



# Installation Guide - Crunchy Containers for PostgreSQL

Crunchy Data Solutions, Inc.

Version 1.4.1, 2017-05-10

# Installation Instructions

The crunchy-containers can run on different environments including:

- standalone docker
- openshift enterprise
- kubernetes 1.2.X

In this document we list the basic installation steps required for these environments.

## Project Setup

This instruction assumes you are installing on Centos 7 or RHEL 7.

## Assumptions

The install assumes the following about your host configuration:

- centos7 (or RHEL7) VM
- user called someuser has been created
- someuser has sudo privileges with no password

## Step 1 - Project Directory Structure

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=1.4.1
export CCP_IMAGE_TAG=$CCP_BASEOS-$CCP_PGVERSION-$CCP_VERSION
export CCROOT=$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 kubernetes
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 1.4.1
godep restore
```

If you are a Crunchy enterprise customer, you will place the **CRUNCHY** repo key and yum repo file into the `$CCPROOT/conf` directory at this point. These files can be obtained through <https://access.crunchydata.com/> on the downloads page.

## Step 2 - Install the Host Dependencies

Next, install system dependencies:

```
sudo yum -y update
sudo groupadd docker
sudo usermod -a -G docker someuser
```

Remember to log out of the **someuser** account for the docker group to be added to your current session. Adding **docker** group to your user account allows you to run docker commands from your user account.

```
su - someuser
```

You can ensure your **someuser** account is added to the **docker** group by running the following command and ensuring **docker** appears as one of the results:

```
groups
```

Before you start docker, you might consider configuring docker storage: This is described if you run:

```
man docker-storage-setup
```

Follow the instructions available [on the main Openshift documentation page](#) to configure docker storage appropriately.

Next, we enable and start up docker:

```
sudo systemctl enable docker.service
sudo systemctl start docker.service
```

## Step 3 - Build the Containers

At this point, you have a decision to make - either download prebuilt containers from dockerhub, **or** build the containers on your local host.

To download the prebuilt containers, make sure you can login to dockerhub.com, and then run the following:

```
docker login
cd $CCPROOT
./bin/pull-from-dockerhub.sh
```

Or if you'd rather build the containers from source, perform a container build as follows:

```
cd $CCPROOT
make setup
make all
```

After this, you will have all the Crunchy containers built and are ready for use in a **standalone docker** environment.

To install an Openshift environment, continue on to Step 4.

## Step 4 - Openshift Installation

See the OSE installation guide for details on how to install OSE on your host. The main instructions are here:

[https://docs.openshift.com/enterprise/3.2/install\\_config/install/index.html](https://docs.openshift.com/enterprise/3.2/install_config/install/index.html)

**Or**, if you'd prefer to install Openshift Origin, the easiest way to get Openshift Origin up and running is found here: [https://github.com/openshift/origin/blob/master/docs/cluster\\_up\\_down.md](https://github.com/openshift/origin/blob/master/docs/cluster_up_down.md)

For examples and tips on how to run Openshift Enterprise & Origin, please look at the openshift.asciidoc documentation.

## Step 5 - Configure NFS for Persistence Examples

NFS is required for some of the Openshift examples, those dealing with backups and restores will require a working NFS for example.

First, if you are running your NFS system with SELinux in enforcing mode, you will need to run the

following command to allow NFS write permissions:

```
sudo setsebool -P virt_use_nfs 1
```

Next, you will need to set the permissions of your NFS path so that your pods can have write access. For the Crunchy examples, the **nfsnobody** GUI was chosen as an example. Pods will reference the **nfsnobody** GID (65534) as a security context **supplementalGroup** attribute. This setting will allow the pod to have group permissions of 65534 and therefore be able to write to the NFS persistent volumes.

The permissions on the NFS path are set as follows:

```
drwxrwx---.  3 nfsnobody nfsnobody  23 Dec 16 11:28 nfsfileshare
```

Most of the Crunchy containers run as the postgres UID (26), but you will notice that when **supplementalGroups** are specified, the pod will include the nfsnobody group in the list of groups for the pod user.

The case of Amazon file systems is different, for that you use the **fsGroup** security context setting but the idea for allowing write permissions is the same.

Here are the instructions I use when setting up NFS:

<http://www.itzgeek.com/how-tos/linux/centos-how-tos/how-to-setup-nfs-server-on-centos-7-rhel-7-fedora-22.html>

Examples of Openshift NFS can be found here:

<https://github.com/openshift/origin/tree/master/examples/wordpress/nfs>

The examples specify a test NFS server running at IP address 192.168.0.103

On that server, the /etc/exports file looks like this:

```
/nfsfileshare *(rw,sync)
```

Test your NFS configuration out by mounting a local directory:

```
mount 192.168.0.114:/nfsfileshare /mnt/nfsfileshare
```

if you are running your client on a VM, you will need to add 'insecure' to the exportfs file on the NFS server, this is because of the way port translation is done between the VM host and the VM instance.

see this for more details:

<http://serverfault.com/questions/107546/mount-nfs-access-denied-by-server-while-mounting>

# Legal Notices

Copyright © 2017 Crunchy Data Solutions, Inc.

CRUNCHY DATA SOLUTIONS, INC. PROVIDES THIS GUIDE "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Crunchy, Crunchy Data Solutions, Inc. and the Crunchy Hippo Logo are trademarks of Crunchy Data Solutions, Inc.