Name: Mayank Tolani

Company Name: Bajaj Finserv

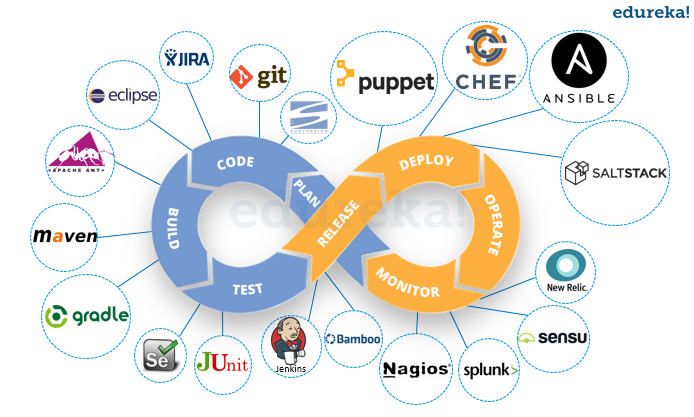
Data Track Batch 2

Date: 05-02-2022

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

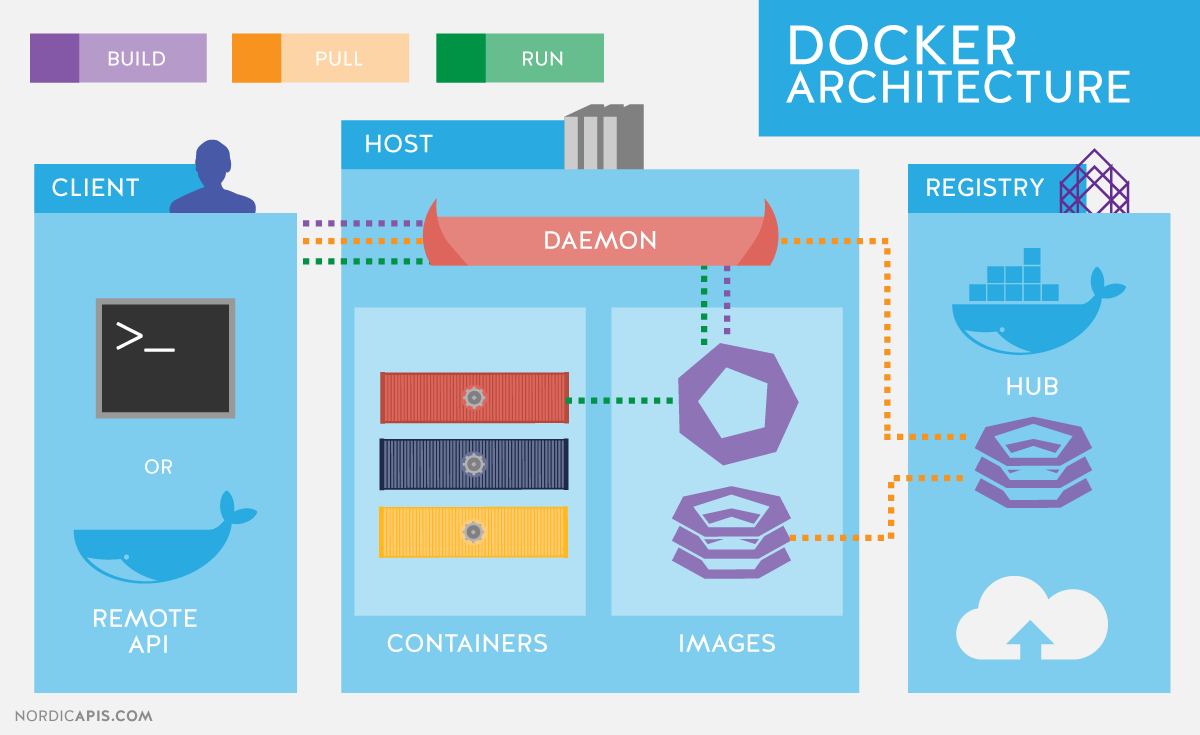
1. DevOps is the combination of cultural philosophies, practices, and tools that increases an organization’s ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management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

Below diagram shows all the devops stages and tools



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

2. Docker is an open-source project for automating the deployment of applications as portable, self-sufficient containers that can run on the cloud or on-premises. Docker is also a company that promotes and evolves this technology, working in collaboration with cloud, Linux, and Windows vendors, including Microsoft.



The Docker architecture uses a client-server model and comprises of the Docker Client, Docker Host, Network and Storage components, and the Docker Registry / Hub. Let’s look at each of these in some detail.

Docker Client

The Docker client enables users to interact with Docker. The Docker client can reside on the same host as the daemon or connect to a daemon on a remote host. A docker client can communicate with more than one daemon. The Docker client provides a command line interface (CLI) that allows you to issue build, run, and stop application commands to a Docker daemon.

DockerHost

The Docker host provides a complete environment to execute and run applications. It comprises of the Docker daemon, Images, Containers, Networks, and Storage. As previously mentioned, the daemon is responsible for all container-related actions and receives commands via the CLI or the REST API. It can also communicate with other daemons to manage its services. The Docker daemon pulls and builds container images as requested by the client. Once it pulls a requested image, it builds a working model for the container by utilizing a set of instructions known as a build file. The build file can also include instructions for the daemon to pre-load other components prior to running the container, or instructions to be sent to the local command line once the container is built.

Docker Registries

Docker registries are services that provide locations from where you can store and download images. In other words, a Docker registry contains Docker repositories that host one or more Docker Images. Public Registries include Docker Hub and Docker Cloud and private Registries can also be used. Common commands when working with registries include:

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

3. Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available. Containers are a good way to bundle and run your applications. In a production environment, you need to manage the containers that run the applications and ensure that there is no downtime. For example, if a container goes down, another container needs to start. Wouldn't it be easier if this behavior was handled by a system?

That's how Kubernetes comes to the rescue! Kubernetes provides you with a framework to run distributed systems resiliently. It takes care of scaling and failover for your application, provides deployment patterns, and more. For example, Kubernetes can easily manage a canary deployment for your system.

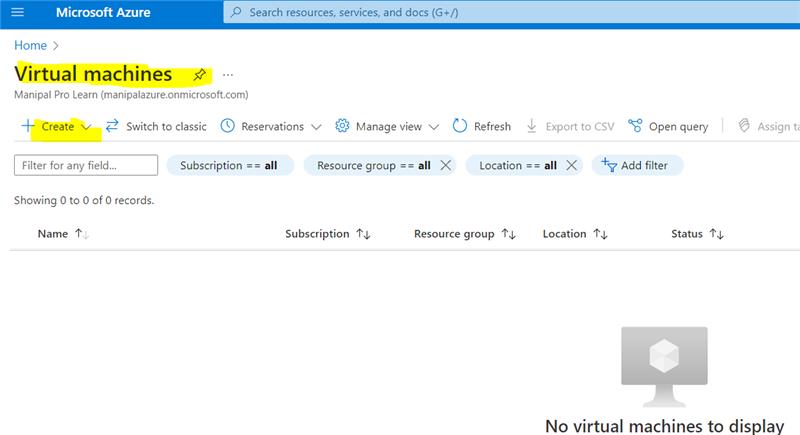
Features:

* Service discovery and load balancing Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
* Storage orchestration Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
* Automated rollouts and rollbacks You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
* Automatic bin packing You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.
* Self-healing Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
* Secret and configuration management Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

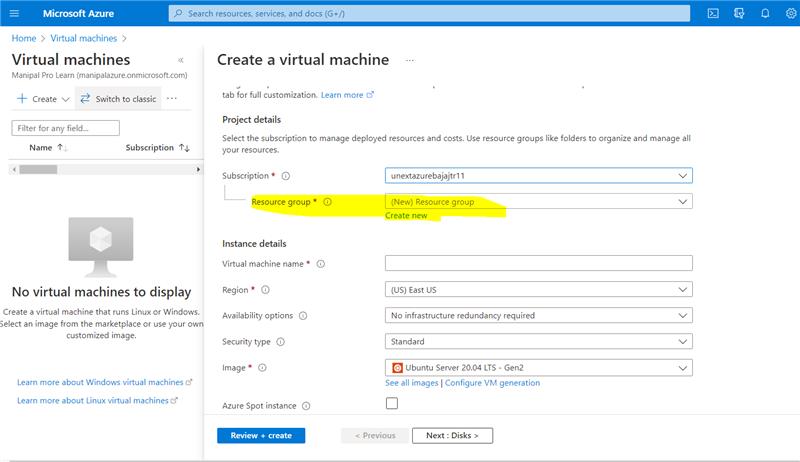
4. Demonstrate docker & kubernetes with screenshots.

Steps:

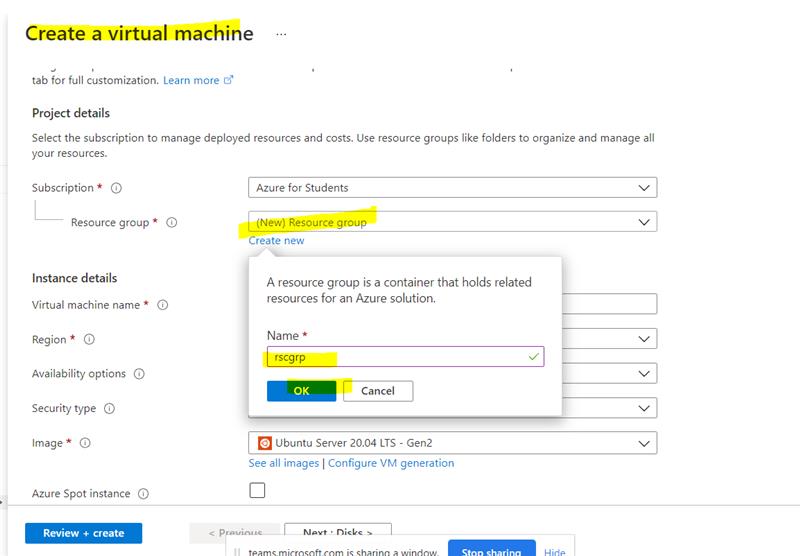
1. Create VM



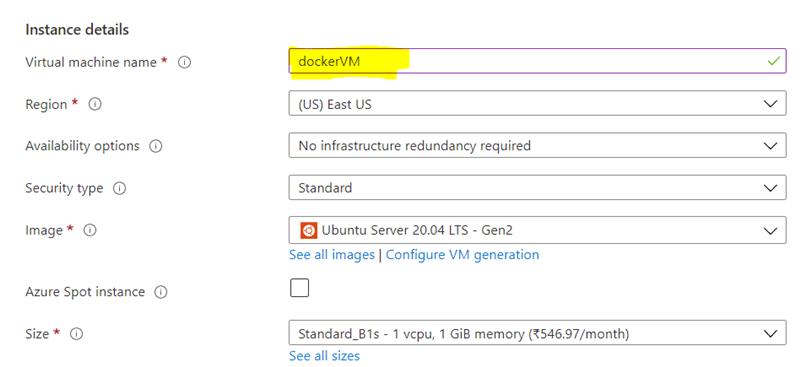
1. Resource group – Contains meta data of all azure services.



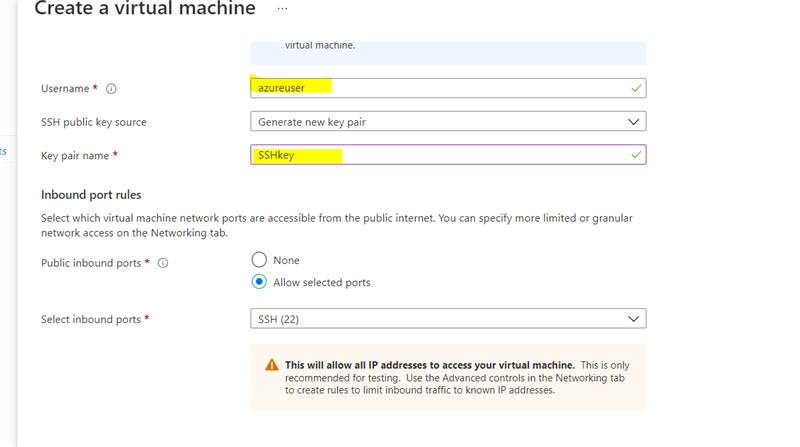
1. Give a name to your new resource group and click ok.



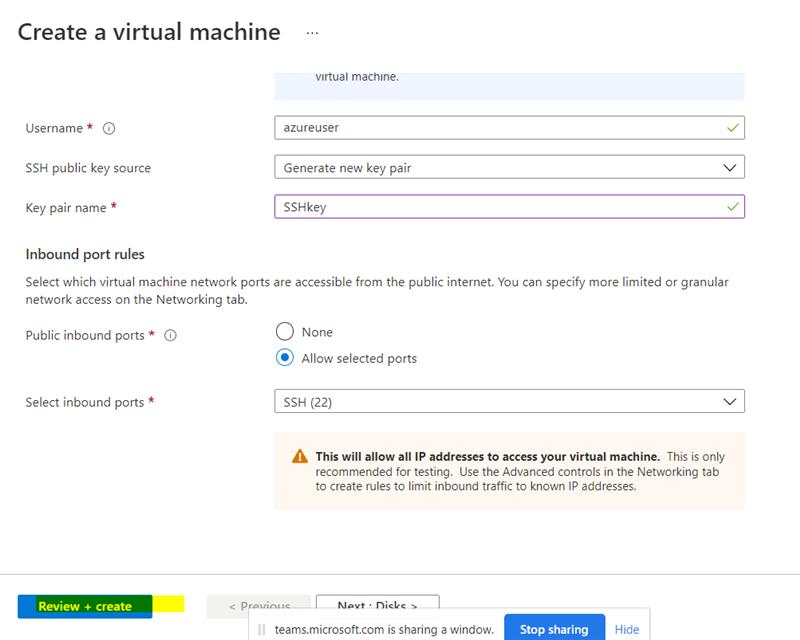
1. Give a name to your virtual machine. In this case, I’ve given dockerVM.



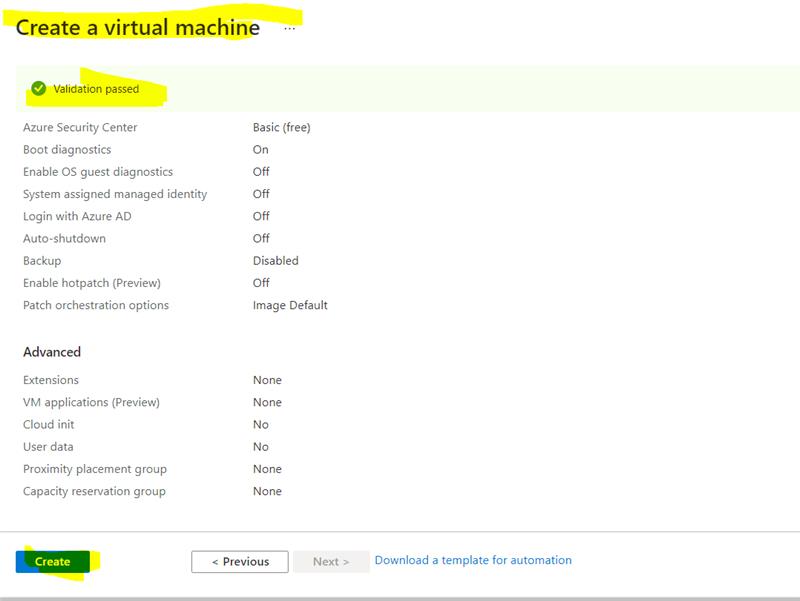
1. Give your username and the key pair name. The SSH key allows you to connect to the VM via your windows cmd in a secure way.



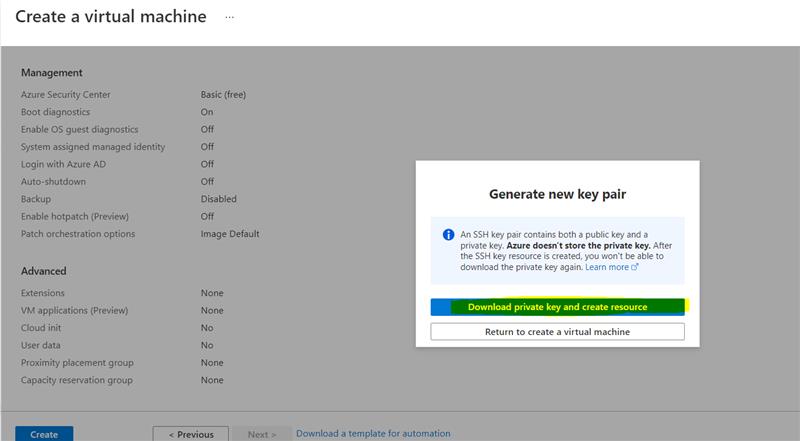
1. Click on review+create and proceed.



1. It will take some time to validate and then the validation will be passed. Finally, click create.

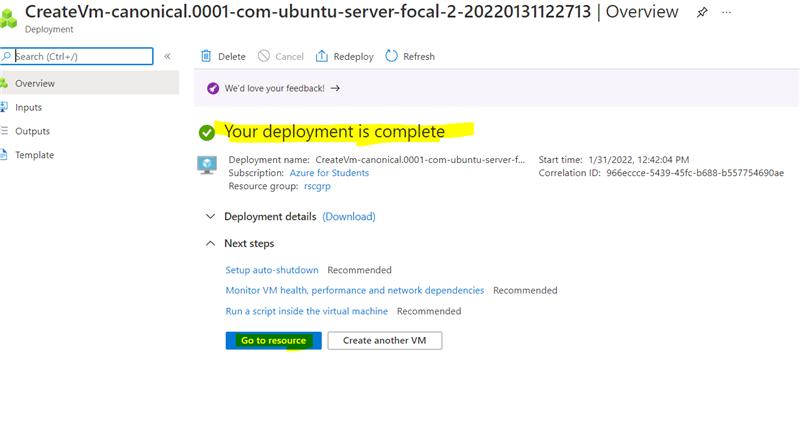


1. Download the key pair into your laptop. This file will be used to connect to the VM via the windows cmd.

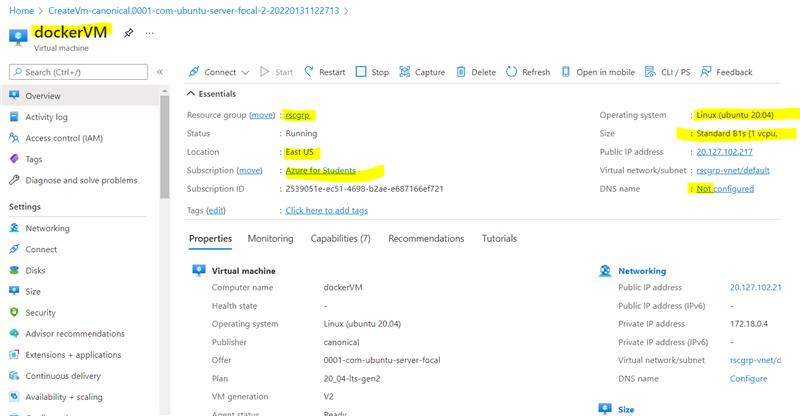


A SSHkey.pem file will be downloaded.

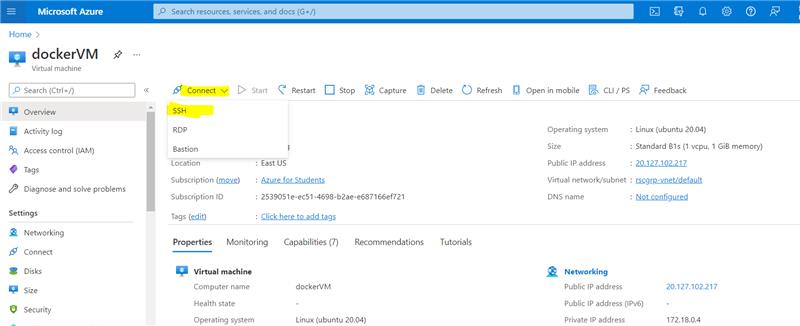
1. If everything goes well, your deployment will be completed successfully. Click on ‘Go to resource’.



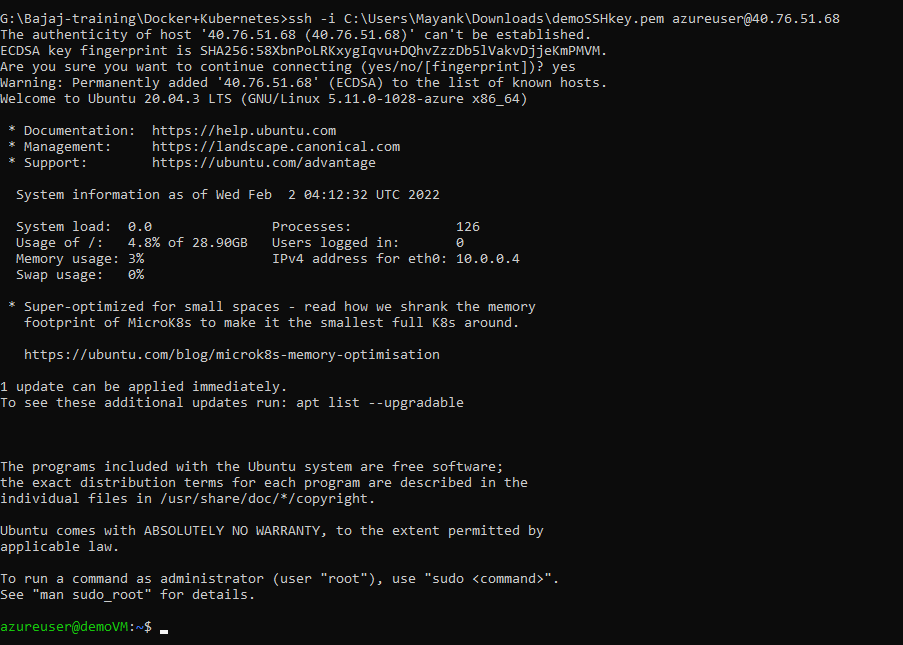
1. Overview of our VM



1. Click on connect and then select SSH. SSH stands for Secured Shell.

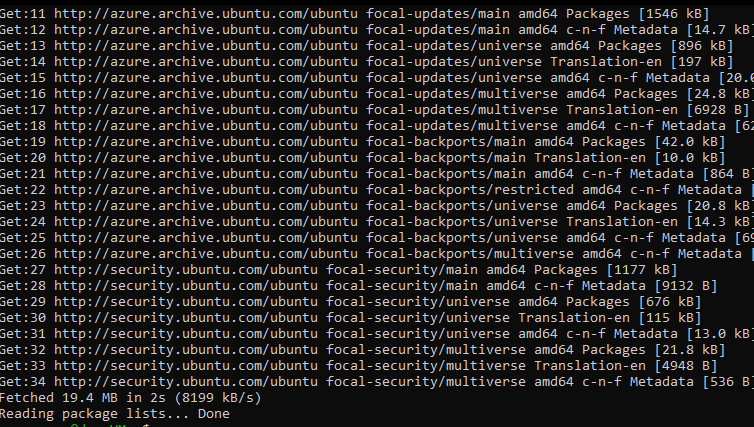


12. Connect to the VM - ssh -i C:\Users\Mayank\Downloads\demoSSHkey.pem azureuser@40.76.51.68



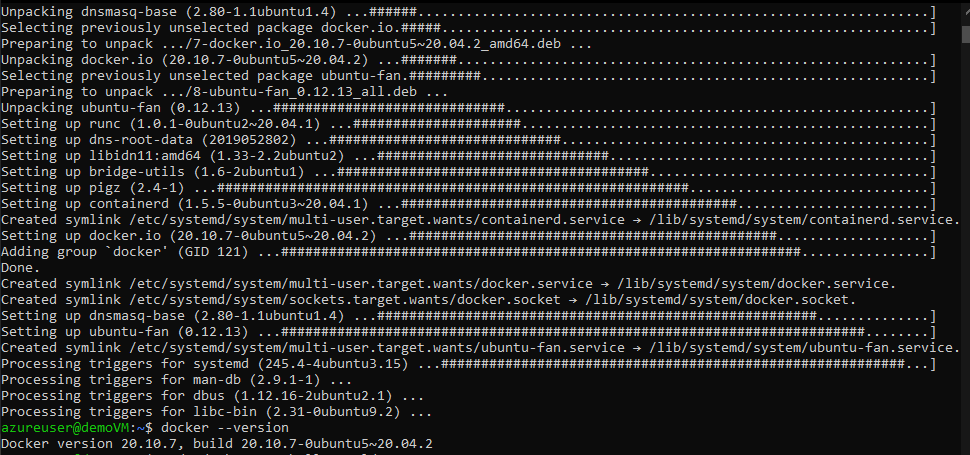
You are successfully connected to the VM.

13. sudo apt-get update – To download the package information from all configured sources.

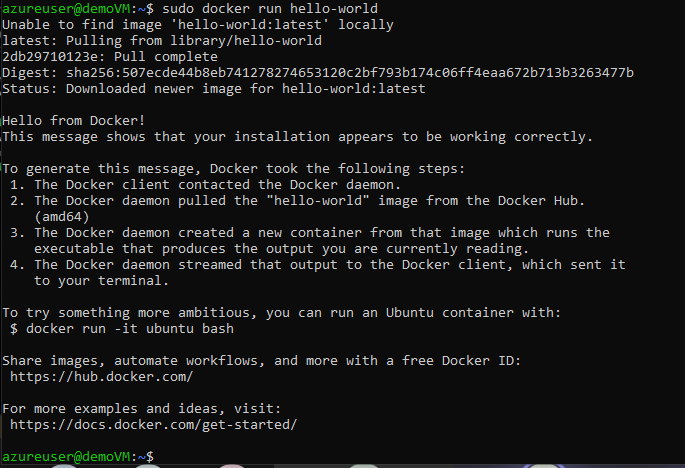


14. sudo apt install docker.io – Install docker via the terminal into your VM.

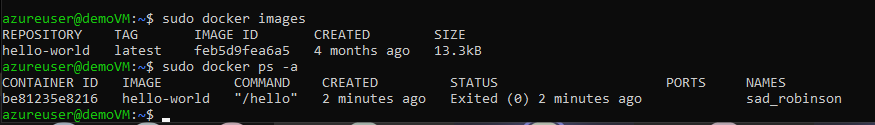
docker –version – To check the docker version. It will display the version if it was successfully installed.



15. sudo docker run hello-world – Running the hello-world docker image. If the image is not present on our system, it will be downloaded from the docker registry.

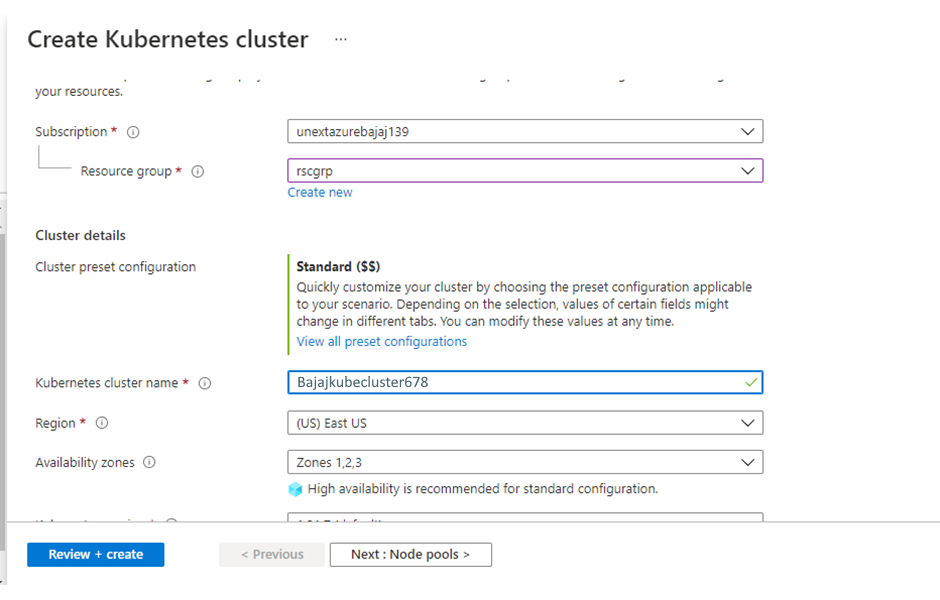


16. sudo docker images – shows installed docker images

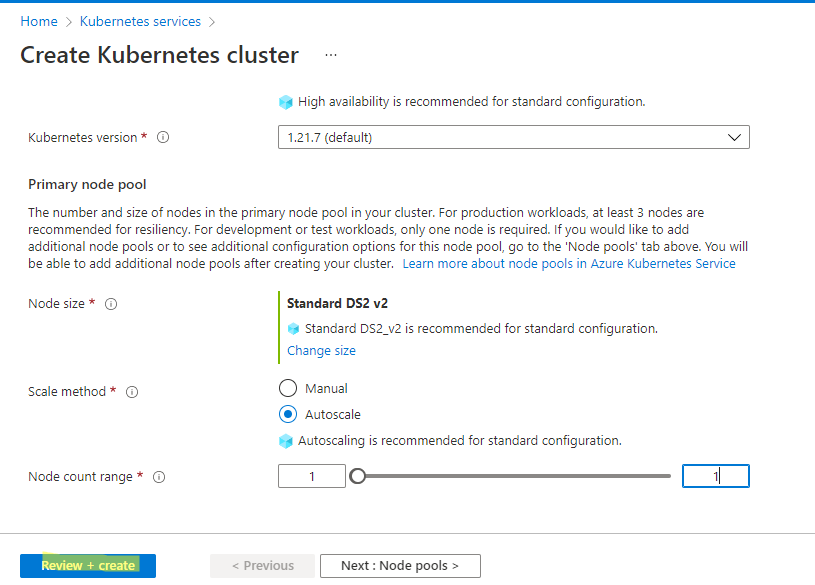


Kubernetes Steps:

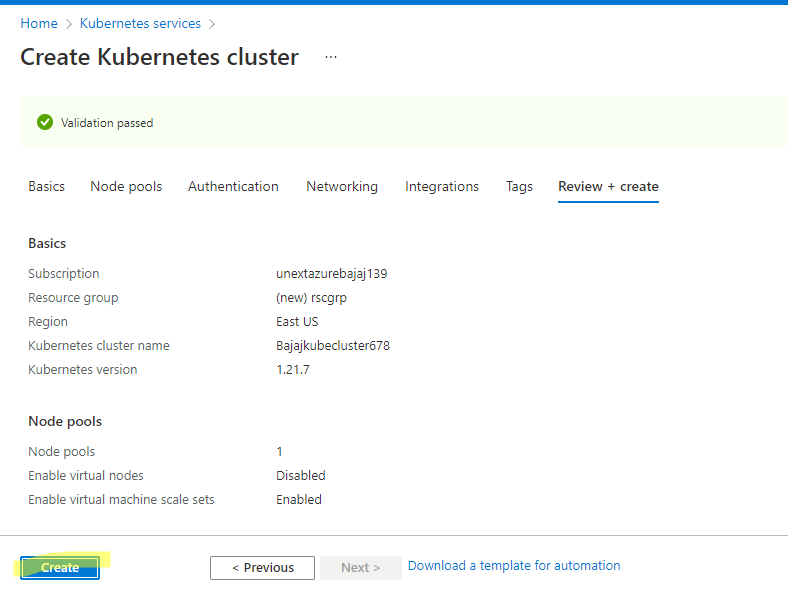
1. Creating a Kubernetes cluster – give the resource group name and the Kubernetes cluster name.



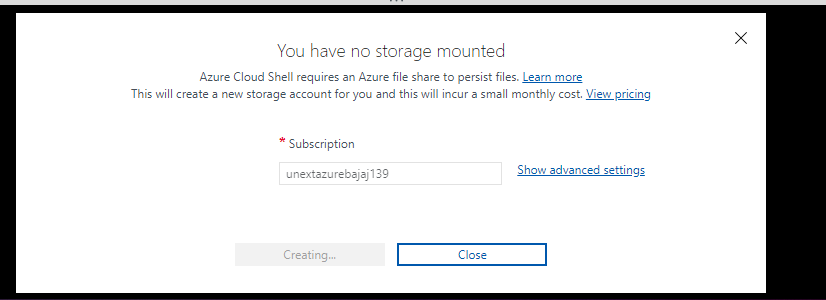
2. Give the node count range as 1. Click on review+create.



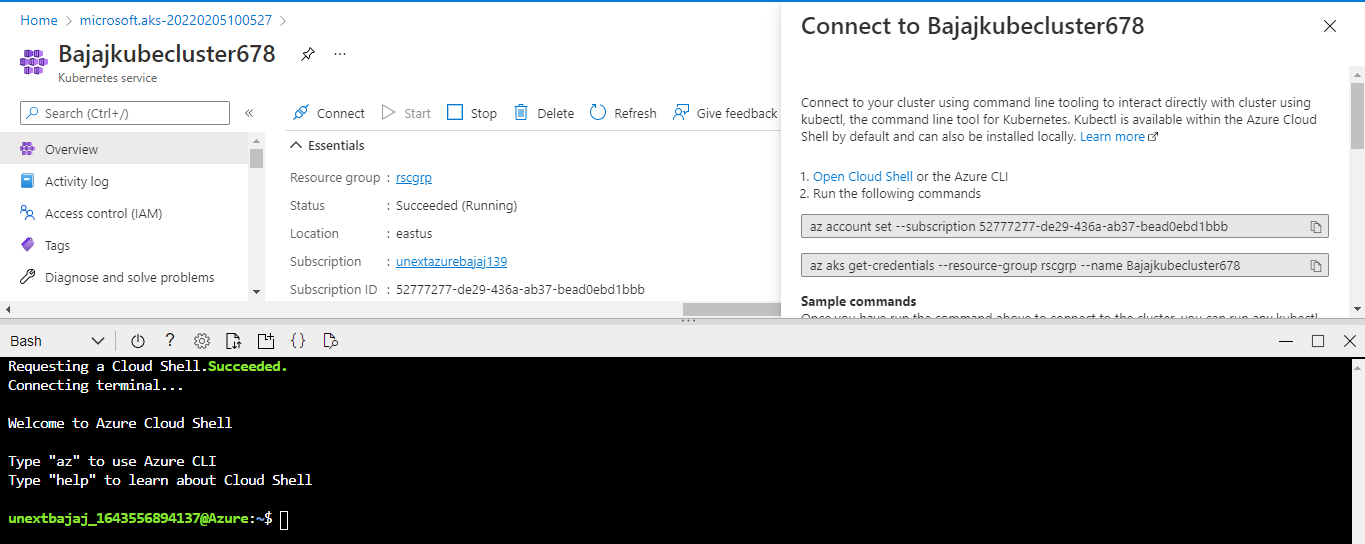
3. After validation is complete you may click on create to complete its creation.



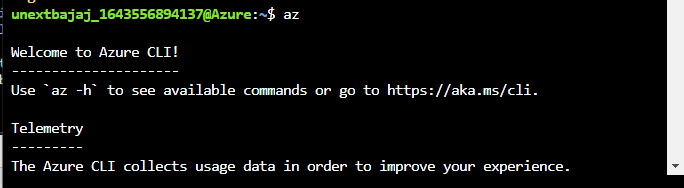
4. The deployment takes some time to complete, in the mean time we can proceed to create our storage account.



5. Open the cloud CLI

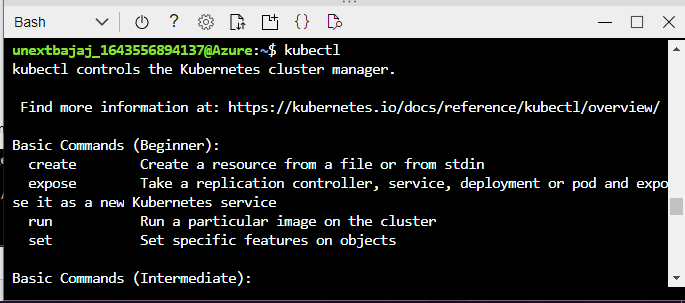


6. az – to open the azure CLI

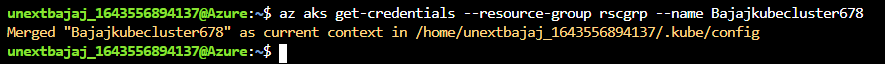


Note: Since the output was very long, some of it has been cropped.

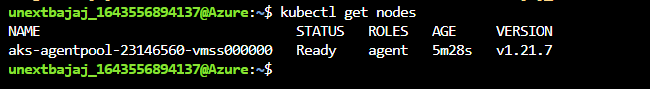
7. kubectl – kubectl is the Kubernetes command-line tool. It allows us to run commands against Kubernetes clusters — deploying applications, inspecting and managing cluster resources, and viewing logs.



8. az aks get-credentials --resource-group rscgrp --name AKScluster



9. kubectl get nodes – shows the nodes in the Kubernetes cluster



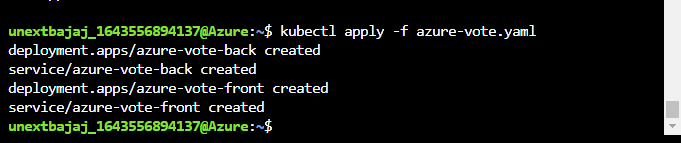
So far, all the commands were for connecting to the cluster.

10. vi azure-vote.yaml – To create a file named azure-vote.yaml and open it in the command line editor.

Paste the below code in the file. First press i to enter into insert mode then paste the text and press esc. :w – to write to the file. :x – To exit.

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  
kind: Service  
metadata:  
name: azure-vote-back  
spec:  
ports:  
- port: 6379  
selector:  
app: azure-vote-back  
---  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
name: azure-vote-front  
spec:  
replicas: 1  
selector:  
matchLabels:  
app: azure-vote-front  
template:  
metadata:  
labels:  
app: azure-vote-front  
spec:  
nodeSelector:  
"kubernetes.io/os": linux  
containers:  
- name: azure-vote-front  
image: mcr.microsoft.com/azuredocs/azure-vote-front:v1  
resources:  
requests:  
cpu: 100m  
memory: 128Mi  
limits:  
cpu: 250m  
memory: 256Mi  
ports:  
- containerPort: 80  
env:  
- name: REDIS  
value: "azure-vote-back"  
---  
apiVersion: v1  
kind: Service  
metadata:  
name: azure-vote-front  
spec:  
type: LoadBalancer  
ports:  
- port: 80  
selector:  
app: azure-vote-front

1. kubectl apply -f azure-vote.yaml



1. kubectl get service azure-vote-front –watch – To watch the service live on the internet using the external IP.

Open the web browser and type the external ip address of your service

