FUJITSU Enterprise Postgres 13 for Kubernetes

Manual Set

Getting Started
- Release Notes >
- Overview >
- Quick Start Guide >

Reference
- User’s Guide >
- Reference >
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</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>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 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 4.0: March 2022
Edition 3.0: December 2021
Edition 2.0: October 2021
Edition 1.0: September 2021

Copyright

Copyright 2021-2022 FUJITSU LIMITED
Chapter 1 New Features and Improvements

1.1 Features Added FEP13 Operator in v3.1.2

1.2 Features Added FEP13 Operator in v3.1.1

1.3 Features Added FEP13 Operator in v3.1.0

1.4 Additional Notes
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 13 Operator</td>
<td>OSS</td>
<td>PostgreSQL Rebase</td>
</tr>
<tr>
<td>image tag:v3.1.2</td>
<td>Platform enhancement</td>
<td>Additional OCP Support</td>
</tr>
<tr>
<td>Container image tag:ubi8-13-1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEP 13 Operator</td>
<td>OSS</td>
<td>PostgreSQL Rebase</td>
</tr>
<tr>
<td>image tag:v3.1.1</td>
<td>Platform enhancement</td>
<td>Additional OCP Support</td>
</tr>
<tr>
<td>Container image tag:ubi8-13-1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FEP 13 Operator</td>
<td>OSS</td>
<td>PostgreSQL Rebase</td>
</tr>
<tr>
<td>image tag:v3.1.0</td>
<td>Platform enhancement</td>
<td>Additional OCP Support</td>
</tr>
<tr>
<td>Container image tag:ubi8-13-1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Operation</td>
<td>OCP Level4 for Monitoring and Alerting Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCP Level4 for Event Notification Feature</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OCP Level5 for Auto Scaling Read Replicas</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Logical Replication Support</td>
</tr>
</tbody>
</table>

1.1 Features Added in FEP13 Operator in v3.1.2

This section explains new features and improvements in FUJITSU Enterprise Postgres for Kubernetes v3.1.2.

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 13.6.

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

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

See

1.2 Features Added in FEP13 Operator in v3.1.1

This section explains new features and improvements in FUJITSU Enterprise Postgres for Kubernetes v3.1.1.

1.2.1 OSS

This section explains the new feature related to OSS:
- PostgreSQL rebase

1.2.1.1 PostgreSQL Rebase

The PostgreSQL version that FUJITSU Enterprise Postgres is based on is 13.5.

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

1.2.2 Platform Enhancement

This section explains the new features related to platform enhancement:
- Additional OCP support

1.2.2.1 Additional OCP Support

The following additional OCP is supported:
- OCP 4.8

See

1.3 Features Added in FEP13 Operator in v3.1.0

This section explains new features and improvements in FUJITSU Enterprise Postgres for Kubernetes v3.1.0.

1.3.1 OSS

This section explains the new feature related to OSS:
1.3.1.1 PostgreSQL Rebase

The PostgreSQL version that FUJITSU Enterprise Postgres is based on is 13.3.

See

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

1.3.2 Platform Enhancement

This section explains the new features related to platform enhancement:

- Additional OCP support
- Additional OCS support

1.3.2.1 Additional OCP Support

The following additional OCP is supported:

- OCP 4.7

See


1.3.2.2 Additional OCS Support

Supports OpenShift Container Storage for storage.

See


1.3.3 Operation

This section explains the new feature related to operation:

- OCP Level4 for Monitoring and Alerting feature
- OCP Level4 for Event Notification feature
- OCP Level5 for Auto Scaling Read Replicas
- Logical Replication support

1.3.3.1 OCP Level4 for Monitoring and Alerting Feature

Provides operator and operand (i.e. FEPCluster) monitoring and alerting as required for OCP Level 4 functionality.

See

Refer to "Monitoring" in the User's Guide for details.
1.3.3.2 OCP Level4 for Event Notification Feature
Provides notification of custom resource creation event as required for OCP Level 4 functionality.

See
Refer to "Event Notification" in the User’s Guide for details.

1.3.3.3 OCP Level5 for Auto Scaling Read Replicas
Provides the auto-scale-out capability of read replicas required for OCP level 5 functionality.

See
Refer to "Scaling Replicas" in the User’s Guide for details.

1.3.3.4 Logical Replication Support
New support for logical replication as a replication method.

See
Refer to "Replication slots" in the User’s Guide for details.

1.4 Additional Notes
Please note the following:
- There is a change in the default value.
  - Exclude pgaudit and vci from 'shared_preload_libraries'
  - Set the default backup retention date (7 days)
- Enable SSL parameters for PGPool2 by default
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 FUJITSU Enterprise Postgres</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td>CR</td>
</tr>
<tr>
<td>Custom Resource</td>
<td>CRD</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>
<tbody>
<tr>
<td>FUJITSU Software Enterprise PostgreSQL for Kubernetes</td>
<td>User's Guide</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.

**Issue date and version**

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0</td>
<td>March 2022</td>
</tr>
<tr>
<td>3.0</td>
<td>December 2021</td>
</tr>
<tr>
<td>2.0</td>
<td>October 2021</td>
</tr>
<tr>
<td>1.0</td>
<td>September 2021</td>
</tr>
</tbody>
</table>

**Copyright**

Copyright 2021-2022 FUJITSU LIMITED
# Contents

**Chapter 1 Know about the Product**

1.1 What is FUJITSU Software Enterprise PostgreSQL for Kubernetes?................................................................. 1
1.2 Operator Features....................................................................................................................................................... 1
   1.2.1 Cluster Deployment............................................................................................................................................ 2
   1.2.1.1 Creating a FEP Cluster................................................................................................................................. 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
1.3 Operator System Configuration................................................................................................................................. 6

**Chapter 2 Know What it does**........................................................................................................................................... 8

2.1 Deployment................................................................................................................................................................. 8
2.2 High Availability (Automatic failover and recovery).............................................................................................. 8
2.3 Configuration Change.................................................................................................................................................. 8
2.4 Minor Version Upgrade............................................................................................................................................. 9
2.5 Configurable Volume Per Cluster............................................................................................................................... 9
2.6 Deploying Pgpool-II and Connect to FEP Cluster from Operator.......................................................................... 9
2.7 Scheduling Backup from Operator............................................................................................................................ 9
2.8 Perform PITR and Latest Backup Restore from Operator........................................................................................ 10
2.9 Monitoring & Alert................................................................................................................................................... 11
2.10 Scaling Replicas....................................................................................................................................................... 12

**Appendix A OSS Supported by FUJITSU Enterprise PostgreSQL for Kubernetes**.......................................................................................................................... 13
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 13 on OpenShift Container Platform.

There are multiple components in the solution.

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

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

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

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

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

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

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

This operator will deploy a standalone as well as highly available FUJITSU Enterprise Postgres cluster with pre-defined configuration to get started with small workload. User can adjust the configuration parameters at the time of deployment and after to make the instance suitable for the workload.

As the name implies, the FEP server container is intended to incorporate the FUJITSU Enterprise Postgres server component.

In principle, a running FEP server container is considered as equivalent to a FUJITSU Enterprise Postgres Server instance.

1.2 Operator Features

This product provides operator services to automate the construction and operation of databases on the customer's container management infrastructure. The features of the operator are as follows:
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 synchronous 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 FEPCfg 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 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.
The minor version upgrade will take effect immediately. There is a momentary outage on the cluster and the users should perform this action at a time that has least disruption to the service.

1.2.6 FEP Features

1.2.6.1 Scope of FEP Feature Support

These features also require the FEP Client ISO.

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.

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.

1.3 Operator System Configuration
The basic relationships among POD, containers and services are as follows.

Example) Deployment with Pgpool-II
In this deployment scenario, Pgpool-II is used to provide connection pooling and load balancing. End user application will point its connection to Pgpool service. Depending on the transaction type, Pgpool will forward the connection to either the Master Pod or the Replica Pod. If a failover/switchover occurs, the FEP pgpool2 will direct traffic to the new FEP master Pod. This is transparent to the end user application.
Example) Deployment without Pgpool-II

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

This chapter explains what you need to do.

2.1 Deployment

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

2.2 High Availability (Automatic failover and recovery)

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

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

2.3 Configuration Change

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

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

2.5 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 tablescape. 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 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.
2.8 Perform PITR and Latest Backup Restore from Operator

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

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

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

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

Prometheus is a condensed way to store time-series metrics. Grafana provides a flexible and visually pleasing interface to view graphs and 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 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.
## Appendix A  OSS Supported by FUJITSU Enterprise Postgres for Kubernetes

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>13.6</td>
<td>Database management system</td>
<td>PostgreSQL Documentation</td>
</tr>
<tr>
<td>orafce</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>2.3.0</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>12.1</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>1.4.6</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>11.5</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>1.2-20200228</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</td>
<td>42.2.18</td>
<td>JDBC driver</td>
<td>“JDBC Driver” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>driver</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>psqlODBC</td>
<td>13.00.0000</td>
<td>ODBC driver</td>
<td>“ODBC Driver” in the FUJITSU Enterprise Postgres Application Development Guide</td>
</tr>
<tr>
<td>pgBackRest</td>
<td>2.30</td>
<td>Backup and restore management</td>
<td>“Scheduling Backup from Operator” in the User’s Guide</td>
</tr>
<tr>
<td>patroni</td>
<td>2.1.0</td>
<td>Postgres cluster management</td>
<td>“High Availability” in the User's Guide</td>
</tr>
<tr>
<td>Python</td>
<td>3.7.5</td>
<td>PL/Python</td>
<td></td>
</tr>
<tr>
<td>postgres-exporter</td>
<td>0.9.0</td>
<td>Postgresql metrics monitoring capabilities for Prometheus with Fujitsu updated queries</td>
<td>“Monitoring” in the User's Guide</td>
</tr>
</tbody>
</table>
FUJITSU Enterprise Postgres 13 for Kubernetes

Quick Start Guide
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. This must be done prior to operator install.

2. On the main menu, click Workspace, click Software, click on the product box of 'FUJITSU Enterprise Postgres for Kubernetes ', and then click Install Operator.

3. On the Update Channel section, select an option.

4. On the Approval Strategy section, select either Automatic or Manual. The approval strategy corresponds to how you want to process operator upgrades.

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

6. Click **Install**. It may take several minutes for installation to complete.

7. Once installation is complete, the status will change from **installing** to **Up to date**.

8. For further information, see the [Red Hat Marketplace Operator documentation](#).

---

**4. Verification of operator installation**

1. Once status changes to **Up to date**, click the vertical ellipsis and select **Cluster Console**.

2. Open the cluster where you installed the product

3. Go to **Operators > Installed Operators**

4. Select the Namespace or Project you installed on

5. Verify status for product is **Succeeded**
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:

![FEPCluster CR YAML](image)

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 13 for Kubernetes Manuals
FUJITSU   Enterprise Postgres 13 for Kubernetes

User's Guide
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 or FUJITSU</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td>Enterprise Postgres</td>
</tr>
<tr>
<td>Vertical Clustered Index</td>
<td>VCI</td>
</tr>
<tr>
<td>Transparent Data Encryption</td>
<td>TDE</td>
</tr>
<tr>
<td>Point in time recovery</td>
<td>PITR</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>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 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 5.0: March 2022
Edition 4.0: December 2021
Edition 3.0: November 2021
Copyright

Copyright 2021-2022 FUJITSU LIMITED
## Contents

### Chapter 1 System Requirements

1.1 Components Embedded................................................................. 1
1.2 CPU............................................................................................... 1
1.3 Supported Platform....................................................................... 1
1.4 Collaboration Tool......................................................................... 1

### Chapter 2 Overview of Operator Design

2.1 Design Task.................................................................................. 2
2.2 System Configuration Design........................................................ 2
2.2.1 Server Configuration............................................................... 2
2.2.2 User Account......................................................................... 4
2.2.3 Basic Information of the Container......................................... 4
2.3 Design Perspective for Each Feature............................................ 6
2.3.1 Deployment................................................................. 7
2.3.2 High Availability............................................................. 7
2.3.3 Configurable Volume per Cluster......................................... 8
2.3.4 Deploying Pgpool-II and Connect to FEPCluster from Operator... 9
2.3.5 Scheduling Backup from Operator..................................... 11
2.3.5.1 Important Setting Items......................................................... 12
2.3.5.2 Parameters that cannot be Set.............................................. 12
2.3.5.3 Restricted Parameters....................................................... 15
2.3.5.4 About Sections in the Config File.................................. 15
2.3.6 Perform PITR and Latest Backup Restore from Operator...... 15
2.3.7 FEP Unique Feature Enabled by Default............................. 15
2.3.8 Monitoring & Alert (FEPExporter)......................................... 16
2.3.8.1 FEPExporter Custom Resource......................................... 16
2.3.8.2 Change to FEPCluster CR - metrics user......................... 16
2.3.8.3 FEPExporter CR auto-create for FEPCluster.................. 16
2.3.9 Scaling Replicas................................................................. 16
2.3.9.1 Change to FEPCluster CR - autoscale......................... 17

### Chapter 3 Operator Installation

3.1 Pre-requisite................................................................................. 18
3.2 Deploying Operator................................................................. 19
3.3 Implement Collaborative Monitoring Tools............................... 21

### Chapter 4 Deployment Container

4.1 Deploying FEPCluster using Operator......................................... 23
4.2 Deploy a Highly Available FEPCluster........................................ 28
4.3 Deploying FEPExporter............................................................. 30
4.4 FEPExporter in Standalone Mode............................................. 31
4.5 Configuration FEP to Perform MTLS........................................ 33
4.5.1 Manual Certificate Management......................................... 34
4.5.2 Automatic Certificate Management..................................... 57
4.5.3 Deploy FEPCluster with MTLS support......................... 40
4.5.4 Configurable Parameters....................................................... 47
4.6 Replication Slots................................................................. 50
4.6.1 Setting Up Logical Replication using MTLS.................. 50

### Chapter 5 Post-Deployment Operations

5.1 Configuration Change............................................................... 53
5.2 FEPPGPool2 Configuration Change........................................ 44
5.3 Scheduling Backup from Operator......................................... 55
5.4 Configure MTLS Setting............................................................. 56
5.4.1 Certification Rotation........................................................... 56
5.5 Monitoring................................................................................. 57
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>13.6</td>
<td>To provide server capabilities</td>
</tr>
<tr>
<td>3</td>
<td>Patroni</td>
<td>2.1.0</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>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OpenShift Container Platform</td>
<td>4.6, 4.7, 4.8, 4.9</td>
</tr>
<tr>
<td>2</td>
<td>OpenShift Container Storage</td>
<td>4.6, 4.7, 4.8</td>
</tr>
</tbody>
</table>

Supports storage supported by OpenShift.

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

1.4 Collaboration Tool

Supports integration with the following tools for monitoring and alerting.

<table>
<thead>
<tr>
<th>No</th>
<th>Platform</th>
<th>Version</th>
<th>How to obtain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Prometheus</td>
<td>The version installed OpenShift</td>
<td>Preinstalled with OpenShift</td>
</tr>
<tr>
<td>2</td>
<td>AlertManager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Grafana</td>
<td>3.10.3 and later</td>
<td>Provided by OperatorHub</td>
</tr>
</tbody>
</table>
Chapter 2 Overview of Operator Design

This chapter describes an overview of the operator design.

2.1 Design Task

This section describes the operation of FEP.

First, determine the configuration. You then design each feature and deploy the container. You can use FEP features immediately after deployment.

<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 FEP Cluster 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 restore container</td>
<td>A container that performs the restore operation. Temporarily created during a restore operation.</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 Exporter container</td>
<td>A container that exposes http/https endpoint for monitoring stats scraping.</td>
</tr>
<tr>
<td>Backup storage</td>
<td>Storage where backup data is stored. If you do not need to obtain a backup, you do not need to create one.</td>
</tr>
<tr>
<td>FEP Cluster</td>
<td>Parent CR for FEP Cluster definition and configuration.</td>
</tr>
<tr>
<td>FEPBackup</td>
<td>Child CR for backup configuration.</td>
</tr>
<tr>
<td>FEPVolume</td>
<td>Child CR for volumes.</td>
</tr>
<tr>
<td>FEPConfig</td>
<td>Child CR for FEP configurations.</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><em>OS</em></td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>FEPBaseVersion</td>
<td>13</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-13-server:ubi8-13-1.0
  - fujitsu-enterprise-postgres-13-server:ubi8-13-1.0-amd64
  - fujitsu-enterprise-postgres-13-server:ubi8-13-1.0-s390x
FEP backup container

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


For each \textit{Version}, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{OS}</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>\textit{FEPBaseVersion}</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>\textit{MajorVersion}</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>\textit{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)

- \texttt{fujitsu-enterprise-postgres-13-backup:ubi8-13-1.0}
  - \texttt{fujitsu-enterprise-postgres-13-backup:ubi8-13-1.0-amd64}
  - \texttt{fujitsu-enterprise-postgres-13-backup:ubi8-13-1.0-s390x}

FEP restore container

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


For each \textit{Version}, specify the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{OS}</td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td>\textit{FEPBaseVersion}</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>\textit{MajorVersion}</td>
<td>1,2, ...</td>
<td>To be used when major change in image, including server patch application</td>
</tr>
<tr>
<td>\textit{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)

- \texttt{fujitsu-enterprise-postgres-13-restore:ubi8-13-1.0}
  - \texttt{fujitsu-enterprise-postgres-13-restore:ubi8-13-1.0-amd64}
  - \texttt{fujitsu-enterprise-postgres-13-restore:ubi8-13-1.0-s390x}

FEP pgpool2 container

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


For each \textit{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>13</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-13-pgpool2:ubi8-13-1.0
  - fujitsu-enterprise-postgres-13-pgpool2:ubi8-13-1.0-amd64
  - fujitsu-enterprise-postgres-13-pgpool2:ubi8-13-1.0-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><strong>OS</strong></td>
<td>ubi8</td>
<td></td>
</tr>
<tr>
<td><strong>FEPBaseVersion</strong></td>
<td>13</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-13-exporter:ubi8-13-1.0
  - fujitsu-enterprise-postgres-13-exporter:ubi8-13-1.0-amd64
  - fujitsu-enterprise-postgres-13-exporter:ubi8-13-1.0-s390x

### 2.3 Design Perspective for Each Feature

This section describes the design of each feature.

**postgresql-cfg format**

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

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

<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>Spec</td>
<td>Example</td>
<td>Comment</td>
</tr>
<tr>
<td>-----</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>The value cannot have space unless quoted.</td>
<td>foo1=bar1</td>
<td></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>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 foo=bar #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 foo2=bar2</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 setting correctly.

### 2.3.1 Deployment

**Information for the FEPCluster**

Equivalent Kubernetes command: kubectl apply -f FEPClusterCR.yaml

This operation will create a FEPCluster with supplied information in FEPClusterCR.yaml. Refer to "FEPCluster parameter" in the Reference for details.

### 2.3.2 High Availability

Describes the settings for using the highly available features.

**Arbitration**

Patroni is used to control and monitor FEP instance startup, shutdown, status and trigger failover should the master instance fails. It plays a significant role in the solution. If the Patroni process dies, especially on master POD, without notice, the POD will not update the Patroni cluster lock. This may trigger an unwanted failover to one of the Replicas, 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
```
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>Field</td>
<td>Mandatory</td>
<td>Sub-Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>-----------</td>
<td>--------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>accessModes</td>
<td></td>
<td></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></td>
<td></td>
<td></td>
</tr>
<tr>
<td>storageClass</td>
<td></td>
<td></td>
<td>Defaults to platform default if omitted</td>
<td>SC is only set at start</td>
</tr>
<tr>
<td>accessModes</td>
<td></td>
<td></td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td>Access mode 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.

<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 you want to deploy the operator.</td>
</tr>
<tr>
<td>spec.image</td>
<td>-</td>
<td>Specifies the FEP pgpool2 container image to provide.</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>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>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>spec.custompcp</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
</tbody>
</table>
spec.customsslkey | "|" | If you want to do it, "|" and the Beethoven key content in the line below.
---|---|---
spec.customsslcert | "|" | If you want to do it, "|" and the contents of the public x 509 certificate from the line below.
---|---|---
spec.customsslcacert | "|" | If you want to do it, "|" and the following lines describe the contents of the CA root certificate in PEM format.
---|---|---
spec.customlogsize | 100 Mi | Specifies the persistent volume size for log output.
---|---|---
spec.storageclassname | - | Specifies the storage class for log output.

**Pgpool-II parameters**

The parameters that can be specified are shown in the table below. For details on the parameters, refer to the Pgpool-II manual.

<table>
<thead>
<tr>
<th>Category</th>
<th>Parameter name (Specified format)</th>
<th>Restart required after change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection settings</strong></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><strong>Authentication settings</strong></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><strong>Backend settings</strong></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><strong>Connection pooling</strong></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><strong>Error reporting and log acquisition</strong></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>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></td>
<td>log_errorverbosity (enum)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_line_prefix (string)</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 Socket 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>

### 2.3.5 Scheduling Backup from Operator

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

A backup definition includes the following:

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

Backup is taken on master POD only.

Backup processing is performed by pgBackRest.
Parameter can be set to pgbackrestParams in CR definition.
The maximum number of backup schedules is 5.
See the pgBackRest User's Guide for details on the parameters.
However, some parameters are limited. Details are given below.

- 2.3.5.1 Important Setting Items
- 2.3.5.2 Parameters that cannot be Set
- 2.3.5.3 Restricted Parameters
- 2.3.5.4 About Sections in the Config File

### 2.3.5.1 Important Setting Items

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

<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</td>
<td>natural number</td>
</tr>
<tr>
<td></td>
<td>No default (should be set according to user backup policy)</td>
<td></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)</td>
<td>time/count</td>
</tr>
<tr>
<td></td>
<td>No default (should be set according to user backup policy)</td>
<td></td>
</tr>
</tbody>
</table>

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>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>Parameter</td>
<td>Overview of parameters</td>
<td>Reason</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>SSH client command Option (<code>--cmd-ssh</code>)</td>
<td>Path to ssh client executable</td>
<td>Not using ssh</td>
</tr>
<tr>
<td>Compress Option (<code>--compress</code>)</td>
<td>Use File Compression</td>
<td>For obsolete options (Use compresse-type option instead)</td>
</tr>
<tr>
<td>Delta Option (<code>--delta</code>)</td>
<td>Restore or Backup with Checksum</td>
<td>For new restores only</td>
</tr>
<tr>
<td>Lock Path Option (<code>--lock-path</code>)</td>
<td>Path where the lock file is stored</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Keep Alive Option (<code>--sck-keep-alive</code>)</td>
<td>Enable keep-alive messages on socket connections</td>
<td>To use internal fixed values</td>
</tr>
<tr>
<td>Spool Path Option (<code>--spool-path</code>)</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>Console Log Level Option (<code>--log-level-console</code>)</td>
<td>Console Log Level</td>
<td>It is not expected to operate on POD.</td>
</tr>
<tr>
<td>Std Error Log Level Option (<code>--log-level-stderr</code>)</td>
<td>Stderr log level</td>
<td>It is not expected to operate on POD.</td>
</tr>
<tr>
<td>Log Path Option (<code>--log-path</code>)</td>
<td>Log File Destination</td>
<td>For automatic determination from FEPCluster CR values</td>
</tr>
<tr>
<td>Azure Repository Account Option (<code>--repo-azure-account</code>)</td>
<td>Azure account used to store the repository</td>
<td>Azure storage is not supported</td>
</tr>
<tr>
<td>Azure Repository TLS CA File Option (<code>--repo-azure-ca-file</code>)</td>
<td>Use a non-default CA file for the Azure Repository TLS CA file system</td>
<td></td>
</tr>
<tr>
<td>Azure Repository TLS CA Path Option (<code>--repo-azure-ca-path</code>)</td>
<td>Use non-default CA path for Azure Repository TLS CA path system</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Container Option (<code>--repo-azure-container</code>)</td>
<td>Azure repository container. Azure container used to store the repository.</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Host Option (<code>--repo-azure-host</code>)</td>
<td>Azure Repository Host</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Key Option (<code>--repo-azure-key</code>)</td>
<td>Azure Repository Shared Key or Shared Access Signature</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Key Type Option (<code>--repo-azure-key-type</code>)</td>
<td>Azure Repository Key Type</td>
<td></td>
</tr>
<tr>
<td>Azure Repository Server Port Option (<code>--repo-azure-port</code>)</td>
<td>Azure Repository Server Port</td>
<td></td>
</tr>
<tr>
<td>Repository Host Option (<code>--repo-host</code>)</td>
<td>Repository host for remote operations via SSH</td>
<td>Repository Host is not used</td>
</tr>
<tr>
<td>Repository Host Command Option (<code>--repo-host-cmd</code>)</td>
<td>Path of pgBackRest on Repository Host</td>
<td></td>
</tr>
<tr>
<td>Repository Host Configuration Option (<code>--repo-host-config</code>)</td>
<td>Repository Host Configuration File Path</td>
<td></td>
</tr>
<tr>
<td>Repository Host Configuration Include Path Option (<code>--repo-host-config-include-path</code>)</td>
<td>Repository hosts configuring include path</td>
<td></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>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>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>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 tablespace required because there is only one tablespace per FEPCluster</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 Port Specification</td>
<td>No SSH connection required</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>Reason</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
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:

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

The user can overwrite these values in config map.

TDE

TDE is enabled by default. For details on how to specify the passphrase, refer to “FEPCluster parameter” in the Reference.

2.3.8 Monitoring & Alert (FEPExporter)

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

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

FEPExporter container by default is configured to extract useful database statistics and make the 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 CPU utilization of the DB container exceeds the threshold.

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

- Transparent Data Encryption (TDE)
Note

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

2.3.9.1 Change to FEPCluster CR - autoscale

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

Refer to "FEPCluster Parameter" in the Reference for FEPCluster parameters.
Chapter 3 Operator Installation

This chapter describes how to install FEP operator in a new namespace on openshift.

3.1 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.

(Images of Create Project dialog)

Note

FEP13 Operator installation needs Prometheus to be pre-installed in the Openshift cluster.
3.2 Deploying Operator

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

On OpenShift platform, logon with credentials that has privileges to install operator. Click on OperatorHub on menu item under Operators and type filter keyword FUJITSU to find FUJITSU Enterprise Postgres 13 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 in cluster" and choose desired namespace. Leave everything else to default and click install.
3.3 Implement Collaborative Monitoring Tools

There is a pre-requisite for running FEPExporter.

- GAP(Grafana, AlertManager, Prometheus) stack is installed on host OpenShift 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 with Openshift. Please refer to the following for deployment information.

(Refer: https://docs.openshift.com/container-platform/4.6/monitoring/understanding-the-monitoring-stack.html)

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.

(Refer: https://fast.fujitsu.com/knowledge-base/knowledge-articles-installationsetup/setting-up-grafana-on-ibm-linuxone)
(Refer: https://fast.fujitsu.com/knowledge-base/knowledge-articles-installationsetup/setting-up-grafana-on-amd64-ocp)

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

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

3. This will bring to "Create FEPCluster" page. Here you have two options to configure. The first one is Form View. At the moment, in Form View, one can change only the name of cluster being deployed. The default name is 'new-fep'. This name must be unique within a namespace.
4. In YAML View, starting value of CR is visible and one can choose to modify parameters before creating CR. Refer to the Reference for details of parameters.

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

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

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.
4.3 Deploying FEPExporter

To deploy a FEPExporter, follow these steps:

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

![Diagram showing monitoring section area](image)

**Note**

- User can set fep.monitoring.enable to true or false on an already instantiated cluster as well to achieve desired results.
- pgMetricsUser can be defined later on a running FEPCluster with monitoring enabled and can force FEPExporter to use pgMetricsUser by mere restarting it (refer restartRequired). However, MTLS can not be configured in this case and user is expected to grant specific permission to pgMetricsUser for all the database objects which are expected to be use while scraping information.
- For MTLS to be forced, ensure usePodName and pg_hba.conf is been set appropriately.
- FEPExporter default metrics expects few following in postgresql.conf
  - pg_stats_statements library pre-loaded
  - track_activities and track_counts are turned on
  - Monitoring user needs permission on pg_stat_* views
- FEPExporter pod specification related to CPU memory can be changed. After changing resources specification, set `restartRequired` flag to true. FEPexporter will be restarted with new specifications.
- FEP Monitoring is closely integrated with Prometheus available on platform. User should ensure that on openshift platform monitoring is enabled for user-defined projects (Refer: https://docs.openshift.com/container-platform/4.6/monitoring/enabling-monitoring-for-user-defined-projects.html). For platforms other than openshift, ensure Prometheus is installed before deployment of FEP operator

### 4.4 FEPExporter in Standalone Mode

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

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

### 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)
```
```
e is 65537 (0x010001)
Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv
Verifying - Enter pass phrase for myca.key: 0okm9ijn8uhb7ygv

```

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

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

2. Create Configmap to store CA certificate

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

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

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

Creating a Server Certificate

4. Create FEP Server private key

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

5. Create FEP Server certificate signing request

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

```shell
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))  # all in one line
```

---

**Note**

The cluster name and namespace must be changed appropriately.

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

---

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

```shell
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.

```shell
openssl genrsa -out patroni.key 2048
```

```shell
generating RSA private key, 2048 bit long modulus
...............................................+++
.......+++
e is 65537 (0x10001)
```

9. Create certificate signing request for Patroni

```shell
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
```

```shell
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

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

11. Create TLS secret to store Patroni certificate and key

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

Creating a User Certificate

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

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

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

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

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

14. Create client certificate for "postgres" user

```
openssl x509 -req -in postgres.csr -CA myca.pem -CAkey myca.key -out postgres.pem -days 365
```

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

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

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

4.5.2 Automatic Certificate Management

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

Note

Note that certificates created in this example are not password protected.
Install cert-manager

```bash
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.

```yaml
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

```yaml
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

```yaml
cat << EOF | oc apply -f -
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: ca-issuer
  namespace: my-namespace
spec:
  ca:
    secretName: cacert
EOF
```
Create FEP Server certificate using above CA Issuer

Assuming FEPCluster name is mydb in namespace my-namespace.

```yaml
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.

```yaml
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
  secretName: mydb-patroni-cert
  issuerRef:
    name: ca-issuer
EOF
```
Create postgres user client certificate

```yaml
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

```yaml
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 rewinduser user client certificate

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

```yaml
apiVersion: fep.fujitsu.io/v2
kind: FEPCluster
metadata:
  name: mydb
  namespace: my-namespace
spec:
  fep:
    usePodName: true
  patroni:
    tls:
      certificateName: mydb-patroni-cert
      caName: cacert
  postgres:
    tls:
      certificateName: mydb-fep-cert
      caName: cacert
      privateKeyPassword: mydb-fep-private-key-password
    forceSsl: true
    podAntiAffinity: false
  mcSpec:
    limits:
      cpu: 500m
      memory: 700Mi
    requests:
      cpu: 200m
      memory: 512Mi
    customAnnotations:
      allDeployments: {}
    servicePort: 27500
    image:
      image: 'quay.io/fujitsu/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_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
goAdminTls:
certificateName: mydb-postgres-cert
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
category: 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'
```

----- END CERTIFICATE-----
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_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'
<table>
<thead>
<tr>
<th>wal_level</th>
<th>replica</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_wal_senders</td>
<td>12</td>
</tr>
<tr>
<td>wal_keep_segments</td>
<td>64</td>
</tr>
</tbody>
</table>

### Storage

<table>
<thead>
<tr>
<th>DataVol</th>
<th>2Gi</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td></td>
</tr>
<tr>
<td>storageClass</td>
<td>nfs-client</td>
</tr>
<tr>
<td>walVol</td>
<td>1200Mi</td>
</tr>
<tr>
<td>size</td>
<td></td>
</tr>
<tr>
<td>storageClass</td>
<td>nfs-client</td>
</tr>
<tr>
<td>logVol</td>
<td>1Gi</td>
</tr>
<tr>
<td>size</td>
<td></td>
</tr>
<tr>
<td>storageClass</td>
<td>nfs-client</td>
</tr>
</tbody>
</table>

### 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

- **tdePassphrase**: tde-passphrase

### SystemCertificates

- **key**: |
- **-----BEGIN RSA PRIVATE KEY-----**

```
MIIeowIBAAKCAQEAOADFKImah8CIJ1VcwxBpBiL+/Ds9/IpRhQQHxfo57x7jG0nne
I/HdFde+Qz2G8kXAKAIhVyk6fkcacwBYTATUX1xGwMT2JKVtRh+kZD1jx2wCj14m
mTPG1e6aZ2maVqDhezHc9F8/dchYj3cw81X0kU6xamqkRQYlxH48NkIOqcn0f6aK
AHF4eWfCr8ot+4rXADlA1JcJcUS1R5SZEctUR2+3pyy+9J0Enjy1y33KUHw30pU
9dpIneyF8XBN/p76ex3M6etv7wpnV/DpjgY8ptxgx6f3yRhRQD5SRc14d1iecaE24j
u7OgotcPkbELH9p6e8galTycG91pAbMQ1150wr0RQ1ADABAc1BADc213quc1mXrXQ
fX4nAc6m4YK4EJqX60o0wF01u4ubnx5V352zHSFWl5+slAIW1bNozScBV0u8G
464WwrA9bv3/c1VqZZ6/1U2TBhHU+Ogh24qhawF5QQUkxEU11To3YsPofUa1gjXKG
F0fJlcVLcnL3K9RlaUxXbEtPWrYuJ39HMFCpAAxV2PPrx5P9PyW3lWhBPc08z5
tFj45b/hnn+j31AVVvgWcZq0Lka57hc4Q7yW/2RoRYqZmdtk7I090LmtkWd8Qvb
qmraoeh2TwGwnN0bQo8X5/l7VktIq778fw96jGsykBr0+DROzj9r110Gg1O6DwL1
daJ5FAEcEyA0q4/fxtPdsN6iaLZ2/eeewtVxjw/woEoFBECb6/y4Roauux9nB
16FCvli79QwpOUT7cmZvY0Mb5GwDE0b1Eo611vm/Qe1M5+sASpS/5rCnHXVe
92CnGh73F4UXEkMaysk8lPj/Tdr5s0XaL+maEuXMY5t1xqDCyEcGyEA1h4X
Cj1fT7aJSCKh72uUtLZDNpIBE6GSK91Q+oHoLF4sKLZ6fu6n8m32sh+Jm0KHtIE
/gkHdqz6jblUWk5UqYEq9Rq8S6Av3Gxqt10Sp003j7BE8RC20Pvf0NNT2mdxf2/
YZx5cvkUxVb9qey7Ye0xWfAchkDmgpYzyIATeEcYnBALD0TpgDrRy1v1M1dMlgH
FF04etk/TBY1XYK1tj8J1qhtibeFzp4q+K7uyUhzj5a4XQOy5lfYHjP/ReTc3ED
r+o2S93ymuyEkgmip22jy jrptmRbWn43t474JhQg6Q0dBd+GdcYyNy9M1n59NPn517E
fuEMd14q3d3h0E177vQAQK8qQRDUk3mXc9oKRNBYderGlJL1QqLBQx4Y18T
ZuFizGWL8+w+PCLMkpxDrVpWqccGpiuiRiElEbPapo0GzegpaY/LJscd/j5zuc8
W3J0tjlnjkaRaf0SF5BPv5tM6YHOZ1F5sVeoiy/a8s1H3NuiqKhn+/TsUHY5JDRh
aek5QK8gOJH4evwR+MYNuwakd61NCBdbH6fEv23WRAAT8BYYz3w9YFnV4U4h/1b
moWYg1K2UprshA8scMUC790FoybQeParQ35x7J19bmTKkCq63fyqqYX3s5XK1
```

---

### Key

```
-----END RSA PRIVATE KEY-----
```

---
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>
</tbody>
</table>

spec.fep.patroni.tls.certificateName: `<secret-name>`

Name of Kubernetes secret that contains the certificate in tls.crt and
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>private key in tls.key for Patroni REST API.</td>
<td></td>
<td>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>
<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.pgrepluserTls.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;pgrepluser&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.pgrepluserTls.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;pgrepluser&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><strong>private key in tls.key for “repluser” user. For MTLS Postgres communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgrepluserTls.caName</code></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><code>spec.fepChildCrVal.sysUsers.pgrepluserTls.sslMode</code></td>
<td><code>verify-full</code></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><code>spec.fepChildCrVal.sysUsers.pgRewindUserTls.certificateName</code></td>
<td><code>&lt;secret-name&gt;</code></td>
<td>Name of Kubernetes secret that contains the certificate in tls.crt and private key in tls.key for “rewinduser” user. For MTLS Postgres communication, this key must be defined. The private key cannot be password protected. When using cert-manager, the secret also contains the CA bundle in ca.crt.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.sysUsers.pgRewindUserTls.caName</code></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><code>spec.fepChildCrVal.sysUsers.pgRewindUserTls.sslMode</code></td>
<td><code>verify-full</code></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 all 0.0.0.0/0 cert hostssl replication all 0.0.0.0/0 cert |

The above setting will force FEP server to perform certification authentication. At the same time verify the authenticity of client certificate.

| spec.fep.customPgHba | hostssl all all 0.0.0.0/0 md5 clientcert=verify-full hostssl replication repluser 0.0.0.0/0 md5 clientcert=verify-full |

The above setting will force 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:
       forceSSL: 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.
   
   caName = enter the name of configmap created for the CA
   
   certificateName = secret created by the end user that contains server certificate

   ```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
   archive_mode = on
   archive_command = 'pgbackrest --stanza=backupstanza
   --config/database/userdata/pgbackrest.conf archive-push %o'
   wal_level = logical
   max_wal_senders = 12
   wal_keep_size = 401
   ```
d. Add entry under spec.fepChildCrVal.customPgHba as shown below.

This requires the client to present a certificate and only certificate authentication is allowed.

Replace "SubClusterName" and "SubNamespace" with the appropriate values as per the Subscriber FEPCluster.

```
RULE

customPgHba: |
# define pg_hba custom rules here to be merged with default rules.
# TYPE DATABASE USER ADDRESS METHOD
hosts all all <SubClusterName>-primary-svc.<SubNamespace>.svc.cluster.local

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.

```yaml
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.

```
sh-4.4$ psql -h /tmp -p 27008 -U postgres
Password for user postgres:
psql (13.1)
Type "help" for help.

postgres=#
```

5. Next, on the publisher side, connect to the database that contains the tables you want to replicate and create a role e.g., logicalrepluser and give the required permissions to this role.

Consider the below image as example only, the privileges to grant may differ as per the requirements.

```
-- Create and Grant Permissions

db1=# CREATE ROLE logicalrepluser WITH REPLICATION LOGIN PASSWORD 'my_password';
CREATE ROLE

db1=# GRANT ALL PRIVILEGES ON DATABASE db1 TO logicalrepluser;
GRANT

db1=# GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO logicalrepluser;
GRANT
```

- 51 -
6. At the Publisher side, create a publication and alter the publication to add the tables that need to be replicated.

```
db1=# create publication my_publication;
CREATE PUBLICATION
```

```
db1=# alter publication my_publication add table my_table;
ALTER PUBLICATION
```

```
db1=#
```

7. At the subscriber side, the custom certificates added in the above step 3.a will be mounted at the path /tmp/custom_certs/ as shown:

```
sh-4.4$ ls -lrt /tmp/custom_certs
total 0
-rwxr-xr-x 1 1001190000 root 103 Aug 10 10:08 logicalrepluser
sh-4.4$ ls -lrt /tmp/custom_certs/logicalrepluser
```

8. The structure of the table to be replicated should be present in the subscriber cluster since logical replication only replicates the data and not the table structure.

Create a subscription as shown below:

```
db1=# CREATE SUBSCRIPTION my_subscription CONNECTION 'host=fepcluster-publisher-primary-svc.ns-a.svc.cluster.local port=27500 sslcert=/tmp/custom_certs/logicalrepluser/tls.crt sslkey=/tmp/custom_certs/logicalrepluser/tls.key sslrootcert=/tmp/custom_certs/logicalrepluser/ca.crt sslmode=verify-full password=my_password user=logicalrepluser dbname=db1' PUBLICATION my_publication WITH (slot_name=myslot1, create_slot=false);
CREATE SUBSCRIPTION
```

The command in the above example is:

```
CREATE SUBSCRIPTION my_subscription CONNECTION 'host=fepcluster-publisher-primary-svc.ns-a.svc.cluster.local port=27500 sslcert=/tmp/custom_certs/logicalrepluser/tls.crt sslkey=/tmp/custom_certs/logicalrepluser/tls.key sslrootcert=/tmp/custom_certs/logicalrepluser/ca.crt sslmode=verify-full password=my_password user=logicalrepluser dbname=db1' PUBLICATION my_publication WITH (slot_name=myslot1, create_slot=false);
```

- Host = primary service of the publisher FEP Cluster
- sslcert, sslkey, sslrootcert = path to certificates mounted on the Subscriber FEP Cluster
- user= Role created on the Publisher side
- password= password for the role
- dbname= database which contains the tables to be replicated

Where

- Host = primary service of the publisher FEP Cluster
- sslcert, sslkey, sslrootcert = path to certificates mounted on the Subscriber FEP Cluster
- user= Role created on the Publisher side and used to establish logical replication connection from Subscriber to Publisher
- dbname= database which contains the tables to be replicated

- 52 -
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>COUNT</td>
<td>.spec.fep.instances</td>
<td>Number of FEP nodes in the cluster</td>
</tr>
</tbody>
</table>

Example)

```
# kubectl get fepclusters -A

NAMESPACE       NAME            AGE
namespace1      ns1fep1         21h
namespace2      ns2fep2         22h
```

Update FEPCluster

Equivalent Kubernetes command: kubectl apply -f <new_spec>

Operations that can be performed here.

<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, FEPConfig, FEPCert & FEPUser) & resources associated with it.

Note

Deleting a FEPCluster will delete all PV associated with the cluster, including backup and archived WAL volumes (unless using 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 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 pgpool2</td>
</tr>
</tbody>
</table>

Example)

```
# kubectl get feppgpool2 -A
```

<table>
<thead>
<tr>
<th>NAMESPACE</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>namespace1</td>
<td>fep1-pgpool2</td>
</tr>
<tr>
<td>namespace2</td>
<td>fep2-pgpool2</td>
</tr>
</tbody>
</table>
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.customparams</td>
<td>Change pgpool2 parameters</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.
$ oc exec pod/fepserver-XXXXX -c FEPbackup - pgbackrest info
stanza: fepbackup
  status: ok
cipher: none
db (current)
  wal archive min/max (12-1): 000000010000000000000001/000000010000000000000005
  full backup: 20201125-025043F
    wal start/stop: 000000010000000000000003 / 000000010000000000000003
    database size: 31.7MB, backup size: 31.7MB
    repository size: 3.9MB, repository backup size: 3.9MB
  incr backup: 20201125-025043F_20201125-025600I
    timestamp start/stop: 2020-11-25 02:56:00 / 2020-11-25 02:56:02
    wal start/stop: 000000010000000000000005 / 000000010000000000000005
    database size: 31.7MB, backup size: 24.3KB
    repository size: 3.9MB, repository backup size: 619B
    backup reference list: 20201125-025043F

Update FEPBackup

Equivalent Kubernetes command: kubectl apply -f <new_spec>

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

<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>

Note
Changes made during the backup are reflected from the next backup.
Changes to the backup schedule do not affect the application.

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>Metrics name</td>
<td>Details</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>- Receive Bandwidth</td>
<td></td>
</tr>
<tr>
<td>- Transmit Bandwidth</td>
<td></td>
</tr>
<tr>
<td>- Rate of Received Packets</td>
<td></td>
</tr>
<tr>
<td>- Rate of Transmitted Packets</td>
<td></td>
</tr>
<tr>
<td>- Rate of Received Packets Dropped</td>
<td></td>
</tr>
<tr>
<td>- Rate of Transmitted Packets Dropped</td>
<td></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 metrices from all FEPCluster nodes

**Note**
- Alert Manager integration to back-office to send mail / message / raising ticket is done by user based on their environment
- Grafana installation and integration is done by user. Use the Grafana Operator provided by OperatorHub.
- Grafana dashboard is created by user based on their requirements and design.

### 5.5.2.2 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_bewriter_*</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 PostgreSQL)</td>
</tr>
<tr>
<td></td>
<td>- scrapes_total (Total number of times PostgreSQL was scraped for metrics)</td>
</tr>
<tr>
<td></td>
<td>last_scrape_error (Whether the last scrape of metrics from PostgreSQL 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>
</tbody>
</table>
### Metrics name

<table>
<thead>
<tr>
<th>Metrics name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<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.

![Example Image](image.png)

### 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>PVCLOWDiskSpace</td>
<td>Warning</td>
<td>5 mins</td>
<td>A FEP PVC (volume) has less that 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>

**Note** 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 pcpool2 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>
<tr>
<td>Column</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>create_time</td>
<td>timestamp with timezo</td>
<td>The creation time and date of the connection</td>
</tr>
<tr>
<td>majorversion</td>
<td>integer</td>
<td>The protocol version numbers used in this connection</td>
</tr>
<tr>
<td>minorversion</td>
<td>integer</td>
<td>The protocol version numbers used in this connection</td>
</tr>
<tr>
<td>pool_counter</td>
<td>integer</td>
<td>Counts the number of times this pool of connections (process) has been used by clients</td>
</tr>
<tr>
<td>pool_connected</td>
<td>boolean</td>
<td>True (1) if a frontend is currently using this backend</td>
</tr>
</tbody>
</table>

### 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.

  Please note the following child CR event are raised as part of Create FEPPgPool2

- `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

1. 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,
2. **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.

![Event Screenshot](image)

5.7 **Scaling Replicas**

5.7.1 **Auto Scale Out**

   Auto-scale-out occurs when the average CPU utilization of the DB container exceeds the threshold.

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

   Specify `spec.fepChildCrVal.autoscale.scaleout` in `FEPClusterCR` when you want to perform Auto scale out.

   ```
   $ 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.
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

Provides major version upgrade procedures for operators and containers.

6.4.1 Install a New Version of the Operator

See “Chapter 3 Operator Installation” to install a new version of the FEP operator.

The Namespace you install can be different from the older version of the operator you are upgrading.

6.4.2 Deploy a New Version of a Container

See “Chapter 4 Deployment Container” to deploy the FEP container.

If necessary, deploy the Pgpool2 container.

6.4.3 Migrating Data Between Containers

Database data migration requires business application outages. Use the pre-verification to understand the migration time in this section and estimate the downtime.

1. Stop the operation to the old DB container in operation.
2. Extract database data from the old database container. Run on the database client where the client corresponding to the old database engine is installed. Get the database data, schema from the old database container by executing the following command.

Example)

```
$CLIENT_DIR=[Directory path where the database engine client was installed]
$export PATH=${CLIENT_DIR}/bin:$PATH
$export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
$IP=[Destination IP address of the old DB container]
$PORT=[ Port number of the old DB container]
$pg_dumpall -h ${IP} -p ${PORT} -U postgres > db.dump
```

Note

Check the database size in advance by connecting to database and executing SQL below.

```
$ SELECT pg_size_pretty(sum(pg_database_size(datname))) AS dbsize FROM pg_database;
```

Since the pg_dumpall command used above outputs the data of the database as an SQL command, the file actually created is:

For example, 2147483647 of type integer is 4 bytes for database data,
However, since SQL commands output them as strings, this is 10 bytes. Therefore, run it in a location where there is sufficient disk space.

3. Populate the new DB container with data.
   Run the new database engine client on the installed database client.
   Populate the new DB container deployed in “6.4.2 Deploy a New Version of a Container” with the schema and data extracted from the old DB container using the pg_dump command.

   **Example**
   ```bash
   $CLIENT_DIR=[ Directory path where the database engine client was installed]
   $export PATH=${CLIENT_DIR}/bin:${PATH}
   $export LD_LIBRARY_PATH=${CLIENT_DIR}/lib:${LD_LIBRARY_PATH}
   $IP=[Destination IP address of the new DB container]
   $PORT=[ Port number of the new DB container]
   $psql -h ${IP} -p ${PORT} -U postgres -d postgres -f db.dump
   ```

   **Note**
   When database data is input, a message that the following extension registration failed is output, but ignore this message.
   - `pgx_cpu`
   - `pgx_disk`
   - `pgx_io`
   - `pgx_log`
   - `pgx_memory`
   - `pgx_network`
   - `pgx_network_err`
   - `pgx_paging`
   - `pgx_process`
   - `pg_stat_statements`

4. Change the destination configured for the application to the new DB container or the new Pgpool2 container.

   **Note**
   - For more information about the pg_dumpall command, see the following in the source database engine documentation ("PostgreSQL 13 documentation").
     * "VI. Reference" - "II. PostgreSQL Client Applications" - "pg_dumpall"
   - For more information on the psql command, see the destination database in the documentation ("PostgreSQL 13 documentation").
     * "VI. Reference" - "II. PostgreSQL Client Applications" - "psql"

### 6.4.4 Removing Old Containers

If necessary, remove the old DB container.

"Operators" > "Installed Operators" > "FUJITSU Enterprise Postgres <Old version> Operator" > "FEPCluster" > "FEPCluster name to delete" > Choose "Delete FEPCluster" from Actions

### 6.4.5 Uninstalling Old Operators

Uninstall the old FEP operator if necessary.
"Operators" > "Installed Operators" > "FUJITSU Enterprise Postgres <Old version> Operator" > Choose "Uninstall Operator" from Actions
Chapter 7 Abnormality

This chapter describes the actions to take when an error occurs in the database or an application, while FEP is operating.

Depending on the type of error, recover from the backed-up material, reserve capacity, check the operator log, and check the FEP log.

7.1 Handling of Data Abnormalities

Recover the database cluster from the backup immediately prior to failure in any of the following cases:

- A hardware failure occurs on the data storage disk or the backup data storage disk.
- If the data on the disk is logically corrupted and the database does not work correctly
- Data corruption caused by user error

Refer to "6.3 Perform PITR and the Latest Backup Restore from Operator" for backup 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 Collecting Fault Investigation Information.

7.5 Collection of Failure Investigation Information

If the cause of the trouble that occurred during the construction or operation of the environment is not identified, information for the initial investigation is collected.

I will explain how to collect information for the initial investigation.

- Product log
- Operator log

Product log

FEP log

Get into the container and collect the log.

The log location is specified by log_directory in the custom resource FEP Clusterspec.startupValues.customPgParam parameter. The default is/database/log.
Pgpool-II log

Get into the container and collect the log.

The log location is /var/log/pgpool/pool.log.

Operator log

Check the operator log as follows.

Verification Example

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

How to check the log

```
$ oc logs pod fep-ansible-operator-7dc5fd9bf7-4 smzk -c manager
```

The log will be output to the console. Please check the file output by redirection.
Appendix A  Quantitative Values and Limitations

A.1 Quantitative Values

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

A.2 Limitations

Note

If you log in to a container and edit the configuration file directly, restarting the container may undo your changes.

If you want to change the settings, modify the custom resource files as described in "5.1 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 doesn’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".
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)

```yaml
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)

```yaml
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 -
```
FUJITSU Enterprise Postgres 13 for Kubernetes

Reference
Preface

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

Intended readers
This document is aimed at people who manage and operate.

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

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

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</td>
</tr>
<tr>
<td>FUJITSU Software Enterprise Postgres</td>
<td>FUJITSU Enterprise Postgres</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**

1.1 FEPCluster Parameter ................................................................................................................................. 1

1.2 Custom Resource Parameters ....................................................................................................................... 13

1.2.1 FEPCluster Custom Resource Parameters ............................................................................................ 13

1.2.2 FEP Cluster Configuration .......................................................................................................................... 19

1.2.3 FEPConfig Child Custom Resource Parameters .................................................................................... 19

1.2.4 FEPUser Child Custom Resource Parameters ......................................................................................... 21

1.2.5 FEPVolume Child Custom Resource Parameters ..................................................................................... 23

1.2.5.1 Create Volumes .................................................................................................................................. 23

1.2.5.2 Delete Volumes ................................................................................................................................ 24

1.2.6 FEPCert Child Custom Resource Parameters .......................................................................................... 24

1.2.6.1 Create/ Update Certificates .................................................................................................................. 24

1.2.6.2 Delete Certificates .............................................................................................................................. 25

1.2.7 FEPBackup Child Custom Resource Parameters ..................................................................................... 26

1.2.8 FEPRestore Custom Resource Parameters .............................................................................................. 27

1.2.9 FEPpgpool2 Custom Resource Parameters ............................................................................................ 29

1.2.10 FEPAction Custom Resource Parameters .............................................................................................. 33

1.2.10.1 FEPAction Specific Operation Details ............................................................................................... 34

1.2.11 FEPExporter Custom Resource ............................................................................................................. 36

1.2.12 FEPAutoscale Custom Resource ............................................................................................................ 38

**Appendix A Default Metrics Queries** .............................................................................................................. 39

**Appendix B Default Alert Rules** ..................................................................................................................... 48
# 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 ‘pod_restart’ and arguments ‘ALL’</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 added to annotations of FEP statefulSet and FEP Pods. If left at default, no annotation is added to Pods and statefulSets</td>
</tr>
<tr>
<td>spec.fep.image.image</td>
<td>&lt;omitted&gt;</td>
<td>FEP server container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-13-server@ubi8-13-0.0. It is optional. Image line is omitted by default. In such a case, it will pick up URL of image from operator container environment. If you specify the image, Operator will take that image to deploy fep container</td>
</tr>
<tr>
<td>spec.fep.image.pullPolicy</td>
<td>IfNotPresent</td>
<td></td>
</tr>
<tr>
<td>spec.fep.mcSpec.limits</td>
<td>cpu: 500m memory: 700Mi</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.fep.mcSpec.requests</td>
<td>cpu: 200m</td>
<td></td>
</tr>
<tr>
<td></td>
<td>memory: 512Mi</td>
<td></td>
</tr>
<tr>
<td>spec.fep.sysExtraLogging</td>
<td>false</td>
<td>To turn extra debugging on, set value to true</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It can be turned on/off at any time</td>
</tr>
<tr>
<td>spec.fep.instances</td>
<td>1</td>
<td>Number of nodes in the cluster, including both Master and Replicas.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Example CR, it is kept at 1 for certification.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>However, user can change it to 3 for 1 master and 2 replicas.</td>
</tr>
<tr>
<td>spec.fep.servicePort</td>
<td>27500</td>
<td>TCP port for FEP master service</td>
</tr>
<tr>
<td>spec.fep.syncMode</td>
<td>off</td>
<td>Replication Mode:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>off - async replication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on - sync replication</td>
</tr>
<tr>
<td>spec.fep.forceSsl</td>
<td>true</td>
<td>Controls that the communication to the server should only be via SSL.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes are reflected in pg_hba.conf</td>
</tr>
<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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fep.monitoring.fepExporter.tls.ca Name</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.ca Name</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>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...</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.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 verify client. The CA is stored in the key ca.crt. This field is optional.</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.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/&lt;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 that contains additional CA for Postgres to verify client. The CA is stored in the key ca.crt. This field is optional.</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.backup</code></td>
<td></td>
<td>that contains CA certificate to verify server. The CA is stored in the key ca.crt.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.image.image</code></td>
<td><code>&lt;omitted&gt;</code></td>
<td>FEP backup container image to be used quay.io/fujitsu/fujitsu-enterprise-postgres-13-backup@ubi8-13-0.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>It is optional. Image line is omitted by default. In such a case, it will pick up URL of image from operator container environment. If you specify the image, Operator will take that image to deploy backup container</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.image.pullPolicy</code></td>
<td><code>IfNotPresent</code></td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.mcSpec.limits</code></td>
<td><code>cpu: 0.2</code>&lt;br&gt;<code>memory: &quot;300Mi&quot;</code></td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.mcSpec.requests</code></td>
<td><code>cpu: 0.1</code>&lt;br&gt;<code>memory: &quot;200Mi&quot;</code></td>
<td></td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.pgbackrestParams</code></td>
<td><code>[global]</code>&lt;br&gt;<code>repo1-retention-full=7</code>&lt;br&gt;<code>repo1-retention-full-type=time</code>&lt;br&gt;<code>log-path=/database/log/backup</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.schedule.num</code></td>
<td><code>0</code></td>
<td>Number of schedules to set. The maximum number of backup schedules is 5.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.scheduleN.schedule</code></td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.backup.scheduleN.type</code></td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.customCertificates</code></td>
<td></td>
<td>List of custom certificates.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.customCertificates.userName</code></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td><code>spec.fepChildCrVal.customCertificates.certificateName</code></td>
<td></td>
<td>The secret name which contains the certificate files</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.customCertificate</td>
<td></td>
<td>If not defined, a folder to store the certificate files will still be created by the name defined in spec.fepChildCrVal.customCertificate</td>
</tr>
<tr>
<td>s.caName</td>
<td></td>
<td>s.userName</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customCertificate</td>
<td></td>
<td>The configmap of CA certificate</td>
</tr>
<tr>
<td>s.privateKeyPassword</td>
<td></td>
<td>The secret which contains the certificate key password</td>
</tr>
<tr>
<td>spec.fepChildCrVal.customPgAudit</td>
<td>[output]</td>
<td>PgAudit file content</td>
</tr>
<tr>
<td>logger = 'auditlog'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_directory = '/database/log/audit'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_truncate_on_rotation = on</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log_filename = 'pgaudit-%a.log'</td>
<td></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>[rule]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.customPgHba</td>
<td># define pg_hba custom rules here to be merged with default rules.</td>
<td>Entries to be inserted into pg_hba.conf</td>
</tr>
<tr>
<td># TYPE DATABASE USER ADDRESS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>METHOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.customPgParams</td>
<td># define custom postgresql.conf parameters below to override defaults.</td>
<td>Postgres configuration in postgresql.conf</td>
</tr>
<tr>
<td># Current values are as per default FEP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>deployment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shared_preload_libraries=pgx_datamaking,pg</td>
<td></td>
<td></td>
</tr>
<tr>
<td>prewarm,pg_stat_statements'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>session_preload_libraries='pg_prewarm'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_prepared_transactions = 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_worker_processes = 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_connections = 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>work_mem = 1MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maintenance_work_mem = 12MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>shared_buffers = 128MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>effective_cache_size = 384MB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>checkpoint_completion_target = 0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td># tcp parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_idle = 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_interval = 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tcp_keepalives_count = 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td># logging parameters in default fep</td>
<td></td>
<td></td>
</tr>
<tr>
<td>installation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# if log volume is not defined, log_directory should be changed to '/database/userdata/data/log'
log_directory = '/database/log'
log_filename = 'logfile-%a.log'
log_file_mode = 0600
log_truncate_on_rotation = on
log_rotation_age = 1d
log_rotation_size = 0
log_checkpoints = on
log_line_prefix = '%e %t [%p]: [%l-1] user=%u,db=%d,app=%a,client=%h'
log_lock_waits = on
log_autovacuum_min_duration = 60s
logging_collector = on
pgaudit.config_file='/opt/app-root/src/pgaudit-cfg/pgaudit.conf'
log_replication_commands = on
log_min_messages = WARNING
log_destination = stderr

# wal_archive parameters in default fep installation
archive_mode = on
archive_command = 'pgbackrest --stanza=backupstanza --config=/database/userdata/pgbackrest.conf
archive-push %p'
wal_level = replica
max_wal_senders = 12
wal_keep_segments = 64
track_activities = on
track_counts = on

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>spec.fepChildCrVal.storage.dataVol</td>
<td>Mandatory volume</td>
<td></td>
</tr>
<tr>
<td>spec.fepChildCrVal.storage.dataVol.size</td>
<td>2Gi</td>
<td>Size of data volume.</td>
</tr>
<tr>
<td></td>
<td>(**)</td>
<td>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></td>
<td>(*)</td>
<td></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]</td>
</tr>
<tr>
<td></td>
<td>(*)</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.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>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>spec.fepChildCrVal.storage.archivewalVol.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.archivewalVol.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.archivewalVol.accessModes</td>
<td>&lt;omitted&gt;</td>
<td>accessModes for Archived WAL 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.logVol</td>
<td></td>
<td>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.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></td>
<td>Mandatory if backup section is defined. Optional otherwise</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: 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.backupVol.accessModes</td>
<td>&lt;omitted&gt;</td>
<td>accessModes for backup volume: Specified as an array of accessModes e.g. [ReadWriteMany] If omitted, it will be treated as [ReadWriteOnce] When the number of instance is more than 1 and backup is not done on S3, both archivewalVol and backupVol must be hosted on Shared storage such</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</td>
<td>admin-password</td>
<td>Password for user &quot;postgres&quot;</td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td>as NFS with accessMode set to [ReadWriteMany]</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.pgreplususer</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 user</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 user</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.tls.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 certificate itself is stored in the key tls.crt. This field is optional.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.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.tls.sslMode</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>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgreplUserTls.certificateName</td>
<td></td>
<td>This points to Kubernetes TLS secret that contains the certificate of Postgres user “repluser”. Patroni will use this for certificate authentication. The certificate itself is stored in the key tls.crt. This field is optional.</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></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>spec.fepChildCrVal.sysUsers.pgRewindUserTls.certificateName</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.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.pgRewindUserTls.sslMode</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>spec.fepChildCrVal.sysUsers.pgMetricsUserTls.certificateName</td>
<td></td>
<td>Optional This points to Kubernetes TLS secret that contains the certificate of Postgres user defined by pgMetricsUser. FEPExporter will use this for certificate authentication. The certificate itself is stored in the key tls.crt.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.sysUsers.pgMetricsUserTls.caName</td>
<td></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>field:spec.fepChildCrVal.sysUsers.pgMetricsUserTls.sslMode</td>
<td>prefer</td>
<td>Optional 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>spec.fepChildCrVal.systemCertificates.key</td>
<td></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.systemCertificates.crt</td>
<td></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.systemCertificates.cacrt</td>
<td></td>
<td>Use spec.fep.postgres.tls specification instead.</td>
</tr>
<tr>
<td>spec.fepChildCrVal.autoscale.scaleout.policy</td>
<td>off</td>
<td>[cpu_utilization/off]</td>
</tr>
<tr>
<td>spec.fepChildCrVal.autoscale.scaleout.threshold</td>
<td>40</td>
<td>Threshold (Average of all replicas) Must be an integer. The unit is %</td>
</tr>
<tr>
<td>spec.fepChildCrVal.autoscale.limits.maxReplicas</td>
<td>2</td>
<td>Maximum number of replicas (0 to 15) (Value out of range) Do not perform auto scale-out</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>
</tbody>
</table>

Definition:

```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: FEPClusters.fep.fujitsu.io
spec:
group: fep.fujitsu.io
names:
  kind: FEPCluster
  listKind: FEPClusterList
  plural: fepclusters
  singular: fepcluster
shortNames:
- fac
scope: Namespaced
conversion:
  strategy: None
versions:
- name: v2
  served: true
  storage: true
schema:
  openAPIV3Schema:
    description: FEPCluster is the Schema for the fepclusters API
    properties:
      apiVersion:
        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: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources'
        type: string
      kind:
        description: 'Kind is a string value representing the REST resource this
```
<table>
<thead>
<tr>
<th>Category</th>
<th>Details</th>
</tr>
</thead>
</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
    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:
```

- 14 -
type: logical
database: postgres
plugin: pgoutput
demo_subscription3:
  type: logical
database: postgres
plugin: pgoutput

defaultCrVal:
customCertificates:
- userName: my_cert_folder1
certificateName: my_cert1_secret
cName: my_ca_configmap
privateKeyPassword: my_cert1_key_secret
- userName: my_cert_folder2
certificateName: my_cert2_secret
cName: 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.
[global]
repol-retention-full = 30
repol-retention-full-type = time
postScript: ""
schedule:
  num: 2
  schedule1:
    schedule: "15 0 * * 0"
    type: "full"
  schedule2:
    schedule: "15 0 * * 1-6"
    type: "incr"
  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
  pgpassword: mydbpassword
  pguser: mydbuser
  pgrepluser: repluser
  pgreplpassword: repluserpwd
tеперсепассы: tde-treepassphrase

systemCertificates:
  key: |
  -----BEGIN RSA PRIVATE KEY-----
  MIIEpAIBAAKCAQQAvhL4BD4LO1YU3nu+jgLd0L4Ee0qWmqlxhspYPRb43paWGF1p
gX1CNP3V01j18ns44LG6W6n7TqV73Mf214NHpuVtjKWT66wtf7dQj7bbKeWQCD5FBK
Qp0keP5HAv45B4x15FppFvmMXWXXUB9niic18X7M1GpQP5u1r7kj8S5S5KgSKQKdpXw
wyGueufbevnaYCI8kapBCAsRIMjWuWngri1n4b8ZIvH0mceuHLrXH8W7mQ7vQbH
9laEwq/K1T0WpV8dC2111tH6gBEC6d6n4q0/vlx0J2McXK63Q+z7Y3z5x5qSNN+
/Kgaesht/K16AcEioJ52P04vneLwErxK6KjMRiWdJQAACAlIF2vH9FRlrr1q4CGyR
6xw11f7j677677r0A0YFRapQ5o0o2sKsvfrBoQ12ly3gDfD0Mq8z4m4bnaq+9H3P31S
72eUSLpJmriZGIxxcDYPFPvuSN6JXMI09M+snxSxzzCFqTmNPbB19LcLSjH0snqQ
G1YHRPe2RqB0rCnD1exsKoj91VNRnIto2ZcArj1ZyjNSS4vPajEjAYSW/XxruRzi
A5smz2XeK55F+Jpc2+H4Q+Rd+OA0LKrRA0yG1CMG3X51YgwBF7jRkMdHJOJIsNim
em+kJx7cSh5JKF+2uxJ+1LW9d+7CtExDyMXKV1Ta7F7a9zgJ71MvotG3tnVnLzATP
KTT0DADcYEA66hsh/V99oB+1WMM4hKAMtqnpwWOhqFQkJMAYIODZXfzWKMKzEKKK8p
k1bZM24xuMs5b/hvKlcvQcrx5Y21LLU0Dcsed280xMG6WnqjAh07kKXhYo1kK1
gVYV3v7Te0Jn3S3191IedC0qVxibPkhD861V79Hdd9W1anp/8RveEcyEAoz9BC
A+ambe3+1xLaQOJ+54aQa/Tb+bBu6Rc1H8OPC1CZ55Ep1p314+ma1tYwM8Renrt
4CmSvphWkmO0nnHCyHerCySjWyV5Mc33RARX5xN107TJ7Bgo2hdaTieEm/KX6
rakgerh7csgXO1Xcbz+NN0ApB2sz7BphtzF3Nm0cCyEYJneotZ71rIO fmKtDIUN
OgQyU09jH5a0dmh0JnTvay8/XKdrgzbhym4GpJa4qt12zn11t0yJrBaW2d6dx6w1IM
q+vEnexFtr0Iw66rlxJm5A/JH37Q8L1DNCbPb7sVjS1U141BD2MRWq498tW22
N7OAIgFZyQuCMtuc8UW0FwvrEcgy07xefOgdZteai1CnUZ7fhh0FjykAK1n0mdYJN32
37Lw9J55xb0bd2U/4Kf0M7n0GavWNRH11eyRQAwyrxd/5FX5LflqP1N1Bm2UH
Pf95/qN05eC6FRfUW0AvtvVosct.XFeBEIWH9ydB7gLQc6m39YKkzUQJURL7pu1
8AcwQBQfQCl149QXqQmknak1hQsY5e+v049G8jRpa7HeE0qge5oJG5shA1mqQJsvu
t+dWgoto944S5jBh9H6hNQLF4x3ZP6vFz8w2FwRCPF7k7bzwqQYv+jp1M4dZhPH6
nSTGz3cdWb7VCNH3APGClZDazEjb0MQLW8q/euhdcLnyuB9Jbbw==

--- END RSA PRIVATE KEY -------
crt: |

---BEGIN CERTIFICATE-----
MIIEhJCCAgawIwBAIGjIWoRzLgsw88HmNAQCSqGSiB3DQEEBcWU/MBYx/FASDBgNV
BAMCM+CwI/gj5MB4XDT1wMTeWQDE0MzUyN1oXDTQ4MDMnN3E0MsUyL1ow
aDELMak4Al1nUEBCOMเตขEFSDBgNBQAgMCMA0IhaYfYXN0dHJmQ0cowDYVDQQfHAQ4Q
N5MR4QgDVQVQKDAd7dpddhPqNMIQxQwGyDVQVQDLANDT0UxFASD8BGmAMcEzW
Ljz6MB4wIdj5MIJiBhBqjkhiG9W0BAEEFAAOCAPQQhIMBqCqCAqAhLvd/01
Lmm/R3nu+jgLd0L4Ee0qWmqlxhspYPRb43paWGF1p缘X1CNAP3Z1Etms4NLVCG6m77q
V73Mf214NHpuVtjKWT66wtf7dQj7bbKeWQCD5FBKQpeK5HAv/5uQ4Bx15FppJv
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
- pgreplpassword: repluserpwd
- tdepassphrase: tde-passphrase
- pgRewindPassword: rewind_password (Optional - if defined)
- pgMetricsPassword: metrics_password (Optional - if defined)
- certificate.key
- certificate.crt
- certificate.cacrt

Values of child CRs at the time of initial deployment of cluster, are stored in FEPCluster under fepChildCrVals, e.g. for Server certificates, Configuration of FEP. User details.

All fields for FEPCluster CR and its child CRs should be managed through FEPCluster CR only. Operator will reflect the changes to respective child CR to be processed. The fields that not allowed to change will not be reflected from parent to child CR and hence will not have any affect.
1.2.2 FEP Cluster Configuration

Configuration of all aspects of FEP Cluster is done through FEPCluster CR only.

All fields for FEPCluster CR and its child CRs should be managed through FEPCluster CR only. Operator will reflect the changes to respective child CR to be processed. The fields that not allowed to change will not be reflected from parent to child CR and hence will not have any affect. Refer to "1.1 FEPCluster Parameter" for details.

All child CRs are marked as internal objects in RedHat OCP and will not appear on console. However, it can be checked on command line using oc or kubectl commands.

Following table shows Child CRs of FEPCluster CR and respective sections in parent CR related to given child CR.

<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 FEPCconfig 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 of FEPCluster CR</td>
<td>Audit rules can be updated in this section. Requires restart. Note: initial values inherited once only at start. Changes to FEPCConfig directly</td>
</tr>
<tr>
<td>spec.customPgHba</td>
<td>All line specified in spec.fepChildCrVal.customPgHba of FEPCluster CR</td>
<td>pg_hba rules can be added in this section Note: Inherited once at start. Changes to FEPCConfig 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 FEPCConfig directly</td>
</tr>
<tr>
<td>spec.replicationSlots</td>
<td>Optional: Details of replication slots if defined in FEPCluster</td>
<td></td>
</tr>
</tbody>
</table>

Example of FEPCconfig CR created

```yaml
apiVersion: fep.fujitsu.io/v1
kind: FEPCConfig
metadata:
  name: new-fep-19ncfg
  namespace: cfg-expt
```
spec:
  sysExtraLogging: false

customPgAudit:
  # define pg audit custom params here to override defaults.
  # if log volume is not defined, log_directory should be
  # changed to '/database/userdata/data/log'

  [output]
  logger = 'auditlog'
  log_directory = '/database/log/audit'
  log_truncate_on_rotation = on
  log_filename = 'pgaudit-%a.log'
  log_rotation_age = 1d
  log_rotation_size = 0

[rule]

customPgHba:
  # define pg_hba custom rules here to be merged with default rules.
  # TYPE DATABASE USER ADDRESS METHOD

customPgParams:
  # define custom postgresql.conf parameters below to override defaults.
  # Current values are as per default FEP deployment

  shared_preload_libraries='pgx_datamasking,vci,pgaudit,pg_prewarm,pg_stat_statements'
  session_preload_libraries='vci,pg_prewarm'
  max_prepared_transactions = 100
  max_worker_processes = 20
  max_connections = 100
  work_mem = 3MB
  maintenance_work_mem = 20MB
  shared_buffers = 128MB
  effective_cache_size = 384MB
  checkpoint_completion_target = 0.8
  pgx_global_metacache = 10MB
  temp_buffers = 10MB

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

  # logging parameters in default fep installation
  # if log volume is not defined, log_directory should be
  # changed to '/database/userdata/data/log'    log_directory = '/database/log'
  log_filename = 'logfile-%a.log'
  log_file_mode = 0600
  log_truncate_on_rotation = on
  log_rotation_age = 1d
  log_rotation_size = 0
  log_checkpoints = on
  log_line_prefix = '%e %t [ %p]: [%d-%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
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</td>
<td>postgres superuser password. Masked once secret is created/changed</td>
</tr>
<tr>
<td></td>
<td>spec.FEPCluster.CR</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</td>
<td>Name of a user database</td>
</tr>
<tr>
<td></td>
<td>spec.FEPCluster.CR</td>
<td>Note: Created once only at start. Cannot be changed</td>
</tr>
<tr>
<td>spec.pgpassword</td>
<td>spec.fepChildCrVal.users.pgpassword</td>
<td>Password for superuser for user database pgdb. Masked once secret is created/changed</td>
</tr>
<tr>
<td></td>
<td>spec.FEPCluster.CR</td>
<td>Note: initial values inherited once only at start. Changes to FEPUser directly</td>
</tr>
<tr>
<td>spec.pguiser</td>
<td>spec.fepChildCrVal.users.pguiser</td>
<td>Name of a user database</td>
</tr>
<tr>
<td></td>
<td>spec.FEPCluster.CR</td>
<td>Note: Created once only at start. Cannot be changed</td>
</tr>
<tr>
<td>spec.pgrepluser</td>
<td>spec.fepChildCrVal.users.pgrepluser</td>
<td>Name of a database user for replication</td>
</tr>
<tr>
<td>spec.pgreplpassword</td>
<td>spec.fepChildCrVal.users.pgreplpassword</td>
<td>Password for pgrepluser</td>
</tr>
<tr>
<td>spec.tdepassphrase</td>
<td>spec.fepChildCrVal.users.tdepassphrase</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.pgAdminTls</td>
<td>spec.fepChildCrVal.sysUsers.pgAdminTls</td>
<td>Optional section See details in FEPCluster CR</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>spec.pgrepluserTls</td>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls</td>
<td>Optional section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgRewindUserTls</td>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls</td>
<td>Optional section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>See details in FEPCluster CR</td>
</tr>
<tr>
<td>spec.pgMetricsUserTls</td>
<td>spec.fepChildCrVal.sysUsers.pgrepluserTls</td>
<td>Optional section</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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
  pgAdminTls:
    certificateName: admin-client-certs-secret
    caName: admin-ssl-rootcert-configmap
    sslMode: prefer
  pgrepluserTls:
    certificateName: repluser-client-certs-secret
    caName: repluser-ca-name-configmap
    sslMode: prefer
  pgRewindUserTls:
    certificateName: rewinduser-client-certs-secret
    caName: rewinduser-ca-name-configmap
    sslMode: prefer
```

**Note**

- Password and Passphrase are masked in output from CR. The original values can still be found in the respective Kubernetes secrets and configmaps.
- TDE is enabled by default with given tdepassphrase and must have a value.
- TDE is enabled by using the key tdepassphrase with the desired passphrase. Do not remove this key once TDE is enabled. Otherwise, the database may go into a crash loop. If the Cluster is running on Async Replication and a failover/switchover 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>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></td>
<td></td>
<td></td>
<td></td>
<td>Additional details in section 3.2</td>
</tr>
<tr>
<td>backupVol</td>
<td>No</td>
<td>size</td>
<td>2Gi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td></td>
</tr>
<tr>
<td>dataVol</td>
<td>Yes</td>
<td>size</td>
<td>2Gi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td></td>
</tr>
<tr>
<td>logVol</td>
<td>No</td>
<td>size</td>
<td>1Gi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td></td>
</tr>
<tr>
<td>tablespaceVol</td>
<td>No</td>
<td>size</td>
<td>512Mi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td>Defaults to ReadWriteOnce if omitted</td>
<td></td>
</tr>
<tr>
<td>walVol</td>
<td>Yes</td>
<td>Size</td>
<td>1200Mi</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>storageClass</td>
<td>Defaults to platform default if omitted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>accessModes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
  - name: log
  sysExtraLogging: false
```

1.2.6 FEPCert Child Custom Resource Parameters

1.2.6.1 Create/ Update Certificates

Certificate secret for the FEP cluster is initially created in accordance with the values set in fepChildCrVal’certs section of the parent FEPCluster CR.

Below is the schema of the FEPCert CR:

<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>Field</td>
<td>Default</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------------------------</td>
<td>------------------------------------------</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.

- `ssl = on`
- `ssl_cert_file = '/fep-certs/fep.crt'`
- `ssl_key_file = '/fep-certs/fep.key'`
- `ssl_ca_file = '/fep-certs/ca.crt'`

It should also be possible to change the certificates by end user, by changing ALL key, crt and cacrt. However, user will need to restart the cluster to let change take effect.

### 1.2.6.2 Delete Certificates

Equivalent Kubernetes command: `kubectl delete FEPCert <cr_name>`

This operation will remove the secret containing the TLS Certificates and keys for the cluster.

Below is an example CR for certificates to be used by FEP server container

```
apiVersion: fep.fujitsu.io/v1
kind: FEPCert
metadata:
  name: new-fep
  namespace: ansible-operator-poc
spec:
  key: |
    -----BEGIN RSA PRIVATE KEY-----
    MIIEowIBAAKCAQEA4AI33yvHZws+jta6qpVw6zjQgF8o6dITiPcrFrVcUUtLFKj1I
    2e4ScTe6I03C/IXuvWipmg1I065s/fQQLO06z1/AuQ7T8YUn/W9m1aHVav4N
    B5JHHWQoJRt3s7nRPGX1ibn0P0rGE2mJjCv9RnExJ3Ieaktg3sdbvlYxtcheyP
    mjdbxf8xv7G0g+/+c/wKrRw0K8U7f7seu0cI/-49J6r5s41gD8sINQCFU1F7
    YvmA7glcSf5B8BNFoUATHEs/fm1W0DRCJWnhTL0ht+6SL/1zWH1LGp2dg6W
    dgmwu5HPm18CDNLNd98AAAj7i+15SRKRCVl1nuQIDAQAAbIAFBPQQY1Kzw/+BAO
    yMUpdct/IMC/r0xWv1bDjg1j1gNVPJHQUvB1B2FA1TQ6qBoj006BaAOQdWN
    Rb0v/+/yYlfNDJF/saJlJAH/0l+2/oEn9R6Fq2gqFVDeh+e1a2r3x7Gsom+p925k30
    l6p0r1Vr1+8J8KsV1V2HwLw/R3J0tPn++mMTzLV/0I+f+y8Q3+S2HuApm49EKXj
    cEmn2J5B7c0ixS5vky0F+s+zbqL1BQKQ2A5Uere1BaDXdBe+RF0IXFvtyu4B72
    voK7+GhEwVf/qywsVl6+l0Ao6tuYm0Y2d3sOGpWkTCQ0MKmEkXZL/WmtCnJn
    9hodZZ/EsCgYEA585MygEOg4f40K5e57Dp697UCVwXLoO5R5FeS58NVScn29j9k0Ig
    OMqog9x0kJTNtzn4Uzdtx/pqM2Nz1lPlfi/jrc0zQ1x3so0O22y0D9W9YN17KtN16
    KJqa356sWezu2OEBuHa2+S3GAlVv1RPeTPnuOumKmP06DjDUGzLNCZy10CgYEa+zFw
    95ZDWhu5U02Z4waVaqEqgQUGQPkHrXTVx/IUunjDkDrLTVo2ZofDNTxrdl+UedMmOOc
    cieZn6Dhczd25fTkysySMH3g/q39pF0bUnGvgcXnEy0EyK0413x1j8TTCQzZXYAcq
    HMSx1s+nRSoncPtxYU05e9q6Q6bPCZC2C3t8FJ0AOGyEASU1AB/jkneFzjx]JG
    PdhQxubw8V9y864A2z1ah9t/k.kjZfYiAziAeq253r7tE74AGFTBTHH18e1Qoemi3P
    Wbc9Gv1Bf1s111bcP1yMrkPEP805QEXtnWxXTFGaJRGKiyi187e8rCAJ+W2Zho
    e/-1550XQfCyQ2yuMOqUCyYANAK2YRlaxaCk+NlSu6o7dmdnu6i5x7?POQ7E0
    OtMaXjua91p1vLZofamDMDuetoE7ZixSwBFpDLDp32Y4Agq1YFpt8gkJ4e
    Tw65y2CwL9P4b8s6m4qKUR6z0Dplwuw6EvvvsFKFQ6g/K3XCHP1akjekJm3Jg3I
    RnWvFqKZBnCeneM5zucei5L5rRttaJw/RB/18g1PMX7Wd7Q3QvLwpm4uOnm5Fp
    P1ZqA4yXK86f4u/+sSr1la9g68uqep6NyWyRqyX1Gza570i50ofwWt8beg+uk
    Z81L1J36xvP7WnujFFH10kk8buZMVK40UWyTeigHERMeP/xw
    -----END RSA PRIVATE KEY-----
  crt: |
    -----BEGIN CERTIFICATE-----
    MIIDUTCCAjmgAwIBAgIATFbA91MocWj9qOHiRd6qRvWMPppM6wDQQJfK9bZCnNQELLBQAw
    NsEQMA4GHAUECgHwNhnuQx4rzdTEjMCXEAGIWEawarkvQ/KFJvB3Q0QEQz9yb5E1
    YmVybmoD2ZMWhcHnNMnwMj42MDQzQHJmJWnehC4mJyMwHJA1MDQeZHmJ2WJ/A/MRAwDQYD

```
Note

This approach of specifying FEPCEerts 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>&quot;&quot;</td>
<td>&quot;&quot; It is fixed, and the parameter set in pgbackrest.conf is described from the line below.</td>
</tr>
<tr>
<td>spec.schedule.num</td>
<td>Integer</td>
<td>Number of schedules to set The maximum number of backup schedules is 5.</td>
</tr>
<tr>
<td>spec.schedule.schedule</td>
<td>-</td>
<td>Write the date and time of the Nth schedule in cron format. The date and time is UTC time.</td>
</tr>
</tbody>
</table>

---

END CERTIFICATE

---

END CERTIFICATE

---

END CERTIFICATE
### Field Details

<table>
<thead>
<tr>
<th>Field</th>
<th>Default</th>
<th>Details</th>
</tr>
</thead>
</table>
| spec.scheduleN.type | full/incr | full: Perform a full backup (Back up the contents of the database cluster).
|                |         | incr — Perform an incremental backup (Back up only the database cluster files that were changed to the last backup migration). |
| spec.preScript  | " "    | This parameter must specify a default value.                           |
| spec.postScript | " "    | This parameter must specify a default value.                           |

Example of FEPBackup CR created

```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"
  preScript: " 
  postScript: " 
pgbackrestParams: |
  # 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>spec.image</td>
<td>&lt;current-released-image&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>spec.imagePullPolicy</td>
<td>IfNotPresent</td>
<td></td>
</tr>
<tr>
<td>spec.mcSpec.limits</td>
<td>cpu: 0.2</td>
<td>memory: &quot;300Mi&quot;</td>
</tr>
</tbody>
</table>
### Field | Default | Details
--- | --- | ---
`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>` | Name of the FEPcluster to restore to
The exact restore destination volume is retrieved from FEPcluster
`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"

#### 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

```yaml
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
```
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.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>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.fepclusternamex</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;|&quot; and the Pgpool-II parameters. Refer to &quot;Pgpool-II parameters&quot; for detail.</td>
</tr>
<tr>
<td>Field</td>
<td>Default</td>
<td>Details</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>backend_weight1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>backend_flag0</td>
<td>‘ALWAYS_PRIMARY’</td>
<td></td>
</tr>
<tr>
<td>backend_flag1</td>
<td>‘DISALLOW_TO_FAILOVER’</td>
<td></td>
</tr>
<tr>
<td>connection_cache</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>max_pool</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>listen_backlog_multiplier</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>serialize_accept</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>child_life_time</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>client_idle_limit</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>child_max_connections</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>connection_life_time</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>reset_query_list</td>
<td>‘ABORT; DISCARD ALL’</td>
<td></td>
</tr>
<tr>
<td>client_min_messages</td>
<td>info</td>
<td></td>
</tr>
<tr>
<td>log_min_messages</td>
<td>debug1</td>
<td></td>
</tr>
<tr>
<td>log_statement</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_per_node_statement</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_client_messages</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_hostname</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_connections</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>log_line_prefix</td>
<td>‘%t: pid %p: ’</td>
<td></td>
</tr>
<tr>
<td>load_balance_mode</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>ignore_leading_white_space</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>white_function_list</td>
<td>”</td>
<td></td>
</tr>
<tr>
<td>black_function_list</td>
<td>‘currval,lastval,nextval,setval’</td>
<td></td>
</tr>
<tr>
<td>black_query_pattern_list</td>
<td>”</td>
<td></td>
</tr>
<tr>
<td>database_redirect_preference_list</td>
<td>”</td>
<td></td>
</tr>
<tr>
<td>app_name_redirect_preference_list</td>
<td>”</td>
<td></td>
</tr>
<tr>
<td>allow_sql_comments</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>disable_load_balance_on_write</td>
<td>”transaction”</td>
<td></td>
</tr>
<tr>
<td>statement_level_load_balance</td>
<td>on</td>
<td></td>
</tr>
<tr>
<td>sr_check_period</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>sr_check_user</td>
<td>‘postgres’</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></td>
<td>delay_threshold = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log_standby_delay = 'none'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssl = on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssl_ciphers = 'HIGH:MEDIUM:+3DES:!aNULL'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssl_prefer_server_ciphers = off</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssl_ecdh_curve = 'prime256v1'</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ssl_dh_params_file = ''</td>
<td></td>
</tr>
<tr>
<td></td>
<td>relcache_expire = 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>relcache_size = 256</td>
<td></td>
</tr>
<tr>
<td></td>
<td>check_temp_table = catalog</td>
<td></td>
</tr>
<tr>
<td></td>
<td>check_unlogged_table = on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>enable_shared_relcache = on</td>
<td></td>
</tr>
<tr>
<td></td>
<td>relcache_query_target = primary</td>
<td></td>
</tr>
<tr>
<td></td>
<td>wd_port0 = 9000</td>
<td></td>
</tr>
<tr>
<td>spec.custompcp</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>spec.customsslkey</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>spec.customsslcert</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>spec.customsslcacert</td>
<td>&quot;</td>
<td>&quot;</td>
</tr>
<tr>
<td>spec.customlogsize</td>
<td>100 Mi</td>
<td>Specifies the persistent volume size for log output.</td>
</tr>
<tr>
<td>spec.storageclassname</td>
<td></td>
<td>Specifies the storage class for log output.</td>
</tr>
</tbody>
</table>

**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></td>
<td>backend_flag0</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>backend_flag1</td>
<td>Y</td>
</tr>
<tr>
<td><strong>Connection pooling</strong></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><strong>Error reporting and log acquisition</strong></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><strong>Load sharing settings</strong></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><strong>Health check</strong></td>
<td>connect_timeout (integer)</td>
<td></td>
</tr>
<tr>
<td><strong>Streaming replication check</strong></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><strong>Secure Socket Layer (SSL)</strong></td>
<td>ssl (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>ssl_ciphers (string)</td>
<td>Y</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></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>recache_expire (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>recache_size (integer)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>enable_shared_recache (boolean)</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>recache_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>Change effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>.spec.targetClusterName</td>
<td>Must specify target FEP Cluster name within namespace mentioned in metadata.</td>
</tr>
<tr>
<td>.spec.targetPgpool2Name</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>Must specify action type.</td>
</tr>
<tr>
<td></td>
<td>Supported action types:</td>
</tr>
<tr>
<td></td>
<td>restart</td>
</tr>
<tr>
<td></td>
<td>pod_restart</td>
</tr>
<tr>
<td></td>
<td>reload</td>
</tr>
<tr>
<td></td>
<td>list</td>
</tr>
<tr>
<td></td>
<td>switchover</td>
</tr>
<tr>
<td></td>
<td>failover</td>
</tr>
<tr>
<td></td>
<td>pgpool2_restart</td>
</tr>
<tr>
<td>.spec.fepAction.args</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.sysExtraLogging</td>
<td>To turn extra debugging on, set value to true.</td>
</tr>
<tr>
<td></td>
<td>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>Failure</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>
apiVersion: fep.fujitsu.io/v1
kind: FEPAction

fepActionStatus:
  fepActionCondition: Success

metadata:
  name: new-fep-reload-action
  namespace: myns

spec:
  fepAction:
    args:
    - new-fep-sts-0
    - new-fep-sts-1
    type: reload
    sysExtraLogging: false
    targetClusterName: new-fep

---

Note
Please do not use the FEPAction to perform a switchover or restart while executing backup. Failed to get the backup.

1.2.10.1 FEPAction Specific Operation Details

**Action type - reload**

The reload action will manually reload the FEP database on the targeted FEPCluster.

“reload” action type expects users to specify the name of individual FEP pods that they want to run the database reload operation on. They specify that in the args section under the FEPAction CR spec as below:

```
spec:
  fepAction:
    args:
    - nf-131851-sts-0
    - nf-131851-sts-1
    type: reload
    targetClusterName: nf-131851
```

**Action type - restart**

The restart action will manually restart the FEP database on the targeted FEPCluster.

“restart” action type expects users to specify the name of individual FEP pods that they want to run the database restart operation on. They specify that in the args section under the FEPAction CR spec as below:

```
spec:
  fepAction:
    args:
    - nf-131851-sts-0
    - nf-131851-sts-1
    type: restart
    targetClusterName: nf-131851
```

**Action type - pod_restart**

The pod_restart action will restart specified list of POD for given target cluster. User can specify key word ‘ALL’ under ‘args’ section to restart all pods in target cluster. Alternatively, user can give the list of pods to be started in target cluster. User should either give ALL or the list of the pods.

This action restarts the replica pods first. Once all replicas have been restarted, it switches over the mastership to one of the replica before restarting old master pod. If it is a single node cluster, master will be restarted in its current state. This action is automatically created to
restart pods when image or machine specs are changed for fep or backup container depending on autoPodRestart flag in FEPCluster CR (see more details in FEPCluster CR section):

```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 current leader/primary pod that they want to switchover from. They specify that in the args section under the FEPAction CR spec as below:

```yaml
spec:
fepAction:
  args:
  - nf-131851-sts-2
  type: switchover
  targetClusterName: nf-131851
```

Here, nf-131851-sts-2 is the current primary.

**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:
spec:
  fepAction:
    type: pgpool2_restart
    targetPgpool2Name: nf-131851-pgpool2

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>spec.prometheus.tls</td>
<td></td>
<td>Prometheus MTLS spec section</td>
</tr>
<tr>
<td>spec.prometheus.tls.certificateName</td>
<td>Optional</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>
</tr>
<tr>
<td>spec.prometheus.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</td>
<td></td>
<td>Exporter spec section</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.secretName</td>
<td></td>
<td>Base Authentication secret to provide username &amp; encrypted password of user</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.userName</td>
<td></td>
<td>Secret name</td>
</tr>
<tr>
<td>spec.fepExporter.authSecret.passwordKey</td>
<td></td>
<td>Key of username 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>Optional</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>
</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not defined or set to false =&gt; Create default queries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Defined and set to true =&gt; Do not create default queries.</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.disableDefaultAlertRules</td>
<td>false</td>
<td>Optional Not 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 Optional If not specified; image name is picked up from operator environment variable</td>
</tr>
<tr>
<td>spec.fepExporter.image.pullPolicy</td>
<td></td>
<td>IfNotPresent Always or IfNotPresent</td>
</tr>
<tr>
<td>spec.fepExporter.mcSpec.limits</td>
<td></td>
<td>Max CPU allocated to exporter container Max memory allocated to exporter container</td>
</tr>
<tr>
<td>spec.fepExporter.mcSpec.requests</td>
<td></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. <em>It can be turned on/off at any time</em></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:
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>Do not perform auto scale-out</td>
<td>[cpu_utilization]</td>
</tr>
<tr>
<td>spec.scaleout.threshold</td>
<td>40</td>
<td>Threshold (Average of all replicas)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Must be an integer. The unit is %.</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>(Value out of range)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Do not perform auto scale-out</td>
</tr>
</tbody>
</table>
Appendix A Default Metrics Queries

pg_capacity_connection:
query: |
select sys, idle, idleintx, idleintx10min, idleintx1hour, idleintx1day, idleintx1week,
(curr.idle + curr.idleintx + curr.active) total, s.setting "max" from
{
    select
    count(CASE WHEN a.state is null THEN 1 END) sys,
    count(CASE WHEN a.state='idle' THEN 1 END) idle,
    count(CASE WHEN a.state='idle in transaction' OR a.state='idle in transaction (aborted)' THEN 1 END) idleintx,
    count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND age(now(), state_change) > interval '10 min' THEN 1 END) idleintx10min,
    count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND age(now(), state_change) > interval '1 hour' THEN 1 END) idleintx1hour,
    count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND age(now(), state_change) > interval '1 day' THEN 1 END) idleintx1day,
    count(CASE WHEN (a.state='idle in transaction' OR a.state='idle in transaction (aborted)') AND age(now(), state_change) > interval '1 week' THEN 1 END) idleintx1week,
    count(CASE WHEN a.state='active' THEN 1 END) active
from pg_stat_activity a
) curr, pg_settings s where name = 'max_connections'
master: true
metrics:
- sys:
    usage: 'GAUGE'
    description: 'Number of system connections.'
- idle:
    usage: 'GAUGE'
    description: 'Number of idle connections.'
- idleintx:
    usage: 'GAUGE'
    description: 'Number of idle in transaction connections.'
- idleintx10min:
    usage: 'GAUGE'
    description: 'Number of idle in transaction connections running longer than 10 min.'
- idleintx1hour:
    usage: 'GAUGE'
    description: 'Number of idle in transaction connections running longer than 1 hour.'
- idleintx1day:
    usage: 'GAUGE'
    description: 'Number of idle in transaction connections running longer than 1 day.'
- idleintx1week:
    usage: 'GAUGE'
    description: 'Number of idle in transaction connections running longer than 1 week.'
- total:
    usage: 'GAUGE'
    description: 'Number of total connections.'
- max:
    usage: 'GAUGE'
    description: 'Max number of connections.'

pg_capacity_schema:
query: |
SELECT current_database() AS database_name, table_schema,
COALESCE(SUM(pg_total_relation_size('"'||table_schema||'".'||table_name||'"')) , 0) AS size
FROM information_schema.tables GROUP BY table_schema
master: true
metrics:
- database_name:
    usage: 'LABEL'
- size:
    usage: 'GAUGE'
    description: 'Size of database.'
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.'
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.relnamename,'?') 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-nullfrac)*avg_width) AS datawidth,
    MAX(nullfrac) AS maxfracsum,
    hdr+
        SELECT 1+count(*)/8
        FROM pg_stats s2
        WHERE nullfrac<>0 AND s2.schemaname = s.schemaname AND s2.tablename = s.tablename
    ) AS nullhdr
FROM pg_stats s, (SELECT
    (SELECT current_setting('block_size')::numeric) AS bs,
    CASE WHEN substring(v,12,3) IN ('8.0','8.1','8.2') THEN 27 ELSE 23 END AS hdr,
    CASE WHEN v ~ 'mingw32' THEN 8 ELSE 4 END AS ma
FROM (SELECT version() AS v) AS foo
) AS constants
GROUP BY 1,2,3,4,5
) AS foo
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: 'LABEL'
    description: 'Database name.'
  - schemaname: 'LABEL'
    description: 'Schema name.'
  - relname: 'LABEL'
    description: 'Name of this table.'
  - wastedbytes: '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.objid IS NOT DISTINCT FROM blocked_locks.objid
      AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
      AND blocking_locks.pid != blocked_locks.pid
      JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid = blocking_locks.pid
      WHERE NOT blocked_locks.GRANTED
    master: true
    metrics:
      - blocked_pid:
          usage: 'LABEL'
          description: 'Blocked process id.'
      - blocked_user:
          usage: 'LABEL'
          description: 'Blocked user.'
      - blocking_pid:
          usage: 'LABEL'
          description: 'Blocking process id.'
      - blocking_user:
          usage: 'LABEL'
          description: 'Blocking user.'
      - blocked_statement:
          usage: 'LABEL'
          description: 'Blocked statement.'
      - locks:
          usage: 'GAUGE'
          description: 'Number of processes in blocked state.'
```

```
pg_performance_locking:
    query: |
      WITH
      locks as (SELECT blocked_locks.DATABASE, count(blocked_locks.pid) locks
      FROM pg_catalog.pg_locks blocked_locks
      JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid = blocked_locks.pid
      JOIN pg_catalog.pg_locks blocking_locks
      ON blocking_locks.locktype = blocked_locks.locktype
      AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
      AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
      AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
      AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
      AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
      AND blocking_locks.transactionid IS NOT DISTINCT FROM blocked_locks.transactionid
      AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
      AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
      JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid = blocking_locks.pid
      WHERE NOT blocked_locks.GRANTED
```

- 42 -
AND blocking_locks.pid != blocked_locks.pid
JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
WHERE NOT blocked_locks.GRANTED group by blocked_locks.DATABASE
),
dbs as {
    select * from pg_catalog.pg_database
}
select dbs.datname, coalesce(locks.locks, 0) locks from dbs left join locks on dbs.oid=DATABASE
master: true
metrics:
  - datname:
      usage: 'LABEL'
      description: 'Database name'
  - locks:
      usage: 'GAUGE'
      description: 'Number of processes in blocked state.'

pg_replication:
query: |
SELECT CASE WHEN pg_last_wal_receive_lsn() = pg_last_wal_replay_lsn() THEN 0 ELSE GREATEST (0, EXTRACT(EPOCH FROM (now() - pg_last_xact_replay_timestamp()))) END AS lag
master: true
metrics:
  - lag:
      usage: "GAUGE"
      description: "Replication lag behind master in seconds"

pg_postmaster:
query: |
SELECT pg_postmaster_start_time as start_time_seconds from pg_postmaster_start_time()
master: true
metrics:
  - start_time_seconds:
      usage: "GAUGE"
      description: "Time at which postmaster started"

pg_stat_user_tables:
query: |
SELECT 
current_database() datname,
schemaname,
relname,
seq_scan,
seq_tup_read,
idx_scan,
idx_tup_fetch,
n_tup_ins,
n_tup_upd,
n_tup_del,
n_tup_hot_upd,
n_live_tup,
n_dead_tup,
n_mod_since_analyze,
last_vacuum,
last_autovacuum,
last_analyze,
last_autoanalyze,
vacuum_count,
autovacuum_count,
analyze_count,
autoanalyze_count
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, min_plan_time / 1000 as min_plan_time_seconds, max_plan_time / 1000 as max_plan_time_seconds, min_exec_time / 1000 as min_exec_time_seconds, max_exec_time / 1000 as max_exec_time_seconds, min_plan_time / 1000 as min_plan_time_seconds, max_plan_time / 1000 as max_plan_time_seconds, mean_plan_time / 1000 as mean_plan_time_seconds, mean_exec_time / 1000 as mean_exec_time_seconds, stddev_plan_time / 1000 as stddev_plan_time_seconds, stddev_exec_time / 1000 as stddev_exec_time_seconds, rows, shared_blks_hit, shared_blks_read, shared_blks_dirtied, shared_blks_written, local_blks_hit, local_blks_read, local_blks_dirtied, local_blks_written, temp_blks_read, temp_blks_written, blk_read_time / 1000 as blk_read_time_seconds, blk_write_time / 1000 as blk_write_time_seconds FROM pg_stat_statements t1 JOIN pg_roles t2 ON (t1.userid=t2.oid) JOIN pg_database t3 ON (t1.dbid=t3.oid) WHERE t2.rolname != 'rdsadmin'

master: true

metrics:
- rolname:
  usage: "LABEL"
  description: "Name of user"
- datname:
  usage: "LABEL"
  description: "Name of database"
- queryid:
  usage: "LABEL"
  description: "Query ID"
- calls:
  usage: "COUNTER"
  description: "Number of times executed"
- total_plan_time_seconds:
  usage: "COUNTER"
  description: "Total plan time spent in the statement, in milliseconds"
- total_exec_time_seconds:
  usage: "COUNTER"
  description: "Total exec time spent in the statement, in milliseconds"
- min_plan_time_seconds:
  usage: "GAUGE"
  description: "Minimum plan time spent in the statement, in milliseconds"
- min_exec_time_seconds:
  usage: "GAUGE"
  description: "Minimum exec time spent in the statement, in milliseconds"
- max_plan_time_seconds:
  usage: "GAUGE"
  description: "Maximum plan time spent in the statement, in milliseconds"
- max_exec_time_seconds:
  usage: "GAUGE"
  description: "Maximum exec time spent in the statement, in milliseconds"
- mean_plan_time_seconds:
  usage: "GAUGE"
  description: "Mean plan time spent in the statement, in milliseconds"
- mean_exec_time_seconds:
  usage: "GAUGE"
  description: "Mean exec time spent in the statement, in milliseconds"
- stddev_plan_time_seconds:
  usage: "GAUGE"
  description: "Population standard deviation of plan time spent in the statement, in milliseconds"
- stddev_exec_time_seconds:
  usage: "GAUGE"
  description: "Population standard deviation of exec time spent in the statement, in milliseconds"
- rows:
  usage: "COUNTER"
  description: "Total number of rows retrieved or affected by the statement"
- shared_blks_hit:
usage: "COUNTER"
description: "Total number of shared block cache hits by the statement"
- shared_blks_read:
  usage: "COUNTER"
description: "Total number of shared blocks read by the statement"
- shared_blks_dirtied:
  usage: "COUNTER"
description: "Total number of shared blocks dirtied by the statement"
- shared_blks_written:
  usage: "COUNTER"
description: "Total number of shared blocks written by the statement"
- local_blks_hit:
  usage: "COUNTER"
description: "Total number of local block cache hits by the statement"
- local_blks_read:
  usage: "COUNTER"
description: "Total number of local blocks read by the statement"
- local_blks_dirtied:
  usage: "COUNTER"
description: "Total number of local blocks dirtied by the statement"
- local_blks_written:
  usage: "COUNTER"
description: "Total number of local blocks written by the statement"
- temp_blks_read:
  usage: "COUNTER"
description: "Total number of temp blocks read by the statement"
- temp_blks_written:
  usage: "COUNTER"
description: "Total number of temp blocks written by the statement"
- blk_read_time_seconds:
  usage: "COUNTER"
description: "Total time the statement spent reading blocks, in milliseconds (if track_io_timing is enabled, otherwise zero)"
- blk_write_time_seconds:
  usage: "COUNTER"
description: "Total time the statement spent writing blocks, in milliseconds (if track_io_timing is enabled, otherwise zero)"
Appendix B Default Alert Rules

```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
          labels:
            severity: warning
        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
          labels:
            severity: warning
        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
          labels:
            severity: warning
        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