



# **EDB™ Postgres on Kubernetes**

*Release 2.7*

**Quick Start**

**Feb 04, 2020**

---

## Contents

---

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>What's New</b>   | <b>2</b>  |
| <b>2</b> | <b>EDB Containers</b>   | <b>3</b>  |
| 2.1      | Architecture Overview . . . . .                                   | 4         |
| <b>3</b> | <b>Deploying on Google Kubernetes</b>                             | <b>5</b>  |
| 3.1      | Deployment using Helm Chart . . . . .                             | 5         |
| <b>4</b> | <b>Deploying on RedHat Openshift</b>                              | <b>7</b>  |
| <b>5</b> | <b>Deploying from the Docker Command Line</b>                     | <b>8</b>  |
| 5.1      | Creating a Container at the Docker Command Line . . . . .         | 8         |
| 5.1.1    | Deploying Advanced Server in a Container (Master Node) . . . . .  | 9         |
| 5.1.2    | Deploying Advanced Server in a Container (Standby Node) . . . . . | 11        |
| 5.1.3    | Deploying pgPool in a Container . . . . .                         | 13        |
| 5.1.4    | Deploying BART in a Container . . . . .                           | 14        |
| <b>6</b> | <b>Appendix</b>   | <b>16</b> |
| 6.1      | Sample GKE Setup . . . . .  | 16        |
| <b>7</b> | <b>Conclusion</b>   | <b>18</b> |
|          | <b>Index</b>  | <b>19</b> |

This document provides shortcuts that allow you to quickly install, configure, and deploy the EDB Postgres Containers in the following environments:

- [Red Hat OpenShift](#)
- [Google Kubernetes Engine \(GKE\)](#)
- [Docker Command Line](#)

For detailed information about Containers installation and configuration, see:

- [EDB™ Postgres Containers and Integration with OpenShift](#)
- [EDB™ Postgres Containers and Integration with Google Kubernetes Engine.](#)

# CHAPTER 1

---

## What's New

---

The following changes are added to EDB™ Postgres on Kubernetes Quick Start Guide to create version 2.7:

- Starting this release, the containers can be deployed in OpenShift using the following methods in addition to using templates:
  - Operator
  - Helm chart
- Locale support for Advanced Server

To deploy a container, you must have access to the registry in which the containers are stored (containers.enterprisedb.com). To receive credentials that allow you to access the container registry, please submit a request at: <https://www.enterprisedb.com/general-inquiry-form>. The following containers are available in the EDB container registry at containers.enterprisedb.com:

**EDB Postgres Advanced Server with Failover Management and Monitoring components:**

- containers.enterprisedb.com/edb/edb-as:v12
- containers.enterprisedb.com/edb/edb-as:v11
- containers.enterprisedb.com/edb/edb-as-gis:v11
- containers.enterprisedb.com/edb/edb-as-gis:v12
- containers.enterprisedb.com/edb/edb-as:v10
- containers.enterprisedb.com/edb/edb-as-gis:v10
- containers.enterprisedb.com/edb/edb-as-gis23:v10
- containers.enterprisedb.com/edb/edb-as:v9.6

**EDB Postgres Advanced Server only:**

- containers.enterprisedb.com/edb/edb-as-lite:v12
- containers.enterprisedb.com/edb/edb-as-lite:v11
- containers.enterprisedb.com/edb/edb-as-lite:v10

**EDB Backup and Recovery Tool (BART):**

- containers.enterprisedb.com/edb/edb-bart:v2.5

**EDB Postgres Enterprise Manager (PEM):**

- containers.enterprisedb.com/edb/edb-pemserver:v7.12

**EDB PgPool:**

- containers.enterprisedb.com/edb/edb-pgpool:v4.0

**Note:** You can also download all of these images as tarballs from the EDB website.

The containers use UBI (Universal Base Image) Minimal image from Red Hat as the base OS. For more information about UBI images, refer to: <https://developers.redhat.com/products/rhel/ubi/>

## 2.1 Architecture Overview

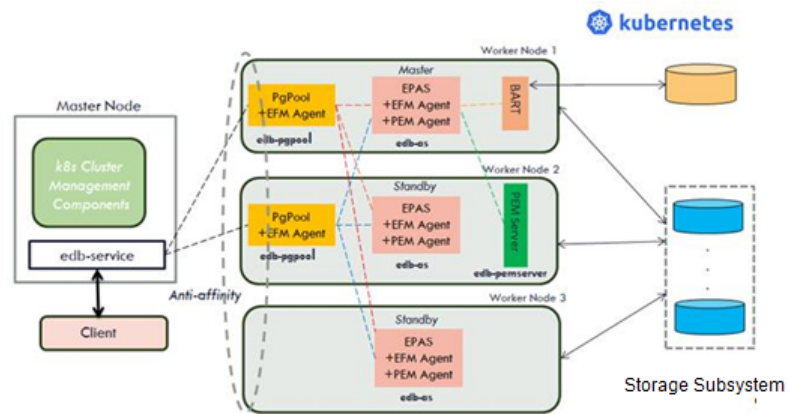


Fig. 1.1: Architecture overview

Fig. 2.1: Architecture overview

---

## Deploying on Google Kubernetes

---

Before deploying the EDB Postgres Platform for Containers on GKE, you must have:

- A fully functional GKE environment with at least three worker nodes.
- A Helm installation. For more information about Helm, see [Helm documentation](#).

Containers can be deployed in GKE using one of the two following methods:

- Helm chart
- EDB Postgres Advanced Server (EPAS) Operator (installed via helm chart)

For deployment using the Advanced Server Operator, refer to the [Advanced Server Operator Guide](#).

### 3.1 Deployment using Helm Chart

To deploy the containers in your GKE environment, complete the following steps:

1. Download the following sample files:
  - Sample helm chart: `edb-2.7.0.tgz`
  - Sample values.yaml file: `sample-values.yaml`
2. Create your own values.yaml file with customized input values using the sample values.yaml file as a template (e.g. myvalues.yaml).
3. Enter the following command to deploy the containers with your custom inputs in your namespace:

```
helm install edb-2.7.0.tgz -f myvalues.yaml
```

**Note:** In the example, myvalues.yaml is the user supplied values.yaml file containing the desired parameter values.

4. Once the helm chart is successfully installed, the following EDB containers should be available in your environment:
  - **EDB Postgres Advanced Server pod (includes EDB Failover Manager)**

- Container image: edb-as:v12/ edb-as:v11 / edb-as:v10
- Number of replicas: 3
- Deployed as: Stateful Set
- Each EPAS pod is deployed on a separate node (anti-affinity rule applied)

- **EDB PgPool pod**

- Container image: edb-pgpool:v4.0
- Number of replicas: 2
- Deployed as: Replica Set
- Each PgPool container is set on a separate node (anti-affinity rule applied)

- **(Optionally) EDB BART (Backup and Recovery Tool) pod**

- Container image: edb-bart:v2.5
- Number of replicas: 1
- Deployed as: Replica Set

5. Refer to *Sample setup on GKE* for more details.

6. Visit the Google Cloud Platform console at <https://console.cloud.google.com>.



---

## Deploying on RedHat Openshift

---

Some prerequisite steps and advanced management tasks must be performed at the OpenShift command line. For detailed information about using the OpenShift command line, see the [OpenShift documentation](#).

Before deploying the EDB Postgres Platform for Containers, you must complete the following steps:

1. Create an OpenShift project that will describe the resource allocations and template information for the container.
2. Grant required privileges to the project.
3. Create and configure the volume resources that the Advanced Server and BART containers will use.
4. Create a registry secret that contains connection information for the container repository.
5. Create a template.

After creating templates, you can customize the parameter values during deployment. You can also download the sample files for your reference.

For detailed information on deployment and using the console, see [EDB™ Postgres Containers and Integration with OpenShift](#).

---

## Deploying from the Docker Command Line

---

You can use the Docker command-line client to deploy, manage, and use an Advanced Server container. We recommend including the following docker command options when using the command line:

`-d`

The `-d` option forces docker to run in the background as a daemon. This option is optional.

`--privileged`

The `--privileged` option may be required if local security settings do not permit mounting a local volume with read-write options. As an alternative, we recommend allowing your security settings to permit the container to have read-write access to the mounted volume. If applicable, adjust your SELinux settings to allow access.

`--restart=always`

This option specifies your restart preferences for the container. Refer to the Docker documentation for details about the supported options. For more information about docker commands and command options, see [Docker base commands](#).

### 5.1 Creating a Container at the Docker Command Line

Use the `docker run` command to deploy a container at the command line. You can specify environment variables at the command line (when invoking the `docker run` command), or (for convenience) in a file located in the network mountpoint. See the [Docker documentation](#) for a complete list of command options.

The number of nodes that you must deploy will depend on the features that you wish to use with your system. For example, if you are using EFM functionality, you will need to create at least one master and two standby nodes.

When you use the Docker command line to create standby nodes or containers that provide complimentary functionality for Advanced Server, you specify information about the configuration of the cluster, and the names of master or standby nodes of the cluster. Please note that information provided in the container configuration must be consistent across all nodes of the cluster.

### 5.1.1 Deploying Advanced Server in a Container (Master Node)

The master node of a cluster contains an instance of Advanced Server and a supporting Failover Manager deployment. When deploying a master node, include the following syntax:

```
sudo docker run \
  --name "master_name" \
  -e DATABASE_NAME="db_name" \
  -e PGPORT=db_listener_port \
  -e DATABASE_USER="db_user_name" \
  -e DATABASE_USER_PASSWORD="db_user_password" \
  -e ENTERPRISEDB_PASSWORD="enterprisedb_password" \
  -e LOCALEPARAMETER="locale" \
  -e NAMESERVER="nameserver_ipaddress" \
  -e REPL_USER="repl_user_name" \
  -e REPL_PASSWORD="repl_user_password" \
  -e ENABLE_HA_MODE=Yes|No \
  -e ENABLE_ARCHIVE_MODE=Yes|No \
  -v host_data_volume:/edbvolume \
  -v host_archive_volume:/edbarchive \
  -p host_port:db_listener_port \
  -e ACCEPT_EULA=Yes|No \
  -d containers.enterprisedb.com/edb/edb-as:v12
```

Include the `--name` option to specify the name of the master node of the cluster. Include the `-e` option with the `docker run` command to specify values for environment variables used to configure your container. Include the following values when deploying a master node that includes Advanced Server and Failover Manager:

Use the `DATABASE_NAME` environment variable to specify the name of the Advanced Server database.

Use the `PGPORT` environment variable to specify the listener port of the Advanced Server database (by default, 5444).

Use the `DATABASE_USER` environment variable to specify the name of a database superuser that will be created when the database is initialized; by default, the database superuser is named `enterprisedb`.

If you specify the default (`enterprisedb`), the user will be associated with the password provided by the `ENTERPRISEDB_PASSWORD` environment variable.

Use the `DATABASE_USER_PASSWORD` environment variable to specify the password associated with the database superuser if you specify a `db_user_name` other than `enterprisedb`. Please note that this password should not be changed after the pod is initialized.

Use the `ENTERPRISEDB_PASSWORD` environment variable to specify the password associated with the default database superuser (`enterprisedb`). During the installation process, the container creates a database superuser named `enterprisedb`. Please note that this password should not be changed after the pod is initialized.

Use the `LOCALEPARAMETER` environment variable to specify the locale that will be used by the container.

Use the `INITDBOPTS` parameter to provide non-default options for the `initdb` operation, e.g., this parameter can be used to disable Oracle compatibility.

Use the `NAMESERVER` environment variable to specify the identity of a name server that will be used for notifications from Failover Manager.

Use the `REPL_USER` environment variable to specify the name of the Postgres streaming replication user.

Use the `REPL_PASSWORD` environment variable to specify the password associated with the replication user.

Use the `ENABLE_HA_MODE` environment variable to specify whether failover should be enabled.

Use the `ENABLE_ARCHIVE_MODE` environment variable to specify whether archiving should be enabled. If `ENABLE_HA_MODE` is set to `Yes`, then `ENABLE_ARCHIVE_MODE` must be set to `Yes` also.

**Map the mount point of the data volume of the container, /edbvolume, to a directory on the host system using the -v option; for**

```
-v /data/edbas:/edbvolume maps the host directory /data/edbas to /  
edbvolume inside the container.
```

If `ENABLE_ARCHIVE_MODE` is set to Yes, then map the mount point of the archive volume of the container, /edbarchive, to a directory on the host system using the -v option; for example, the following option:

```
-v /archive/edbas:/edbarchive maps the host directory /archive/edbas  
to /edbarchive inside the container.
```

Use the `ACCEPT_EULA` environment variable to indicate if you accept or decline the EnterpriseDB license. The license is available for review at:

<https://www.enterprisedb.com/node/4509>

Note: You must accept the license agreement to deploy a container.

Include the `-p` option to enable port forwarding from the container to the host machine; for example, `-p 5445:5444` makes the listener port 5444 in the container available on port 5445 on the host system.

Include the `-d` option to indicate that the container should be deployed in detached mode; this enables background processing for the deployment.

After providing values for the environment variable and including any docker options, specify the repository name and image you wish to use.

## 5.1.2 Deploying Advanced Server in a Container (Standby Node)

A standby container hosts an instance of Advanced Server that is replicating the Advanced Server master node. If Failover Manager detects a failure on the part of the master node, a standby will assume the role of master node for the cluster. When deploying a standby node, include the following syntax:

```
sudo docker run --name "standby_name" \
  -e DATABASE_NAME="db_name" \
  -e PGPORT=db_listener_port \
  -e DATABASE_USER="db_user_name" \
  -e DATABASE_USER_PASSWORD="db_user_password" \
  -e ENTERPRISEDB_PASSWORD="enterprisedb_password" \
  -e LOCALEPARAMETER="locale" \
  -e NAMESERVER="nameserver_ipaddress" \
  -e REPL_USER="repl_user_name" \
  -e REPL_PASSWORD="repl_user_password" \
  -e MASTER_HOST "`sudo docker inspect --format '{{
  -e MASTER_PORT=5444 \
  -e ENABLE_HA_MODE=Yes|No \
  -e ENABLE_ARCHIVE_MODE=Yes|No \
  -v host_data_volume:/edbvolume \
  -v host_archive_volume:/edbarchive\
  -p host_port:db_listener_port \
  -e ACCEPT_EULA=Yes|No \
  -d containers.enterprisedb.com/edb/edb-as:v12
```

Include the `--name` option to specify the name of the master node of the cluster. Include the `-e` option with the `docker run` command to specify values for environment variables used to configure your container. Include the following values when deploying a master node that includes Advanced Server and Failover Manager:

Use the `DATABASE_NAME` environment variable to specify the name of the Advanced Server database.

Use the `PGPORT` environment variable to specify the listener port of the Advanced Server database (by default, 5444).

Use the `DATABASE_USER` environment variable to specify the name of a database superuser that will be created when the database is initialized; by default, the database superuser is named `enterprisedb`.

If you specify the default (`enterprisedb`), the user will be associated with the password provided by the `ENTERPRISEDB_PASSWORD` environment variable.

Use the `DATABASE_USER_PASSWORD` environment variable to specify the password associated with the database superuser if you specify a `db_user_name` other than `enterprisedb`. Please note that this password should not be changed after the pod is initialized.

Use the `ENTERPRISEDB_PASSWORD` environment variable to specify the password associated with the default database superuser (`enterprisedb`). During the installation process, the container creates a database superuser named `enterprisedb`. Please note that this password should not be changed after the pod is initialized.

Use the `LOCALEPARAMETER` environment variable to specify the locale that will be used by the container.

Use the `INITDBOPTS` parameter to provide non-default options for the `initdb` operation, e.g., this parameter can be used to disable Oracle compatibility.

Use the `NAMESERVER` environment variable to specify the identity of a name server that will be used for notifications from Failover Manager.

Use the `REPL_USER` environment variable to specify the name of the Postgres streaming replication user.

Use the `REPL_PASSWORD` environment variable to specify the password associated with the replication user.

Use the `MASTER_HOST` environment variable to specify the ip address of the master node of the replication cluster. The ip address of the master node can be retrieved using the `docker inspect` command the cluster based on the name of the master node as shown below:

```
sudo docker inspect --format '{{ .NetworkSettings. IPAddress }}'
master_name
```

Use the `STANDY_HOSTS` environment variable to specify a comma-separated list of ip addresses of the standby nodes in the replication cluster.

Use the `ENABLE_HA_MODE` environment variable to specify whether failover should be enabled.

Use the `ENABLE_ARCHIVE_MODE` environment variable to specify whether archiving should be enabled. If `ENABLE_HA_MODE` is set to Yes, then `ENABLE_ARCHIVE_MODE` must be set to Yes also.

Map the mount point of the data volume of the container, `/edbvolumes`, to a directory on the host system using the `-v` option; for example, the following option:

**`-v /data/edbas:/edbvolumes`** maps the host directory `/data/edbas` to `/edbvolumes` inside the container.

If `ENABLE_ARCHIVE_MODE` is set to Yes, then map the mount point of the archive volume of the container, `/edbarchive`, to a directory on the host system using the `-v` option; for example, the following option:

**`-v /archive/edbas:/edbarchive`** maps the host directory `/archive/edbas` to `/edbarchive` inside the container.

Use the `ACCEPT_EULA` environment variable to indicate if you accept or decline the EnterpriseDB license. The license is available for review at:

<https://www.enterprisedb.com/node/4509>

Note: You must accept the license agreement to deploy a container.

Include the `-p` option to enable port forwarding from the container to the host machine; for example, `-p 5445:5444` makes the listener port 5444 in the container available on port 5445 on the host system.

Include the `-d` option to indicate that the container should be deployed in detached mode; this enables background processing for the deployment.

After providing values for the environment variable and including any docker options, specify the repository name and image you wish to use.

Replace `master_name` with the name of the master node of the cluster.

### 5.1.3 Deploying pgPool in a Container

pgPool is a connection pooler for Advanced Server. When deploying a pgPool container, include the following syntax:

```
docker run --name "container_name" \
  -e DATABASE_NAME="db_name" \
  -e PGPORT=db_listener_port \
  -e ENTERPRISEDB_PASSWORD="enterprisedb_password" \
  -e REPL_USER="repl_user_name" \
  -e REPL_PASSWORD="repl_user_password" \
  -e MASTER_HOST=master_ipaddress \
  -e STANDBY_HOSTS=standby_ipaddress_list \
  -e ACCEPT_EULA=Yes|No \
  -d containers.enterprisedb.com/edb/edb-pgpool:v4.0
```

Include the `--name` option to specify the name of the container in which pgPool resides.

Use the `-e` option with the `docker run` command to specify values for environment variables used to configure your container. Include the following values when deploying a master node that includes Advanced Server and Failover Manager:

Use the `DATABASE_NAME` environment variable to specify the name of the Advanced Server database.

Use the `PGPORT` environment variable to specify the listener port of the Advanced Server database (by default, 5444).

Use the `ENTERPRISEDB_PASSWORD` environment variable to specify the password associated with the default database superuser (`enterprisedb`). During the installation process, the container creates a database superuser named `enterprisedb`. Please note that this password should not be changed after the pod is initialized.

Use the `EPL_USER` environment variable to specify the name of the Postgres streaming replication user.

Use the `REPL_PASSWORD` environment variable to specify the password associated with the replication user.

Use the `MASTER_HOST` environment variable to identify the master node of the replication cluster. The clause uses a call to `docker inspect` to retrieve the address of the master node of the cluster:

```
MASTER_HOST="`sudo docker inspect --format '{{ .NetworkSettings.
IPAddress }}'master_name`".
```

Use the `STANDY_HOSTS` environment variable to specify a comma-separated list of ip addresses of the standby nodes in the replication cluster.

Use the `ACCEPT_EULA` environment variable to indicate if you accept or decline the EnterpriseDB license. The license is available for review at:

<https://www.enterprisedb.com/node/4509>

Note: You must accept the license agreement to deploy a container.

Include the `-d` option to indicate that the container should be deployed in detached mode; this enables background processing for the deployment.

After providing values for the environment variable and including any docker options, specify the repository name and image you wish to use.

## 5.1.4 Deploying BART in a Container

EDB Postgres Backup and Recovery Tool (BART) simplifies backup management for Advanced Server databases. When deploying BART in a container, include the following syntax:

```
docker run \
  --name "container_name" \
  -e DATABASE_NAME=db_name \
  -e BART_HOST_ADDRESS=host_address \
  -e PGHOST="`sudo docker inspect --format '{{.NetworkSettings.IPAddress }}'
  → 'master_name`" \
  -e PGPORT=db_listener_port \
  -e REPL_USER="repl_user_name" \
  -e REPL_PASSWORD="repl_user_password" \
  -e BART_AUTOMATE_BACKUP=Yes|No \
  -e BART_NUM_BACKUPS_TO_KEEP=2 \
  -e BART_COMPRESS_BACKUP=Yes|No \
  -e ACCEPT_EULA=Yes \
  -v hostvolume:/edbbakup \
  -d containers.enterprisedb.com/edb/edb-bart:v2.5
```

Include the `--name` option to specify the name of the container in which BART will reside.

Include the `-e` option with the `docker run` command to specify values for environment variables used to configure your container. Include the following values when deploying a master node that includes Advanced Server and Failover Manager:

Use the `DATABASE_NAME` environment variable to specify the name of the Advanced Server database that will be backed up.

Use the `BART_HOST_ADDRESS` environment variable to provide the IP address or location of the BART host.

Use the `PGHOST` environment variable to identify the master node of the replication cluster. The clause uses a call to `docker inspect` to retrieve the address of the master node of the cluster:

```
PGHOST="`sudo docker inspect --format '{{ .NetworkSettings.IPAddress }}'
  → 'master_name`".
```

Replace `master_name` with the name of the master node of the cluster.

Use the `PGPORT` environment variable to specify the listener port of the Advanced Server database (by default, 5444).

Use the `REPL_USER` environment variable to specify the name of the Postgres streaming replication user.

Use the `REPL_PASSWORD` environment variable to specify the password associated with the replication user.

Use the `BART_AUTOMATE_BACKUP` environment variable to specify if BART should automate the backup process.

If `BART_AUTOMATE_BACKUP` is set to `Yes`, use the `BART_BACKUP_SCHEDULE` environment variable to specify the schedule for the backup process in cron format.

Use the `BART_NUM_BACKUPS_TO_KEEP` environment variable to specify the number of backups retained by BART.

Use the `BART_COMPRESS_BACKUP` environment variable to specify if BART should compress the backup. Specify `yes` to indicate that backups should be compressed, or `no` to indicate that they should not be compressed.



Use the `ACCEPT_EULA` environment variable to indicate if you accept or decline the EnterpriseDB license. The license is available for review at:

<https://www.enterprisedb.com/node/4509>

Note: You must accept the license agreement to deploy a container.

Map the mount point of the backup volume of the container, `/eddbackup`, to a directory on the host system using the `-v` option; for example, the following option:

```
-v /backup/edbas:/eddbackup maps the host directory /backup/edbas to /eddbackup inside the container.
```

Include the `-d` option to indicate that the container should be deployed in detached mode; this enables background processing for the deployment.

After providing values for the environment variable and including any docker options, specify the repository name and image you wish to use.

## 6.1 Sample GKE Setup

To deploy EDB containers on GKE, you must complete the following steps:

1. Set up a client machine with the following tools:
  - Google Cloud SDK - <https://cloud.google.com/sdk/docs/quickstarts>
  - Kubernetes CLI Tool (kubectl) - <https://kubernetes.io/docs/tasks/tools/install-kubectl>
  - Helm - [https://helm.sh/docs/using\\_helm](https://helm.sh/docs/using_helm)
2. Get GKE cluster credentials to access the cluster from your client machine.

**Note:** If you don't have an existing Kubernetes cluster to work with, you can setup a 3 node cluster in the [Google Cloud Platform](#).

**Tip:** Use defaults but increase number of CPUs to 2 and increase memory to 13G

To get the credentials for the GKE cluster, run the following command on the client machine after replacing the appropriate values for cluster-name, time-zone, and project-name:

```
gcloud container clusters get-credentials cluster-name \  
--zone time-zone \  
--project project-name
```

3. Setup Helm on the client machine by running the following commands:

```
kubectl -n kube-system create serviceaccount tiller  
kubectl create clusterrolebinding tiller \  
--clusterrole cluster-admin \  
--serviceaccount=kube-system:tiller  
helm init --service-account tiller
```

4. Create the Kubernetes registry secret using your credentials for the EDB container registry after replacing registry-username, registry-password.

```
kubectl create secret docker-registry regcred \
--docker-server=containers.enterprisedb.com/edb \
--docker-username=registry-username \
--docker-password=registry-password
```

5. Download the sample-values.yaml and make the required modifications.

<https://get.enterprisedb.com/helm-charts/sample-values.yaml>

In the sample-values.yaml:

- Search for the key 'pullSecret', and replace the value with 'regcred' to match the name of the secret you created in Step 4
- Search for 'nfsServer' and modify the path and ipAddress variables if needed

6. Download the sample helm chart:

<https://get.enterprisedb.com/helm-charts/edb-2.6.0.tgz>

7. Deploy the EDB containers by installing the helm chart:

```
Helminstalledb-2.6.0.tgz-fsample-values.yaml
```

8. Remote access to the database cluster (optional): To access the database cluster remotely (i.e. using pgAdmin or psql), expose the pgpool service:

```
kubectl expose deployment edb1-pgpool \
--port=5444 --target-port=9999 \
--name=edb1-ext-service \
--type=LoadBalancer
```

Wait until external ip address is displayed after running the following command: (takes ~30 secs)

```
kubectl get service edb1-ext-service
```

8. Helpful commands:

- **checking status of the deployed pods**

```
kubectl get pods
```

- **checking error messages to troubleshoot issues**

```
kubectl describe pod pod-name
```

- **list of all the helm charts**

```
helm list
```

- **delete chart (to cleanup and restart)**

```
helm delete chart-name
```

EDB™ Postgres Containers Quick Start

Copyright © 2007 - 2020 EnterpriseDB Corporation. All rights reserved.

EnterpriseDB® Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E [info@enterprisedb.com](mailto:info@enterprisedb.com) [www.enterprisedb.com](http://www.enterprisedb.com)

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.
- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres products where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

## A

Appendix, 16

## C

Conclusion, 18

Creating a Container at the Docker Command Line, 8

## D

Deploying Advanced Server in a Container (Master Node), 9

Deploying Advanced Server in a Container (Standby Node), 11

Deploying BART in a Container, 14

Deploying from the Docker Command Line, 8

Deploying on Google Kubernetes, 5

Deploying on RedHat Openshift, 7

Deploying pgPool in a Container, 13

## E

EDB Containers, 3

## S

Sample GKE Setup, 16

## W

What's New, 2