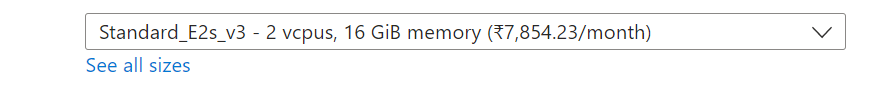
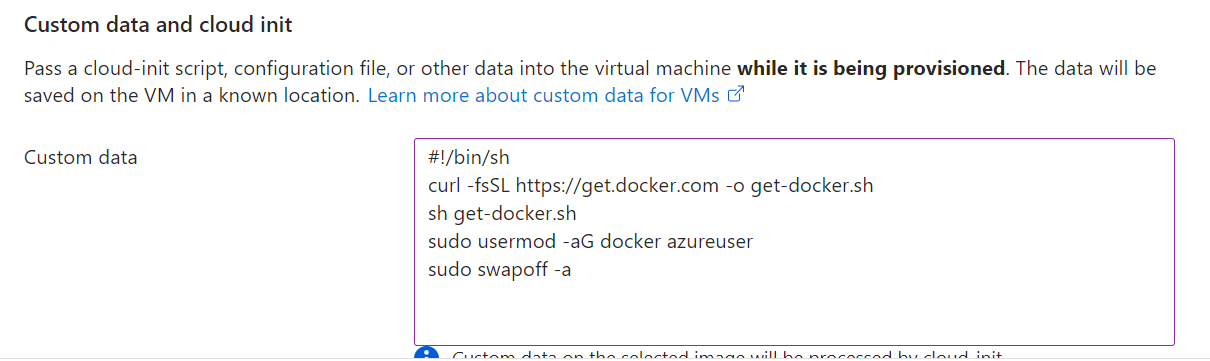
|  |
| --- |
| #!/bin/sh  curl -fsSL https://get.docker.com -o get-docker.sh  sh get-docker.sh  sudo usermod -aG docker azureuser  sudo swapoff -a |

Installation of Kubernetes

Create 2 or 3 VM’s by giving the following commands in advance



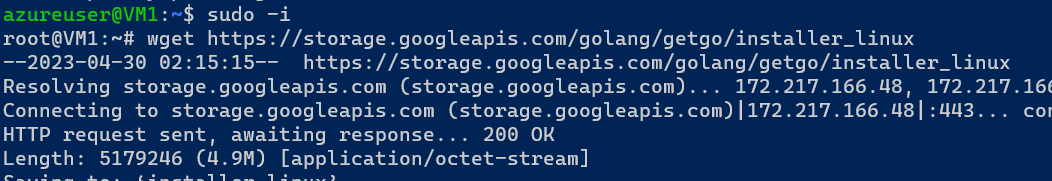


vm's connect to PowerShell

docker info

sudo -i

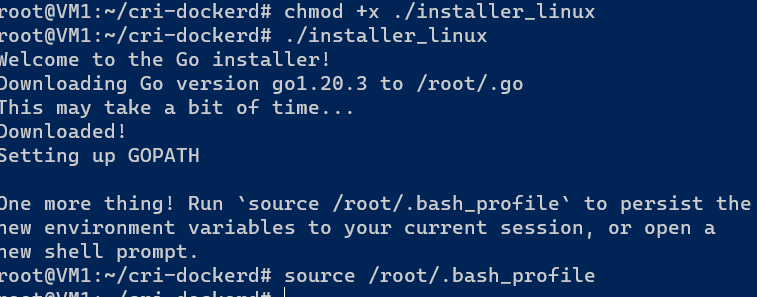
wget <https://storage.googleapis.com/golang/getgo/installer_linux>



chmod +x ./installer\_linux

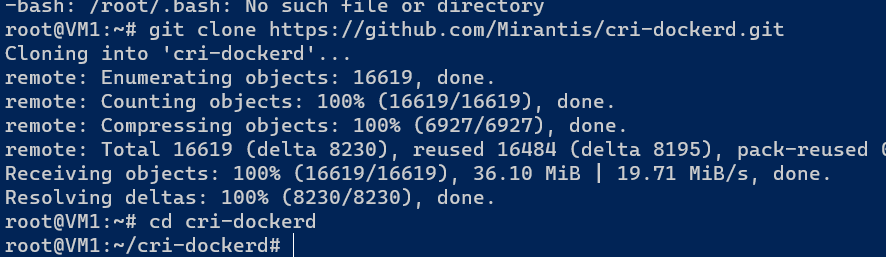
./installer\_linux

source /root/.bash\_profile



git clone https://github.com/Mirantis/cri-dockerd.git

cd cri-dockerd



mkdir bin

go build -o bin/cri-dockerd

mkdir -p /usr/local/bin

install -o root -g root -m 0755 bin/cri-dockerd /usr/local/bin/cri-dockerd

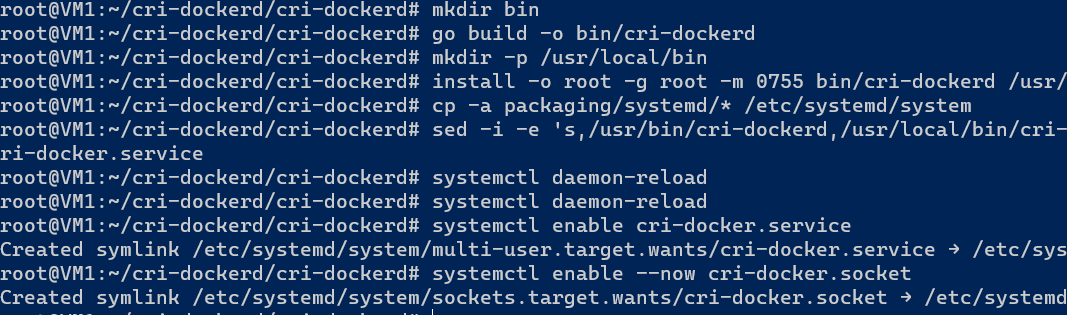
cp -a packaging/systemd/\* /etc/systemd/system

sed -i -e 's,/usr/bin/cri-dockerd,/usr/local/bin/cri-dockerd,' /etc/systemd/system/cri-docker.service

systemctl daemon-reload

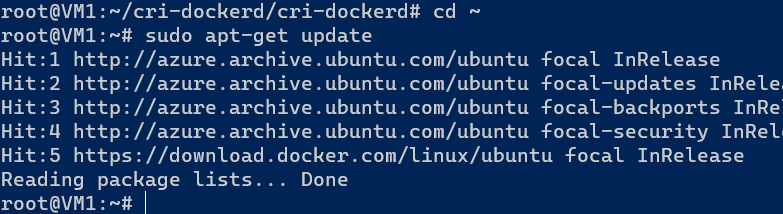
systemctl enable cri-docker.service

systemctl enable --now cri-docker.socket

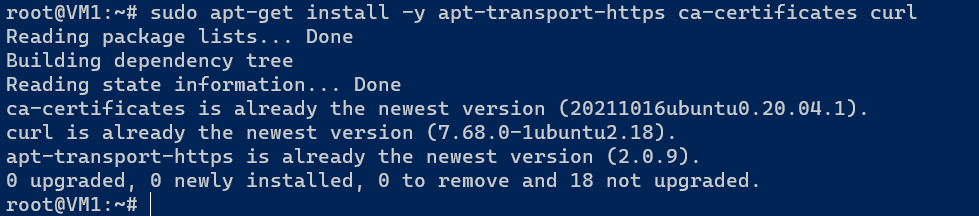


cd ~

sudo apt-get update

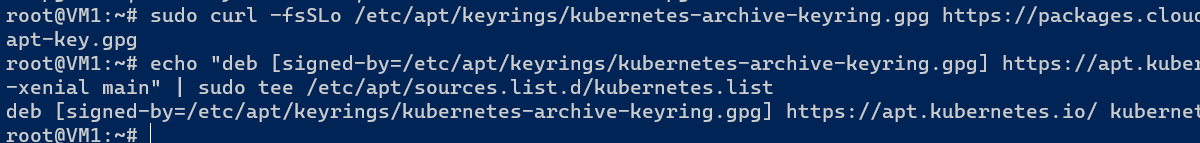


sudo apt-get install -y apt-transport-https ca-certificates curl



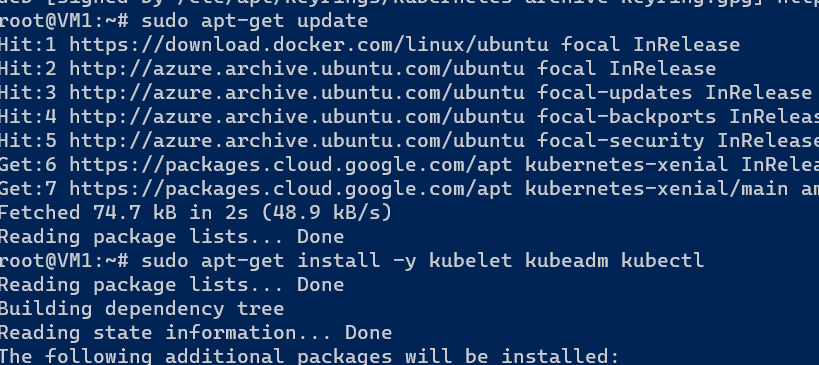
sudo curl -fsSLo /etc/apt/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg

echo "deb [signed-by=/etc/apt/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list

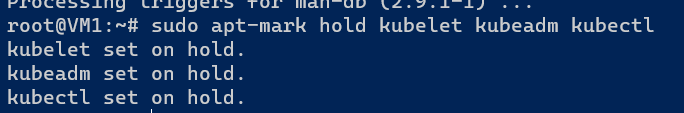


sudo apt-get update

sudo apt-get install -y kubelet kubeadm kubectl

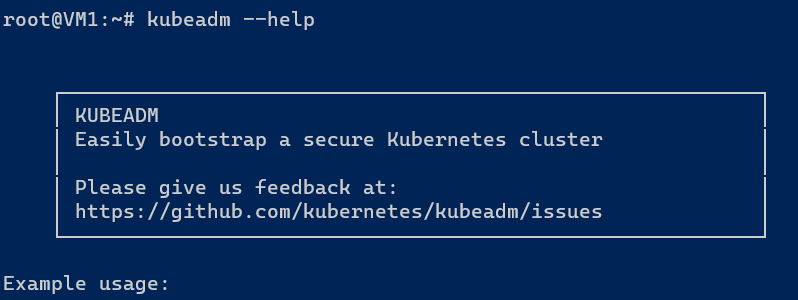


sudo apt-mark hold kubelet kubeadm kubectl

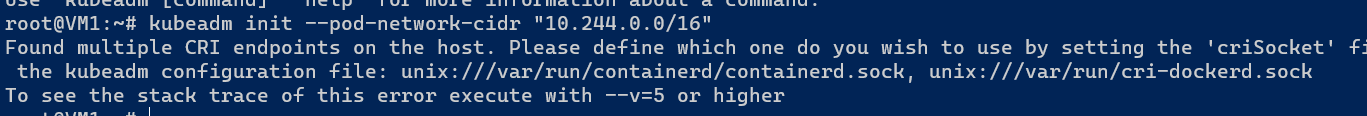


### now doing only master node

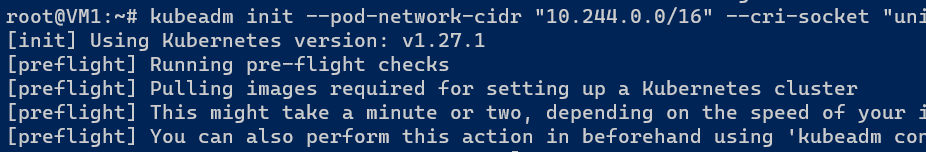
kubeadm --help

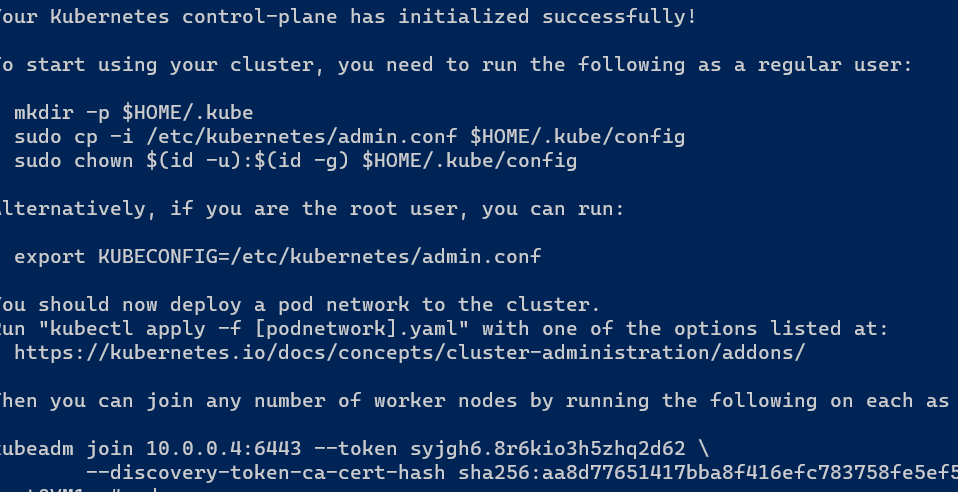


kubeadm init --pod-network-cidr "10.244.0.0/16"



kubeadm init --pod-network-cidr "10.244.0.0/16" --cri-socket "unix:///var/run/cri-dockerd.sock"





Exit

mkdir -p $HOME/.kube

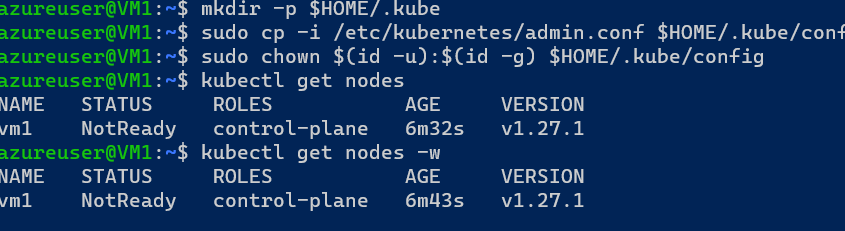
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

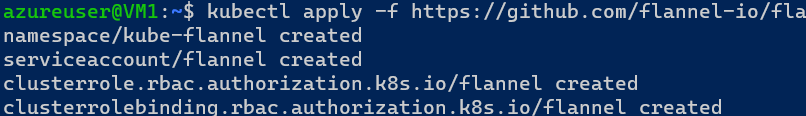
### above three commands from notepad only

kubectl get nodes

kubectl get nodes -w



kubectl apply -f <https://github.com/flannel-io/flannel/releases/latest/download/kube-flannel.yml>



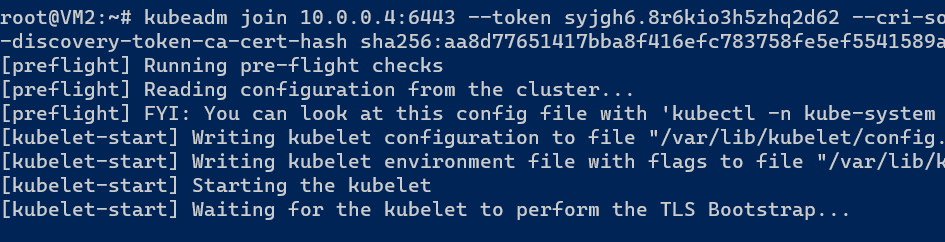
kubectl get nodes

kubectl get nodes -w

### Now doing in Nodes

Paste the command from notepad and add this in between

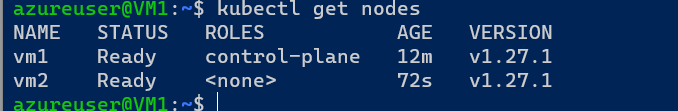
--cri-socket "unix:///var/run/cri-dockerd.sock"



### Then check in master node by using these commands

kubectl get nodes

kubectl get nodes -w



### create manifest files and execution

curl <https://jsonplaceholder.typicode.com/todos/1>

curl https://jsonplaceholder.typicode.com/todos

kubectl api-resources

### go to kubectl cheat sheet

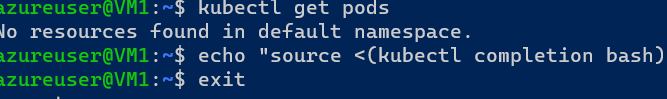
Kubectl (tab button)

source <(kubectl completion bash)

Kubectl get p

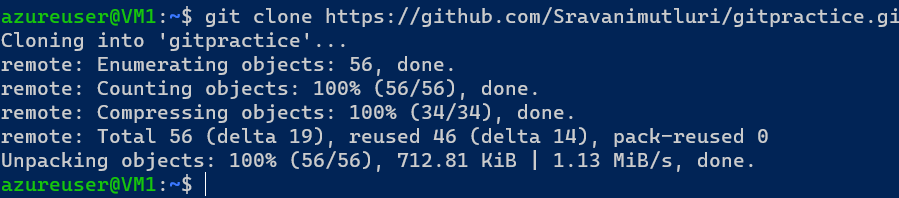
echo "source <(kubectl completion bash)" >> ~/.bashrc

Exit



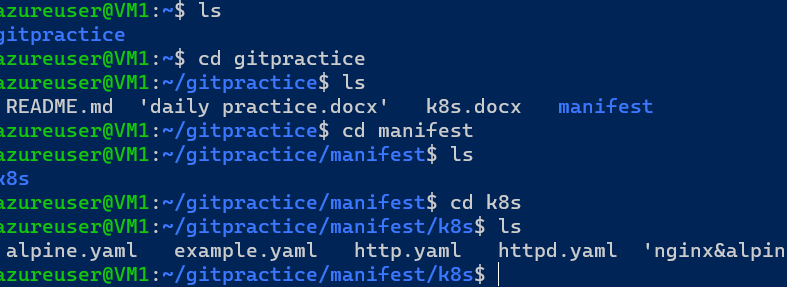
###Reconnect the VM’s again. Create a manifest file and upload into git hub

git clone (paste github code)



cd (manifest file)

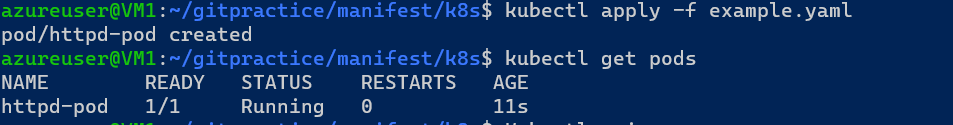




Ls

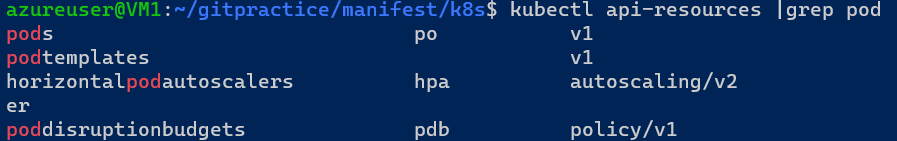
kubectl apply -f (yaml file name)

kubectl get pods



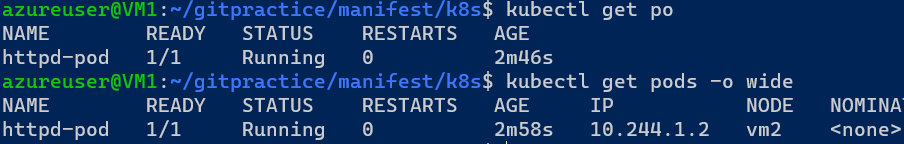
kubectl api-resources

kubectl api-resources |grep pod



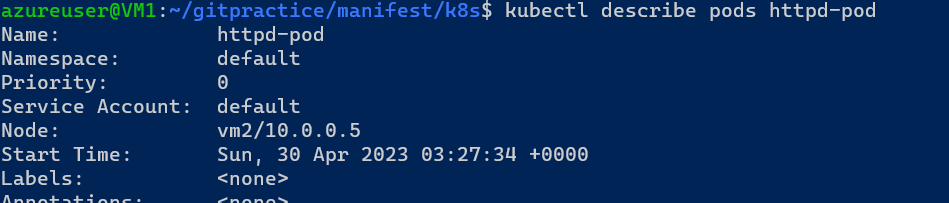
kubectl get po (or) kubectl get pods

kubectl get pods -o wide

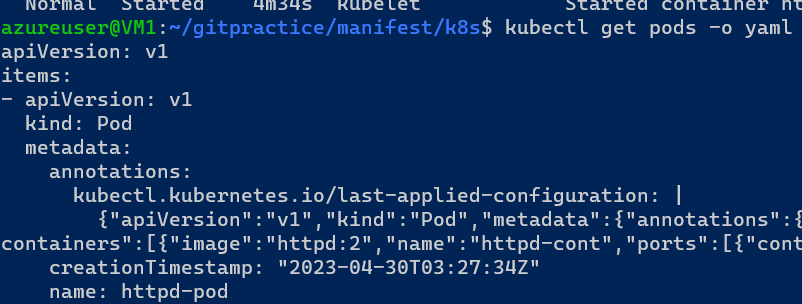


kubectl describe pods

kubectl describe pods (name in yaml file)



kubectl get pods -o yaml



kubectl delete -f <yaml file name>



kubectl get po



kubectl apply -f (yaml file name)

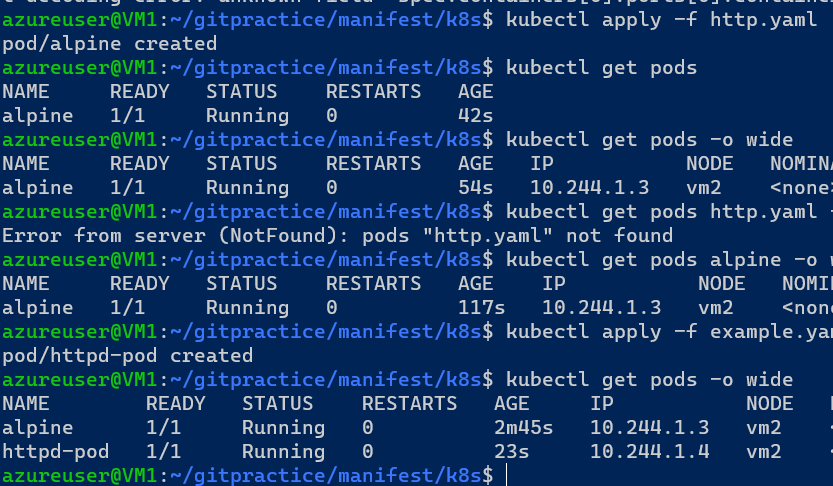
kubectl get pods

kubectl get pods -o wide

kubectl get po (name in yaml file) -o yaml

kubectl apply -f (yaml file name)

kubectl get pods -o wide (check ip address)



kubectl get po

kubectl delete pods alpine pod <names in yamlfile>

git pull

### Apply the following commands to excute the manifest file

kubectl apply -f (file name)

kubectl get po

kubectl get po -o wide

kubectl apply ----> to create

kubectl delete----> to delete

kubectl apply ----> to update also

kubectl get----> to retrive

kubectl describe --> to more information

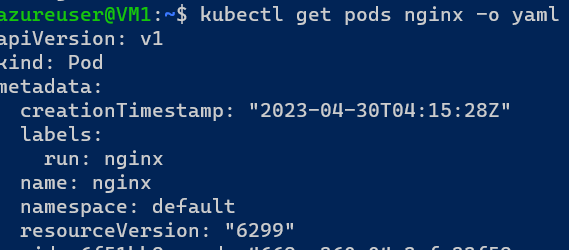
kubectl … -w --> to watch the updates

kubectl api-resource

kubectl run <name in yamefile> --image nginx

kubectl get po

kubectl get pods <name in yamefile> -o yaml



### in above command by using pod name only get the output

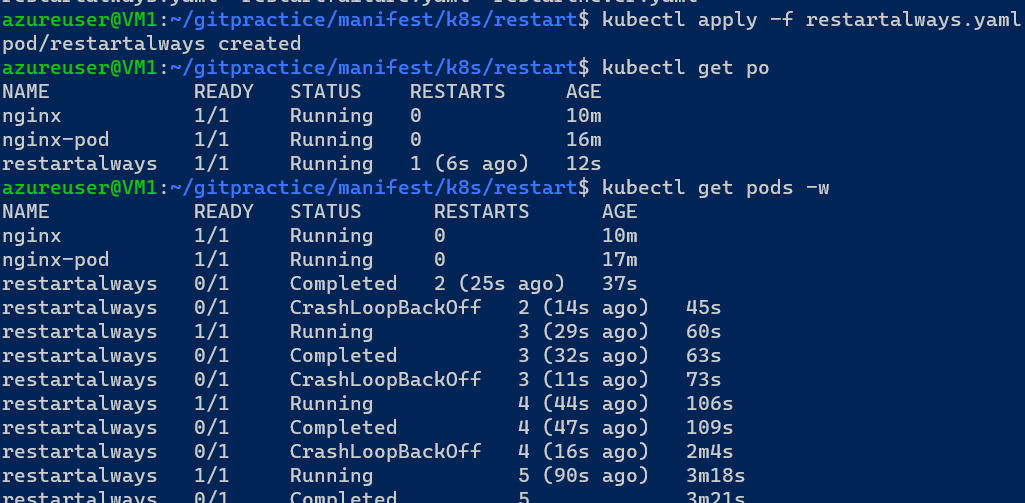
LIFE CYCLE of pods

git pull

kubectl apply -f restartalways.yaml

kubectl get po

kubectl get pods -w

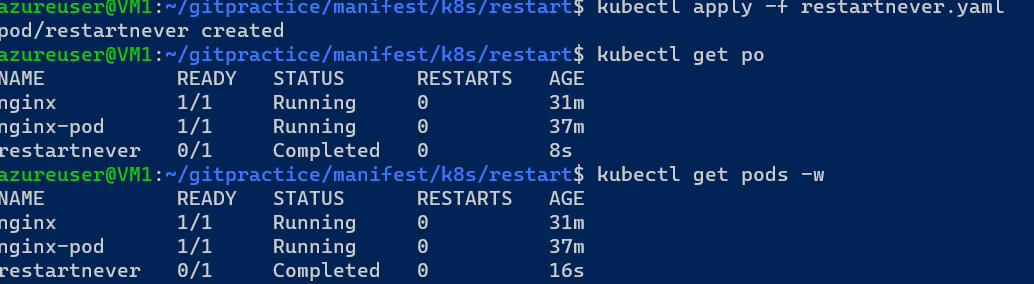


kubectl delete pods restartalways

kubectl apply -f restartnever.yaml

kubectl get po

kubectl get pods -w



kubectl delete pods restartnever

kubectl apply -f restartfailure.yaml

kubectl get po

kubectl get pods -w

