FUJITSU Enterprise Postgres 14 for Kubernetes

Manual Set

Linux

March 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:

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td>FEP or FUPU</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td></td>
</tr>
<tr>
<td>Custom Resource</td>
<td>CR</td>
</tr>
<tr>
<td>Universal Base Image</td>
<td>UBI</td>
</tr>
<tr>
<td>OpenShift Container Platform</td>
<td>OCP</td>
</tr>
<tr>
<td>Mutual TLS</td>
<td>MTLS</td>
</tr>
</tbody>
</table>

Abbreviations of manual titles

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

<table>
<thead>
<tr>
<th>Full Manual Title</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td></td>
</tr>
<tr>
<td>Release Notes</td>
<td></td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td></td>
</tr>
<tr>
<td>Overview</td>
<td></td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td></td>
</tr>
<tr>
<td>User's Guide</td>
<td></td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td></td>
</tr>
<tr>
<td>Reference</td>
<td></td>
</tr>
</tbody>
</table>

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.
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

<table>
<thead>
<tr>
<th>Version and level</th>
<th>Classification</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEP 14 Operator</td>
<td>OSS</td>
<td>PostgreSQL Rebase</td>
</tr>
<tr>
<td>image tag:v4.1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Container image</td>
<td>Platform enhancement</td>
<td>Additional OCP Support</td>
</tr>
<tr>
<td>tag:ubi8-14-1.0</td>
<td></td>
<td>Additional OCS Support</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Additional Kubernetes Support</td>
</tr>
<tr>
<td>Collaboration tools</td>
<td></td>
<td>Installing Operator with Helm Chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Additional Rancher Support</td>
</tr>
<tr>
<td>Operation</td>
<td></td>
<td>Disaster Recovery</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Server Log Monitoring</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changing the Switchover FEPAction Interface</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Major Version Upgrade</td>
</tr>
<tr>
<td></td>
<td></td>
<td>On-Demand Backup</td>
</tr>
</tbody>
</table>

1.1 Features Added FEP14 Operator in v4.1.0

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

1.1.1 OSS

This section explains the new feature related to OSS:

- PostgreSQL rebase

1.1.1.1 PostgreSQL Rebase

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

See

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

1.1.2 Platform Enhancement

This section explains the new features related to platform enhancement:

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

1.1.2.1 Additional OCP Support

The following additional OCP is supported:

- OCP 4.9
1.1.2.2 Additional OCS Support

The following additional OCS is supported:
- OCS 4.8
- OCS 4.9

1.1.2.3 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)

1.1.3 Collaboration Tools

This section explains the new features related to installation:
- Installing Operator with Helm Chart
- Additional Rancher support

1.1.3.1 Installing Operator with Helm Chart

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

1.1.3.2 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.
1.1.4 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

1.1.4.1 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.

<table>
<thead>
<tr>
<th>Feature</th>
<th>FEP12</th>
<th>FEP13</th>
<th>FEP14</th>
</tr>
</thead>
<tbody>
<tr>
<td>fep server</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>backup sidecar</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>restore</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>pgpool2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>fep exporter</td>
<td>-</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>fep logging (fluentD)</td>
<td>-</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>fep logging sidecar (fluentbit)</td>
<td>-</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>pgBadger</td>
<td>-</td>
<td>-</td>
<td>Y</td>
</tr>
</tbody>
</table>

Y: Supported
-: Not supported

1.1.4.2 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.

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

1.1.4.3 Server Log Monitoring

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

Refer to "FEP Logging" in the User's Guide for details.
1.1.4.4 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.

1.1.4.5 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.

1.1.4.6 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
Preface

Purpose of this document
This document explains the FUJITSU Enterprise Postgres for Kubernetes concepts to those who are to operate databases using it.
This document explains the features of FUJITSU Enterprise Postgres for Kubernetes.

Intended readers
This document is intended for people who are:
- Considering installing FUJITSU Enterprise Postgres for Kubernetes
- Using FUJITSU Enterprise Postgres for Kubernetes for the first time
- Wanting to learn about the concept of FUJITSU Enterprise Postgres for Kubernetes
- Wanting to see a functional overview of FUJITSU Enterprise Postgres for Kubernetes

Readers of this document are also assumed to have general knowledge of:
- Linux
- Kubernetes
- Containers
- Operators

Structure of this document
This document is structured as follows:

Chapter 1 Know about the Product
Explains the features of FUJITSU Enterprise Postgres for Kubernetes.

Chapter 2 Know What it does
Explains what you need to do.

Appendix A OSS Supported by FUJITSU Enterprise Postgres for Kubernetes
Explains the OSS supported by FUJITSU Enterprise Postgres for Kubernetes.

Abbreviations
The following abbreviations are used in this manual:

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td>FEP or FEP</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td>FUJITSU Enterprise Postgres</td>
</tr>
<tr>
<td>Custom Resource</td>
<td>CR</td>
</tr>
<tr>
<td>Custom Resource Definition</td>
<td>CRD</td>
</tr>
<tr>
<td>Persistent Volume</td>
<td>PV</td>
</tr>
<tr>
<td>GAP</td>
<td>Grafana, Alert Manager, Prometheus</td>
</tr>
</tbody>
</table>

Abbreviations of manual titles
The following abbreviations are used in this manual as manual titles:
<table>
<thead>
<tr>
<th>Full Manual Title</th>
<th>Abbreviations</th>
</tr>
</thead>
</table>

**Trademarks**

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

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

**Export restrictions**

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

**Issue date and version**

Edition 1.0: March 2022

**Copyright**

Copyright 2021-2022 FUJITSU LIMITED
Contents

Chapter 1 Know about the Product.......................................................................................................................................... 1
  1.1 What is FUJITSU Software Enterprise Postgres for Kubernetes?.......................................................................................... 1
  1.2 Operator Features.................................................................................................................................................................... 1
    1.2.1 Cluster Deployment.............................................................................................................................................................. 2
      1.2.1.1 Creating a FEPCluster................................................................................................................................................... 2
      1.2.1.2 Creating a FEP Pgpool2 Container.................................................................................................................................. 3
    1.2.2 Highly Available Feature....................................................................................................................................................... 3
      1.2.2.1 Automatic Failover......................................................................................................................................................... 3
      1.2.2.2 Automatic Recovery..................................................................................................................................................... 3
      1.2.2.3 Manual Switchover...................................................................................................................................................... 3
    1.2.3 Backup Recovery.................................................................................................................................................................. 4
      1.2.3.1 Automatic Backup.......................................................................................................................................................... 4
      1.2.3.2 Point-in-time Recovery.................................................................................................................................................. 4
    1.2.4 Configuration Change........................................................................................................................................................... 4
      1.2.4.1 Parameter Change.......................................................................................................................................................... 4
      1.2.4.2 Resource Change.......................................................................................................................................................... 4
    1.2.5 Minor Version Upgrade......................................................................................................................................................... 4
      1.2.5.1 Minor Version Upgrade................................................................................................................................................ 4
    1.2.6 FEP Features......................................................................................................................................................................... 5
      1.2.6.1 Scope of FEP Feature Support..................................................................................................................................... 5
    1.2.7 Monitoring & Alert............................................................................................................................................................... 5
      1.2.7.1 Monitoring.................................................................................................................................................................... 5
      1.2.7.2 Alert and Event............................................................................................................................................................. 6
    1.2.8 Scaling Replicas................................................................................................................................................................... 6
      1.2.8.1 Automatic Scale out.................................................................................................................................................... 6
      1.2.8.2 Manual Scale in/out................................................................................................................................................... 6
    1.2.9 Disaster Recovery............................................................................................................................................................... 6
  1.3 Operator System Configuration....................................................................................................................................... 6

Chapter 2 Know What it does............................................................................................................................................... 9
  2.1 Deployment........................................................................................................................................................................ 9
  2.2 High Availability (Automatic failover and recovery)...................................................................................................... 9
  2.3 Configuration Change.................................................................................................................................................... 9
  2.4 Upgrade............................................................................................................................................................................. 10
    2.4.1 Minor Version Upgrade.................................................................................................................................................. 10
    2.4.2 Major Version Upgrade............................................................................................................................................... 10
  2.5 Configurable Volume Per Cluster..................................................................................................................................... 10
  2.6 Deploying Pgpool-II and Connect to FEPCluster from Operator..................................................................................... 10
  2.7 Backup............................................................................................................................................................................... 11
    2.7.1 Scheduling Backup from Operator...................................................................................................................................... 11
    2.7.2 On-Demand Backup..................................................................................................................................................... 11
  2.8 Perform PITR and Latest Backup Restore from Operator................................................................................................ 11
  2.9 Monitoring & Alert...................................................................................................................................................... 12
  2.10 Server Log Monitoring.................................................................................................................................................... 13
    2.10.1 pgBadger Log Monitoring............................................................................................................................................. 13
    2.10.2 Prometheus and Elasticsearch Log Monitoring.................................................................................................................. 14
  2.11 Scaling Replicas.............................................................................................................................................................. 14
    2.11.1 Using the automatic scale out function........................................................................................................................... 15
  2.12 Disaster Recovery..................................................................................................................................................... 16

Appendix A OSS Supported by FUJITSU Enterprise Postgres for Kubernetes................................................................. 17
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.

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.
1.2.6 FEP Features

1.2.6.1 Scope of FEP Feature Support

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.

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

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
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 FEP Cluster without configuring Pgpool-II. In this deployment scenario, end user application will point its connection to the FEP master service. If a failover/switchover occurs, the FEP master service will point to the new FEP master Pod automatically. The end user application will experience a disconnection. When it re-establishes the connection, it will be connected to the new FEP master Pod. There is no need to reconfigure the application connection string.
Chapter 2 Know What it does

This chapter explains what you need to do.

2.1 Deployment

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

2.2 High Availability (Automatic failover and recovery)

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

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

2.3 Configuration Change

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

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

2.4.1 Minor Version Upgrade

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

2.4.2 Major Version Upgrade

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

Application outages are required during major version upgrades.

2.5 Configurable Volume Per Cluster

To improve performance, may want to separate the volume storing database files and WAL files. Similarly, one may want to use a dedicated volume for a new tablesapce. 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.
Monitoring & Alert 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 & FEP Cluster 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 gauges 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 metrics 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.
The OSS supported by FUJITSU Enterprise Postgres for Kubernetes is listed below.

<table>
<thead>
<tr>
<th>OSS name</th>
<th>Version and level</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgreSQL</td>
<td>14.0</td>
<td>Database management system</td>
<td>PostgreSQL Documentation</td>
</tr>
<tr>
<td>oracle</td>
<td>3.17.0</td>
<td>Oracle-compatible SQL features</td>
<td>“Compatibility with Oracle Databases” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>Pgpool-II</td>
<td>4.2.6</td>
<td>Failover, connection pooling, load balancing, etc.</td>
<td>“Pgpool-II” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>oracle_fdw</td>
<td>2.4.0</td>
<td>Connection to the Oracle database server</td>
<td>“oracle_fdw” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>pg_statsinfo</td>
<td>13.0</td>
<td>Collection and accumulation of statistics</td>
<td>“pg_statsinfo” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>pg_hint_plan</td>
<td>13.1.3.7</td>
<td>Tuning (statistics management, query tuning)</td>
<td>-  “pg_hint_plan” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-  “Optimizer Hints” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>pg_dbms_stats</td>
<td>1.5.0</td>
<td></td>
<td>-  “pg_dbms_stats” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-  “Locked Statistics” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>pg_repack</td>
<td>1.4.7</td>
<td>Table reorganization</td>
<td>“pg_repack” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>pg_rman</td>
<td>1.3.13</td>
<td>Backup and restore management</td>
<td>“pg_rman” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>pgBadger</td>
<td>11.6</td>
<td>Log analysis</td>
<td>“pgBadger” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>pg_bigm</td>
<td>1.2</td>
<td>Full-text search (multibyte)</td>
<td>“pg_bigm” in the FUJITSU Enterprise Postgres Installation and Setup Guide for Server</td>
</tr>
<tr>
<td>PostgreSQL JDBC driver</td>
<td>42.2.23</td>
<td>JDBC driver</td>
<td>“JDBC Driver” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>psqlODBC</td>
<td>13.02.0000</td>
<td>ODBC driver</td>
<td>“ODBC Driver” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>pgBackRest</td>
<td>2.36</td>
<td>Backup and restore management</td>
<td>“Scheduling Backup from Operator” in the User’s Guide</td>
</tr>
<tr>
<td>patroni</td>
<td>2.1.2</td>
<td>Postgres cluster management</td>
<td>“High Availability” in the User’s Guide</td>
</tr>
<tr>
<td>postgres-exporter</td>
<td>0.10.0</td>
<td>Postgresql metrics monitoring capabilities for Prometheus with Fujitsu updated queries</td>
<td>“Monitoring” in the User's Guide</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td>FEP orFUJITSU Enterprise Postgres</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td>VCI</td>
</tr>
<tr>
<td>Vertical Clustered Index</td>
<td>TDE</td>
</tr>
<tr>
<td>Transparent Data Encryption</td>
<td>PITR</td>
</tr>
<tr>
<td>Point in time recovery</td>
<td>CR</td>
</tr>
<tr>
<td>Custom Resource</td>
<td>CRD</td>
</tr>
<tr>
<td>Custom Resource Definition</td>
<td>PV</td>
</tr>
<tr>
<td>Persistent Volume</td>
<td>UBI</td>
</tr>
<tr>
<td>Universal Base Image</td>
<td>OCP</td>
</tr>
<tr>
<td>OpenShift Container Platform</td>
<td>MTLS</td>
</tr>
</tbody>
</table>

Abbreviations of manual titles

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

<table>
<thead>
<tr>
<th>Full Manual Title</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes Release Notes</td>
<td>Release Notes</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes Overview</td>
<td>Overview</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes Reference</td>
<td>Reference</td>
</tr>
</tbody>
</table>

Trademarks

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

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

Export restrictions

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

Issue date and version

Edition 1.0: March 2022
Chapter 5 Post-Deployment Operations

5.1 Configuration Change
5.2 FEPPool2 Configuration Change
5.3 Scheduling Backup from Operator
5.4 Configure MTLS Setting
5.5 Monitoring
5.5.1 Monitoring FEP Operator and Operands
5.5.2 Monitoring FEP Server
5.5.2.1 Architecture
5.5.2.2 Default Server Metrics Monitoring
5.5.2.3 Default Alerts
5.5.2.4 Graphical user interface
5.5.3 Monitoring FEP Backup
5.5.3.1 pgbackrest_info_backup view
5.5.4 Monitoring FEP PGPool2
5.5.4.1 pgpool2_stat_load_balance view
5.5.4.2 pgpool2_stat_conn_pool view
5.5.4.3 pgpool2_stat_sql_command view
5.6 Event Notification
5.6.1 Events raised
5.6.2 Viewing the custom events
5.7 Scaling Replicas
5.7.1 Auto Scale Out
5.7.2 Manual Scale In/Out
5.8 Backing Up to Object Storage
5.8.1 Storing CA Files (Root Certificates)
5.8.2 Defining a FEPCluster Custom Resource
5.9 Disaster Recovery
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 6 Maintenance Operations</td>
<td>92</td>
</tr>
<tr>
<td>6.1 Minor Version Upgrade</td>
<td>92</td>
</tr>
<tr>
<td>6.2 Cluster Master Switchover</td>
<td>92</td>
</tr>
<tr>
<td>6.3 Perform PITR and the Latest Backup Restore from Operator</td>
<td>92</td>
</tr>
<tr>
<td>6.3.1 Setting Item</td>
<td>93</td>
</tr>
<tr>
<td>6.3.2 After Restore</td>
<td>93</td>
</tr>
<tr>
<td>6.4 Major Version Upgrade</td>
<td>93</td>
</tr>
<tr>
<td>6.4.1 Pre-work on the Data Source FEP Cluster</td>
<td>93</td>
</tr>
<tr>
<td>6.4.2 Operator Upgrade</td>
<td>93</td>
</tr>
<tr>
<td>6.4.2.1 Uninstalling the Old Operator</td>
<td>94</td>
</tr>
<tr>
<td>6.4.2.2 Installing a New Version of the Operator</td>
<td>94</td>
</tr>
<tr>
<td>6.4.3 Major Version Upgrade of FEP</td>
<td>94</td>
</tr>
<tr>
<td>6.4.3.1 Creating a New FEPCluster CR</td>
<td>94</td>
</tr>
<tr>
<td>6.4.3.2 Verifying FEP Major Upgrade Complete</td>
<td>97</td>
</tr>
<tr>
<td>6.4.4 Updating Each Custom Resource</td>
<td>97</td>
</tr>
<tr>
<td>6.4.4.1 Removing a FEPClusterCR for a Data Source</td>
<td>97</td>
</tr>
<tr>
<td>6.4.4.2 FEPPgpool2</td>
<td>97</td>
</tr>
<tr>
<td>6.4.4.3 FEPExporter Built in Standalone Mode</td>
<td>97</td>
</tr>
<tr>
<td>Chapter 7 Abnormality</td>
<td>98</td>
</tr>
<tr>
<td>7.1 Handling of Data Abnormalities</td>
<td>98</td>
</tr>
<tr>
<td>7.2 Handling when the Capacity of the Data Storage Destination or Transaction Log Storage Destination is Insufficient</td>
<td>98</td>
</tr>
<tr>
<td>7.3 What to do when the Capacity of the Backup Data Storage Area is Insufficient</td>
<td>98</td>
</tr>
<tr>
<td>7.4 Handling Access Abnormalities When Instance Shutdown Fails</td>
<td>98</td>
</tr>
<tr>
<td>7.5 Collection of Failure Investigation Information</td>
<td>98</td>
</tr>
<tr>
<td>Appendix A Quantitative Values and Limitations</td>
<td>100</td>
</tr>
<tr>
<td>A.1 Quantitative Values</td>
<td>100</td>
</tr>
<tr>
<td>A.2 Limitations</td>
<td>100</td>
</tr>
<tr>
<td>Appendix B Adding Custom Annotations to FEPCluster Pods using Operator</td>
<td>101</td>
</tr>
<tr>
<td>Appendix C Utilize Shared Storage</td>
<td>103</td>
</tr>
<tr>
<td>C.1 Creating a StorageClass</td>
<td>103</td>
</tr>
<tr>
<td>C.2 Creating a PersistentVolume</td>
<td>103</td>
</tr>
<tr>
<td>C.3 Creating FEPCluster</td>
<td>104</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>No</th>
<th>Component</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Red Hat UBI minimal</td>
<td>8</td>
<td>Meant to provide base OS image for the container</td>
</tr>
<tr>
<td>2</td>
<td>FUJITSU Enterprise Postgres Server</td>
<td>14.0</td>
<td>To provide server capabilities</td>
</tr>
<tr>
<td>3</td>
<td>Patroni</td>
<td>2.1.2</td>
<td>To provide HA capabilities and other management to the Cluster</td>
</tr>
</tbody>
</table>

1.2 CPU

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

<table>
<thead>
<tr>
<th>No</th>
<th>CPU architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x86</td>
</tr>
<tr>
<td>2</td>
<td>s390x</td>
</tr>
</tbody>
</table>

1.3 Supported Platform

It supports running on the following platforms.

<table>
<thead>
<tr>
<th>No</th>
<th>Platform</th>
<th>Version</th>
<th>Support environments are as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OpenShift Container Platform</td>
<td>4.6, 4.7, 4.8, 4.9</td>
<td>- Azure Kubernetes Service</td>
</tr>
<tr>
<td>2</td>
<td>OpenShift Container Storage</td>
<td>4.6, 4.7, 4.8, 4.9</td>
<td>- Amazon EKS</td>
</tr>
<tr>
<td>3</td>
<td>Kubernetes</td>
<td>1.21, 1.22</td>
<td>- Rancher Kubernetes Engine (on Linux hosts)</td>
</tr>
</tbody>
</table>

Supports storage supported by OpenShift or Kubernetes (AKS, EKS and RKE).

However, backup and archive WAL volumes require shared storage, such as NFS.

1.4 Collaboration Tool

Supports integration with the following tools.

<table>
<thead>
<tr>
<th>No</th>
<th>Tool</th>
<th>Version</th>
<th>How to obtain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prometheus</td>
<td>- OpenShift</td>
<td>- OpenShift</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The version installed OpenShift</td>
<td>Preinstalled with OpenShift</td>
</tr>
<tr>
<td>No</td>
<td>Tool</td>
<td>Version</td>
<td>How to obtain</td>
</tr>
<tr>
<td>----</td>
<td>----------------------</td>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Kubernetes</td>
<td>- Kubernetes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v0.52.1 and later</td>
<td>prometheus-operator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Rancher</td>
<td><a href="https://github.com/prometheus-operator/prometheus-operator">https://github.com/prometheus-operator/prometheus-operator</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The version provided by Rancher Monitoring Chart</td>
<td>Rancher Using the Rancher Monitoring Chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Grafana</td>
<td>- OpenShift and Kubernetes</td>
<td>OpenShift Provided by OperatorHub</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.10.3 and later</td>
<td>Kubernetes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Rancher</td>
<td>grafana-operator</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The version provided by Rancher Monitoring Chart</td>
<td><a href="https://github.com/grafana-operator/grafana-operator">https://github.com/grafana-operator/grafana-operator</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Rancher Using the Rancher Monitoring Chart</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Helm</td>
<td>3.7.2 and later</td>
<td>Kubernetes only on only Helms Web Site</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://helm.sh/docs/intro/install/">https://helm.sh/docs/intro/install/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Rancher</td>
<td>v2.6.2 and later</td>
<td>Rancher Web Site</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="https://rancher.com/">https://rancher.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Prometheus Adapter</td>
<td>- OpenShift and Kubernetes</td>
<td>OpenShift and Kubernetes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Confirmed the operation with v0.9.1.</td>
<td>Prometheus Adapter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Rancher</td>
<td><a href="https://github.com/kubernetes-sigs/prometheus-adapter">https://github.com/kubernetes-sigs/prometheus-adapter</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The version provided by Rancher Monitoring Chart</td>
<td>Rancher Using the Rancher Monitoring Chart</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Design required to operate FEP</th>
<th>Where to find</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEP setup</td>
<td>Required.</td>
<td>2.3.1 Deployment</td>
</tr>
<tr>
<td>High availability configuration</td>
<td>Optional. (When checking or changing the behavior of high availability. However, even by default, constant high availability operation is possible.)</td>
<td>2.3.2 High Availability</td>
</tr>
<tr>
<td>Volume settings</td>
<td>Optional. (When setting the volume. However, even by default, allocate a fixed volume.)</td>
<td>2.3.3 Configurable Volume per Cluster</td>
</tr>
<tr>
<td>Pgpool-II setup</td>
<td>Optional. (When using Pgpool-II.)</td>
<td>2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator</td>
</tr>
<tr>
<td>Backup/restore settings</td>
<td>Optional. (When using a backup and restore.)</td>
<td>2.3.5 Scheduling Backup from Operator</td>
</tr>
<tr>
<td>Monitoring &amp; Alert(FEPExporter)</td>
<td>Optional. (When using Monitoring and Alert )</td>
<td>2.3.6 Perform PITR and Latest Backup Restore from Operator</td>
</tr>
<tr>
<td>Scaling Replicas</td>
<td>Optional. (When using scaling feature )</td>
<td>2.3.9 Scaling Replicas</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Configuration server type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEP operator</td>
<td>A container that accepts user requests and is responsible for automating database construction and operational operations.</td>
</tr>
<tr>
<td>FEP server container</td>
<td>A container for the FEP server.</td>
</tr>
<tr>
<td>FEP backup container</td>
<td>A container that performs scheduled backup operations. Created on the same Pod as the FEP server container.</td>
</tr>
<tr>
<td>FEP Fluentbit container</td>
<td>A container that collects FEP database CSV log and forwards it to Fluentd container for processing.</td>
</tr>
<tr>
<td>FEP pgpool2 container</td>
<td>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.</td>
</tr>
<tr>
<td>FEP restore container</td>
<td>A container that performs the restore operation. Temporarily created during a restore operation.</td>
</tr>
<tr>
<td>FEP Exporter container</td>
<td>A container that exposes http/https endpoint for monitoring stats scraping.</td>
</tr>
<tr>
<td>FEP Fluentd container</td>
<td>A container that summarises FEP log severity as metrics for Prometheus to consume. Optionally, forwards log entries to Elasticsearch for detailed log analysis.</td>
</tr>
</tbody>
</table>
### 2.2.2 User Account

The user accounts used by this product are as follows.

<table>
<thead>
<tr>
<th>User type</th>
<th>User name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure administrator</td>
<td>Mandatory</td>
<td>A system administrator (superuser) who has root privileges on all the servers that make up this product.</td>
</tr>
<tr>
<td>Database administrator</td>
<td>Mandatory</td>
<td>Install, set up, start, stop, and perform operation and maintenance of this product.</td>
</tr>
<tr>
<td>Application developer</td>
<td>Mandatory</td>
<td>Develops and executes database applications.</td>
</tr>
</tbody>
</table>

### 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.

```
```

For each `Version`, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

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

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


For each Version, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-14-backup:ubi8-14-1.0
  - fujitsu-enterprise-postgres-14-backup:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-14-backup:ubi8-14-1.0-s390x

FEP restore container

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


For each Version, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-14-restore:ubi8-14-1.0
  - fujitsu-enterprise-postgres-14-restore:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-14-restore:ubi8-14-1.0-s390x
FEP pgpool2 container

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


For each **Version**, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.0
  - fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-14-pgpool2:ubi8-14-1.0-s390x

FEP Exporter container

FEP Exporter container as for FEP server containers.


For each **Version**, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-exporter:ubi8-14-1.0
  - fujitsu-enterprise-postgres-exporter:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-exporter:ubi8-14-1.0-s390x

FEP Fluentd container

FEP Fluentd container as for FEP server containers.


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:ubi8-14-1.0
  - fujitsu-enterprise-postgres-fluentd:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-fluentd:ubi8-14-1.0-s390x

**FEP Fluentbit container**
FEP Fluentbit container as for FEP server containers.


For each Version, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS</strong></td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td><strong>FEPBaseVersion</strong></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><strong>MajorVersion</strong></td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td><strong>MinorVersion</strong></td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

The first publishing will expect following names / tagging (Manifest and Child images)
- fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.0
  - fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.0-amd64
  - fujitsu-enterprise-postgres-fluentbit:ubi8-14-1.0-s390x

**FEP Cronjob container**
FEP Cronjob container as for FEP server containers.


For each Version, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OS</strong></td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td><strong>FEPBaseVersion</strong></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><strong>MajorVersion</strong></td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
</tbody>
</table>

- 8 -
<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-cronjob:ubi8-14-1.0
- fujitsu-enterprise-postgres-cronjob:ubi8-14-1.0-amd64
- fujitsu-enterprise-postgres-cronjob:ubi8-14-1.0-s390x

**FEP Upgrade container**

FEP Upgrade container as for FEP server containers.


For each Version, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEBaseVersion</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>MajorVersion</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>MinorVersion</td>
<td>0,1,2 ...</td>
<td>To be used when minor changes in image, e.g. bug fix in container script</td>
</tr>
</tbody>
</table>

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

- fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.0
- fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.0-amd64
- fujitsu-enterprise-postgres-14-upgrade:ubi8-14-1.0-s390x

### 2.3 Design Perspective for Each Feature

This section describes the design of each feature.

**postgresql-cfg format**

A postgresql-cfg represents 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:

<table>
<thead>
<tr>
<th>Spec</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>The content may have multiple key/value pairs</td>
<td>foo=bar</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>foo1=bar1</td>
<td></td>
</tr>
<tr>
<td>The value cannot have space unless quoted.</td>
<td>foo=bar bar2</td>
<td>Invalid</td>
</tr>
<tr>
<td>The quoted value cannot have another value after</td>
<td>foo='bar bar2' something</td>
<td>Invalid</td>
</tr>
<tr>
<td>Spec</td>
<td>Example</td>
<td>Comment</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The key value pair must have a '=' sign</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>White spaces are allowed before/after/between the key value pair</td>
<td>foo = bar</td>
<td></td>
</tr>
<tr>
<td>Any content after '#' will be ignored</td>
<td># this is a comment</td>
<td>-</td>
</tr>
<tr>
<td>The value may be quoted by single quotes</td>
<td>foo='bar bar2'</td>
<td></td>
</tr>
<tr>
<td>Single quote can be escaped by two single quotes</td>
<td>foo='It’s ok'</td>
<td>Note: single quotes are not supported by Patroni edit-config command</td>
</tr>
<tr>
<td>Backslash '' will be replaced by '\' when invoking patronictl edit-config command</td>
<td>-</td>
<td>To avoid command line escape</td>
</tr>
<tr>
<td>When a key value pair is invalid, it will be ignored. The update continue to process next pair</td>
<td>foobar</td>
<td>The 'foobar' will be ignored</td>
</tr>
<tr>
<td>The container script does not validate the key and value as long as they are in correct format.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

It is recommended to use the psql's show command to verify parameter is set 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 liveness probe. Important to note that this is not expected to be configured by end user.

```yaml
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

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(FEPVolume Controller) takes care of creating the required volumes.

Any subsequent change to the volumes need to go thru the FEPVolume CR thus created.

Below is the schema of the FEPVolume CR:

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
<th>Sub-Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>archivewalVol</td>
<td>No</td>
<td>size</td>
<td>1Gi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode is only set at start</td>
</tr>
<tr>
<td>backupVol</td>
<td>No</td>
<td>size</td>
<td>2Gi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode is only set at start</td>
</tr>
<tr>
<td>dataVol</td>
<td>Yes</td>
<td>size</td>
<td>2Gi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode is only set at start</td>
</tr>
<tr>
<td>logVol</td>
<td>No</td>
<td>size</td>
<td>1Gi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode is only set at start</td>
</tr>
<tr>
<td>tablespaceVol</td>
<td>No</td>
<td>size</td>
<td>512Mi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode is only set at start</td>
</tr>
<tr>
<td>walVol</td>
<td>Yes</td>
<td>size</td>
<td>1200Mi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
</tbody>
</table>
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 “FEP pgpool2 Custom Resource Parameters” in the “Reference” for more information.

### 2.3.5 Scheduling Backup from Operator

When creating a FEP Cluster, 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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Overview of parameters</th>
<th>Setting value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Retention Option (repo retention -full)</td>
<td>Specify number of full backups to keep No default (should be set according to user backup policy)</td>
<td>natural number</td>
</tr>
<tr>
<td>Full Retention Type Option (repo retention-full-type)</td>
<td>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)</td>
<td>time/count</td>
</tr>
</tbody>
</table>

- 12 -
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.

```yaml
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Overview of parameters</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>Copy Archive Option (--archive-copy)</td>
<td>Copy the WAL segments needed for consistency to the backup</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Check Archive Mode Option (--archive-mode-check)</td>
<td>Check the PostgreSQL archive_mode setting.</td>
<td>Limited to backup from master</td>
</tr>
<tr>
<td>Backup from Standby Option (--backup-standby)</td>
<td>Back up from the standby cluster</td>
<td>Limited to backup from master</td>
</tr>
<tr>
<td>Stop Auto Option (--stop-auto)</td>
<td>Stops a previously failed backup on a new backup.</td>
<td>Because they are 9.6 not supported in</td>
</tr>
<tr>
<td>SSH client command Option (--cmd-ssh)</td>
<td>Path to ssh client executable</td>
<td>Not using ssh</td>
</tr>
<tr>
<td>Compress Option (--compress)</td>
<td>Use File Compression</td>
<td>For obsolete options (Use compresse-type option instead)</td>
</tr>
<tr>
<td>Config Option (--config)</td>
<td>pgBackRest configuration file.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Config Include Path Option (--config-include-path)</td>
<td>Path to additional pgBackRest configuration files.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Config Path Option (--config-path)</td>
<td>Base path of pgBackRest configuration files.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Delta Option (--delta)</td>
<td>Restore or Backup with Checksum</td>
<td>For new restores only</td>
</tr>
<tr>
<td>Dry Run Option (--dry-run)</td>
<td>Execute a dry-run for the command.</td>
<td>command-line only option</td>
</tr>
<tr>
<td>Lock Path Option (--lock-path)</td>
<td>Path where the lock file is stored</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Keep Alive Option (--sck-keep-alive)</td>
<td>Enable keep-alive messages on socket connections</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Spool Path Option (--spool-path)</td>
<td>Path to store temporary data for asynchronous archive-push and archive-get commands</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>Stanza Option (--stanza)</td>
<td>Defines the stanza.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Console Log Level Option (--log-level-console)</td>
<td>Console Log Level</td>
<td>It is not expected to operate on Pod.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Overview of parameters</td>
<td>Reason</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Std Error Log Level Option (--)log-level-stderr</td>
<td>Stderr log level</td>
<td>It is not expected to operate on Pod.</td>
</tr>
<tr>
<td>Log Path Option (--)log-path</td>
<td>Log File Destination</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>Azure Repository Account Option (--)repo-azure-account</td>
<td>Azure account used to store the repository</td>
<td>Azure storage is not supported</td>
</tr>
<tr>
<td>Azure Repository Container Option (--)repo-azure-container</td>
<td>Azure repository container.Azue container used to store the repository.</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Endpoint Option (--)repo-azure-endpoint</td>
<td>Azure repository endpoint.</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Key Option (--)repo-azure-key</td>
<td>Azure Repository Shared Key or Shared Access Signature</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Key Type Option (--)repo-azure-key-type</td>
<td>Azure Repository Key Type</td>
<td></td>
</tr>
<tr>
<td>Azure Repository URI Style Option (--)repo-azure-uri-style</td>
<td>Azure URI Style.</td>
<td></td>
</tr>
<tr>
<td>GCS Repository Bucket Option (--)repo-gcs-bucket</td>
<td>GCS repository bucket.</td>
<td>GCS is not supported</td>
</tr>
<tr>
<td>GCS Repository Endpoint Option (--)repo-gcs-endpoint</td>
<td>GCS repository endpoint.</td>
<td></td>
</tr>
<tr>
<td>GCS Repository Key Option (--)repo-gcs-key</td>
<td>GCS repository key.</td>
<td></td>
</tr>
<tr>
<td>GCS Repository Key Type Option (--)repo-gcs-key-type</td>
<td>GCS repository key type.</td>
<td></td>
</tr>
<tr>
<td>Repository Host Option (--)repo-host</td>
<td>Repository host for remote operations via SSH</td>
<td>Repository Host is not used</td>
</tr>
<tr>
<td>Repository Host Command Option (--)repo-host-cmd</td>
<td>Path of pgBackRest on Repository Host</td>
<td></td>
</tr>
<tr>
<td>Repository Host Configuration Option (--)repo-host-config</td>
<td>Repository Host Configuration File Path</td>
<td></td>
</tr>
<tr>
<td>Repository Host Configuration Include Path Option (--)repo-host-config-include-path</td>
<td>Repository hosts configuring include path</td>
<td></td>
</tr>
<tr>
<td>Repository Host Configuration Path Option (--)repo-host-config-path</td>
<td>Repository Host Configuration Path</td>
<td></td>
</tr>
<tr>
<td>Repository Host Port Option (--)repo-host-port</td>
<td>Repository host port when &quot;repo-host&quot; is configured</td>
<td></td>
</tr>
<tr>
<td>Repository Host User Option (--)repo-host-user</td>
<td>Repository host user when &quot;repo-host&quot; is configured</td>
<td></td>
</tr>
<tr>
<td>Repository Path Option (--)repo-path</td>
<td>Path where backups and archives are stored</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>Archive Retention Option (--)repo-retention-archive</td>
<td>The number of consecutive WAL backups to keep.</td>
<td>This option is not recommended, and WAL retention is controlled by the Full Retention Option and Full Retention Type Option.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Overview of parameters</td>
<td>Reason</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Archive Retention Type Option (--repo-retention-archive-type)</td>
<td>Backup Type for WAL Retention</td>
<td>It is recommended not to change from the default.</td>
</tr>
<tr>
<td>Differential Retention Option (--repo-retention-diff)</td>
<td>Number of incremental backups to keep</td>
<td>No incremental backups</td>
</tr>
<tr>
<td>Archive Mode Option (--archive-mode)</td>
<td>Retains or disables the archive for the restored cluster.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Exclude Database Option (--db-exclude)</td>
<td>Restore excluding the specified databases.</td>
<td>To restore the entire FEP cluster, including all databases</td>
</tr>
<tr>
<td>Include Database Option (--db-include)</td>
<td>Restore only the specified database</td>
<td>To restore the entire FEP cluster, including all databases</td>
</tr>
<tr>
<td>Link All Option (--link-all)</td>
<td>Restore all symbolic links.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Link Map Option (--link-map)</td>
<td>Changes the destination of a symbolic link.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Recovery Option Option (--recovery-option)</td>
<td>Setting options in postgresQL recovery.conf</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Tablespace Map Option (--tablespace-map)</td>
<td>Restoring tablespace to a specified directory</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>Map All Tablespaces Option (--tablespace-map-all)</td>
<td>Restores all tablespaces to the specified directory</td>
<td>No tables required because there is only one tablespace per FEPCluster</td>
</tr>
<tr>
<td>PostgreSQL Database Option (--pg-database)</td>
<td>PostgreSQL database.</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>PostgreSQL Host Option (--pg-host)</td>
<td>PostgreSQL host for remote operations via SSH</td>
<td>No SSH connection required</td>
</tr>
<tr>
<td>PostgreSQL Host Command Option (--pg-host-cmd)</td>
<td>Path of pgBackRest exe on the PostgreSQL host</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>PostgreSQL Host Configuration Option (--pg-host-config)</td>
<td>Path of the pgBackRest configuration file</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>PostgreSQL Host Configuration Include Path Option (--pg-host-config-include-path)</td>
<td>Setting pgBackRest on PostgreSQL host include path</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>PostgreSQL Host Configuration Path Option (--pg-host-config-path)</td>
<td>Path to configure pgBackRest on the PostgreSQL host</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>PostgreSQL Host Port Option (--pg-host-port)</td>
<td>SSH PortSpecification</td>
<td>No SSH connection required</td>
</tr>
<tr>
<td>PostgreSQL Host User Option (--pg-host-user)</td>
<td>The logon user when hosting PostgreSQL, if pg-host is set.</td>
<td>No SSH connection required</td>
</tr>
<tr>
<td>PostgreSQL Path Option (--pg-path)</td>
<td>PostgreSQL data directory.</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>PostgreSQL Port Option (--pg-port)</td>
<td>PostgreSQL Ports</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>PostgreSQL Socket Path Option (--pg-socket-path)</td>
<td>PostgreSQL Unix socket path</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>PostgreSQL Database User Option (--pg-user)</td>
<td>PostgreSQL database user</td>
<td>To use internal fixed values</td>
</tr>
</tbody>
</table>
2.3.5.3 Restricted Parameters

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Overview of parameters</th>
<th>Possible Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Type Option (--repo-type)</td>
<td>Type of storage to use for the repository</td>
<td>posix/s3</td>
</tr>
</tbody>
</table>

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 FEPCluster CR (in openshift UI). The postgresql.conf in container contains the following parameters:

```bash
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 a 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 metrics 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 default 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 metrics 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.

Note

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 (pg_backrest) 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.
Chapter 3 Operator Installation

This chapter describes how to install FEP operator.

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.
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 platforms.

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 14” to find FUJITSU Enterprise Postgres 14 Operator.

Click on FEP 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. Updating Custom Resource Definitions

   Update custom resource definitions before upgrading Helm Chart.

   ```
   helm show crds fep-repo/fujitsu-enterprise-postgres-operator > crds.yaml
   kubectl apply -f crds.yaml
   ```
4. 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 prerequisite 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, so use Helm to build Grafana. Detailed instructions are available at the following site for your reference.


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.

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 thereafter 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.

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
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>READY</th>
<th>STATUS</th>
<th>RESTARTS</th>
<th>AGE</th>
<th>FEPROLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>fep-ansible-operator-88f7fb4b-5jh85</td>
<td>1/1</td>
<td>Running</td>
<td>0</td>
<td>24m</td>
<td>master</td>
</tr>
<tr>
<td>new-fep-sts-0</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>17m</td>
<td>master</td>
</tr>
<tr>
<td>new-fep-sts-1</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>15m</td>
<td>replica</td>
</tr>
<tr>
<td>new-fep-sts-2</td>
<td>2/2</td>
<td>Running</td>
<td>0</td>
<td>13m</td>
<td>replica</td>
</tr>
</tbody>
</table>

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

- 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 FEP Cluster 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 FEP Cluster 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 datasource 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

```bash
openssl genrsa -aes256 -out myca.key 4096
Generating RSA private key, 4096 bit long modulus (2 primes)
```
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
................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................................+++ ................................�
```

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: 0kn9ijn8ubh7ygv

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
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

```bash
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

```bash
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.

```bash
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

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

14. Create client certificate for "postgres" user

```bash
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

```bash
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
EOF
```
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.

```bash
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.

```bash
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-primary-svc.my-namespace.svc"
    - "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
EOF
```
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'
EOF
Create FEPLogging client(prometheus) certificate

```yaml
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

```yaml
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`.

```yaml
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: fepexporter-cert
  namespace: my-namespace
spec:
  subject:
    commonName: exp1-service
EOF
```

- `nfl-fluentd-headless-service.feplogging-dev.svc.cluster.local`
duration: 8760h
usages:
- server auth
secretName: fluentd-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
cacert
    postgres:
      tls:
        certificateName: mydb-fep-cert
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/fep-server-test:ubi8-mtls-enhance_15'
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,vci,pgaudit,pg_prewarm'
  session_preload_libraries='vci,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_rotation_age = 1d
  log_rotation_size = 0
  log_checkpoint = 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

# vci parameters in default fep installation
vci.enable = on
vci.maintenance_work_mem = 256MB
vci.max_local_ros = 64MB
vci.force_max_parallelism = off

# 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-----
MIIEowIBAAKCAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEOAQEE-----------
ND4oCw==
-----END RSA PRIVATE KEY-----
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.

```yaml
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/fep-server-test:ubi8-mtls-enhance_13'
    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'
    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,vci,pgaudit,pg_prewarm'
      session_preload_libraries='vci,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

# vci parameters in default fep installation
vci.enable = on
vci.maintenance_work_mem = 256MB
vci.max_local_ros = 64MB
vci.force_max_parallelism = off

# 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/ipRhQOHxf05x7j90nse
    mTP61a6aGvDhezr98/FdchYj3cw81X0kU6xamqrrKQY1lxQG48N10qow0s6ak
    AHF4emFcr80t44xA1DAljC2S1lRSEctU32+30j+y9j07Enj1R32KUHw30pU
    9pDneyInexynB/P76cx3MetWtgmvp/vkJ18bp6qG6wYRshQsR1C14dtLeecEa24j
    uJ0toctFkZL6hNP8eu9galcyG9lPbAMQ15w8r8IQAQABAcI1ACq1213qPocimExrQ
    fXJ4nMqNdyK4jQCX6o0u4e6kk532hsLshsLe8a1IA1bNvOoBv0u8G
    64Vwr99bc3/c79VZ62/Us2bHNo+Qg824qawfSQBkZSEU11o7YsPofaalgX9S
    FGO6eCVC8h63Lk3jalDxXEpETYrEyu393MCFPaA2Z2rPxRm9S9yfWb58c08z5
    Ff45/jb/hJ+m1AVyVw3q0plks57hc4Q7yW/2oRyRq2md1K1790OLwtekWD0Vg
    qmoaraeh2TgWnA0bso5x3/LjVxKtIq778fW6jgYqkbbr8d0DKozj9r1tG60YKD01D
    ns3ZPACEcYgEAOqf/fxtDsPnGiaLZ2/hhewtzaxj/WeoVBECb6/Y4R0tauxn8B
    16Fcv1jcpWOUT7cmzv58m5kGWEObDEIAe611vm/QetLM5+uApDs/TchKXLYe
    92onXmq73p4FUXEkMaya81pzu/TdMs50uaf+ma4EuXMY51xgqCnyEcYgEAI4X
    4FjCI740sA3C8Kh7a2U4TL2DNIP6EG6SK9j90+0LFlT6et7k2Sf6un8m2zhj+M0KHTIIE
    /gk6Hf2QSwuUJlhfoYEcg398S6Av3gSrSp003j78C802Vp0FNJT2mdmef2/
    ZYX3CkuVo1h9qee7y7teon8AW8ckDGMHPyVLaETCgyBALD0TDPgDr7ly1MIdm1gh
    FFG04etk/T7BY1X1tgj8j1KhtlhefZp4qk4W7Yuhyjz5a4Q4xy0hJfp7Re7C3Jrd
    +r2o3h5ymuEkqmupL2jyjptMrBWN4g3t4jDaHoq6Q+BdCZzyN9mInp5N1s7e
    fuEM1d4g6d3H0e9s7V0AABKQ8XRDrU3mLTx9c0RKNByeRgLLJiLQlBQXtY181T
    ZufiGWLW8+PC1AMkpoxDrWqwcpiuiRi2ElBfPao3o2gEpeAY/LJscd/j5j6ecu8/
    W3J3OnjnpKora4sfo57BPv5m6THYoI1F5epey/o/sa13hrNriqkJh/+mSUHYF5DRh
    a4Ek4QkBfCO1FeVrv+MNwuaexD61NCbB8hf6v23W8AT8Yy3sW9Yfnv4J4lh/b1
    mocWy1iZkprhA8nMUC790FPeyqB3parQ3s5X7j191bhTkmQxe63fyqjyhX35XR1
    JSKtm6E2c0m6soZisj8B7COKHR32w0U5CgaInQxj1dbGwsh5Opn
  crt:  |

--- BEGIN CERTIFICATE ---
MIID2DCCaC6gwu1389gbQdFFytE4dKjz2iyljJ1TANBqkgixkI9gW0BAPwFADBX
M9rgWqfVDQWGEKWE5h9lBAP6cmdb16XKZpR2b24xCaAYBqVBAzA1anBMBs44LAVYDVQD
EYeNe5BPCcmdb16XLKzR2b24qQVyd1maWNhduGv0Q9yX5R5MX4D1txIDxMDQy
MDADQ08Q9A6xTiXDMQ90A6xMDaMQOQ5hDFEmEDM8QQGAA1UEawNk15a3gCtnHrJ BN
ZDDCAs1wQyJKo2IhvcqABEABQOAgqEPDCCAAQcggEEBANXzJC0wciAyIX1FM2W
TS/4vkx5fQYUEBBXzucue40j7pH1B3XrEkvMdh1/Cq1G1Cvph+pm2hN6MEWwElN
C4QMF5xQvMZZsuUfHfoYEG3tQ98S6Av3gSrSp003j7BE802Vp0FNTJ2mdmef2/
ZYX3CkuV0Qh9qee7y7teon8AW8ckDGMHPyVLaETCgyBALD0TDPgDr7ly1MIdm1gh

--- END CERTIFICATE ---
### 4.5.4 Configurable Parameters

To enable MTLS, make changes to the following parameters.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fep.usePodName</td>
<td>True</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.patroni.tls.certificateName</td>
<td>&lt;secret-name&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.patroni.tls.caName</td>
<td>&lt;configmap-name&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.postgres.tls.certificateName</td>
<td>&lt;secret-name&gt;</td>
<td>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.</td>
</tr>
</tbody>
</table>

-----BEGIN CERTIFICATE-----
MIIDXDCCAkSgAwIBAgIRAMPzF3MTkgFAMbGjMCqQvQwDQYJKoZIhvcNAQELBQAw
VzEYMXcKMA0GA1UEChMVMCMQ0EwCPBgNVHSAEBmAdgZDAtMQswCQYDVQQGEwJF
Q0ExCAYDVQQHEwJFQ0ExMwYwJAYDVQQHEwJFQ0ExMw0xIzAFBgNVHSMEHGw0h
GjQGCCsGAQUFBwIBBhkgGjQRBlkBLkBUUjA9QCSCek631Jm8e8G4qXszW54a1pv
mDkKNdFJ0UzimfAc7Ze+uniOzYfJ3RJYu+xi0TPk11G0XPhQgsQhehz4Kepc8Q
jg0Jum1oByX2Z8Sb1o9rNmFuVUL5BovcVoZv2sRgXjlWJLbFr5h8N2xf2dbvKw+D
K41f9O094G6a127tuJierzNC3nAgBAAA6G1ha9AhA4A1UdWwEB/wQEAwICcDAP
HRMBAfFEBTADAQH/MA0GCSqGSIb3DQEBCwUA4IBAQM0Cn35/COT4uI24ewKk
r6mANBFPM9u6MBj80U62hGqBvCJuUzZI1sQaFX6ZJ2177T5y52gEK5A
QgsB8j3p/vJ5DJ7v0698+1Q19B9K3smQdksAim19FbqzB7J4zK/+8aqj/q2IFv
Jk3ekwQ3xfgkk1BBUv76glr1v0uy1PtPffPflfcGZ0616m6mgbajenXo1RIPx
0+zyCS8dK9tDuP1ruwwXCFMYw9TPbxxK1t77i9qRJqg1LnxWJDzMr0YOCCn+sDm
cxenV91r89qZ0X1YUzYRkka5N4dhIhrzTAlNd05gzynXGz67u/1efsz11K92cdE3
-----END CERTIFICATE-----
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fep.postgres.tls.caName</td>
<td>&lt;configmap-name&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.postgres.tls.privateKeyPassword</td>
<td>&lt;secret-name&gt;</td>
<td>Name of Kubernetes secret that contains the password for the private key for Postgres Server.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName</td>
<td>&lt;secret-name&gt;</td>
<td>Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for &quot;postgres&quot; 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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdminTls.caName</td>
<td>&lt;configmap-name&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdminTls.sslMode</td>
<td>verify-full</td>
<td>For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls.certificateNa me</td>
<td>&lt;secret-name&gt;</td>
<td>Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for &quot;repluser&quot; 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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls.caName</td>
<td>&lt;configmap-name&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls.sslMode</td>
<td>verify-full</td>
<td>For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTls.certificateN ame</td>
<td>&lt;secret-name&gt;</td>
<td>Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for &quot;rewinduser&quot; 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.</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
<td>Details</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTls.caName</td>
<td><code>&lt;configmap-name&gt;</code></td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode</td>
<td>verify-full</td>
<td>For MTLS, this value must be set to verify-full. If only TLS is required, this can be set to verify-ca or prefer.</td>
</tr>
</tbody>
</table>

It is also required to customize `pg_hba.conf` to perform MTLS. Below are two possible settings.

| spec.fep.customPgHba | hostssl 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 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.

   ```yaml
   spec:
     fep:
       forceSel: true
       replicationSlots:
         myslot1:
           type: logical
           database: db1
           plugin: pgoutput
         myslot2:
           type: logical
           database: db1
           plugin: pgoutput
         podAntiAffinity: false
   ```

   b. Add section postgres under spec.fep as shown below.

   ```yaml
   postgres:
     tls:
       caName: cacert
       certificateName: my-fep-cert
   ```

   c. Change the value of wal_level parameter under spec.fepChildCrVal.customPgParams from replica to logical.

   ```yaml
   wal_level = logical
   ```

   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.

   ```yaml
   [rule]
   customPgHba:
     hosts1 all all $SubClusterName-prim-svc.$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

4. Connect to the pod terminal of the Publisher FEPCluster and then connect to the postgres database as shown below.

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.

6. At the Publisher side, create a publication and alter the publication to add the tables that need to be replicated.
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 -l /tmp/custom_certs/
```

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:

```
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);
```

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);
```

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
- **password** = password for the role
- **dbname** = database which contains the tables to be replicated

4.7 FEP Logging

FEPCluster generates log files over the lifetime of execution. 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 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 aggregate 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 monitoring the FEP server log file 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
```
Below is the list of all parameters defined in the `fepLogging` section, along with their brief description

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Required/Optional</th>
<th>Change Effect</th>
<th>Updating value allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>spec.fepLogging.image.image</code></td>
<td>Optional</td>
<td>Fluentd Image of FEPLogging</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.image.pullPolicy</code></td>
<td>Required</td>
<td>Fluentd Image pull policy of FEPLogging</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.limits.cpu</code></td>
<td>Required</td>
<td>Max CPU allocated to fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.limits.memory</code></td>
<td>Required</td>
<td>Max memory allocated to fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.requests.cpu</code></td>
<td>Required</td>
<td>CPU allocation at start for fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.requests.memory</code></td>
<td>Required</td>
<td>Memory allocation at start for fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.sysExtraLogging</code></td>
<td>Required</td>
<td>To turn on extra debugging messages for operator, set value to true. It can be turned on/off at any time</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.restartRequired</code></td>
<td>Required</td>
<td>To restart FEPLogging instance for applying any new configuration for example after certificate rotation</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.scrapeInterval</code></td>
<td>Optional</td>
<td>Scrape interval for Prometheus to fetch metrics from FEPLogging instance</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.scrapeTimeout</code></td>
<td>Optional</td>
<td>Scrape Timeout for Prometheus to fetch metrics from FEPLogging instance</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.host</code></td>
<td>Optional</td>
<td>Target Elasticsearch host name</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.port</code></td>
<td>Optional</td>
<td>Target Elasticsearch port number</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.authSecret.secretName</code></td>
<td>Optional</td>
<td>Secret name which contains Elasticsearch authentication username &amp; password</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.authSecret.userName</code></td>
<td>Optional</td>
<td>Username key specified in Elasticsearch authentication secret</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.authSecret.passwordKey</code></td>
<td>Optional</td>
<td>Password key specified in Elasticsearch authentication secret</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.logstashPrefix</code></td>
<td>Optional</td>
<td>Logstash prefix to differentiate index pattern in elastic search. Default value is postgres</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.scheme</code></td>
<td>Optional</td>
<td>Connection scheme between FEPLogging &amp; Elasticsearch. Possible options http &amp; https</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.sslVerify</code></td>
<td>Optional</td>
<td>Set to true if you want to verify ssl certificate. If set to false then will not consider TLS certificate</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.tls.certificateName</code></td>
<td>Optional</td>
<td>Kubernetes secret name which holds fluentd certificate</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.tls.caName</code></td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Elasticsearch to verify Elasticsearch TLS connection</td>
<td>Yes</td>
</tr>
<tr>
<td><code>spec.fepLogging.tls.certificateName</code></td>
<td>Optional</td>
<td>Kubernetes secret name which holds Fluentd certificate</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Custom Resource Spec

<table>
<thead>
<tr>
<th>Custom Resource Spec</th>
<th>Required/Optional</th>
<th>Change Effect</th>
<th>Updating value allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepLogging.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.prometheus.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds Prometheus certificate</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.prometheus.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
<td>Yes</td>
</tr>
</tbody>
</table>

#### 4.7.1.1 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)

```yaml
spec:
  fepLogging:
    image:
      image: 'quay.io/fujitsu/fujitsu-enterprise-postgres-14-fluentbit:ubi814-0.0'
      pullPolicy: IfNotPresent
```

#### 4.7.1.2 Define fepLogging mcSpec

FEPLogging container Memory & CPU configuration can be provided by mcSpec properties.

Example)

```yaml
spec:
  fepLogging:
    mcSpec:
      limits:
        cpu: 500m
        memory: 700Mi
      requests:
        cpu: 200m
        memory: 512Mi
```

#### 4.7.1.3 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)

```yaml
spec:
  fepLogging:
    restartRequired: true
```

#### 4.7.1.4 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.
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
- scheme
- sslVerify
- tls(if sslVerify set to true)

Configure Elasticsearch server and use its 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)

```yaml
spec:
  fepLogging:
    elastic:
      authSecret:
        passwordKey: password
        secretName: elastic-auth
        userKey: username
      host: elastic-passthrough.apps.openshift.com
      logstashPrefix: 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)

```yaml
kind: Secret
apiVersion: v1
metadata:
  name: elastic-auth
  namespace: my-namespace
```
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)

```yaml
spec:
  fepLogging:
    tls:
      certificateName: fluentd-cert
      caName: cacert
```

4.7.1.8 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)

```yaml
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:

```yaml
spec:
  fep
    ...
  remoteLogging:
    enable: true
    fluentdName: new-fep-logging
    tls:
      certificateName: fluentbit-cert
```
Below is the list of all parameters defined in the remoteLogging section, along with their brief description:

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Required/Optional</th>
<th>Change Effect</th>
<th>Updating value allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>remoteLogging.enable</td>
<td>Required</td>
<td>The ‘enable’ is set to true for enabling Logging feature</td>
<td>No</td>
</tr>
<tr>
<td>remoteLogging.fluentdName</td>
<td>Required</td>
<td>The ‘fluentdName’ is the name of the FEPLogging CR where logs will be forwarded</td>
<td>Yes</td>
</tr>
<tr>
<td>remoteLogging.tls.secretName</td>
<td>Optional</td>
<td>Secret name which contains MTLS certs of fluentbit</td>
<td>No</td>
</tr>
<tr>
<td>remoteLogging.tls.caName</td>
<td>Optional</td>
<td>Cacert of Fluentd for ssl verification</td>
<td>No</td>
</tr>
</tbody>
</table>

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)

```yaml
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)

```yaml
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)
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 logs to that FEPLogging instance, those FEP logs will be visible on Elasticsearch stack or Elastic Cloud. Assuming Elasticsearch has been configured with Kibana then logs will be visible to Kibana Dashboard. User can use fep log csv fields to create various Dashboard in Kiabana as well. LogstashPrefix value 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 severity 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.

4.7.4 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

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Change Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgBadger.schedules.create</td>
<td>The ‘create’ schedule to create report and upload it to endpoint</td>
</tr>
<tr>
<td>pgBadger.schedules.cleanup</td>
<td>The ‘cleanup’ schedule to delete the report left in container</td>
</tr>
<tr>
<td>pgBadger.options.incremental</td>
<td>Default: false; When set to True: create incremental report in pgbadger</td>
</tr>
<tr>
<td>pgBadger.endpoint.authentication</td>
<td>a secret to contain authentication info to access endpoint support basic auth only</td>
</tr>
<tr>
<td>pgBadger.endpoint.customCertificateName</td>
<td>Client certificate reference in customCertificate CR</td>
</tr>
<tr>
<td>pgBadger.endpoint.fileUploadParameter</td>
<td>The file upload parameter defined by the web server</td>
</tr>
<tr>
<td>Custom Resource spec</td>
<td>Change Effect</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>Default: 'file'</td>
<td></td>
</tr>
<tr>
<td>pgBadger.endpoint.insecure</td>
<td>equivalent to curl -insecure option, default to false</td>
</tr>
<tr>
<td>pgBadger.endpoint.url</td>
<td>Web server url to upload the report file</td>
</tr>
</tbody>
</table>

### 4.8.2 Define pgBadger Schedules

The schedules are used to create and run a job periodically, written in Cron format.

If the schedule format is invalid, the cronjob will not be created, so no pgBadger report will be created and uploaded.

**Example**

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

### 4.8.3 Define pgBadger Options

When the incremental option is set to false, pgBadger will create normal html report and upload the html file to the web server.

When the incremental option is set to true, pgBadger will create incremental report and upload a zip file to the web server.

**Example**

```yaml
pgBadger:
  options:
    incremental: true
```

### 4.8.4 Define Endpoint for Uploading Report

**Web server url**

Both http and https are supported.

**Example**

```yaml
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**

```bash
$ echo -ne "myuser:mypass" | base64
amFzb2530mphc29udw==
```

Wrap the output with base64 for creating a secret

**Example**

```bash
$ echo -ne "amFzb2530mphc29udw==" | base64
```
Create a secret by using the wrapped text. The key must be 'basic_auth'.

Example:

```yaml
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:

```yaml
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:

```bash
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:

```bash
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:

```yaml
spec:
  fepChildCrVal:
    customCertificates:
    - userName: pgbadger-custom
certificateName: webserver-cert
caName: webserver-cacert
```
Refer the custom certificate name in pgbadger endpoint.

Example)

```yaml
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)

```yaml
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)

```yaml
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 endpoint section will be converted to curl command parameters. The following table shows the mapping:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>User configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>[URL]</td>
<td>Endpoint url</td>
</tr>
<tr>
<td>--cert</td>
<td>webserver.pem</td>
</tr>
<tr>
<td></td>
<td>included in the secret referred in customCertificateName</td>
</tr>
<tr>
<td>--key</td>
<td>webserver.key</td>
</tr>
<tr>
<td></td>
<td>included in the secret referred in customCertificateName</td>
</tr>
<tr>
<td>--cacert</td>
<td>webca.pem</td>
</tr>
<tr>
<td></td>
<td>included in the configmap referred in customCertificateName</td>
</tr>
<tr>
<td>--form “uploadfile=@/path/to/report”</td>
<td>Endpoint fileUploadParameter</td>
</tr>
<tr>
<td>--header “Authorization: Basic passxxx”</td>
<td>Endpoint authentication configmap</td>
</tr>
<tr>
<td>--insecure</td>
<td>When endpoint.insecure is set to true</td>
</tr>
</tbody>
</table>

### 4.8.5 Uploaded File on Web Server

The FEP cluster uploads the pgbadger report according to the incremental mode:

<table>
<thead>
<tr>
<th>Incremental mode</th>
<th>Uploaded file name</th>
<th>Example</th>
</tr>
</thead>
</table>
| 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  |
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.
Chapter 5 Post-Deployment Operations

This chapter describes the operation after deploying the container.

5.1 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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>.metadata.name</td>
<td>Name of Cluster</td>
</tr>
<tr>
<td>AGE</td>
<td>Elapsed time</td>
<td>Indicates the amount of time that has elapsed since the cluster was created</td>
</tr>
</tbody>
</table>

Example)

# kubectl get fepclusters -A

<table>
<thead>
<tr>
<th>NAMESPACE</th>
<th>NAME</th>
<th>AGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace1</td>
<td>ns1fep1</td>
<td>21h</td>
</tr>
<tr>
<td>namespace2</td>
<td>ns2fep2</td>
<td>22h</td>
</tr>
</tbody>
</table>

Update FEPCluster

Equivalent Kubernetes command: kubectl apply -f <new_spec>

Operations that can be performed here.

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Change effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>.spec.fep.instances: n</td>
<td>Increase the number of nodes in the cluster to n.</td>
</tr>
</tbody>
</table>

This will impact behaviour for values in fep section only.

All other values except spec.fepChildCrVal.backup.image.image must be changed from respective child CR.

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, FEPCfg, FEPCert & FEPUser) & resources associated with it.

Note

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.
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 the "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 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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>.metadata.name</td>
<td>Name of pgpool2</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Change Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>.spec.count: n</td>
<td>Increase the number of nodes in the cluster to n.</td>
</tr>
<tr>
<td>.spec.serviceport</td>
<td>Change the TCP port for connecting to the Pgpool-II.</td>
</tr>
<tr>
<td>.spec.statusport</td>
<td>Change the TCP port for connecting to the PCP process.</td>
</tr>
<tr>
<td>.spec.limits.cpu</td>
<td>Change limits of cpus.</td>
</tr>
<tr>
<td>.spec.limits.memory</td>
<td>Change limits of memory.</td>
</tr>
<tr>
<td>.spec.requests.cpu</td>
<td>Change requests of cpus.</td>
</tr>
<tr>
<td>.spec.requests.memory</td>
<td>Change requests of memory.</td>
</tr>
<tr>
<td>.spec.fepclustername</td>
<td>Change fepcluster to connect.</td>
</tr>
<tr>
<td>.spec.customhba</td>
<td>Change pool_hba.conf file.</td>
</tr>
<tr>
<td>.spec.custompcp</td>
<td>Change pcp.conf file.</td>
</tr>
<tr>
<td>.spec.customsslkey</td>
<td>Change key content</td>
</tr>
<tr>
<td>.spec.customsslcert</td>
<td>Change the contents of the public x 509 certificate.</td>
</tr>
<tr>
<td>.spec.customsslcacert</td>
<td>Change the contents of the CA root certificate in PEM format.</td>
</tr>
</tbody>
</table>

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:

```yaml
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.

5.3 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.
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.

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Change Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.schedule.num</td>
<td>Change the Number of Registered Backup Schedules</td>
</tr>
<tr>
<td>spec.scheduleN.schedule</td>
<td>Change the scheduled backup time</td>
</tr>
<tr>
<td>spec.scheduleN.type</td>
<td>Change the scheduled backup type</td>
</tr>
<tr>
<td>spec.pgBackrestParams</td>
<td>Change pgBackRest parameters</td>
</tr>
</tbody>
</table>
| spec.scheduleN.repo        | If you specified more than one repository for spec.pgBackrestParams, select the repository in which to store the backup data.  
                             | The default is 1.                                                            |

- Changes made during the backup are reflected from the next backup.
- Changes to the backup schedule do not affect the application.
- If you perform any of the following update operations, be sure to obtain a backup after the update.
  - When the master encryption key is updated with pgx_set_master_key
  - When the encryption passphrase for transparent data encryption is updated (can be updated by the tdeppassphrase parameter of FEPCluster CR)
5.4 Configure MTLS Setting

5.4.1 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.

5.5 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.
5.5.1 Monitoring FEP Operator and Operands

The monitoring of FEP operators and operands are achieved by Prometheus' standard alive and resource monitoring.

<table>
<thead>
<tr>
<th>Metrics name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alive monitoring</td>
<td>Can monitor Pod status</td>
</tr>
<tr>
<td>Resource monitoring</td>
<td>You can monitor the following resource status</td>
</tr>
<tr>
<td></td>
<td>- CPU Usage</td>
</tr>
<tr>
<td></td>
<td>- CPU Quota</td>
</tr>
<tr>
<td></td>
<td>- Memory Usage</td>
</tr>
<tr>
<td></td>
<td>- Memory Quota</td>
</tr>
<tr>
<td></td>
<td>- Current Network Usage</td>
</tr>
<tr>
<td></td>
<td>- Receive Bandwidth</td>
</tr>
<tr>
<td></td>
<td>- Transmit Bandwidth</td>
</tr>
<tr>
<td></td>
<td>- Rate of Received Packets</td>
</tr>
<tr>
<td></td>
<td>- Rate of Transmitted Packets</td>
</tr>
<tr>
<td></td>
<td>- Rate of Received Packets Dropped</td>
</tr>
<tr>
<td></td>
<td>- Rate of Transmitted Packets Dropped</td>
</tr>
</tbody>
</table>

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 Catalog page to see which version you are currently using, which can be updated, and to check for security vulnerabilities.

5.5.2 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 metrics 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.

5.5.2.1 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 metrics 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.

5.5.2.2 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

<table>
<thead>
<tr>
<th>Type</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default mandatory</td>
<td>Are collected by FEP Exporter. These are kept enabled by default by FEP Exporter and can not be disabled by end user.</td>
</tr>
<tr>
<td>Default useful</td>
<td>Useful focused metrics for health and performance metrics. Can be disabled by end user.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Metrics name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>pg_stat_bgwriter_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_stat_database_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_stat_database_conflicts_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_stat_archiver_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_stat_activity_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_stat_replication_*</td>
<td>Maps to view in Statistic Collector</td>
</tr>
<tr>
<td>pg_replication_slots_*</td>
<td>Maps to System Catalog pg_replication_slots</td>
</tr>
<tr>
<td>pg_settings_*</td>
<td>Maps to System Catalog pg_settings</td>
</tr>
<tr>
<td>pg_locks_*</td>
<td>Maps to System Catalog pg_locks</td>
</tr>
<tr>
<td>pg_exporter_*</td>
<td>Exposes exporter metrics:</td>
</tr>
<tr>
<td></td>
<td>- last_scrape_duration_seconds (Duration of the last scrape of metrics from PostgresSQL)</td>
</tr>
<tr>
<td></td>
<td>- scrapes_total (Total number of times PostgresSQL was scraped for metrics)</td>
</tr>
<tr>
<td></td>
<td>last_scrape_error (Whether the last scrape of metrics from PostgresSQL resulted in an error; 1 for error &amp; 0 for success)</td>
</tr>
<tr>
<td>pg_*</td>
<td>Exposes exporter metrics</td>
</tr>
<tr>
<td></td>
<td>- pg_up ( set to 1 if the connection to service is success, 0 otherwise )</td>
</tr>
<tr>
<td></td>
<td>- pg_static ( can be used to fetch label short_version / version containing postgres server version information )</td>
</tr>
</tbody>
</table>

Default useful metrics

There are certain useful queries which are additionally added to evaluate the health of the Database system.

<table>
<thead>
<tr>
<th>Metrics name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>pg_capacity_connection_*</td>
<td>Metrics on connections e.g. txns running for 1 hour</td>
</tr>
<tr>
<td>pg_capacity_schema_*</td>
<td>Metrics on disk space of schema</td>
</tr>
<tr>
<td>pg_capacity_tblspace_*</td>
<td>Metrics on disk space of tablespace</td>
</tr>
<tr>
<td>pg_capacity_tblvacuum_*</td>
<td>Metrics on tables without vacuum for days</td>
</tr>
<tr>
<td>pg_capacity_longtx_*</td>
<td>Number of transactions running longer than 5 minutes</td>
</tr>
<tr>
<td></td>
<td>Review the information and consider SQL tuning and resource enhancements.</td>
</tr>
<tr>
<td>pg_performance_locking_detail_*</td>
<td>Details of processes in blocked state</td>
</tr>
<tr>
<td>pg_performance_locking_*</td>
<td>Number of processes in blocked state</td>
</tr>
<tr>
<td>pg_replication_*</td>
<td>Replication lag behind master in seconds</td>
</tr>
<tr>
<td></td>
<td>Provides the ability to check for the most current data in a reference replica</td>
</tr>
<tr>
<td></td>
<td>To solve the problem, it is necessary to consider measures such as increasing network resources and reducing the load</td>
</tr>
<tr>
<td>pg_postmaster_*</td>
<td>Time at which postmaster started</td>
</tr>
<tr>
<td>pg_stat_user_tables_*</td>
<td>Important statistics from pg_stat_user_tables</td>
</tr>
<tr>
<td>Metrics name</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>pg_statio_user_tables_*</td>
<td>Important statistics from pg_statio_user_tables</td>
</tr>
<tr>
<td>pg_database_*</td>
<td>Database size</td>
</tr>
<tr>
<td></td>
<td>If the database runs out of space, database restore is required</td>
</tr>
<tr>
<td>pg_stat_statements_*</td>
<td>Statistics of SQL statements executed by server</td>
</tr>
<tr>
<td>pg_capacity_tblbloat_*</td>
<td>Fetched bloat in tables</td>
</tr>
</tbody>
</table>

**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.

5.5.2.3 Default Alerts

There are few basic alert rules which are setup by the FEP Operator as below

<table>
<thead>
<tr>
<th>Alert rule</th>
<th>Alert Level</th>
<th>Condition persistence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContainerHighCPUUsage</td>
<td>Warning</td>
<td>5 mins</td>
<td>FEP server container/Pod CPU usage is exceeding 80% of the resource limits</td>
</tr>
<tr>
<td>ContainerHighRAMUsage</td>
<td>Warning</td>
<td>30 mins</td>
<td>FEP server container/Pod memory usage is exceeding 80% of the resource limits</td>
</tr>
<tr>
<td>PVCCLowDiskSpace</td>
<td>Warning</td>
<td>5 mins</td>
<td>A FEP PVC (volume) has less than 10% disk available</td>
</tr>
<tr>
<td>ContainerDisappeared</td>
<td>Warning</td>
<td>60 seconds</td>
<td>FEP server container/Pod has disappeared since last 60 seconds</td>
</tr>
<tr>
<td>PostgresqlDown</td>
<td>Error</td>
<td>-</td>
<td>FEP server apparently went down or not accessible</td>
</tr>
<tr>
<td>PostgresqlTooManyConnections</td>
<td>Warning</td>
<td>-</td>
<td>FEP server container/Pod connection usage is beyond 90% of its available capacity</td>
</tr>
</tbody>
</table>

**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.

5.5.2.4 Graphical user interface

User can build their custom dashboard using default and custom metrics.

An example Grafana dashboard screenshot is shown below

![Grafana Dashboard Screenshot]

5.5.3 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>
<thead>
<tr>
<th>Table/View name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgbackrest_info_backup</td>
<td>Backup Processing Status</td>
</tr>
</tbody>
</table>

5.5.3.1 pgbackrest_info_backup view

Contains one line per backup for information about the state of the backup.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>text</td>
<td>Information identifying the backup</td>
</tr>
<tr>
<td>type</td>
<td>text</td>
<td>full: full backup, incr: incremental backup</td>
</tr>
<tr>
<td>prior</td>
<td>text</td>
<td>Label of the backup that should be applied first (For incremental backups only)</td>
</tr>
<tr>
<td>database_size</td>
<td>bigint</td>
<td>Database size</td>
</tr>
<tr>
<td>database_size_comp</td>
<td>bigint</td>
<td>Database size (After Compression)</td>
</tr>
<tr>
<td>backup_size</td>
<td>bigint</td>
<td>Backup size</td>
</tr>
<tr>
<td>backup_size_comp</td>
<td>bigint</td>
<td>Backup size (After Compression)</td>
</tr>
<tr>
<td>archive_start</td>
<td>text</td>
<td>Range of WALs required for restore (Start)</td>
</tr>
<tr>
<td>archive_stop</td>
<td>text</td>
<td>Range of WALs required for restore (End)</td>
</tr>
</tbody>
</table>
### 5.5.4 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 fepExporter schema in the postgres database on the FEP server.

<table>
<thead>
<tr>
<th>Table/View name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgpool2_stat_load_balance</td>
<td>Load Balance Information in pgpool2</td>
</tr>
<tr>
<td>pgcluster_stat_replication</td>
<td>Replication State</td>
</tr>
<tr>
<td>pgpool2_stat_conn_pool</td>
<td>Connection Pool State for pgpool2</td>
</tr>
<tr>
<td>pgpool2_stat_sql_command</td>
<td>SQL Command Statistics</td>
</tr>
</tbody>
</table>

#### 5.5.4.1 pgpool2_stat_load_balance view

Contains one row for MasterService and one row for ReplicaService.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_id</td>
<td>integer</td>
<td>database node id (0 or 1)</td>
</tr>
<tr>
<td>status</td>
<td>text</td>
<td>status (up or down)</td>
</tr>
<tr>
<td>lb_weight</td>
<td>double precision</td>
<td>load-balancing weight</td>
</tr>
<tr>
<td>role</td>
<td>text</td>
<td>role (primary or standby)</td>
</tr>
<tr>
<td>last_status_change</td>
<td>timestamp with time zone</td>
<td>last status change time</td>
</tr>
</tbody>
</table>

#### 5.5.4.2 pgpool2_stat_conn_pool view

Indicates the state of the connection pool. Contains connection pool information for each pgpool2 instance.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgpool2_node_id</td>
<td>integer</td>
<td>pgpool2 node id (0 - the number of pgpool2 instance -1)</td>
</tr>
<tr>
<td>pool_pid</td>
<td>integer</td>
<td>The PID of the displayed Pgpool-II process</td>
</tr>
<tr>
<td>start_time</td>
<td>timestamp with timezone</td>
<td>The timestamp of when this process was launched</td>
</tr>
<tr>
<td>pool_id</td>
<td>integer</td>
<td>The pool identifier (should be between 0 and max_pool - 1)</td>
</tr>
<tr>
<td>backend_id</td>
<td>integer</td>
<td>The backend identifier (should be between 0 and the number of configured backends minus one)</td>
</tr>
<tr>
<td>role</td>
<td>text</td>
<td>role (primary or standby)</td>
</tr>
<tr>
<td>database</td>
<td>text</td>
<td>The database name for this process's pool id connection</td>
</tr>
<tr>
<td>username</td>
<td>text</td>
<td>The user name for this process's pool id connection</td>
</tr>
</tbody>
</table>
### Column | Type | Description
--- | --- | ---
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

#### 5.5.4.3 pgpool2_stat_sql_command view

Represents SQL command statistics.

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_id</td>
<td>integer</td>
<td>The backend identifier (should be between 0 and the number of configured backends minus one)</td>
</tr>
<tr>
<td>role</td>
<td>text</td>
<td>role (primary or standby)</td>
</tr>
<tr>
<td>select_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: SELECT</td>
</tr>
<tr>
<td>insert_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: INSERT</td>
</tr>
<tr>
<td>update_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: UPDATE</td>
</tr>
<tr>
<td>delete_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: DELETE</td>
</tr>
<tr>
<td>ddl_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: DDL</td>
</tr>
<tr>
<td>other_cnt</td>
<td>integer</td>
<td>The numbers of SQL command: others</td>
</tr>
<tr>
<td>panic_cnt</td>
<td>integer</td>
<td>The numbers of failed commands</td>
</tr>
<tr>
<td>fatal_cnt</td>
<td>integer</td>
<td>The numbers of failed commands</td>
</tr>
<tr>
<td>error_cnt</td>
<td>integer</td>
<td>The numbers of failed commands</td>
</tr>
</tbody>
</table>

#### 5.6 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.

**5.6.1 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.

### 5.6.2 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,

<table>
<thead>
<tr>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEPCert CR creation successful</td>
</tr>
<tr>
<td>FEPCert CR creation failed</td>
</tr>
<tr>
<td>FEPConfig CR creation successful</td>
</tr>
<tr>
<td>FEPConfig CR creation failed</td>
</tr>
<tr>
<td>FEPVolume CR creation successful</td>
</tr>
<tr>
<td>FEPVolume CR creation failed</td>
</tr>
<tr>
<td>FEPBackup cronjob1 creation successful</td>
</tr>
<tr>
<td>FEPBackup cronjob1 creation failed</td>
</tr>
<tr>
<td>FEPBackup cronjob2 creation successful</td>
</tr>
<tr>
<td>FEPBackup cronjob2 creation failed</td>
</tr>
<tr>
<td>FEPBackup cronjob3 creation successful</td>
</tr>
<tr>
<td>FEPBackup cronjob3 creation failed</td>
</tr>
<tr>
<td>FEPBackup cronjob4 creation successful</td>
</tr>
<tr>
<td>FEPBackup cronjob4 creation failed</td>
</tr>
<tr>
<td>FEPBackup cronjob5 creation successful</td>
</tr>
<tr>
<td>FEPBackup cronjob5 creation failed</td>
</tr>
<tr>
<td>FEPPgPool2 CR creation successful</td>
</tr>
<tr>
<td>FEPPgPool2 CR creation failed</td>
</tr>
<tr>
<td>FEPPgPool2Cert CR creation successful</td>
</tr>
<tr>
<td>FEPPgPool2Cert CR creation failed</td>
</tr>
<tr>
<td>FEPRestore CR creation successful</td>
</tr>
<tr>
<td>FEPRestore CR creation failed</td>
</tr>
</tbody>
</table>

**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.
5.7 Scaling Replicas

5.7.1 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>
```

5.7.2 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.

5.8 Backing Up to Object Storage

Describes how to store backup data in object storage.

5.8.1 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
```
5.8.2 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 “5.8.1 Storing CA Files (Root Certificates)” for caName.

FEPCluster Custom Resource Example: Only Object Storage Used for Backup Repository

```yaml
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=/etc/pki/tls/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

```yaml
... 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=/etc/pki/tls/certs/ca.crt
      pgbackrestKeyParams: |
        repo2-s3-key=SAMPLEKEY
        repo2-s3-key-secret=SAMPLESECRET
      caName: |
        - storage-cacert
  ...
```

5.9 Disaster Recovery

By using OSS (pg_backrest) 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.

### 5.9.1 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.

![Configuration Diagram]

**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.

### 5.9.2 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.

#### 5.9.2.1 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
```

#### 5.9.2.2 Defining a FEPCluster Custom Resource

In addition to the FEPCluster settings, specify the Restore settings below.

FEPCluster Custom Resource Example

```yaml
apiVersion: fep.fujitsu.io/v1
kind: FEPCluster
metadata:
  ...
spec:
  fepChildCrVal:
    restore:
      pgbackrestParams: |
        repo1-type=s3
        repo1-path=/backup/cluster1
```
### Setting value

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepChildCrVal.restore</td>
<td></td>
<td>Define when restoring by specifying the backup data stored in the object storage.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.pgbackrestParams</td>
<td></td>
<td>Optional &quot;</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.pgbackrestKeyParams</td>
<td></td>
<td>Optional &quot;</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.caName</td>
<td></td>
<td>Optional Specify when you use a CA file other than the system default. Specify the name of the created ConfigMap in list format. The specified ConfigMap will be mounted in /pgbackrest/storage-certs.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.mcspec.limits</td>
<td>cpu: 200m memory: 300Mi</td>
<td>Optional CPU and memory allocated to the container performing the restore.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.mcspec.requests</td>
<td>cpu: 100m memory: 200Mi</td>
<td>Optional CPU and memory allocated to the container performing the restore.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoretype</td>
<td>latest</td>
<td>Optional Restore Type (latest or PITR)</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoredate</td>
<td></td>
<td>Optional Specify the date to restore when spec.fepChildCrVal.restorerestoretype is &quot;PITR&quot;.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoretime</td>
<td></td>
<td>Optional Specify the time to restore when spec.fepChildCrVal.restorerestoretype is &quot;PITR&quot;.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.image</td>
<td></td>
<td>Optional Image of the container to perform the restore.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.imagePullPolicy</td>
<td>IfNotPresent</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is omitted by default. In this case, the URL for image is obtained from the operator container environment.</td>
</tr>
</tbody>
</table>
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:

```yaml
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:

```yaml
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.
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:

```yaml
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"**

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
</table>
| spec.fepChildCrVal.upgrade|         | Optional
<p>|                           |         | When this field is defined, a major version upgrade is performed. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepChildCrVal.upgrade.sourceCluster</td>
<td></td>
<td>Specify the FEPCluster CR name of the data migration source. Be sure to specify spec.fepChildCrVal.upgrade when defining it.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.mcSpec.limits</td>
<td>cpu: 200m memory: 300Mi</td>
<td>Optional Specify the maximum number of resources allocated to the upgrade execution container.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.mcSpec.requests</td>
<td>cpu: 100m memory: 200Mi</td>
<td>Optional Specify the lower limit of resources allocated to the upgrade execution container.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.image</td>
<td></td>
<td>Optional If omitted, the URL of the image is obtained from the operator container environment.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.imagePullPolicy</td>
<td>IfNotPresent</td>
<td>Optional Specify the pull policy for the container image.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Always</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IfNotPresent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Never</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.source.pgAdminTls.certificateName</td>
<td></td>
<td>Optional If the data source FEPCluster used &quot;cert&quot; 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 &quot;postgres&quot; in the data source. Refer to &quot;4.5.1 Manual Certificate Management&quot; for information about creating secrets.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.destination.pgAdminTls.certificateName</td>
<td></td>
<td>Optional No details provided.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.storage</td>
<td>Optional</td>
<td>Defines storage for storing dump files.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.storage.storageClass</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, the default storage class of the operating environment will be used.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.storage.size</td>
<td>2Gi</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify the size of the storage to store the dump file.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.storage.accessModes</td>
<td>ReadWriteOnce</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Storage access mode for storing dump files</td>
</tr>
<tr>
<td></td>
<td></td>
<td>As an array of access modes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>e.g. [ReadWriteMany]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, it is treated as [ReadWriteOnce].</td>
</tr>
</tbody>
</table>

**Note**

Connect to the database and run the following SQL to check the size of the database in advance:

```sql
$ 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.
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
- FEP 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

```bash
$ oc get po
NAME                                     READY   STATUS    RESTARTS   AGE
fep-ansible-operator-7dc5fd9bf7-4 smzk   1/1     Running   0          20m
```

How to check the log

```bash
$ 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 "5.1 Configuration Change" and reapply. Depending on the parameters to be changed, the container may be redeployed. Refer to "5.1 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 don’t allow it to be 100% functionally capable to VM based server instance. The known limitations are as below.

<table>
<thead>
<tr>
<th>No</th>
<th>Limitation</th>
<th>Reason for Limitation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No Support for JIT</td>
<td>Since UBI8 is not having requisite LLVM libraries</td>
<td>It is not possible to enable JIT in postgreSQL.conf. Impact for the customer is that they are not able to achieve maximum performance capabilities on given CPU and memory</td>
</tr>
<tr>
<td>2</td>
<td>FEP parallelism improvements</td>
<td>Since UBI8 is not hosting dstat binaries</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>Crypto Express cards are not supported</td>
<td>IBM LinuxOne doesn’t support CryptoExpress cards in Openshift container platform at this stage.</td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>No Support for Oracle foreign data wrapper</td>
<td>Oracle foreign data wrapper has dependency on Instant Client package, which is not available.</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 annotations as below and then click on Create.
2. Both the Statefulset and its resulting pods will be annotated with your provided 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:
```
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

```yaml
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 -
```
Preface

Purpose of this document
This document is a reference, and explains parameter.

Intended readers
This document is aimed at people who manage and operate.
Readers of this document are also assumed to have general knowledge of:
- Linux
- Kubernetes
- Containers
- Operators

Structure of this document
This document is structured as follows:
Chapter 1 Custom Resource Parameters
Expects the parameter.
Appendix A Default Metrics Queries
Explains the Default Metrics Queries
Appendix B Default Alert Rules
Explains the Default Alert Rules

Abbreviations
The following abbreviations are used in this manual:

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUJITSU Software Enterprise Postgres for Kubernetes</td>
<td>FEP or FEP or FUJITSU Enterprise Postgres</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td></td>
</tr>
<tr>
<td>Transparent Data Encryption</td>
<td>TDE</td>
</tr>
<tr>
<td>Custom Resource</td>
<td>CR</td>
</tr>
<tr>
<td>Custom Resource Definition</td>
<td>CRD</td>
</tr>
<tr>
<td>Persistent Volume</td>
<td>PV</td>
</tr>
</tbody>
</table>

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.
# Contents

**Chapter 1 Custom Resource Parameters**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 FEPCluster Parameter</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Custom Resource Parameters</td>
<td>19</td>
</tr>
<tr>
<td>1.2.1 FEPCluster Custom Resource Parameters</td>
<td>19</td>
</tr>
<tr>
<td>1.2.2 FEP Cluster Configuration</td>
<td>25</td>
</tr>
<tr>
<td>1.2.3 FEPConfig Child Custom Resource Parameters</td>
<td>25</td>
</tr>
<tr>
<td>1.2.4 FEPUser Child Custom Resource Parameters</td>
<td>27</td>
</tr>
<tr>
<td>1.2.5 FEPVolume Child Custom Resource Parameters</td>
<td>29</td>
</tr>
<tr>
<td>1.2.5.1 Create Volumes</td>
<td>29</td>
</tr>
<tr>
<td>1.2.5.2 Delete Volumes</td>
<td>30</td>
</tr>
<tr>
<td>1.2.6 FEP Cert Child Custom Resource Parameters</td>
<td>31</td>
</tr>
<tr>
<td>1.2.6.1 Create/Update Certificates</td>
<td>31</td>
</tr>
<tr>
<td>1.2.6.2 Delete Certificates</td>
<td>31</td>
</tr>
<tr>
<td>1.2.7 FEPBackup Child Custom Resource Parameters</td>
<td>33</td>
</tr>
<tr>
<td>1.2.8 FEP Restore Custom Resource Parameters</td>
<td>33</td>
</tr>
<tr>
<td>1.2.9 FEP Pgpool2 Custom Resource Parameters</td>
<td>36</td>
</tr>
<tr>
<td>1.2.10 FEP Action Custom Resource Parameters</td>
<td>40</td>
</tr>
<tr>
<td>1.2.10.1 FEP Action Specific Operation Details</td>
<td>42</td>
</tr>
<tr>
<td>1.2.11 FEP Exporter Custom Resource</td>
<td>44</td>
</tr>
<tr>
<td>1.2.12 FEP Autoscale Custom Resource</td>
<td>46</td>
</tr>
<tr>
<td>1.2.13 FEP Upgrade Custom Resource</td>
<td>46</td>
</tr>
<tr>
<td>1.2.14 FEP Logging Custom Resource</td>
<td>48</td>
</tr>
<tr>
<td>1.2.15 FEP Custom Resources - spec.fep.pgBadger</td>
<td>49</td>
</tr>
<tr>
<td>Appendix A Default Metrics Queries</td>
<td>51</td>
</tr>
<tr>
<td>Appendix B Default Alert Rules</td>
<td>60</td>
</tr>
</tbody>
</table>
# 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata.name</td>
<td>new-fep</td>
<td>Name for the Cluster. FEP server container will use this value for Patroni scope. e.g. new-fep</td>
</tr>
<tr>
<td>spec.fep.autoPodRestart</td>
<td>&lt;omitted&gt;</td>
<td>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 minimise 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 <code>pod_restart</code> and arguments <code>ALL</code></td>
</tr>
<tr>
<td>spec.fep.fepVersion</td>
<td>&lt;omitted&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.customAnnotation.allDeployments</td>
<td>{} (*)</td>
<td>Contents under this are optional. User can remove {} and add multiple key-value pairs. All of these pair will be</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| spec.fep.image.image  | <omitted> | FEP server container image to be used quay.io/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  
<p>|                        |         | Default - C |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fep.monitoring</td>
<td></td>
<td>This is an Optional section. This defines whether monitoring enabled(true) or disabled(false) , MTLS enabled or disabled &amp; Basic authentication enabled or not</td>
</tr>
<tr>
<td>spec.fep.monitoring.enable</td>
<td>false</td>
<td>If set true, the operator will create FEPExporter with given spec</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter</td>
<td></td>
<td>This is Optional section. Exporter spec section applied only if enable: true</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.authSecret</td>
<td></td>
<td>This is Optional section. Base Authentication secret to provide username &amp; encrypted password of user</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.authSecret.secretName</td>
<td>(created by user)</td>
<td>Mandatory Name of secret that contains username and password</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.authSecret.userKey</td>
<td>(created by user)</td>
<td>Mandatory Key of username in specified secret</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.authSecret.passwordKey</td>
<td>(created by user)</td>
<td>Mandatory Key of password in specified secret</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.tls</td>
<td></td>
<td>This is optional section. FEPExporter MTLS specs. Mandatory if tls specs defined for Prometheus specs</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.tls.certificateName</td>
<td>(created by user)</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.tls.caName</td>
<td>(created by user)</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.monitoring.prometheus</td>
<td></td>
<td>This is Optional section. Prometheus specs are mandatory if tls specs defined for FEPExporter</td>
</tr>
<tr>
<td>spec.fep.monitoring.prometheus.tls</td>
<td></td>
<td>Prometheus MTLS specs</td>
</tr>
<tr>
<td>spec.fep.monitoring.prometheus.tls.certificateName</td>
<td>(created by user)</td>
<td>This is an Optional parameter. These points to Kubernetes TLS secret that contains the certificate of Prometheus. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.</td>
</tr>
<tr>
<td>spec.fep.monitoring.prometheus.tls.caName</td>
<td>(created by user)</td>
<td>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.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fep.podAntiAffinity</td>
<td>false</td>
<td>Defines that all the pods should not run on same worker node</td>
</tr>
<tr>
<td>spec.fep.podDisruptionBudget</td>
<td>false</td>
<td>Allows to maintain minimum number of pods of an application even when some nodes are voluntarily drained for say, maintenance</td>
</tr>
<tr>
<td>spec.fep.replicationSlots</td>
<td></td>
<td>List of Patroni permanent replication slots.</td>
</tr>
<tr>
<td>spec.fep.replicationSlots.demo_subscription1</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.replicationSlots.type</td>
<td>logical</td>
<td>Must be 'logical' for logical replication</td>
</tr>
<tr>
<td>spec.fep.replicationSlots.database</td>
<td>postgres</td>
<td>Specify the database name for logical replication</td>
</tr>
<tr>
<td>spec.fep.replicationSlots.plugin</td>
<td>pgoutput</td>
<td>FEP supports 'pgoutput' by default.</td>
</tr>
<tr>
<td>spec.fep.usePodName</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.patroni.tls.certificateName</td>
<td>(created by user)</td>
<td>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</td>
</tr>
<tr>
<td>spec.fep.patroni.tls.caName</td>
<td>(created by user)</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fep.postgres.tls.certificateName</td>
<td>(created by user)</td>
<td>Optional This points to Kubernetes TLS secret that contains the certificate for Postgres. The certificate itself is stored in the key tls.crt. This field is optional. When this key is set, Operator will ignore the value in systemCertificates</td>
</tr>
<tr>
<td>spec.fep.postgres.tls.caName</td>
<td>(created by user)</td>
<td>Optional This point to Kubernetes configmap that contains additional CA for Postgres to</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fep.postgres.tls.privateKeyPassword</td>
<td>(created by user)</td>
<td>Optional This points to Kubernetes secret that contains the password for the above private key. This field is optional.</td>
</tr>
<tr>
<td>spec.fep.pgBadger.schedules.create</td>
<td></td>
<td>The 'create' schedule to create report and upload it to endpoint</td>
</tr>
<tr>
<td>spec.fep.pgBadger.schedules.cleanup</td>
<td></td>
<td>The 'cleanup' schedule to delete the report left in container</td>
</tr>
<tr>
<td>spec.fep.pgBadger.options.incremental</td>
<td>false</td>
<td>Default: false; When set to True: create incremental report in pgbadger</td>
</tr>
<tr>
<td>spec.fep.pgBadger.endpoint.authentication</td>
<td></td>
<td>a secret to contain authentication info to access endpoint support basic auth only</td>
</tr>
<tr>
<td>spec.fep.pgBadger.endpoint.customCertificateName</td>
<td></td>
<td>Client certificate reference in customCertificate CR</td>
</tr>
<tr>
<td>spec.fep.pgBadger.endpoint.fileUploadParameter</td>
<td></td>
<td>The file upload parameter defined by the web server Default: 'file'</td>
</tr>
<tr>
<td>spec.fep.pgBadger.endpoint.insecure</td>
<td></td>
<td>equivalent to curl -insecure option, default to false</td>
</tr>
<tr>
<td>spec.fep.pgBadger.endpoint.url</td>
<td></td>
<td>Web server url to upload the report file</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customCertificates</td>
<td></td>
<td>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</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customCertificates.userName</td>
<td></td>
<td>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&gt;. 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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customCertificates.certificateName</td>
<td>(created by user)</td>
<td>Optional This points to Kubernetes TLS secret that contains the custom certificate. The certificate itself is stored in the key tls.crt.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customCertificates.caName</td>
<td>(created by user)</td>
<td>Optional This points to Kubernetes configmap</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup</td>
<td></td>
<td>Optional  &lt;br&gt; This section is defined to enable fepbackup sidecar for cluster backup feature.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.image.image</td>
<td>&lt;omitted&gt;</td>
<td>FEP backup container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-13-backup@ubi8-13-0.0  &lt;br&gt; It is optional.  &lt;br&gt; Image line is omitted by default. In such a case, it will pick up URL of image from operator container environment.  &lt;br&gt; If you specify the image, Operator will take that image to deploy backup container</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.image.pullPolicy</td>
<td>IfNotPresent</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.mcSpec.limits</td>
<td>cpu: 0.2  &lt;br&gt; memory: &quot;300Mi&quot;</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.mcSpec.requests</td>
<td>cpu: 0.1  &lt;br&gt; memory: &quot;200Mi&quot;</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.pgbackrestParams</td>
<td>[global]  &lt;br&gt; repo1-retention-full=7  &lt;br&gt; repo1-retention-full-type=time  &lt;br&gt; log-path=/database/log/backup</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.pgbackrestKeyParams</td>
<td>]“</td>
<td>When nothing is specified, and the parameter set in pgbackrest.conf is described from the line below.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.caName</td>
<td></td>
<td>Optional  &lt;br&gt; Set to use a CA file other than the system default.  &lt;br&gt; Specifies the name of the Configmap you created.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.schedule.num</td>
<td>0</td>
<td>Number of schedules to set  &lt;br&gt; The maximum number of backup schedules is 5.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.scheduleN.schedule</td>
<td>]“</td>
<td>Backup schedule in cron format.  &lt;br&gt; The date and time is UTC time.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.backup.scheduleN.type</td>
<td>]“</td>
<td>full: Perform a full backup (Back up the contents of the database cluster).  &lt;br&gt; incr = Perform an incremental backup (Back up only the database cluster files</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| `spec.fepChildCrVal.backup.scheduleN.repo` | 1 | Optional  
Gets a backup in the specified repository.  
The range is 1 to 256. |
| `spec.fepChildCrVal.customCertificates` | | List of custom certificates. |
| `spec.fepChildCrVal.customCertificate.s.userName` | | The username of the certificates. It is also the folder name in container to store the certificate files  
If not defined, the item on the list will be ignored. |
| `spec.fepChildCrVal.customCertificate.s.certificateName` | | The secret name which contains the certificate files  
If not defined, a folder to store the certificate files will still be created by the name defined in `spec.fepChildCrVal.customCertificate.s.userName` |
| `spec.fepChildCrVal.customCertificate.s.caName` | | The configmap of CA certificate |
| `spec.fepChildCrVal.customCertificate.s.privateKeyPassword` | | The secret which contains the certificate key password |
| `spec.fepChildCrVal.customPgAudit` | [output]  
logger = 'auditlog'  
log_directory = '/database/log/audit'  
log_truncate_on_rotation = on  
log_filename = 'pgaudit-%a.log'  
log_rotation_age = 1d  
log_rotation_size = 0  
[rule] | PgAudit file content |
| `spec.fepChildCrVal.customPgHba` | # define pg_hba custom rules here to be merged with default rules.  
# TYPE DATABASE USER ADDRESS METHOD | Entries to be inserted into pg_hba.conf |
| `spec.fepChildCrVal.customPgParams` | # define custom postgresql.conf parameters below to override defaults.  
# Current values are as per default FEP deployment  
shared_preload_libraries='pgx_datamasking,pg_prewarm,pg_stat_statements'  
session_preload_libraries='pg_prewarm'  
max_prepared_transactions = 100  
max_worker_processes = 30 | Postgres configuration in postgresql.conf |
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_connections</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>work_mem</td>
<td>1MB</td>
<td></td>
</tr>
<tr>
<td>maintenance_work_mem</td>
<td>12MB</td>
<td></td>
</tr>
<tr>
<td>shared_buffers</td>
<td>128MB</td>
<td></td>
</tr>
<tr>
<td>effective_cache_size</td>
<td>384MB</td>
<td></td>
</tr>
<tr>
<td>checkpoint_completion_target</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td># tcp parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_idle</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_interval</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_count</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td># logging parameters in default fep installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td># if log volume is not defined, log_directory should be</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_directory = /database/log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_filename = logfile-%a.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_file_mode = 0600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_truncate_on_rotation</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_rotation_age = 1d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_rotation_size = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_checkpoints</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_line_prefix = %e %t [%p]: [%l-1] user=%u,db=%d,app=%a.client=%h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_lock_waits</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_autovacuum_min_duration</td>
<td>60s</td>
<td></td>
</tr>
<tr>
<td>logging_collector</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>pgaudit.config_file=/opt/app-root/src/pgaudit-cfg/pgaudit.conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_replication_commands</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_min_messages</td>
<td>WARNING</td>
<td></td>
</tr>
<tr>
<td>log_destination</td>
<td>stderr</td>
<td></td>
</tr>
<tr>
<td># wal_archive parameters in default fep installation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>archive_mode</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>archive_command</td>
<td>'pgbackrest --stanza=backupstanza --config=/database/userdata/pgbackrest.conf archive-push %p'</td>
<td></td>
</tr>
<tr>
<td>wal_level</td>
<td>replica</td>
<td></td>
</tr>
<tr>
<td>max_wal_senders</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.dataVol</td>
<td></td>
<td>Mandatory volume</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.dataVol.size</td>
<td>2Gi (**)</td>
<td>Size of data volume. Data volume must be specified</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.dataVol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>StorageClass for data volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.dataVol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>accessModes for data volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.walVol</td>
<td></td>
<td>Mandatory volume</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.walVol.size</td>
<td>1200Mi (**)</td>
<td>Size of WAL volume. WAL volume must be specified</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.walVol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>StorageClass for WAL volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.walVol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>accessModes for WAL volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.tablespaceVol</td>
<td></td>
<td>Optional volume</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.tablespaceVol.size</td>
<td>512Mi (**)</td>
<td>Size of tablespace volume. This volume is optional and can be omitted</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.tablespaceVol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>StorageClass for tablespace volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.tablespaceVol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>accessModes for tablespace volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.archivewalVol</td>
<td></td>
<td>Mandatory if backup section is defined. Optional otherwise</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.archivewal Vol.size</td>
<td>1Gi (***)</td>
<td>Size of archivewal volume. This volume is optional and can be omitted.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.archivewal Vol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.archivewal Vol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>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].</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.logVol</td>
<td>Optional volume</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.logVol.size</td>
<td>1Gi (***)</td>
<td>Size of log volume. This volume is optional and can be omitted.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.logVol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>StorageClass for log volume: When this line is omitted, the PV created will use default storage class in the Kubernetes cluster.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.logVol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>accessModes for log volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce].</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.backupVol</td>
<td>Mandatory if backup section is defined. Optional otherwise</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.backupVol.size</td>
<td>2Gi (***)</td>
<td>Size of backup volume. This volume is optional and can be omitted.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.backupVol.storageClass</td>
<td>&lt;omitted&gt; (*)</td>
<td>StorageClass for backup volume:</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.backupVol.accessModes</td>
<td>&lt;omitted&gt; (*)</td>
<td>accessModes for backup volume:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specified as an array of accessModes e.g. [ReadWriteMany]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If omitted, it will be treated as [ReadWriteOnce]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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]</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdminPassword</td>
<td>admin-password</td>
<td>Password for user &quot;postgres&quot;</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgdb</td>
<td>mydb (*)</td>
<td>Database to be created during provisioning</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pguser</td>
<td>mydbuser (*)</td>
<td>Database user to be created during provisioning</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgpassword</td>
<td>mydbpassword</td>
<td>Password for database user pguser</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgrepluser</td>
<td>repluser (*)</td>
<td>Database user for replication</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgreplpassword</td>
<td>repluserpwd</td>
<td>Password for database user repluser</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.tdepassphrase</td>
<td>tde-passphrase</td>
<td>TDE keystore passphrase</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUser</td>
<td>rewind_user</td>
<td>Database user for Rewind</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserPassword</td>
<td>rewind_password</td>
<td>Password for database user rewinduser</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgMetricsUser</td>
<td></td>
<td>Optional user for FEPExporter connection. Can be defined afterwards</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgMetricsUserPassword</td>
<td></td>
<td>Optional Password for metrics user. Can be defined afterwards</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdminTls.certificateName</td>
<td></td>
<td>This points to Kubernetes TLS secret that contains the certificate of Postgres user &quot;postgres&quot;. Patroni will use this for certificate authentication. The</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgAdmin.Tls.caName</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>.spec.fepChildCrVal.sysUsers.pgAdmin.Tls.sslMode</td>
<td>prefer</td>
<td>Specify the type of TLS negotiation with the server.</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- disable</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- allow</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- prefer</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- require</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-ca</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-full</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgreplUserTls.caName</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgreplUserTls.sslMode</td>
<td>prefer</td>
<td>Specify the type of TLS negotiation with the server.</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- disable</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- allow</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- prefer</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- require</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-ca</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-full</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTls.caName</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode</td>
<td>prefer</td>
<td>Specify the type of TLS negotiation with the server.</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- disable</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- allow</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- prefer</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- require</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-ca</td>
</tr>
<tr>
<td>Specified values:</td>
<td></td>
<td>- verify-full</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode</code></td>
<td>prefer</td>
<td>Specify the type of TLS negotiation with the server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- allow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- prefer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- require</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- verify-ca</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- verify-full</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgMetricUserTls.certificateName</code></td>
<td>Optional</td>
<td>This points to Kubernetes TLS secret that contains the certificate of Postgres user defined by pgMetricsUser. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgMetricUserTls.caName</code></td>
<td>Optional</td>
<td>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.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgMetricUserTls.sslMode</code></td>
<td>prefer</td>
<td>Specify the type of TLS negotiation when FEPExporter connects to FEP server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- disable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- allow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- prefer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- require</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- verify-ca</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- verify-full</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.systemCertificates.key</code></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.systemCertificates.crt</code></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.systemCertificates.ca.crt</code></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.autoscale.scaleout.policy</code></td>
<td>off</td>
<td>Specifies whether to use the automatic scale out feature and the metric to base on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cpu_utilization (if based on CPU utilization)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- connection_number (if based on number of connections)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- off (without automatic scale out)</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| spec.fepChildCrVal.autoscale.scaleout.threshold | 40      | Specifies an integer as the threshold for performing scale out.  
- When cpu_utilization is specified for policy  
  Specifies the average CPU utilization as a percentage for the threshold. If this option is omitted, 40 (40%) is assumed.  
- When connection_number is specified for policy  
  Specifies the average value of the number of connections as a threshold. If you omit this option, 40 is assumed.                                                                                      |
| spec.fepChildCrVal.autoscale.scaleout.metricName | pg_capacity_connection_average | Specify this parameter if policy is connection_number. Ignored if policy is cpu_utilization.  
  The custom metrics server must publish the average number of connections in the FEP cluster under this name.  
  If omitted, pg_capacity_connection_average is assumed.                                                                                           |
| spec.fepChildCrVal.autoscale.scaleout.stabilizationWindowSeconds | 0       | This parameter controls the stability of scaling (variation in the number of replicas). Scale out is not performed unless the metric exceeds the threshold for more than the number of seconds specified for this parameter.  
  If omitted, 0 is assumed.                                                                                                                                                                                                                                                       |
| spec.fepChildCrVal.autoscale.limits.maxReplicas  | 2       | Maximum number of replicas (0 to 15)  
  (Value out of range)  
  Do not perform auto scale out                                                                                                                  |
| spec.fepChildCrVal.restore                        | Optional | Defines to restore specified backup data stored in object storage.                                                                                                                                                                                                                                                           |
| spec.fepChildCrVal.restore.pgbackrest.Params      | 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. |
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepChildCrVal.restore.pgbackrest</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>KeyParams</td>
<td></td>
<td>’</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.caName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to use a CA file other than the system default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the name of the ConfigMap created, in list format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The ConfigMap specified is mounted in /pgbackrest/storage-certs.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.mcSpec.limits</td>
<td>cpu: 200m</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>memory: 300Mi</td>
<td>CPU and memory allocated to the container performing the restore.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.mcSpec.requests</td>
<td>cpu: 100m</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>memory: 200Mi</td>
<td>CPU and memory allocated to the container performing the restore.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoretype</td>
<td>latest</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select the type of restore (latest or PITR).</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoredate</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the date to restore when spec.fepChildCrVal.restore.restoretype is ‘PITR’.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.restoretme</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies the time to restore when spec.fepChildCrVal.restore.restoretype is ‘PITR’.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.image</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image of the container to perform the restore.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is omitted by default. In this case, the URL for image is obtained from the operator container environment.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.restore.imagePullPo licy</td>
<td>IfNotPresent</td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When this field is defined, a major version upgrade is performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>However, if spec.fepChildCrVal.restore is defined, the FEPCluster build stops.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.upgrade.sourceCluster</td>
<td></td>
<td>Specifies the FEPClusterCR name from which to migrate data.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| spec.fepChildCrVal.upgrade.mcSpec.limits | cpu: 200m  
memory: 300Mi | Required if spec.fepChildCrVal.upgrade is defined.  
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 execution container uses the certificate defined as secret. |
<p>| spec.fepChildCrVal.upgrade.storage | | Optional |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>spec.fepChildCrVal.upgrade.storage.storageClass</code></td>
<td></td>
<td>Defines the storage for storing dump files.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.upgrade.storage.size</code></td>
<td>2Gi</td>
<td>Optional If omitted, the default storage class for your environment is used.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.upgrade.storage.accessModes</code></td>
<td>ReadWriteOnce</td>
<td>Optional accessModes for store the dump file e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce]</td>
</tr>
<tr>
<td><code>spec.fepLogging.image.image</code></td>
<td></td>
<td>Optional Fluentd Image of FEPLogging</td>
</tr>
<tr>
<td><code>spec.fepLogging.image.pullPolicy</code></td>
<td>IfNotPresent</td>
<td>Fluentd Image pull policy of FEPLogging</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.limits.cpu</code></td>
<td>500m</td>
<td>Max CPU allocated to fluentd container</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.limits.memory</code></td>
<td>700Mi</td>
<td>Max memory allocated to fluentd container</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.requests.cpu</code></td>
<td>200m</td>
<td>CPU allocation at start for fluentd container</td>
</tr>
<tr>
<td><code>spec.fepLogging.mcSpec.requests.memory</code></td>
<td>512Mi</td>
<td>Memory allocation at start for fluentd container</td>
</tr>
<tr>
<td><code>spec.fepLogging.sysExtraLogging</code></td>
<td>true</td>
<td>To turn on extra debugging messages for operator, set value to true. It can be turned on/off at any time</td>
</tr>
<tr>
<td><code>spec.fepLogging.restartRequired</code></td>
<td></td>
<td>To restart FEPLogging instance for applying any new configuration for example after certificate rotation</td>
</tr>
<tr>
<td><code>spec.fepLogging.scrapeInterval</code></td>
<td></td>
<td>Optional Scrape interval for Prometheus to fetch metrics from FEPLogging instance</td>
</tr>
<tr>
<td><code>spec.fepLogging.scrapeTimeout</code></td>
<td></td>
<td>Optional Scrape Timeout for Prometheus to fetch metrics from FEPLogging instance</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.host</code></td>
<td></td>
<td>Optional Target Elasticsearch host name</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.port</code></td>
<td></td>
<td>Optional Target Elasticsearch port number</td>
</tr>
<tr>
<td><code>spec.fepLogging.elastic.authSecret.secretName</code></td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.authSecret.useKey</td>
<td>Optional</td>
<td>Username key specified in Elasticsearch authentication secret</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.authSecret.passwordKey</td>
<td>Optional</td>
<td>Password key specified in Elasticsearch authentication secret</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.logstashPrefix</td>
<td>Optional</td>
<td>Logstash prefix to differentiate index pattern in elastic search. Default value is postgres</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.scheme</td>
<td>Optional</td>
<td>Connection scheme between FEPLogging &amp; Elasticsearch. Possible options http &amp; https</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.sslVerify</td>
<td>Optional</td>
<td>Set to true if you want to verify ssl certificate. If set to false then will not consider TLS certificate</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds fluentd certificate</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Elasticsearch to verify Elasticsearch TLS connection</td>
</tr>
<tr>
<td>spec.fepLogging.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds Fluentd certificate</td>
</tr>
<tr>
<td>spec.fepLogging.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
</tr>
<tr>
<td>spec.prometheus.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds Prometheus certificate</td>
</tr>
<tr>
<td>spec.prometheus.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
</tr>
</tbody>
</table>
**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

<table>
<thead>
<tr>
<th>Field name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>status.fepStatus.fepClusterReady</td>
<td>Will be true or false to reflect if the whole cluster is ready. Kubernetes cluster information is fetched to check number of instances 'READY' &amp; 'RUNNING' is equal to number of Configured instances.</td>
</tr>
</tbody>
</table>

**Note**

"fepClusterReady" flag will be set at first FEPCluster creation time only. fepClusterReady flag does not participate in the next reconciliation loop)

### 1.2 Custom Resource Parameters

This section explains the Custom Resource Parameters.

#### 1.2.1 FEPCluster Custom Resource Parameters

<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRD Name</td>
<td>FEPCluster</td>
</tr>
<tr>
<td>Category</td>
<td>Details</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>versions:</td>
<td>- name: v2</td>
</tr>
<tr>
<td></td>
<td>served: true</td>
</tr>
<tr>
<td></td>
<td>storage: true</td>
</tr>
<tr>
<td></td>
<td>schema:</td>
</tr>
<tr>
<td></td>
<td>openAPIV3Schema:</td>
</tr>
<tr>
<td></td>
<td>description: FEPCluster is the Schema for the fepclusters API</td>
</tr>
<tr>
<td></td>
<td>properties:</td>
</tr>
<tr>
<td></td>
<td>apiVersion:</td>
</tr>
<tr>
<td></td>
<td>description: 'APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources</a>'</td>
</tr>
<tr>
<td></td>
<td>type: string</td>
</tr>
<tr>
<td></td>
<td>kind:</td>
</tr>
<tr>
<td></td>
<td>description: 'Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: <a href="https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds">https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds</a>'</td>
</tr>
<tr>
<td></td>
<td>type: string</td>
</tr>
<tr>
<td></td>
<td>metadata:</td>
</tr>
<tr>
<td></td>
<td>spec:</td>
</tr>
<tr>
<td></td>
<td>description: Spec defines the desired state of FEPCluster</td>
</tr>
<tr>
<td></td>
<td>type: object</td>
</tr>
<tr>
<td></td>
<td>x-kubernetes-preserve-unknown-fields: true</td>
</tr>
<tr>
<td></td>
<td>status:</td>
</tr>
<tr>
<td></td>
<td>description: Status defines the observed state of FEPCluster</td>
</tr>
<tr>
<td></td>
<td>type: object</td>
</tr>
<tr>
<td></td>
<td>x-kubernetes-preserve-unknown-fields: true</td>
</tr>
<tr>
<td></td>
<td>type: object</td>
</tr>
<tr>
<td></td>
<td>subresources:</td>
</tr>
<tr>
<td></td>
<td>status: {}</td>
</tr>
</tbody>
</table>

**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**

```yaml
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: new-fep
  namespace: new-fep
spec:
  fep:
    forceSsl: true
```

- 20 -
image: 'quay.io/fujitsu/fep-server-test:ubi8-patroni_72'
pullPolicy: IfNotPresent
mcSpec:
  limits:
    cpu: 500m
    memory: 700Mi
  requests:
    cpu: 200m
    memory: 512Mi
podAntiAffinity: true
podDisruptionBudget: true
instances: '3'
servicePort: 27500
syncMode: 'on'
sysExtraLogging: false
replicationSlots: |
  demo_subscription1:
    type: logical
    database: postgres
    plugin: pgoutput
  demo_subscription2:
    type: logical
    database: postgres
    plugin: pgoutput
  demo_subscription3:
    type: logical
    database: postgres
    plugin: pgoutput
fepChildCrVal:
customCertificates:
  - userName: my_cert_folder1
    certificateName: my_cert1_secret
    caName: my_ca_configmap
    privateKeyPassword: my_cert1_key_secret
  - userName: my_cert_folder2
    certificateName: my_cert2_secret
    caName: my_ca2_configmap
    privateKeyPassword: my_cert2_key_secret
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,vci,pgaudit,pg_prewarm, pg_stat_statements'
  session_preload_libraries='vci,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
pgx_global_metacache = 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

# vci parameters in default fep installation
vci.enable = on
vci.maintenance_work_mem = 256MB
vci.max_local_ros = 64MB
vci.force_max_parallelism = off

# 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 = 10
wal_keep_segments = 64
wal_sender_timeout = 60s
track_activities = on
track_counts = on

backup:
image:
  image: 'quay.io/fujitsu/fep-backup@sha256:3434345'
pullPolicy: IfNotPresent
mcSpec:
  limits:
    cpu: 200m
    memory: 300Mi
  requests:
    cpu: 100m
    memory: 200Mi
pgbackrestParams: |
# define custom pgbackrest.conf parameters below to override defaults.

[globals]
repo1-retention-full = 30
repo1-retention-full-type = time
preScript: " "
postScript: " "
schedule:
  num: 2
  schedule1:
    schedule: "15 0 * * 0"
    type: "full"
schedule2:
  schedule: "15 0 * 1-6"
  type: "incx"
schedule3:
  schedule: " "
  type: " "
schedule4:
  schedule: " "
  type: " "
schedule5:
  schedule: " "
  type: " "

storage:
dataVol:
  size: 2Gi

tablespaceVol:
  size: 512Mi

walVol:
  size: 1200Mi

archiveWalVol:
  size: 1Gi

backupVol:
  size: 2Gi

logVol:
  size: 1Gi

sysUsers:
pgAdminPassword: admin-password
pgdb: mydb
pgpassword: mydbpassword
pguser: mydbuser
pgrepluser: repluser
pgreplpassword: repluserpwd
tdepassphrase: tde-passphrase

systemCertificates:
  key: |-
    MIIBpAIBAAKCAQEAvhL4D/01Lmm/RY3nu+jgLOdLYEEg0wgMhxspyPRb43paWSFlp
gX1CNAPzB1tnn4LVGo6n7Tq7V73Mz41NhpuTvjYHTS6wtf7dQJ7bbReWQCD5f5hK
    QpekJ5hAv/5uQ8x154FppJvmX6CtUb89i8X717GPr05uiz7kj83S5kSpX6Kp
    wgwuEufbvenaYCI8Kap8c7A8RIMjW2#W9ri1n4b82Yi1Vh0mchLrX88Hw7qJvqBh
    9laEwq1/K7tpWQVp8d3l1Lt-H6gBECd6n4q9/*v1x0J2MoVK63Q+z773Y/3x5gBNN
    +/K9scht91L6AcEzIoJ352pA4vmeLwrkK6kJN8w1DAQABAnIBAF2v9HR1r1q4Byr
    6xw1Zfj776Z7nOAYPRAp30Q0O2aKsvfrBhQ12yn3DF0b9bWp8z4ubnq4A+9HP31S
    72eUSIp3mizG8l1c3y8FYPVfuBn6JKMFQ29M+snSXzCFmTmHl5915bLsJ70nq4Q
    GY1JHRP8ezgbArocD1exK0j9IVRNIWOz2CajztyJyNSS4vPaExAy5W/SXuRz1
    A5mxmz2xVn55+FJpc2+H4Q+ Rd+0AdLKrRA0yGCMG3x5iygwb7tjzKmdHiJi5nI
    em+kYxChGd4FK2u2zJ1+L1W9d+77CtEDxIYMKv81taF79aQz7J7MvotGtvVn4L
    KTDOAECqYEA5hSh/0P90b+1WM4xhQAmntpwW0h0QPJKjMYI0XZF3wMK8brzEKKk
    8p
    klbZIM24xUtM5b/hKvcqcraxY211UmGDspbu0xM6G6WnqjAh0T8K8VhY0ihhK1

- 23 -
It should also be noted that all the passwords / passphrase and certificates will be masked after the creation of the CR. This includes
- Also, initial pgAdminPassword: admin-password
- pgpassword: mydbpassword
- pgrepplpassword: repluserpwd
- tdepassphrase: tde-passphrase
- pgRewindPassword: rewind_passwordword (Optional - if defined)
- pgMetricsPassword: metrics_password (Optional - if defined)
- certificate.key
- certificate.crt
- certificate.ca.crt

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.

<table>
<thead>
<tr>
<th>Child CR Name</th>
<th>Relevant sections in FEP Cluster CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEPBackup</td>
<td>spec.fepChildCrVal.backup</td>
</tr>
<tr>
<td>FEPCert</td>
<td>spec.fepChildCrVal.systemCertificates</td>
</tr>
<tr>
<td>FEPConfig</td>
<td>spec.fepChildCrVal.customPgAudit</td>
</tr>
<tr>
<td></td>
<td>spec.fepChildCrVal.customPgHba</td>
</tr>
<tr>
<td></td>
<td>spec.fepChildCrVal.customPgParams</td>
</tr>
<tr>
<td>FEPUser</td>
<td>spec.fepChildCrVal.sysUsers</td>
</tr>
<tr>
<td>FEPVolume</td>
<td>spec.fepChildCrVal.storage</td>
</tr>
</tbody>
</table>

### 1.2.3 FEPCfg Child Custom Resource Parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata.name</td>
<td>&lt;same-as-in-FEPCluster&gt;</td>
<td>This value is inherited from parent FEPCluster CR</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>&lt;same-as-in-FEPCluster&gt;</td>
<td>This value is inherited from parent FEPCluster CR</td>
</tr>
<tr>
<td>spec.customPgAudit</td>
<td>All line specified in spec.fepChildCrVal.customPgAudit</td>
<td>Audit rules can be updated in this section. Requires restart. Note: initial values inherited once only at start. Changes to FEPCfg directly</td>
</tr>
<tr>
<td>spec.customPgHba</td>
<td>All line specified in pg_hba rules can be added in this section</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customPgHba of FEPCluster CR</td>
<td></td>
<td>Note: Inherited once at start. Changes to FEPConfig directly</td>
</tr>
<tr>
<td>spec.customPgParams</td>
<td>All line specified in spec.fepChildCrVal.customPgParams of FEPCluster CR</td>
<td>All postgres parameters are listed here to overwrite defaults. Note: Inherited once at start. Changes to FEPConfig directly</td>
</tr>
<tr>
<td>spec.replicationSlots</td>
<td></td>
<td>Optional: Details of replication slots if defined in FEPCluster</td>
</tr>
</tbody>
</table>

Example of FEPConfig CR created

```yaml
apiVersion: fep.fujitsu.io/v1
gkind: 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_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,vci,pgaudit,pg_prewarm,pg_stat_statements'
    session_preload_libraries='vci,pg_prewarm'
    maxprepared_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_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' or 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
- log_replication_commands = on
- log_min_messages = WARNING
- log_destination = stderr
- vci.enable = on
- vci.maintenance_work_mem = 256MB
- vci.max_local_ros = 64MB
- vci.force_max_parallelism = off
- 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

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>metadata.name</td>
<td>&lt;same-as-in-FEPCluster&gt;</td>
<td>This value is inherited from parent FEPCluster CR</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>&lt;same-as-in-FEPCluster&gt;</td>
<td>This value is inherited from parent FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgAdminPassword</td>
<td>spec.fepChildCrVal.users.pgAdminPassword of FEPCluster CR</td>
<td>postgres superuser password. Masked once secret is created/changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: initial values inherited once only at start. Changes to FEPUser directly</td>
</tr>
<tr>
<td>spec.pgdb</td>
<td>spec.fepChildCrVal.users.pgdb of FEPCluster CR</td>
<td>Name of a user database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Created once only at start. Cannot be changed</td>
</tr>
<tr>
<td>spec.pgpassword</td>
<td>spec.fepChildCrVal.users.pgpassword of FEPCluster CR</td>
<td>Password for superuser for user database pgdb. Masked once secret is created/changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: initial values inherited once only at start. Changes to FEPUser directly</td>
</tr>
<tr>
<td>spec.pguser</td>
<td>spec.fepChildCrVal.users.pguser of FEPCluster CR</td>
<td>Name of a user database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: Created once only at start. Cannot be changed</td>
</tr>
<tr>
<td>spec.pgrepluser</td>
<td>spec.fepChildCrVal.users.pgrepluser of FEPCluster CR</td>
<td>Name of a database user for replication</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>spec.pgreplpassword</td>
<td>spec.fepChildCrVal.users.pgreplpassword of FEPCluster CR</td>
<td>Password for pgrepluser</td>
</tr>
<tr>
<td>spec.tdepassphrase</td>
<td>spec.fepChildCrVal.users.tdepassphrase of FEPCluster CR</td>
<td>Passphrase for encrypting/decrypting keystore file which contains the TDE encryption key</td>
</tr>
<tr>
<td>spec.pgRewindUser</td>
<td>rewind_user</td>
<td>Database user for Rewind</td>
</tr>
<tr>
<td>spec.pgRewindUserPassword</td>
<td>rewind_password</td>
<td>Password for database user rewinduser</td>
</tr>
<tr>
<td>spec.pgMetricsUser</td>
<td>spec.fepChildCrVal.sysUsers.pgMetricsUser</td>
<td>Optional See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgMetricsPassword</td>
<td>spec.fepChildCrVal.sysUsers.pgMetricsPassword</td>
<td>Optional See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgAdminTlss</td>
<td>spec.fepChildCrVal.sysUsers.pgAdminTlss</td>
<td>Optional section See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgrepluserTlss</td>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTlss</td>
<td>Optional section See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgRewindUserTlss</td>
<td>spec.fepChildCrVal.sysUsers.pgRewindUserTlss</td>
<td>Optional section See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgMetricsUserTlss</td>
<td>spec.fepChildCrVal.sysUsers.pgMetricsUserTlss</td>
<td>Optional section See details in FEPCluster CR</td>
</tr>
</tbody>
</table>

Example of FEPUser CR created

```yaml
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
  pgAdminTlss:
    certificateName: admin-client-certs-secret
    caName: admin-ssl-rootcert-configmap
    sslMode: prefer
  pgrepluserTlss:
    certificateName: repluser-client-certs-secret
    caName: repluser-ca-name-configmap
    sslMode: prefer
  pgRewindUserTlss:
    certificateName: rewinduser-client-certs-secret
```
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/swwitchover occurred 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 (FEPVolume Controller) takes care of creating the required volumes. After initial FEPCluster create, new volume cannot be added later and storageClass or accessModes can not be changed.

Only size of an initially created volume can be changed if and only if underlying storageClass supports dynamic change of size.

Below is the schema of the FEPVolume CR:

<table>
<thead>
<tr>
<th>Field</th>
<th>Mandatory</th>
<th>Sub-Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>archivewalVol</td>
<td>No</td>
<td>size</td>
<td>1Gi</td>
<td>Size of the volume, expandable later</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td></td>
<td>Defaults to platform default if omitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td></td>
<td>Defaults to ReadWriteOnce if omitted</td>
</tr>
<tr>
<td>backupVol</td>
<td>No</td>
<td>size</td>
<td>2Gi</td>
<td>-do-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td></td>
<td>Defaults to platform default if omitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td></td>
<td>Defaults to ReadWriteOnce if omitted</td>
</tr>
<tr>
<td>dataVol</td>
<td>Yes</td>
<td>size</td>
<td>2Gi</td>
<td>-do-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td></td>
<td>Defaults to platform default if omitted</td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td></td>
<td>Defaults to ReadWriteOnce if omitted</td>
</tr>
</tbody>
</table>
### Field | Mandatory | Sub-Field | Default | Description
--- | --- | --- | --- | ---
logVol | No | size | 1Gi | -do-
 |  | storageClass | Defaults to platform default if omitted | |
 |  | accessModes | Defaults to ReadWriteOnce if omitted | |
tablespaceVol | No | size | 512Mi | -do-
 |  | storageClass | Defaults to platform default if omitted | |
 |  | accessModes | Defaults to ReadWriteOnce if omitted | |
walVol | Yes | Size | 1200Mi | -do-
 |  | storageClass | Defaults to platform default if omitted | |
 |  | accessModes | Defaults to ReadWriteOnce if omitted | |

#### 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

```yaml
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
```
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's certs section of the parent FEPCluster CR.

Below is the schema of the FEPCert CR:

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacrt</td>
<td>Defaults to dummy self signed crt from parent FEPCluster CR</td>
<td>Can be replaced with customer's own CA cert</td>
</tr>
<tr>
<td>crt</td>
<td>Defaults to dummy self signed crt from parent FEPCluster CR</td>
<td>Can be replaced with customer's own trusted cert</td>
</tr>
<tr>
<td>key</td>
<td>Defaults to dummy key from parent FEPCluster CR</td>
<td>Can be replaced with customer's own key</td>
</tr>
</tbody>
</table>

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.

```plaintext
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

```plaintext
apiVersion: fep.fujitsu.io/v1
kind: FEPCert
metadata:
  name: new-fep
  namespace: ansible-operator-poc
spec:
  key: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIEowIBAAKCAQEA4AI33yvH2wsj+ta6gqV6wzJqF8odIftIpCfbrVcUUtLFKJ1I
    2e4SecTiK603C/11Vvu1W1pqg5I065+fQQLO06z1/AuQT78YUn/Wlm9x1aHVsv4AN
    B5JWwD0jrR73o7nRPGXflabP0rGE2mJJcV9nExJ3leaktgT3s8bY1XvchyYP
    mjdbFxbTb07iq+6+cv4oReRxOK9U7f75euE0iC/490J6r5Rs41gO8aJQNCUF1TF
    YvmAh7gcsdnSFB8N1UfH5o8m1W0DCJFJWhHtL08H+s6L1izwTHLj5G2pdk66W
    dgmuSH2pDm18CDNLDv98A4j7i+ISSRKJcVPlnuQIDAQABaoI8AFPQK70x2w/+BA0b
    yMIUupdctIMb/54Cr/xR0MwVvDbsjigNVPjH0vqB8Y1B2FAIT9Q0bgJo068AvQ0dWN
    Rbo/0/yYiNDFjaWJ1aIAH10/oCWxBfAEqgqVDjhB+e1xaw2r2x7Gxm+p925k30
    l6pIVY+18J RJkliV1V2hwL/R3J0tPr+f+mZtLJyVOI+f+ySqiJ+T2HuAjm4E9Kxj
    cE9mMj2BP7QzixXsvxy0f+zbgL1B89QdZAFCU5eErhRsDRxDRW+KfO1Vftuy48JZ
    voK7VGHvF/qyssW4s+61AO6tpuYnnM0Y23sOGoWPKTcQK0MekYksLb1mCj9N
    9hodJteEgYEA5E6hE0f4u0KeSTDP697UCvXLo0R58FDe/S8XNvScn29jj0xqIg
    -----END RSA PRIVATE KEY-----
```

- name: log
  sysExtraLogging: false
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

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Fixed</td>
</tr>
<tr>
<td>kind</td>
<td>FEPBackup</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.name</td>
<td>&lt;clustername&gt;</td>
<td>Enter the CR name.</td>
</tr>
<tr>
<td>spec.pgbackrestParams</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.schedule.num</td>
<td>Integer</td>
<td>Number of schedules to set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The maximum number of backup schedules is 5.</td>
</tr>
<tr>
<td>spec.scheduleN.schedule</td>
<td>-</td>
<td>Write the date and time of the Nth schedule in cron format.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The date and time is UTC time.</td>
</tr>
<tr>
<td>spec.scheduleN.type</td>
<td>full/incr</td>
<td>full: Perform a full backup (Back up the contents of the database cluster).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>incr — Perform an incremental backup (Back up only the database cluster files that were changed to the last backup migration).</td>
</tr>
<tr>
<td>spec.preScript</td>
<td>&quot; &quot;</td>
<td>This parameter must specify a default value.</td>
</tr>
<tr>
<td>spec.postScript</td>
<td>&quot; &quot;</td>
<td>This parameter must specify a default value.</td>
</tr>
</tbody>
</table>

Example of FEPBackup CR created

```yaml
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"
pbbackrestParams: |
  # define custom pgbackrest.conf parameters below to override defaults.
  [global]
  repol-retention-full = 30
  repol-retention-full-type = time
  ...
```

### 1.2.8 FEPRestore Custom Resource Parameters

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Fixed</td>
</tr>
<tr>
<td>kind</td>
<td>FEPRestore</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.name</td>
<td>-</td>
<td>Enter the CR name.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepVersion</td>
<td>Optional</td>
<td>To use FEPRestore image of given version. Possible values: 12, 13 &amp; 14</td>
</tr>
</tbody>
</table>
| spec.image                  | <current-released-image> | FEP restore container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-12-restore:ubi8-12-1.1  
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.  
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  
Example) "2020-11-25"                                                                                                                  |
| spec.restoreTime            | -                       | If spec.restoreType is PITR, specifies the PITR time (UTC) in HH: MM: SS format  
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.storage.caName | Optional |                                                                                                                                          |
Specify this to change the `spec.fepChildCrVal.storage.caName` setting in FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this to change the `spec.fepChildCrVal.storage.backupVol` setting in FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this option to change the `spec.fepChildCrVal.storage.archivewalVol` setting for FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this to change the `spec.fepChildCrVal.storage.dataVol` setting for FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this to change the `spec.fepChildCrVal.storage.walVol` setting for FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this to change the `spec.fepChildCrVal.storage.logVol` setting for FEP Cluster CR when restoring to a new DB cluster.

**Optional**
Specify this to change the `spec.fepChildCrVal.storage.tablespaceVol` setting for FEP Cluster CR when restoring to a new DB cluster.

---

**Example of FEPRestore CR created**

```yaml
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
  restoretimel: 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Fixed</td>
</tr>
<tr>
<td>kind</td>
<td>FEPPgpool2</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.name</td>
<td>-</td>
<td>List the name of the FEP Pgpool2 container.</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>-</td>
<td>Specify the namespace of the environment where you want to deploy the operator.</td>
</tr>
<tr>
<td>spec.fepVersion</td>
<td>Optional</td>
<td>To use FEPPgpool2 image of given version. Possible values: 12, 13 &amp; 14</td>
</tr>
<tr>
<td>spec.image</td>
<td>&lt;current-released-image&gt;</td>
<td>FEPPgpool2 container image to be used</td>
</tr>
<tr>
<td></td>
<td></td>
<td>quay.io/fujitsu/fujitsu-enterprise-postgres-12-pgpool2:ubi8-12-1.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is optional.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Image is left blank by default. In such a case, it will pick up URL of image from operator container environment.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify the image, Operator will take that image to deploy container.</td>
</tr>
<tr>
<td>spec.count</td>
<td>2</td>
<td>List the number of FEP Pgpool2 containers to create.</td>
</tr>
<tr>
<td>spec.serviceport</td>
<td>9999</td>
<td>Describes the TCP port for connecting to the FEP Pgpool2 container.</td>
</tr>
<tr>
<td>spec.statusport</td>
<td>9898</td>
<td>Identifies the TCP port for connecting to the PCP process.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.limits.cpu</td>
<td>400m</td>
<td>List the number of CPUs (restriction) to allocate to resources.limits.cpu.</td>
</tr>
<tr>
<td>spec.limits.memory</td>
<td>512Mi</td>
<td>Specifies the memory size (restriction) to allocate to resources.limits.memory.</td>
</tr>
<tr>
<td>spec.requests.cpu</td>
<td>200m</td>
<td>List the number of CPUs (request) to allocate to resources.requests.cpu.</td>
</tr>
<tr>
<td>spec.requests.memory</td>
<td>256Mi</td>
<td>Specifies the memory size (request) to allocate to resources.requests.memory</td>
</tr>
<tr>
<td>spec.fepclustername</td>
<td>new-fep</td>
<td>Enter the FEPCluster name to connect to.</td>
</tr>
<tr>
<td>spec.customhba</td>
<td></td>
<td>If you want to use pool _ hba.conf, describe what pool _ hba.conf should contain from the line below.</td>
</tr>
<tr>
<td>spec.customparams</td>
<td></td>
<td>&quot;</td>
</tr>
</tbody>
</table>

- 37 -
<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>log_connections = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_line_prefix = '%t: pid %p:'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>load_balance_mode = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ignore_leading_white_space = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>white_function_list = &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>black_function_list = 'currval,lastval,nextval,setval'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>black_query_pattern_list = &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>database_redirect_preference_list = &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>app_name_redirect_preference_list = &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>allow_sql_comments = off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>disable_load_balance_on_write = 'transaction'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>statement_level_load_balance = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sr_check_period = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sr_check_user = 'postgres'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>delay_threshold = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_standby_delay = 'none'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl_prefer_server_ciphers = off</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl_ecdh_curve = 'prime256v1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl_dh_params_file = &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relcache_expire = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relcache_size = 256</td>
<td></td>
<td></td>
</tr>
<tr>
<td>check_temp_table = catalog</td>
<td></td>
<td></td>
</tr>
<tr>
<td>check_unlogged_table = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enable_shared_relcache = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>relcache_query_target = primary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wd_port0 = 9000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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. |
If you want to do it, "|" and the contents of the public x 509 certificate from the line below.

If you want to do it, "|" and the following lines describe the contents of the CA root certificate in PEM format.

Specifies the persistent volume size for log output.

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.

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter name (Specified format)</th>
<th>Restart required after change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection settings</td>
<td>listen_addresses (string)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>pcp_listen_addresses (string)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>num_init_children (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>reserved_connections (integer)</td>
<td>Y</td>
</tr>
<tr>
<td>Authentication settings</td>
<td>enable_pool_hba (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>allow_clear_text_frontend_auth (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>authentication_timeout (integer)</td>
<td></td>
</tr>
<tr>
<td>Backend settings</td>
<td>backend_weight0 (floating point)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>backend_weight1 (floating point)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>backend_flag0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>backend_flag1</td>
<td></td>
</tr>
<tr>
<td>Connection pooling</td>
<td>connection_cache (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>max_pool (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>listen_backlog_multiplier (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>serialize_accept (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>child_life_time (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>client_idle_limit (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>child_max_connections (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>connection_life_time (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>reset_query_list (string)</td>
<td></td>
</tr>
<tr>
<td>Error reporting and log acquisition</td>
<td>client_min_messages (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_min_messages (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_statement (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_per_node_statement (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_client_messages (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_hostname (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_connections (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_error_verbosity (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_line_prefix (string)</td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>Parameter name (Specified format)</td>
<td>Restart required after change</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Load sharing settings</td>
<td>load_balance_mode (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ignore_leading_white_space (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>white_function_list (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>black_function_list (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>black_query_pattern_list (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>database_redirect_preference_list (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>app_name_redirect_preference_list (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>allow_sql_comments (boolean)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>disable_load_balance_on_write (string)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>statement_level_load_balance (boolean)</td>
<td></td>
</tr>
<tr>
<td>Health check</td>
<td>connect_timeout (integer)</td>
<td></td>
</tr>
<tr>
<td>Streaming replication check</td>
<td>sr_check_period (integer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sr_check_user (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sr_check_password (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sr_check_database (string)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>delay_threshold (integer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_standby_delay (string)</td>
<td></td>
</tr>
<tr>
<td>Secure Sockect Layer (SSL)</td>
<td>ssl (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ssl_ciphers (string)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ssl_prefer_server_ciphers (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ssl_ecdh_curve (string)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ssl_dh_params_file (string)</td>
<td>Y</td>
</tr>
<tr>
<td>Other parameters</td>
<td>relcache_expire (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>relcache_size (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>enable_shared_relcache (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>relcache_query_target (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>check_temp_table (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>check_unlogged_table (boolean)</td>
<td></td>
</tr>
</tbody>
</table>

### 1.2.10 FEPAction Custom Resource Parameters

Specify parameters in the format described below.

<table>
<thead>
<tr>
<th>Custom resource spec</th>
<th>Default</th>
<th>Change effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>.spec.targetClusterName</td>
<td></td>
<td>Must specify target FEP Cluster name within namespace mentioned in metadata.</td>
</tr>
<tr>
<td>.spec.targetPgpool2Name</td>
<td></td>
<td>Must specify target FEPPgpool2 name within namespace mentioned in metadata when using pgpool2_restart.</td>
</tr>
<tr>
<td>.spec.fepAction.type</td>
<td></td>
<td>Must specify action type. Supported action types are: restart.</td>
</tr>
<tr>
<td>Custom resource spec</td>
<td>Default</td>
<td>Change effect</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
<td>---------------</td>
</tr>
<tr>
<td>.spec.fepAction.args</td>
<td></td>
<td>Must specify arguments needed for given action. For details of args corresponding to each action refer to &quot;1.2.10.1 FEPAction Specific Operation Details&quot;.</td>
</tr>
<tr>
<td>.spec.fepAction.backupType</td>
<td>full</td>
<td>Options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specify backup for fepAction.type, the type of backup is used.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>full: Performs a full backup (backs up the contents of the database cluster).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>incr: Perform an incremental backup (Back up only the database cluster files that were changed during the last backup migration).</td>
</tr>
<tr>
<td>.spec.fepAction.backupRepo</td>
<td>1</td>
<td>Options</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gets a backup in the specified repository.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The range is 1 to 256.</td>
</tr>
<tr>
<td>.spec.sysExtraLogging</td>
<td></td>
<td>To turn extra debugging on, set value to true. It can be turned on/off at any time.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>fepStatus (with possible values)</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>fepActionStatus:</td>
<td>fepStatus is inserted at the top of FEPAction CR</td>
</tr>
<tr>
<td>fepActionCondition: Success</td>
<td>This flag is inserted in fepAction CR to reflect success or failure of requested action</td>
</tr>
<tr>
<td>fepActionResult: &gt; &quot;details&quot;</td>
<td>The result contains verbose details corresponding to the specific action been executed. Should be noted that it is either plain text of HTTP output.</td>
</tr>
<tr>
<td>processedTimestamp: &lt;time stamp&gt;</td>
<td>Denotes time of action execution by the Operator</td>
</tr>
</tbody>
</table>

```yaml
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
```
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:

```yaml
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:

```yaml
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):

```yaml
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:

```yaml
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:

```yaml
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:

```yaml
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:

```yaml
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:

```yaml
spec:
  targetClusterName: new-fep
  fepAction:
```
- 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.

### 1.2.11 FEPExporter Custom Resource

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Mandatory as it is</td>
</tr>
<tr>
<td>kind</td>
<td>FEPExporter</td>
<td>Mandatory as it is</td>
</tr>
<tr>
<td>metadata.name</td>
<td>fep-monitor</td>
<td>Name of FEPExporter CR - must be unique in namespace</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>fep-ns</td>
<td>Namespace - OCP populates it as current</td>
</tr>
<tr>
<td>spec.prometheus</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Prometheus MTLS spec section</td>
<td></td>
</tr>
<tr>
<td>spec.prometheus.tls</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.prometheus.tls.certificateName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>spec.prometheus.tls.caName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>spec.fepExporter.authSecret</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>Base Authentication secret to provide username &amp; encrypted password of user</td>
<td></td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.secretName</td>
<td></td>
<td>Secret name</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.userKey</td>
<td></td>
<td>Key of username in specified secret</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.passwordKey</td>
<td></td>
<td>Key of password in specified secret</td>
</tr>
<tr>
<td>spec.fepExporter.tls</td>
<td></td>
<td>FEPEXporter MTLS specs</td>
</tr>
<tr>
<td>spec.fepExporter.tls.certificateName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepExporter.tls.caName</td>
<td>Optional</td>
<td>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.</td>
</tr>
<tr>
<td>spec.fepExporter.disableDefaultQueries</td>
<td>false</td>
<td>Optional No defined or set to false =&gt; Create default queries Defined and set to true =&gt; Do not create default queries.</td>
</tr>
<tr>
<td>spec.fepExporter.disableDefaultAlertRules</td>
<td>false</td>
<td>Optional No defined or set to false =&gt; Create default alert rules Defined and set to true =&gt; Do not create default alert rules. If Default queries are disabled =&gt; Do not create default alert rule.</td>
</tr>
<tr>
<td>spec.fepExporter.exporterLogLevel</td>
<td>error</td>
<td>Set logging level: one of debug, info, warn, error</td>
</tr>
<tr>
<td>spec.fepExporter.fepClusterList</td>
<td></td>
<td>Array of FEPCluster to monitor</td>
</tr>
<tr>
<td>spec.fepExporter.image.image</td>
<td></td>
<td>quay.io/fujitsu/fep-exporter-test:v1.0.0</td>
</tr>
<tr>
<td>spec.fepExporter.image.pullPolicy</td>
<td>IfNotPresent</td>
<td>Always or IfNotPresent</td>
</tr>
<tr>
<td>spec.fepExporter.mcSpec.limits</td>
<td>cpu: 500m memory: 700Mi</td>
<td>Max CPU allocated to exporter container Max memory allocated to exporter container</td>
</tr>
<tr>
<td>spec.fepExporter.mcSpec.requests</td>
<td>cpu: 200m memory: 512Mi</td>
<td>CPU allocation at start for exporter container memory allocation at start for exporter container</td>
</tr>
<tr>
<td>spec.fepExporter.scrapeInterval</td>
<td>30s</td>
<td>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</td>
</tr>
<tr>
<td>spec.fepExporter.scrapeTimeout</td>
<td>30s</td>
<td>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</td>
</tr>
<tr>
<td>spec.fepExporter.sysExtraLogging</td>
<td>true</td>
<td>To turn on extra debugging messages for operator, set value to true It can be turned on/off at any time</td>
</tr>
<tr>
<td>spec.fepExporter.restartRequired</td>
<td>false</td>
<td>True: To restart FEPExporter, when there is any change found in CR or FEPCluster False: Will not restart FEPExporter</td>
</tr>
<tr>
<td>spec.fepExporter.userCustomQueries</td>
<td></td>
<td>Optional Section Example user’s custom query to extract additional metrics.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Fixed</td>
</tr>
<tr>
<td>kind</td>
<td>FEPAutoscale</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.name</td>
<td>Same as FEPClusterCR</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>Same as FEPClusterCR</td>
<td>Fixed</td>
</tr>
<tr>
<td>spec.scaleout.policy</td>
<td>off</td>
<td>[cpu_utilization/connection_number/off]</td>
</tr>
<tr>
<td>spec.scaleout.threshold</td>
<td>cpu_utilization: 40</td>
<td>Threshold</td>
</tr>
<tr>
<td></td>
<td>connection_number: 40</td>
<td></td>
</tr>
<tr>
<td>spec.scaleout.metricName</td>
<td>pg_capacity_connection_average</td>
<td>Specify this parameter if policy is connection_number.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The custom metrics server must publish the average number of connections in the FEP cluster under this name.</td>
</tr>
<tr>
<td>spec.scaleout.stabilizationWindowSeconds</td>
<td>0</td>
<td>If the duration (seconds) threshold of this parameter has been exceeded continuously, a scale out is performed.</td>
</tr>
<tr>
<td>spec.limits.maxReplicas</td>
<td>2</td>
<td>Maximum number of replicas (0 to 15)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the value is out of range, no automatic scale out is performed.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiVersion</td>
<td>fep.fujitsu.io/v1</td>
<td>Fixed</td>
</tr>
<tr>
<td>kind</td>
<td>FEPUpgrade</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.name</td>
<td>Same as FEPClusterCR</td>
<td>Fixed</td>
</tr>
<tr>
<td>metadata.namespace</td>
<td>Same as FEPClusterCR</td>
<td>Fixed</td>
</tr>
<tr>
<td>spec.upgrade</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.upgrade.sourceCluster</td>
<td></td>
<td>Specifies the FEPClusterCR name from which to migrate data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Required.</td>
</tr>
<tr>
<td>spec.upgrade.mcSpec.limits</td>
<td>cpu: 200m</td>
<td>Optional</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.upgrade.mcSpec.requests</td>
<td>cpu: 100m, memory: 200Mi</td>
<td>Specifies the maximum number of resources to allocate to the upgrade execution container.</td>
</tr>
<tr>
<td>spec.upgrade.image</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>spec.upgrade.imagePullPolicy</td>
<td>IfNotPresent</td>
<td>Specifies the pull policy for the container image.</td>
</tr>
<tr>
<td>spec.upgrade.source.pgAdminTls.certificateName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>spec.upgrade.destination.pgAdminTls.certificateName</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>spec.upgrade.storage</td>
<td></td>
<td>Defines the storage for storing dump files.</td>
</tr>
<tr>
<td>spec.upgrade.storage.storageClass</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>spec.upgrade.storage.size</td>
<td>2Gi</td>
<td>Specifies the size of the storage to store the dump file.</td>
</tr>
<tr>
<td>spec.upgrade.storage.accessModes</td>
<td>ReadWriteOnce</td>
<td>Optional</td>
</tr>
</tbody>
</table>

If you do not define the 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 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.
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:

```yaml
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:

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Required/ Optional</th>
<th>Change Effect</th>
<th>Updating value allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepLogging.image.image</td>
<td>Optional</td>
<td>Fluentd Image of FEPLogging</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.image.pullPolicy</td>
<td>Required</td>
<td>Fluentd Image pull policy of FEPLogging</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.mcSpec.limits.cpu</td>
<td>Required</td>
<td>Max CPU allocated to fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.mcSpec.limits.memory</td>
<td>Required</td>
<td>Max memory allocated to fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.mcSpec.requests.cpu</td>
<td>Required</td>
<td>CPU allocation at start for fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.mcSpec.requests.memory</td>
<td>Required</td>
<td>Memory allocation at start for fluentd container</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.sysExtraLogging</td>
<td>Required</td>
<td>To turn on extra debugging messages for operator, set value to true. It can be turned on/off at any time</td>
<td>Yes</td>
</tr>
<tr>
<td>Custom Resource spec</td>
<td>Required/Optional</td>
<td>Change Effect</td>
<td>Updating value allowed</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>spec.fepLogging.restartRequired</td>
<td>Required</td>
<td>To restart FEPLogging instance for applying any new configuration for example after certificate rotation</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.scraperInterval</td>
<td>Optional</td>
<td>Scrape interval for Prometheus to fetch metrics from FEPLogging instance</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.scraperTimeout</td>
<td>Optional</td>
<td>Scrape Timeout for Prometheus to fetch metrics from FEPLogging instance</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.host</td>
<td>Optional</td>
<td>Target Elasticsearch host name</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.port</td>
<td>Optional</td>
<td>Target Elasticsearch port number</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.authSecret.secretName</td>
<td>Optional</td>
<td>Secret name which contains Elasticsearch authentication username &amp; password</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.authSecret.userKey</td>
<td>Optional</td>
<td>Username key specified in Elasticsearch authentication secret</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.authSecret.passwordKey</td>
<td>Optional</td>
<td>Password key specified in Elasticsearch authentication secret</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.logstashPrefix</td>
<td>Optional</td>
<td>Logstash prefix to differentiate index pattern in elastic search. Default value is postgres</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.scheme</td>
<td>Optional</td>
<td>Connection scheme between FEPLogging &amp; Elasticsearch. Possible options http &amp; https</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.sslVerify</td>
<td>Optional</td>
<td>Set to true if you want to verify ssl certificate. If set to false then will not consider TLS certificate</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds fluentd certificate</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.elastic.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Elasticsearch to verify Elasticsearch TLS connection</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds Fluentd certificate</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.fepLogging.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.prometheus.tls.certificateName</td>
<td>Optional</td>
<td>Kubernetes secret name which holds Prometheus certificate</td>
<td>Yes</td>
</tr>
<tr>
<td>spec.prometheus.tls.caName</td>
<td>Optional</td>
<td>Kubernetes configmap which holds cacert of Fluentd to configure MTLS between FEPLogging &amp; Prometheus</td>
<td>Yes</td>
</tr>
</tbody>
</table>

1.2.15 FEP Custom Resources - spec.fep.pgBadger

<table>
<thead>
<tr>
<th>Custom Resource spec</th>
<th>Change Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgBadger.schedules.create</td>
<td>The 'create' schedule to create report and upload it to endpoint</td>
</tr>
<tr>
<td>pgBadger.schedules.cleanup</td>
<td>The 'cleanup' schedule to delete the report left in container</td>
</tr>
<tr>
<td>pgBadger.options.incremental</td>
<td>Default: false; When set to True: create incremental report in pgbadger</td>
</tr>
<tr>
<td>pgBadger.endpoint.authentication</td>
<td>a secret to contain authentication info to access endpoint</td>
</tr>
<tr>
<td>Custom Resource spec</td>
<td>Change Effect</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>support basic auth only</td>
<td></td>
</tr>
<tr>
<td>pgBadger.endpoint.customCertificateName</td>
<td>Client certificate reference in customCertificate CR</td>
</tr>
</tbody>
</table>
| 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         |
Appendix A Default Metrics Queries

pg_capacity_connection:
query:

```sql
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:

```sql
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 nullhdr2
FROM pg_stats s,
(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)
GROUP BY 1,2,3,4,5
) AS foo)
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.'
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
)
select * from pg_catalog.pg_database
) select dbs.datname, coalesce(locks.locks, 0) locks from dbs left join locks on dbs.oid=DATABASE

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

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()

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,
average_tup_count,
average_tup_delete_count,
average_tup_hot_update_count,
average_tup_insert_count,
average_tup_update_count,
average_distribute_count,
autoanalyze_count

metrics:
  - datname:
      usage: 'LABEL'
      description: 'Database name'
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"
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'
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

```yaml
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
1. **Prerequisites**
   - Registered OpenShift cluster with Red Hat Marketplace
   - Buy or try the product ‘FUJITSU Enterprise Postgres for Kubernetes ’ from Red Hat Marketplace

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.

<table>
<thead>
<tr>
<th>No</th>
<th>CPU architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>x86</td>
</tr>
<tr>
<td>2</td>
<td>s390x</td>
</tr>
</tbody>
</table>

2.2. **Supported Platform**

It supports running on the following platforms.

<table>
<thead>
<tr>
<th>No</th>
<th>Platform</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OpenShift Container Platform</td>
<td>4.6, 4.7, 4.8, 4.9</td>
</tr>
</tbody>
</table>

3. **Operator installation from Red Hat Marketplace**

1. For information on registering your cluster and creating a namespace, see [Red Hat Marketplace Docs](https://marketplace.redhat.com/en-us/documentation/clusters#register-openshift-cluster-with-red-hat-marketplace). 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.
Quick Start Guide
FUJITSU Enterprise Postgres for Kubernetes

- 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**
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](#)