|  |  |
| --- | --- |
|  |  |
| **Hüseyin Məmmədli** | **DevOps-2** |
| **#**export KUBEVIRT\_VERSION=$(curl -s https://api.github.com/repos/kubevirt/kubevirt/releases/latest | jq -r .tag\_name)  echo $KUBEVIRT\_VERSION |  |
| #kubectl create -f https://github.com/kubevirt/kubevirt/releases/download/${KUBEVIRT\_VERSION}/kubevirt-operator.yaml | deploy the KubeVirt Operator |
| # kubectl create -f https://github.com/kubevirt/kubevirt/releases/download/${KUBEVIRT\_VERSION}/kubevirt-cr.yaml |  |
| # wget -O virtctl https://github.com/kubevirt/kubevirt/releases/download/${KUBEVIRT\_VERSION}/virtctl-${KUBEVIRT\_VERSION}-linux-amd64 | Install Virtctl |
| # chmod +x virtctl |  |
| # kubectl get pods -n kubevirt | Check deployment |
| # kubectl -n kubevirt get kubevirt |  |
| # kubectl apply -f https://kubevirt.io/labs/manifests/vm.yaml |  |
| # kubectl get vms  # kubectl get vms -o yaml testvm | grep -E 'running:.\*|$' | Create VM |
| # kubectl get vms  # kubectl get vmis | Check VM status |
| # ./virtctl console testvm  # ./virtctl stop testvm  # kubectl delete vms testvm | Shutdown and cleanup |
| \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |
| #kubectl get storageclass |  |
| #export VERSION=$(curl -Ls https://github.com/kubevirt/containerized-data-importer/releases/latest | grep -m 1 -o "v[0-9]\.[0-9]\*\.[0-9]\*")  echo $VERSION | Install the Containerized Data Importer |
| #kubectl create -f https://github.com/kubevirt/containerized-data-importer/releases/download/$VERSION/cdi-operator.yaml  #kubectl -n cdi scale deployment/cdi-operator --replicas=1 |  |
| #kubectl create -f <https://github.com/kubevirt/containerized-data-importer/releases/download/$VERSION/cdi-cr.yaml>  #kubectl get cdi -n cdi | Create CRD to trigger operator deployment of CDI |
| #kubectl wait -n cdi --for=jsonpath='{.status.phase}'=Deployed cdi/cdi |  |
| #kubectl -n cdi get pods |  |
| #cat <<EOF > pvc\_cirros.yml  apiVersion: v1  kind: PersistentVolumeClaim  metadata:  name: "cirros"  labels:  app: containerized-data-importer  annotations:  cdi.kubevirt.io/storage.import.endpoint: "http://download.cirros-cloud.net/0.5.2/cirros-0.5.2-x86\_64-disk.img"  kubevirt.io/provisionOnNode: node01  spec:  accessModes:  - ReadWriteOnce  resources:  requests:  storage: 120Mi  EOF  kubectl create -f pvc\_cirros.yml | Use CDI to upload a VM image |
| #kubectl get pod |  |
| #kubectl logs importer-cirros -f |  |
| #cat <<EOF > vm1.yml  apiVersion: kubevirt.io/v1  kind: VirtualMachine  metadata:  labels:  kubevirt.io/os: linux  name: vm1  spec:  running: true  template:  metadata:  creationTimestamp: null  labels:  kubevirt.io/domain: vm1  spec:  domain:  cpu:  cores: 1  devices:  disks:  - disk:  bus: virtio  name: disk0  - cdrom:  bus: sata  readonly: true  name: cloudinitdisk  resources:  requests:  memory: 128M  volumes:  - name: disk0  persistentVolumeClaim:  claimName: cirros  - cloudInitNoCloud:  userData: |  #cloud-config  user: cirros  password: gocubsgo  hostname: vm1  ssh\_pwauth: True  disable\_root: false  name: cloudinitdisk  EOF  kubectl create -f vm1.yml | create a virtual machine that makes use of our new PVC. |
| #kubectl get pod -o wide  #kubectl get vmi |  |
| #IP=$(kubectl get vmi vm1 -o jsonpath='{.status.interfaces[0].ipAddress}') |  |
| #ssh cirros@${IP} |  |
| #ssh-copy-id -i ~/.ssh/id\_rsa.pub cirros@${IP} |  |
| #ssh cirros@${IP} hostname  #virtctl stop vm1  #kubectl get vmi |  |
| #virtctl start vm1  #IP=$(kubectl get vmi vm1 -o jsonpath='{.status.interfaces[0].ipAddress}'  #ssh cirros@${IP} hostname | Start the VM back up |