OpenShift Dedicated Examples - Crunchy Containers for PostgreSQL

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 OpenShift Dedicated Environment

Here are instructions for running examples on an OpenShift 3.4 Dedicated environment. For Dedicated, we have built a set of templates that can be installed into a Dedicated instance to help automate the creation of Crunchy Container Suite containers.

A defined set of templates are supported including:

- master database
- replica database
- full database backup
- database restore
- pgadmin4

Each template is described below.

There are some limitations presented by OpenShift 3.4 in the way in which we can scale up the replica databases. Without support for dynamic volume provisioning and stateful sets, the replica databases can be scaled in a more manual way by creating a new replica database using the provided template. If you want say 3 replica databases, you would use the template 3 times to create the 3 replicas.

When OpenShift supports StatefulSets and Dynamic Provisioning, the replica can be included within a Deployment and/or StatefulSet which will allow for scaling to be done by manipulating the Replica Count within the Deployment or StatefulSet.

2 Installation

Users can install the templates into their OpenShift environment using the following commands.

First, add the following lines to your .bashrc file to set the project paths:

```
export GOPATH=$HOME/cdev
export GOBIN=$GOPATH/bin
export PATH=$PATH:$GOBIN
export CCP_BASEOS=centos7
export CCP_PGVERSION=9.6
export CCP_VERSION=1.5
export CCP_IMAGE_TAG=$CCP_BASEOS-$CCP_PGVERSION-$CCP_VERSION
export CCPROOT=$GOPATH/src/github.com/crunchydata/crunchy-containers
```

You will then need to log out and back in for the changes to your .bashrc file to take effect.

Next, set up a project directory structure and pull down the project:

```
mkdir $HOME/cdev $HOME/cdev/src $HOME/cdev/pkg $HOME/cdev/bin
cd $GOPATH
sudo yum -y install golang git docker postgresql
go get github.com/tools/godep
cd src/github.com
mkdir crunchydata
cd crunchydata
git clone https://github.com/crunchydata/crunchy-containers
cd crunchy-containers
git checkout 1.5
godep restore
```

Finally, you'll change directories to your Dedicated examples repository and create all the templates stored there.

```
cd crunchy-containers/examples/dedicated
./create-all.sh
```

2.1 Example Details

Each example will build a template to be later used by users when they want to deploy a Crunchy container.

The templates are installed by running the following script within each example directory:

```
./run.sh
```

When you run the examples, there are variable substitutions taking place to set the image path and image tags within the OpenShift templates. This substitution allows for better support of different deployments and deployment environments.

You can either use the templates within the OpenShift Web Console using the **Add to Project** functionality or use the **oc** CLI locally to use the templates to deploy databases.

Within each template directory, there is an **example.sh** script that shows how to use the template using the **oc** CLI.

2.2 Deploying Images to OpenShift Registry

You can deploy the Crunchy built container images to the OpenShift registry by running the following script:

```
cd $CCPROOT/examples/dedicated
./push-images.sh
```

You will first need to login to the OpenShift registry to perform this script.

The script will create the appropriate image tag and push the image to the remote registry.

If you are a Crunchy enterprise customer, you will need to change the $TOKEN and $REG values in the push-images.sh script to the ones that correspond to your company's registries along with your personal access token. The necessary token and server information can be obtained either through the web console (console.$YOURCOMPANY.openshift.com) or by visiting your company's API server and requesting a token (https://api.$YOURCOMPANY.openshift.com/oath/token/-request).

As you use the templates, you can specify the images in your templates using the OpenShift registry URL as follows for the **default** OpenShift project:

```
172.30.149.135:5000/default
```

3 Containers Used

3.1 crunchy-postgres container

The crunchy-postgres container executes the Postgres database.

Packages

The container image is built using either the Crunchy Postgres release or the community version based upon a flag in the Makefile.

The Crunchy Postgres RPMs are available to Crunchy customers only. The Crunchy release is meant for customers that require enterprise level support.

The PGDG community RPMs can be used as well by simply commenting out the Crunchy yum repo within the Dockerfiles and uncommenting the PGDG yum repo.

setup.sql

The **setup.sql** script is used to define startup SQL commands that are executed when the database is first created.

Environment Variables

- PG_MODE - either **master**, **slave** or **set**, this value determines whether the database is set up as a master or slave instance, in the case of **set**, it means the container is started within a StatefulSet in a Kubernetes cluster.

- PG_MASTER_USER - the value to use for the user ID created as master. The **master** user has super user privileges.
- PG_MASTER_PASSWORD - the password for the PG_MASTER_USER database user
- PG_USER - the value to use for the user ID created as a normal user. This user is created as part of the setup.sql script upon database creation and allows users to predefine an application user.
- PG_PASSWORD - the password for the PG_USER database user that is created
- PG_DATABASE - a database that is created upon database initialization
- PG_ROOT_PASSWORD - the postgres user password set up upon database initialization
- PG_LOCALE - if set, the locale you want to create the database with, if not set, the default locale is used
- SYNC_SLAVE - if set, this value is used to specify the application_name of a slave that will be used for a synchronous replication
- CHECKSUMS - if set, this value is used to enable the **--data-checksums** option when initdb is executed at initialization, if not set, the default is to **not** enable data checksums
- XLOGDIR - if set, initdb will use the specified directory for WAL
- ARCHIVE_MODE - if set to **on**, will enable continuous WAL archiving by setting the value within the postgresql.conf file **archive_mode** setting, if not set, the default is **off**
- ARCHIVE_TIMEOUT - if set to a number (in seconds) , will specify the postgresql.conf **archive_timeout** setting, if not set, the default value of **60** is used.
- PGAUDIT_ANALYZE - if set, will cause the container to also start the pgaudit_analyze program in the background
- PGDATA_PATH_OVERRIDE - if set, will cause the container to use a /pgdata path name of your choosing rather than the hostname of the container which is the default... this is useful for a master in a deployment.

Features

The following features are supported by the crunchy-postgres container:

- use of OpenShift secrets
- ability to restore from a database backup
- use of custom pg_hba.conf and postgresql.conf files
- ability to override postgresql.conf configuration parameters
- ability to override the default setup.sql script
- ability to set the database locale
- ability to specify a synchronous slave application_name
- ability to specify a recovery using PITR and WAL files, see pitr.adoc for a detailed design explanation of how PITR is implemented within the container suite

Locale Support

Adding locale support to the container is accomplished by running *yum reinstall glibc_common* within the container, this increases the size of the container image and can be removed if you do not require specific locale support.

You can specify the PG_LOCALE env var which is passed to the initdb command when the initial data files are created, for example:

```
"name": "PG_LOCALE",
"value": "fr_BE.UTF-8"
```

By default, no locale is specified when the initdb command is executed.

3.2 crunchy-backup

The crunchy-backup container executes a pg_basebackup against another database container. The backup is a full backup using the standard utility included with postgres, pg_basebackup.

Backup Location

Backups are stored in a mounted backup volume location, using the database host name plus **-backups** as a sub-directory, then followed by a unique backup directory based upon a date/timestamp. It is left to the user to perform database backup archives in this current version of the container. This backup location is referenced when performing a database restore.

Dependencies

The container is meant to be using a NFS or similar network file system to persist database backups.

Environment Variables

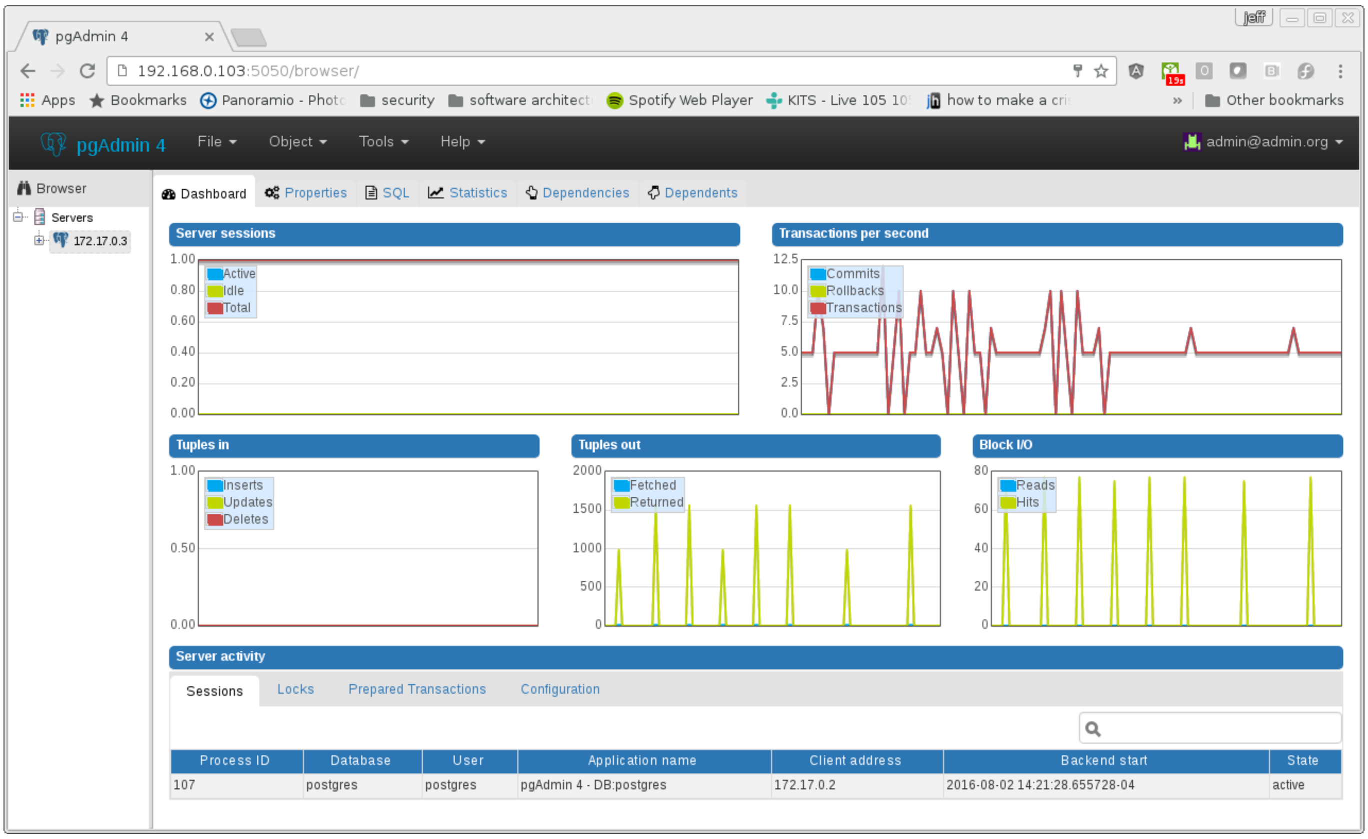
- BACKUP_LABEL - when set, will set the label of the backup, if not set the default label used is **crunchy-backup**
- BACKUP_HOST - required, this is the database we will be doing the backup for
- BACKUP_USER - required, this is the database user we will be doing the backup with
- BACKUP_PASS - required, this is the database password we will be doing the backup with
- BACKUP_PORT - required, this is the database port we will be doing the backup with

3.3 crunchy-pgadmin4

The crunchy-ppgadmin4 container executes the pgadmin4 web application.

The pgadmin4 project is found at the following location: https://www.pgadmin.org/

pgadmin4 provides a web user interface to PostgreSQL databases. A sample screenshot is below:



Process ID	Database	User	Application name	Client address	Backend start	State
107	postgres	postgres	pgAdmin 4 - DB:postgres	172.17.0.2	2016-08-02 14:21:28.655728-04	active

Environment Variables

- None

Features

The following features are supported by the crunchy-pgadmin4 container:

- mount config_local.py and pgadmin4.db to /data volume inside the container to support customization and store the pgadmin4 database file
- expose port 5050 which is the web server port
- a sample pgadmin4 database is provided with an initial administrator user **admin@admin.org** and password of **password**

Restrictions

- None

4 OpenShift Dedicated Template Examples

4.1 Master Database Template

Template Name is **crunchy-master**

Example is found here:

`examples/dedicated/crunchy-master`

This template will create the following:

- database container crunchy-postgres as the master running within a Deployment
- database service for the master

This example deploys a master database configuration which uses a Persistent Volume Claim for persistence.

Table 1: Table Template Parameters

Parameter	Description	Default
NAME	the database service name	example
PGDATA_PATH_OVERRIDE	should match the name of the NAME parameter in most cases	example
PG_MASTER_PORT	the postgres port to use	5432
PG_MASTER_USER	the user name to create and use for a master user	master
PG_MASTER_PASSWORD	the password to use for the master user	password
PG_USER	the user name to create as a normal user	testuser
PG_PASSWORD	the password to use for the normal user	password
PG_DATABASE	the name of the the normal user database which will be created	userdb
PG_ROOT_PASSWORD	the password of the postgres user	password
SYNC_SLAVE	the name of a sync replica that will be allowed to connect if any	
CCP_IMAGE_TAG	the image version to use for the container	rhel7-9.6-1.5

Table 1: (continued)

CCP_IMAGE_PREFIX	the image prefix to use, typically the image stream prefix of your registry	172.30.149.135:5000/default
CCP_IMAGE_NAME	the image name to use, either crunchy-postgres or crunchy-postgres-gis	crunchy-postgres
PVC_NAME	the name to assign to the PVC created for this database typically NAME-pvc	example-pvc
PVC_SIZE	the size of the PVC to create	300M
PVC_ACCESS_MODE	the PVC access mode to use for the created PVC	ReadWriteMany
TEMP_BUFFERS	the postgres temp_buffers configuration setting	9MB
MAX_CONNECTIONS	the postgres max_connections setting	101
SHARED_BUFFERS	the postgres shared_buffers configuration setting	129MB
MAX_WAL_SENDERS	the postgres max_wal_senders configuration setting	7
WORK_MEM	the postgres work-mem configuration setting	5MB

4.2 Database Backup Template

Template Name is **crunchy-backup**

Example is found here:

```
examples/dedicated/crunchy-backup
```

This template will create the following:

- Job which generates a backup container

This example deploys a Job which results in a Pod created which will run the **crunchy-backup** container. It will create a backup of a database and store the backup files in a PVC.

Table 2: Table Template Parameters

Parameter	Description	Default
JOB_NAME	the job name	backupjob
DB_NAME	the service name of the database to backup	master
PVC_NAME	the PVC name to use to store the backup files	backup-pvc
PVC_SIZE	the PVC size to allocate	500M
PVC_ACCESS_MODE	the PVC access mode to use in the creation of the PVC	ReadWriteMany
BACKUP_USER	the postgres user to use when performing the backup	master
BACKUP_PASS	the postgres user password to use when performing the backup	master
CCP_IMAGE_PREFIX	the container image prefix to use, typically the registy IP address and namespace	172.30.149.135:5000/default
CCP_IMAGE_TAG	the container image version to use	rhel7-9.6-1.5

4.3 Restore Database Template

Template Name is **crunchy-restore**

Example is found here:

```
examples/dedicated/crunchy-restore
```

This template will create the following:

- database container crunchy-postgres
- database service

This example performs a database restore using a backup archive found in a PVC.

Table 3: Table Template Parameters

Parameter	Description	Default
NAME	the job name	restoredb
PG_MASTER_PORT	the postgres port to use	5432
PG_MASTER_USER	the user name to create and use for a master user	master
PG_MASTER_PASSWORD	the password to use for the master user	password
PG_USER	the user name to create as a normal user	testuser
PG_PASSWORD	the password to use for the normal user	password
PG_DATABASE	the name of the the normal user database which will be created	userdb
PG_ROOT_PASSWORD	the password of the postgres user	password
PGDATA_PATH_OVERRIDE	the name to overide the pgdata path with typically the NAME value	restoredb
PVC_NAME	the PVC name to use when creating the new PVC typically NAME-pvc	restoredb-pvc
PVC_SIZE	the PVC size to allocate	500M
PVC_ACCESS_MODE	the PVC access mode to use in the creation of the PVC	ReadWriteMany
BACKUP_PATH	the backup archive path to restore from	master7-backups/2017-04-04-09-42-53
BACKUP_PVC	the backup archive PVC to restore from	backup-pvc
CCP_IMAGE_PREFIX	the container image prefix to use, typically the registy IP address and namespace	172.30.149.135:5000/default
CCP_IMAGE_NAME	the container image name to use, must match the image name used in the original db	crunchy-postgres
CCP_IMAGE_TAG	the container image version to use	rhel7-9.6-1.5

4.4 Replica Database Template

Template names is **crunchy-replica**

Example is found here:

```
examples/dedicated/crunchy-replica
```

These templates create the following:

- replica database container crunchy-postgres using Persistent Volume Claim
- service for replica

Table 4: Table Template Parameters

Parameter	Description	Default
SERVICE_NAME	the name to use for the database service	replica
PG_MASTER_HOST	the postgres master service name the replica will connect to	master
PG_MASTER_PORT	the postgres port to use	5432
PG_MASTER_USER	the user name to create and use for a master user	master
PG_MASTER_PASSWORD	the password to use for the master user	password
PVC_NAME	the PVC name to use when creating the new PVC typically NAME-pvc	restoredb-pvc
PVC_SIZE	the PVC size to allocate	500M
PVC_ACCESS_MODE	the PVC access mode to use in the creation of the PVC	ReadWriteMany
CCP_IMAGE_PREFIX	the container image prefix to use, typically the registy IP address and namespace	172.30.149.135:5000/default
CCP_IMAGE_NAME	the container image name to use, must match the image name used in the original db	crunchy-postgres
CCP_IMAGE_TAG	the container image version to use	rhel7-9.6-1.5

4.5 pgadmin4 Web User Interface Template

Template Name is **crunchy-pgadmin4**

Example is found here:

```
examples/dedicated/crunchy-pgadmin4
```

This template will create the following:

- PVC for the pgadmin4 configuration files and database
- pod containing the crunchy-pgadmin4 container
- service for the pgadmin4 container

Table 5: Table Template Parameters

Parameter	Description	Default
NAME	the name to use for the pgadmin4 service	pgadmin4
PVC_NAME	the name to assign to the PVC created for this pgadmin4 typically NAME-pvc	pgadmin4-pvc
PVC_SIZE	the size of the PVC to create	300M
PVC_ACCESS_MODE	the PVC access mode to use for the created PVC	ReadWriteMany
CCP_IMAGE_PREFIX	the container image prefix to use, typically the registy IP address and namespace	172.30.149.135:5000/default
CCP_IMAGE_TAG	the container image version to use	rhel7-9.6-1.5

5 Legal Notices