

# Openshift Demo

Jeff.McCormick@crunchydata.com

## Install Openshift

1. Install openshift from github releases page, binary install:
  - a. <https://github.com/openshift/origin/releases>
2. Download, untar....
3. Set /etc/resolv.conf to match local host for Openshift DNS server
4. Make sure docker is running
5. Configure Docker group for dev user
6. Turn off firewall to allow access to Openshift ports (development env only)
7. `sudo ./openshift start`
8. `sudo chmod +rw admin.kubeconfig`
9. `oc login`
10. `oc new-project pgproject`
11. `oc edit scc restricted --config=./openshift.local.config/master/admin.kubeconfig`

## crunchy-pg (v1.1.0)

A simple postgresql container that will run in various configurations within Openshift.

<https://github.com/CrunchyData/crunchy-postgres-container-94>

## Dockerfile overview

1. either centos 7 or rhel 7 can be the container image base
2. installs postgresql RPMs from the PGDG repo, 9.4.5 now, 9.5 soon, Crunchy-PG RPMs build is available with support
3. includes postgis and pgrouting
4. features a volume for overriding postgres config files with local files
5. features env vars to override certain postgres tuning parameters
6. features a volume for using local host file system for postgres data files (max I/O)
7. Runs as postgres user, requires postgres user (uid/gid) to be defined on local host for setting file ownership
8. Bash script used to initialize and start postgres daemon

## Standalone Example

standalone.json - runs a single postgres container (pod + service):

```
oc process -f standalone.json | oc create -f -  
  
psql -h pg-standalone.pgproject.svc.cluster.local -U testuser  
userdb
```

The password is generated by Openshift 'process' and can be found using:  
oc describe pod pg-standalone | grep PASSWORD

ping is blocked by openshift

## Master Slave Example

master-slave-rc.json - creates a single master container and a single standby/slave container for a simple replication scenario, the slave is read-only

This example creates the following Openshift objects:

- replication controllers
- master pod
- master service
- slave pod
- slave service

oc process -f master-slave-rc.json | oc create -f -

```
oc get pods  
oc get services  
oc get rc  
oc get dc
```

scaling up can be performed

```
oc scale rc pg-slave-rc-1 --replicas=2
```

You can see the generated master password as follows:

```
oc describe pod pg-master | grep MASTER
```

You can access the databases as follows:

```
psql -h pg-master.pgproject.svc.cluster.local -U master userdb  
psql -h pg-slave.pgproject.svc.cluster.local -U master userdb
```

Demonstrate with inserts/selects and `pg_stat_replication` that replication is working.

Demonstrate with golang client example that slave is behind a round-robin proxy.

Demonstrate web console.

Demonstrate where data is stored for emptyDir.

## Other examples

- NFS for file system instead of EmptyDir
- storing passwords as Openshift secrets
- Openshift 'RunAsUser' setting
- Passing environment variables to containers for tuning/configuration

More details on how Openshift 'secrets' can be used:

<https://blog.openshift.com/openshift-ecosystem-crunchy-postgresql-integration/>

## Crunchy Postgresql Manager

A Docker-based Postgresql-as-a-Service project that Crunchy is developing:

<https://github.com/CrunchyData/crunchy-postgresql-manager>