DevOps Challenge / İlhan Furkan Sadıkoğlu

Kubernetes ve Helm kurulumu

```
sudo su
touch k8s_install.sh
chmod u+x k8s_install.sh
nano k8s_install.sh
```

HOSTNAME değişkeninin düzeltilmesi gerekir.

k8s_install.sh:

```
#! /bin/bash -xe
apt-get update -y
apt-get upgrade -y
apt-get -y install linux-image-generic-hwe-18.04
apt-get install -y sshpass
ssh-keygen -b 2048 -t rsa -f ~/.ssh/id_rsa -q -N ""
sed -ie 's/PasswordAuthentication no/PasswordAuthentication yes/g' /etc/ssh/sshd_config
sed -ie 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/g' /etc/ssh/sshd_config
systemctl restart sshd.service
echo wordpress > /root/password.txt
echo "root:wordpress" | chpasswd
su -c "sshpass -f /root/password.txt ssh-copy-id root@HOSTNAME -o StrictHostKeyChecking=no"
apt -y install python3-pip
cd /root/
git clone https://github.com/kubernetes-sigs/kubespray.git
cd /root/kubespray
pip3 install -r requirements.txt
cp -rfp inventory/sample inventory/mycluster
PRVT=$(hostname -I)
declare -a IPS=($PRVT)
CONFIG_FILE=inventory/mycluster/hosts.yaml python3 contrib/inventory_builder/inventory.py $IPS
sed -ie 's/node1/HOSTNAME/g' inventory/mycluster/hosts.yaml
#sed -ie 's/PasswordAuthentication yes/PasswordAuthentication no/g' /etc/ssh/sshd_config
#sed -ie 's/PermitRootLogin yes/#PermitRootLogin prohibit-password/g' /etc/ssh/sshd_config
systemctl restart sshd.service
ansible-playbook -i inventory/mycluster/hosts.yaml --become-user=root cluster.yml
curl -fsSL -o get_helm.sh https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
```

```
bash k8s_install.sh
```

Not: Version check'te hata verirse requirements.txt ve cluster.yml'daki version değerlerini düzeltmek gerekir. (ansible 6.7.0, core: 2.13.7)

Not: kubespray altındaki playbooks altında ansible_version.yml altında minimal ve maximal değerin arasında kalması gereken değer: 2.13.7 dir. yani 2.13.0 ve 2.14.0 verilebilir.

Düzeltmeler sonrasında işlemlere 16. satırdaki `cd /root/kubespray` komutundan devam edilir.

K9S kurulumu (Management UI)

```
sudo su
curl -sS https://webi.sh/k9s | sh
source ~/.config/envman/PATH.env
```

Postgresql Installation

```
helm repo add bitnami https://charts.bitnami.com/bitnami
nano postgres-pv.yaml
```

postgres-pv.yaml:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: postgresql-pv
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data1"
```

```
kubectl apply -f postgres-pv.yaml
nano postgres-pvc.yaml
```

postgres-pvc.yaml:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: postgresql-pv-claim
spec:
  storageClassName: manual
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
```

```
kubectl apply -f postgres-pvc.yaml
kubectl get pv
kubectl get pvc
```

Son 2 komutun çıktısında volume'ün bound olduğundan emin olunmalıdır.

Ardından:

```
helm install postgresql bitnami/postgresql --set persistence.existingClaim=postgresql-pv-claim --set volumePermissions.enabled=true
```

Bu adımların ardından kurulum tamamlanır. K9S üzerinden kontrol yapılabilir.

Root parolasını almak için:

```
export POSTGRES_PASSWORD=$(kubectl get secret --namespace default psql-test-postgresql -o jsonpath="{.data.postgresql-password}" | base64 --decode)
```

Son olarak **k edit svc postgresql** komutu ile service düzenlenir ve nodeport ayarlanır:

```
nodePort=31532
type: NodePort
```

Postgre'ye şu komutla bağlanılır:

```
psql --host 127.0.0.1 -U postgres -d postgres -p 31532
```

Backup

Aşağıdaki işlemler backup sunucusunda yapılır. `USERNAME`, `KUBERNETESIP`, `/path/to/db-backups` ve `/path/to/mnt` düzenlenmelidir. Ayrıca backup sunucusunun public key'i kubernetes makinesine eklenmelidir. Backup sunucusunda pg_dump'un aynı versiyonu bulunmalıdır.

```
mkdir /path/to/db-backups
touch db-backups.sh
chmod u+x db-backups.sh
nano db-backups.sh
```

```
#! /bin/bash -xe
ssh USERNAME@KUBERNETESIP 'tar -zcf postgres_data-$(hostname)-$(date +%d-%m-%y).gz /path/to/mnt && echo "data alındı"'
PGPASSWORD="$POSTGRES_PASSWORD" pg_dumpall -h KUBERNETESIP -U postgres -p 31532 -f postgres_dump-$(hostname)-$(date +%d-%m-%y).sql'
```

```
crontab -e
```

Aşağıdaki satır eklenir:

```
0 0 * * * bash /path/to/db-backups/db-backups.sh
```

Redis Kurulumu

```
helm install redis bitnami/redis
export REDIS_PASSWORD=$(kubectl get secret --namespace default redis -o jsonpath="{.data.redis-password}" | base64 --decode)
```

```
nano redis-pv.yaml
```

redis-pv.yaml:

```
apiVersion: v1
kind: PersistentVolume
metadata:
name: redis-data-redis-master0
spec:
capacity:
storage: 8Gi
accessModes:
- ReadWriteOnce
hostPath:
path: "/storage/data-master0"

```

```
---
apiVersion: v1
kind: PersistentVolume
metadata:
name: redis-data-redis-replicas0
spec:
capacity:
storage: 8Gi
accessModes:
- ReadWriteOnce
hostPath:
path: "/storage/data-replicas0"
```

```
kubectl apply -f redis-pv.yaml
```

K9s üzerinden podun çalıştığı teyit edilir.

Son olarak **k edit svc redis** komutu ile service düzenlenir ve nodeport ayarlanır:

```
nodePort=31679
type: NodePort
```

Test için redis-cli kullanılabilir.

```
sudo apt install redis-tools
redis-cli -h KUBERNETESIP -p 31679 ping
```