FUJITSU Enterprise Postgres 14
for Kubernetes

Manual Set

Linux

FUJITSU

June-2023

Release Notes >

Overview >

User's Guide >

Reference Guide >

Quick Start Guide >

FUJITSU Enterprise Postgres 14 for Kubernetes

Release Notes

Linux

FUJITSU

J2UL-RN14-08ENZ0(00)
October-2022

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 Software Enterprise Postgres for Kubernetes FUJITSU Software Enterprise Postgres	FEP or FUJITSU Enterprise Postgres
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 Software Enterprise Postgres for Kubernetes Release Notes	Release Notes
FUJITSU Software Enterprise Postgres for Kubernetes Overview	Overview
FUJITSU Software Enterprise Postgres for Kubernetes User's Guide	User's Guide
FUJITSU Software 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 8.0: October 2022
Edition 7.0: September 2022
Edition 6.0: August 2022
Edition 5.0: July 2022
Edition 4.0: June 2022
Edition 3.0: May 2022
Edition 2.0: April 2022
Edition 1.0: March 2022
```


Copyright

Contents

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 14 Operator image tag:v4.1.7 Container image tag:ubi8-14-1.7	Container	Update Container Images
	Platform enhancement	Patching the Container's Base OS Image
FEP 14 Operator image tag:v4.1.6 Container image tag:ubi8-14-1.6	Container	Update Container Images
	OSS	Patching the PostgreSQL JDBC Driver
	Platform enhancement	Patching the Container's Base OS Image
	Security	Supports KMIP (Key Management Interoperability Protocol)
		Audit Log Logging Feature
FEP 14 Operator image tag:v4.1.5 Container image tag:ubi8-14-1.5	Container	Update Container Images
	Platform enhancement	Patching the Container's Base OS Image
FEP 14 Operator image tag:v4.1.4 Container image tag:ubi8-14-1.4	Container	Update Container Images
	Platform enhancement	Patching the Container's Base OS Image
FEP 14 Operator image tag:v4.1.3 Container image tag:ubi8-14-1.3	Container	Update Container Images
	OSS	Patching PostgreSQL
	Platform enhancement	Additional OCP Support
		Additional Kubernetes Support
		Additional Storage Support
FEP 14 Operator image tag:v4.1.2 Container image tag:ubi8-14-1.2	Platform enhancement	Patching the Container's Base OS Image
FEP 14 Operator image tag:v4.1.1 Container image tag:ubi8-14-1.1	Platform enhancement	Additional Platform Support
FEP 14 Operator image tag:v4.1.0 Container image tag:ubi8-14-1.0	OSS	PostgreSQL Rebase
	Platform enhancement	Additional OCP Support
		Additional OCS Support
		Additional Kubernetes Support
	Collaboration tools	Installing Operator with Helm Chart
		Additional Rancher Support
	Operation	Disaster Recovery
		Server Log Monitoring

Version and level	Classification	Feature
		Changing the Switchover FEPAction Interface
		Major Version Upgrade
		On-Demand Backup

1.1 Features Added FEP14 Operator in v4.1.7

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.7.

1.1.1 Container

This section explains the new feature related to container:

- Update container images

1.1.1.1 Update Container Images

The version of each container image has been updated to:

Container Name	Version Tag
fujitsu-enterprise-postgres-operator	v4.1.7
fujitsu-enterprise-postgres-14-server	ubi8-14-1.7
fujitsu-enterprise-postgres-14-backup	ubi8-14-1.7
fujitsu-enterprise-postgres-14-restore	ubi8-14-1.7
fujitsu-enterprise-postgres-14-pgpool2	ubi8-14-1.7
fujitsu-enterprise-postgres-exporter	ubi8-14-1.6
fujitsu-enterprise-postgres-fluentd	ubi8-14-1.6
fujitsu-enterprise-postgres-fluentbit	ubi8-14-1.6
fujitsu-enterprise-postgres-cronjob	ubi8-14-1.6
fujitsu-enterprise-postgres-14-upgrade	ubi8-14-1.6

1.1.2 Platform Enhancement

This section explains the improvement related to platform enhancement:

- Patching the Container's Base OS image

1.1.2.1 Patching the Container's Base OS Image

Patched the Container's Base OS image.

1.2 Features Added FEP14 Operator in v4.1.6

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.6.

1.2.1 Container

This section explains the new feature related to container:

- Update container images

1.2.1.1 Update Container Images

The version of each container image has been updated to:

Container Name	Version Tag
fujitsu-enterprise-postgres-operator	v4.1.6
fujitsu-enterprise-postgres-14-server	ubi8-14-1.6
fujitsu-enterprise-postgres-14-backup	ubi8-14-1.6
fujitsu-enterprise-postgres-14-restore	ubi8-14-1.6
fujitsu-enterprise-postgres-14-pgpool2	ubi8-14-1.6
fujitsu-enterprise-postgres-exporter	ubi8-14-1.5
fujitsu-enterprise-postgres-fluentd	ubi8-14-1.5
fujitsu-enterprise-postgres-fluentbit	ubi8-14-1.5
fujitsu-enterprise-postgres-cronjob	ubi8-14-1.5
fujitsu-enterprise-postgres-14-upgrade	ubi8-14-1.5

1.2.2 OSS

This section explains the improvement related to OSS:

- Patching the PostgreSQL JDBC driver

1.2.2.1 Patching the PostgreSQL JDBC Driver

Patched the PostgreSQL JDBC driver.

1.2.3 Platform Enhancement

This section explains the improvement related to platform enhancement:

- Patching the Container's Base OS image

1.2.3.1 Patching the Container's Base OS Image

Patched the Container's Base OS image.

1.2.4 Security

This section explains the new feature related to security:

- Supports KMIP (Key Management Interoperability Protocol)
- Audit log logging feature

1.2.4.1 Supports KMIP (Key Management Interoperability Protocol)

Supported KMIP (Key Management Interoperability Protocol).

An encryption key stored in a key management system that supports KMIP can be used as a master encryption key for transparent data encryption.

 See

Refer to "Transparent Data Encryption Using a Key Management System" in the User's Guide for details.

1.2.4.2 Audit Log Logging Feature

Regarding the FEP log feature, it is now possible to log the audit log in addition to the conventional server log.

See

Refer to "FEP Logging" in the User's Guide for details.

1.3 Features Added FEP14 Operator in v4.1.5

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.5.

1.3.1 Container

This section explains the new feature related to container:

- Update container images

1.3.1.1 Update Container Images

The version of each container image has been updated to:

Container Name	Version Tag
fujitsu-enterprise-postgres-operator	v4.1.5
fujitsu-enterprise-postgres-14-server	ubi8-14-1.5
fujitsu-enterprise-postgres-14-backup	ubi8-14-1.5
fujitsu-enterprise-postgres-14-restore	ubi8-14-1.5
fujitsu-enterprise-postgres-14-pgpool2	ubi8-14-1.5
fujitsu-enterprise-postgres-exporter	ubi8-14-1.4
fujitsu-enterprise-postgres-fluentd	ubi8-14-1.4
fujitsu-enterprise-postgres-fluentbit	ubi8-14-1.4
fujitsu-enterprise-postgres-cronjob	ubi8-14-1.4
fujitsu-enterprise-postgres-14-upgrade	ubi8-14-1.4

1.3.2 Platform Enhancement

This section explains the improvement related to platform enhancement:

- Patching the Container's Base OS image

1.3.2.1 Patching the Container's Base OS Image

Patched the Container's Base OS image.

1.4 Features Added FEP14 Operator in v4.1.4

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.4.

1.4.1 Container

This section explains the new feature related to container:

- Update container images

1.4.1.1 Update Container Images

The version of each container image has been updated to:

Container Name	Version Tag
fujitsu-enterprise-postgres-operator	v4.1.4
fujitsu-enterprise-postgres-14-server	ubi8-14-1.4
fujitsu-enterprise-postgres-14-backup	ubi8-14-1.4
fujitsu-enterprise-postgres-14-restore	ubi8-14-1.4
fujitsu-enterprise-postgres-14-pgpool2	ubi8-14-1.4
fujitsu-enterprise-postgres-exporter	ubi8-14-1.3
fujitsu-enterprise-postgres-fluentd	ubi8-14-1.3
fujitsu-enterprise-postgres-fluentbit	ubi8-14-1.3
fujitsu-enterprise-postgres-cronjob	ubi8-14-1.3
fujitsu-enterprise-postgres-14-upgrade	ubi8-14-1.3

1.4.2 Platform Enhancement

This section explains the improvement related to platform enhancement:

- Patching the Container's Base OS image

1.4.2.1 Patching the Container's Base OS Image

Patched the Container's Base OS image.

1.5 Features Added FEP14 Operator in v4.1.3

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.3.

1.5.1 Container

This section explains the new feature related to container:

- Update container images

1.5.1.1 Update Container Images

The version of each container image has been updated to:

Container Name	Version Tag
fujitsu-enterprise-postgres-operator	v4.1.3
fujitsu-enterprise-postgres-14-server	ubi8-14-1.3
fujitsu-enterprise-postgres-14-backup	ubi8-14-1.3
fujitsu-enterprise-postgres-14-restore	ubi8-14-1.3
fujitsu-enterprise-postgres-14-pgpool2	ubi8-14-1.3
fujitsu-enterprise-postgres-exporter	ubi8-14-1.2
fujitsu-enterprise-postgres-fluentd	ubi8-14-1.2
fujitsu-enterprise-postgres-fluentbit	ubi8-14-1.2
fujitsu-enterprise-postgres-cronjob	ubi8-14-1.2

Container Name	Version Tag
fujitsu-enterprise-postgres-14-upgrade	ubi8-14-1.2

1.5.2 OSS

This section explains the improvement related to OSS:

- Patching PostgreSQL

1.5.2.1 Patching PostgreSQL

Patched the PostgreSQL.

1.5.3 Platform Enhancement

This section explains the new feature related to platform enhancement:

- Additional OCP support
- Additional Kubernetes support
- Additional storage support

1.5.3.1 Additional OCP Support

The following additional OCP is supported:

- OCP 4.10

 See

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

1.5.3.2 Additional Kubernetes Support

The following additional Kubernetes is supported:

- Kubernetes 1.23

Support environments are as follows:

- Alibaba Cloud Container Service for Kubernetes
- Google Kubernetes Engine
- IBM Cloud Kubernetes Service

 See

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

1.5.3.3 Additional Storage Support

The following additional storage is supported:

- Google Cloud Storage

1.6 Features Added FEP14 Operator in v4.1.2

This section explains the improvement in FUJITSU Enterprise Postgres for Kubernetes v4.1.2.

1.6.1 Platform Enhancement

This section explains the improvement related to platform enhancement:

- Patching the Container's Base OS image

1.6.1.1 Patching the Container's Base OS Image

Patched the Container's Base OS image.

1.7 Features Added FEP14 Operator in v4.1.1

This section explains new features and improvements in FUJITSU Enterprise Postgres for Kubernetes v4.1.1.

1.7.1 Platform Enhancement

This section explains the new features related to platform enhancement:

- Additional platform support

1.7.1.1 Additional Platform Support

The following additional Platform is supported:

- IBM Power Systems

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

1.8 Features Added FEP14 Operator in v4.1.0

This section explains new features and improvements in FUJITSU Enterprise Postgres for Kubernetes v4.1.0.

1.8.1 OSS

This section explains the new feature related to OSS:

- PostgreSQL rebase

1.8.1.1 PostgreSQL Rebase

The PostgreSQL version that FUJITSU Enterprise Postgres is based on is 14.0.

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

1.8.2 Platform Enhancement

This section explains the new features related to platform enhancement:

- Additional OCP support
- Additional OCS support
- Additional Kubernetes support

1.8.2.1 Additional OCP Support

The following additional OCP is supported:

- OCP 4.9

 See

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

1.8.2.2 Additional OCS Support

The following additional OCS is supported:

- OCS 4.8

 See

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

1.8.2.3 Additional Kubernetes Support

The following additional Kubernetes is supported:

- Kubernetes 1.21
- Kubernetes 1.22

Support environments are as follows:

- Azure Kubernetes Service
- Amazon EKS
- Rancher Kubernetes Engine (on Linux hosts)

 See

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

1.8.3 Collaboration Tools

This section explains the new features related to installation:

- Installing Operator with Helm Chart
- Additional Rancher support

1.8.3.1 Installing Operator with Helm Chart

Supports Operator installation and upgrades with Helm Chart. This makes it easy to deploy operators to your Kubernetes cluster.

 See

Refer to "Using the Helm Chart" in the User's Guide for details.

1.8.3.2 Additional Rancher Support

Supports Rancher. This makes you to deploy FEP Operators on Kubernetes clusters and deploy/update various custom resources (such as FEPCluster) from the Rancher UI.

 See

Refer to "Using the Rancher UI" in the User's Guide for details.

1.8.4 Operation

This section explains the new features related to operation:

- Disaster Recovery
- Server Log Monitoring
- Changing the Switchover FEPAction Interface
- Major Version Upgrad
- On-Demand Backup

1.8.4.1 Operator Support Feature Matrix

This Operator release supports deploying FEP14 and earlier release (FEP13 and FEP12) Cluster. For supported feature on each release, please refer to table below.

Feature	FEP12	FEP13	FEP14
fep server	Y	Y	Y
backup sidecar	Y	Y	Y
restore	Y	Y	Y
pgpool2	Y	Y	Y
fep exporter	-	Y	Y
fep logging (fluentD)	-	-	Y
fep logging sidecar (fluentbit)	-	-	Y
pgBadger	-	-	Y

Y: Supported

-: Not supported

1.8.4.2 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.

 See

Refer to "Disaster Recovery" in the User's Guide for details.

1.8.4.3 Server Log Monitoring

Adds server log monitoring to the monitoring item. You can also use pgBadger to analyze logs.

See

Refer to "FEP Logging" in the User's Guide for details.

1.8.4.4 Changing the Switchover FEPAction Interface

Previously, the parameter required the specification of the master POD, but this is no longer necessary.

See

Refer to "FEPAction Custom Resource Parameters" in the Reference for details.

1.8.4.5 Major Version Upgrade

You can upgrade a major version of FEP by specifying the upgrade parameters when you create the latest FEP cluster.

See

Refer to "Major Version Upgrade" in the Overview.

1.8.4.6 On-Demand Backup

On-demand backup can be taken at any time other than a preset schedule.

See

Refer to "On-Demand Backup" in the Overview.

FUJITSU Enterprise Postgres 14 for Kubernetes

Overview

Linux

FUJITSU

J2UL-OV14-03ENZ0(00)
September-2022

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:

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
FUJITSU Software Enterprise Postgres for Kubernetes FUJITSU Software Enterprise Postgres	FEP or FUJITSU Enterprise Postgres
Custom Resource	CR
Custom Resource Definition	CRD
Persistent Volume	PV
GAP	Grafana, Alert Manager, Prometheus

Abbreviations of manual titles

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

Full Manual Title	Abbreviations
FUJITSU Software 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 3.0: September 2022
Edition 2.0: April 2022
Edition 1.0: March 2022
```

Copyright

Contents

Chapter 1 Know about the Product

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

1.1 What is FUJITSU Software Enterprise Postgres for Kubernetes?

FUJITSU Software Enterprise Postgres for Kubernetes provides automated operations for installing and managing your FUJITSU Enterprise Postgres 14 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.

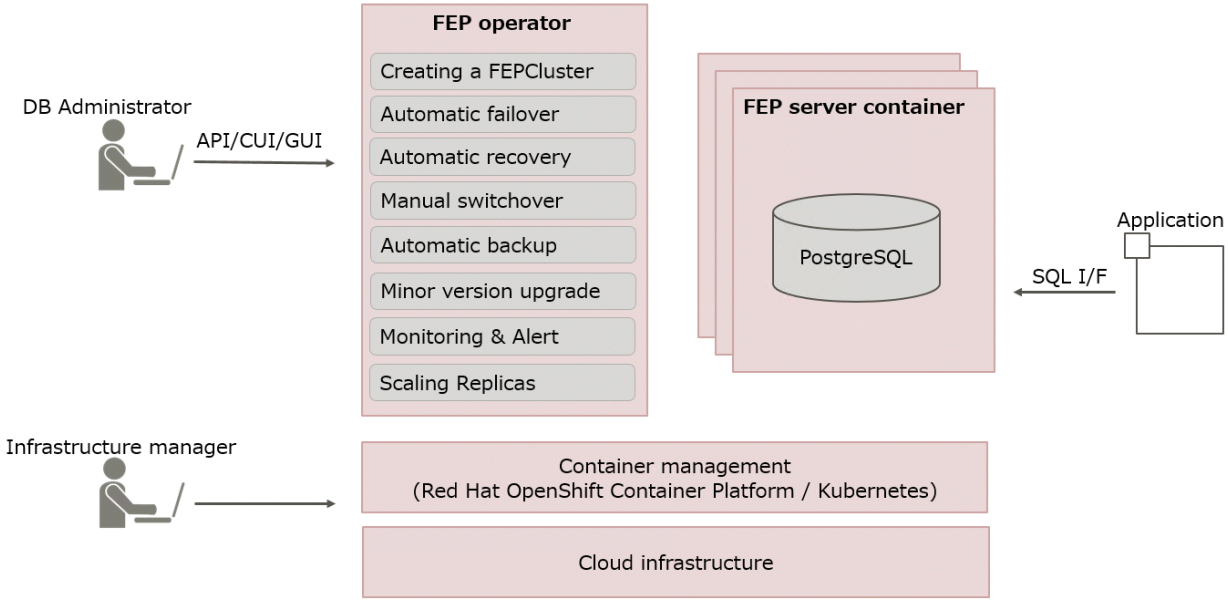
FEP backup container: Contains the FEP server software to perform scheduled backup operations.

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 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 server component.

In principle, a running FEP server container is considered as equivalent to a FUJITSU 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
- Minor Version Upgrade
- FEP Features
- Monitoring & Alert
- Scaling Replicas
- Disaster Recovery

1.2.1 Cluster Deployment

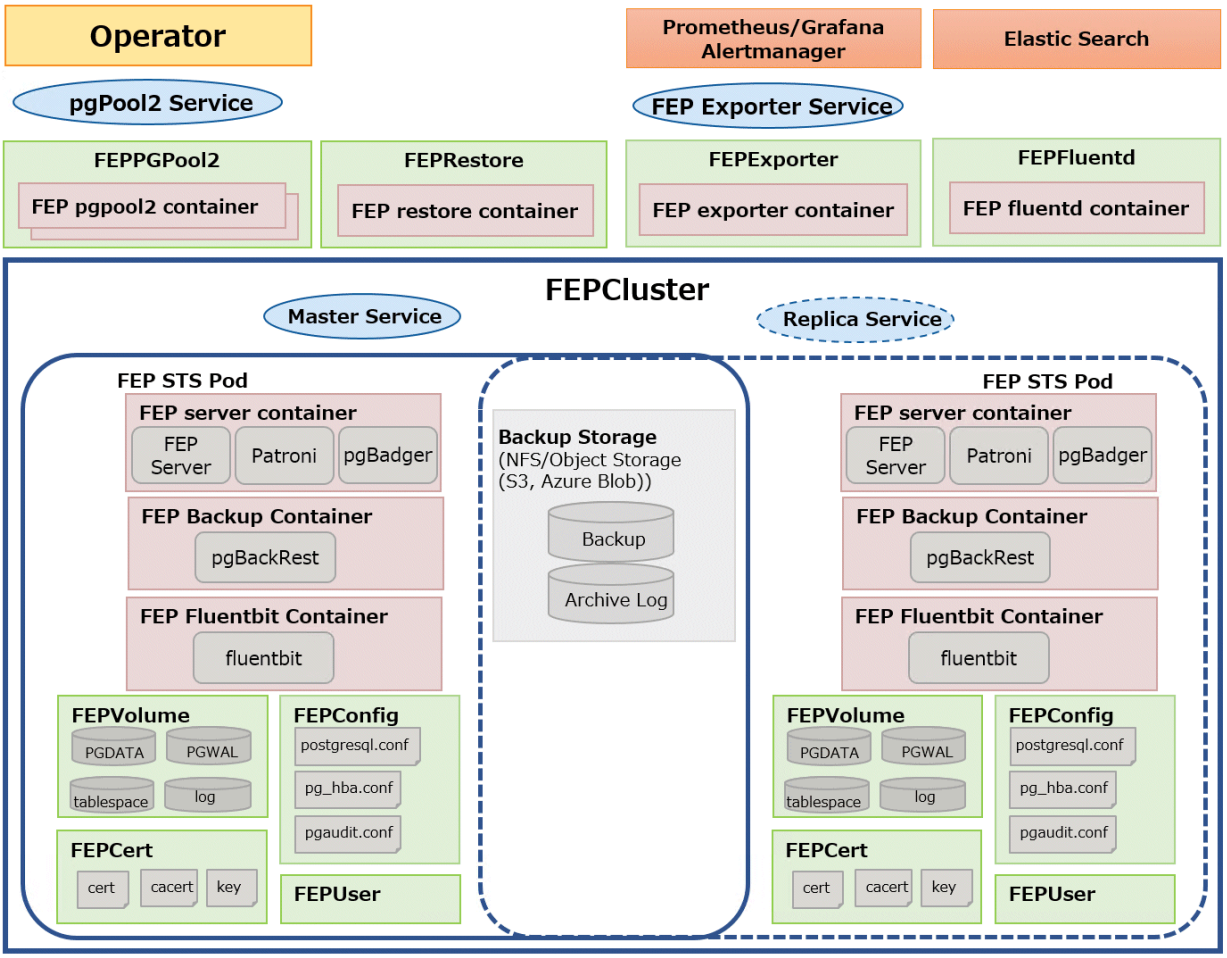
1.2.1.1 Creating a FEPCluster

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. 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.

1.2.5 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.

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 Monitoring & Alert

1.2.7.1 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

1.2.7.2 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.

1.2.8 Scaling Replicas

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

1.2.8.1 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.

Note

- 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.

1.2.8.2 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.

1.2.9 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.

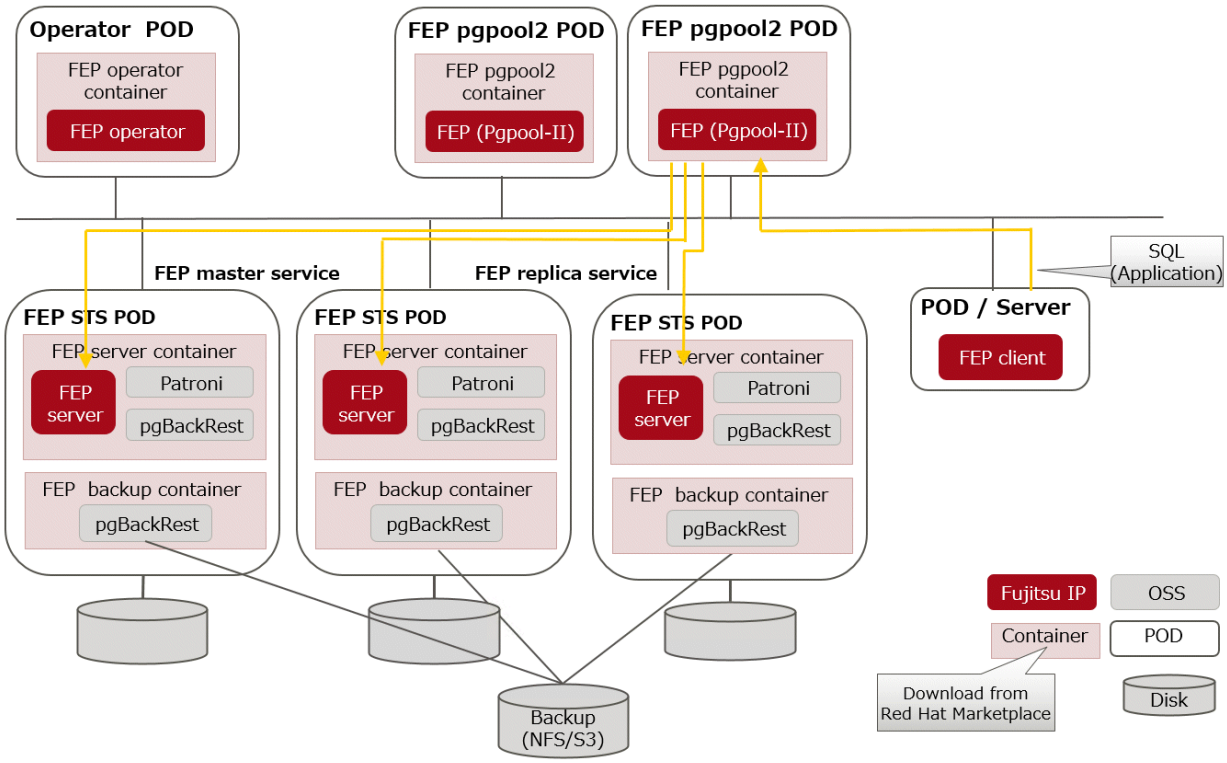
Even if it is difficult to operate in an OCP environment with a database cluster due to a disaster, it is possible to continue operating in a different OCP environment.

1.3 Operator System Configuration

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

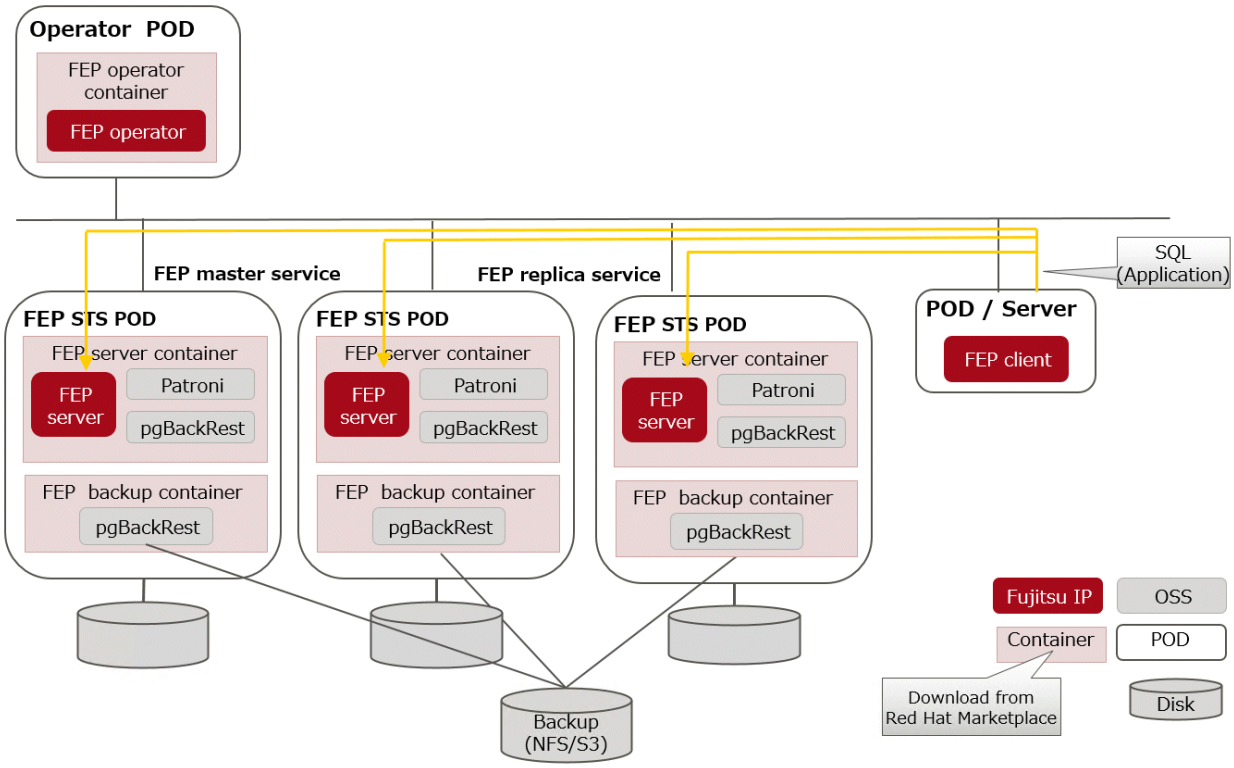
Example) Deployment with Pgpool-II

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.


Operator POD
FEP operator container
FEP operator
FEP master service
FEP replica service
SQL (Application)
FEP STS POD
FEP server container
FEP server
Patroni
pgBackRest
FEP backup container
pgBackRest
FEP STS POD
FEP server container
FEP server
Patroni
pgBackRest
FEP backup container
pgBackRest
FEP STS POD
FEP server container
FEP server
Patroni
pgBackRest
FEP backup container
pgBackRest
POD / Server
FEP client
Backup (NFS/S3)
Fujitsu IP
OSS
Container
POD
Download from Red Hat Marketplace
Disk

Chapter 2 Know What it does

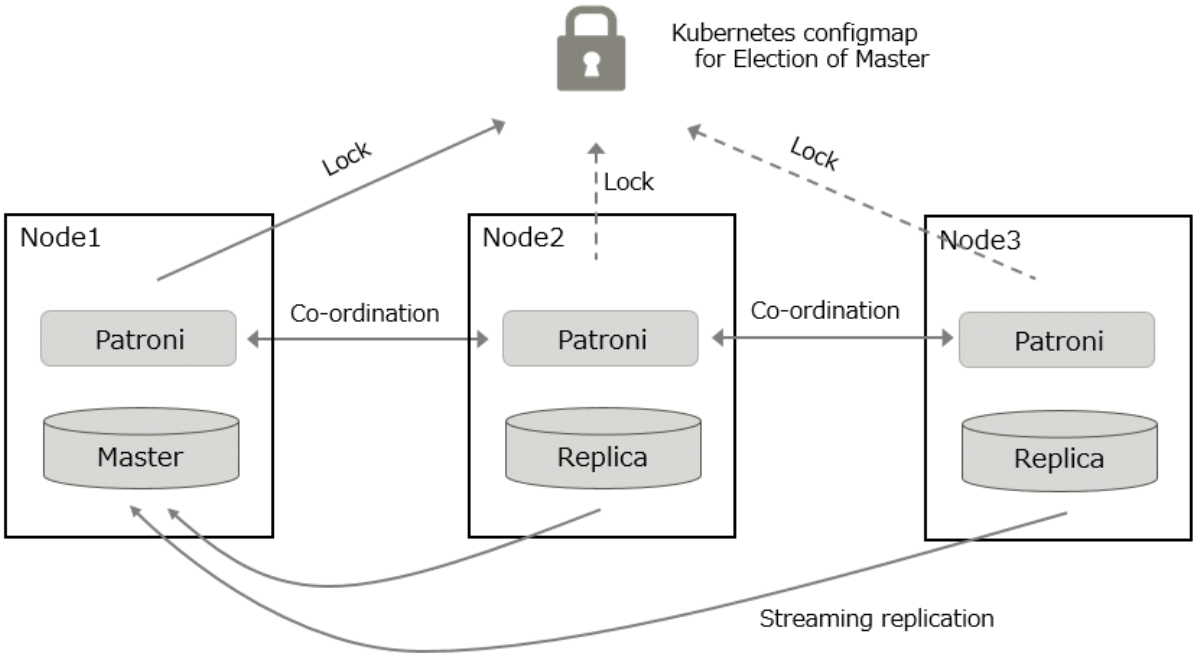
This chapter explains what you need to do.

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.

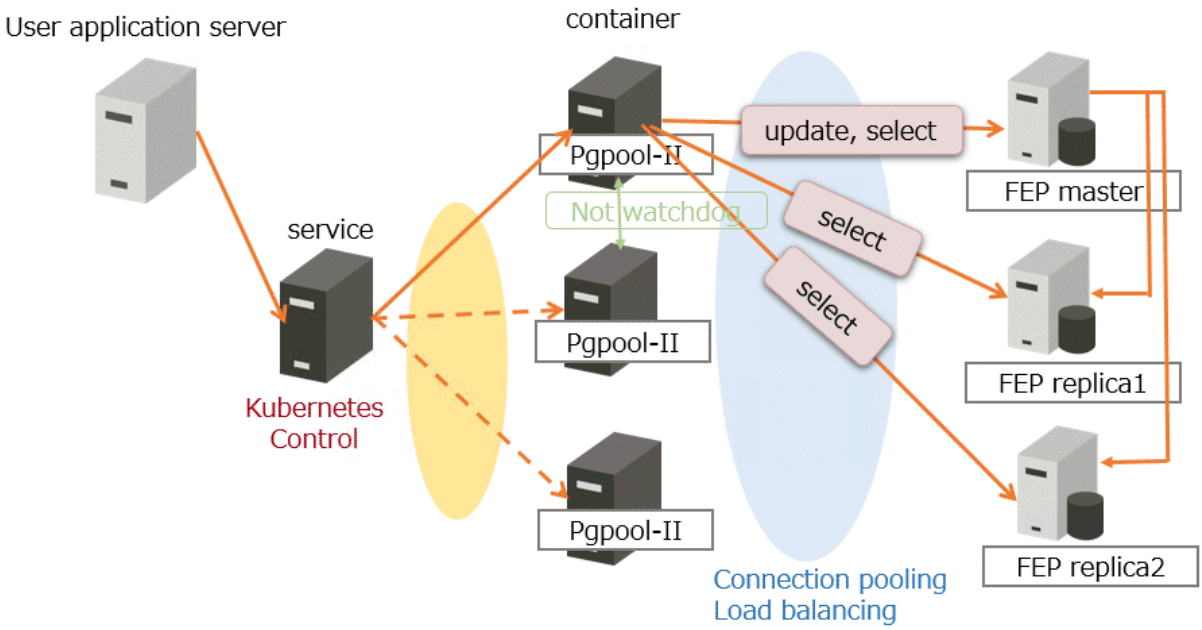
2.5 Configurable Volume Per Cluster

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.

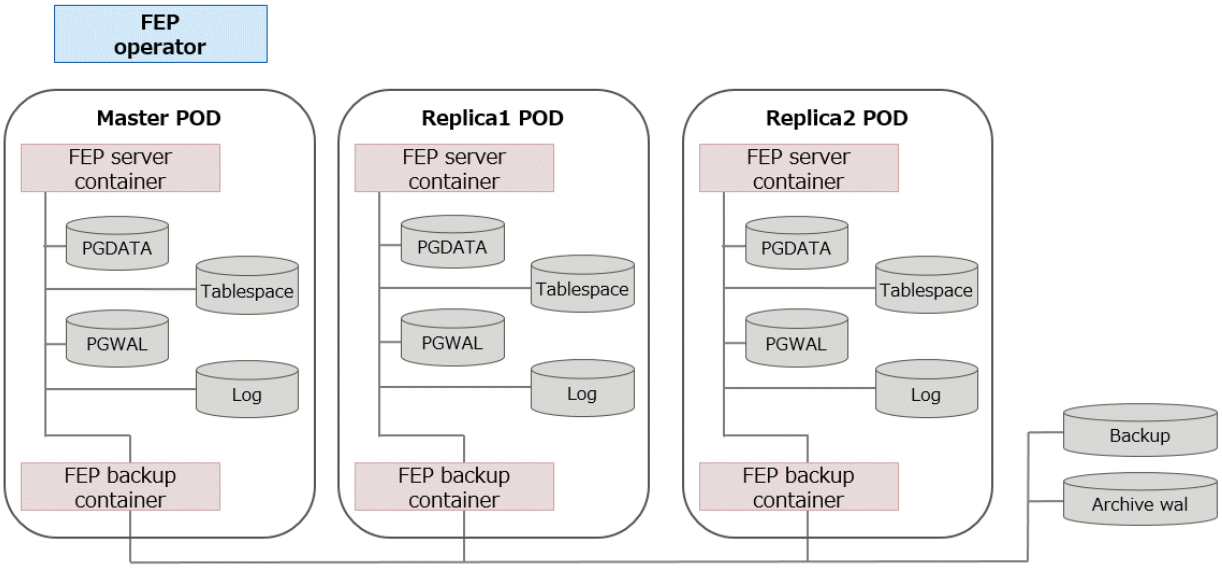


2.7 Backup

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.

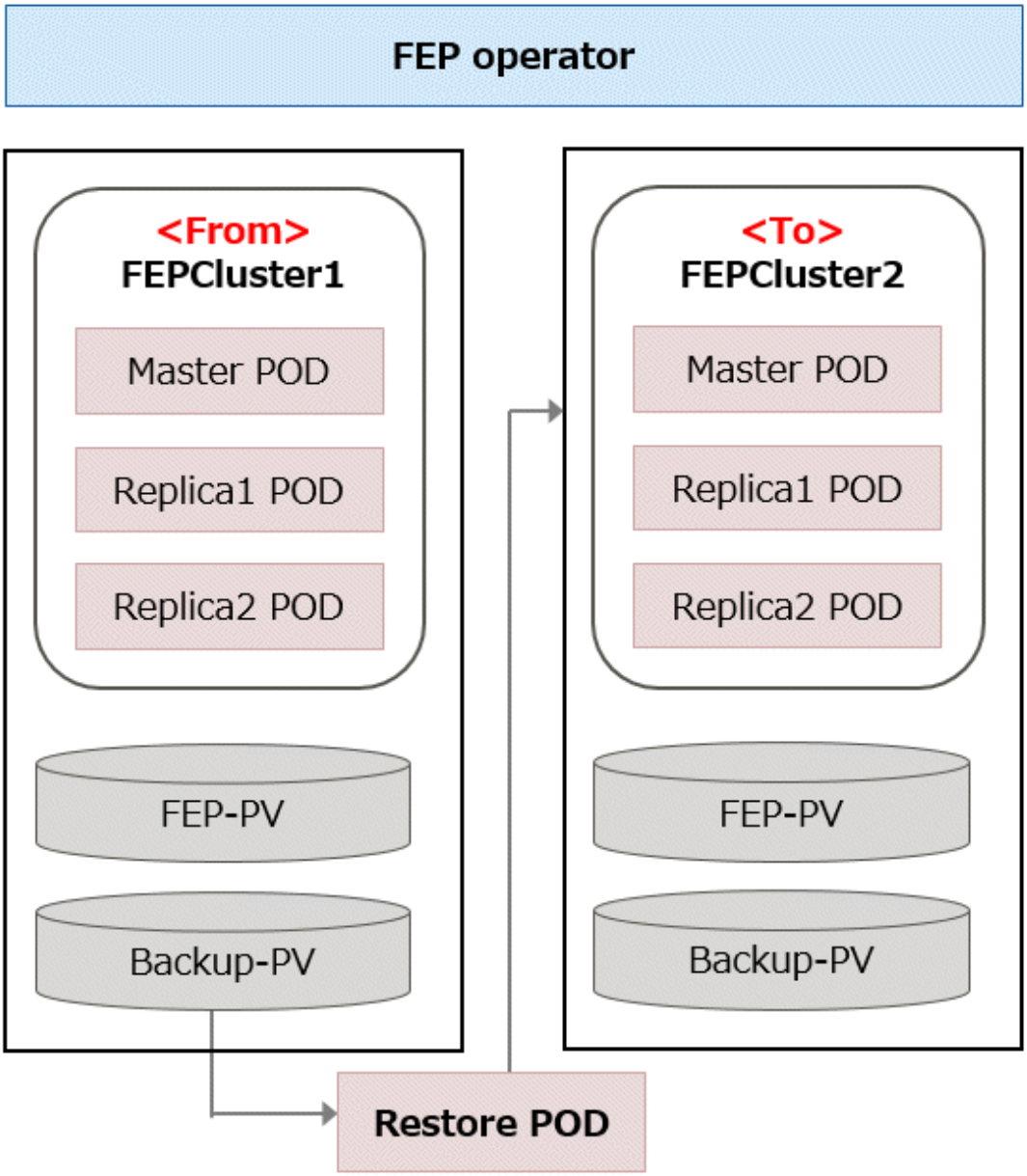
The backup storage location and retention period settings are the same as those for scheduled backups.

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.

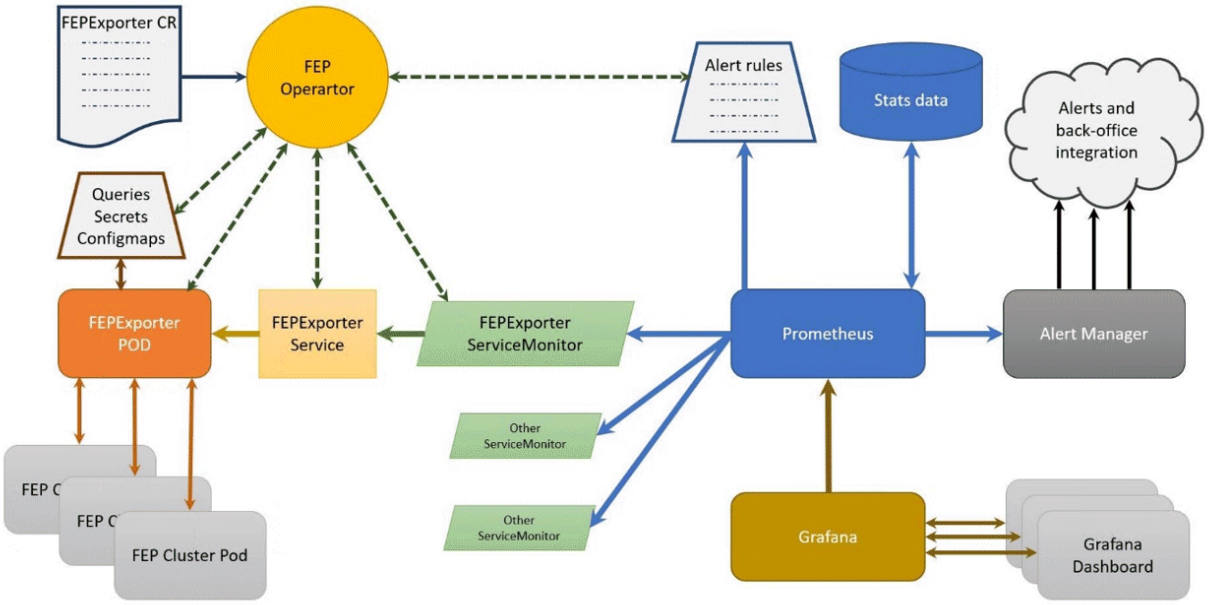
Prometheus is a condensed way to store time-series metrics. Grafana provides a flexible and visually pleasing interface to view graphs and guages of FEP metrics stored in Prometheus.

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.

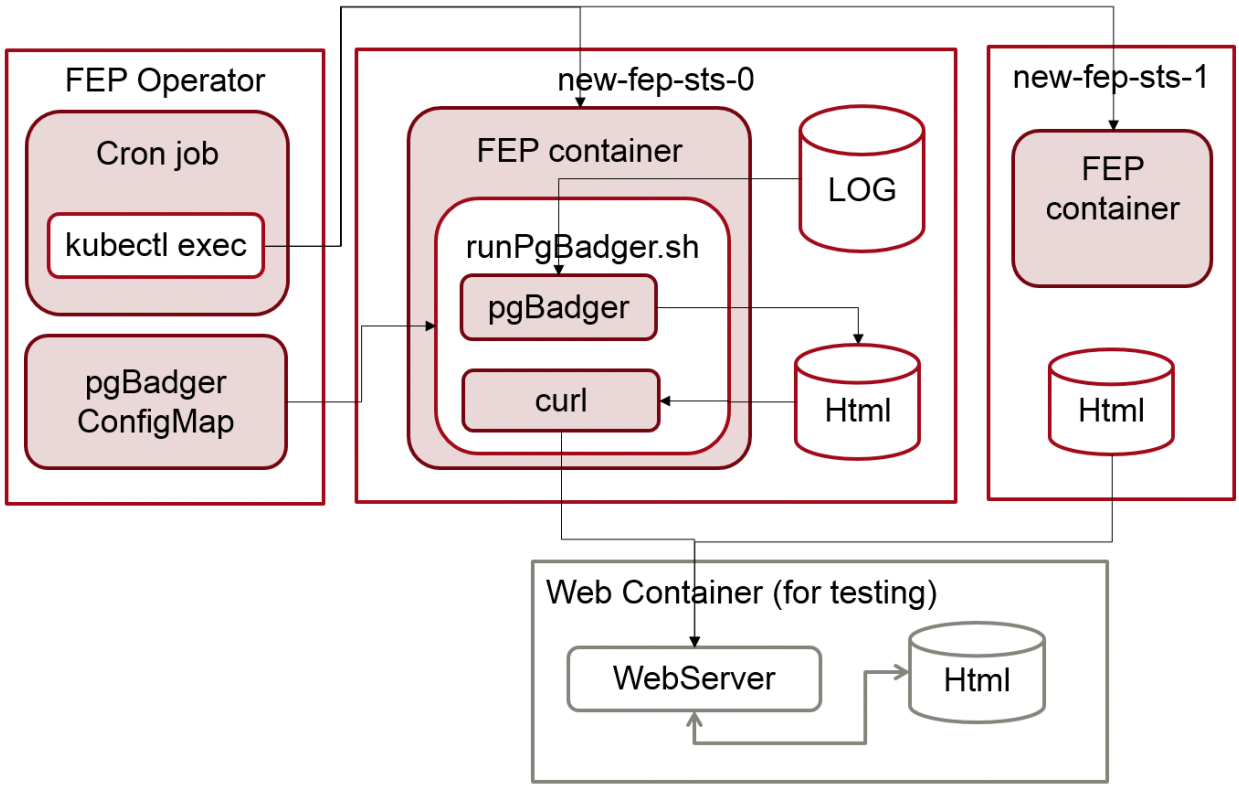
Metrics from FEP Cluster(s) is collected by Prometheus through optional components deployed using FEP Exporter with default set of metrices and corresponding Prometheus rules to raise alerts. User may extend or overwrite metrics by defining their custom metrics queries and define their custom Prometheus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

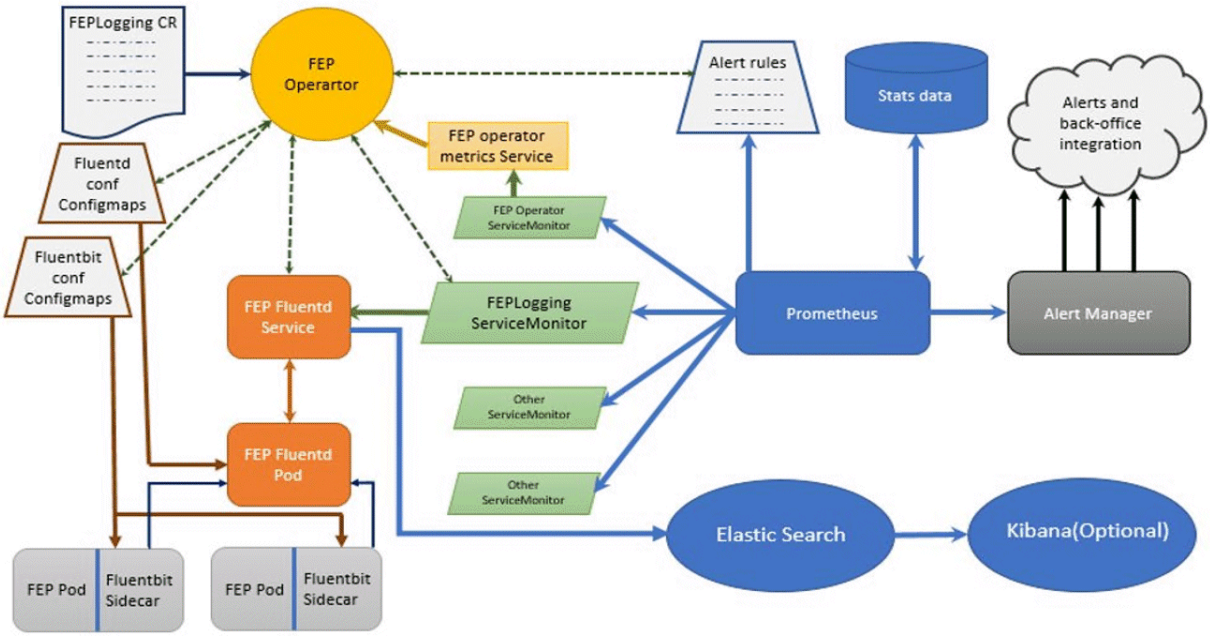


2.10.2 Prometheus and Elasticsearch Log Monitoring

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.

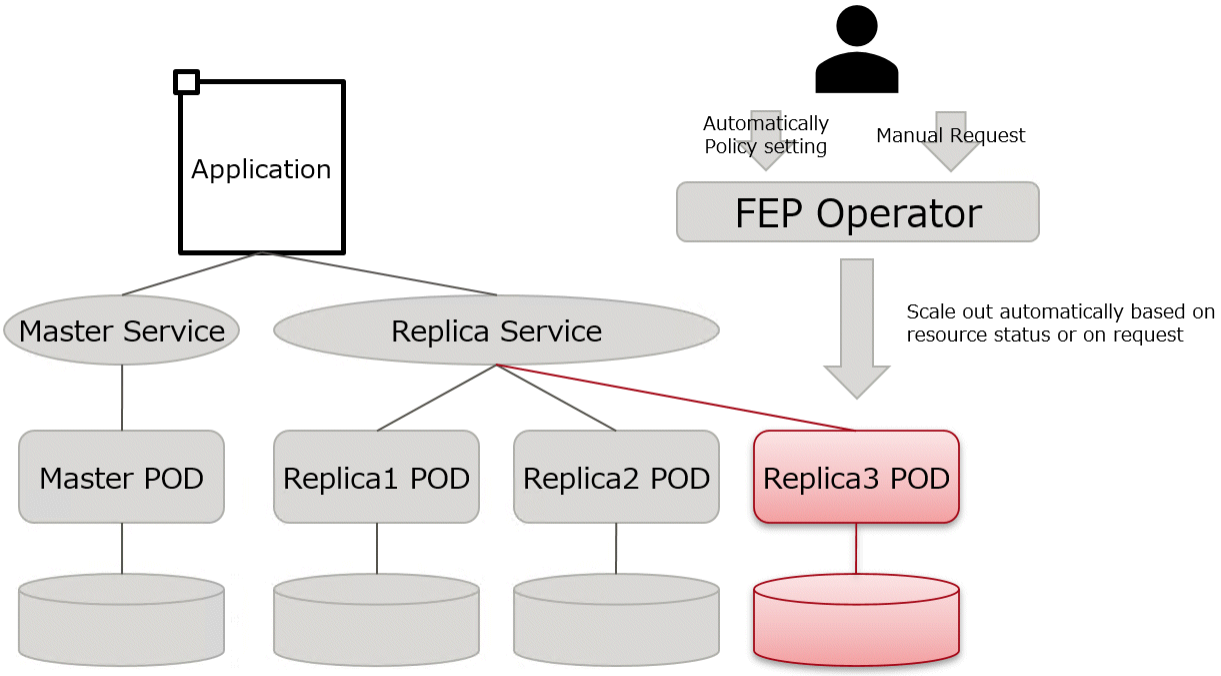
Fluentd counts the occurrence of different severity levels (PANIC, FATAL, ERROR, WARNING, DEBUG etc) by filtering the incoming log records and the counts are passed onto Prometheus along with the logfile name. The log file name makes it easier to investigate the reason of severity/issue.

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.

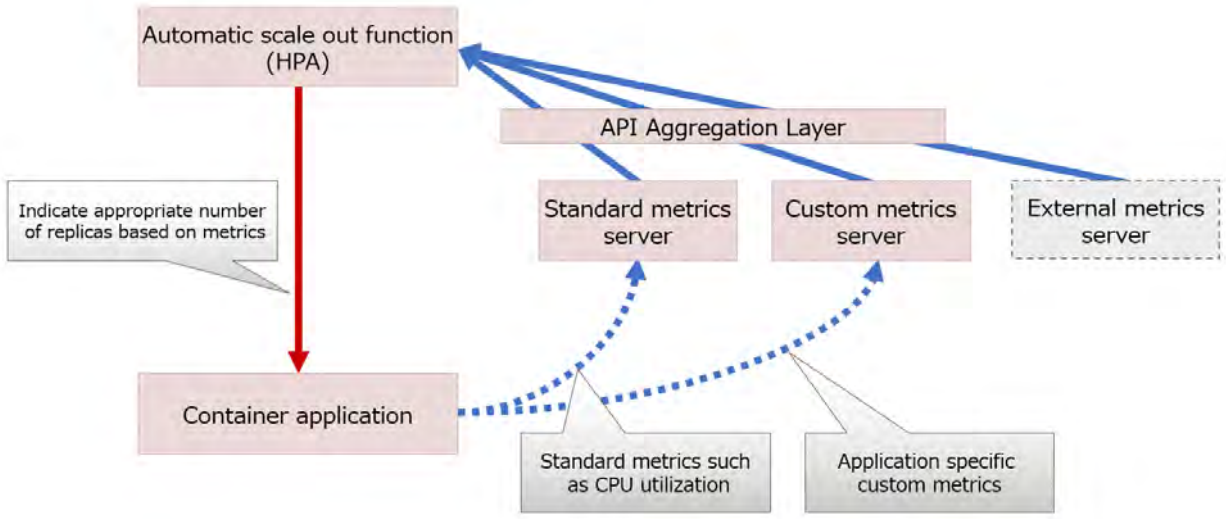
- Standard metrics server
- Custom metrics server
- External metrics server

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.

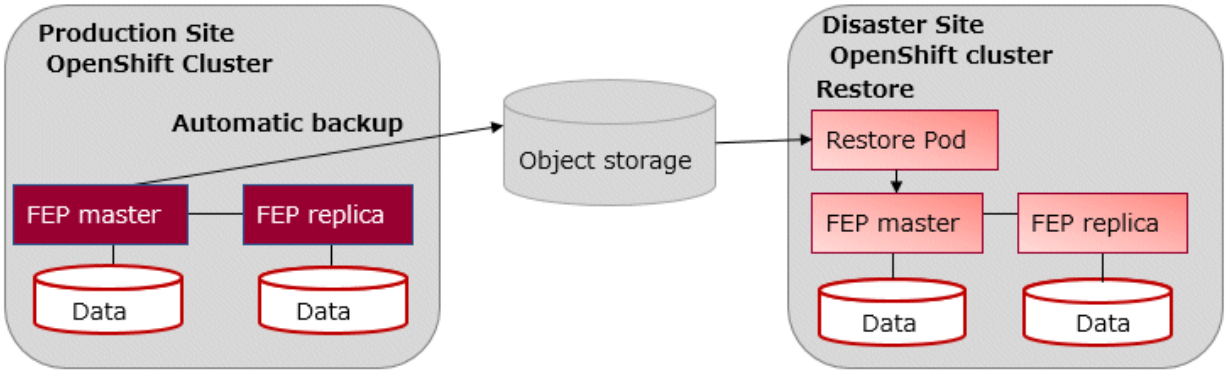
The custom metrics server must also reference Prometheus, which collects custom metrics for FEP clusters, and must be configured to publish custom metrics for FEP clusters.



2.12 Disaster Recovery

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

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



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	14.0	Database management system	PostgreSQL Documentation
orafce	3.17.0	Oracle-compatible SQL features	"Compatibility with Oracle Databases" in the FUJITSU Enterprise Postgres Application Development Guide
Pgpool-II	4.2.6	Failover, connection pooling, load balancing, etc.	"Pgpool-II" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
pg_statsinfo	13.0	Collection and accumulation of statistics	"pg_statsinfo" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
pg_hint_plan	13.1.3.7	Tuning (statistics management, query tuning)	- "pg_hint_plan" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server - "Optimizer Hints" in the FUJITSU Enterprise Postgres Application Development Guide
pg_dbms_stats	1.5.0		- "pg_dbms_stats" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server - "Locked Statistics" in the FUJITSU Enterprise Postgres Application Development Guide
pg_repack	1.4.7	Table reorganization	"pg_repack" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
pg_rman	1.3.13	Backup and restore management	"pg_rman" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
pgBadger	11.6	Log analysis	"pgBadger" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
pg_bigm	1.2	Full-text search (multibyte)	"pg_bigm" in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server
PostgreSQL JDBC driver	42.2.23	JDBC driver	"JDBC Driver" in the FUJITSU Enterprise Postgres Application Development Guide
psqlODBC	13.02.0000	ODBC driver	"ODBC Driver" in the FUJITSU Enterprise Postgres Application Development Guide
pgBackRest	2.36	Backup and restore management	"Scheduling Backup from Operator" in the User's Guide
patroni	2.1.2	Postgres cluster management	"High Availability" in the User's Guide
postgres_exporter	0.9.0	Postgresql metrics monitoring capabilities for Prometheus with Fujitsu updated queries	“Monitoring" in the User's Guide

FUJITSU Enterprise Postgres 14 for Kubernetes

User's Guide

Linux

FUJITSU

J2UL-UG14-06ENZ0(00)
June-2023

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.

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
FUJITSU Software Enterprise Postgres for Kubernetes FUJITSU Software Enterprise Postgres	FEP or FUJITSU Enterprise Postgres
Vertical Clustered Index	VCI
Transparent Data Encryption	TDE
Point in time recovery	PITR
Custom Resource	CR
Custom Resource Definition	CRD
Persistent Volume	PV
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 Software Enterprise Postgres for Kubernetes Release Notes	Release Notes
FUJITSU Software Enterprise Postgres for Kubernetes Overview	Overview
FUJITSU Software Enterprise Postgres for Kubernetes User's Guide	User's Guide
FUJITSU Software 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 6.0: June 2023
Edition 5.0: October 2022
Edition 4.0: September 2022
Edition 3.0: June 2022
```

Edition 2.0: April 2022
Edition 1.0: March 2022

Copyright

Contents

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	8	Meant to provide base OS image for the container
2	FUJITSU Enterprise Postgres Server	14.0	To provide server capabilities
3	Patroni	2.1.2	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		4.10, 4.11, 4.12
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	1.24, 1.25, 1.26

*1: Kubernetes 1.24 - 1.26

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	3.7.2 and later	- Kubernetes only Helm Web Site https://helm.sh/docs/intro/install/
5	Rancher	v2.6.2 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

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.

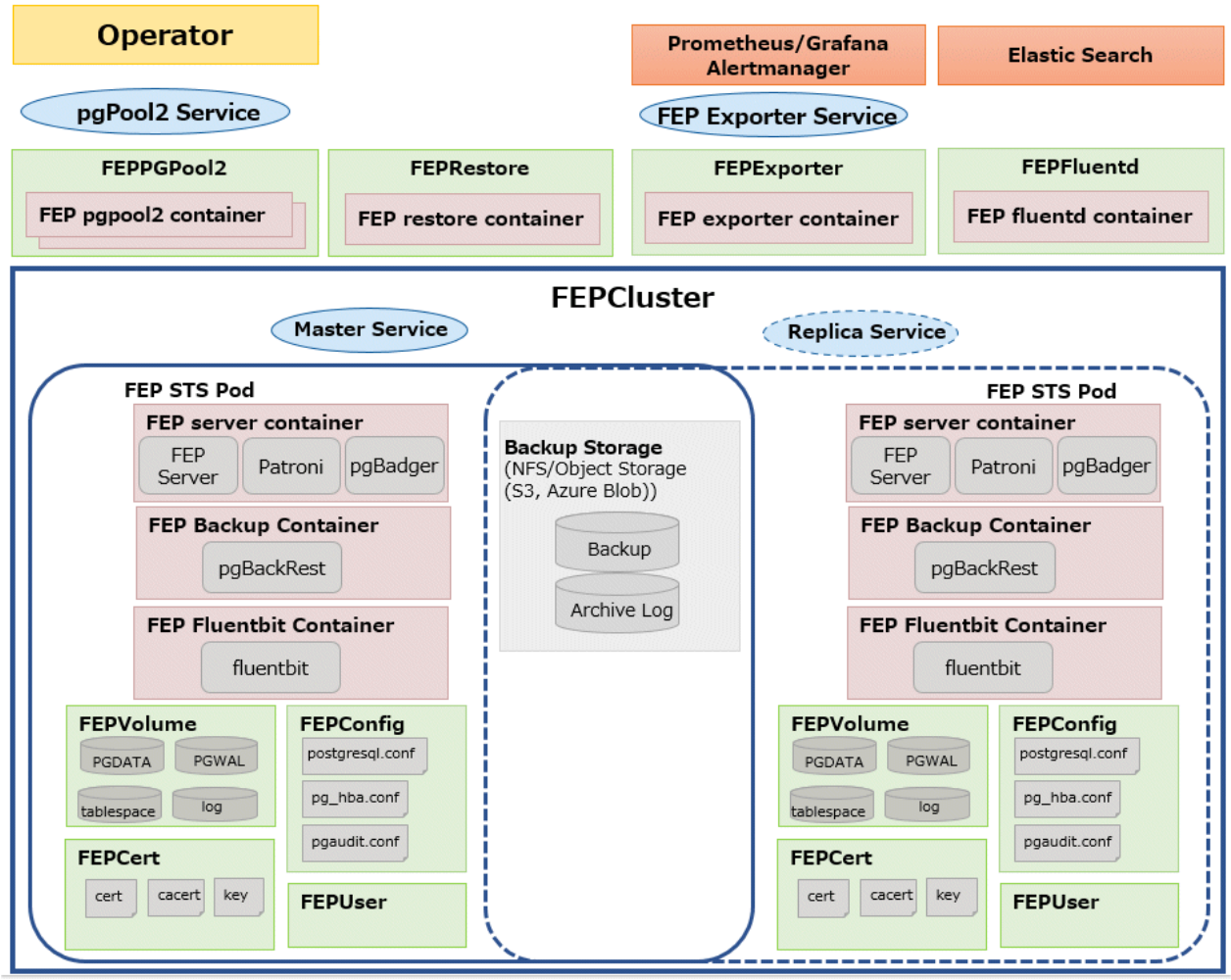
Task	Design required to operate FEP	Where to find
FEP setup	Required.	2.3.1 Deployment
High availability configuration	Optional. (When checking or changing the behavior of high availability. However, even by default, constant high availability operation is possible.)	2.3.2 High Availability
Volume settings	Optional. (When setting the volume. However, even by default, allocate a fixed volume.)	2.3.3 Configurable Volume per Cluster
Pgpool-II setup	Optional. (When using Pgpool-II.)	2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator
Backup/restore settings	Optional. (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)	Optional. (When using Monitoring and Alert)	2.3.8 Monitoring & Alert (FEPExporter)
Scaling Replicas	Optional. (When using scaling feature)	2.3.9 Scaling Replicas

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.

Configuration server type	Description
Backup storage	Storage where backup data is stored. If you do not need to obtain a backup, you do not need to create one.
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.
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.

User type	User name	Description
Infrastructure administrator	Mandatory	A system administrator (superuser) who has root privileges on all the servers that make up this product.
Database administrator	Mandatory	Install, set up, start, stop, and perform operation and maintenance of this product.
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-14-server:*OS*-*FEPBaseVersion*-*MajorVersion*.*MinorVersion*-*ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi8	
FEPBaseVersion	14	
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-14-server:ubi8-14-1.1
  - fujitsu-enterprise-postgres-14-server:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-14-server:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-14-server:ubi8-14-1.1-ppc64le

FEP backup container

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

fujitsu-enterprise-postgres-14-backup:*OS*-*FEPBaseVersion*-*MajorVersion*.*MinorVersion*-*ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi8	
FEPBaseVersion	14	
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-14-backup:ubi8-14-1.1
  - fujitsu-enterprise-postgres-14-backup:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-14-backup:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-14-backup:ubi8-14-1.1-ppc64le

FEP restore container

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

fujitsu-enterprise-postgres-14-restore:*OS*-*FEPBaseVersion*-*MajorVersion*.*MinorVersion*-*ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi8	
FEPBaseVersion	14	
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-14-restore:ubi8-14-1.1
  - fujitsu-enterprise-postgres-14-restore:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-14-restore:ubi8-14-1.1-s390x

- fujitsu-enterprise-postgres-14-restore:ubi8-14-1.1-ppc64le

FEP pgpool2 container

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

fujitsu-enterprise-postgres-14-pgpool2: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi8	
FEPBaseVersion	14	
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-14-pgpool2:ubi8-14-1.1
  - fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.1-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	ubi8	
FEPBaseVersion	14	
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:ubi8-14-1.1
  - fujitsu-enterprise-postgres-exporter:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-exporter:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-exporter:ubi8-14-1.1-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	ubi8	
FEPBaseVersion	14	
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:ubi8-14-1.1
  - fujitsu-enterprise-postgres-fluentd:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-fluentd:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-fluentd:ubi8-14-1.1-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	ubi8	
FEPBaseVersion	14	
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:ubi8-14-1.1
  - fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.1-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	ubi8	
FEPBaseVersion	14	
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:ubi8-14-1.1
  - fujitsu-enterprise-postgres-cronjob:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-cronjob:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-cronjob:ubi8-14-1.1-ppc64le

FEP Upgrade container

FEP Upgrade container as for FEP server containers.

fujitsu-enterprise-postgres-14-upgrade: *OS-FEPBaseVersion-MajorVersion.MinorVersion-ARCH*

For each *Version*, specify the following:

Field	Values	Description
OS	ubi8	
FEPBaseVersion	14	
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-14-upgrade:ubi8-14-1.1
  - fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.1-amd64
  - fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.1-s390x
  - fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.1-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
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

Field	Mandatory	Sub-Field	Default	Description
		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.
		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.

2.3.4 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.

- 2.3.5.1 Important Setting Items
- 2.3.5.2 Parameters that cannot be Set
- 2.3.5.3 Restricted Parameters
- 2.3.5.4 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 times (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 a FEPCluster deployment by setting the above parameters.

```
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
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
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	

Parameter	Overview of parameters	Reason
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	
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
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
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

Parameter	Overview of parameters	Reason
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)

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. For details on how to specify the passphrase, refer to "FEPCluster parameter" in the Reference.

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 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.

2.3.9.1 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.

2.3.10 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.

2.3.11 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.

Refer to the FUJITSU Enterprise Postgres Installation and Setup Guide for Server for the requirements of the 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.

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.

Even if the master encryption key is updated by key rotation, if you have encrypted backup data with the old encryption key, you must keep the old encryption key.

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.

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.

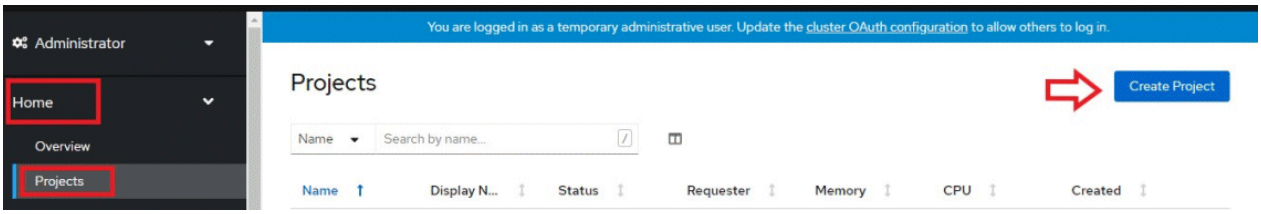
3.1 Using the OperatorHub

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

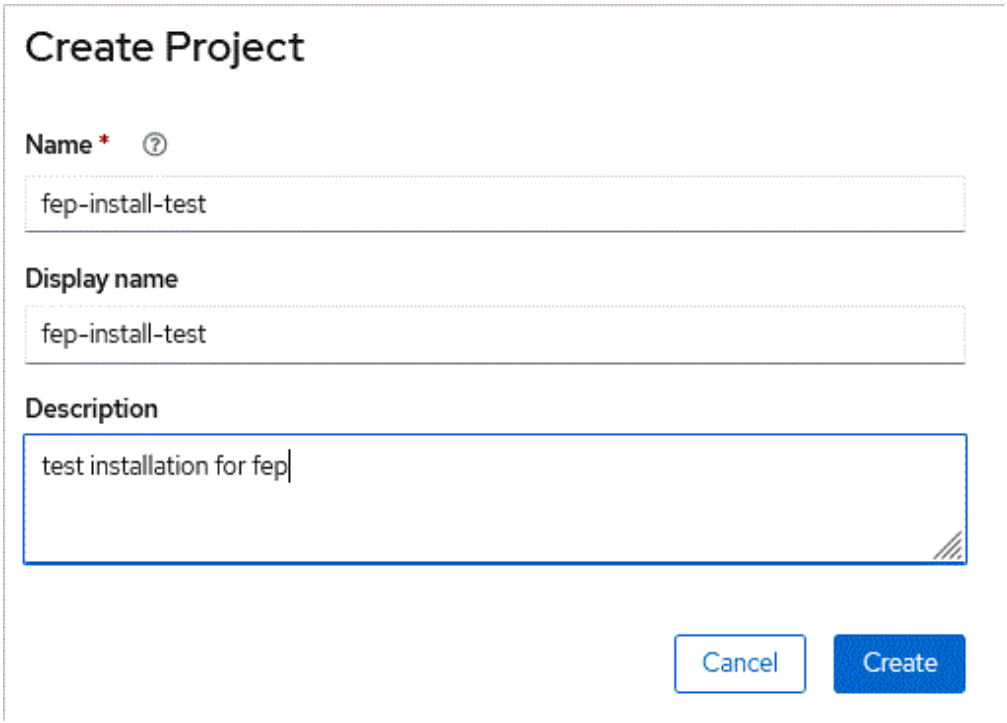
3.1.1 Pre-requisite

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.



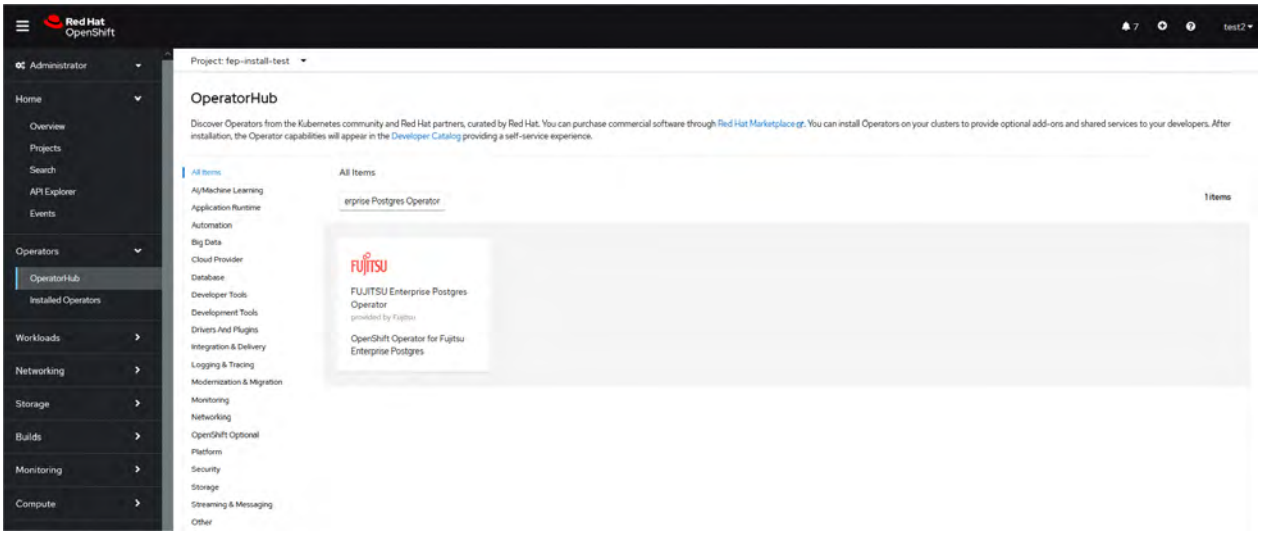
Note

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

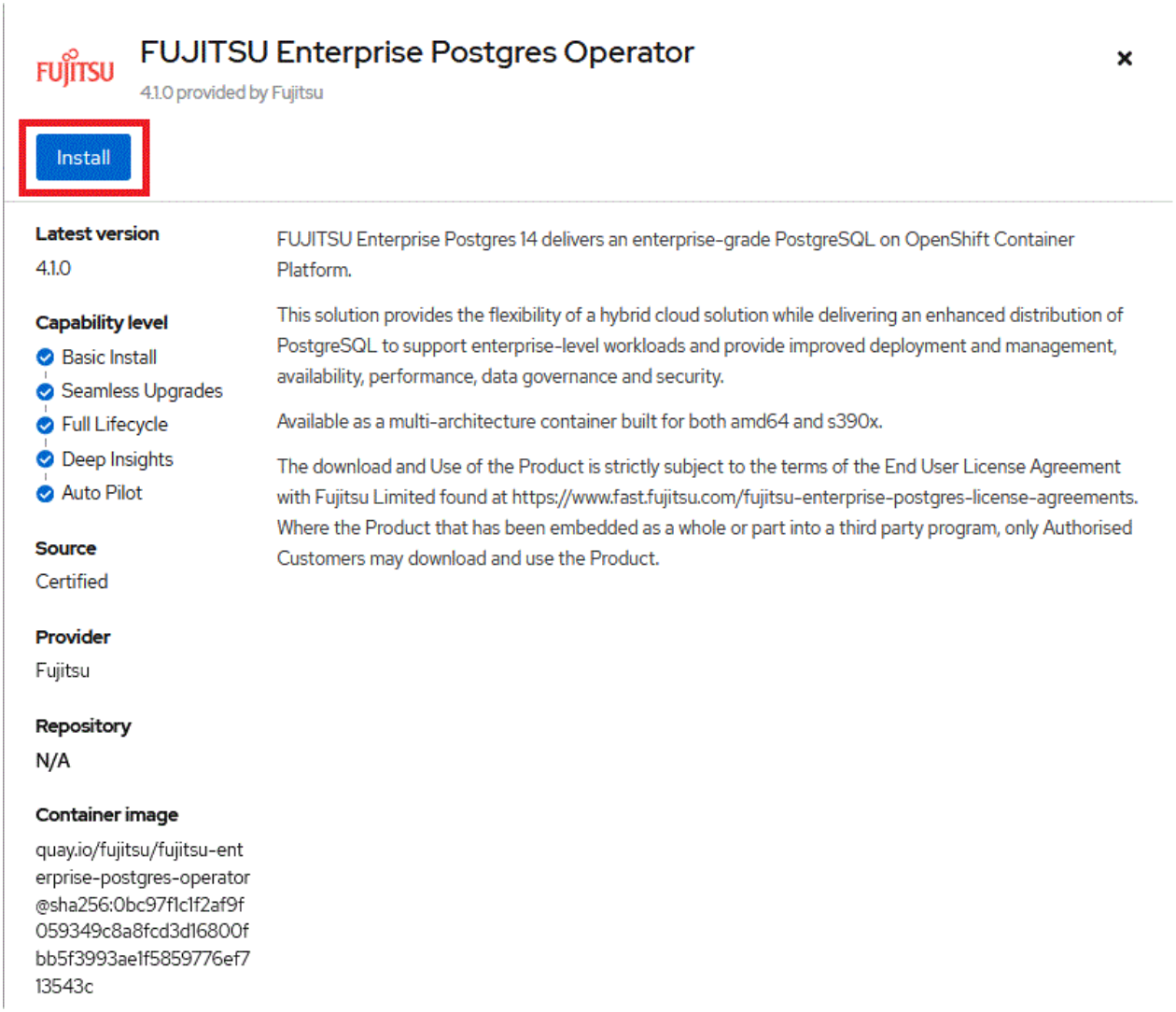
3.1.2 Deploying Operator

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

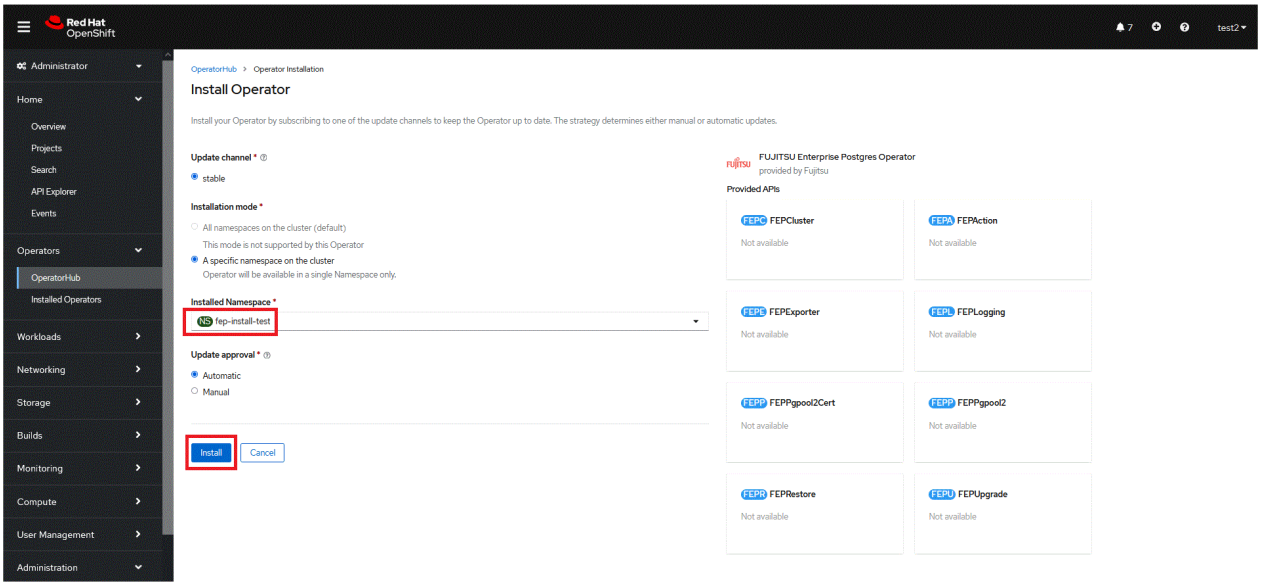
On OpenShift platform, logon with credentials that has privileges to install operator. Click on OperatorHub on menu item under Operators and type filter keyword "FUJITSU Enterprise Postgres Operator" to find FUJITSU Enterprise Postgres Operator.



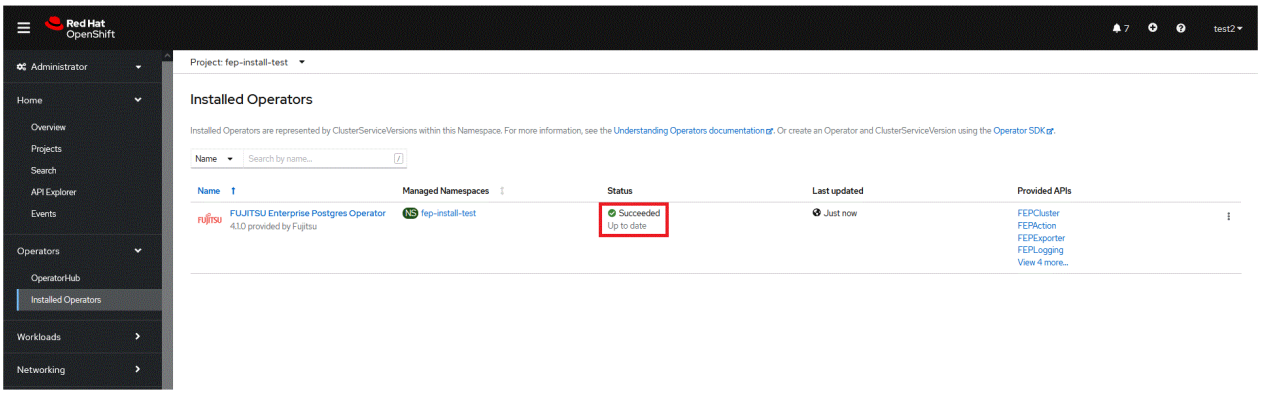
Click on FUJITSU Enterprise Postgres Operator to install operator. It will bring up details page with install button as below.



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".



3.2 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.

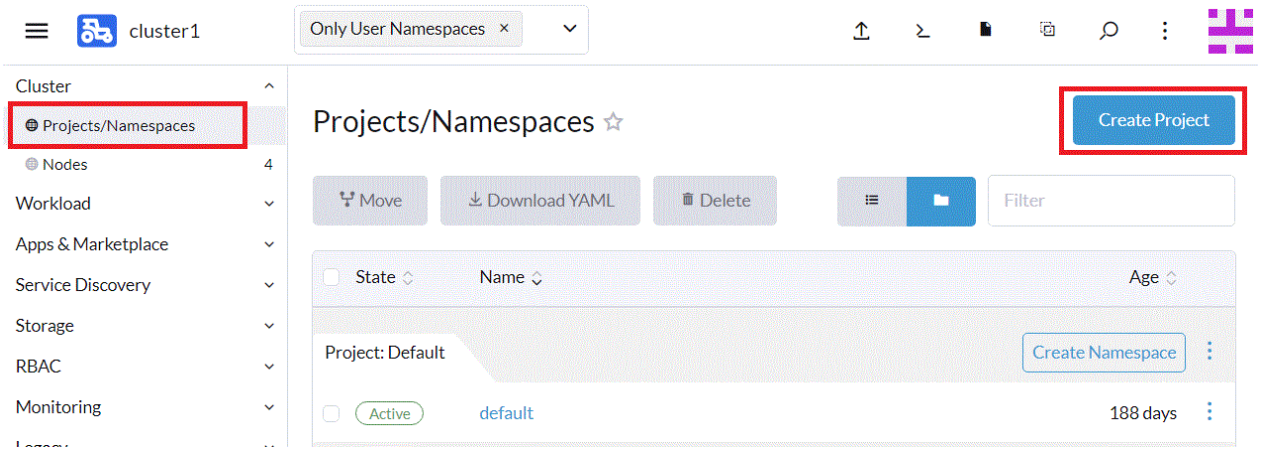
```
helm upgrade fep-operator-release fep-repo/fujitsu-enterprise-postgres-operator --namespace fep-operator
```

3.3 Using the Rancher UI

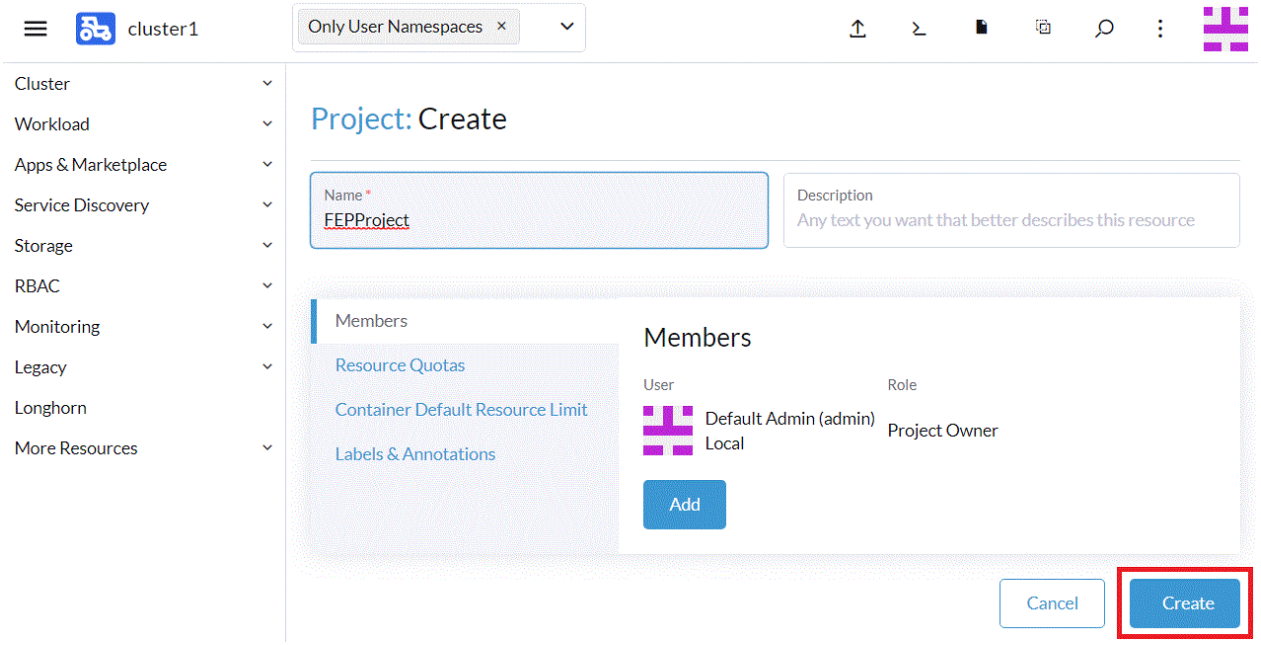
Describes how to install FEP operators into a new namespace on Rancher UI.

3.3.1 Pre-requisite

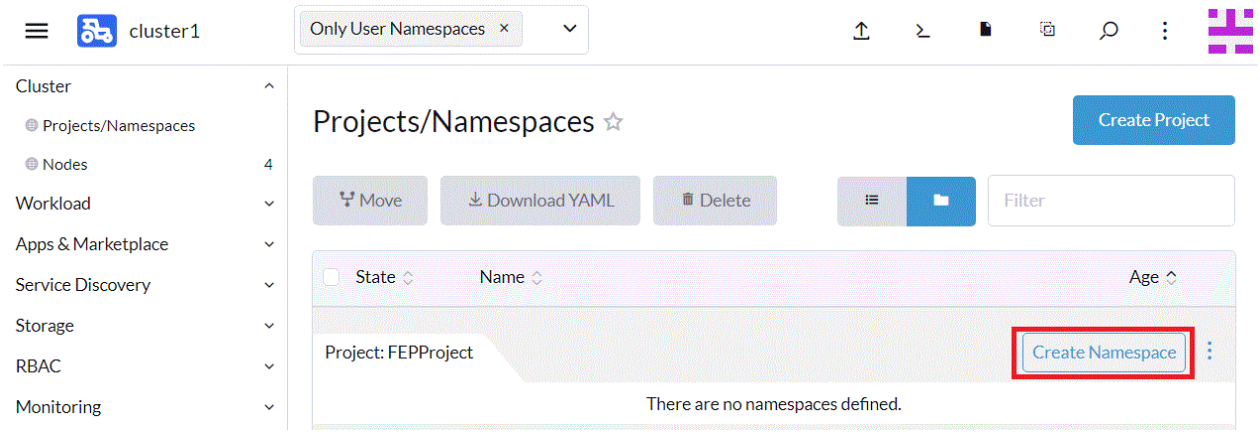
Create a project and its associated namespace on the Rancher UI. We recommend that you install FEP in a different namespace. In the Rancher UI, click [Projects/Namespaces], then click [Create Project] that appears.



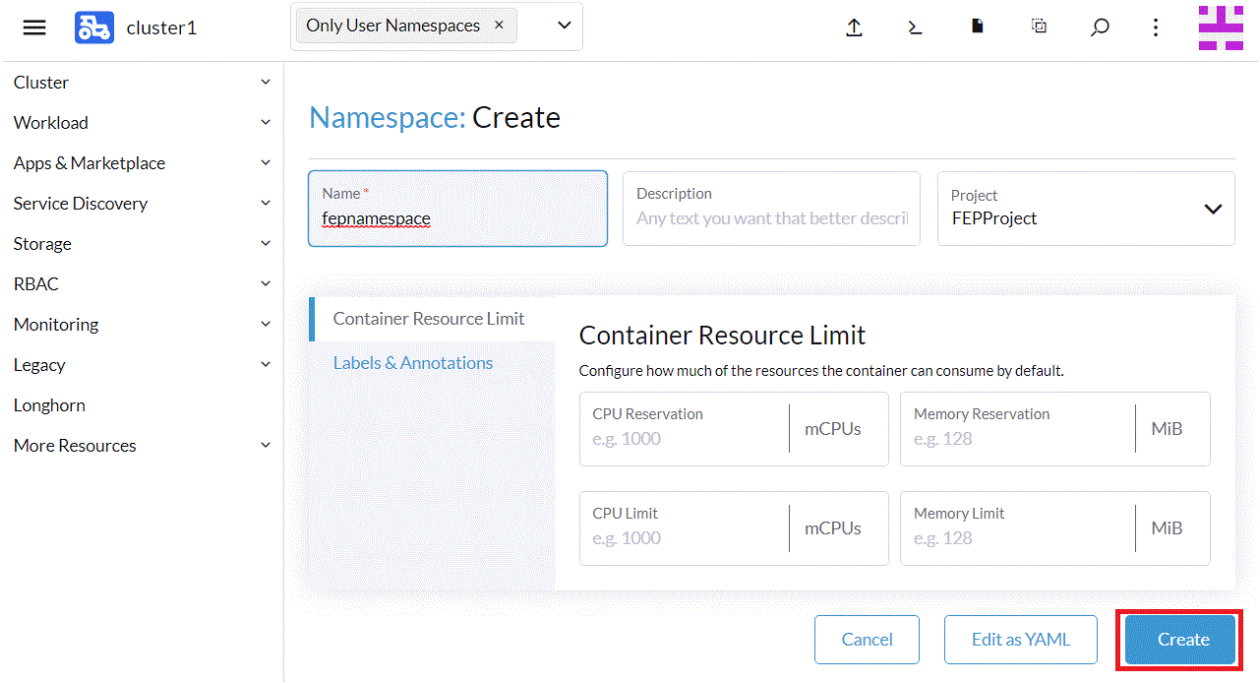
Specify a unique name for the project and click [Create].



Click [Create Namespace] displayed on the specified project.



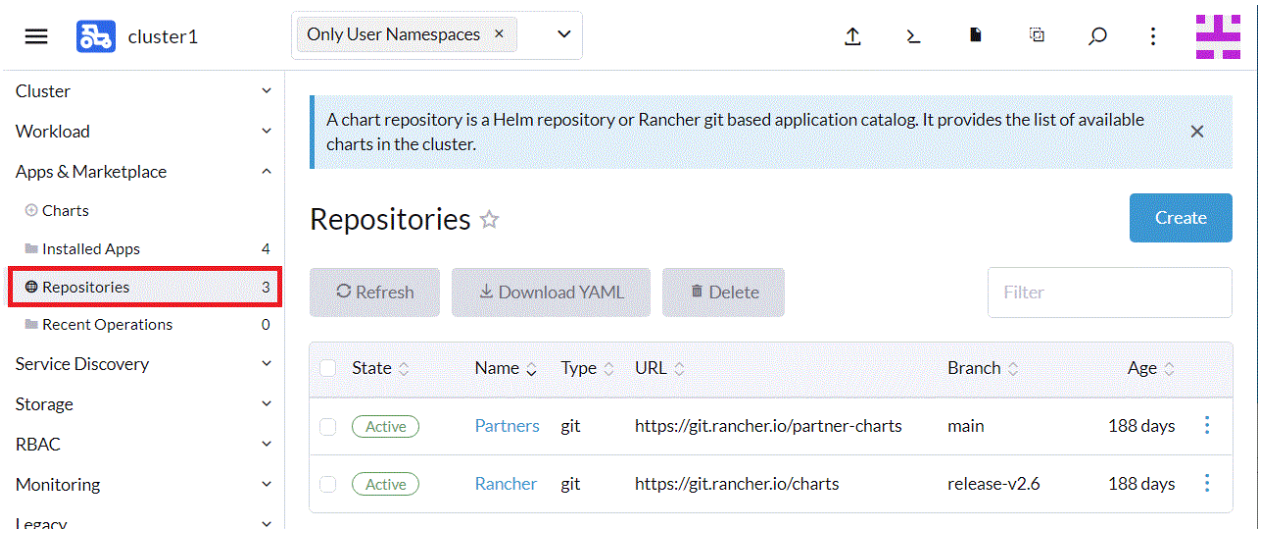
Specify a unique name in the namespace and click [Create].



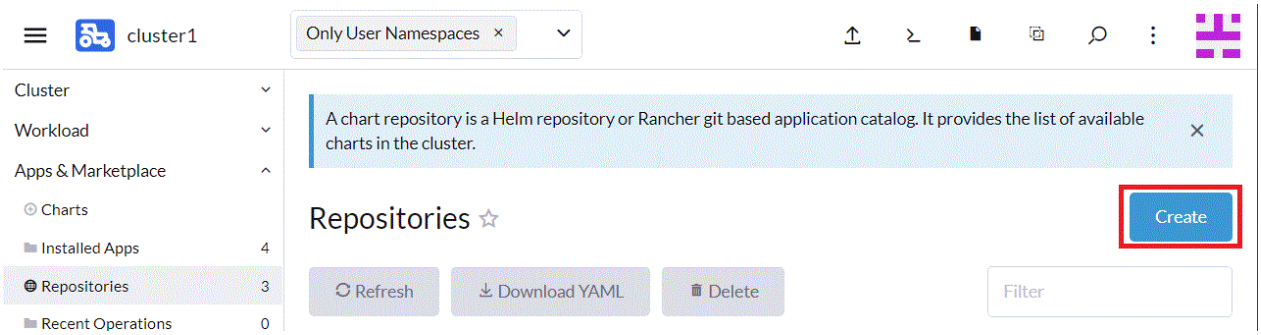
3.3.2 Register Helm Chart Repository

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.

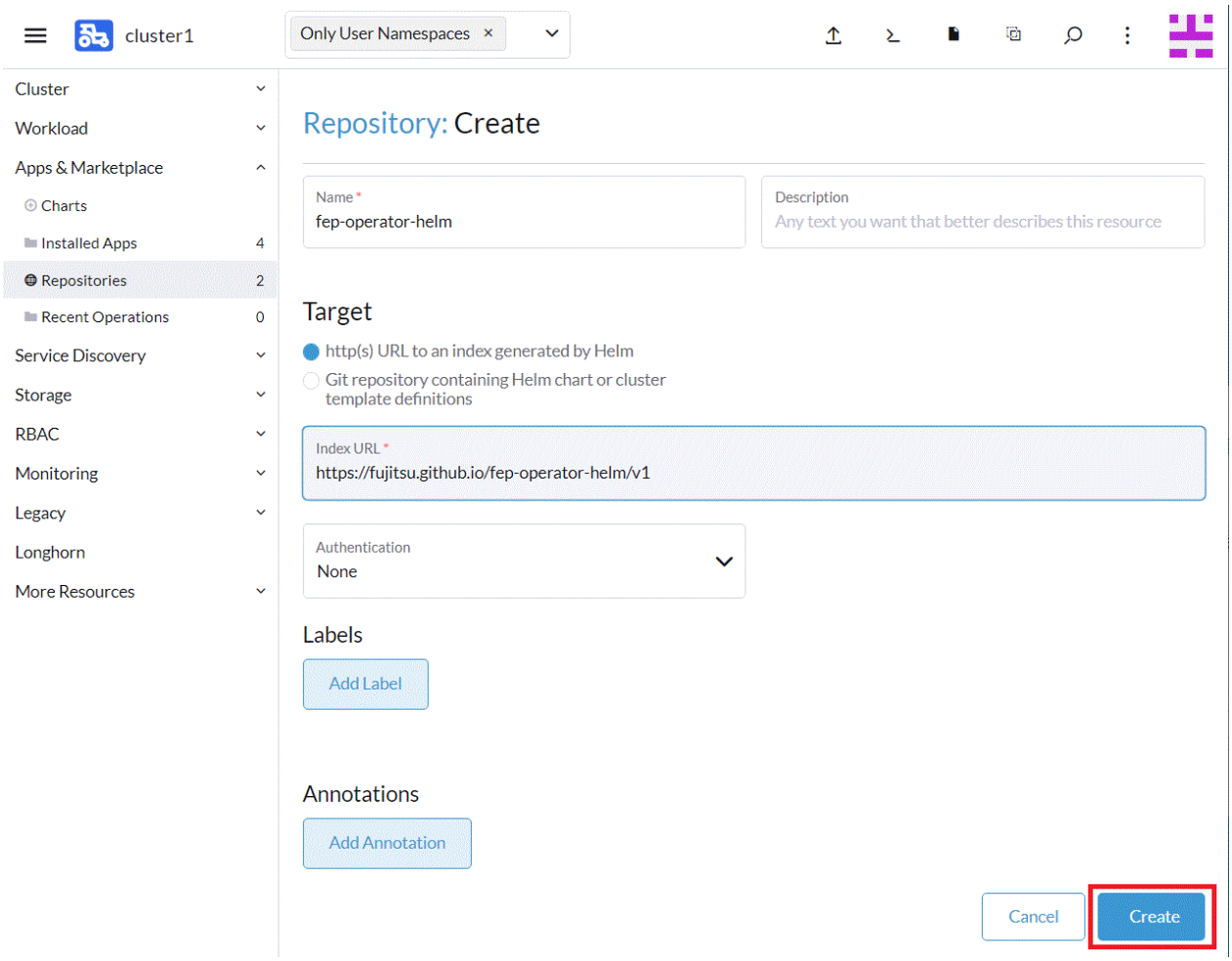


Click [Create] to create the Helm Chart repository.



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

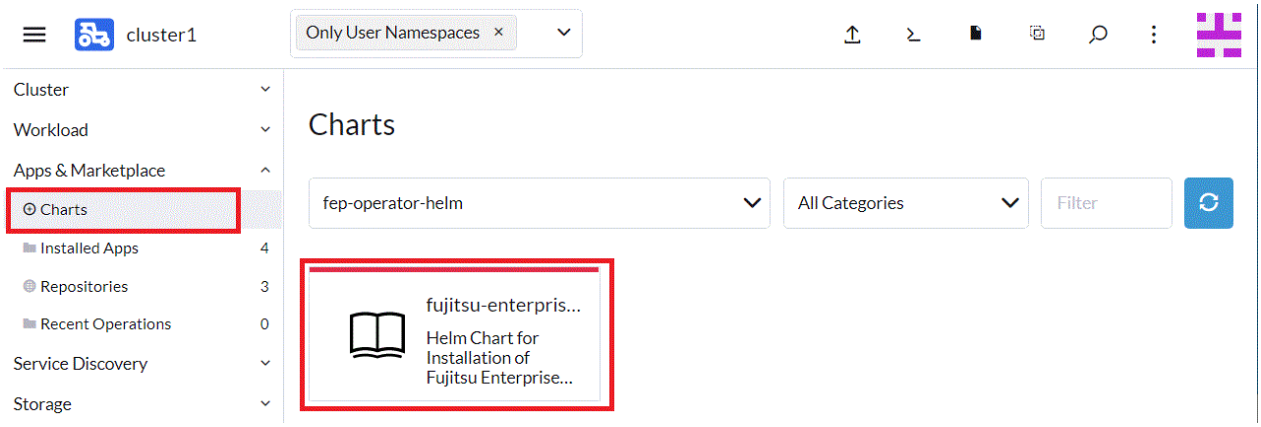
```
https://fujitsu.github.io/fep-operator-helm/v1
```



3.3.3 Deploying Operator

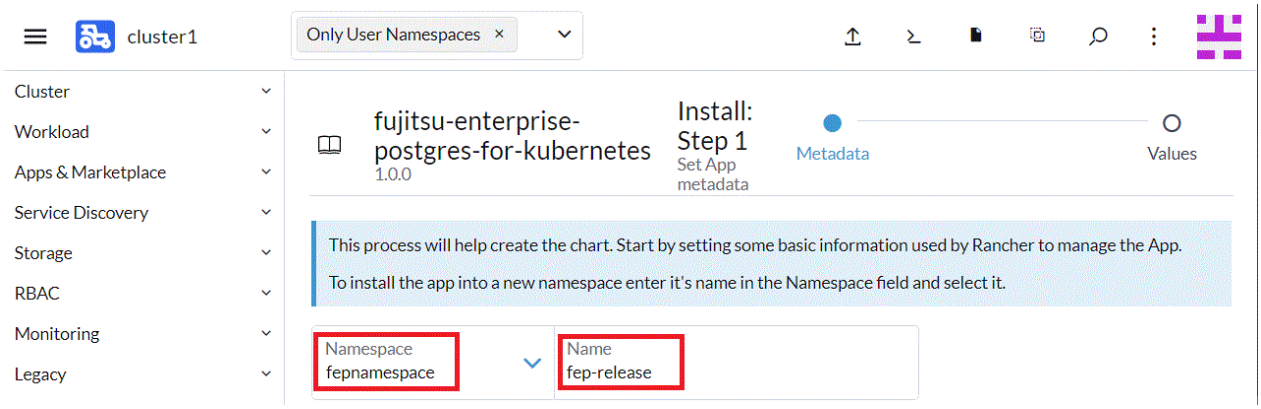
On the Rancher UI, apply the operator function Helm Chart to the project / namespace created in "3.3.1 Pre-requisite" and install the operator.

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

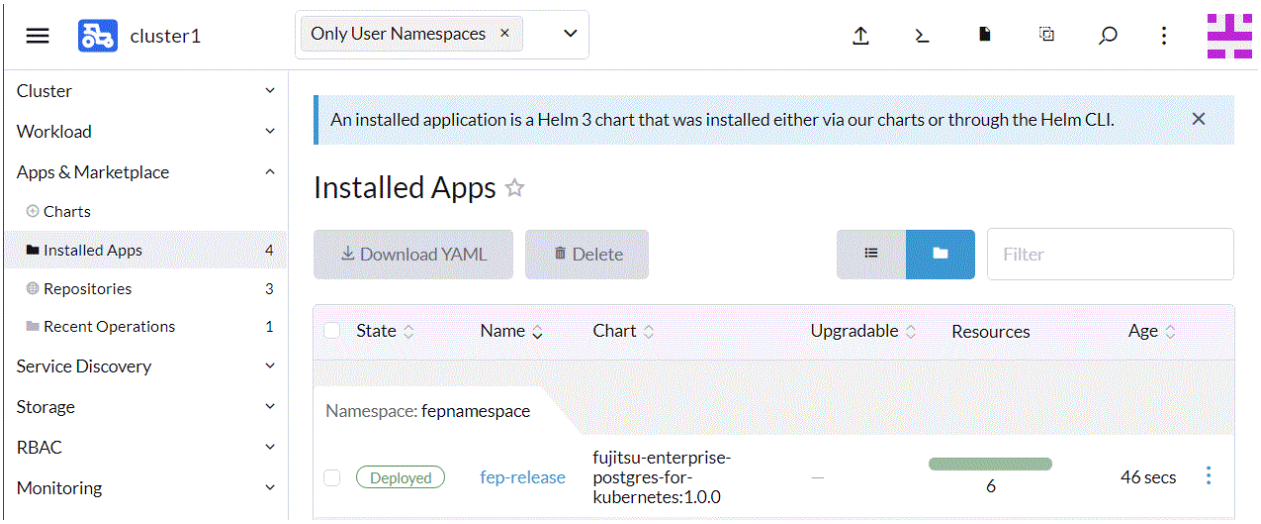


Click Install on the screen that appears.

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.



3.4 Implement Collaborative Monitoring Tools

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/4.9/monitoring/monitoring-overview.html#understanding-the-monitoring-stack_monitoring-overview)

For Grafana, install and use the Grafana Operator provided by OperatorHub for x86. 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.

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.

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.

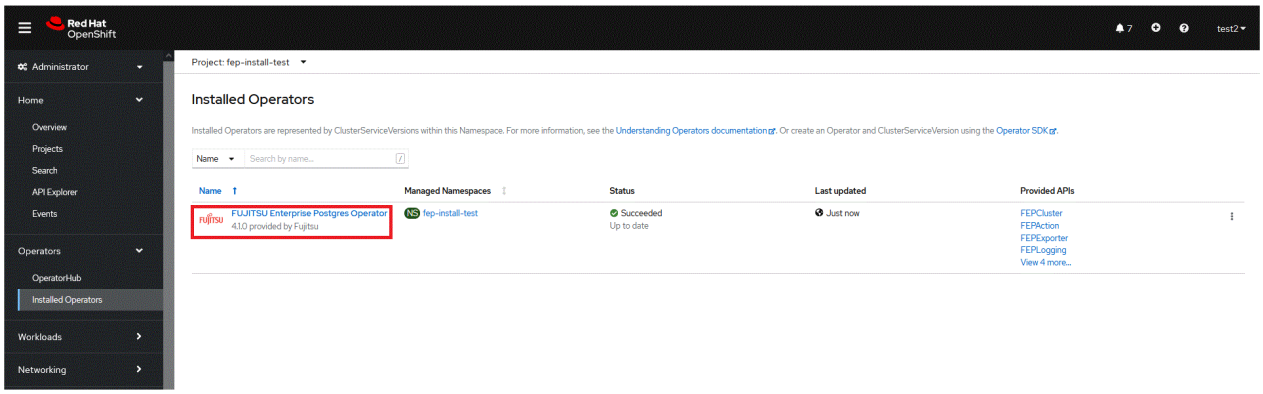
4.1 Deploying FEPCluster using Operator

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

Note

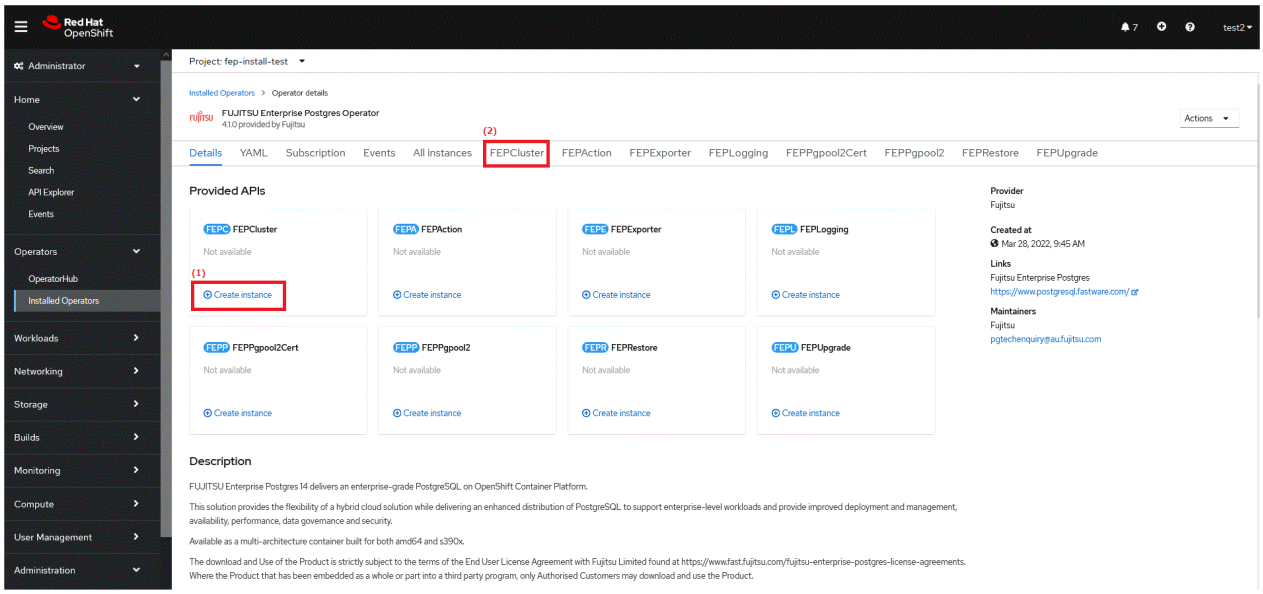
If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

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.

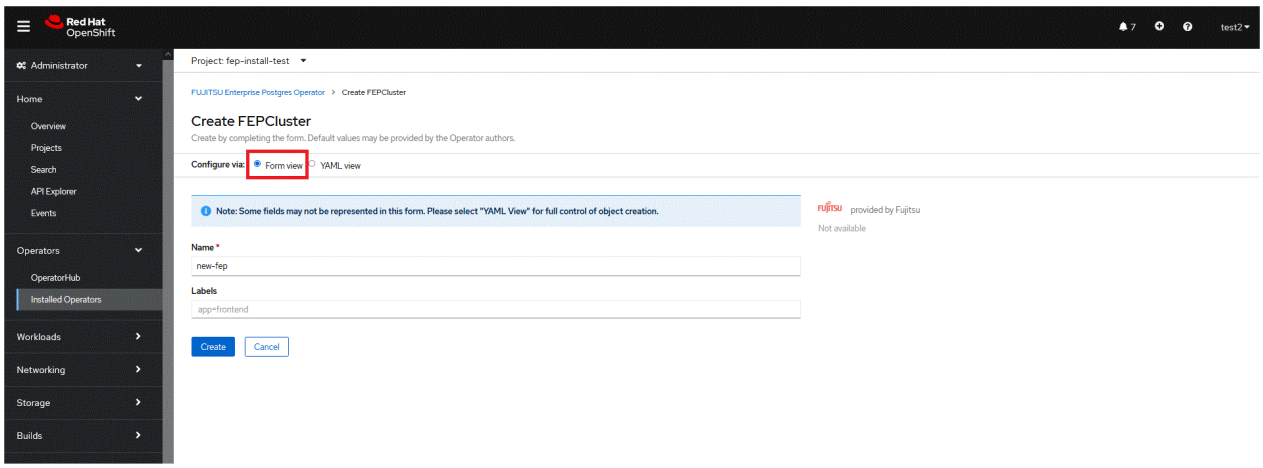


2. It will display a page with all CRs this operator supports. FEPCluster is the main CR and all others are child CR. We would create the main CR and all other CRs will be created automatically by Operator.
   To create Cluster CR, either
   (1) Click on "**Create Instance**" under FEPCluster.

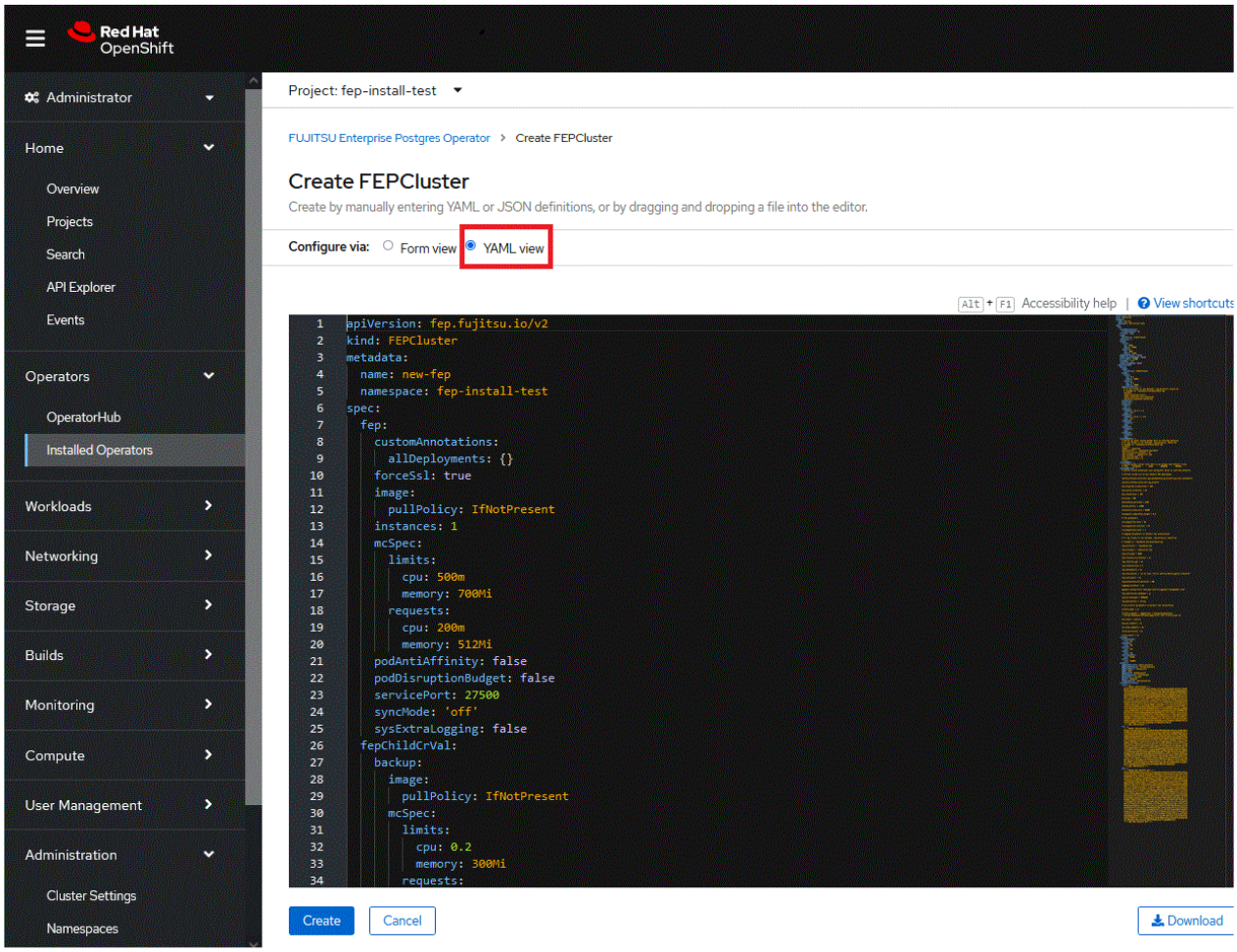
OR
(2) Click on "**FEPCluster**" on top and then click on "**Create FEPCluster**" on the next page.



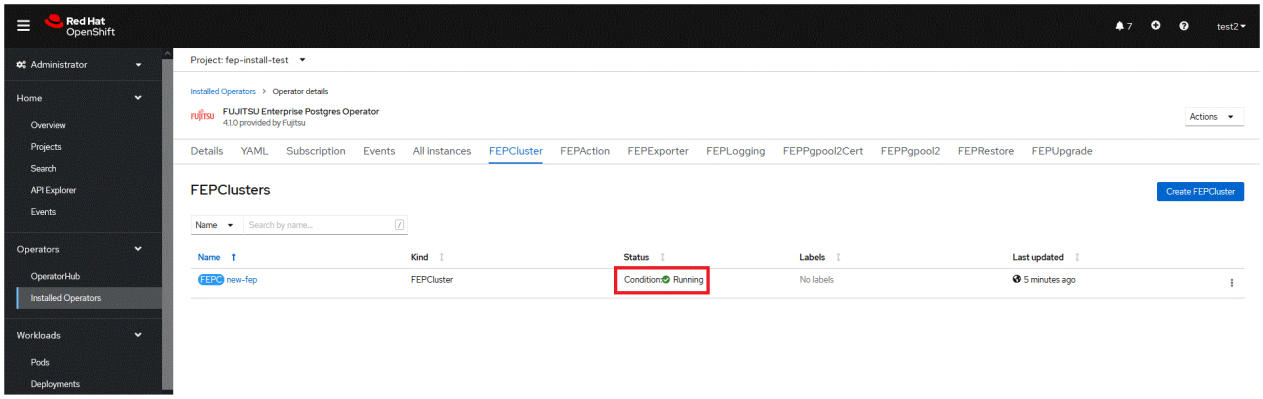
3. This will bring to "Create FEPCluster" page. Here you have two options to configure. The first one is Form View. At the moment, in Form View , one can change only the name of cluster being deployed. The default name is "new-fep".
This name must be unique within a namespace.



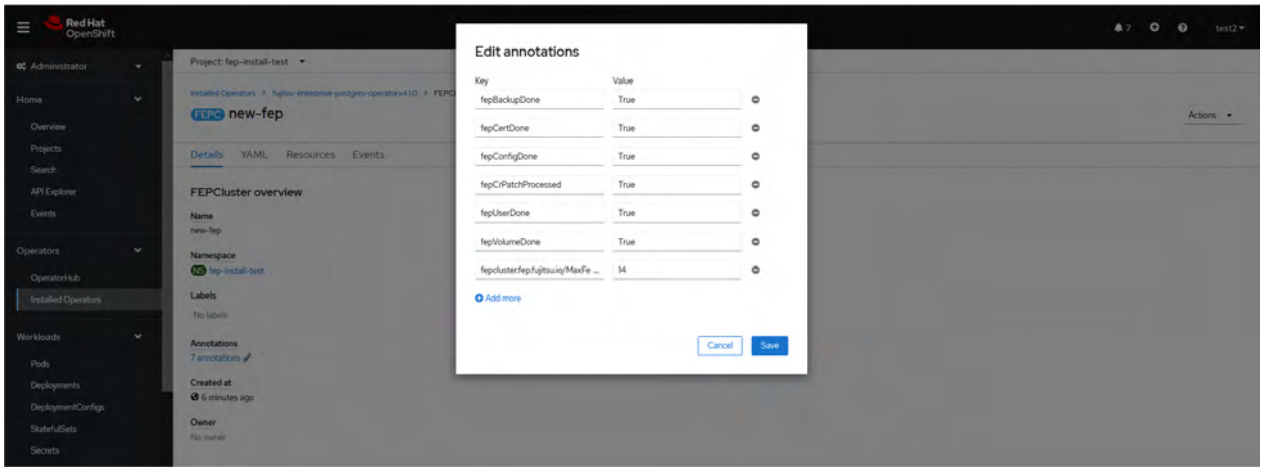
4. In YAML View, starting value of CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.



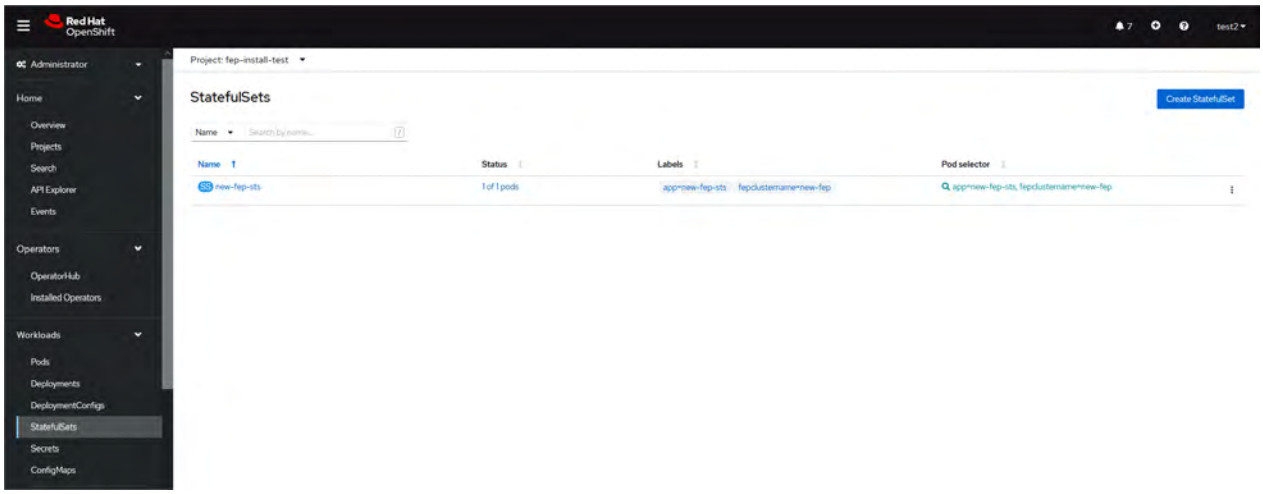
5. When "Create" is clicked on either of the two pages above, the operator creates FEPCluster CR, and there after one by one FEPBackup, FEPConfig, FEPVolume, FEPUser, and FEPCert child CRs are created automatically.
The starting values for child CRs are taken from the "fepChildCrVal" section of the FEPCluster 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.



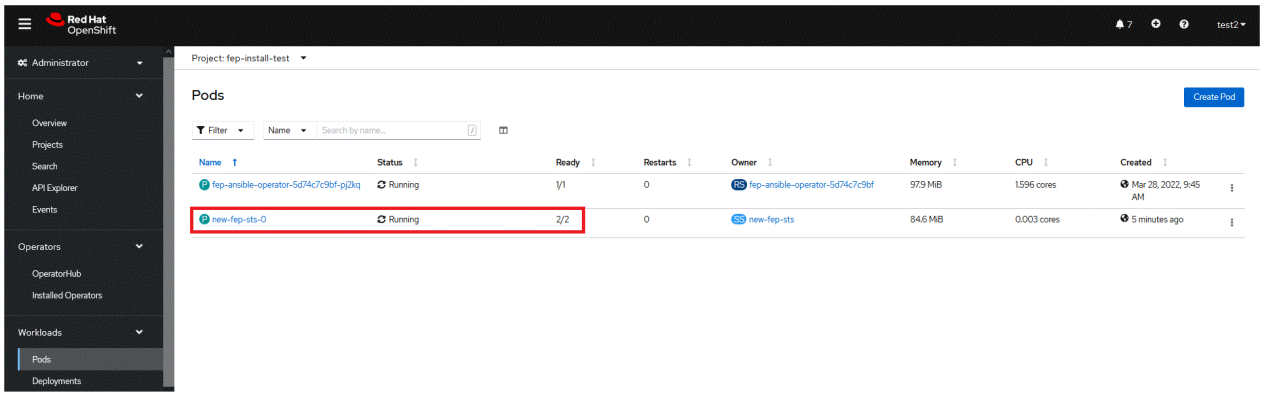
6. In FEPCluster CR, annotations are added to indicate that child CRs are created successfully and has initialised properly. It may take some time to complete.



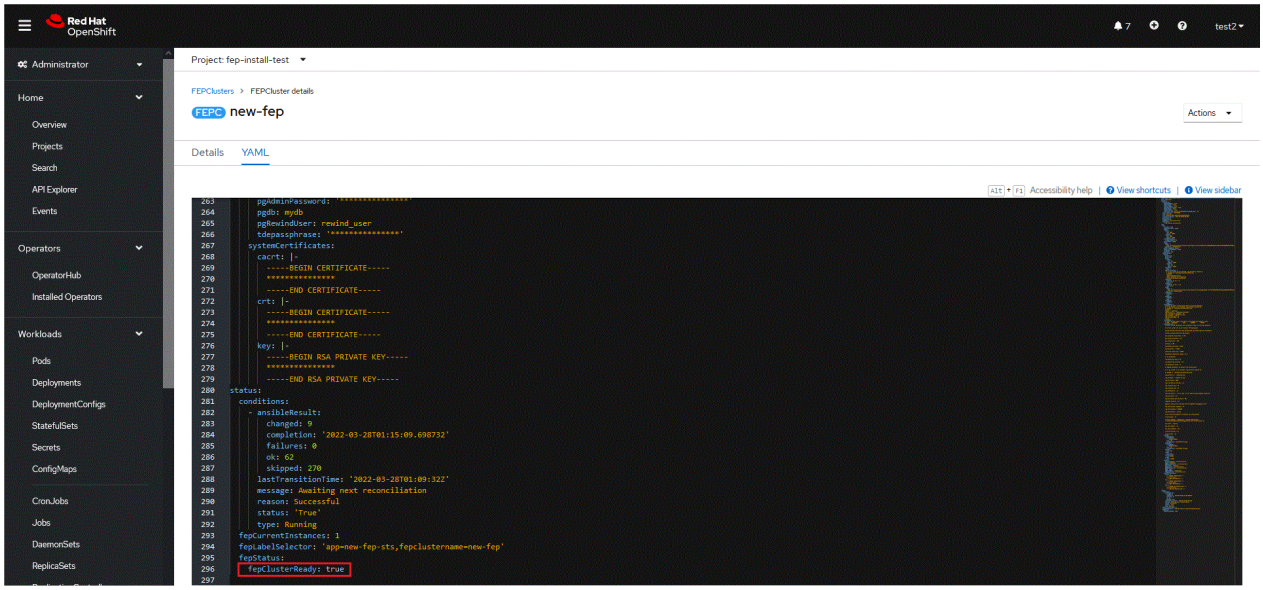
7. Once child CRs are marked done in annotations, operator creates StatefulSet for the cluster.



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.

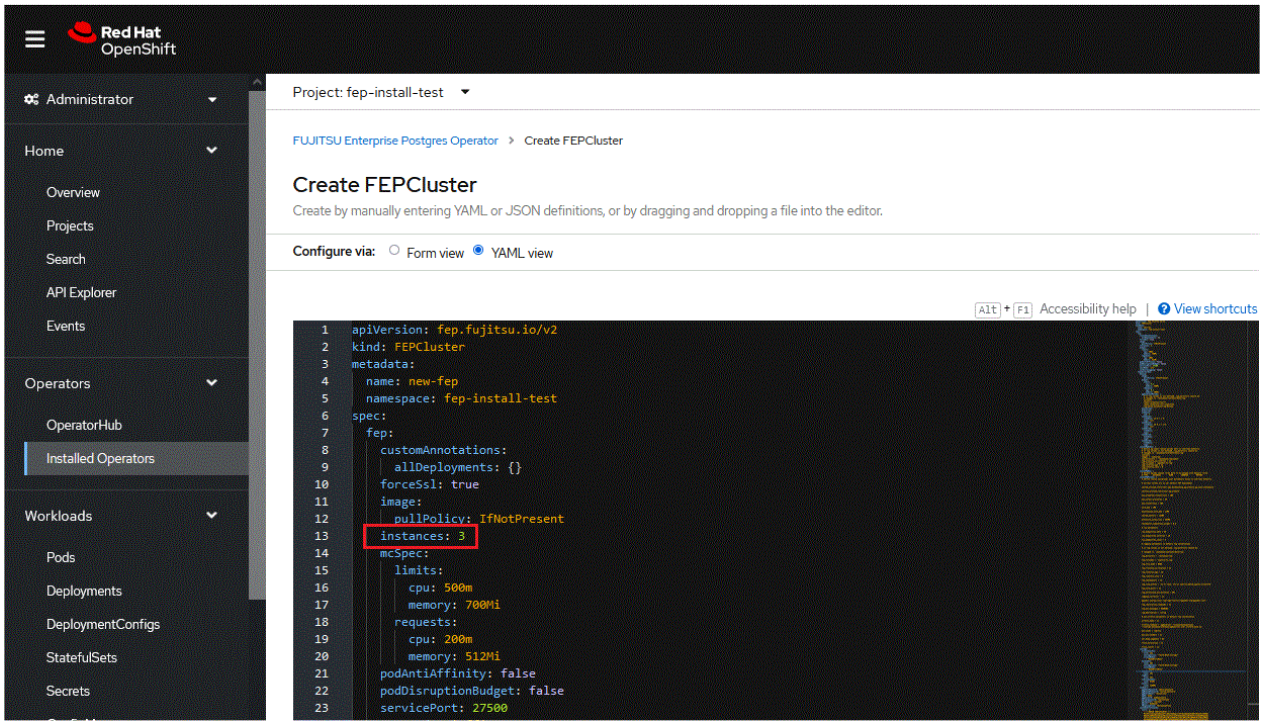
If you do not have shared storage, you can remove the backup section and the backup and archive volume sections to disable the backup feature and deploy the FEP cluster.


Note

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

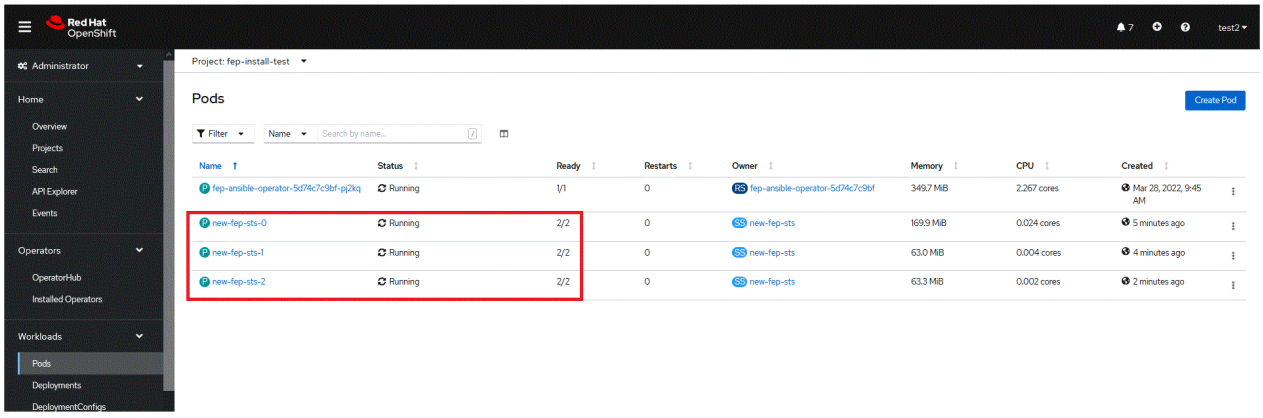
1. It is the same as the procedure from step 1 to step 3 in "4.1 Deploying FEPCluster using Operator".

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
```

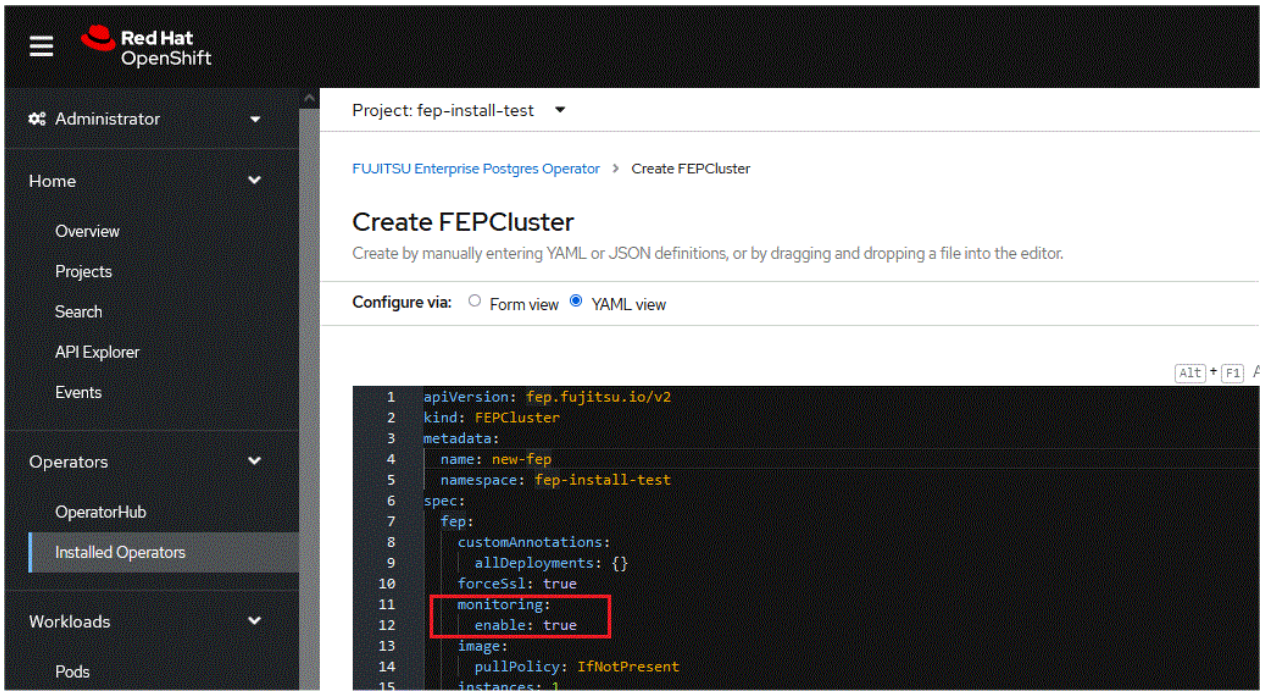
4.3 Deploying FEPExporter

To deploy a FEPExporter, follow these steps:

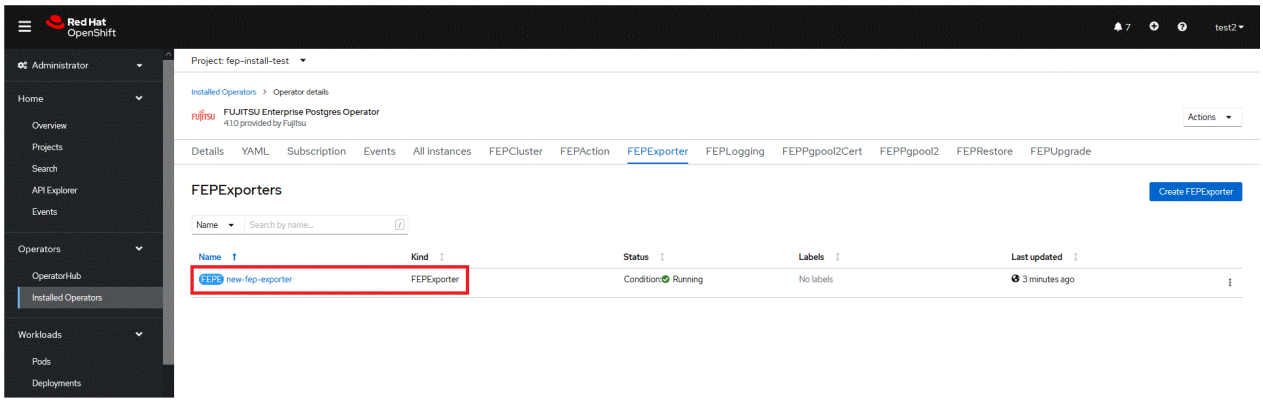
Note

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.

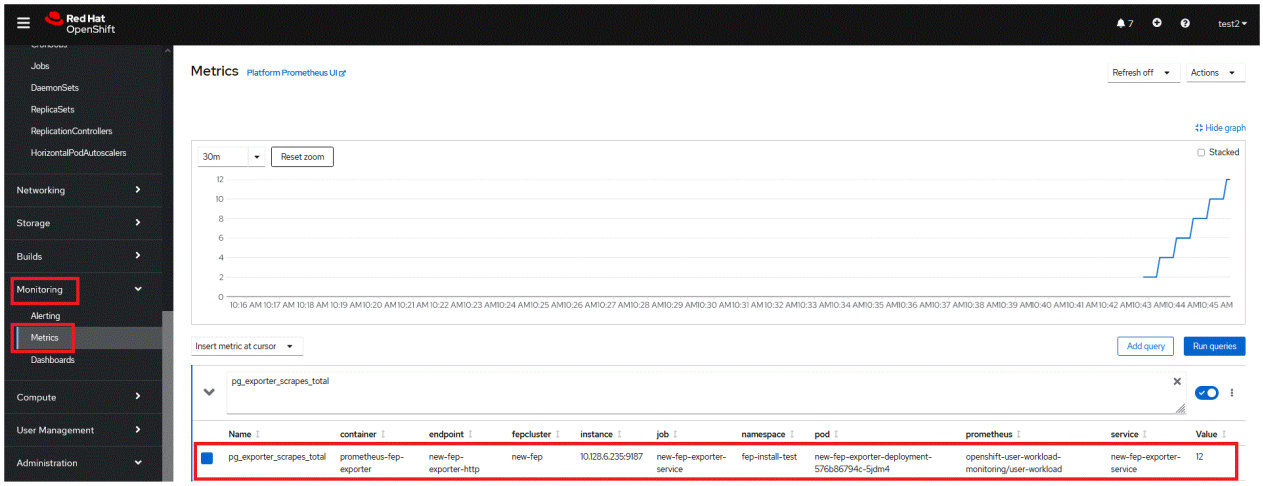


2. FEPExporter will be created automatically under the name <cluster-name>-fepexporter. And it will list show all the database with statistics of specified FEPcluster.



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/4.6/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

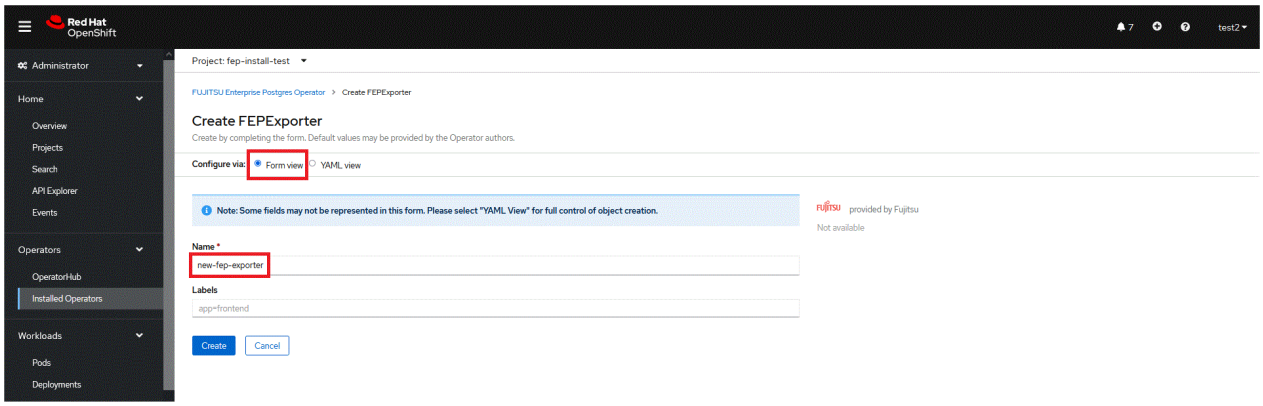
If you are deploying on a Kubernetes cluster, Refer to "Custom Resource Parameters" in the Reference to create and apply a yaml file.

1. To create FEPExporter CR, either
   (1) Click on "**Create Instance**" under FEPExporter.

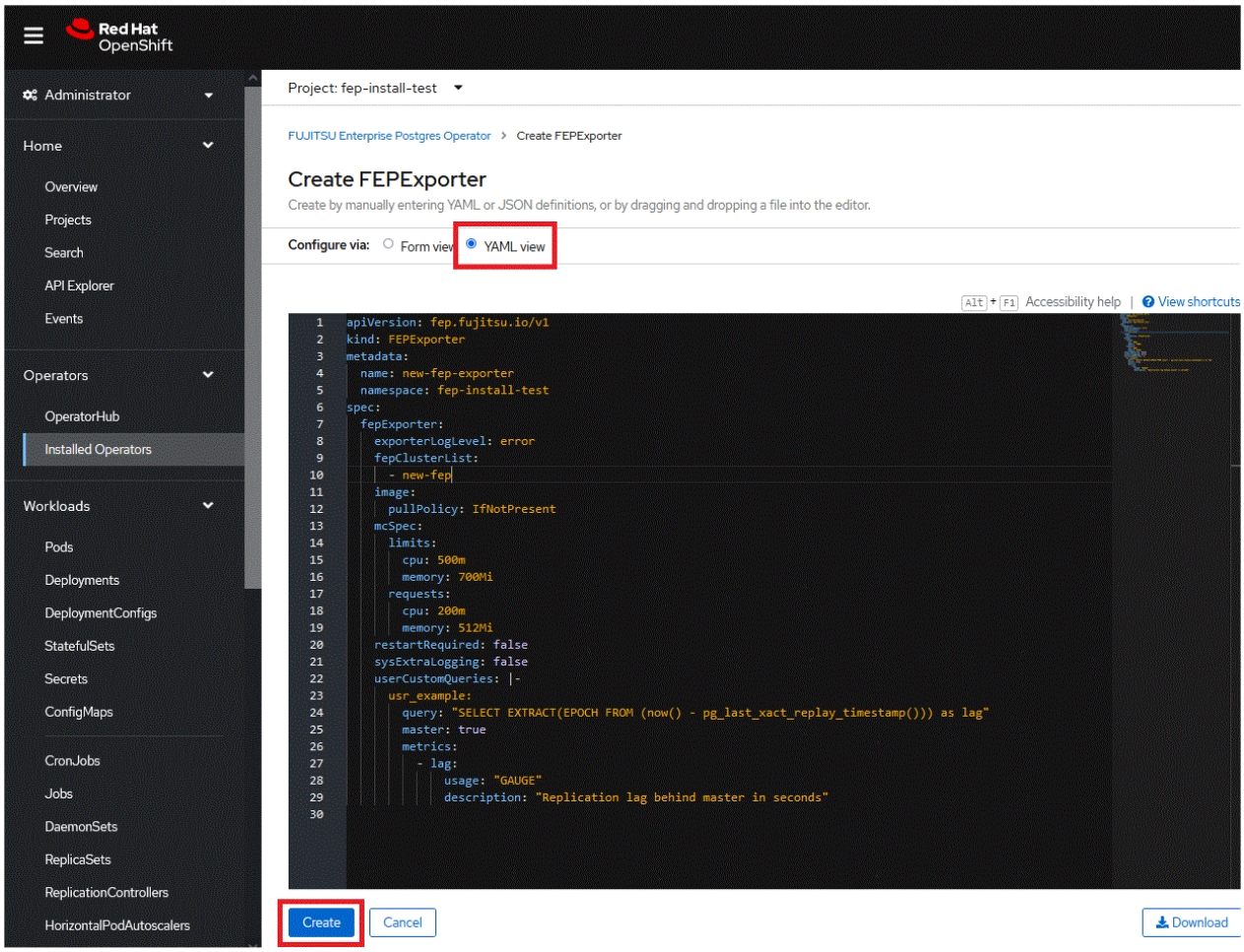
OR
(2) Click on "**FEPExporter**" on top and then click on "**Create FEPExporter**" on the next page.

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.

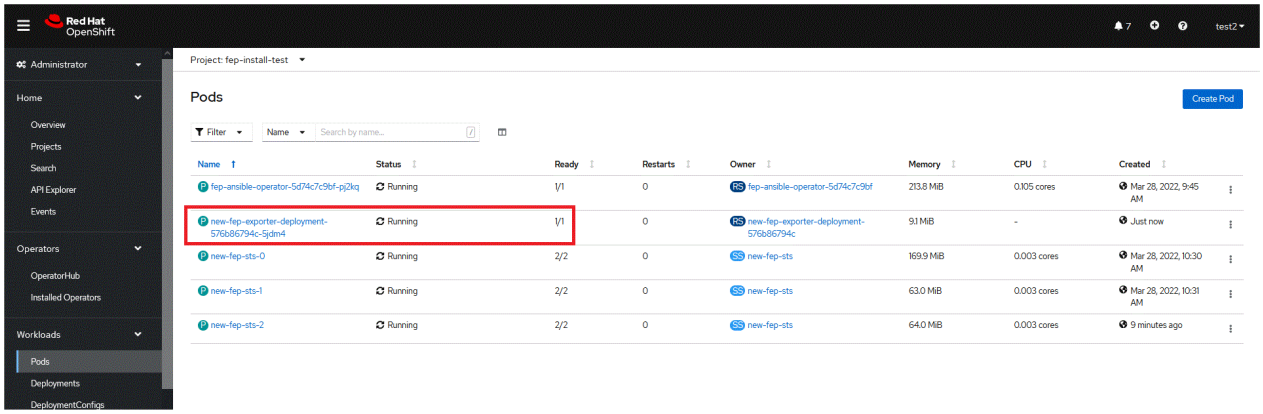
3. FEPExporter scrapes metrics for FEPCluster within same namespace.



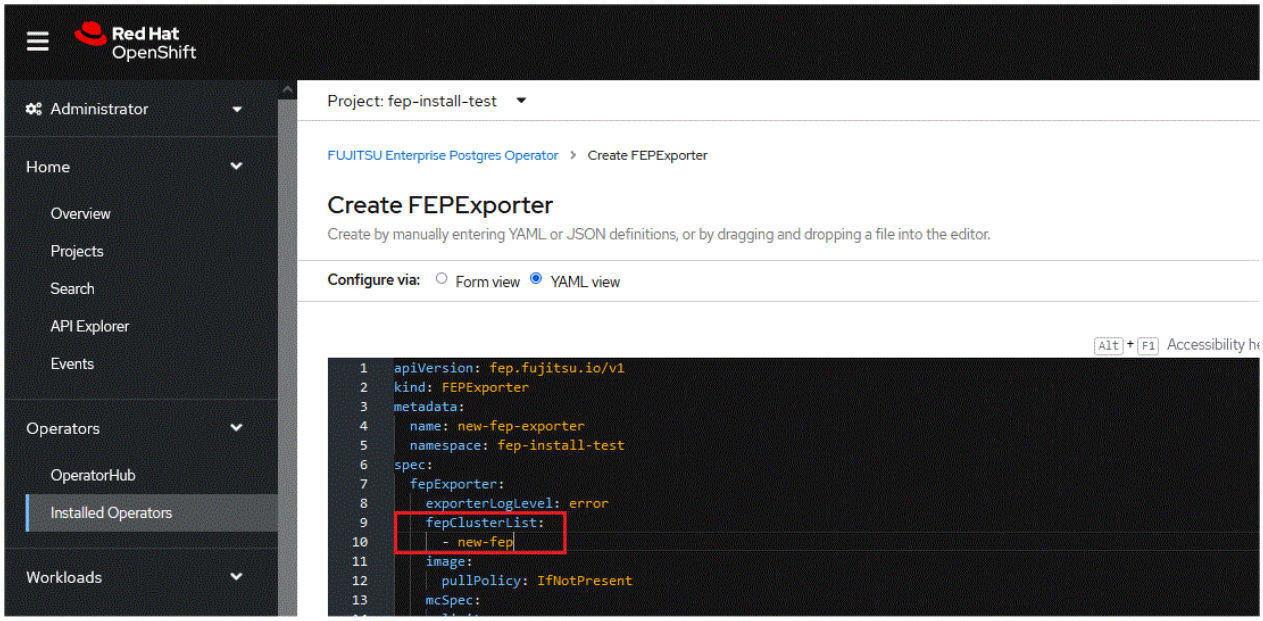
4. In YAML View, starting value of FEPExporter CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.



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



6. Targeting the name of FEPcluster in FEPExporter cluster list. 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 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)

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.

4.5.1 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"

Note

- 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"

4.5.2 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

- 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
```

4.5.3 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-14-server:ubi8-14-0.0'
    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-14-server:ubi8-14-0.0'
      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-----
```

4.5.4 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.

Key	Value	Details
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 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

Key	Value	Details
		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.

4.6 Replication Slots

4.6.1 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

```
79    customAnnotations:
80      allDeployments: {}
81    servicePort: 27500
82    postgres:
83      tls:
84        caName: cacert
85        certificateName: my-fep-cert
86    image:
```

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=veri
fy-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

4.7 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.

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 "4.5.2 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.

4.7.1 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.

4.7.1.1 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

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
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. 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

4.7.1.1.1 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:ubi8-14-0.0'
      pullPolicy: IfNotPresent
```

4.7.1.1.2 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
```

4.7.1.1.3 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
```

4.7.1.1.4 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
```

4.7.1.1.5 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
```

4.7.1.1.6 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: kubernetes.io/basic-auth
```

4.7.1.1.7 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
```

4.7.1.1.8 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
```

4.7.2 FEPCluster Configuration

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.

4.7.2.1 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	Required	The ‘fluentdName’ is the name of the FEPLogging CR where logs will be forwarded	Yes
remoteLogging.tls.secretName	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

4.7.2.1.1 Define remoteLogging enable and fluentdName

The enable flag is used to describe that FEPCluster will enable log monitoring 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 log monitoring 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
       ...
```

4.7.2.1.2 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.

4.7.2.1.3 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)

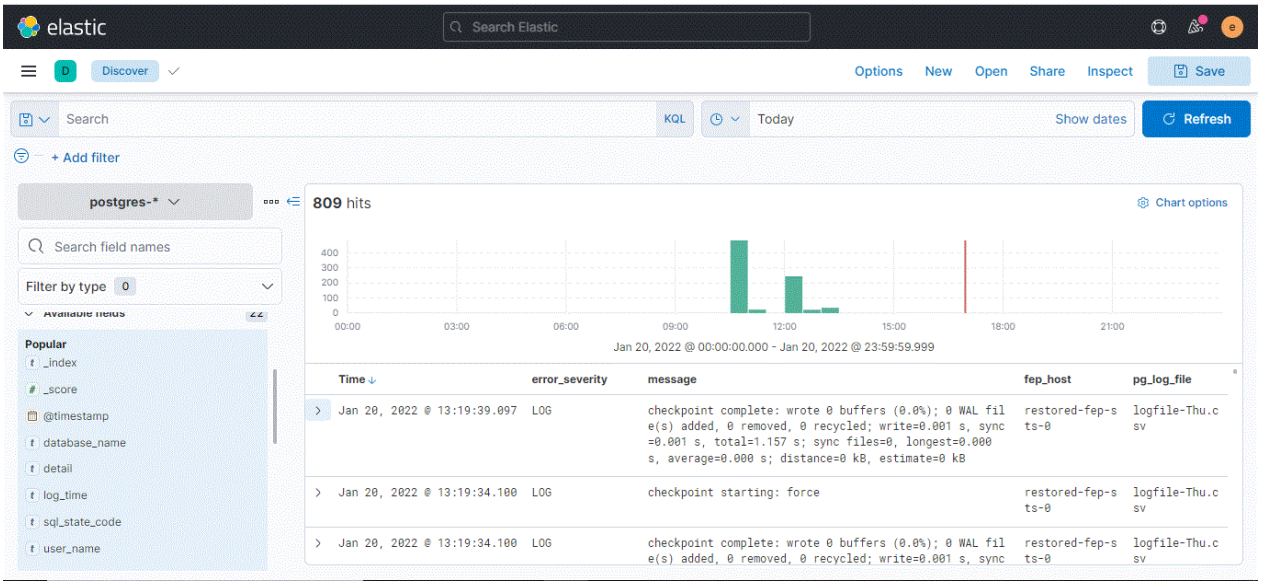
```
spec:
  fep:
    remoteLogging:
      image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.3'
      pullPolicy: IfNotPresent
```

4.7.3 FEPLogging Operations

4.7.3.1 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.

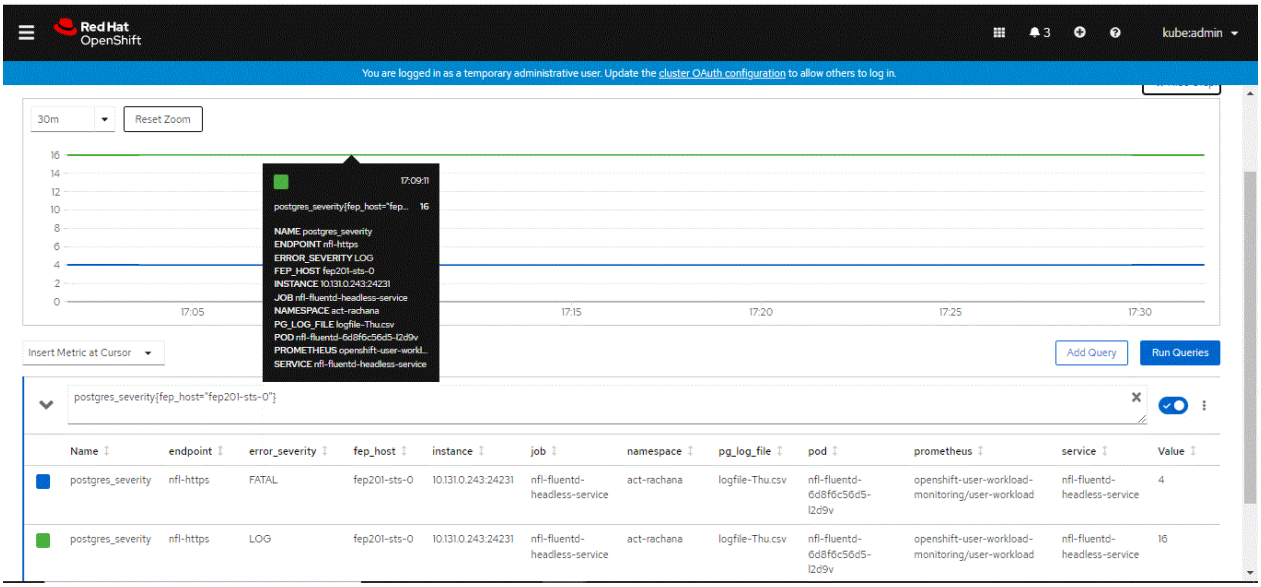


4.7.3.2 Log severity based Alarms/Metrics

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.



4.7.3.3 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'
```

4.7.4 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.

4.8 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.

4.8.1 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

4.8.2 Define pgBdager 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 pgBdager report will be created and uploaded.

Example)

```
pgBadger:
  schedules:
    cleanup: '10 * * * *'
    create: '50 * * * *'
```

4.8.3 Define pgBdager 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
```

4.8.4 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
--cacert	webca.pem included in the configmap referred in customCertificateName
--form “uploadfile=@/path/to/report”	Endpoint fileUploadParameter
--header "Authorization: Basic passxxx”	Endpoint authentication configmap

curl command parameter	User configuration
--insecure	When endpoint.insecure is set to true

4.8.5 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.

4.9 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.

4.9.1 Certificate Registration

Save the certificate used for TLS communication between key management systems 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 your key management system.

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 key management system
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
```

4.9.2 Configuring FEPCluster Custom Resources

To enable TDE using a key management system, you need to set “spec.fepChildCrVal.customPgParams” and “spec.fepChildCrVal.sysTde”.

4.9.2.1 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.

4.9.2.2 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, TDE with a passphrase is implemented). 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

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.

Example)

```
    sysTde:
      tdeType: tdek
      tdek:
        targetKmsName: kms_conninfo1
        kmsDefinition:
          - name: kms_conninfo1
            type: kmip
...
```

Refer to the Reference for details of each parameter.

For the parameter under cert of fepChildCrVal.sysTde.tdek.kmsDefinition, specify the Secret or ConfigMap name created in "4.9.1 Certificate Registration".

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
```

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-14-server:ubi8-14-1.1'	Minor upgrade of FEP image to ubi8-14-1.1.
spec.fepChildCrVal.backup.image.image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-14-backup:ubi8-14-1.1'	Minor upgrade of Backup image to ubi8-14-1.1.

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>

Specify updated parameters in the format described in "2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator". Only following parameters would change for Operations that can be performed here.

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.

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>

Specify updated parameters in the format described in "2.3.5 Scheduling Backup from Operator". Only following parameters would change for Operations that can be performed here.

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.

Note

- 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.

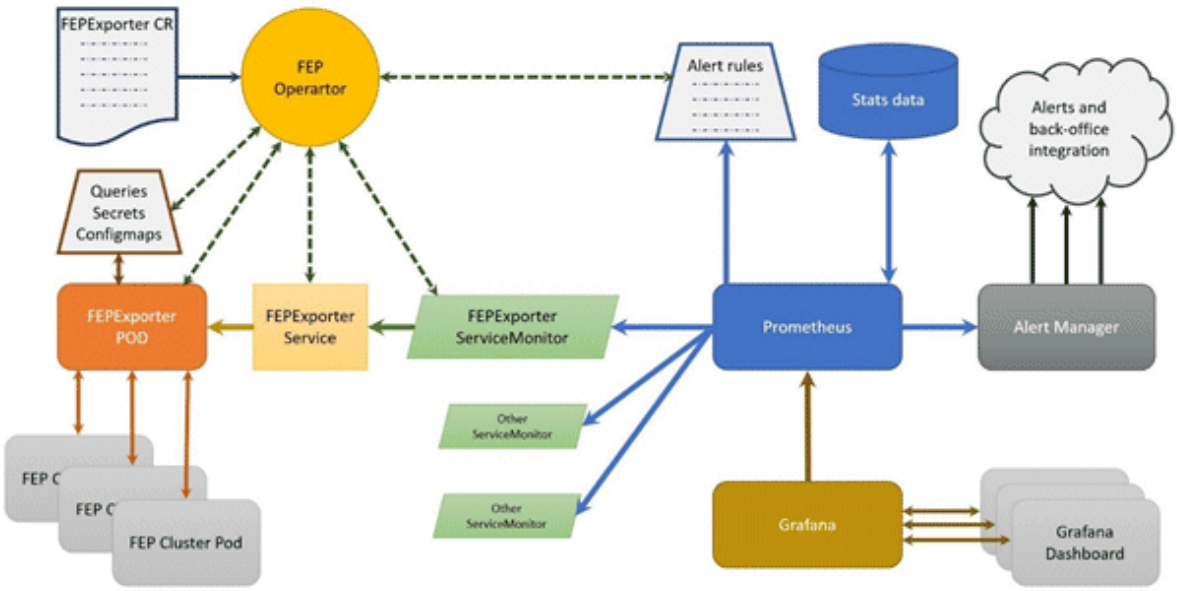
The Prometheus acts as storage and a polling consumer for the time-series data of FEP container. Grafana queries Prometheus to displaying informative and very pretty graphs.

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.

There are 2 levels 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) - 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.

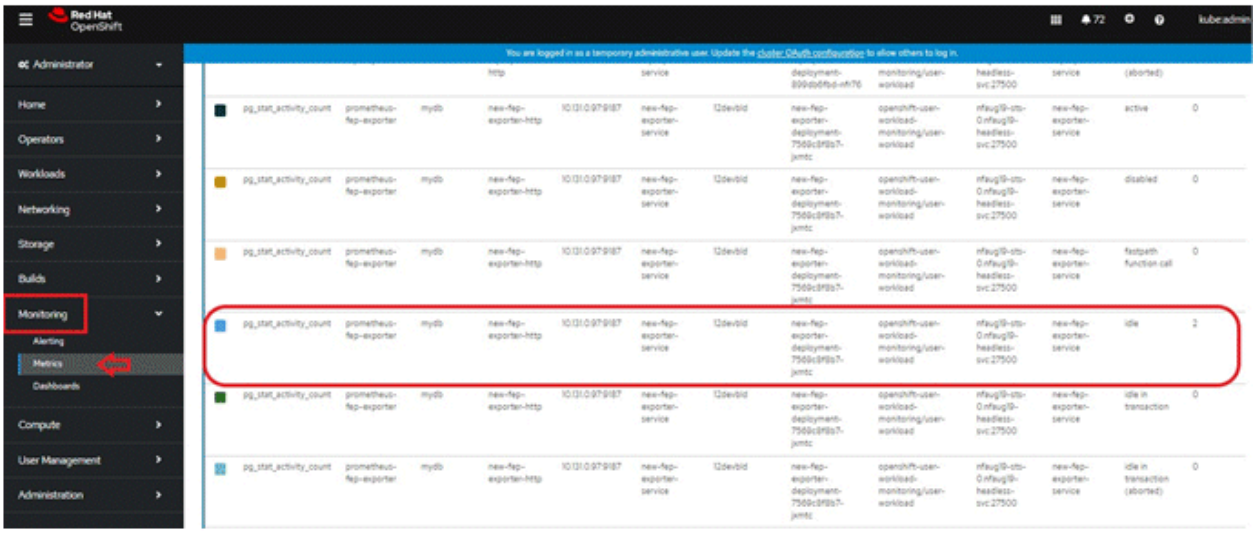
Metrics name	Details
pg_capacity_connection_*	Metrics on connections e.g. txns running for 1 hour
pg_capacity_schema_*	Metrics on disk space of schema
pg_capacity_tblspace_*	Metrics on disk space of tablespace
pg_capacity_tblvacuum_*	Metrics on tables without vacuum for days
pg_capacity_longtx_*	Number 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

Metrics name	Details
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

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.

Refer an example below.



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

Alert rule	Alert Level	Condition persistence	Description
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

** The alerts are based on statistics/metrics. If a platform statistics are incorrect, it may raise an incorrect alarm.

e.g. if the Storage Driver is not showing correct metrics for bytes usage for a PV, system may end up raising incorrect alarm of PVCLowDiskSpace. This behaviour can be seen with NFS storage.

You can configure any alert by adding alert rules to other monitoring items.

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

5.7.3 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

Column	Type	Description
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
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

Column	Type	Description
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.

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 Viewing the custom events

The custom events can be viewed on CLI as well as the Openshift console

On cli

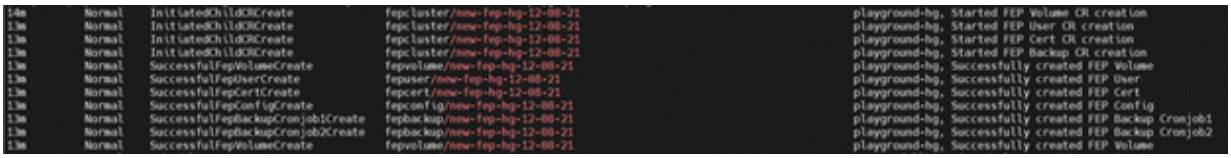
Executing the command

kubectl get events

OR

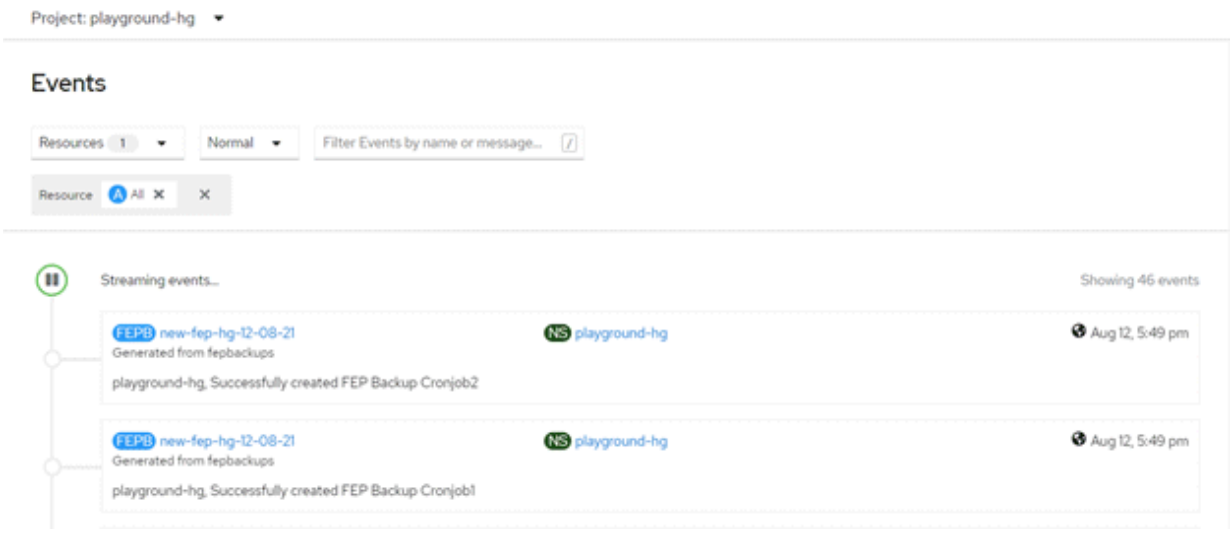
oc get events

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 Auto Scale Out

Auto 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 Auto 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

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.

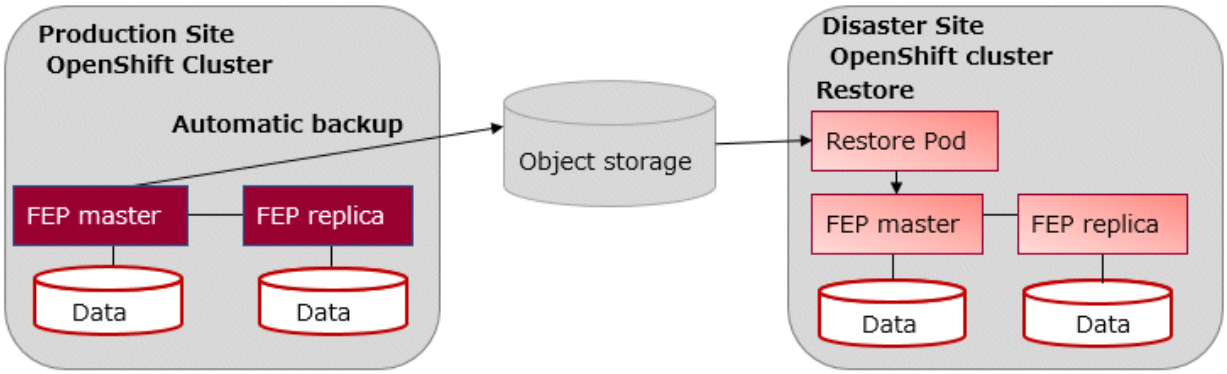
Even if it is difficult to operate in an OCP environment with a database cluster due to a disaster, it is possible to continue operating in a different OCP environment.

5.11.1 Disaster Recovery Prerequisites

The configuration diagram of the pod placement and backup repository, which are prerequisites for the backup feature for performing disaster recovery, 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.



Note

The definition of the FEPCluster custom resource is not inherited when performing disaster recovery.

We recommend that you save your production site FEPCluster custom resource definitions in case of a disaster.

5.11.2 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.

5.11.2.1 Pre-creation of Resources

5.11.2.1.1 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
```

5.11.2.1.2 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
```

5.11.2.2 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 "5.11.2.1.2 Storing 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. 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.

Field	Default	Details
		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

5.12 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 to the key management system are updated and the passphrase of the private key is updated, update the sslpassphrase value under fepChildCrVal.sysTde.tdek.kmsDefinition of the FEPCluster custom resource. The operator automatically performs a keystore open when this value is 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名> <namespace>, Successfully
set TDE masterKey

# When re-enabling TDE fails
$ kubectl get event
LAST SEEN   TYPE    REASON                      OBJECT                      MESSAGE
164m       Warning  FailedTdeSetMasterKey      fepconfig/<FEPClusterCR名> <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 there is an error in the Secret or ConfigMap that stores the private key for TLS communication with the key management system, and there is no correction for the custom resource, use the FEPAction custom resource described in "5.12.2 Update Credentials", to open the keystore.

5.12.2 Update Credentials

If the credentials of the key management system are updated, update the Secret/ConfigMap values specified under cert in sysTde.tdek.kmsDefinition and apply the FEPAction custom resource to update the credentials.

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).

When a master encryption key newer than the time of backup is specified in sysTde.tdek.targetKeyId newer than the backup time of the source FEPCluster custom resource, the value is inherited to the restore destination FEPCluster custom resource, and the operator automatically restores the data after the data is restored. Temporarily re-enable TDE with the new master encryption key.

Also, before executing the restore, update the authentication information (sslpassphrase value of sysTde.tdek.kmsDefinition or the value held in the Secret/ConfigMap specified under cert) to the key management system of the source FEPCluster custom resource. please give me. If your authentication information is not updated, you will not be able to connect to the key management service and restore your data.

If parameters other than "sysTde.tdek.targetKeyId" and "sysTde.tdek.kmsDefinition sslpassphrase" under sysTde.tdek are incorrectly updated after building FEPCluster, the key management system cannot be referenced when restoring data. Before executing the restore process, confirm that the correct values are described in the FEPCluster custom resource.

Chapter 6 Maintenance Operations

This chapter describes the maintenance operation after deploying the container.

6.1 Minor Version Upgrade

Minor FEP version upgrade is done by replacing the image in FEPCluster customer resource with a new one. For the procedure, refer to "Minor Version Upgrade" in the Overview.

Update information can be found in the Red Hat catalog to see if a new FEP database server container has been released.

Upgrades are rolling updated, so you can localize downtime, but it is recommended that you avoid running during business hours as connected applications will result in connection errors.

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>

"switchover" action type expects users to specify the name of the current leader/primary pod that they want to switchover from. Specify the name in the args section under the FEPAction CR spec as below:

```
spec:
fepAction:
 args:
 - new-fep-sts-2
 type: switchover
 targetClusterName: new-fep
```

Here, new-fep-sts-2 is the current primary.

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.

Field	Default	Details
		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. Refer to "4.5.1 Manual Certificate Management" for information about creating secrets.
spec.fepChildCrVal.upgrade.destination.pgAdminTls.certificateName		Optional

Field	Default	Details
		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 "4.5.1 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.

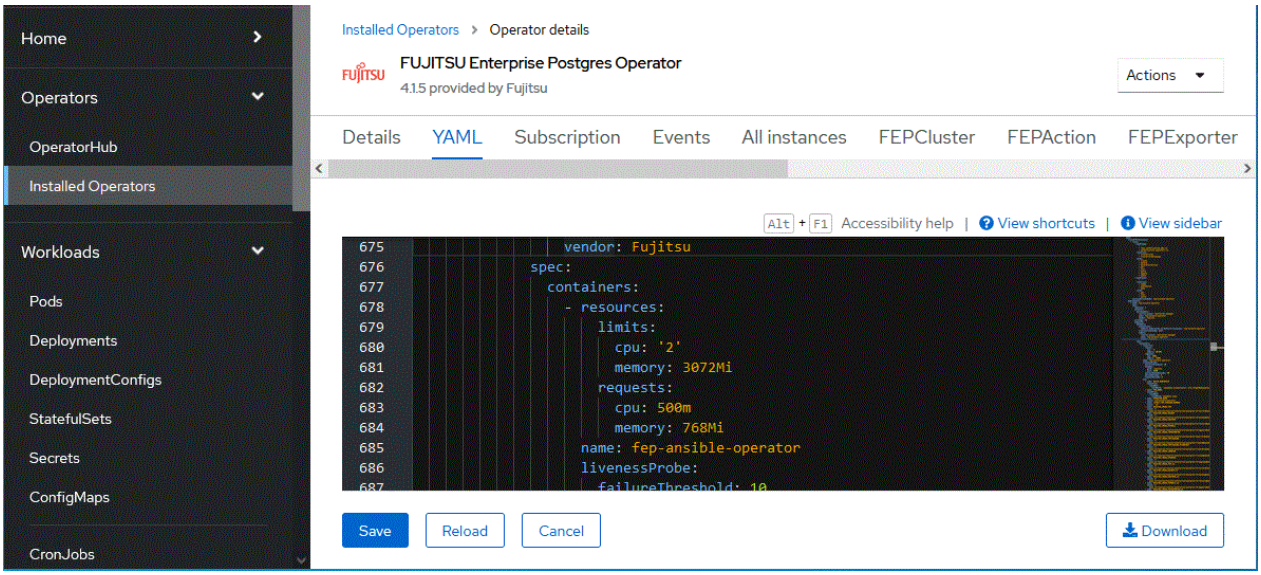
6.5.1.1 When installing using OperatorHub

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
```

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.

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.

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.

Appendix A Quantitative Values and Limitations

A.1 Quantitative Values

Refer to the FUJITSU Software 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	No Support for JIT	Since UBI8 is not having requisite LLVM libraries	It is not possible to enable JIT in postgreql.conf. Impact for the customer is that they are not able to achieve maximum performance capabilities on given CPU and memory
2	FEP parallelism improvements	Since UBI8 is not hosting dstat binaries	FEP parallelism improvement is to restrict number of parallel workers in case the CPU is already busy because of other tasks/processes. It is unlikely to have too much impact on FEP container, since container is running only one process.
3	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.
4	No Support for Oracle foreign data wrapper	Oracle foreign data wrapper has dependency on Instant Client package, which is not available.	Oracle InstantClient package is not redistributed by FUJITSU Enterprise Postgres leading to this limitation. The functionality of Oracle Foreign data wrapper is not available to FUJITSU Enterprise Postgres on Openshift environment.

Fixed parameter

Some parameters cannot be changed. Refer to "2.3.5.2 Parameters that cannot be Set".

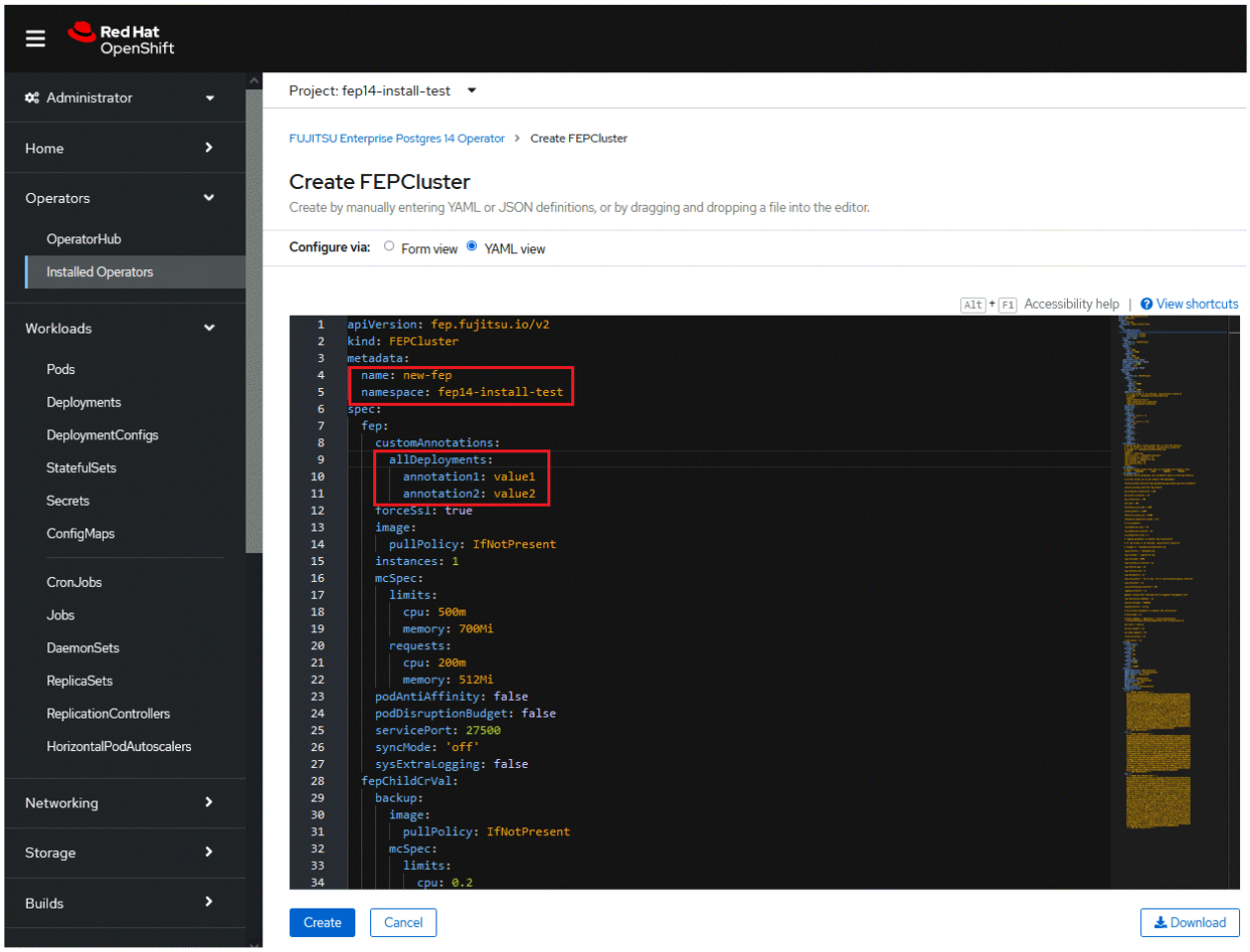
FEP features that needs to be set when using

Refer to "2.3.7 FEP Unique Feature Enabled by Default".

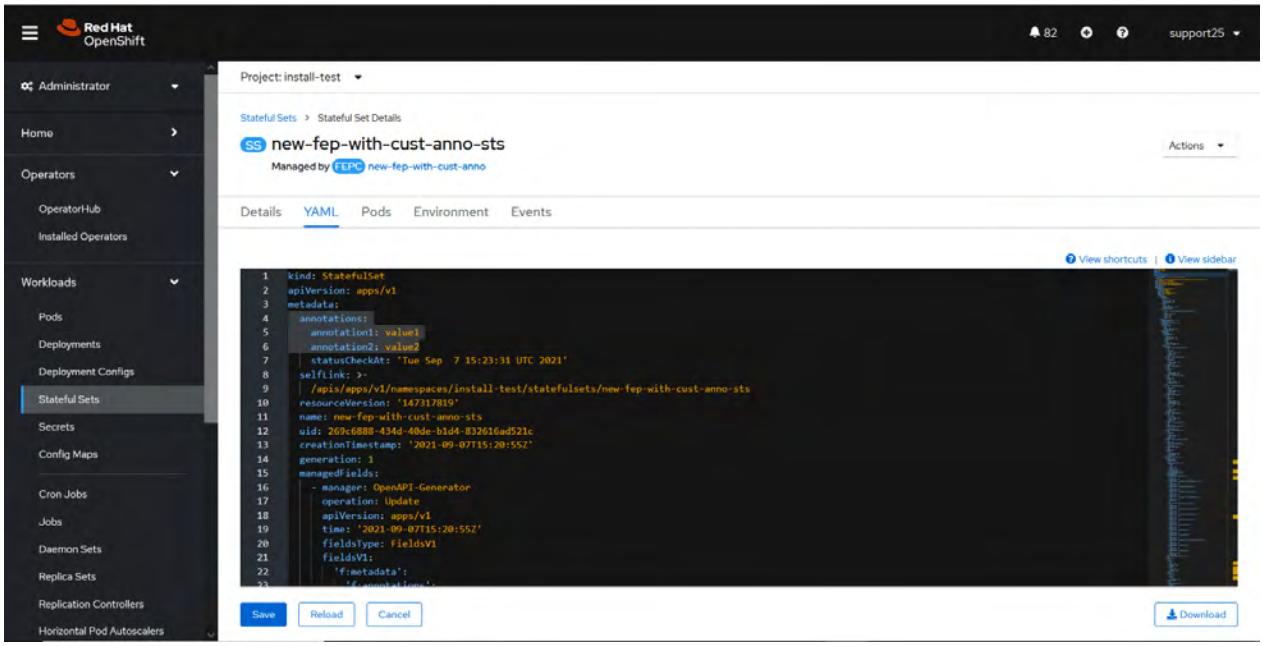
Appendix B Adding Custom Annotations to FEPCluster Pods using Operator

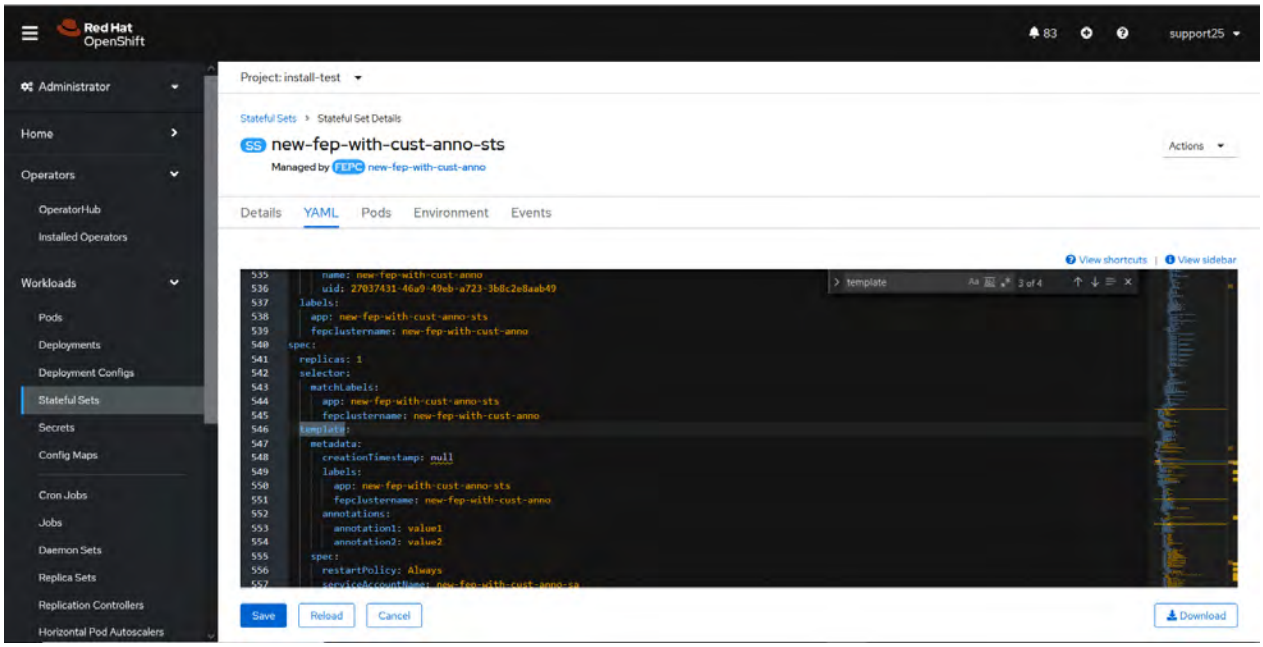
This section describes instructions for adding custom annotations to a FEPCluster pod.

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 ~
```

FUJITSU Enterprise Postgres 14 for Kubernetes

Reference Guide

Linux

FUJITSU

J2UL-RF14-05ENZ0(00)
June-2023

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:

Abbreviations

The following abbreviations are used in this manual:

Full Name	Abbreviations
FUJITSU Software Enterprise Postgres for Kubernetes FUJITSU Software Enterprise Postgres	FEP or FUJITSU Enterprise Postgres
Transparent Data Encryption	TDE
Custom Resource	CR
Custom Resource Definition	CRD
Persistent Volume	PV

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 5.0: June 2023
Edition 4.0: October 2022
Edition 3.0: September 2022
Edition 2.0: June 2022
Edition 1.0: March 2022
```

Copyright

Contents

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: 12, 13, 14 Note: Changing fepVersion from one version to another version is not supported after deployment.
spec.fep.customAnnotation.allDeployments	{} (*)	Contents under this are optional. User can remove {} and add multiple key-value pairs. All of these pair will be

Field	Default	Details
		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-14-server:ubi8-14-0.0 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	
spec.fep.mcSpec.requests	cpu: 200m memory: 512Mi	
spec.fep.sysExtraLogging	false	To turn extra debugging on, set value to true It can be turned on/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.forceSsl	true	Controls that the communication to the server should only be via SSL. Changes are reflected in pg_hba.conf
spec.fep.locale	<omitted> (*)	Optional Can only be specified when creating a FEPCluster. Database Cluster Locale Settings: ja_JP - Japanese locale Default - C

Field	Default	Details
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.
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

Field	Default	Details
		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.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, 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.

Field	Default	Details
		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		Directory where auditlog files are stored. This should match the one defined in fepChildCrVal.customPgAudit.log_directory
spec.fep.pgAuditLog.schedules		Schedule to upload auditlog
spec.fep.pgAuditLog.schedules.upload		Upload schedule in crontab format
spec.fep.pgAuditLog.endpoint.protocol		Web server protocol to upload auditlog file. Supported type is http or https
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 Name of secret that contains the username and password for Basic authentication to Web server
spec.fep.pgAuditLog.endpoint.fileUploadParameter	file	Optional The file upload parameter defined by the web server
spec.fep.pgBadger.schedules.create		The 'create' schedule to create report and upload it to endpoint
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

Field	Default	Details
spec.fep.pgBadger.endpoint.fileUpload Parameter	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.fepChildCrVal.customCertificates		Optional This is an optional parameter, which comprises of the parameters mentioned below. It is an array of elements to define certificates. Used to setup SSL connection between publisher and subscriber clusters for logical replication
spec.fepChildCrVal.customCertificate s.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.customCertificate s.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.customCertificate s.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.ima ge	<omitted>	FEP backup container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-14-backup:ubi8-14-0.0 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.pull Policy	IfNotPresent	
spec.fepChildCrVal.backup.mcSpec.li mits	cpu: 0.2	

Field	Default	Details
	memory: "300Mi"	
spec.fepChildCrVal.backup.mcSpec.requests	cpu: 0.1 memory: "200Mi"	
spec.fepChildCrVal.backup.pgbackrestParams	[global] repo1-retention-full=7 repo1-retention-full-type=time log-path=/database/log/backup	"|" When nothing is specified, and the parameter set in pgbackrest.conf is described from the line below.
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	Number of schedules to set The maximum number of backup schedules is 5.
spec.fepChildCrVal.backup.scheduleN.schedule	"|"	Backup schedule in cron format. The date and time is UTC time.
spec.fepChildCrVal.backup.scheduleN.type	"|"	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	PgAudit file content

Field	Default	Details
	[rule]	
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
	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
		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.

Field	Default	Details

	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: 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 (**)	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]

Field	Default	Details
spec.fepChildCrVal.storage.tablespaceVol		Optional volume
spec.fepChildCrVal.storage.tablespaceVol.size	512Mi (**)	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
spec.fepChildCrVal.storage.archivewalVol.size	1Gi (**)	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 (**)	Size of log volume. This volume is optional and can be omitted

Field	Default	Details
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]
spec.fepChildCrVal.storage.backupVol		Mandatory if backup section is defined. Optional otherwise
spec.fepChildCrVal.storage.backupVol.size	2Gi (**)	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]
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

Field	Default	Details
		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.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
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	Password for database user repluser Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), symbols (~! @ # $^ & * () - = < >,.? ; : |/+)
spec.fepChildCrVal.sysUsers.tdepassphrase	tde-passphrase	TDE keystore passphrase
spec.fepChildCrVal.sysUsers.pgRewindUser	rewind_user	Database user for Rewind Available character types Alphanumeric characters (A-Z, a-z), numbers (0 -9), and underscores (_)

Field	Default	Details
		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.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

Field	Default	Details
		- 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.
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

Field	Default	Details
		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 the KMS. Required when tdek is specified for spec.fepChildCrVal.sysTde.tdeType.
spec.fepChildCrVal.sysTde.tdek.targetKmsName	(*)	Specifies the KMS name. It has the same value as name in sysTde.tdek.kmsDefinition.
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.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].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

Field	Default	Details
		a-z, consisting of a-z, numbers (0 -9), and underscores. Upper and lower case letters are the same.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].type	(*)	Specifies the type of KMS. Currently, only kmip is available.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].address	(*)	Specifies the host name or IP address of the KMS.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].port	(*)	Specifies the KMS port.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].authMethod	(*)	Specifies the authentication method in KMS. Currently, the only possible value is cert.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].sslpassphrase		Optional Specifies the passphrase of the client certificate private key file when connecting to KMS. This can be omitted if no passphrase is set in the private key file.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].cert		Optional Specifies the name of the Secret/ ConfigMap containing the certificate file, etc., when cert is specified as authMethod.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].cert.certificateName	(*)	Specifies the TLS Secret name that contains the client certificate and private key for TLS communication with KMS.
spec.fepChildCrVal.sysTde.tdek.kmsDefinition[0].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.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) If omitted, off is assumed.
spec.fepChildCrVal.autoscale.scaleout.threshold	40	Specifies an integer as the threshold for performing scale out.

Field	Default	Details
		- 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.
spec.fepChildCrVal.restore.pgbackrestKeyParams		Optional "|" is fixed, and the following line describes the parameters to be set in

Field	Default	Details
		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
spec.fepChildCrVal.upgrade		Optional When this field is defined, a major version upgrade is performed.

Field	Default	Details
		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 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

Field	Default	Details
		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

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

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)
- 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
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

Field	Default	Details
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
spec.pgRewindUserPassword	rewind_password	Password for database user rewinduser
spec.pgMetricsUser	spec.fepChildCrVal.sysUsers.pgMetricsUser	Optional See details in FEPCluster CR

Field	Default	Details
spec.pgMetricsPassword	spec.fepChildCrVal.sysUsers.pgMetricsPassword	Optional See details in FEPCluster CR
spec.pgAdminTls	spec.fepChildCrVal.sysUsers.pgAdminTls	Optional section See details in FEPCluster CR
spec.pgrepluserTls	spec.fepChildCrVal.sysUsers.pgrepluserTls	Optional section See details in FEPCluster CR
spec.pgRewindUserTls	spec.fepChildCrVal.sysUsers.pgRewindUserTls	Optional section See details in FEPCluster CR
spec.pgMetricsUserTls	spec.fepChildCrVal.sysUsers.pgMetricsUserTls	Optional section See details in FEPCluster CR

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.

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.

Below is the schema of the FEPVolume CR:

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

Field	Mandatory	Sub-Field	Default	Description
		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), 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.
		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.

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
    NzEQMA4GA1UEChwHRnVqaXRzdTEjMCEGA1UEAwwaRkVQIFJvb3QgQ0EgZm9yIEt1
    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
    BQAwNzEQMA4GA1UEChwHRnVqaXRzdTEjMCEGA1UEAwwaRkVQIFJvb3QgQ0EgZm9y
    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 To use FEPRestore image of given version. Possible values: 12, 13 & 14
spec.image	<current-released-image>	FEP restore container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-14-restore:ubi8-14-0.0 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'

Field	Default	Details
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.
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.

Field	Default	Details
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.

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
spec.image	<current-released-image>	FEPPgpool2 container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-0.0 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 backend_weight0 = 1 backend_weight1 = 1 backend_flag0 = 'ALWAYS_PRIMARY'	"|" and the Pgpool-II parameters. Refer to "Pgpool-II parameters" for detail.

Field	Default	Details
	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 sr_check_user = 'postgres' delay_threshold = 0 log_standby_delay = 'none' ssl = on	

Field	Default	Details
	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.

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	

Category	Parameter name (Specified format)	Restart required after change
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)	
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

Category	Parameter name (Specified format)	Restart required after change
	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 pod_restart reload list switchover failover pgpool2_restart backup open_tde_masterkey
.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 (see more details in FEPCluster CR section):

```
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
```

- 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
```

1.2.11 FEPExporter Custom Resource

Field	Default	Details
apiVersion	fep.fujitsu.io/v1	Mandatory as it is
kind	FEPExporter	Mandatory as it is

Field	Default	Details
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.
spec.fepExporter.disableDefaultQueries	false	Optional Not defined or set to false => Create default queries Defined and set to true => Do not create default queries.

Field	Default	Details
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:ubi8-14-0.0 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.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 memory: 300Mi	Optional 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.

Field	Default	Details
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
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

Custom Resource spec	Required/ Optional	Change Effect	Updating value allowed
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.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'

Custom Resource spec	Change Effect
pgBadger.endpoint.insecure	equivalent to curl -insecure option, default to false
pgBadger.endpoint.url	Web server url to upload the report file

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)"
```

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
```

FUJITSU Enterprise Postgres 14 for Kubernetes

Quick Start Guide

Linux

FUJITSU

J2UL-QK14-05ENZ0(00)
June-2023

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 requirements

2.1. 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

2.2. Supported Platform

It supports running on the following platforms.

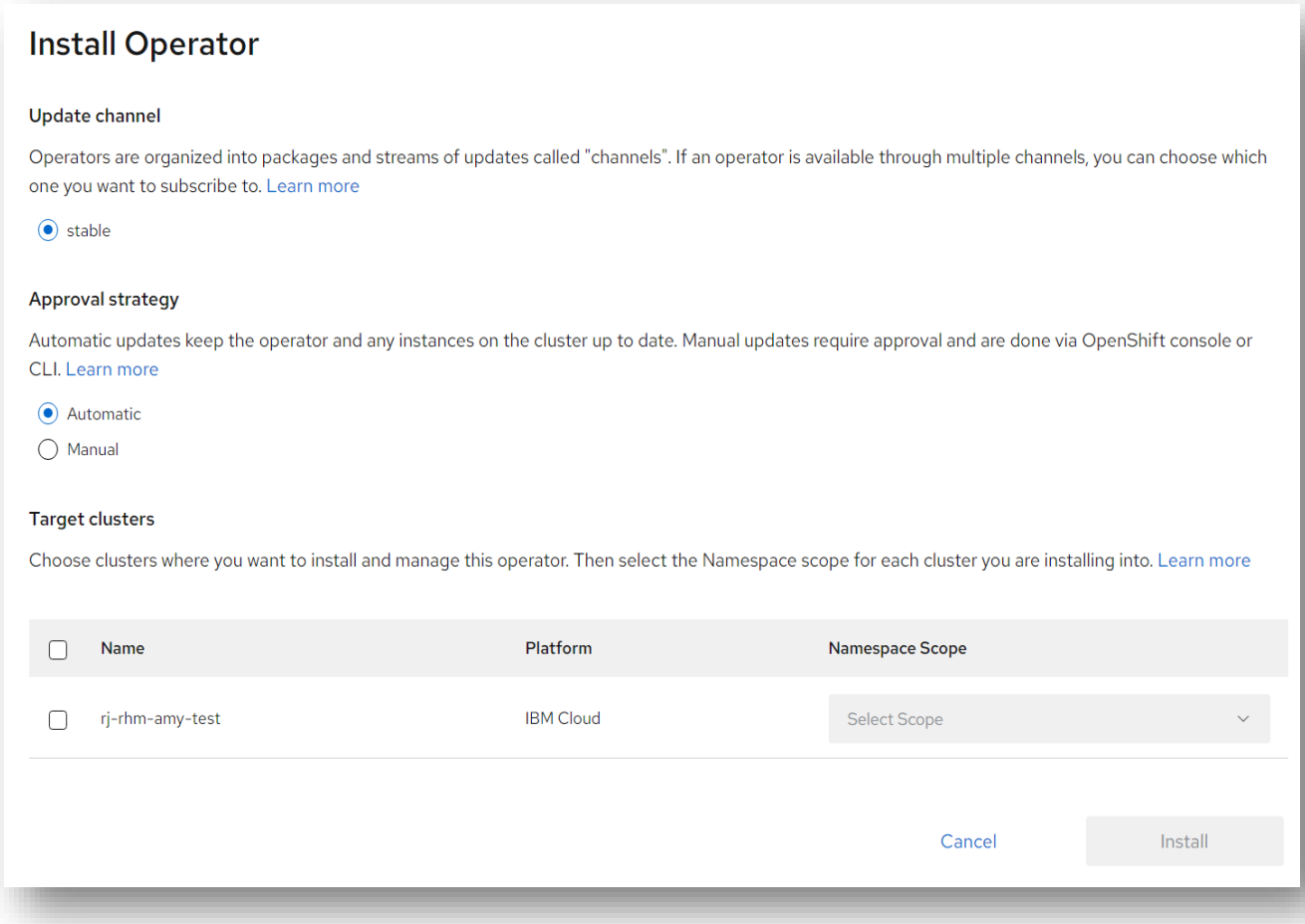
No	Platform	Version
1	OpenShift Container Platform	4.10, 4.11, 4.12

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 operator install.
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:

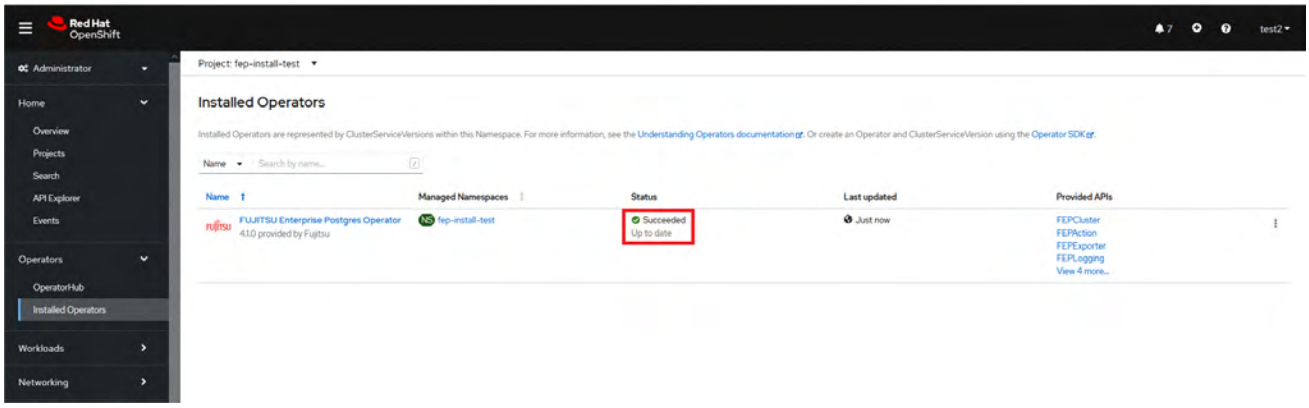
- Click the checkbox next to the clusters where you want to install the Operator.
- For each cluster you selected, under **Namespace Scope**, on the **Select Scope** list, select an option.

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**

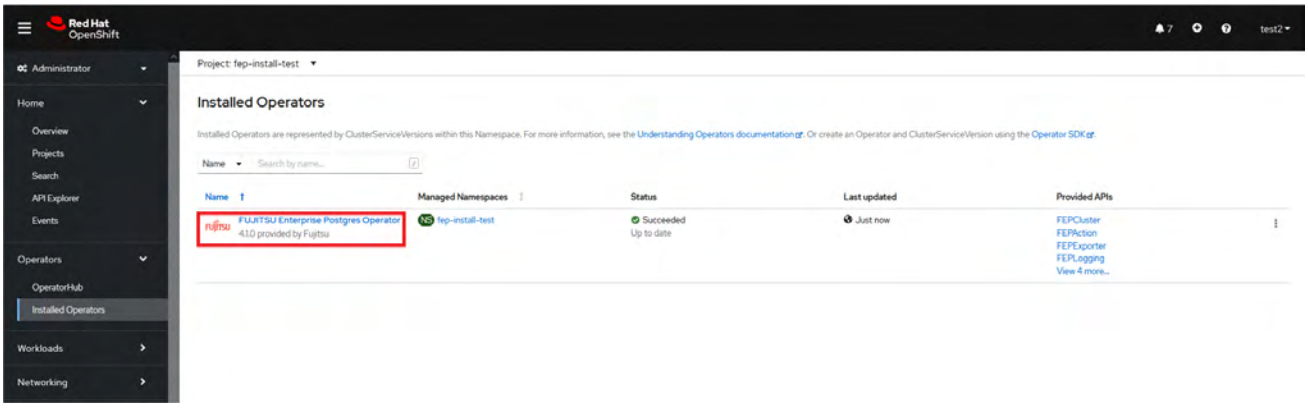


Installed operators status changes to “Succeeded”

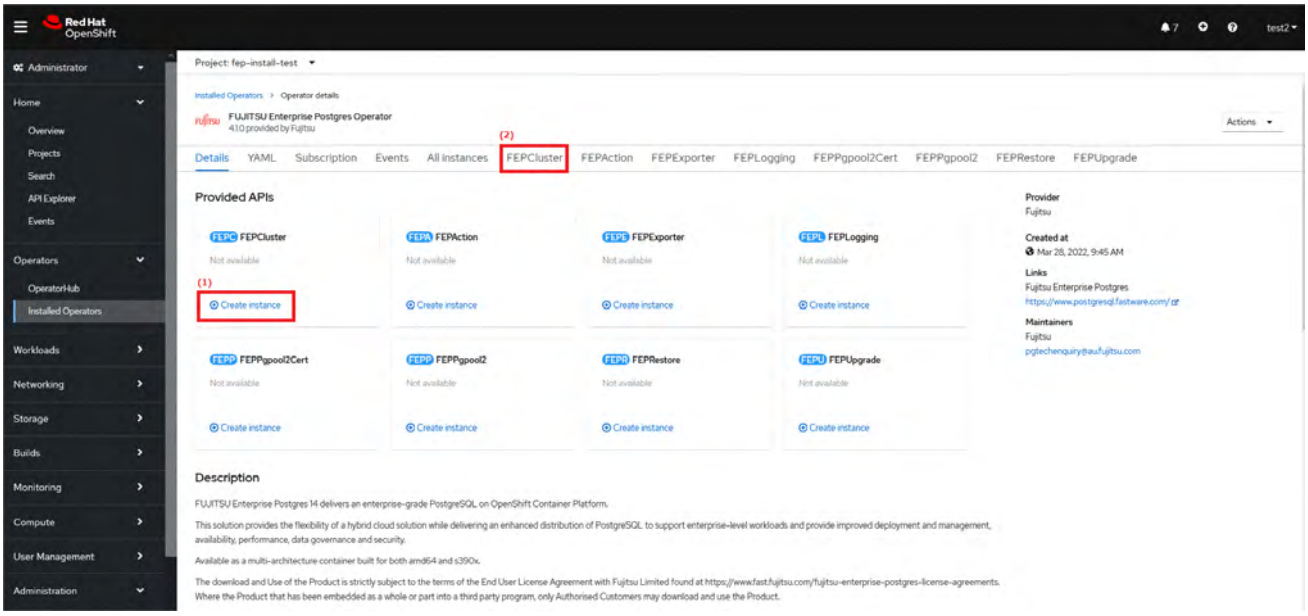
5. Deploying FEPCluster using Operator

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

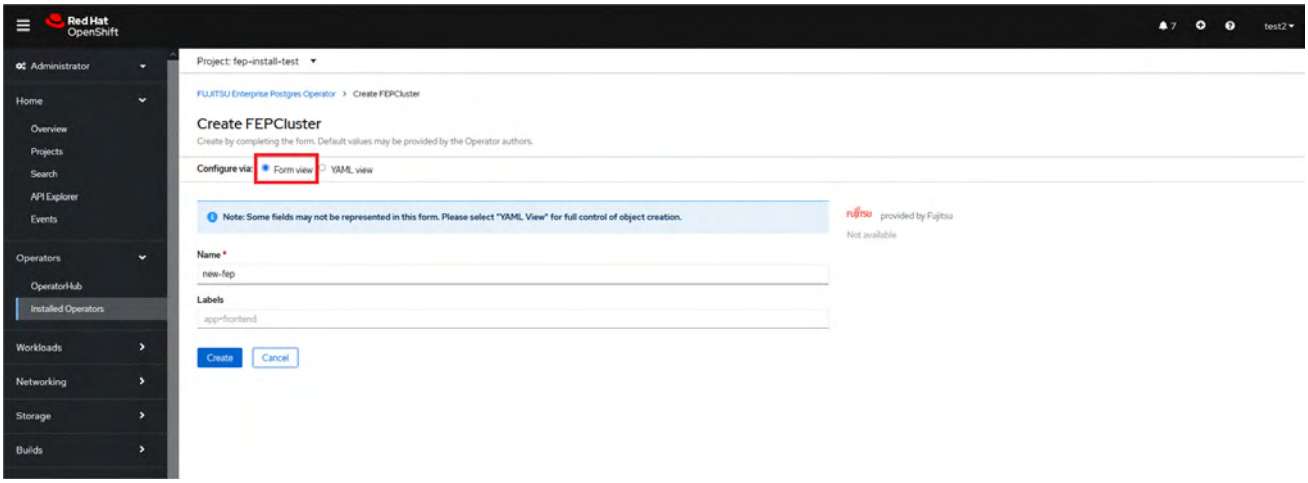
1. Under "Operators" menu item, click on "**Installed Operators**". You would see the installed FEP operator. Click on the name of operator.



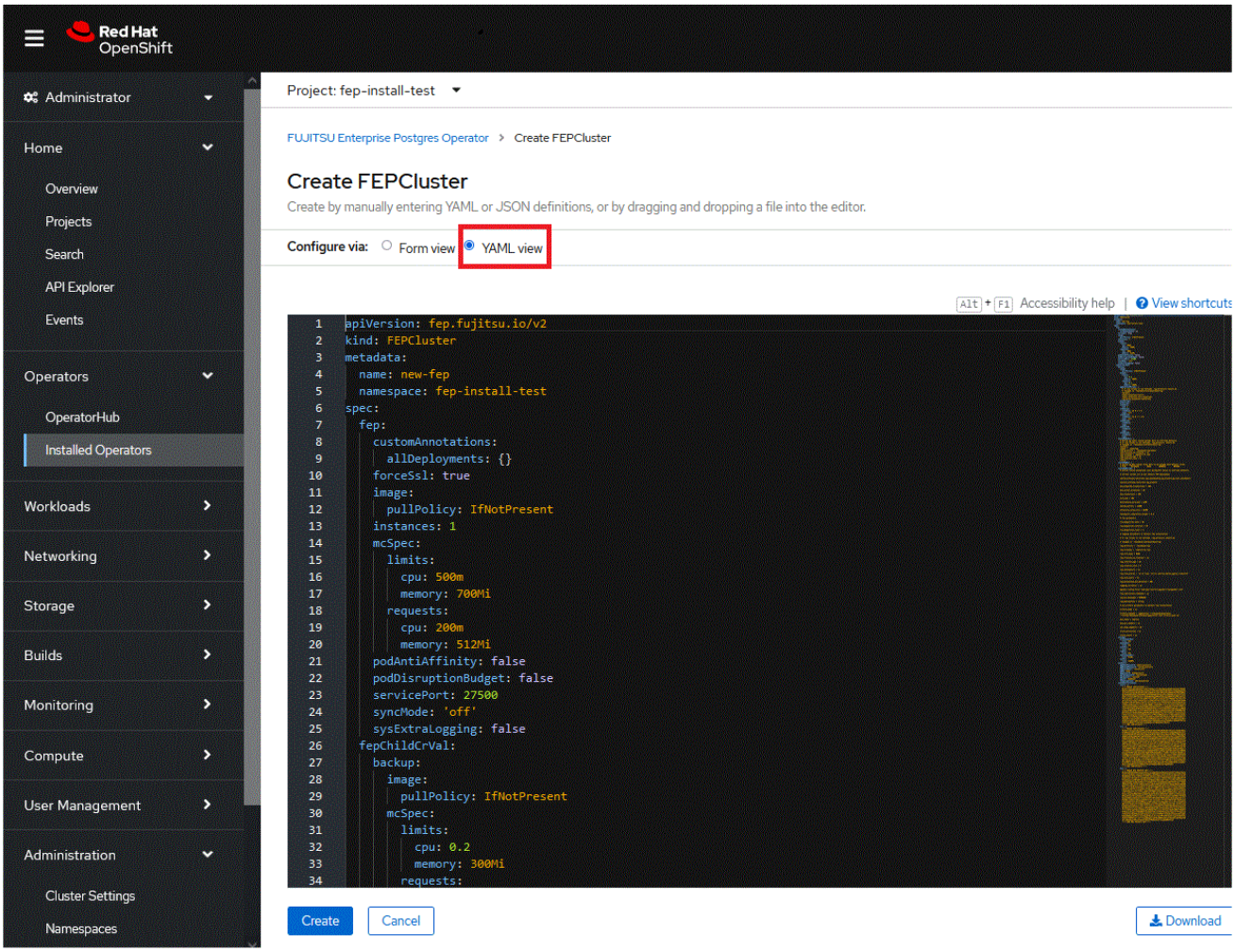
2. It will display a page with all CRs this operator supports. FEPCluster is the main CR and all others are child CR. We would create main CR and all other CRs will be created automatically by Operator.
   To create Cluster CR, either
   (1) Click on "**Create Instance**" under FEPCluster.
   OR
   (2)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



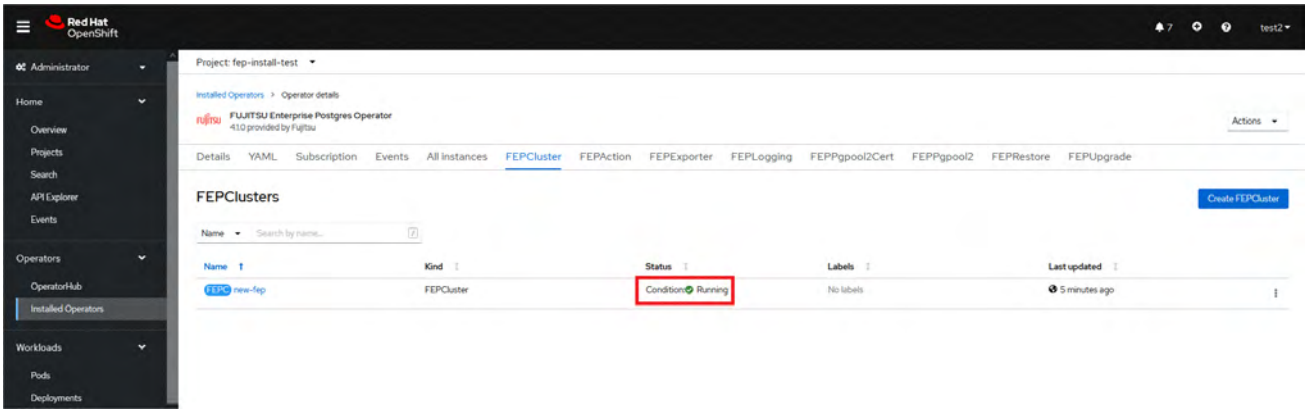
3. This will bring to "Create FEPCluster" page. Here you have two options to configure. The first one is Form View. At the moment, in Form View, one can change only the name of cluster being deployed. Default name is "new-fep".
   This name must be unique within a namespace.



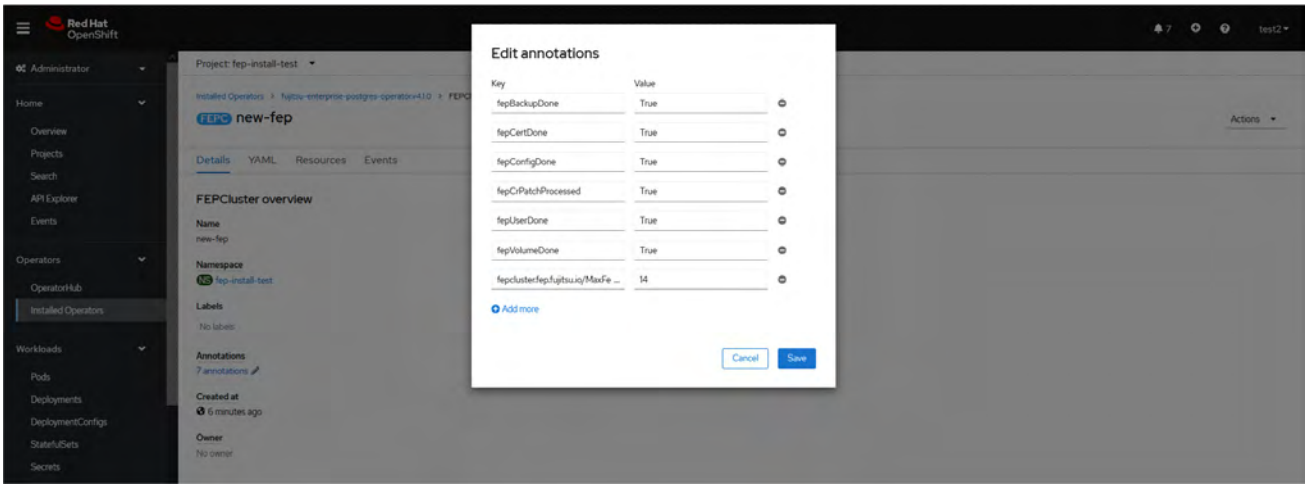
4. In YAML View, starting value of CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.For example, add a configuration value for the customPgHba parameter according to your environment.



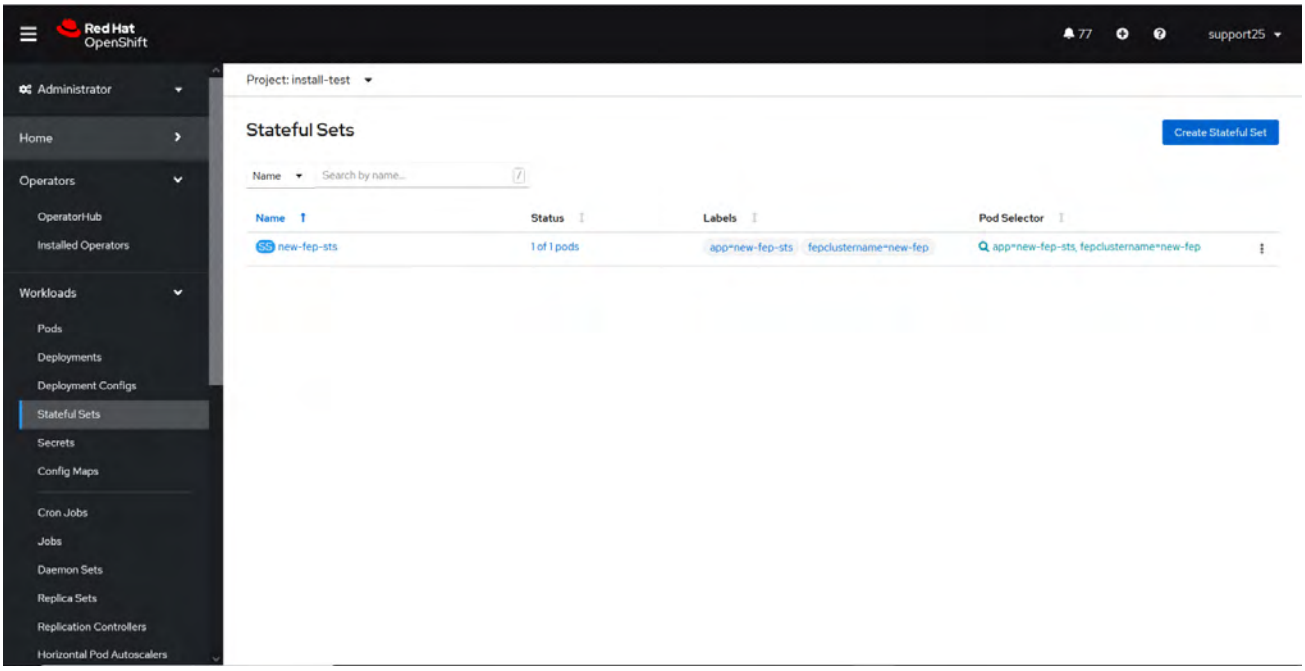
5. When "**Create**" is clicked on either of two pages above, operator creates FEPCluster CR and there after one by one FEPBackup, FEPConfig, FEPVolume, FEPUser and FEPCert child CRs are created automatically.
The starting values for child CRs are taken from "fepChildCrVal" section of FEPCluster CR yaml 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.



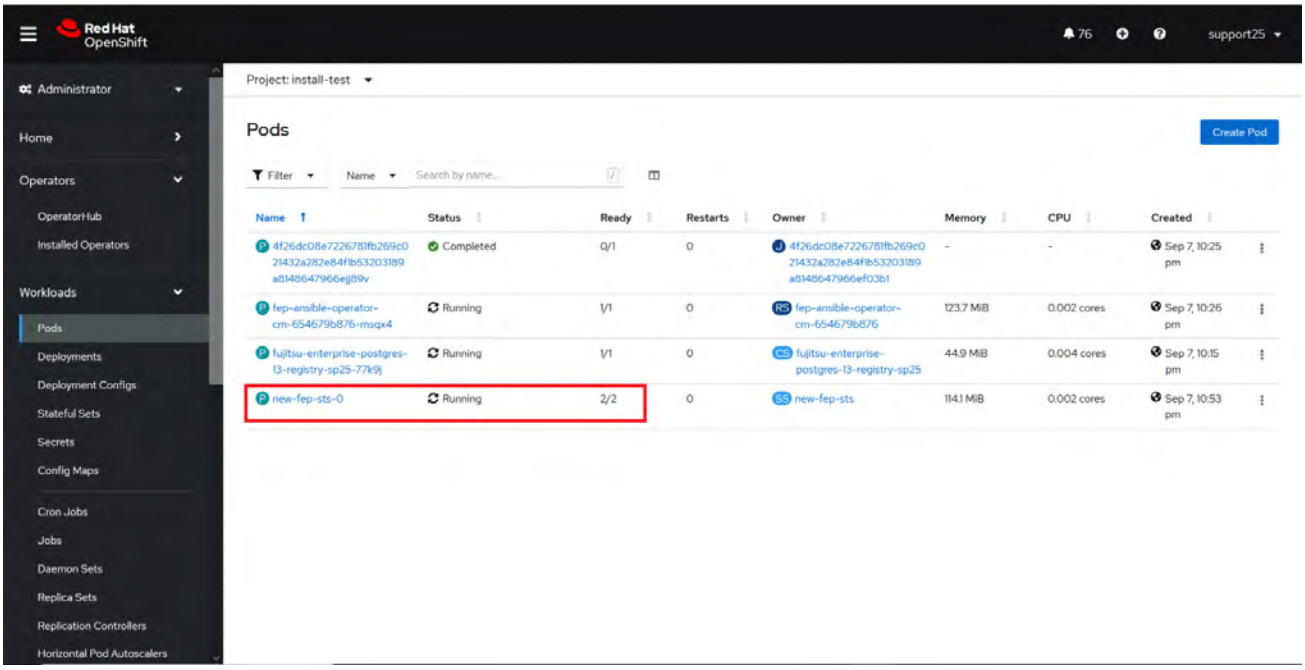
6. In FEPCluster CR, annotations are added to indicate that child CRs are created successfully and has initialized properly. It may take some time to complete.



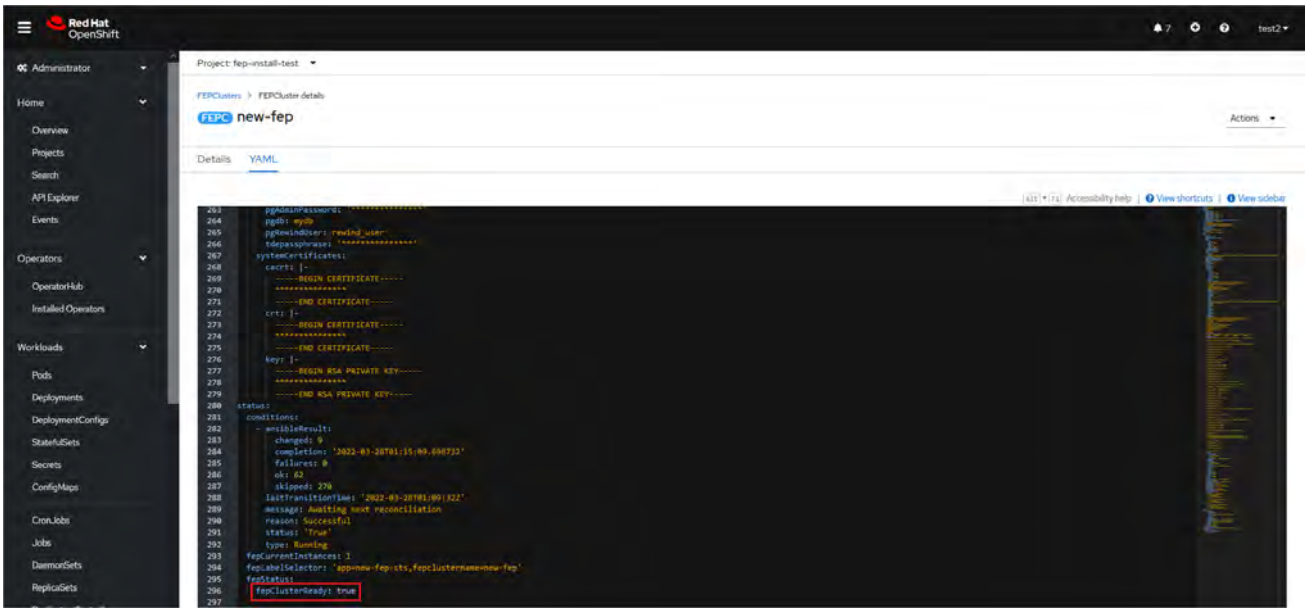
7. Once all four child CRs are marked done in annotations, operator creates StatefulSet for the cluster.



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, 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:



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 14 for Kubernetes Manuals