Standard Operating Procedure

*New Node Addition to OpenShift Cluster*

## PURPOSE

The purpose of this standard operating procedure (SOP) is to provide a detailed step-by-step procedure to addition of new nodes.

## PROCEDURE

Prechecks (all must pass before proceeding):

1. Verify Vault access by ssh-ing to all new nodes.
2. Try to ssh from the master node to the new node to check that the passwdless ssh works.

ssh <node hostname>

* 1. If ssh does not work, copy id\_rsa.pub file from the master node and put it in the new node

cd .ssh from the master node

cat id\_rsa.pub to copy the key

cd .ssh and vi authorized\_keys on the new node and paste the key

chomd 640 authorized\_keys to modify read/write permissions if necessary

1. For AWS nodes, check EFS entry (Also in Checking and Updating EFS Entry for AWS Nodes SOP):
   1. Go to master01 node and run the command below as svc-aws-ose user:

sudo su – svc-aws-ose

for i in $appnodes; do ssh $i "uname -n; sudo grep efs /etc/dnsmasq.conf"; done

i.e.

[svc-aws-ose@master01-prod-red ~]$ for i in $appnodes; do ssh $i "uname -n; sudo grep efs /etc/dnsmasq.conf"; done

node01-prod-red.aws-use1.cloud.marriott.com

server=/efs.us-east-1.amazonaws.com/169.254.169.253

node02-prod-red.aws-use1.cloud.marriott.com

server=/efs.us-east-1.amazonaws.com/169.254.169.253

node03-prod-red.aws-use1.cloud.marriott.com

server=/efs.us-east-1.amazonaws.com/169.254.169.253

…

node40-prod-red.aws-use1.cloud.marriott.com

server=/efs.us-east-1.amazonaws.com/169.254.169.253

node41-prod-red.aws-use1.cloud.marriott.com

node42-prod-red.aws-use1.cloud.marriott.com

node43-prod-red.aws-use1.cloud.marriott.com

node44-prod-red.aws-use1.cloud.marriott.com

node45-prod-red.aws-use1.cloud.marriott.com

node46-prod-red.aws-use1.cloud.marriott.com

node47-prod-red.aws-use1.cloud.marriott.com

node48-prod-red.aws-use1.cloud.marriott.com

node49-prod-red.aws-use1.cloud.marriott.com

node50-prod-red.aws-use1.cloud.marriott.com 

Here, nodes 41-50’s /etc/dnsmasq.conf are not updated to use EFS.

* 1. Go into each node and update /etc/dnsmasq.conf by adding

server=/efs.us-east-1.amazonaws.com/169.254.169.253

in the file:

ssh <node name>

vi /etc/dnsmasq.conf, add server=/efs.us-east-1.amazonaws.com/169.254.169.253, and save and exit

* 1. Restart dnsmasq.service:

sudo systemctl restart dnsmasq.service

* 1. After updating all nodes, run the command from Step 1 in the master01 node to validate that all nodes are updated.

1. Login to new node and check the NTP and DNS on nodes and should be matching with already added hosts.

chronyc activity

shows all online servers.

chronyc sources

Shows what source servers are.

Ping app hostname and check the app node is resolving to correct IP or not.

ping <node hostname>

nslookup <node hostname>

And check if the IP address displayed matches with the app node’s IP address

1. Check /etc/sysctl.conf for fs.inotify.max\_user\_instances = 8192
2. Confirm atomic RPM version is at least **atomic-openshift-3.3.1.46.45-1**

rpm -qa | grep atomic

1. Check for leap status on all nodes. It should be normal.

chronyc tracking

Leap status should be normal.

* 1. If Leap status is not normal, run the commands below to sync.

chronyc sync

systemctl status chronyd.service

check this is active.

systemctl is-enabled chronyd

Check if the returned value is “enabled”.

If above steps all passed, do the following steps node by node:

1. Ping the master URL and the IP address to confirm that it is reachable.

ping <master hostname>

1. Login to master node and see all routes.

oc get routes --all-namespaces

Grab one of the routes with marriott.com and ping it from the new node

ping <route>

to see if the route is reachable from the new node.

1. Run docker-storage-setup (Also in Docker Storage Setup SOP):
   1. Stop Docker Service

systemctl stop docker

* 1. To see if a process has been stopped

systemctl status docker

* 1. Check docker volume

lvdisplay  
vgdisplay

* 1. Delete docker volume

vgremove docker-vg

* 1. Check the volumes

lsblk

Check which volume has docker-vg:

[svc-aws-ose@node01-dev2-red ~]$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

nvme0n1 259:4 0 25G 0 disk

├─nvme0n1p1 259:3 0 1M 0 part

└─nvme0n1p2 259:6 0 25G 0 part /

nvme1n1 259:0 0 40G 0 disk

└─log\_vg-log\_lv 253:0 0 40G 0 lvm /var/log

nvme2n1 259:1 0 100G 0 disk

└─origin\_vg-origin\_lv 253:1 0 100G 0 lvm /var/lib/o

nvme3n1 259:2 0 100G 0 disk

└─nvme3n1p1 259:5 0 100G 0 part

├─docker--vg-docker--pool\_tmeta 253:2 0 104M 0 lvm

│ └─docker--vg-docker--pool 253:4 0 39.8G 0 lvm

* 1. Delete the respective docker volume

fdisk /dev/<docker volume>

i.e.

fdisk /dev/nvme3n1

To delete:  
d

To save:  
w

* 1. Validate the partition deletion

lsblk

To save:

partprobe

* 1. Perform docker partition clean-up

wipefs -a /dev/<docker volume>

i.e.

wipefs –a /dev/nvme3n1

* 1. Clean-up docker folders

rm -rf /var/lib/docker/\*

* 1. Remove /etc/sysconfig/docker-storage

rm /etc/sysconfig/docker-storage

* 1. Run docker storage setup

docker-storage-setup

* 1. Check the partition

lsblk

* 1. Check if docker-vg shows up

vgs

* 1. Start docker

systemctl start docker

* 1. Check docker status  
     systemctl status docker

1. Disable swap on each VM.

To check the status:

free -m

swapon –sc

To disable:

swapoff <device name>

**comment** /dev/mapper/rhel-swap **out** from /etc/fstab

1. From the node, check the user svc-vxby-ose (change *vxby* based on location) is available and check the passwdless | sudo access for that user from master
   1. Every environment but Perf:

Id svc-vxby-ose;

grep -i svc-vxby-ose /etc/passwd

grep -i svc-vxby-ose /etc/sudoers

* 1. Perf:

id root;

grep -i root /etc/passwd

grep -i root /etc/sudoers

1. Check for DTR certificates from root.

[root@node51-devtest-vxbyr ~]# ll /etc/pki/ca-trust/source/anchors/

total 4

-rw-r--r--. 1 root root 2102 Oct 9 11:09 dtr-556876752.us-east-1.elb.amazonaws.com.crt

Check if DTR cron job is working.

sh /home/svc-vxby-ose/OpenShift/scripts/CheckDTRAccess.ksh

1. Create /var/log/OpenShift for crons to work under a particular node.

The scale up should happen at once for all set of new nodes. Repeat all previous steps until all nodes passed prechecks and are ready for scale up.

1. Login to Master server and add [new\_nodes] section under inventory file /etc/ansible/hosts.

Populate [new\_nodes] section with new nodes with appropriate node labels.

oc get nodes --show-labels

vi /etc/ansible/hosts

[new\_nodes]

nodeX.example.com openshift\_node\_labels="{'region': 'primary', 'zone': 'XXXX}"

1. Run ansible playbook to scale up the cluster. Must be in svc-vx\* user (except Perf2)

screen (Do yum install screen if it is not installed)

ansible-playbook -i /etc/ansible/hosts /usr/share/ansible/openshift-ansible/playbooks/byo/openshift-node/scaleup.yml

This can take up to 30 minutes.

1. In .bashrc of master node populate APP\_NODE with new node name

vi .bashrc

source .bashrc

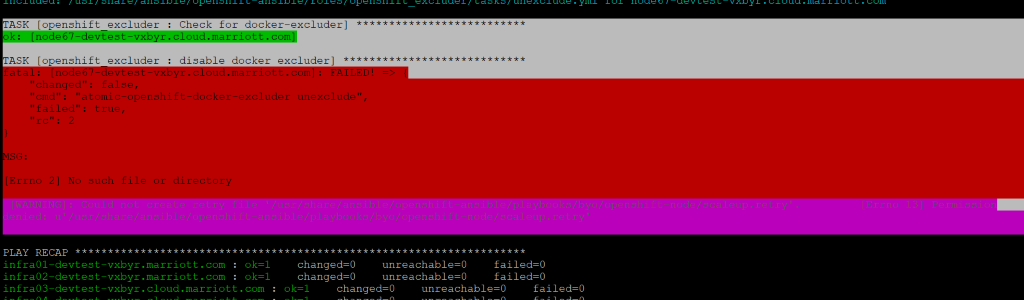
1. Login with root and check if the $appnodes are updated from the master node.

echo $appnodes

(Skip steps 19-23 if excluder error does not exist)

If there is excluder error while scaling up (Also in Excluder Error While Scaling Up SOP):

1. Check if the error below is seen:



1. Execute this command:

atomic-openshift-docker-excluder unexcluded

1. Put flags below:

openshift\_enable\_docker\_excluder=false

openshift\_enable\_openshift\_excluder=false

1. If there is a package error, rebuild rpm package:
2. vi /etc/yum.conf
3. Comment out exclude=

#exclude=

1. Remove rpm

rm -f /var/lib/rpm/\_\_db\*

/usr/lib/rpm/rpmdb\_verify /var/lib/rpm/Packages

1. yum clean all

1. Proceed with scale up. When it finishes, comment out excluder flags (step 21) and uncomment exclude section in /etc/yum.conf (steps 22a and 22b) and rebuild rpm db (Below is also in RPM Rebuild SOP):
2. Rebuild RPM

rpm --rebuilddb

1. Cleanup everything

yum clean all

Postchecks:

1. Log into an existing node to copy all cron tabs and put the cron tabs in /var/spool/cron/root. Look at the cron tab to find the path.

crontab -l

* 1. Every environment but Perf:

scp -r \* svc-vxby-ose@<node name>:/home/svc-vxby-ose

* 1. Perf:

scp -r \* root@<node name>:/root/OpenShift/ose-perf2-vxby<r/b>

cd /var/spool/cron

vi root

1. Check the crontab from new nodes and update if it is not,

crontab -l

1. Validate /etc/sysconfig/docker registry from new nodes.

grep -i ADD\_REGISTRY /etc/sysconfig/docker

1. Move the entry from [new-nodes] to [nodes] in /etc/ansible/hosts file from master server. Comment out [new-nodes] at the beginning and end of the hosts file.
2. Delete pods (that are not deploy) that have crashed or have errors to regenerate them in the new node.

oadm manage-node <node hostname> --list-pods

oc get pods --all-namespaces | grep -v Running

oc delete pod <pod name> -n <project name>

oc get pods --all-namespaces | grep <pod name>

oc get pods --all-namespaces | grep Crash to get all CrashedLoopBackOff pods

## VALIDATION

Follow the validation steps in “Procedure”.

Validate the new servers using oc get nodes commands.

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Effective Date | Reviewed by | Approved by |
| 1.0 | 10/1/2018 | KyungIn Kim | Yashi Kumar |
|  |  |  |  |

Document Modification History & Revision Log

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Version | Date Modified | Revised by | Role | Section Affected | Reference | Remarks |
| 1.0 |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |