

Name : leburu gokul
Official mail : gokul.leburu@bajajfinserv.in
Date : 05-feb-2022

Assignment - 1

1. Explain devops.Mention the devops stages and tools.

DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes. This speed enables organizations to better serve their customers and compete more effectively in the market.

Stages in devops:-

- 1) Plan
- 2) Code
- 3) Build
- 4) Test
- 5) Integrate
- 6) Deploy
- 7) Operate
- 8) Monitor

Tools used in devops:-

Git and Jira are used in the plan and code phase.

Gradle and maven are used in building phase.

Selenium is used in Testing phase. Jenkins and

Bmboo are used to Integrate. Puppet,docker

chef are used in deploy and operate stages.

Splunk and nagios are used for monitoring

2. What is docker? explain the architecture & the components of docker.

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

Using Docker lets you ship code faster, standardize application operations, seamlessly move code, and save money by improving resource utilization. With Docker, you get a single object that can reliably run anywhere. Docker's simple and straightforward syntax gives you full control. Wide adoption means there's a robust ecosystem of tools and off-the-shelf applications that are ready to use with Docker.

Architecture of docker:-

(DOCKER CLIENT){build,run,pull}-->(SERVER){Docker daemon[containers] <--> Registry[Images]}

Components of docker include:-

1) Docker client and server:-

Docker Client is accessed from the terminal and a Docker Host runs the Docker Daemon and registry. A user can build Docker Images and run Docker Containers by passing commands from the Docker Client to Docker server.

2) Docker image:-

Docker Image is a template with instructions, which is used for creating Docker Containers.

3) Docker container:-

Docker Container is a standalone, executable software package which includes applications and their dependencies.

4) Docker registry:-

It is a server side service used for hosting and distributing images

3. What is kubernetes? Explain the features of kubernetes.

You can cluster together groups of hosts running Linux® containers, and Kubernetes helps you easily and efficiently manage those clusters.

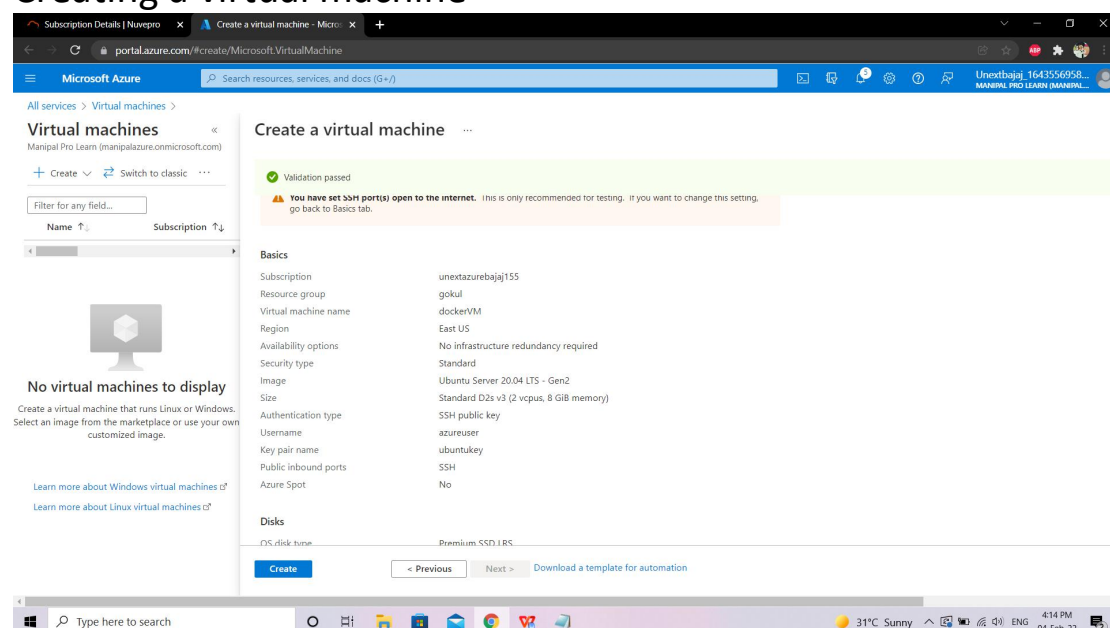
Kubernetes clusters can span hosts across on-premise, public, private, or hybrid clouds. For this reason, Kubernetes is an ideal platform for hosting cloud-native applications that require rapid scaling, like real-time data streaming through Apache Kafka

Features of kuberentes:-

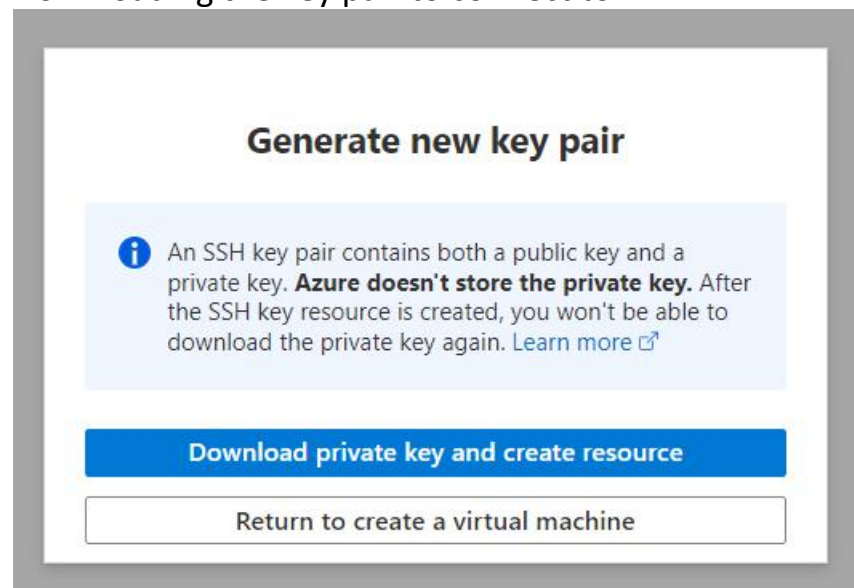
- 1) Automatic binpacking
- 2) Load balancing
- 3) Service discovery
- 4) Storage orchestration
- 5) Self healing
- 6) Batch execution
- 7) Horizontal scaling
- 8) Secret and Configuration management
- 9) Automatic rollbacks and rollouts.

4.Demonstrate docker & kubernetes with screenshots.

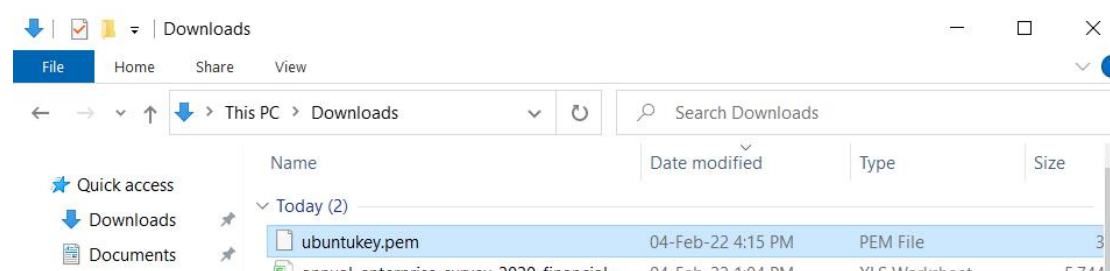
Creating a virtual machine



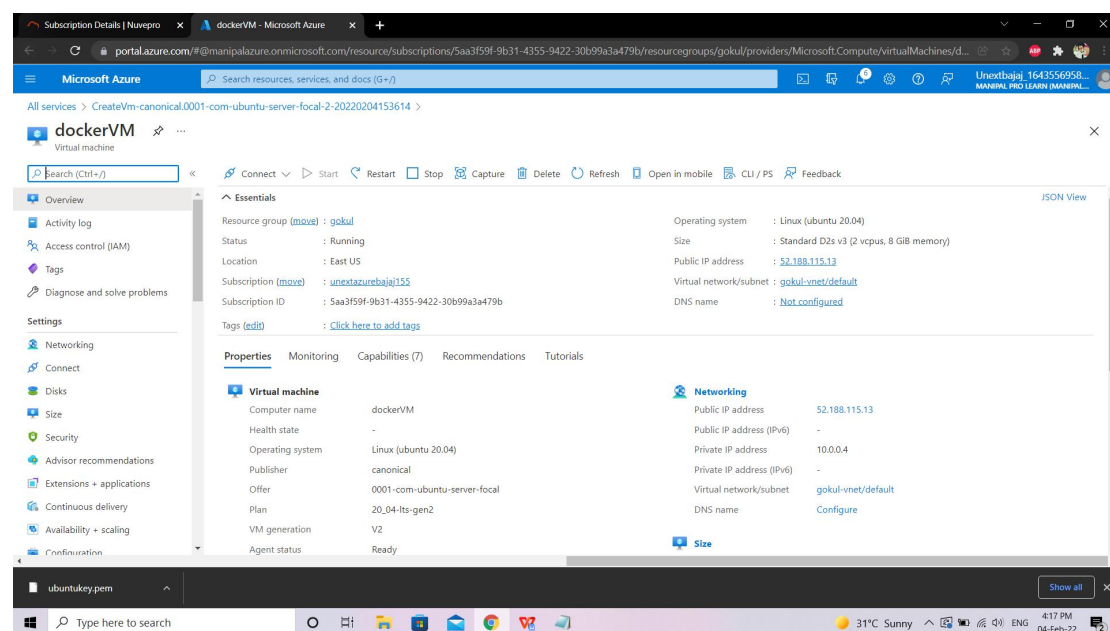
Downloading the key pair to connect to VM



Location of file to be remembered for future reference



Successfully deployed a VM in azure



Connecting to VM via SSH

Microsoft Azure

All services > CreateVm-canonical.0001-com-ubuntu-server-focal-2-20220204153614 > dockerVM

dockerVM | Connect

Virtual machine

Search (Ctrl+/)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Networking

Connect

Disks

Size

Security

Advisor recommendations

Extensions + applications

Continuous delivery

Availability + scaling

To improve security, enable just-in-time access on this VM. →

RDP SSH Bastion

Connect via SSH with client

1. Open the client of your choice, e.g. PuTTY or other clients.
2. Ensure you have read-only access to the private key.
`chmod 400 azureuser.pem`
3. Provide a path to your SSH private key file. ⓘ
Private key path
`~/.ssh/azureuser`
4. Run the example command below to connect to your VM.
`ssh -i <private key path> azureuser@52.188.115.13`

Can't connect?

Test your connection

Troubleshoot SSH connectivity issues

Connection and dockers commands:

```
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\gokul>ssh -i C:\Users\gokul\Downloads\ubuntukey.pem azureuser@52.188.115.13

Warning: Permanently added '52.188.115.13' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1028-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Fri Feb  4 10:49:30 UTC 2022

System load:  0.12               Processes:    130
Usage of /:   4.8% of 28.90GB    Users logged in:  0
Memory usage: 3%                IPv4 address for eth0: 10.0.0.4
Swap usage:  0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@dockerVM:~$
```


azure.txt - Notepad

File Edit Format View Help

```
ssh -i C:\Users\gokul\Downloads\dockerVM_key.pem azureuser@20.106.156.125
```

sudo apt-get update - is used to download package information

sudo apt install docker.io - to install dockers in VM

docker --version - to know the version of the docker app

sudo docker run hello-world - to get helloworld container

sudo docker images - to view which docker images are present

sudo docker ps -a - to check what containers are present

```
azureuser@dockerVM: ~
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azureuser@dockerVM:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1550 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [300 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [14.7 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [896 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [197 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [20.0 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [24.8 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [6928 B]
Get:19 http://azure.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [620 B]
Get:20 http://azure.archive.ubuntu.com/ubuntu focal-backports/main amd64 Packages [42.0 kB]
Get:21 http://azure.archive.ubuntu.com/ubuntu focal-backports/main Translation-en [10.0 kB]
Get:22 http://azure.archive.ubuntu.com/ubuntu focal-backports/main amd64 c-n-f Metadata [864 B]
Get:23 http://azure.archive.ubuntu.com/ubuntu focal-backports/restricted amd64 c-n-f Metadata [116 B]
Get:24 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe amd64 Packages [20.8 kB]
Get:25 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe Translation-en [14.3 kB]
Get:26 http://azure.archive.ubuntu.com/ubuntu focal-backports/universe amd64 c-n-f Metadata [692 B]
Get:27 http://azure.archive.ubuntu.com/ubuntu focal-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:28 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [1210 kB]
Get:29 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [213 kB]
Get:30 http://security.ubuntu.com/ubuntu focal-security/main amd64 c-n-f Metadata [9136 B]
Get:31 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [718 kB]
Get:32 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [103 kB]
Get:33 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [676 kB]
Get:34 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [115 kB]
Get:35 http://security.ubuntu.com/ubuntu focal-security/universe amd64 c-n-f Metadata [13.0 kB]
Get:36 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [21.8 kB]
Get:37 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [4948 B]
Get:38 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 c-n-f Metadata [536 B]
Fetched 20.8 MB in 3s (6062 kB/s)
```

```

azureuser@dockerVM:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 10 not upgraded.
Need to get 74.5 MB of archives.
After this operation, 361 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu focal/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu focal/main amd64 bridge-utils amd64 1.6-2ubuntu1 [30.5 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 runc amd64 1.0.1-0ubuntu2~20.04.1 [4155 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 containerd amd64 1.5.5-0ubuntu3~20.04.1 [33.0 MB]
Get:5 http://azure.archive.ubuntu.com/ubuntu focal/main amd64 dns-root-data all 2019052802 [5300 B]
Get:6 http://azure.archive.ubuntu.com/ubuntu focal/main amd64 libidn11 amd64 1.33-2.2ubuntu2 [46.2 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 dnsmasq-base amd64 2.80-1.1ubuntu1.4 [315 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu focal-updates/universe amd64 docker.io amd64 20.10.7-0ubuntu5~20.04.2 [36.9 MB]
Get:9 http://azure.archive.ubuntu.com/ubuntu focal/main amd64 ubuntu-fan all 0.12.13 [34.5 kB]
Fetched 74.5 MB in 2s (30.9 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 90%

```

```

Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
azureuser@dockerVM:~$ docker --version
Docker version 20.10.7, build 20.10.7-0ubuntu5~20.04.2
azureuser@dockerVM:~$

```

```

azureuser@dockerVM:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:507ecde44b8eb741278274653120c2bf793b174c06ff4eaa672b713b3263477b
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

azureuser@dockerVM:~$

```



```

azureuser@dockerVM:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  4 months ago  13.3kB
azureuser@dockerVM:~$ sudo docker ps -a
CONTAINER ID   IMAGE        COMMAND        CREATED        STATUS        PORTS        NAMES
70ea31f91ef3   hello-world  "/hello"       34 seconds ago Exited (0) 33 seconds ago          condescending_jennings
azureuser@dockerVM:~$

```

Kubernetes:

Microsoft Azure

Home > Kubernetes services >

Kubernetes services

Manipal Pro Learn (manipalazure@microsoft.com)

+ Create Manage view

Filter for any field...

Name ↑ Type ↑

No Kubernetes services to display

Use Azure Kubernetes Service to create and manage Kubernetes clusters. Azure will handle cluster operations, including creating, scaling, and upgrading, freeing up developers to focus on their application. To get started, create a cluster with Azure Kubernetes Service.

Learn more

Create Kubernetes cluster

Validation passed

Basics Node pools Authentication Networking Integrations Tags Review + create

Basics

Subscription unextazurebaj155

Resource group (new) gokul

Region East US

Kubernetes cluster name kcluster

Kubernetes version 1.21.7

Node pools

Node pools 1

Enable virtual nodes Disabled

Enable virtual machine scale sets Enabled

Authentication

Authentication method System-assigned managed identity

Role-based access control (RBAC) Enabled

AKS-managed Azure Active Directory Disabled

Create < Previous Next > Download a template for automation

Home >

microsoft.aks-20220204213441 | Overview

Deployment

Search (Ctrl+ /)

Delete Cancel Redeploy Refresh

Overview

Inputs

Outputs

Template

We'd love your feedback! →

✓ Your deployment is complete

Deployment name: microsoft.aks-20220204213441

Subscription: unextazurebaj155

Resource group: gokul

Start time: 2/4/2022, 9:36:55 PM

Correlation ID: 1ebd5e5e-85e7-48b

Deployment details (Download)

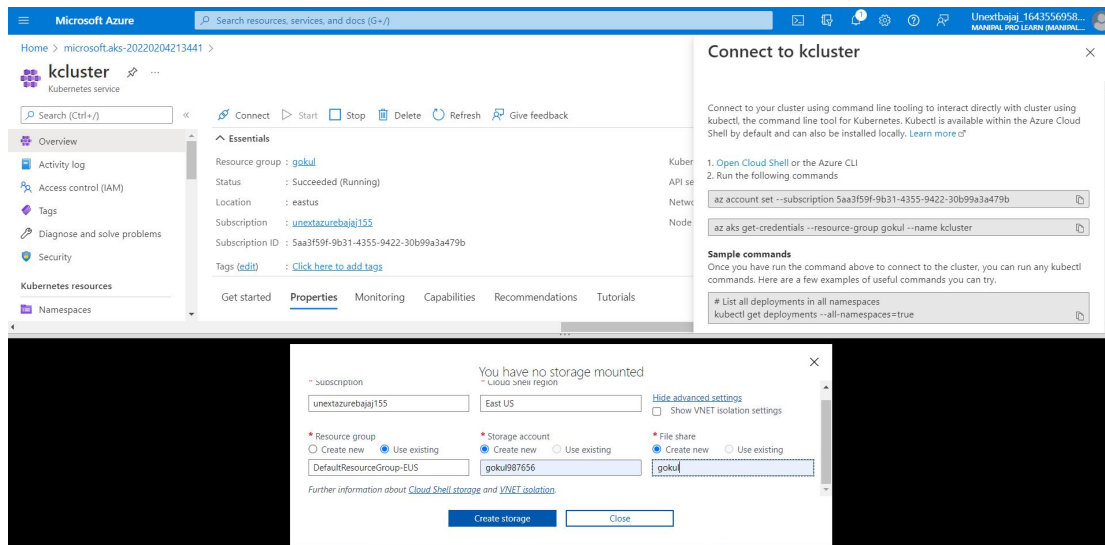
Next steps

Create a Kubernetes deployment Recommended

Integrate automatic deployments within your cluster Recommended

Connect to cluster Recommended

Go to resource Connect to cluster



Commands:

kubernetes:

```
kubectl - is used to deploy applications
az aks get-credentials - resource-group rscgrp --name AKSculster
kubectl get nodes - it show the acitve nodes information
vi azure-vote.yaml - to open and edit a file to run the application
kubectl apply -f azure-vote --watch - we can create a deployment using it
kubectl get service azure-vote-front --watch - to get the website address or we can say the info of deployment of the file
```

```
Bash
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

unextbajaj_1643556958007@Azure:~$ az account set --subscription 5aa3f59f-9b31-4355-9422-30b99a3a479b
unextbajaj_1643556958007@Azure:~$ az aks get-credentials --resource-group gokul --name kcluster
Merged "kcluster" as current context in /home/unextbajaj_1643556958007/.kube/config
unextbajaj_1643556958007@Azure:~$
```

```
unextbajaj_1643556958007@Azure:~$ az aks get-credentials --resource-group gokul --name kcluster
Merged "kcluster" as current context in /home/unextbajaj_1643556958007/.kube/config
unextbajaj_1643556958007@Azure:~$ kubectl get nodes
NAME                                STATUS    ROLES    AGE    VERSION
aks-agentpool-72335525-vmss000000  Ready    agent    10m    v1.21.7
aks-agentpool-72335525-vmss000001  Ready    agent    10m    v1.21.7
aks-agentpool-72335525-vmss000002  Ready    agent    10m    v1.21.7
unextbajaj_1643556958007@Azure:~$
```

```
Microsoft Azure Search resources, services, and docs (G+/)

Bash

apiVersion: apps/v1
kind: Deployment
metadata:
  name: azure-vote-back
spec:
  replicas: 1
  selector:
    matchLabels:
      app: azure-vote-back
  template:
    metadata:
      labels:
        app: azure-vote-back
    spec:
      nodeSelector:
        "kubernetes.io/os": linux
      containers:
        - name: azure-vote-back
          image: mcr.microsoft.com/oss/bitnami/redis:6.0.8
          env:
            - name: ALLOW_EMPTY_PASSWORD
              value: "yes"
          resources:
            requests:
              cpu: 100m
              memory: 128Mi
            limits:
              cpu: 250m
              memory: 256Mi
          ports:
            - containerPort: 6379
              name: redis
---
apiVersion: v1
```

```
unextbajaj_1643556958007@Azure:~$ kubectl apply -f azure-vote.yaml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
unextbajaj_1643556958007@Azure:~$

unextbajaj_1643556958007@Azure:~$ kubectl get service azure-vote-front --watch
NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
azure-vote-front    LoadBalancer  10.0.56.128   <pending>      80:32688/TCP     32s
```

