows you to define the container's CPU, Memory, disk size, etc.

You can define each resource size individually, or you can use the following parameters to define the allocations for each resource in bulk.

Parameter	Description
spec.fep.databaseSize	Small, medium, and large define the following values for cpu/memory: small: 500m/700Mi medium: 2/4Gi large: 4/16Gi
spec.fepChildCrVal.storage.dataSize	Specifies the size of the data storage PV. Estimate and define the size of the backup storage area when the backup is enabled

We recommend that you use transparent data encryption to store your data.
Tablespace PVs are mounted in/database/tablespaces/tbspace1.
After building the database cluster, create tablespaces and tables as follows:

```
# Creating an Encrypted Tablespace
CREATE TABLESPACE secure_tablespace LOCATION '/database/tablespaces/tbspace1' WITH (tablespace_encryption_algorithm ='AES256');

# Create Table in Encrypted Tablespace
CREATE TABLE secure_table (id int, pref text, city text, data text) TABLESPACE secure_tablespace;
```

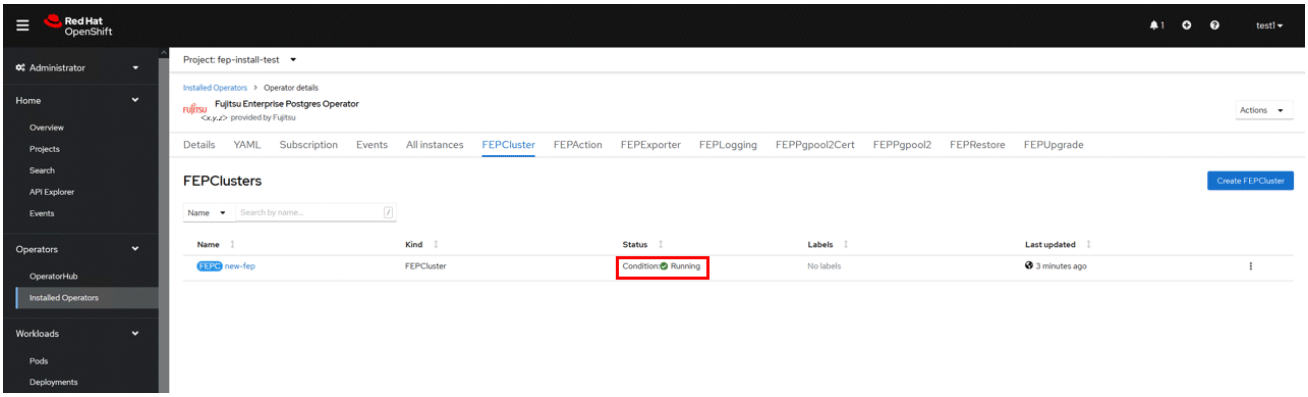
Operator provides features such as backup, audit logging, and monitoring.
These features can be enabled with the following parameters:

Copyright 2022-2024 Fujitsu Limited

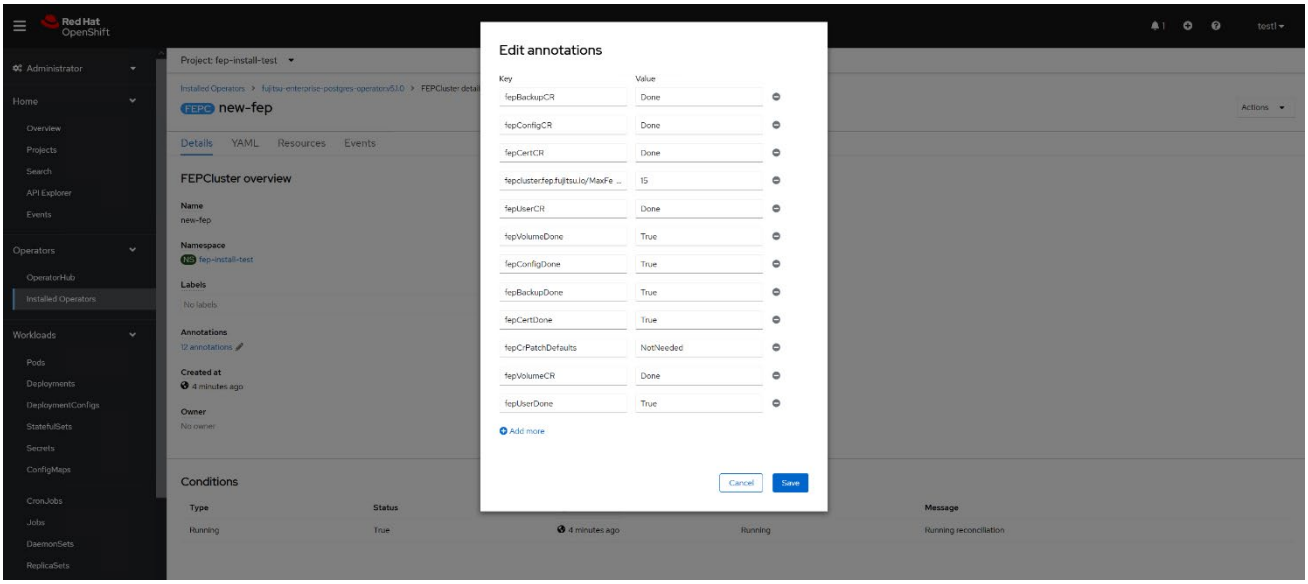
Parameter	Description
spec.fep.monitoring.enable	When true, monitoring feature is enabled.
spec.fep.pgAuditLog.enable	When true, audit log collection is enabled.
spec.fepChildCrVal.backup.type	local enables backup. The backup data is stored in the PV.
spec.fepChildCrVal.autoscale.scaleout.policy	cpu_utilization enables the ability to autoscale out replicas when CPU utilization exceeds a threshold.

Refer to the User's Guide for detailed settings and usage information for each function.

5. When "**Create**" is clicked on either of two pages above, operator creates FEPCluster CR and there after one by one FEPBackup, FEPConfig, FEPVolume, FEPUser and FEPCert child CRs are created automatically.
The starting values for child CRs are taken from "fepChildCrVal" section of FEPCluster CR yaml file. Once child CRs are created, respective values are managed through child CRs only. If you want to change the value, modify the value in FEPCluster "fepChildCrVal" section. Operator reflects changes from FEPCluster parent CR to respective child CRs. Only allowable changes are reflected in child CRs. Child CRs are marked internal objects and hence will not be visible on OCP console. However, you can check child CRs using command line tools.

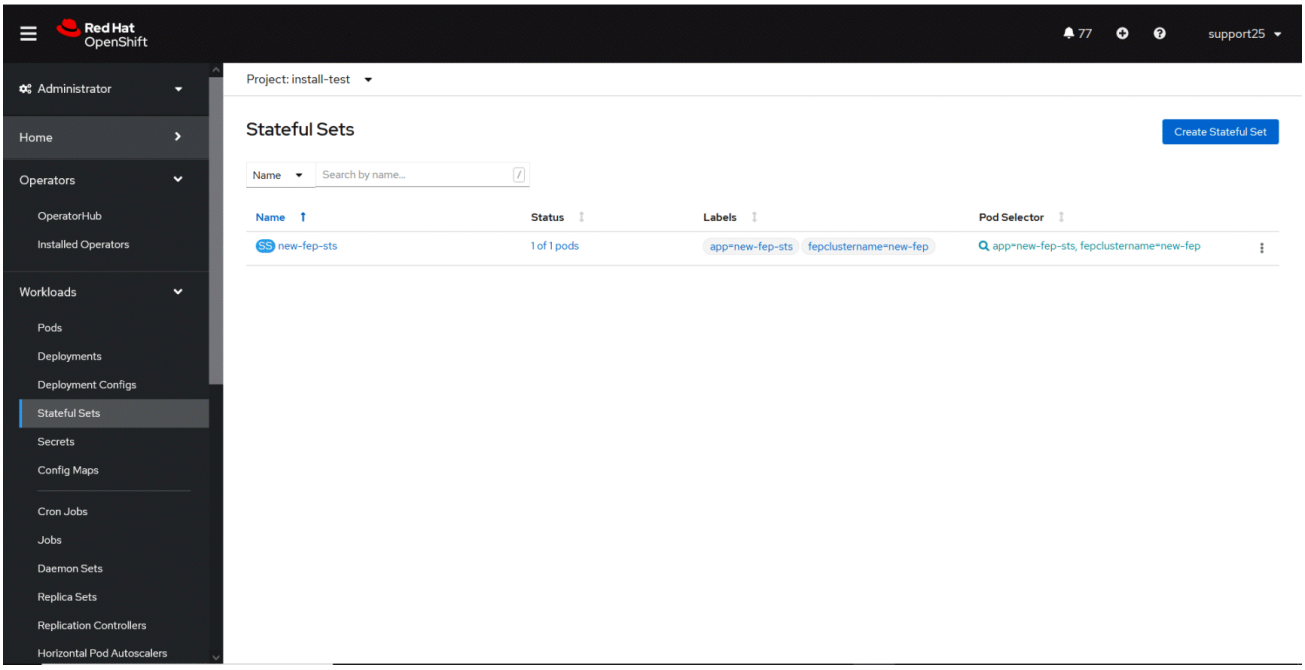


6. In FEPCluster CR, annotations are added to indicate that child CRs are created successfully and have initialized properly. It may take some time to complete.

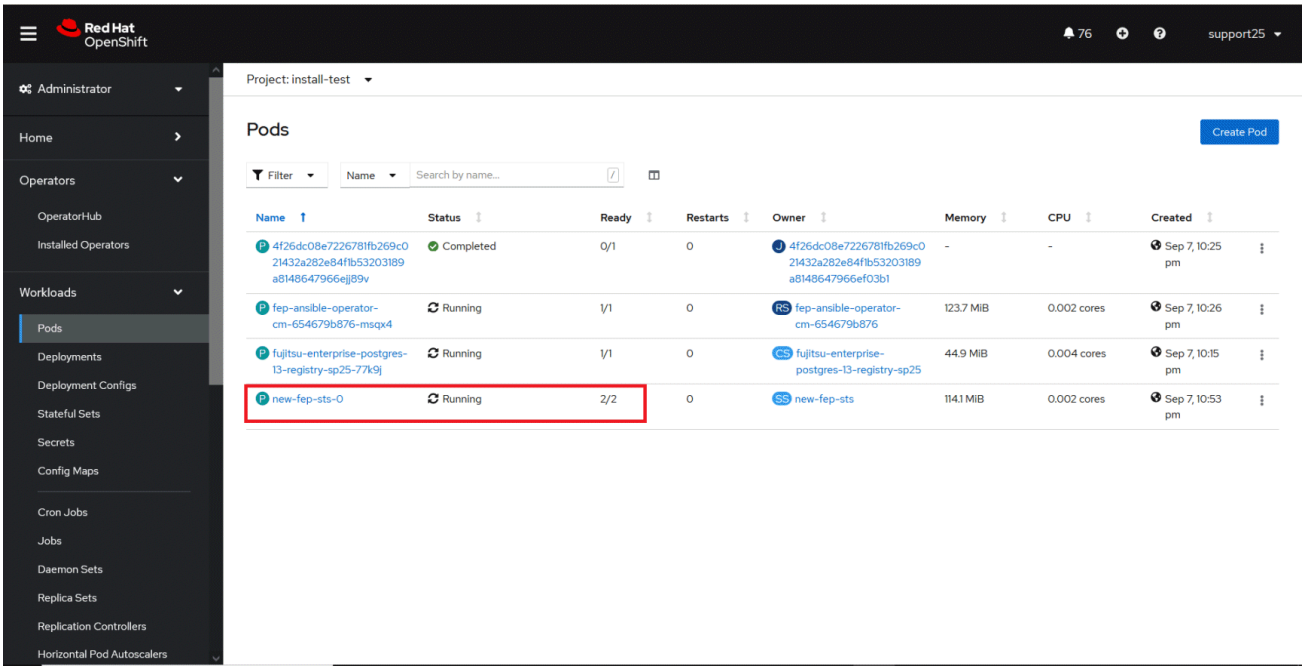


7. Once all four child CRs are marked done in annotations, operator creates StatefulSet for the cluster.

Copyright 2022-2024 Fujitsu Limited

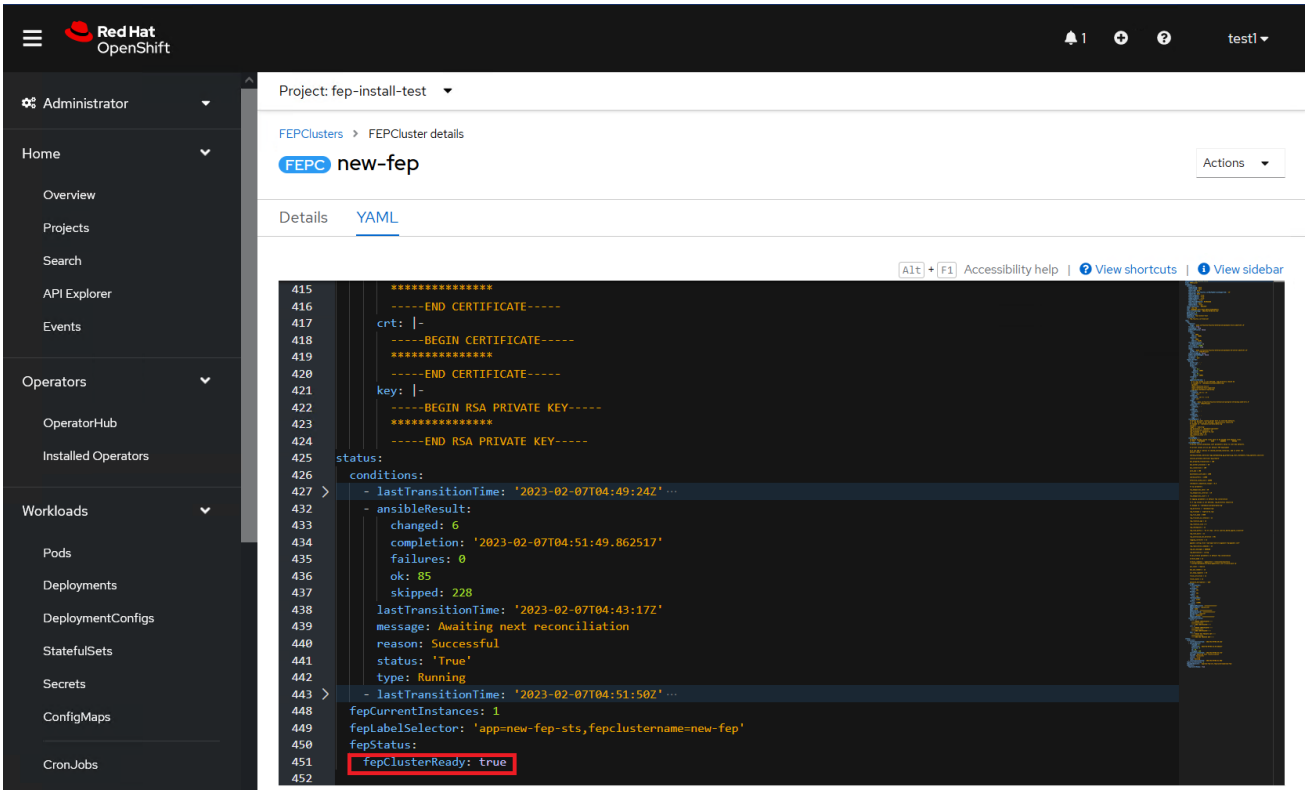


8. StatefulSet will start one FEP instance at a time and will wait for each to be ready before starting next one.



9. Once all instances of FEP servers are started, operator marks a flag "fepClusterReady" in "fepStatus" section of CR to be **true**, indicating that FEPCluster is ready for use. Looking at YAML of FEPCluster CR, it would look like as below:

Copyright 2022-2024 Fujitsu Limited



10. Operator also masks the sensitive fields like passwords, passphrase, certificates and keys in FEPCluster fepChildCrVal and also in child CRs.

11. For further information, see the Fujitsu Enterprise Postgres 17 for Kubernetes Manuals .

Copyright 2022-2024 Fujitsu Limited