**1.)** Környezet ellenőrzés

oshift-master:

* publicIP (eth0): 140.86.55.167
* privateIP (eth1): 10.1.1.101 —> oshift-master-priv
* user: centos

oshift-compute1:

* publicIP (eth0): 140.86.55.77
* privateIP (eth1): 10.1.1.11 —> oshift-compute1-priv
* user: centos

oshift-compute2:

* publicIP (eth0): 140.86.49.95
* privateIP (eth1): 10.1.1.12 —> oshift-compute2-priv
* user: centos

oshift-xe:

* publicIP (eth0): 140.86.55.172
* privateIP (eth1): 10.1.1.201 —> oshift-xe-priv
* user: centos

**2.)**

Telepítéshez szükséges csomagok telepítése

sudo yum install wget git net-tools bind-utils iptables-services bridge-utils bash-completion

Csomagok frissességének ellenőrzése

(mindegyik gépen)

sudo yum update

Epel Repository telepítése

(mindegyik gépen)

sudo yum -y install \

<https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm>

Az Epel repository alapértelmezett használatának tiltása.

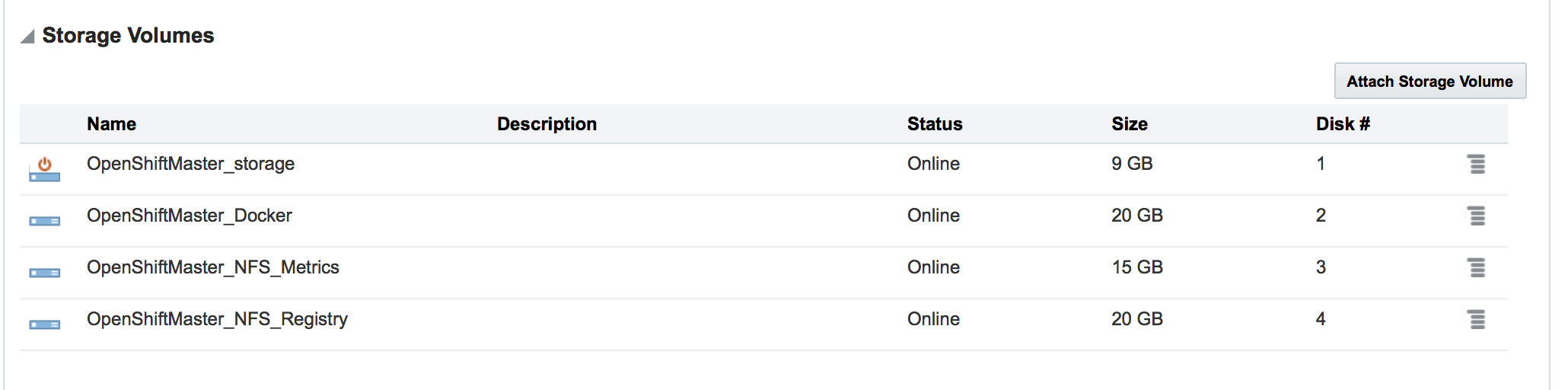
(mindegyik gépen)

sudo sed -i -e "s/^enabled=1/enabled=0/" /etc/yum.repos.d/epel.repo

**3.)**

A Master gép dedikált diszkekkel rendelkezik az alábbi funkciókra:

* docker
* registry
* metrics



sudo fdisk -l

Minden diszket particionálni kell és beállítani “Linux LVM” típusra

pl.:

sudo fdisk /dev/xvdc

az alábbi billentyűvel új partíciólétrehozása: **n**

… itt enter… enter… az alapértelmezett válaszok megfelelőek

az alábbi billentyűvel típus definiálása: **t**

Linux LVM definiálása: **8e**

a partíció ellenőrzése: **p**

kilépés és mentés: **w**

A maradék kettő diszken ugyan ennek a megismétlése

Ha nem elérhető a “vgcreate” parancs, akkor:

sudo yum install lvm2

VG létrehozása a Registry és Metrics részére

sudo vgcreate vg\_registry /dev/xvdc1

sudo vgcreate vg\_nfs\_metric /dev/xvdd1

sudo vgcreate vg\_nfs\_registry /dev/xvde1

LV létrehozása (csak az NFS VG-k részére)

sudo lvcreate -n lv\_metrics -l 100%VG vg\_nfs\_metric

sudo lvcreate -n lv\_registry -l 100%VG vg\_nfs\_registry

FileSystem létrehozása (csak az NFS LV-k részére)

sudo mkfs.xfs /dev/vg\_nfs\_registry/lv\_registry

sudo mkfs.xfs /dev/vg\_nfs\_metric/lv\_metrics

Mapp könyvtárak létrehozása

sudo mkdir /nfs

sudo mkdir -p /nfs/registry/registry

sudo mkdir -p /nfs/metrics/metrics

sudo chmod -R 777 /nfs

fstab szerkesztése az automatikus mount miatt

sudo vi /etc/fstab

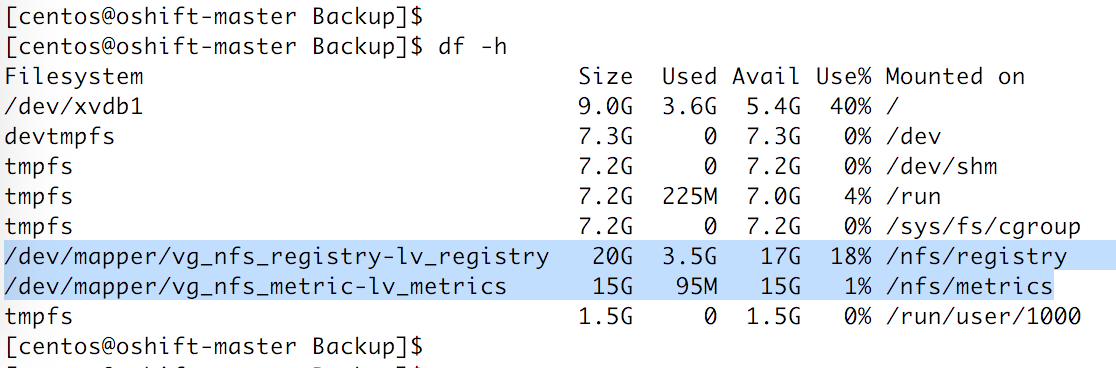
Az alábbi kettő sorral kell kiegészíteni (TAB pozicionált sorok!)

/dev/vg\_nfs\_metric/lv\_metrics /nfs/metrics xfs defaults 0 0

/dev/vg\_nfs\_registry/lv\_registry /nfs/registry xfs defaults 0 0

Mount…

sudo mount -a



**4.) NFS**

NFS package telepítése (CentOS esetén az nfs-utils-lib nem kell)

yum install nfs-utils nfs-utils-lib

sudo systemctl start nfs-server

sudo systemctl start rpcb

sudo systemctl start nfs-lock

sudo systemctl start nfs-idmap

sudo systemctl enable nfs-server

sudo systemctl enable rpcb

sudo systemctl enable nfs-lock

sudo systemctl enable nfs-idmap

Az export fájl létrehozása és szerkesztése

sudo vi /etc/exports

Az alábbi kettő sort kell beleírni (a privátIP tartományból enged NFS csatlakozási kérelmet)

/nfs/metrics 10.1.1.0/24(rw,sync,no\_root\_squash,no\_all\_squash)

/nfs/registry 10.1.1.0/24(rw,sync,no\_root\_squash,no\_all\_squash)

Az iptables állomány lecserélése a mellékelt és szerkesztett verzióra.

Majd újra kell indítani az iptables -t

Ha az sikeres és a “sudo iptables -L” mutatja az NFS portokat, akkor a NFS szervert

(feltételezve, hogy a másolandó és szerkesztett iptables fáj a /tmp alatt található…

sudo cp /tmp/iptables /etc/sysconfig/iptables

sudo systemctl restart iptables

sudo iptables -L

NFS restart…

sudo systemctl restart nfs-server

**5)**

A host saját docker volume konfigurálása (nem az OpenShift belső Docker diszkterület !!!)

Docker csomag telepítése

sudo yum install docker

Ha ez a telepítés utáni állapot (a docker még nem volt elindítva illetve használva), akkor ez a lépés nem szükséges!

sudo rm /var/lib/docker/

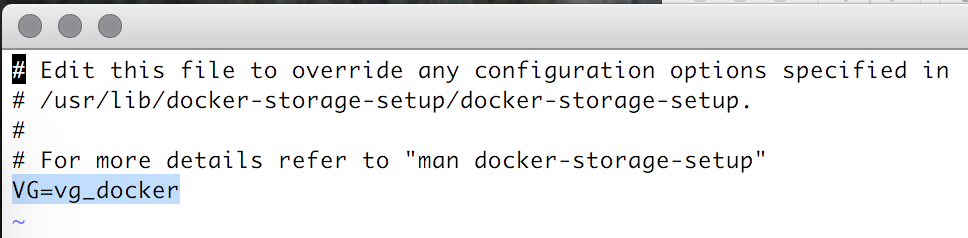
sudo mkdir /var/lib/docker/

A saját VG definiálása a Docker storage számára…

sudo vi /etc/sysconfig/docker-storage-setup

Az alábbi sort kell elhelyezni

VG=vg\_docker



A docker storage telepítése a megadott VG -re…

sudo docker-storage-setup

Sikeres lefutást követően Docker indítás és storage ellenőrzés

sudo systemctl start docker

sudo systemctl status docker

sudo systemctl enable docker

sudo docker info

Annak érdekében, hogy nem megbízható tanúsítvány alatt is el lehessen érni a Docker -t (az OpenShift hálózatból), az INSECURE tartományt definiálni kell

(később ez korrigálható…)

sudo vi /etc/sysconfig/rocker

Az alábbi sort kell elhelyezni (egy kommentezett sor már létezik, azt kell kiegészíteni)

INSECURE\_REGISTRY='--insecure-registry 172.30.0.0/16

Docker daemon újraindítása

sudo systemctl restart docker

**6.)**

NetworkManager szükséges az OpenShift működéséhez

sudo yum install NetworkManager

sudo systemctl start NetworkManager

sudo systemctl enable NetworkManager

**7.)**

A Compute gépeket is elő kell készíteni

A 2.lépés (csomag telepítés) megismétlése mindegyik compute gépen

A 3. lépés (Diszk konfigurálás) megismétlése mindegyik gépen

* A compute gépek esetén csak gép saját, belső docker diszkjét kell konfigurálni

sudo fdisk /dev/xvdc

sudo vgcreate vg\_docker /dev/xvdc1

* Ezeken a gépeken a /nfs könyvtárat és alkönyvtárakat létre kell hozni, hiszen ide történik majd az NFS mapp

sudo mkdir /nfs

sudo mkdir -p /nfs/registry/registry

sudo mkdir -p /nfs/metrics/metrics

sudo chmod -R 777 /nfs

A 4. lépés (NFS) végrehajtása, de itt csak klienst konfirurálunk

yum install nfs-utils nfs-utils-lib

sudo systemctl start nfs-server

sudo systemctl start rpcb

sudo systemctl start nfs-lock

sudo systemctl start nfs-idmap

sudo systemctl enable nfs-server

sudo systemctl enable rpcb

sudo systemctl enable nfs-lock

sudo systemctl enable nfs-idmap

Az fstab szerkesztése, az alábbi kettő sor hozzáadása

10.1.1.101:/nfs/metrics /nfs/metrics nfs rw,sync,hard,intr 0 0

10.1.1.101:/nfs/registry /nfs/registry nfs rw,sync,hard,intr 0 0

Az NFS diszkek felcsatolása…

sudo mount -a

Tesztelés…

touch /nfs/registry/test\_registry.txt

touch /nfs/metrics/test\_metrics.txt

Az 5. (Docker) és 6. (NetworkManager) lépések megismétlése mindegyik compute gépen

**8.)**

Ansible és annak szükséges csomagjának a telepítése a master gépen

sudo yum -y --enablerepo=epel install ansible pyOpenSSL

**9.)**

Master gépen, ha nem a $HOME a jelenleg aktuális könyvtár, akkor

cd ~

OpenShift-Ansible telepítő csomag klónozása a hivatlasos GitHub elérésről

git clone https://github.com/openshift/openshift-ansible -b release-1.4

**10.)**

A gyakorlat során az OpenShift környezet felhasználó azonosítását a http\_auth biztosítja.

Ennek használatához szükséges segédprogram telepítése

(Csak a Master gépen)

sudo yum install -y httpd-tools

**11.)**

Az Ansible telepítő konfigurációs állományokat (hosts és ansible.cfg) le kell cserélni a mellékelt állományokra (master gép)

Feltételezve, hogy az állományok a /tmp alatt találhatóak…

sudo cp /tmp/hosts /etc/ansible

sudo cp /tmp/ansible.cfg /etc/ansible

**12.)**

Kucs-cserés SSH konfigurálása a Master és Compute gépek között (**centos** felhasználó!)

cd ~

ssh-keygen -t rsa

touch /home/centos/.ssh/authorized\_keys

sudo chmod u=rwx,g=,o= /home/centos/.ssh

sudo chmod u=rw,g=,o= /home/centos/.ssh/authorized\_keys

cat .ssh/id\_rsa.pub >> .ssh/authorized\_keys

A Master gépen a **/home/centos/.ssh/id\_rsa.pub** file tartalmát bele kell illeszteni a **/home/centos/.ssh/authorized\_keys** állományba a Compute gépeken.

Ugyan ezt a lépés a Compute gépek és Master között is meg kell ismételni.

Mind a kettő gép között, egymásra bejelentkezve a kucs-cserét ellenőrizni kell.

(első bejelentkezés során a “yes” válasz megadása szükséges)

Compute… ->

**ssh centos@oshift-master-priv**

Master ->

**ssh centos@oshift-compute1-priv**

**ssh centos@oshift-compute2-priv**

A Master gépről önmagára bejelentkezve szintén ellenőrizni kell az SSH bejelentkezést

Master ->

**ssh centos@oshift-master-priv**

**13.)**

Az Ansible és a gépek konfigurációjának ellenőrzése

cd openshift-ansible/

ansible -i /etc/ansible/hosts nodes -a 'hostnamectl'

**14.)**

Sikeres teszt esetén a telepítés elindítása

ansible-playbook -i /etc/ansible/hosts ~/openshift-ansible/playbooks/byo/config.yml

**15.)**

Az “admin” felhasználónak, akit a telepítés során az Ansible hozott lére “cluster-admin” szerepkört kell adományozni

sudo oadm policy add-cluster-role-to-user cluster-admin admin

**16.)**

Nézzük meg, milyen service-account alatt ténykedünk jelenleg

oc whoami -c

**17.)**

Ellenőrizzük a node-k állapotát

oc get nodes

**18.)**

Ellenőrizzük a projekteket és a jelenleg aktív projektünket.

oc get projects

oc project

**19)**

Alap esetben a “Default” projectben kezdjük a munkát telepítést követően.

A prokekt státuszát kérdezzük le.

oc status

**20)**

Ellenőrizzük a külső \* [xip.io](http://xip.io) doamin elérhetőségét és a névfelfoldást.

nslookup 140.86.55.167.xip.io

**21.)**

Ellenőrizzük a console elérhetőségét

<https://oc-140-86-55-167.compute.oraclecloud.com:8443/console/>

vagy…

<https://140.86.55.167.xip.io:8443/console>

felhasználó: admin

jelszó: welcome1

**22.)**

Az “openshift-infra” project alatt a hawkular-metrics és heapster konfigurálása általában nem sikerül…

Kézzel kell helyrehozni…

Metrics deploy by hand

----------------

oc process -f ./roles/openshift\_hosted\_templates/files/v1.4/origin/metrics-deployer.yaml -v HAWKULAR\_METRICS\_HOSTNAME=hawkular-metrics-openshift-infra.apps.140.86.55.167.xip.io -v CASSANDRA\_PV\_SIZE=10Gi -v MASTER\_URL=https://10.1.1.101:8443 -v IMAGE\_VERSION=v1.4.1 | oc create -f -

EZT A FEJEZETET KÜLÖN SZÓBAN KELLENE ÁTBESZÉLNI!!!!

**23.)**

Ebben a pontban a Docker image -ek fordítása és az OpenShift belső registry alá töltésre kerül sor.

FONTOS!!! a lentebb jelzett pontokban ki kell nyerni az aktuális Docker-Registry által birtokolt IP címet és azt esetleges visszatöltés miatt érdemes megjegyezni!

cd Docker/

cd OracleJDK/java-8/

./build.sh

sudo docker images

cd ../../Coherence/

cd base/

sudo ./builDockerImage.sh -s

sudo docker images

cd ../demo/

sudo docker build -t oracle/coherence-server:v1.0 -f Dockerfile .

sudo docker images

cd ../../Wildfly/10.1.0/

sudo docker build -t oracle/wildfly-coherence-server:v1.0 -f Dockerfile .

sudo docker images

cd ../../Apache/

sudo docker build -t oracle/apache-demo:v1.0 -f Dockerfile .

sudo docker images

oc project default

oc get svc *——-> itt az aktuális docker-registry IP címe kell !!!!!!!!!*

oc login -u admin https://oshift-master-priv:8443

oc whoami -t

*(itt és ezt követően az IP cím a korábban meghatározott docker-registry IP címe!!!)*

*(A token az előző pontban visszaadott aktuális token!!!)*

sudo docker login -u admin -p IpJ-BanuklXFYY5YAwGc2rD\_vxrWvj8vY91lpY2KXIQ https://172.30.192.247:5000

oc new-project coherence-demo --display-name="Coherence Demo"

oc project

sudo docker tag oracle/coherence-server:v1.0 172.30.192.247:5000/coherence-demo/coherence-server:v1.0

sudo docker tag oracle/wildfly-coherence-server:v1.0 172.30.192.247:5000/coherence-demo/wildfly-coherence-server:v1.0

sudo docker tag oracle/apache-demo:v1.0 172.30.192.247:5000/coherence-demo/apache-demo:v1.0

sudo docker images

sudo docker push 172.30.192.247:5000/coherence-demo/coherence-server:v1.0

sudo docker push 172.30.192.247:5000/coherence-demo/wildfly-coherence-server:v1.0

sudo docker push 172.30.192.247:5000/coherence-demo/apache-demo:v1.0

oc get images |grep coherence

oc get istag

**24.)**

cd ~/Template/

oc create -f oc\_coherence\_headless\_service.yaml

oc get svc

oc create -f template/coherence-demo.yaml

oc new-app --template=coherence-demo -p CLUSTER\_PORT=7574 -p CLUSTER\_NAME=DemoCoherCluster

oc status

oc get pods

oc describe coherence-server-1-deploy - - - > itt a név az aktuális generált pod név!!!!

**25.)**

Indításkor felfelé skálázás

(fontos!!! A doksi rossz karaktert másolhat. a replicas előtt kettő minusz karakter szerepel!)

oc scale dc/coherence-server —replicas=2

oc scale dc/wildfly-coherence —replicas=1

oc scale dc/apache-coherence —replicas=1

**26.)**

Leálláskor lefelé skálázás

oc scale dc/apache-coherence —replicas=0

oc scale dc/wildfly-coherence —replicas=1

oc scale dc/coherence-server —replicas=1

**27.)**

ha minden pod elindult és fut a környezet, akkor a Coherence szerver adatokkal feltöltése

(az IP cím a Widfly szerver által birtokolt és SVC által kisorsolt IP elérés)

curl <http://172.30.211.203:8080/DemoTelekom2/rest/Manage/preload>

Ellenőrizni a pod logokat pl. így lehet…

oc logs -f wildfly-coherence-1-1idbz

**28.)**

Elérések:

OpenShift Web Console:

<https://oc-140-86-55-167.compute.oraclecloud.com:8443/console/>

Coherence Apache alkalmazás:

<http://coherence-demo-coherence-demo.apps.140.86.55.167.xip.io/demo>

GitLab:

<http://oc-140-86-53-111.compute.oraclecloud.com:4080/>