Red Hat
OpenShift Container Platform

User Experiences in Setting Up OpenShift on z/VM

Samuel D. Cohen
Levi, Ray & Shoup, Inc.
sam.cohen@lrs.com
(217) 862-9227

What I was trying to do

- Understand Openshift better
  - Learn more about the technology and terminology
- See what it takes to setup a standalone demo environment under z/VM
- Try to understand the Redhat-provided documentation
- Provide “lessons learned” for others that want to try this at home
- I can’t answer what you would do with this, other than impress your Kubernetes-centric colleagues
- I also can’t answer what you would do after installation other than setup multipathing

Credits

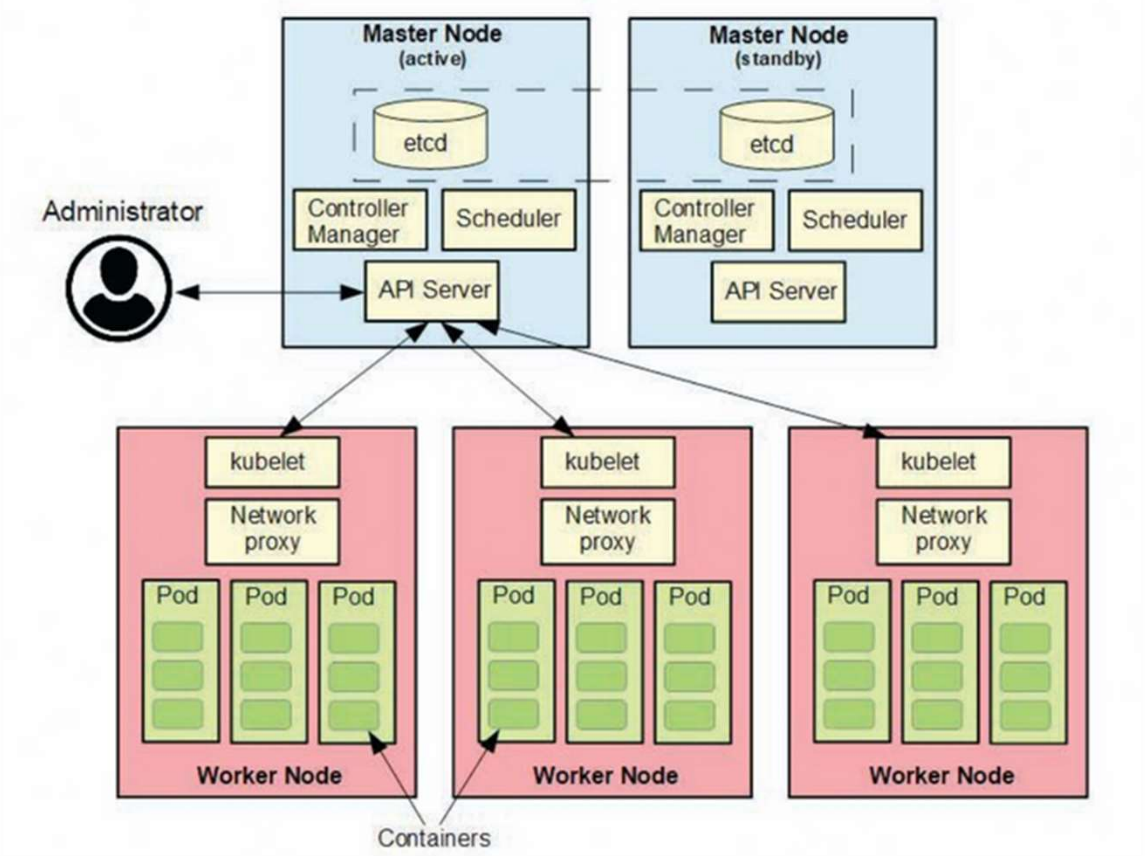
- S. Michael Benson, *Docker Swarm or Kubernetes?*, Enterprise Tech Journal, 2019 Issue 6.
- Filipe Miranda, IBM, published articles in Linkedin on his experience and provided sample settings for DNS, Load Balancer. Also provided 1:1 assistance
  - https://www.linkedin.com/pulse/red-hat-openshift-installation-process-experiences-ibm-filipe-miranda/
  - https://www.linkedin.com/pulse/understanding-network-definitions-from-openshift-4-ibm-filipe-miranda/ describes network settings used internally by Openshift

Understand OpenShift Better

- Redhat uses Kubernetes as a deployment and management engine for “pods”
- Terminology
  - Container: Think of it like a stand-alone application with static links
  - Pod: a business application, made up of one or more containers
  - Service: multiple identical pods distributed throughout the cluster for load balancing and higher availability
  - Node: a virtual machine (in our case) where multiple pods can be deployed
  - Master Node: schedules and controls pods across multiple worker nodes
  - Worker Node: where pods are actually executing
  - Kubelet: an agent in each worker node for communication with the master nodes

Reference: S. Michael Benson, *Docker Swarm or Kubernetes?*, Enterprise Tech Journal, 2019 Issue 6. Used with permission

Better shown in a picture



Reference: S. Michael Benson, *Docker Swarm or Kubernetes?*, Enterprise Tech Journal, 2019 Issue 6. Used with permission

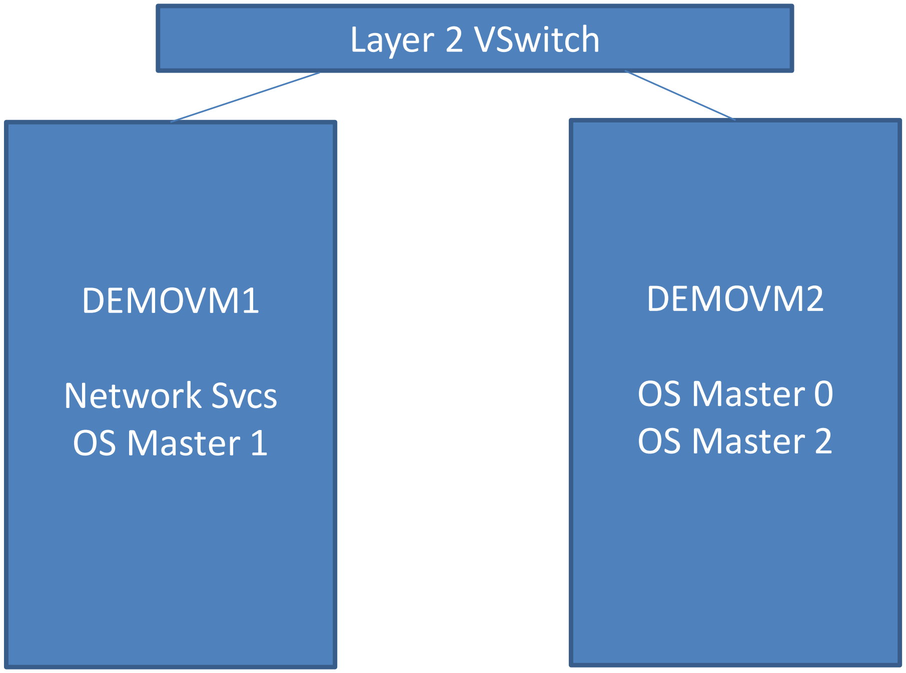
Deployment process

- Master nodes and worker nodes use CoreOS as the operating system
  - CoreOS is lighter weight than RHEL (even in minimal installation)
- A “boot” node is created which will create the master/worker nodes
  - Node is initially installed like RHEL via VM Reader with boot, parmfile and initrd
  - After code is laid down from a base image, CoreOS uses an “ignition file” to customize the base image
  - Ignition files are created using an Openshift-provided tool based on customer-provided instructions
    - Instructions written in YAML (recursive acronym: YAML Ain’t Markup Language)
    - Similar to XML in concept
    - Positional input, very picky

For this Proof of Concept

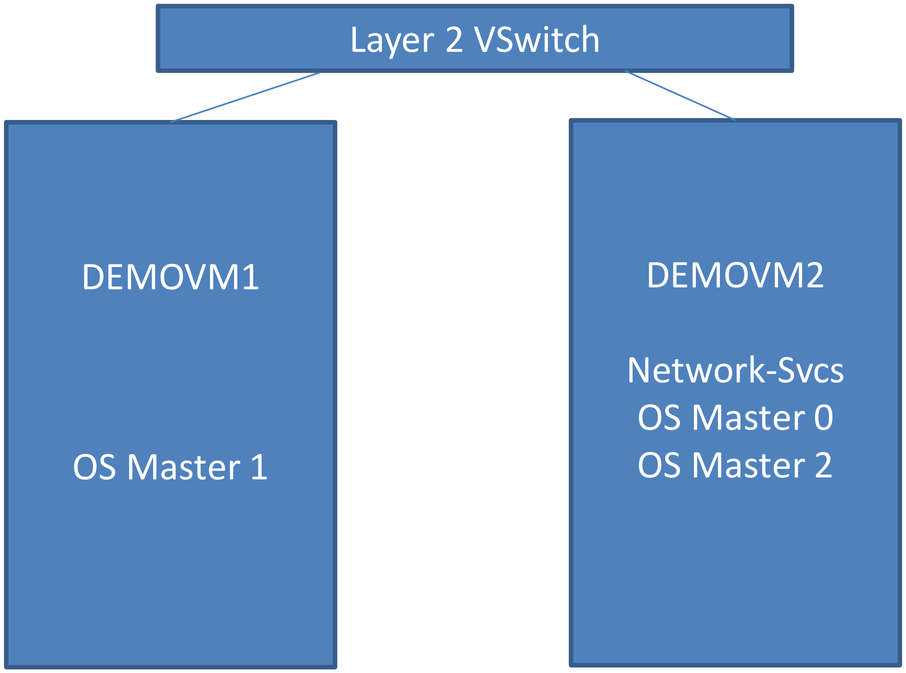
- I defined only 3 master nodes
  - Client Nodes are recommended for production systems
  - If you plan to show this environment to someone, also allocate at least 2 client nodes
- I split master nodes between 2 different SSI cluster members
- I used FB disks attached via 3 FCP channels
  - NPIV is active so subchannel traffic only goes to the appropriate VM guest
  - I added multipath connections for both SSI cluster members in each node
- I created a single RHEL 8 guest to provide front-end interfaces:
  - dns server for routing by name to Openshift components
  - http server for reading ignition scripts
  - ftp server for reading installation images
  - haproxy server for sending incoming messages to multiple nodes

Openshift Demo Environment (Initial)



Management network:	10.96.64.192/28
DNS Domain:	osdemo.lrsinc.org
Kubernetes “service” network:	internal
DNS Domain:	democluster.osdemo.lrsinc.org

Openshift Demo Environment (Bonus #1)



Layer 2 VSwitch

DEMOVM1

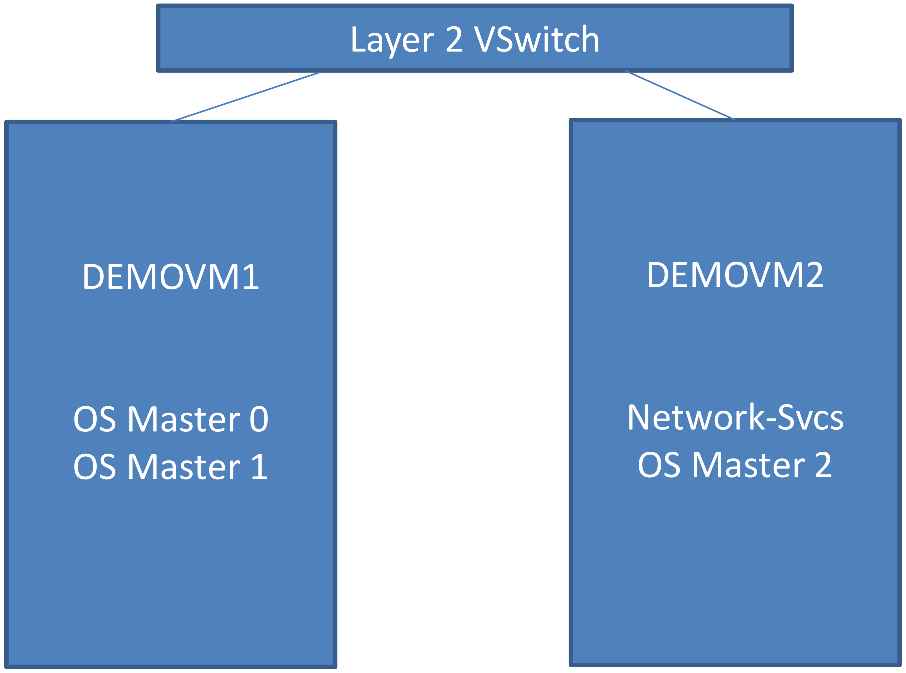
OS Master 1

DEMOVM2

Network-Svcs
OS Master 0
OS Master 2

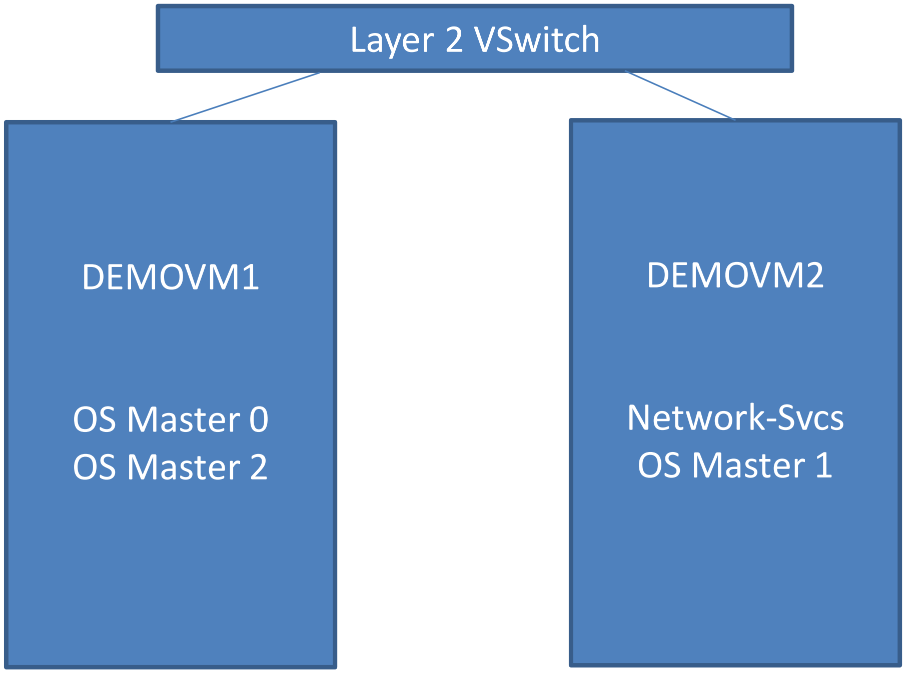
Management network:	10.96.64.192/28
DNS Domain:	osdemo.lrsinc.org
Kubernetes “service” network:	internal
DNS Domain:	democluster.osdemo.lrsinc.org

Openshift Demo Environment (Bonus #2)



Management network:	10.96.64.192/28
DNS Domain:	osdemo.lrsinc.org
Kubernetes “service” network:	internal
DNS Domain:	democluster.osdemo.lrsinc.org

Openshift Demo Environment (Bonus #3)



Layer 2 VSwitch

DEMOVM1

OS Master 0
OS Master 2

DEMOVM2

Network-Svcs
OS Master 1

Management network:	10.96.64.192/28
DNS Domain:	osdemo.lrsinc.org
Kubernetes “service” network:	internal
DNS Domain:	democluster.osdemo.lrsinc.org

Getting Started

- Prepare front-end virtual machine
- Define OpenShift Master virtual machines
  - Each has 4 Virtual IFLs, 21G of Memory, VNIC to a Layer 2 Vswitch
  - Worker virtual machines are similar, but “only” 16G of Memory
- Register with Redhat so you can download the code
- Download Openshift (https://try.openshift.com)
  - (https://cloud.redhat.com/openshift/install/ibmz/user-provisioned)
  - Also download a file called “pull-secret”, generated by your userid/password
- Print documentation:
  - https://docs.openshift.com/container-platform/latest/installing/installing_ibm_z/installing-ibm-z.html

Layer 2 Vswitch (from SYSTEM CONFIG)

```
Rdevice  0200-022F EQID ETH00200 Type OSA
Define VSwitch OSVSW    Rdev 0223.P00 Ethernet VLAN Unaware
```

Directory Profiles for Openshift Virtual Machines

```
PROFILE OSMASTR
CLASS GL
STORAGE 21G
MAXSTORAGE 24G
ACCOUNT OPNSHIFT
ACIGROUP OPNSHIFT
COMMAND SET VCONFIG MODE LINUX
COMMAND DEFINE CPU 00 TYPE IFL
COMMAND DEFINE CPU 01 TYPE IFL
COMMAND DEFINE CPU 02 TYPE IFL
COMMAND DEFINE CPU 03 TYPE IFL
COMMAND DEFINE STORAGE INITIAL STANDBY REMAINDER
CRYPTO    APVIRT
IPL CMS
IUCV ALLOW
IUCV *IDENT RESANY GLOBAL
MACHINE ESA 4
OPTION TODENABLE APPLMON
CONSOLE 0009 3215 T OPERATOR
NICDEF 1000 TYPE QDIO LAN SYSTEM OSVSW
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
(CMS Minidisk Links)
```

```
PROFILE OSWRKR
CLASS GL
STORAGE 16G
MAXSTORAGE 20G
ACCOUNT OPNSHIFT
ACIGROUP OPNSHIFT
COMMAND SET VCONFIG MODE LINUX
COMMAND DEFINE CPU 00 TYPE IFL
COMMAND DEFINE CPU 01 TYPE IFL
COMMAND DEFINE CPU 02 TYPE IFL
COMMAND DEFINE CPU 03 TYPE IFL
COMMAND DEFINE STORAGE INITIAL STANDBY REMAINDER
CRYPTO    APVIRT
IPL CMS
IUCV ALLOW
IUCV *IDENT RESANY GLOBAL
MACHINE ESA 4
OPTION TODENABLE APPLMON
CONSOLE 0009 3215 T OPERATOR
NICDEF 1000 TYPE QDIO LAN SYSTEM OSVSW
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
(CMS Minidisk Links)
```

Four Types of Virtual Machines Defined

- Front-End Services

  USER OSDHCPD LBYONLY

  - IP Address = 10.96.64.193

    INCLUDE OSMASTR

    MACHINE ESA 1

    DEFINE STORAGE 2G

    (attach FCP channels)

- Bootstrap Node

  USER OSBOOT LBYONLY

  - IP Address = 10.96.64.194

    INCLUDE OSMASTR

    (attach FCP channels)

- Master Node(s)

  USER OSMASTR[0-2] LBYONLY

  - IP Addresses 10.96.64.195-197

    INCLUDE OSMASTR

    (attach FCP channels)

- Worker Node(s)

  USER OSWRKR[0-1] LBYONLY

  - IP Addresses 10.96.64.198-99

    INCLUDE OSWRKR

    (attach FCP channels)

1. Define DNS/DHCP/Load Balancer/FTP Server

- Built based on SLES15 SP3
  - Easier to setup and configure via YaST than RedHat
  - Enabled DNS with 1 new zone:
    - osdemo.lrsinc.org (10.96.64.192/26)
      - Using existing network 10.96.64.0/24 with gateway 10.96.64.254
  - Add required DNS entries to appropriate zone
  - Using nginx for load balancer
- Built alternate server based on RHEL 8.5
  - Same DNS config as on SLES
  - Using haproxy instead of nginx
  - Didn’t configure DHCP server
- Download Openshift code
  - Include openshift-install tar file
  - Include pull-secret
  - Make available via anonymous FTP via bind-mount
- Define user “core” with password “corepass”
  - Defined ssh key for this user for copying “public” key to ignition file
  - Command used:

```
ssh-keygen -t rsa -b 4096 -N ‘ ’  -f /home/core/.ssh/id_rsa
```

2. Create installation ignition files

- Untar openshell-install in the FTP server
  - Executable and readme files
- Verify the pull-secret file
  - Go to a website like *jsonlint.com* to verify the JSON in this file.
  - Verify you aren’t missing any delimiters or curly braces (easily missed in a cut-and-paste)
- Create an “install-config.yaml” file
  - Position sensitive
  - Ensure “architecture: s390x” under computer and controlPlane tags
  - Imbed the pull-secret file
  - Imbed the public key from /home/core/.ssh/id_rsa.pub
  - Note: the YAML file will disappear after it is used to create the manifests and ignition files
    - Openshell-install program uses the term “consumed”
    - Make a copy so you can repeat the process more easily
  - Don’t change the cluster network and service network parameters

2. Create installation ignition files (cont’d)

```
apiVersion: v1
baseDomain: osdemo.lrsinc.org
compute:
- architecture: s390x
  hyperthreading: Enabled
  name: worker
  replicas: 0
controlPlane:
  architecture: s390x
  hyperthreading: Enabled
  name: master
  replicas: 3
metadata:
  name: democluster
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  networkType: OpenShiftSDN
  serviceNetwork:
  - 172.30.0.0/16
platform:
  none: {}
fips: false
pullSecret: ‘contents of pull-secret file’
sshkey: ‘contents of /home/core/.ssh/id_rsa.pub’
```

2. Create installation ignition files (cont'd)

- Execute `openshift-install create manifests --dir=(`***`install_directory`***`)`
- If defining worker guests, change manifests/cluster-scheduler-02-config.yml file
  - Set **mastersSchedulable** to False
- Execute `openshift-install create ignition-configs --dir=(`***`install_directory`***`)`
  - 3 ignition files created: bootstrap.ign, master.ign, worker.ign
- Enable access to files via anonymous ftp
  - enable world-readable for created files
    - `chmod +r *.ign`
    - `chmod +r -R auth/`
  - mount /mnt *mount-point* --bind
    - /srv/ftp in SLES
    - /var/ftp in RHEL

Note: security certificates used in this process expire after 24 hours

- Don't get too distracted

3. Build “bootstrap” machine

- Determine proper parameters vs. “ignition” file
  - Parm file used to define hardware (FCP) and network, points to ignition file
  - Ignition file used to customize virtual machine
  - Review the doc, not everything you need is in one place
- Parmfile that worked:

```
rd.neednet=1 console=ttysclp0 !condef zfcp.allow_lun_scan=0
rd.znet=qeth,0.0.1000,0.0.1001,0.0.1002,layer2=1,portno=0
ip=10.96.64.194::10.96.64.254:255.255.255.192:osboot.osdemo.lrsinc.org::none
nameserver=10.96.64.193
coreos.inst.ignition_url=http://10.96.64.193:81/mnt/bootstrap.ign
coreos.live.rootfs_url=http://10.96.64.193:81/mnt/rhcos-live-rootfs.s390x.img
coreos.inst.install_dev=sda
rd.zfcp=0.0.2100,0x5005076812161BCB,0x0000000000000000
rd.zfcp=0.0.2100,0x5005076812161BCC,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCB,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCC,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCB,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCC,0x0000000000000000
```

- Use of a separate CONFIG file to hold parameters in RHEL doesn’t work with RHCOS

4. Build “master” machines

- Create a unique Parmfile for each machine:

```
rd.neednet=1 console=ttysclp0 coreos.inst=yes zfcp.allow_lun_scan=0
rd.znet=qeth,0.0.1000,0.0.1001,0.0.1002,layer2=1,portno=0
ip=10.96.64.195::10.96.64.254:255.255.255.192:master0.osdemo.lrsinc.org::none
nameserver=10.96.64.193
coreos.inst.ignition_url=http://10.96.64.193:81/mnt/master.ign
coreos.live.rootfs_url=http://10.96.64.193:81/mnt/rhcos-live-rootfs.s390x.img
coreos.inst.install_dev=sda
rd.zfcp=0.0.2100,0x5005076812161BCB,0x0000000000000000
rd.zfcp=0.0.2100,0x5005076812161BCC,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCB,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCC,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCB,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCC,0x0000000000000000
```

- IP address is different for each master, other parameters stay the same
- Real FCP subchannels are unique for each v.m. but are defined with the same virtual subchannels for ease of definition

5a. Build “worker” machines (optional)

- Create a unique Parmfile for each machine:

```
rd.neednet=1 console=ttysclp0 coreos.inst=yes zfcp.allow_lun_scan=0
rd.znet=qeth,0.0.1000,0.0.1001,0.0.1002,layer2=1,portno=0
ip=10.96.64.198::10.96.64.254:255.255.255.192:worker0.osdemo.lrsinc.org::none
nameserver=10.96.64.193
coreos.inst.ignition_url=http://10.96.64.193:81/mnt/worker.ign
coreos.live.rootfs_url=http://10.96.64.193:81/mnt/rhcos-live-rootfs.s390x.img
coreos.inst.install_dev=sda
rd.zfcp=0.0.2100,0x5005076812161BCB,0x0000000000000000
rd.zfcp=0.0.2100,0x5005076812161BCC,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCB,0x0000000000000000
rd.zfcp=0.0.2200,0x5005076812171BCC,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCB,0x0000000000000000
rd.zfcp=0.0.2300,0x5005076812181BCC,0x0000000000000000
```

- IP address is different for each worker, other parameters stay the same
- Real FCP subchannels are unique for each v.m. but are defined with the same virtual subchannels for ease of definition

5b. Add worker machines to the cluster (4.6+)

- New for OpenShift 4.6, the worker machines are not automatically added to the cluster
- Must issue commands to add them to the cluster
  - **oc get csr**
    - Note csr name associated with worker node
  - **oc set csr *csr-name* approve**
  - Wait a few minutes, then issue **oc get nodes** and see if the worker nodes are up and running

6. Enable multipathing (4.8+)

- New for OpenShift 4.9, if you are using FB disks and predefined the paths during creation, you must enable the multipath daemon.
  - Updating /etc/multipath.conf doesn’t work
  - Creating/updating /etc/zfcp.conf doesn’t work
  - You must tell openshift to do it for you
- Build a YAML file named 99-master-kargs-mpath.yaml

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: "master"
  name: 99-master-kargs-mpath
spec:
  kernelArguments:
    - 'rd.multipath=default'
    - 'root=/dev/disk/by-label/dm-mpath-root’
```

- Worker nodes have a slightly different YAML file
  - Replace “master” in the YAML above with “worker”
- Implement the update
  - oc create -f ./99-master-kargs-mpath.yaml

Progress:

```
./openshift-install create manifests --dir=/mnt
time="2023-06-16T12:59:46-05:00" level=info msg="Consuming Install Config from target directory"
time="2023-06-16T12:59:46-05:00" level=warning msg="Making control-plane schedulable by setting MastersSchedulable to true for Scheduler cluster settings"
time="2023-06-16T12:59:47-05:00" level=info msg="Manifests created in: /mnt/manifests and /mnt/openshift"
./openshift-install create ignition-configs --dir=/mnt
time="2023-06-16T13:00:10-05:00" level=info msg="Consuming Openshift Manifests from target directory"
time="2023-06-16T13:00:10-05:00" level=info msg="Consuming Worker Machines from target directory"
time="2023-06-16T13:00:10-05:00" level=info msg="Consuming Common Manifests from target directory"
time="2023-06-16T13:00:10-05:00" level=info msg="Consuming Master Machines from target directory"
time="2023-06-16T13:00:10-05:00" level=info msg="Consuming OpenShift Install (Manifests) from target directory"
time="2023-06-16T13:00:10-05:00" level=info msg="Ignition-Configs created in: /mnt and /mnt/auth"
chmod +r -R auth/
chmod +r rhcos-live*
(start OSBOOT, OSMASTR[0-2])
./openshift-install wait-for bootstrap-complete --dir=/mnt
time="2023-06-16T13:17:41-05:00" level=info msg="Waiting up to 20m0s (until 1:37PM) for the Kubernetes API at https://api.democluster.osdemo.lrsinc.org:6443..."
time="2023-06-16T13:17:41-05:00" level=info msg="API v1.26.5+7a891f0 up"
time="2023-06-16T13:17:41-05:00" level=info msg="Waiting up to 30m0s (until 1:47PM) for bootstrapping to complete..."
time="2023-06-16T13:29:21-05:00" level=info msg="It is now safe to remove the bootstrap resources"
time="2023-06-16T13:29:21-05:00" level=info msg="Time elapsed: 11m41s"
(Update haproxy.cfg and restart haproxy service)
(Shutdown OSBOOT)
./openshift-install wait-for install-complete --dir=/mnt
time="2023-06-16T13:30:58-05:00" level=info msg="Waiting up to 40m0s (until 2:10PM) for the cluster at https://api.democluster.osdemo.lrsinc.org:6443 to initialize..."
time="2023-06-16T13:38:19-05:00" level=info msg="Checking to see if there is a route at openshift-console/console..."
time="2023-06-16T13:38:19-05:00" level=info msg="Install complete!"
time="2023-06-16T13:38:19-05:00" level=info msg="To access the cluster as the system:admin user when using 'oc', run 'export KUBECONFIG=/mnt/auth/kubeconfig'"
time="2023-06-16T13:38:19-05:00" level=info msg="Access the OpenShift web-console here: https://console-openshift-console.apps.democluster.osdemo.lrsinc.org"
time="2023-06-16T13:38:19-05:00" level=info msg="Login to the console with user: \"kubeadmin\", and password: \"RGWUR-FqvDx-bkhIy-RVHoo\""
time="2023-06-16T13:38:19-05:00" level=info msg="Time elapsed: 7m21s"
```

Progress:

Now check on status with the ***oc*** command:

```
[root@osdhcpd mnt]# export KUBECONFIG=/mnt/auth/kubeconfig

[root@osdhcpd mnt]# ./oc status
In project default on server https://api.democluster.osdemo.lrsinc.org:6443

svc/openshift - kubernetes.default.svc.cluster.local
svc/kubernetes - 172.30.0.1:443 -> 6443

View details with 'oc describe <resource>/<name>' or list resources with 'oc get all'.

[root@osdhcpd mnt]# ./oc get all
NAME               TYPE         CLUSTER-IP   EXTERNAL-IP                          PORT(S)   AGE
service/kubernetes   ClusterIP      172.30.0.1   <none>                               443/TCP   3h17m
service/openshift    ExternalName   <none>       kubernetes.default.svc.cluster.local   <none>    3h8m

[root@osdhcpd mnt]# oc whoami
system:admin

[root@osdhcpd mnt]# ./oc get nodes
NAME                      STATUS   ROLES                          AGE     VERSION
master0.osdemo.lrsinc.org   Ready    control-plane,master,worker   3h16m   v1.26.5+7a891f0
master1.osdemo.lrsinc.org   Ready    control-plane,master,worker   3h12m   v1.26.5+7a891f0
master2.osdemo.lrsinc.org   Ready    control-plane,master,worker   3h15m   v1.26.5+7a891f0
```

Progress:

```
[root@osdhcpd mnt]# watch -n5 oc get clusteroperators

Every 5.0s: oc get clusteroperators
osdhcpd.osdemo.lrsinc.org: Fri Jun 16 16:31:48 2023

NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
authentication                             4.13.3    True        False         False      124m
baremetal                                  4.13.3    True        False         False      3h9m
cloud-controller-manager                   4.13.3    True        False         False      3h17m
cloud-credential                           4.13.3    True        False         False      3h19m
cluster-autoscaler                         4.13.3    True        False         False      3h9m
config-operator                            4.13.3    True        False         False      3h11m
console                                    4.13.3    True        False         False      131m
control-plane-machine-set                  4.13.3    True        False         False      3h9m
csi-snapshot-controller                    4.13.3    True        False         False      3h10m
dns                                        4.13.3    True        False         False      3h10m
etcd                                       4.13.3    True        False         False      3h8m
image-registry                             4.13.3    True        False         False      3h1m
ingress                                    4.13.3    True        False         False      3h3m
insights                                   4.13.3    True        False         False      3h5m
kube-apiserver                             4.13.3    True        False         False      176m
kube-controller-manager                    4.13.3    True        False         False      3h7m
kube-scheduler                             4.13.3    True        False         False      3h6m
kube-storage-version-migrator              4.13.3    True        False         False      131m
machine-api                                4.13.3    True        False         False      3h10m
machine-approver                           4.13.3    True        False         False      3h9m
machine-config                             4.13.3    True        False         False      3h6m
marketplace                                4.13.3    True        False         False      3h10m
monitoring                                 4.13.3    True        False         False      179m
network                                    4.13.3    True        False         False      3h10m
node-tuning                                4.13.3    True        False         False      129m
openshift-apiserver                        4.13.3    True        False         False      124m
openshift-controller-manager               4.13.3    True        False         False      3h4m
openshift-samples                          4.13.3    True        False         False      3h3m
operator-lifecycle-manager                 4.13.3    True        False         False      3h9m
operator-lifecycle-manager-catalog         4.13.3    True        False         False      3h10m
operator-lifecycle-manager-packageserver   4.13.3    True        False         False      3h4m
service-ca                                 4.13.3    True        False         False      3h11m
storage                                    4.13.3    True        False         False      3h11m
```

Progress:

- Enable multipathing by running the YAML created earlier

```
[root@osdhcpd mnt]#./oc create -f ./99-master-kargs-mpath.yaml
machineconfig.machineconfiguration.openshift.io/99-master-kargs-mpath created
```

- Verify multipathing was dispatched

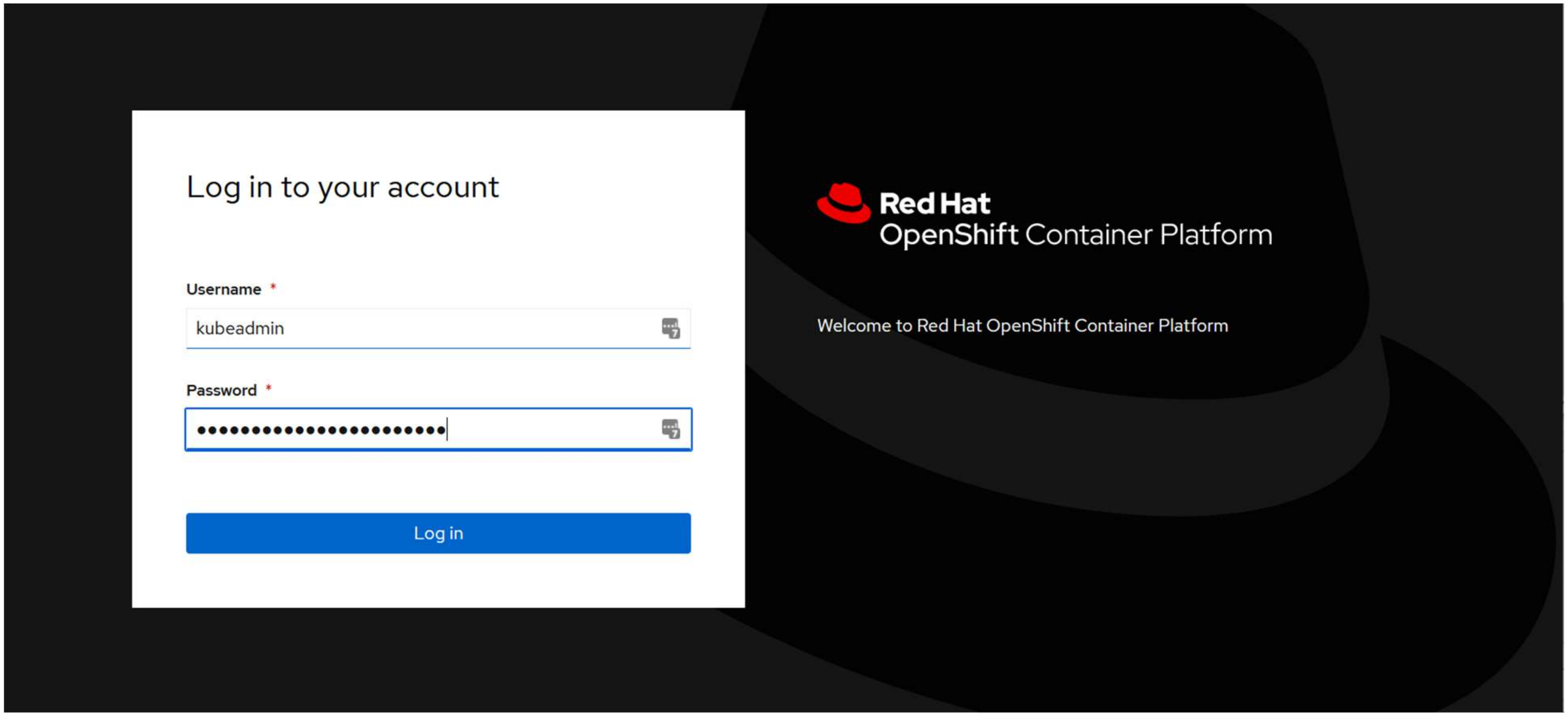
```
[root@osdhcpd mnt]#./oc get MachineConfig
NAME                                               GENERATEDBYCONTROLLER                      IGNITIONVERSION   AGE
00-master                                          14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
00-worker                                          14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
01-master-container-runtime                        14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
01-master-kubelet                                  14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
01-worker-container-runtime                        14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
01-worker-kubelet                                  14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
99-master-generated-registries                     14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
99-master-kargs-mpath                                                                                           98s
99-master-ssh                                                                                 3.2.0             50m
99-worker-generated-registries                     14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
99-worker-ssh                                                                                 3.2.0             50m
rendered-master-29a3cf34678fbeacd78e485aaddf6621   14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
rendered-master-d7882b903adcc79bf0c2886249662fb4   14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             93s
rendered-worker-cc42cfdf6f0fc9d7cde5aed9d8d6af10   14a1ca2cb91ff7e0faf9146b21ba12cd6c652d22   3.2.0             43m
```

- Log into each node to run multipath command and verify multipathing

```
su - core
eval "$(ssh-agent -s)"
ssh-add /home/core/.ssh/id_rsa
ssh osmastr0
        sudo multipath –l
        exit
ssh osmastr1
        sudo multipath -l
        exit
ssh osmastr1
        sudo multipath -l
        exit
```

Progress:

- Finally, open a browser, get the login screen and sign on:



Log in to your account

Username *

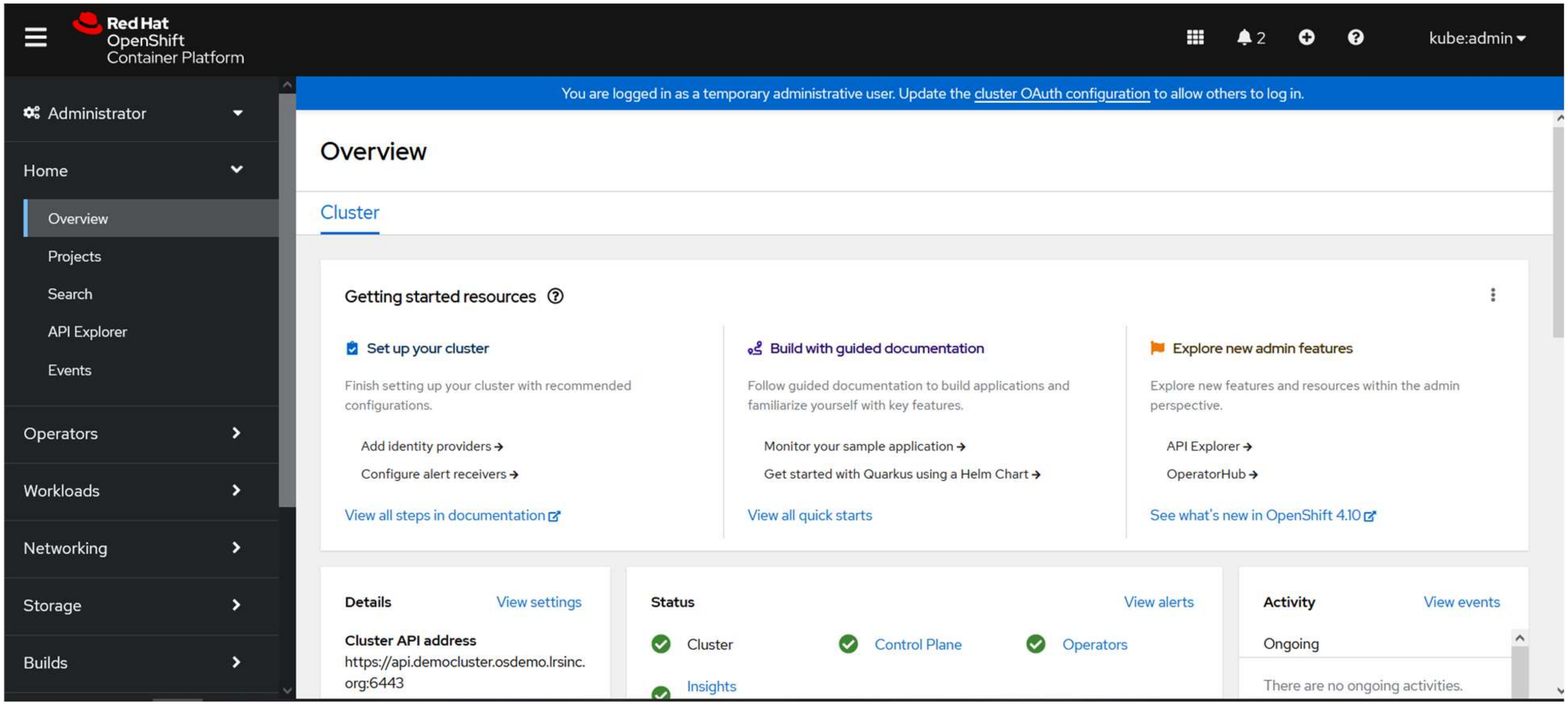
kubeadmin

Password *

••••••••••••••••••••••••

Welcome to Red Hat OpenShift Container Platform

Progress:



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Overview

Cluster

Getting started resources

Set up your cluster

Finish setting up your cluster with recommended configurations.

- Add identity providers →
- Configure alert receivers →

View all steps in documentation

Build with guided documentation

Follow guided documentation to build applications and familiarize yourself with key features.

- Monitor your sample application →
- Get started with Quarkus using a Helm Chart →

View all quick starts

Explore new admin features

Explore new features and resources within the admin perspective.

- API Explorer →
- OperatorHub →

See what's new in OpenShift 4.10

Details View settings

Cluster API address
https://api.democluster.osdemo.lrsinc.org:6443

Status View alerts

✓ Cluster ✓ Control Plane ✓ Operators
✓ Insights

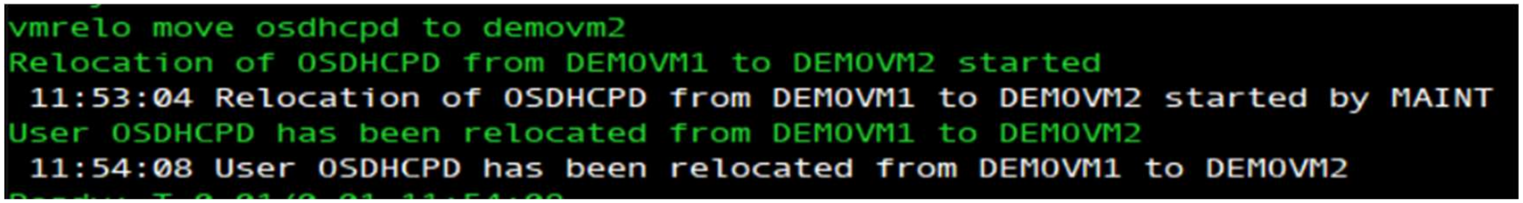
Activity View events

Ongoing

There are no ongoing activities.

Bonus Actions

- VMRELO of front-end (OSBOOT) was uneventful



```
vmrelo move osdhcpd to demovm2
Relocation of OSDHCPD from DEMOVM1 to DEMOVM2 started
 11:53:04 Relocation of OSDHCPD from DEMOVM1 to DEMOVM2 started by MAINT
User OSDHCPD has been relocated from DEMOVM1 to DEMOVM2
 11:54:08 User OSDHCPD has been relocated from DEMOVM1 to DEMOVM2
```

- VMRELO of master nodes was more interesting
  - VMRELO completed, but path switching took a while
    - 2 paths were defined for each VM LPAR during initial config
    - Lost 8 ping packets after relocation completed before ping resumed successfully
    - multipath –l eventually came back with both paths
    - ./oc get nodes also took a while to come back, as did ssh to the relocated node

Lessons Learned

- Couldn’t login as “core” after bootstrap machine created
  - Had to start ssh-agent from user “core” and load that agent every time I rebooted the DDNS/DHCP server:

    eval "$(ssh-agent -s)“

    ssh-add *path/filename of private key*
  - Connected to bootstrap machine and watched progress via command journalctl -b -f -u bootkube.service
    - Started to build master and worker machines once bootkube service settled down
    - My clue was to look for “SELinux: mount invalid” messages on the boot server after reboot
- To handle anonymous ftp, bind mount /mnt to /var/ftp on RHEL or /srv/ftp on SLES
- Had to change *.ign files to permissions 644 (added world-readable for anonymous ftp) for ignition files to be read during image creation
- Had to allow world-readable auth subdirectory to let userid “core” get the credentials

Lessons Learned

- CNAMEs didn’t always work
  - Gave explicit assignment to master[0-2], worker[0-1], CNAME to actual v.m. name
- Using SCSI-disk means that IND USER won’t show I/O counts
  - Harder to tell if you’re stuck; had to use SCIF to monitor along with watching bootkube.service (which goes in fits and starts)
- SLES vs RHEL for DNS/Load balancer/Load source
  - haproxy (RHEL) vs. nginx (SLES), nginx didn’t like to simply listen on ports 80/443
  - YaST DNS-Server dialog doesn’t like “*” in the dns config file; had to manually edit the osdemo.lrsinc.org file
  - Ended up staying with RHEL due to issues with nginx for http/https
    - Probably could have gotten it to work with more knowledge of nginx

Results

- Cluster was successfully built
- Front-end machine could be relocated between SSI members
- RHOCP Masters could be started in either SSI member and relocated
  - Several OpenShift processes were unhappy after the virtual machine moved from one LPAR to another
  - Needs more research by someone more familiar with RHOCP internals and processes
- Graphical interface worked (once I pointed to the new DNS in my Windows network settings)
- Someone else gets to figure out what to do with it

Questions?

Additional File Settings

/var/named/dynamic/osdemo.lrsinc.org

```
$TTL 2d
@                IN SOA          osdhcpd.osdemo.lrsinc.org.      root.osdhcpd.osdemo.lrsinc.org. (
                                 2020030101      ; serial
                                 3h              ; refresh
                                 1h              ; retry
                                 1w              ; expiry
                                 1d )            ; minimum
osdemo.lrsinc.org.         IN NS   osdhcpd.osdemo.lrsinc.org.
$ORIGIN osdemo.lrsinc.org.
helper                 IN A     10.96.64.193
osdhcpd                IN A     10.96.64.193
bootstrap              IN A     10.96.64.194
master0                IN A     10.96.64.195
master1                IN A     10.96.64.196
master2                IN A     10.96.64.197
worker0                IN A     10.96.64.198
worker1                IN A     10.96.64.199
helper.democluster     IN CNAME helper.osdemo.lrsinc.org.
api.democluster        IN CNAME osdhcpd.osdemo.lrsinc.org.
api-int.democluster    IN CNAME osdhcpd.osdemo.lrsinc.org.
*.apps.democluster     IN CNAME osdhcpd.osdemo.lrsinc.org.
osboot                 IN CNAME bootstrap.osdemo.lrsinc.org.
osmastr0               IN CNAME master0.osdemo.lrsinc.org.
osmastr1               IN CNAME master1.osdemo.lrsinc.org.
osmastr2               IN CNAME master2.osdemo.lrsinc.org.
oswrkr0                IN CNAME worker0.osdemo.lrsinc.org.
oswrkr1                IN CNAME worker1.osdemo.lrsinc.org.
```

Additional File Settings

/var/named/dynamic/64.96.10.in-addr.arpa

```
$TTL 2d
@               IN SOA          osdhcpd.osdemo.lrsinc.org.      root.osdhcpd.osdemo.lrsinc.org. (
                                2020041400      ; serial
                                3h              ; refresh
                                1h              ; retry
                                1w              ; expiry
                                1d )            ; minimum

64.96.10.in-addr.arpa.  IN NS           osdhcpd.osdemo.lrsinc.org.
$ORIGIN 64.96.10.in-addr.arpa.
193             IN PTR          osdhcpd.osdemo.lrsinc.org.
194             IN PTR          bootstrap.osdemo.lrsinc.org.
195             IN PTR          master0.osdemo.lrsinc.org.
196             IN PTR          master1.osdemo.lrsinc.org.
197             IN PTR          master2.osdemo.lrsinc.org.
198             IN PTR          worker0.osdemo.lrsinc.org.
199             IN PTR          worker1.osdemo.lrsinc.org.
```

Additional File Settings

/etc/vsftpd/vsftpd.conf

```
anonymous_enable=YES
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
ftpd_banner=Welcome to osdhcpd FTP service
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=NO
```

Additional File Settings

/etc/haproxy/haproxy.conf

```
global
 log         127.0.0.1 local2
 chroot      /var/lib/haproxy
 pidfile     /var/run/haproxy.pid
 maxconn     4000
 user        haproxy
 group       haproxy
 daemon
 stats socket /var/lib/haproxy/stats
 ssl-default-bind-ciphers PROFILE=SYSTEM
 ssl-default-server-ciphers PROFILE=SYSTEM
defaults
    mode                    http
    log                     global
    option                  tcplog
    option                  dontlognull
    option                  redispatch
    retries                 3
    timeout http-request    10s
    timeout queue           1m
    timeout connect         10s
    timeout client          1m
    timeout server          1m
    timeout http-keep-alive 10s
    timeout check           10s
    maxconn                 3000
```

```
listen ingress-http
    bind *:80
    mode tcp
    server master0 10.96.64.195:80 check inter 1s
    server master1 10.96.64.196:80 check inter 1s
    server master2 10.96.64.197:80 check inter 1s
listen ingress-https
    bind *:443
    mode tcp
    server master0 10.96.64.195:80 check inter 1s
    server master1 10.96.64.196:80 check inter 1s
    server master2 10.96.64.197:80 check inter 1s
listen api
    bind *:6443
    mode tcp
#   server bootstrap 10.96.64.194:6443 check
    server master0 10.96.64.195:6443 check
    server master1 10.96.64.196:6443 check
    server master2 10.96.64.197:6443 check

listen api-int
    bind *:22623
    mode tcp
#   server bootstrap 10.96.64.194:22623 check
    server master0 10.96.64.195:22623 check
    server master1 10.96.64.196:22623 check
    server master2 10.96.64.197:22623 check
```