

**Jenkins Integration with Docker image and AKS Cluster**

**Part of Prerequisite for Ubnutu 22.0 version**

**Step 1: Installation of dotnetcore 6.0**

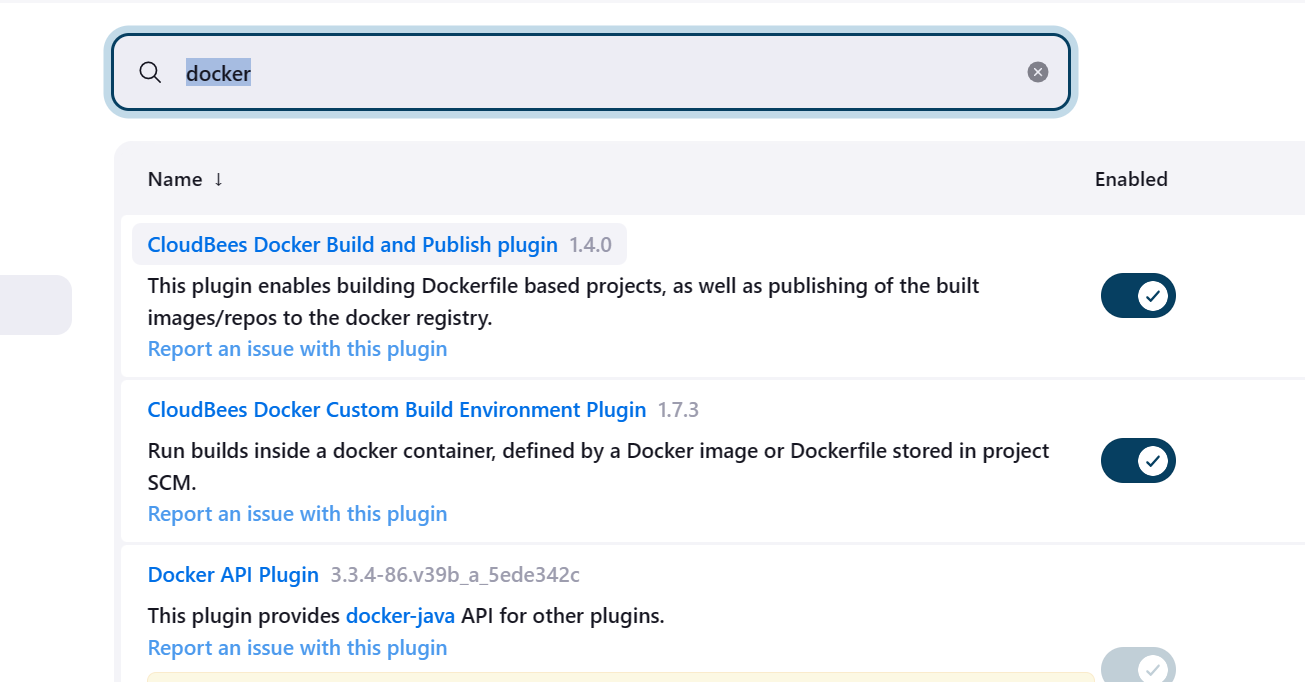
**If you are using Ubuntu 20, you need to download the package from this repo:**

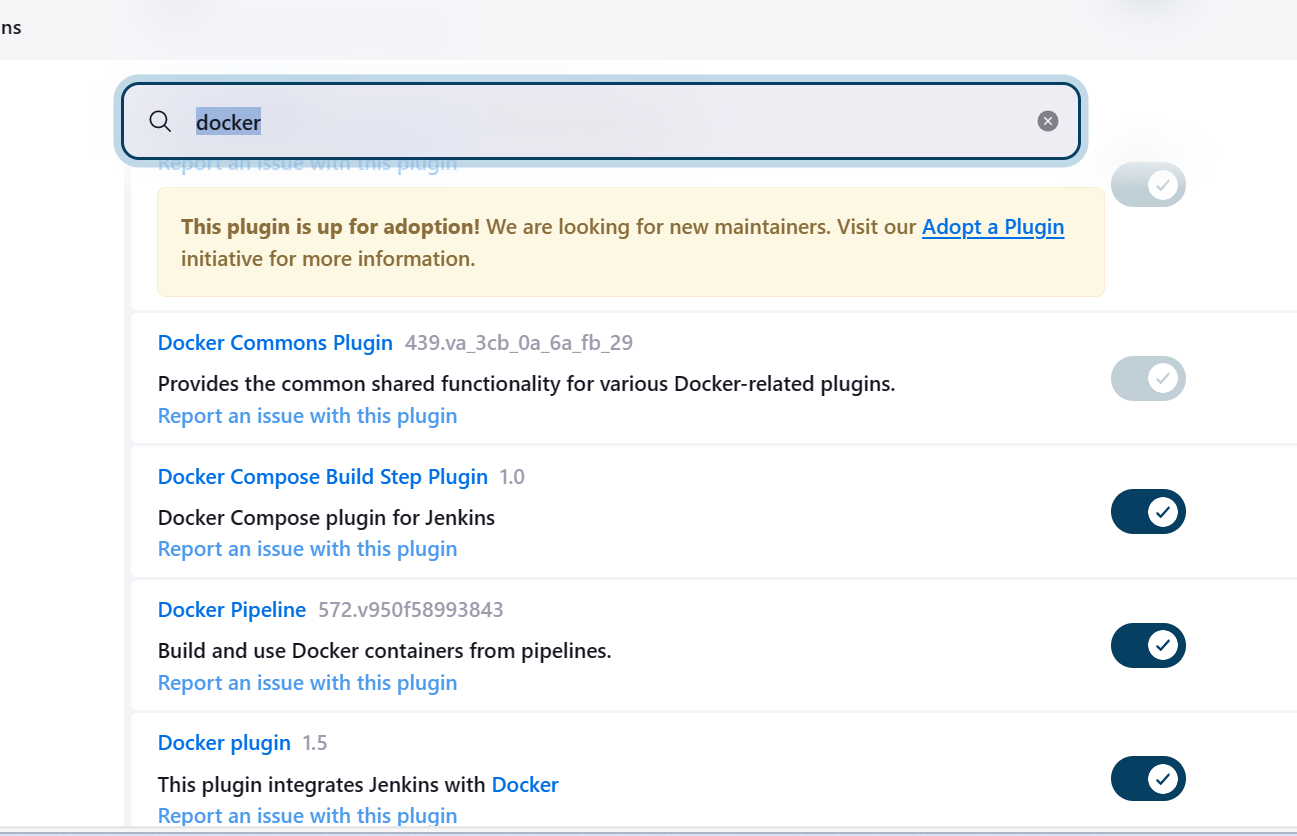
wget https://packages.microsoft.com/config/ubuntu/20.04/packages-microsoft-prod.deb -O packages-microsoft-prod.deb sudo dpkg -i packages-microsoft-prod.deb rm packages-microsoft-prod.deb

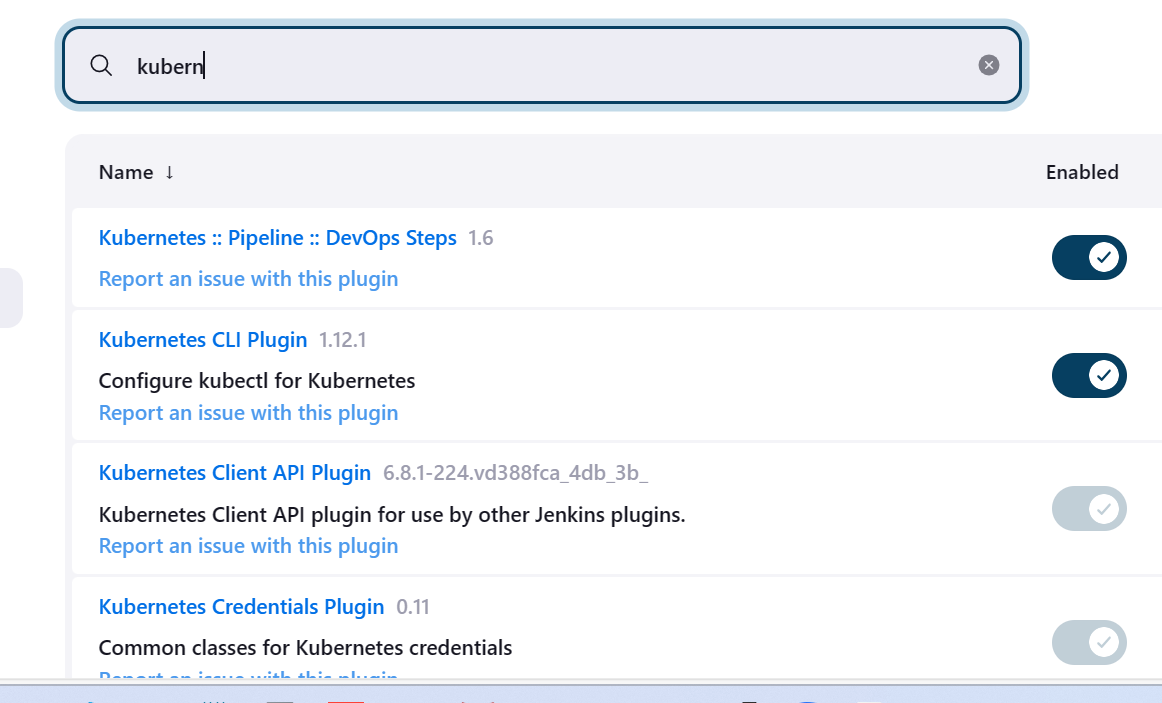
sudo apt-get update && \

sudo apt-get install -y dotnet-sdk-6.0

Plugins







Step 2: Docker Step

**Docker Installation Setup for Ubuntu 22**

*# Add Docker's official GPG key:* sudo apt-get update sudo apt-get install ca-certificates curl gnupg sudo install -m **0755** -d /etc/apt/keyrings curl -fsSL <https://download.docker.com/linux/ubuntu/gpg> **|** sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg sudo chmod a+r /etc/apt/keyrings/docker.gpg *# Add the repository to Apt sources:* echo \ "deb [arch=**$(**dpkg --print-architecture**)** signed-by=/etc/apt/keyrings/docker.gpg] <https://download.docker.com/linux/ubuntu> \ **$(**. /etc/os-release **&&** echo "$VERSION\_CODENAME"**)** stable" **|** \ sudo tee /etc/apt/sources.list.d/docker.list > /dev/null sudo apt-get update

sudo service docker stop

sudo service docker start

**Add jenkins user to Docker group**

**sudo usermod -a -G docker jenkins**  
  
**Restart Jenkins service**  
**sudo service jenkins restart**  
  
**Reload system daemon files**

sudo systemctl daemon-reload  
  
Restart Docker service as well

**Step 3: Kubectl Installation :**

Make sure to Install Docker, Docker pipeline and Kubectl CLI plug-ins are installed in Jenkins

***How to install Kubectl in Ubuntu instance***

**Download keys from google website**

curl -s <https://packages.cloud.google.com/apt/doc/apt-key.gpg> | sudo apt-key add -

sudo touch /etc/apt/sources.list.d/kubernetes.list

echo "deb <http://apt.kubernetes.io/> kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list

**Create the below file**

**Update package manager**

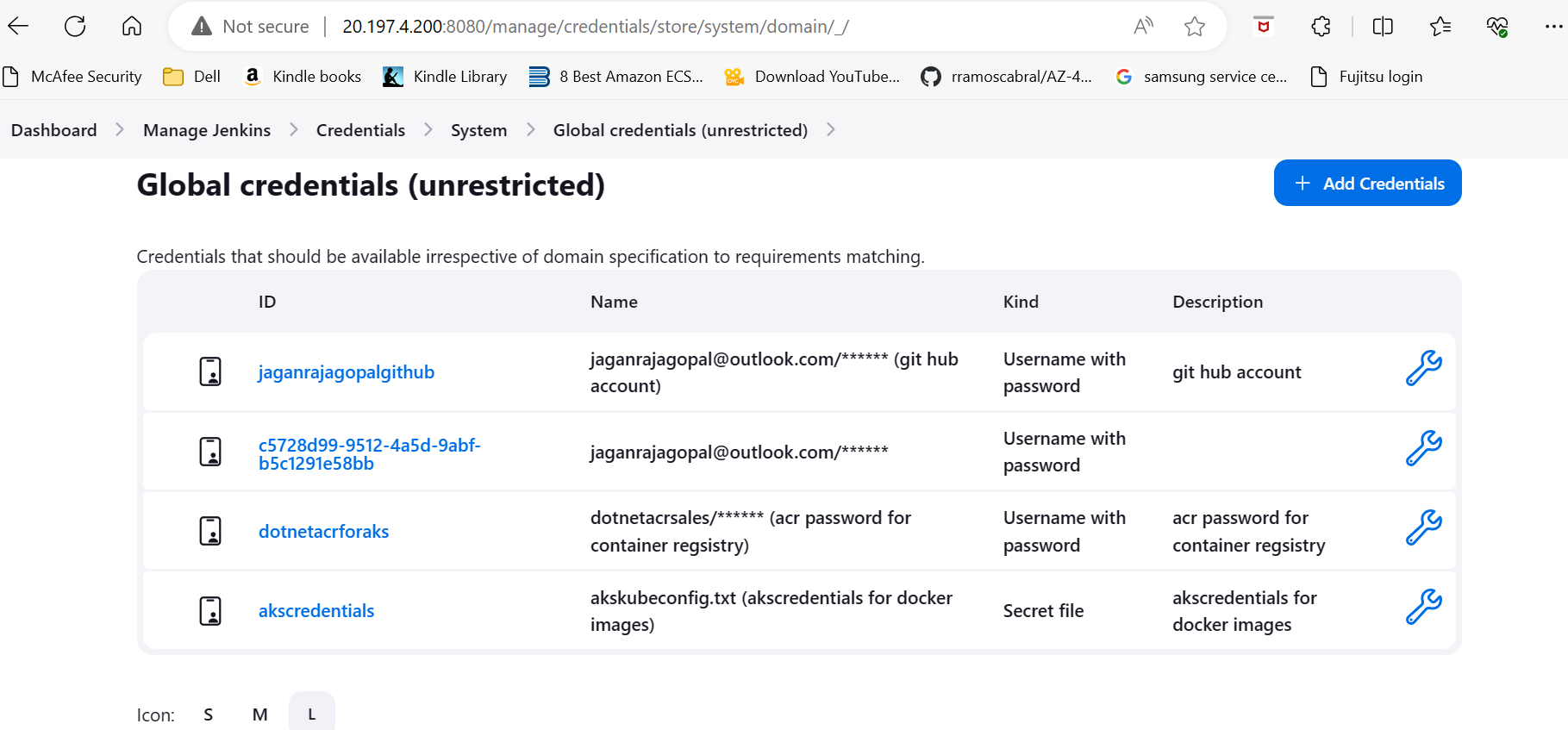
sudo apt-get update

**Install**

sudo apt-get install -y kubectl

**Verify if kubectl got installed**

kubectl version –client

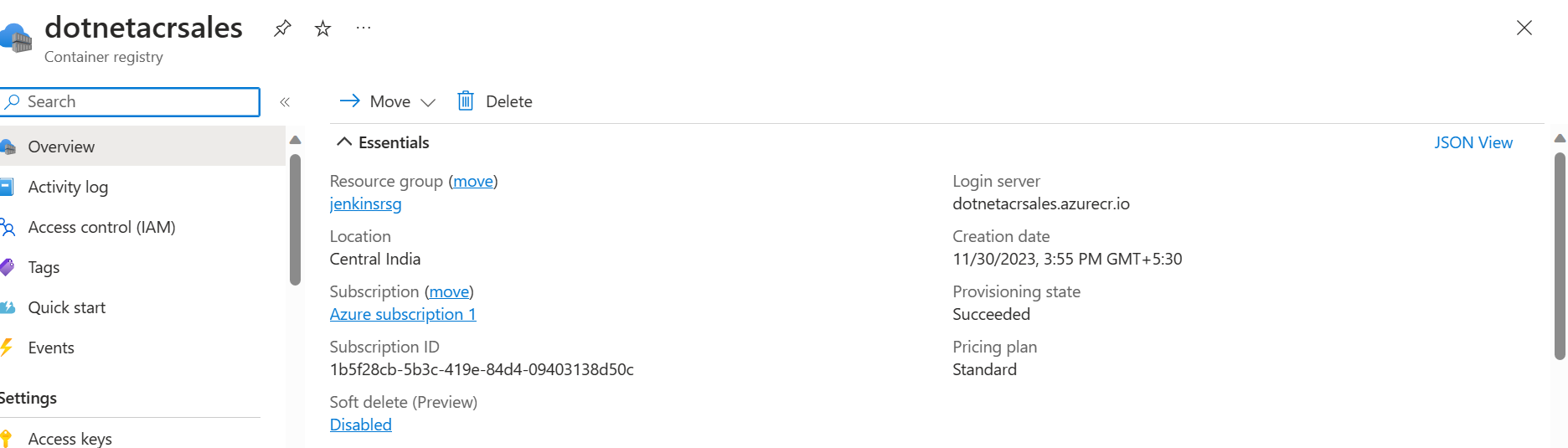


**Step 4: AKS Cluster creation and ACR on Azure cloud account**

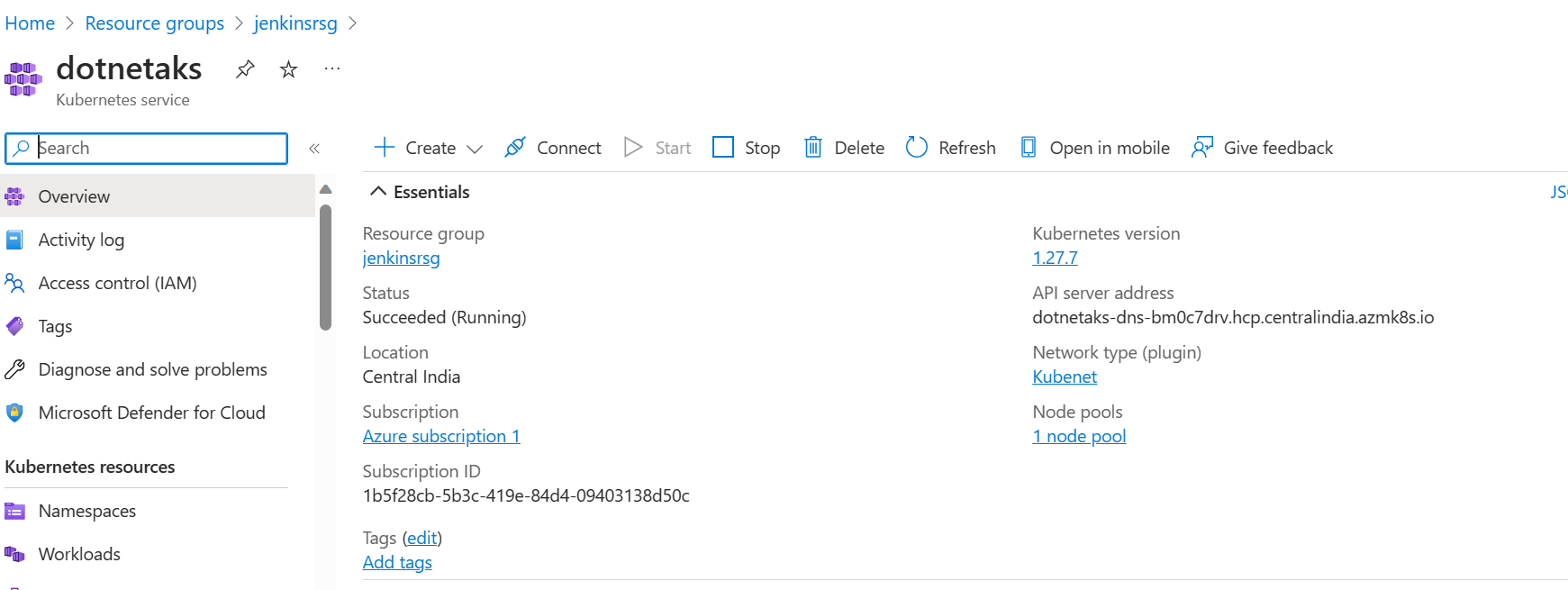
**Step 6: Attach the aks cluster with ACR Register**

az aks update -n aksclustername -g resourcegroup --attach-acr acrname

**Step 5: Get the Kubeconfig get secret key**



Step 3: Create the AKS Cluster on Azure portal



--- Get credentials for AKS cluster

az aks get-credentials --name dotnetaks --resource-group jenkinsrsg

---Get credential in the local file :

cat C:\Users\jagan\.kube\config

Step 6: configure the Jenkins pipeline

**Jenkins pipeline for dotnetcore on AKS cluster**

pipeline {

agent any

environment {

registryUrl='http://dotnetacrsales.azurecr.io'

acrCredentials='dotnetacrforaks'

registerName='dotnetacrsales'

}

stages {

stage('Checkout ') {

steps {

git changelog: false, credentialsId: 'jaganrajagopalgithub', poll: false, url: 'https://github.com/jaganrajagopal/Jenkinswithdockercomposeup.git'

}

}

stage('docker build') {

steps {

script {

dockerImage=docker.build registerName

}

}

}

stage('docker push to ACR') {

steps {

script{

docker.withRegistry(registryUrl,acrCredentials){

dockerImage.push()

}

}

echo 'Pushing to ACR'

}

}

stage('Docker deploy to aks cluster') {

steps {

script{

withKubeConfig(caCertificate: '', clusterName: '', contextName: '', credentialsId: 'akscredentials', namespace: '', restrictKubeConfigAccess: false, serverUrl: '') {

// some block

sh 'kubectl apply -f manifests/deployment.yml'

sh 'kubectl apply -f manifests/service.yml'

}

}

echo 'Publish code'

}

}

}

}

After successful the deployment on the aks cluster

