RED HAT®
CONSULTING

redhat.

Oracle Cartridge Installation Guide

PREPARED FOR - FRIT

endif::[]l

Table of Contents

1. Synopsis

The purpose of this document is to present the guidance to install and configure an Oracle OpenShift Enterprise V2.2 cartridge. It is meant to be loaded into OpenShift from source code.

The cartridge currently supports the following features:

- Creating an Oracle Database tenant instance via a call to a remote script
- Setting environment variables to allow web servers to call newly created tenant in a database

2. Installation

2.1. Setup OSE Environment

The setup of the OSE Environment can be accomplished as per your usual way of deploying broker and nodes. This could be via the OSE install script, or any other CM tools like Puppet and Ansible

2.2. Oracle Provisioning Script Requirements

This cartridge is in and of itself not responsible for configuring a remote tenant, instead that responsibility lies to a script that resides on a remote host that will be called by the install script when the gear is instantiated. Script is called via a remote sshpass call as the **OPENSHIFT_ORACLE_DB_SCRIPT_USER**. The script accepts username, password, and SID. It will then return:

SUCCESS/FAIL@@HOST@@PORT@@TENNANT_ID

2.3. Service Account Access Requirements

The service account used to remotely login to the host with the Oracle provisioning script needs to be able to be to **sudo** without a password to the **OPENSHIFT_ORACLE_DB_SCRIPT_USER**. This will require changes to the /etc/sudoers file on the **OPENSHIFT_ORACLE_DB_SCRIPT_HOST** The following shows an example configuration that will allow the **serviceaccountuser** user to call the **/u02/app/oracle/frit/bin/frit_dba_cdb.pl** script with any arguments (the *, wildcard flag) from the *test.hosts.example.com* host without a password prompt (**NOPASSWD**).

```
User_Alias      OSEORAGRP = serviceaccountuser
Host_Alias      OSEORAGRP_HOSTS = test.hosts.example.com
Cmnd_Alias      OSEORAGRP_CMNDS = /u02/app/oracle/frit/bin/frit_dba_cdb.pl*
OSEORAGRP       OSEORAGRP_HOSTS = (oracle) NOPASSWD: OSEORAGRP_CMNDS
```

3. Cartridge Installation

The cartridge can be installed as any other OSE cartridge.

3.1. Cartridge Extraction

Extract the zipped source code of the Oracle cartridge under

`/usr/libexec/openshift/cartridges`

You will also need to set the correct SELinux Context on the cartridge so that it is consistent with the rest of the cartridges on each node. This file context is:

`system_u:object_r:bin_t:s0`

To set this context run the following command:

```
chcon -R -u system_u /usr/libexec/openshift/cartridges/ose2-oracle-frb-cart-frb-was*/
```

3.2. Cartridge Environment Variables

You now need to set the environment variables on each Node. Please note that the script does expect the password variable value to be a base64 hash of the plaintext password. Modify as needed:

```
echo "oraclescripthost.example.com" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_HOST
echo "serviceaccountuser" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_HOST_SERVICE_ACCOUNT
echo "oracle" > /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_USER
echo "/OracleProvisioningScript.sh" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_LOC
echo "@@" > /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_DELIMINATOR
echo "/usr/libexec/openshift/cartridges/ose2-oracle-frb-cart/id_rsa" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE
echo "/usr/libexec/openshift/cartridges/ose2-oracle-frb-cart/id_rsa.pub" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PUBLIC
```

OPENSHIFT_ORACLE_DB_SCRIPT_HOST	This is the hostname that the remote Oracle configuration script resides on.
OPENSHIFT_ORACLE_DB_SCRIPT_HOST_SERVICE_ACCOUNT	This is the service account user name that will be used for the ssh call to the **OPENSHIFT_ORACLE_DB_SCRIPT_HOST**.
OPENSHIFT_ORACLE_DB_SCRIPT_USER	This is the user that the **OPENSHIFT_ORACLE_DB_SCRIPT_HOST_SERVICE_ACCOUNT** will sudo to to run the **OPENSHIFT_ORACLE_DB_SCRIPT_LOC**.

OPENSHIFT_ORACLE_DB_SCRIPT_LOC	This is the location on **OPENSHIFT_ORACLE_DB_SCRIPT_HOST** where the remote configuration script resides.
OPENSHIFT_ORACLE_DB_SCRIPT_DELIMINATOR	This is the deliminator used in the return value coming from the remote configuration script. This should be set to '@@'
OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE	This points to the location on the filesystem where the private ssh key that will be used to call the script on **OPENSHIFT_ORACLE_DB_SCRIPT_HOST** will reside.
OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PUBLIC	This points to the location on the filesystem where the public ssh key that will be used to call the script on **OPENSHIFT_ORACLE_DB_SCRIPT_HOST** will reside.

Optional Variable:

If on the deletion of the Oracle Database Datasource Gear you want the back end tenant PDB to remain in place, set the following optional variable, otherwise the tenant PDB will be deleted on gear teardown by a call to the OPENSHIFT_ORACLE_DB_SCRIPT_LOC.

```
echo "true" >
/etc/openshift/env/OPENSHIFT_ORACLE_DB_PRESERVE_ON_TEARDOWN
```

Once the environment variables have been set, run the following to apply the appropriate file permissions:

```
chmod 644 /etc/openshift/env/OPENSHIFT_ORACLE_DB_*
```

3.3. SSH Key Exchange

Now use the following command to create the needed SSH keys and exchange them with **OPENSHIFT_ORACLE_DB_SCRIPT_HOST**. You will be asked twice to enter in the password for the **OPENSHIFT_ORACLE_DB_SCRIPT_HOST_SERVICE_ACCOUNT@OPENSHIFT_ORACLE_DB_SCRIPT_HOST**.

```
rm -rf $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE)
ssh-keygen -t rsa -f $(cat
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE) -N "" -q
ssh $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_USER)@$(cat
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_HOST) 'mkdir -p .ssh; chmod 700 .ssh'
cat $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PUBLIC) | ssh $(cat
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_USER)@$(cat
/etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_HOST) 'cat >> .ssh/authorized_keys;
chmod 600 .ssh/authorized_keys'
chmod 755 $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE)
chmod 755 $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PUBLIC)
```

3.4. Verifying environment variables and SSH Key Exchange

Now that the environment variables have been set and the SSH key exchange completed, it behooves the administrator to test the settings before proceeding further. Run the following command and confirm that no SSH, SSL, or user permissions errors are thrown.

---

ssh -i $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SSH_IDENTITY_PRIVATE) -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIP_HOST_SERVICE_ACCOUNT)@$(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_HOST) "echo \"$(echo $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIP_HOST_SERVICE_ACCOUNT_ENC_PASSWORD) | base64 -d)\" | sudo -u oracle -S $(cat /etc/openshift/env/OPENSHIFT_ORACLE_DB_SCRIPT_LOC) -op get_pdb_list" ""

3.5. Cartridge OpenShift Registry Installation

Execute the following commands to add the cartridge to the node:

```
cd /usr/libexec/openshift/cartridges
oo-admin-cartridge --action install --recursive --source
/usr/libexec/openshift/cartridges
```

If you are updating the cartridge, make sure to remove the previously installed cartridge from the broker's internal registry by running the following on the broker:

```
oo-admin-ctl-cartridge -c deactivate --name frb-oracle-12.0
oo-admin-ctl-cartridge -c delete --name frb-oracle-12.0
```

To make the cartridge available run this command from the broker:

```
oo-admin-ctl-cartridge --activate -c import-node node.hostname
oo-admin-broker-cache --clear && oo-admin-console-cache --clear
```

4. Reference Information

OpenShift V2

- Cartridge Developers Guide
- How to expose more than one public port in cartridge