Name: Mayank Tolani

Company Name: Bajaj Finserv

Data Track Batch 2

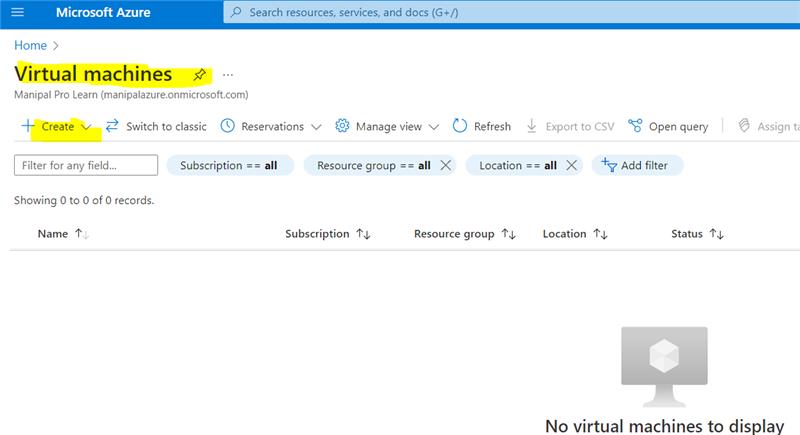
Date: 05-02-2022

Case Study - 5th Feb Dec 2021:  
----------------------------------------  
Global-tech incorporation is leading Biotech & Medical distribution  
company, has decided to migrate their data warehouse (around volume  
of 300TB uncompressed) to Cloud. Also, this organization has decided to  
migrate all downstream applications to Azure. Since its COVID –  
pandemic situation, hence its critical time & ETA is very less, the whole  
migration had to happen seamlessly, Using Azure cloud Service – we  
have to develop solutions for Global-tech. and migration activity to be  
performed.  
  
Cases: -

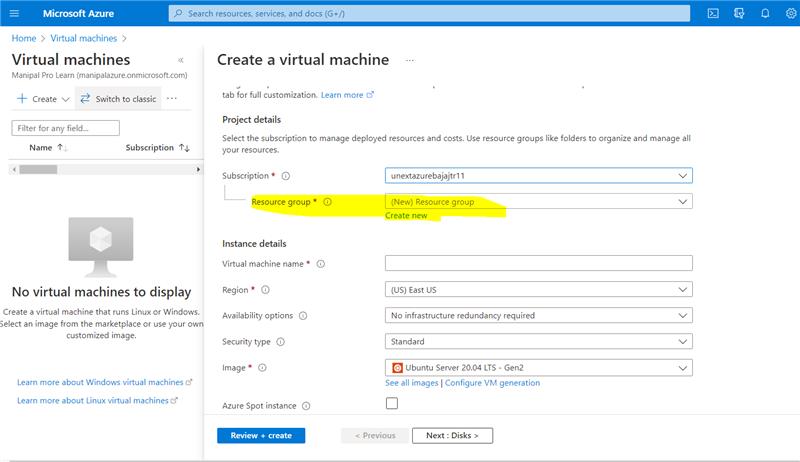
Q1. Create new Azure VM Instance (instance name: **globetechvm231**) & install docker into it,  
explain all steps with screenshots.

Steps:

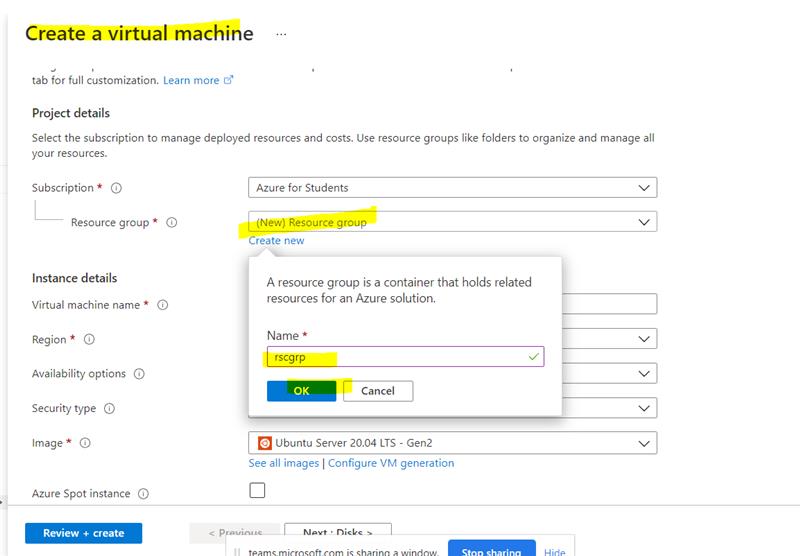
1. Create VM



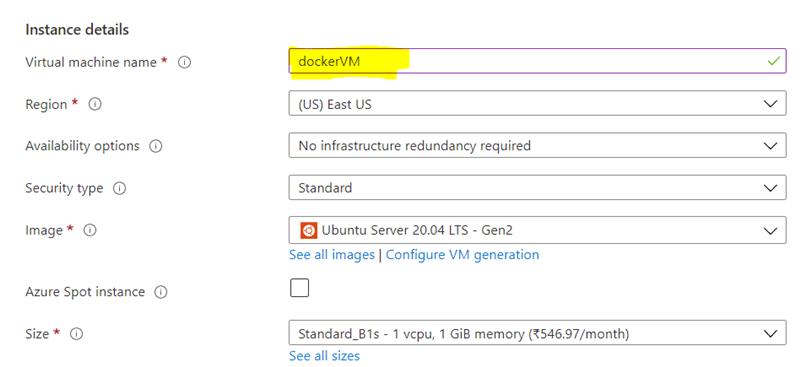
1. Resource group – Contains meta data of all azure services.



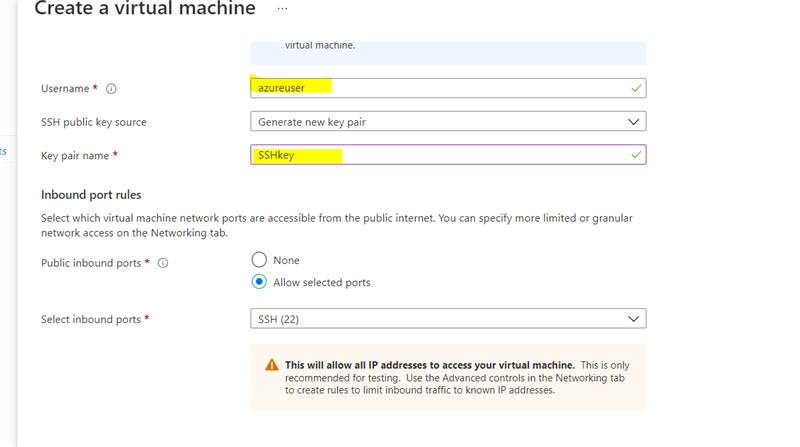
1. Give a name to your new resource group and click ok.



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



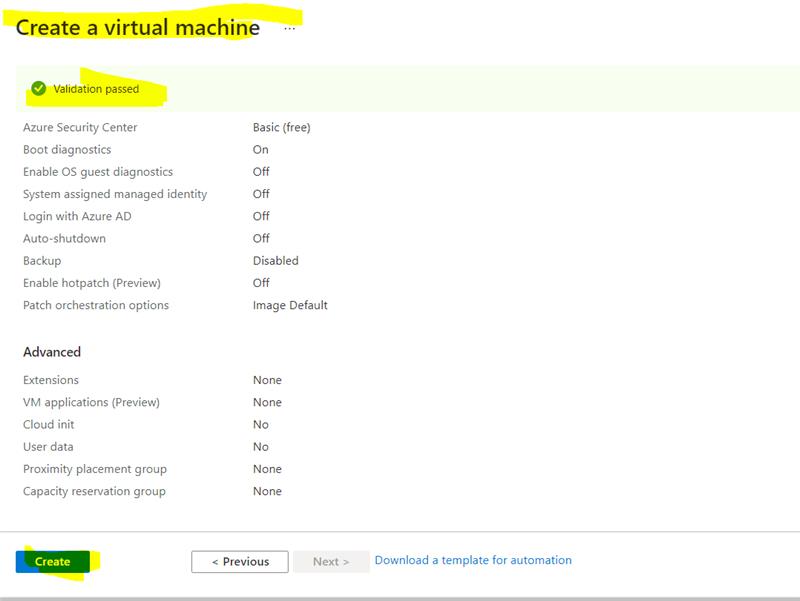
1. Give your username and the key pair name. The SSH key allows you to connect to the VM via your windows cmd in a secure way.



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.

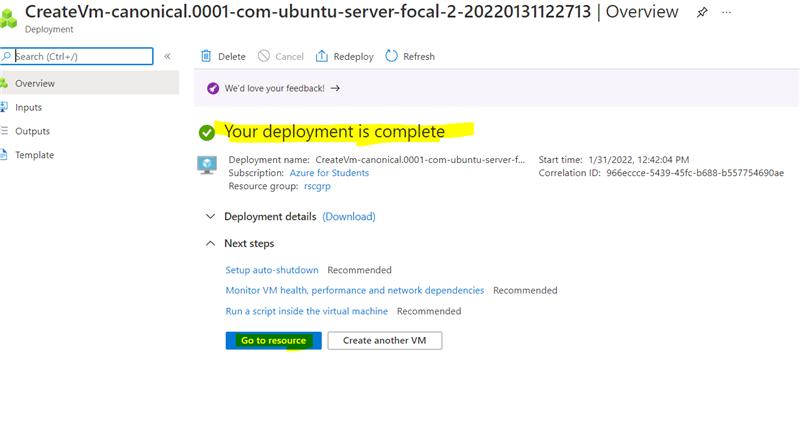


1. Download the key pair into your laptop. This file will be used to connect to the VM via the windows cmd.

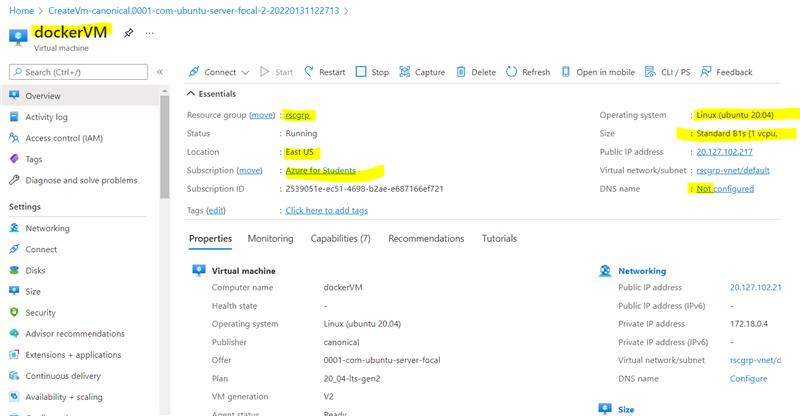


A SSHkey.pem file will be downloaded.

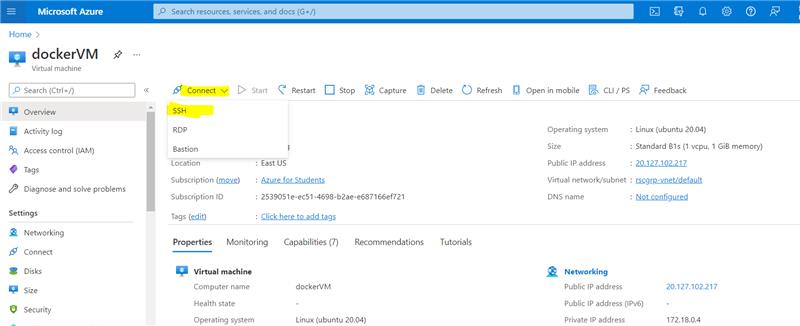
1. If everything goes well, your deployment will be completed successfully. Click on ‘Go to resource’.



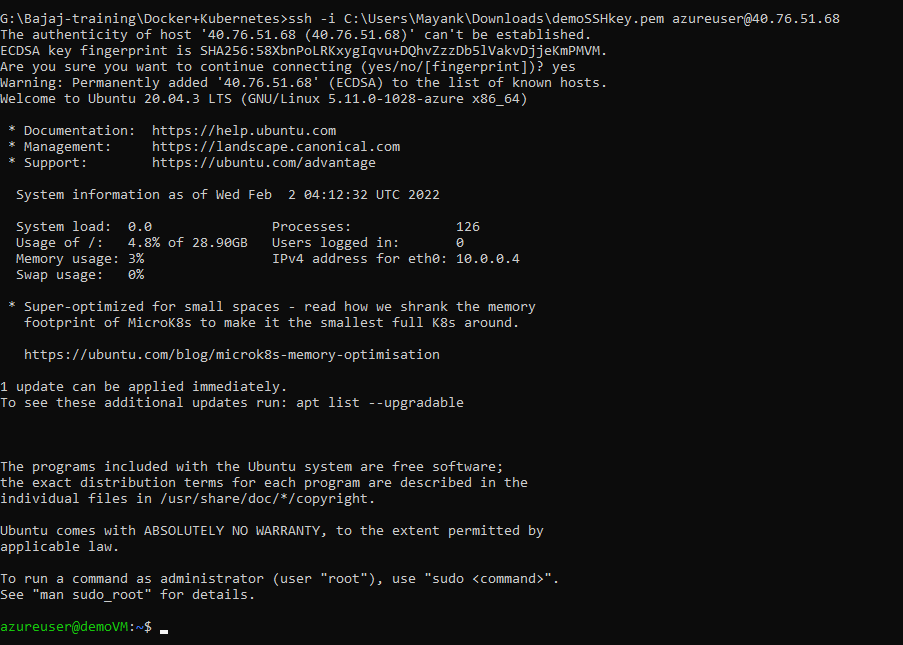
1. Overview of our VM



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

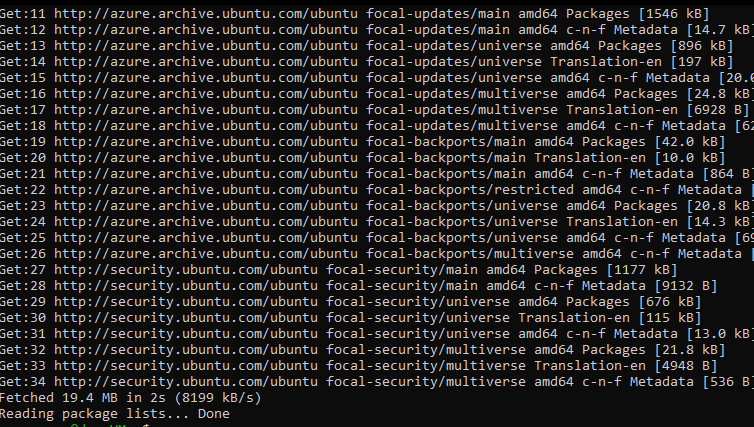


You are successfully connected to the VM.

2. Explain with screenshots - how Docker images will be installed in a VM & explain various features of Docker?

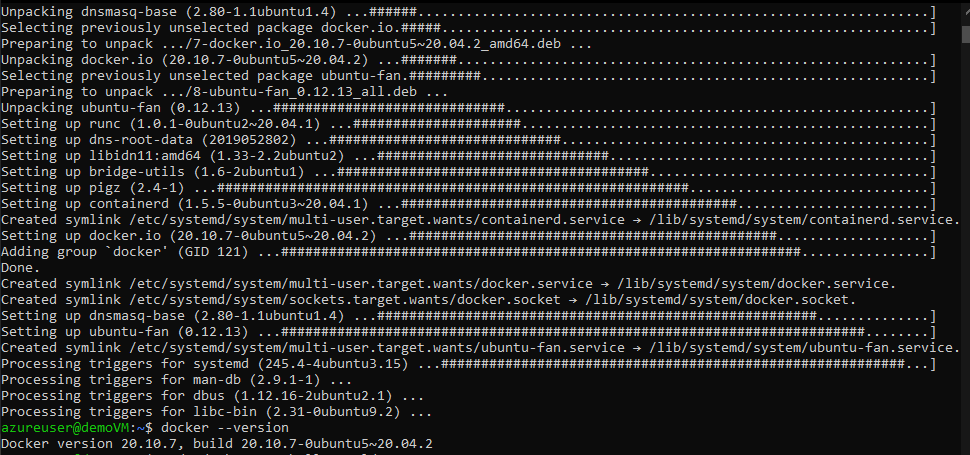
Steps:

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

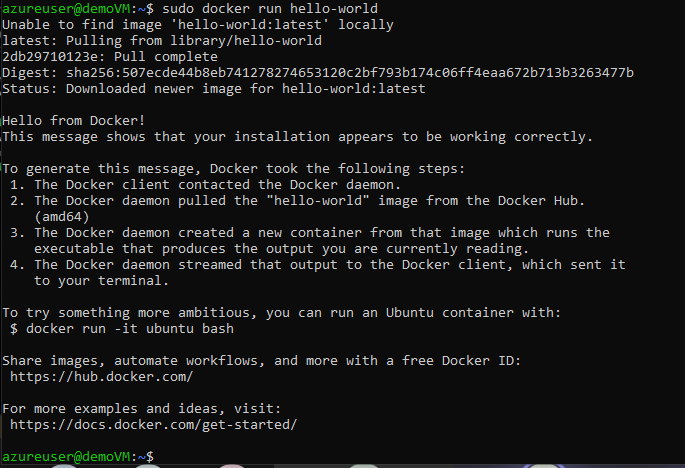


2. sudo apt install docker.io – Install docker via the terminal into your VM.

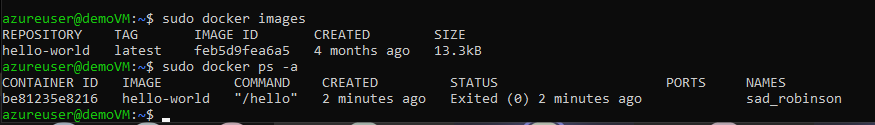
docker –version – To check the docker version. It will display the version if it was successfully installed.



3. 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.



4. sudo docker images – shows installed docker images



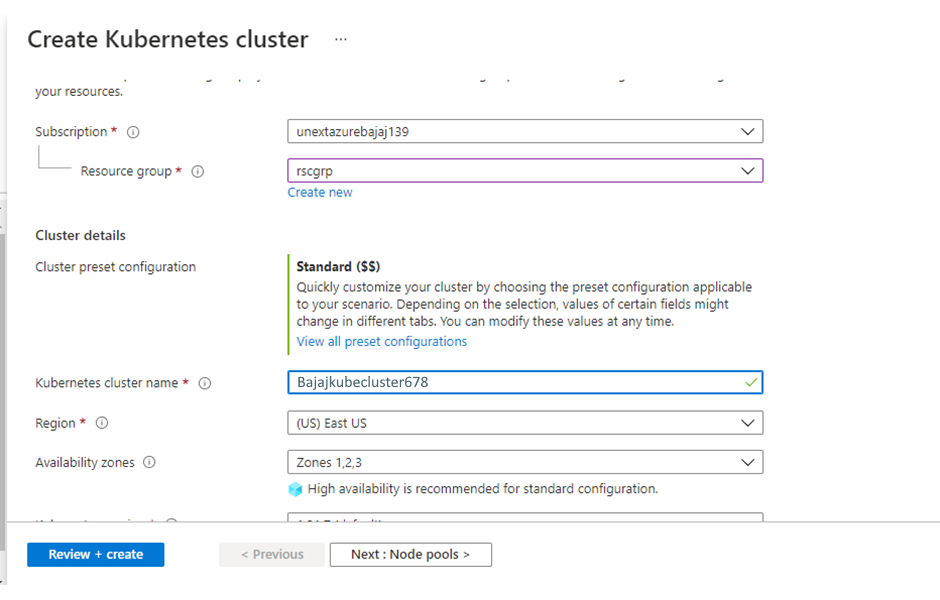
Features of docker:

* **Fast and Easy to configure:** It is one of the key features of Docker that helps you in configuring the system in a faster and easier manner. Due to this feature, codes can be deployed in less time and with fewer efforts.
* **Application isolation:** Docker provides containers that are used to run applications in an isolated environment. Since each container is independent, Docker can execute any kind of application.
* **Swarm:** Swarm is a clustering and scheduling tool for Docker containers. At the front end, it uses the Docker API, which helps us to use various tools to control it. It is a self-organizing group of engines that enables pluggable backends.
* **Security Management:** It saves secrets into the swarm and chooses to give services access to certain secrets, including a few important commands to the engine such as secret inspect, secret create, etc.
* **Better Software Delivery:** Software Delivery with the help of containers is said to be more efficient. Containers are portable, self-contained and include an isolated disk volume. This isolated volume goes along with the container as it develops and is deployed to various environments.

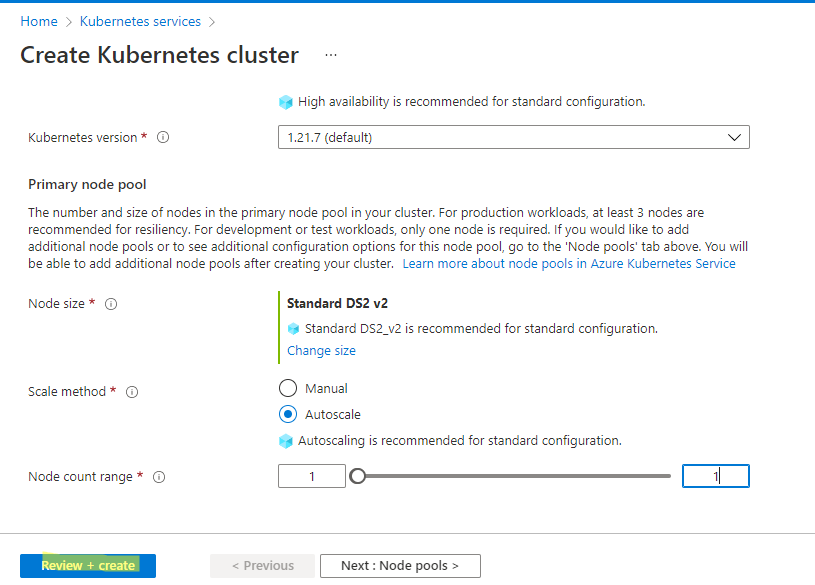
3. In Azure - please create Kubernetes cluster called: Bajajkubecluster678 & Create a sample Voting app & explain all steps with screenshots.

Steps:

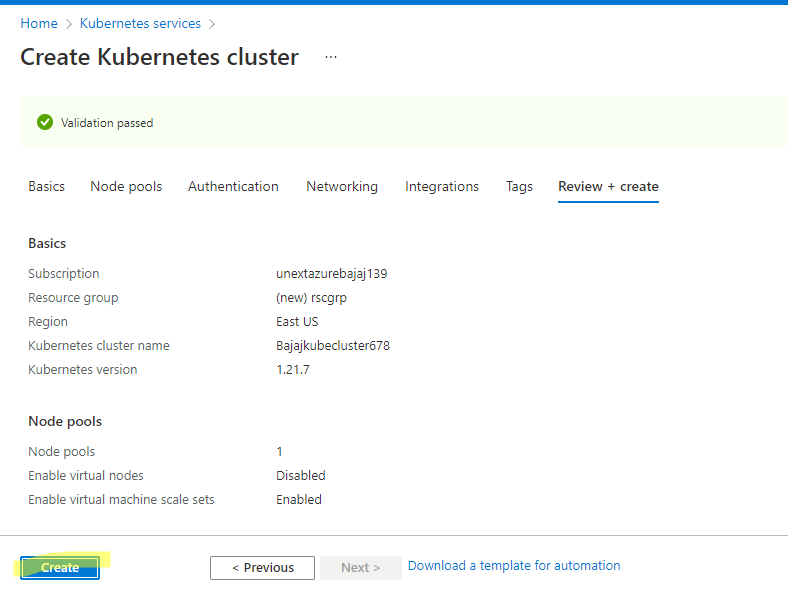
1. Creating a Kubernetes cluster – give the resource group name and the Kubernetes cluster name.



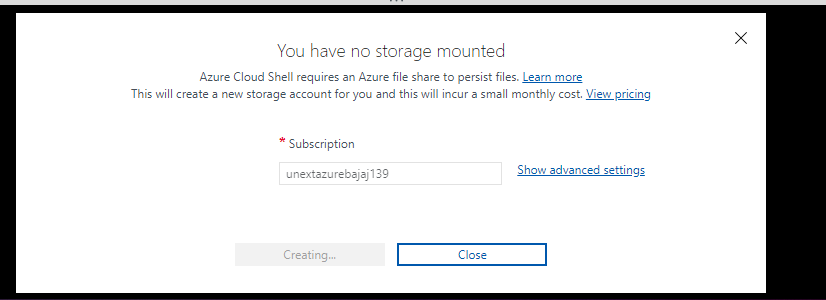
2. Give the node count range as 1. Click on review+create.



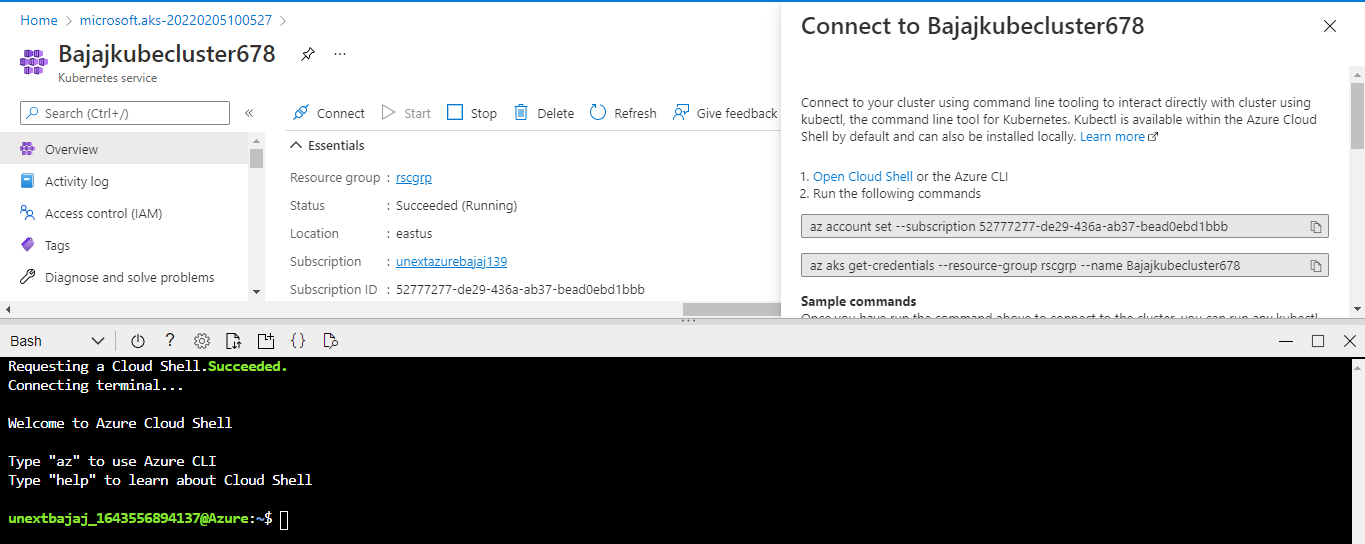
3. After validation is complete you may click on create to complete its creation.



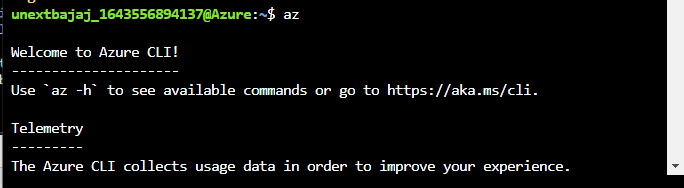
4. The deployment takes some time to complete, in the mean time we can proceed to create our storage account.



5. Open the cloud CLI

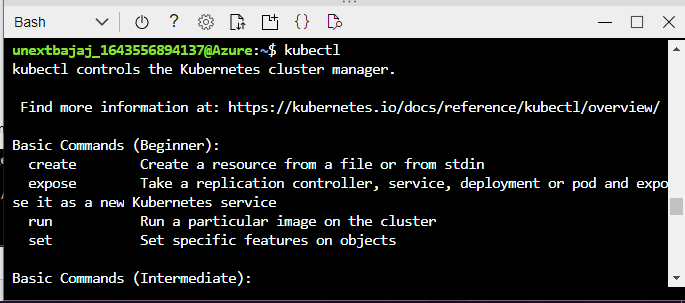


6. az – to open the azure CLI

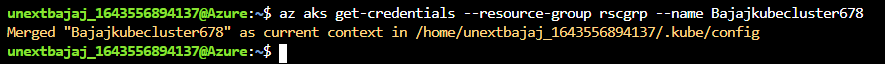


Note: Since the output was very long, some of it has been cropped.

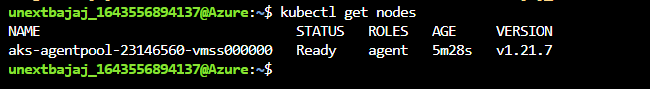
7. kubectl – kubectl is the Kubernetes command-line tool. It allows us to run commands against Kubernetes clusters — deploying applications, inspecting and managing cluster resources, and viewing logs.



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



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



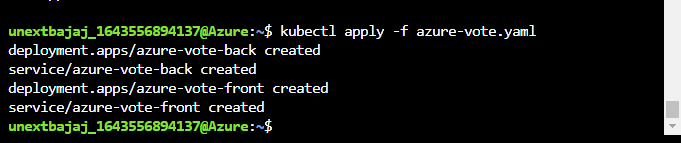
So far, all the commands were for connecting to the cluster.

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.

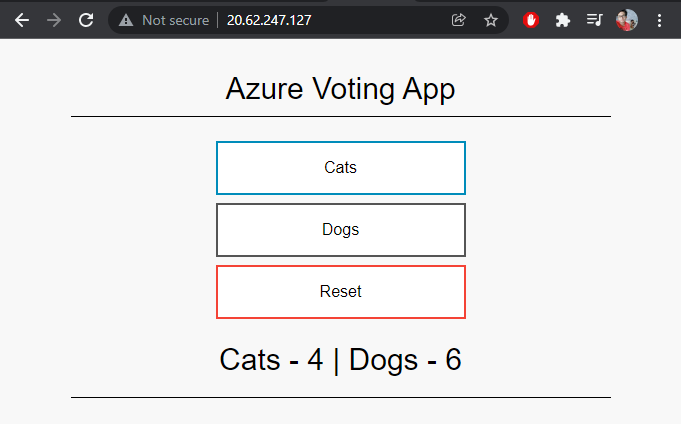
apiVersion: apps/v1  
kind: Deployment  
metadata:  
name: azure-vote-back  
spec:  
replicas: 1  
selector:  
matchLabels:  
app: azure-vote-back  
template:  
metadata:  
labels:  
app: azure-vote-back  
spec:  
nodeSelector:  
"kubernetes.io/os": linux  
containers:  
- name: azure-vote-back  
image: 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.

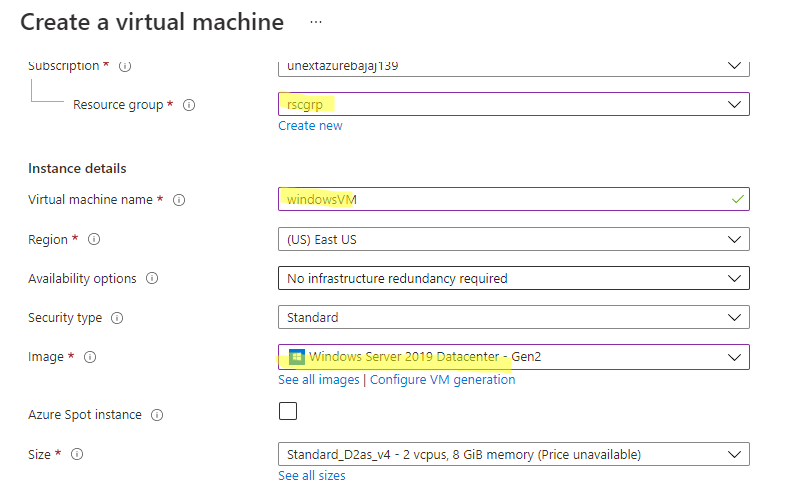
Open the web browser and type the external ip address of your service



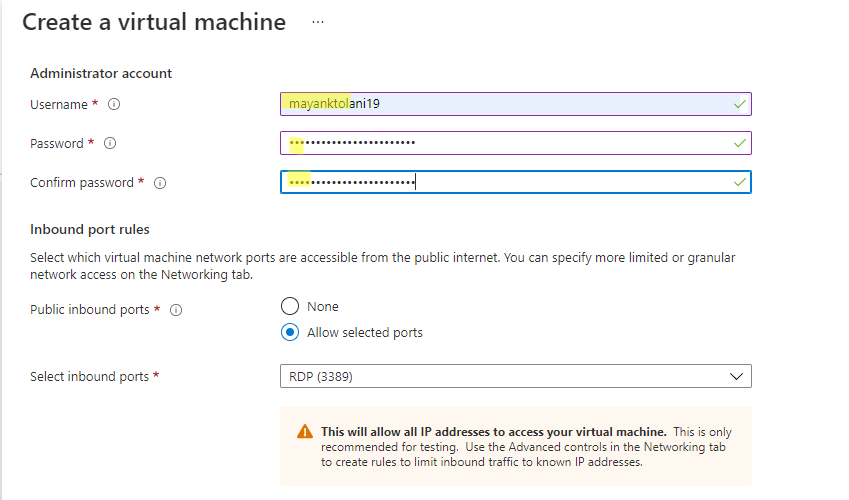
4. In Azure, please create Windows Server 2019 Datacenter VM & connect the VM with RDP & explain all steps with screenshots.

Steps:

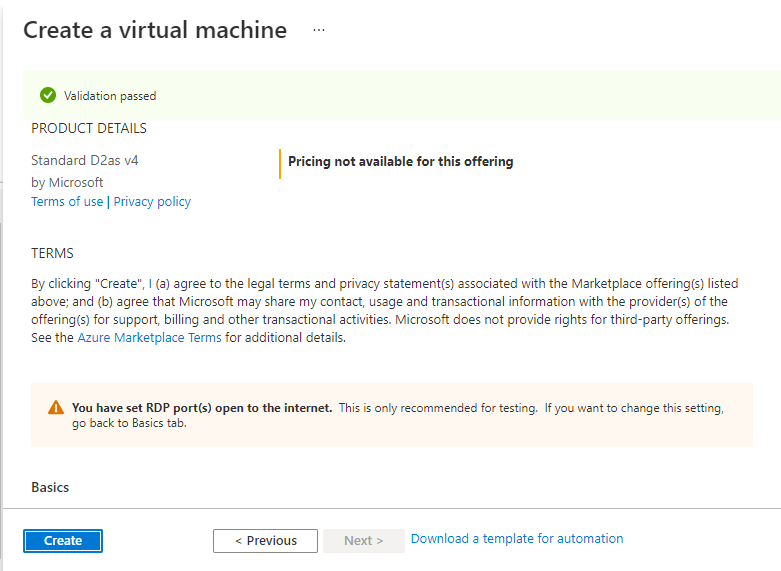
1. Create the windows VM. Select windows server 2019 datacenter as the image.



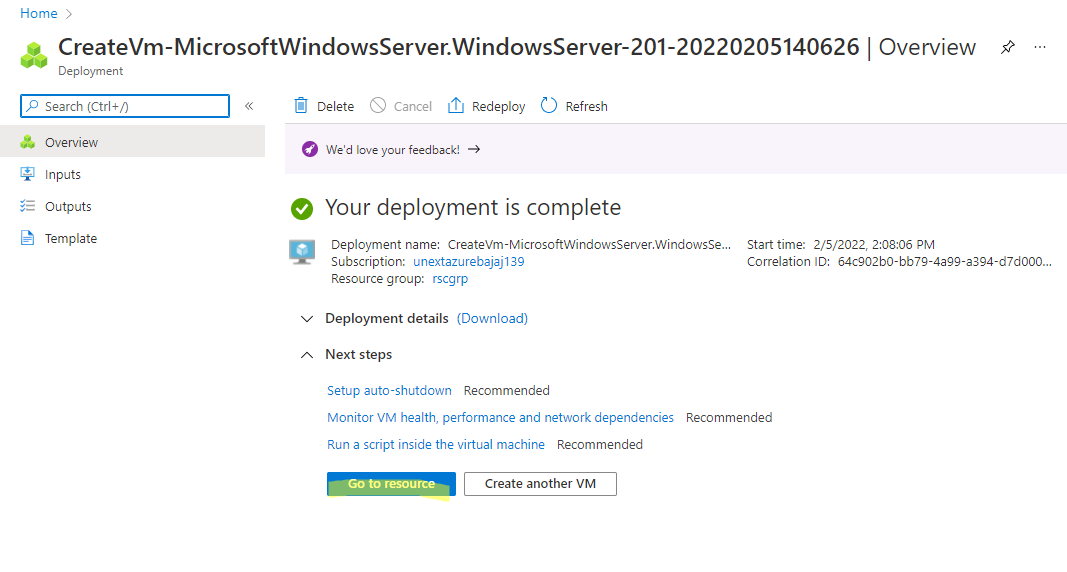
1. Give a username and password and click on review+create to proceed.



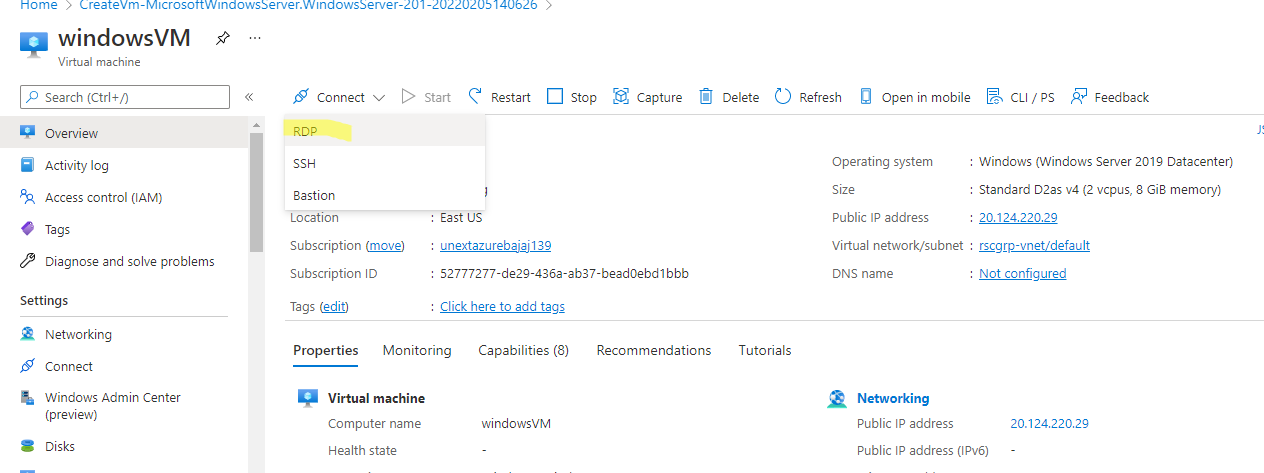
1. When validation is passed, click on create.



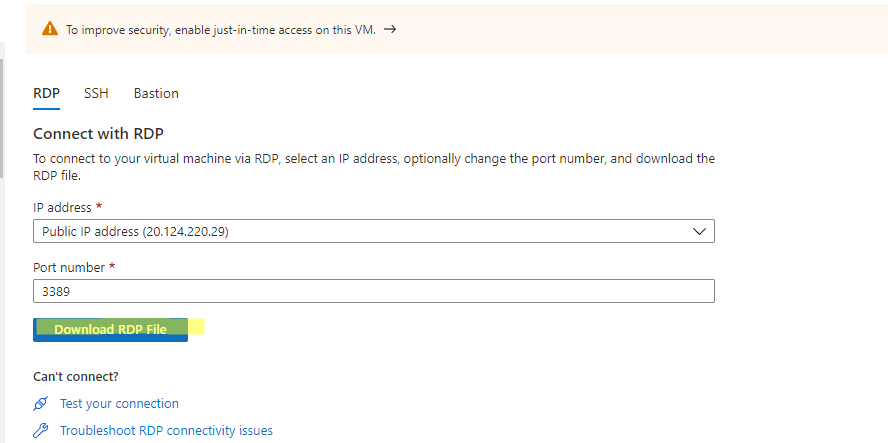
1. Once deployment is complete, go to resource.



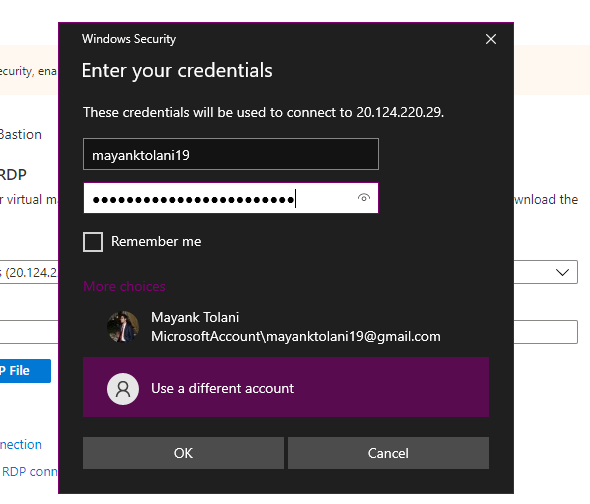
1. Connect via RDP now



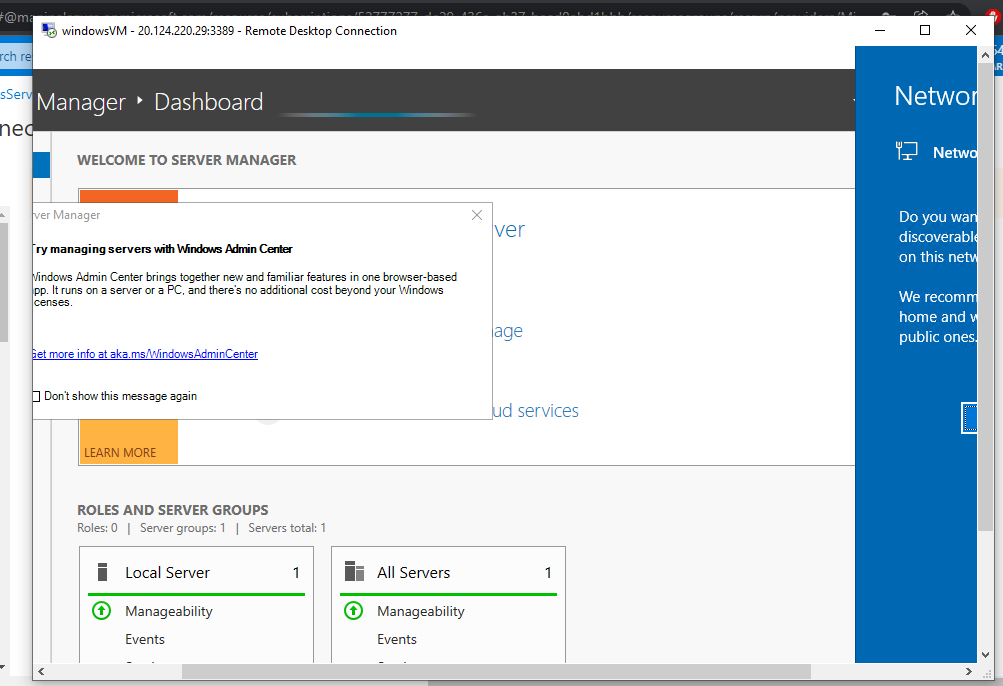
1. Download the RDP file



1. On opening the RDP file, a prompt will come to login to the account.



1. The remote desktop connection is now active.



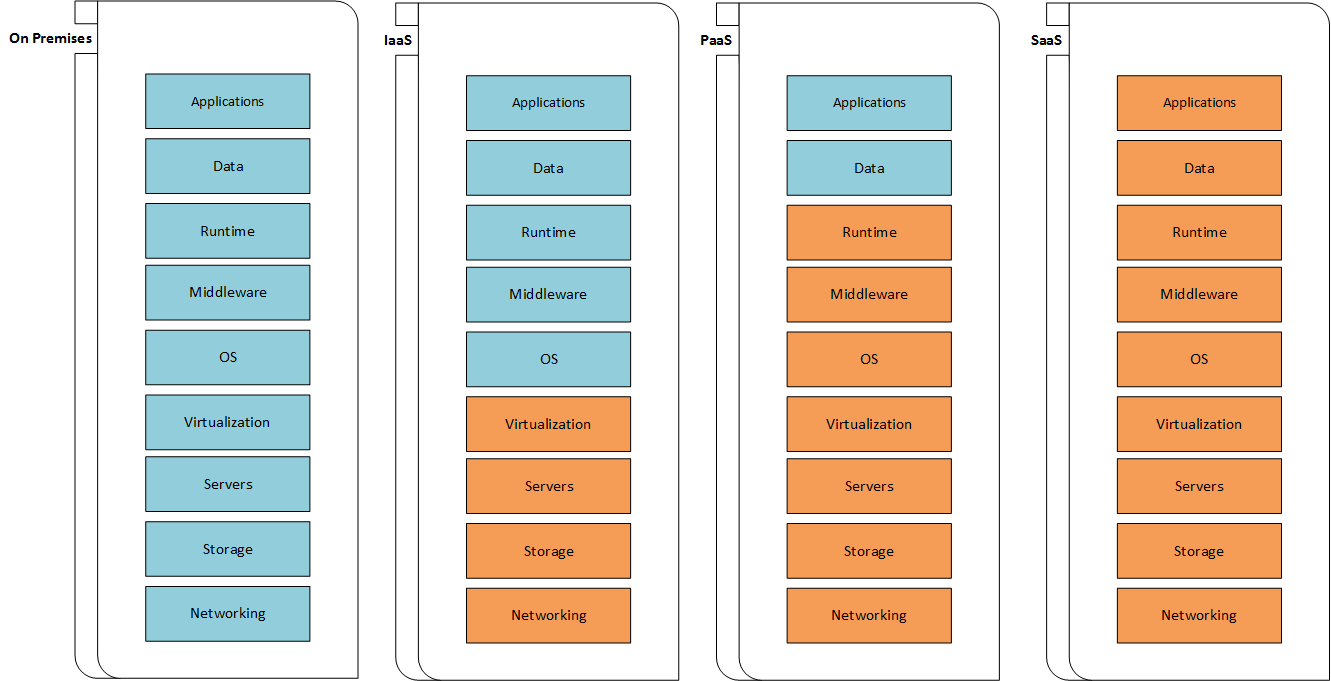
5. Explain various cloud Service & deployment models with block diagrams & examples.

Cloud service models

All workloads in a cloud scenario use resources from an extremely large resource pool that is operated (managed) by you or a cloud service provider. These resources include servers, storage, networks, applications, services, and much more.

The cloud service models describe to what extent your resources are managed by yourself or by your cloud service providers.

Let's look at the available service models. In the following diagram, you will find a comparison of the models and the existing management responsibilities. Areas that are colored in blue are managed by you: all others are the responsibility of your provider:



The offers are mainly categorized into the following service models:

On-premises: On-premises describes a model in which the user manages all resources alone.

Infrastructure as a Service (IaaS): IaaS describes a model in which the cloud provider gives the consumer the ability to create and configure resources from the computing layer upwards. This includes virtual machines, containers, networks, appliances, and many other infrastructure-related resources.

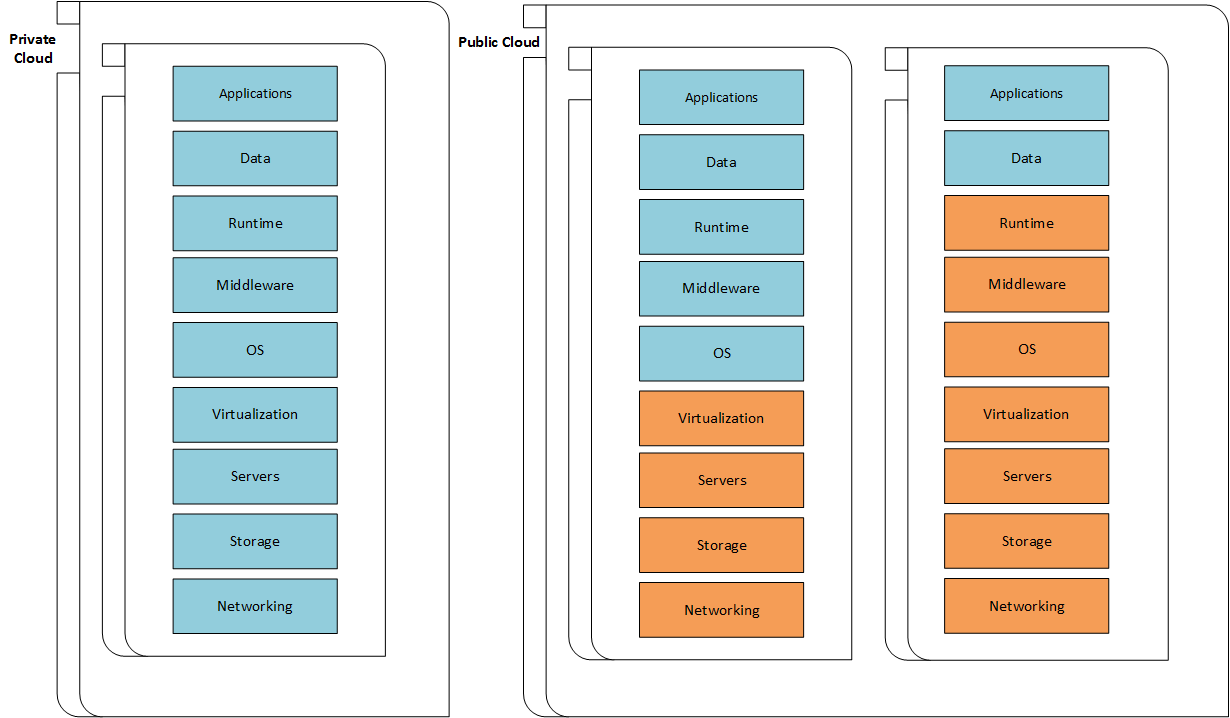
Platform as a Service (PaaS): PaaS gives the consumer an environment from the operating system upwards. So the consumer is not responsible for the underlying infrastructure.

Software as a Service (SaaS): SaaS is the model with the lowest levels of control and required management. A SaaS application is reachable from multiple clients and consumers, and the owning consumer doesn't have any control over the backend, except for application-related management tasks.

Cloud deployment models

Cloud deployment models describe the way in which resources are provided in the cloud.

Let's look at the following diagram first:



The deployment model based on the on-premises service model is called the private cloud. A private cloud is an environment/infrastructure, built and operated by a single organization, which is only for internal use.

In the context of this book, you should know that the Windows Azure Pack (a free add-on for the Windows server) gives you the opportunity to deploy Azure technologies in a private cloud environment.

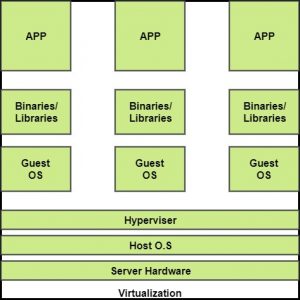
The deployment model based on the IaaS and the PaaS service model is called the public cloud. A public cloud is an offer from a service provider (for example, Microsoft Azure), that can be accessed by the public. This includes individuals as well as companies.

6. Explain what is cloud security & how virtualization is achieved with block diagram.

Cloud security involves the procedures and technology that secure cloud computing environments against both external and insider cybersecurity threats. Cloud computing, which is the delivery of information technology services over the internet, has become a must for businesses and governments seeking to accelerate innovation and collaboration. Cloud security and security management best practices designed to prevent unauthorized access are required to keep data and applications in the cloud secure from current and emerging cybersecurity threats. Most cloud providers attempt to create a secure cloud for customers. Their business model hinges on preventing breaches and maintaining public and customer trust. Cloud providers can attempt to avoid cloud security issues with the service they provide, but can’t control how customers use the service, what data they add to it, and who has access. Customers can weaken cybersecurity in cloud with their configuration, sensitive data, and access policies.

Virtualization allows to share a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to a physical storage and providing a pointer to that physical resource on demand. With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

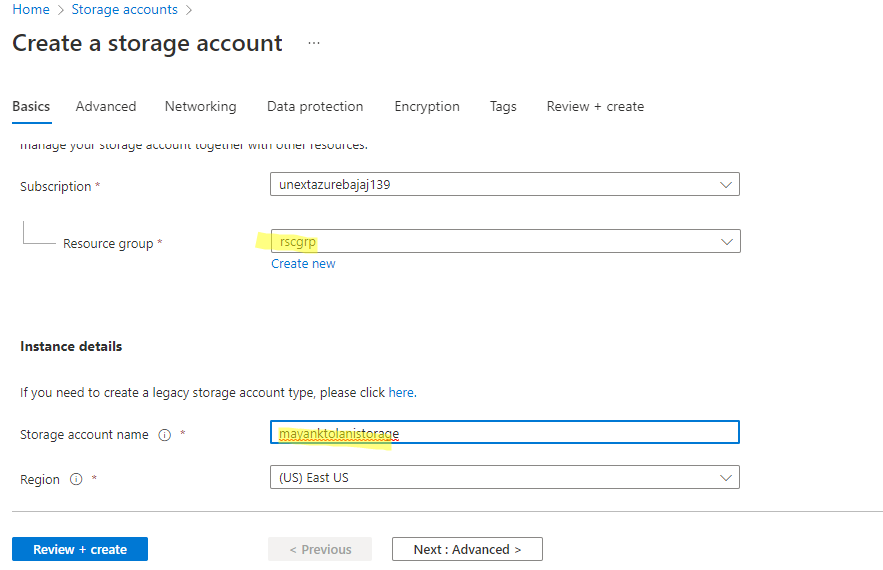
The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.



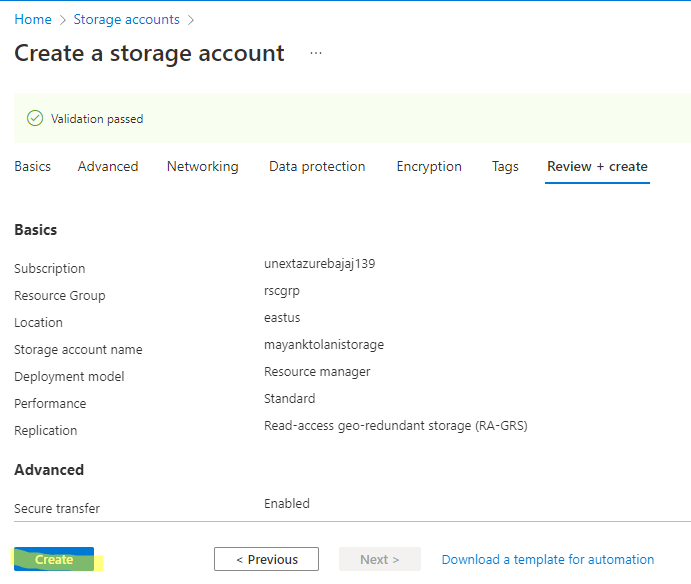
7. In azure please create Azure Blob & upload sample CSV, Excell, TXT files in it & explain all steps with screenshots.

Steps:

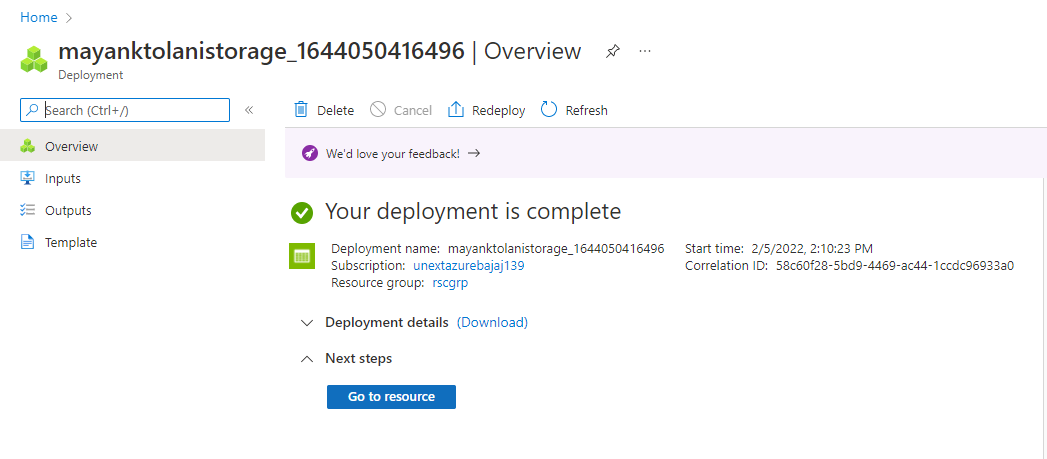
1. Create a storage account. Give resource group and storage account name.



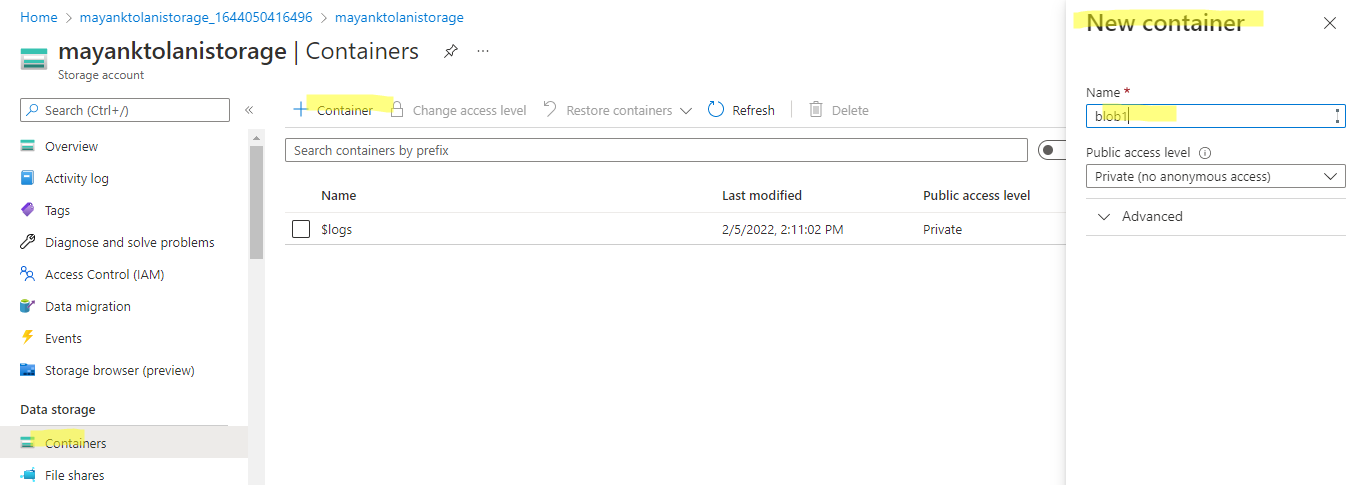
1. Click on create to complete the storage account creation.



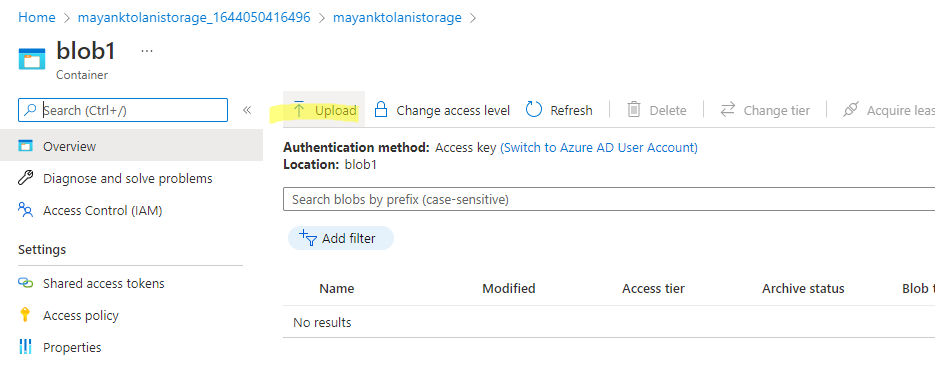
1. Once the deployment is complete, go to the resource.



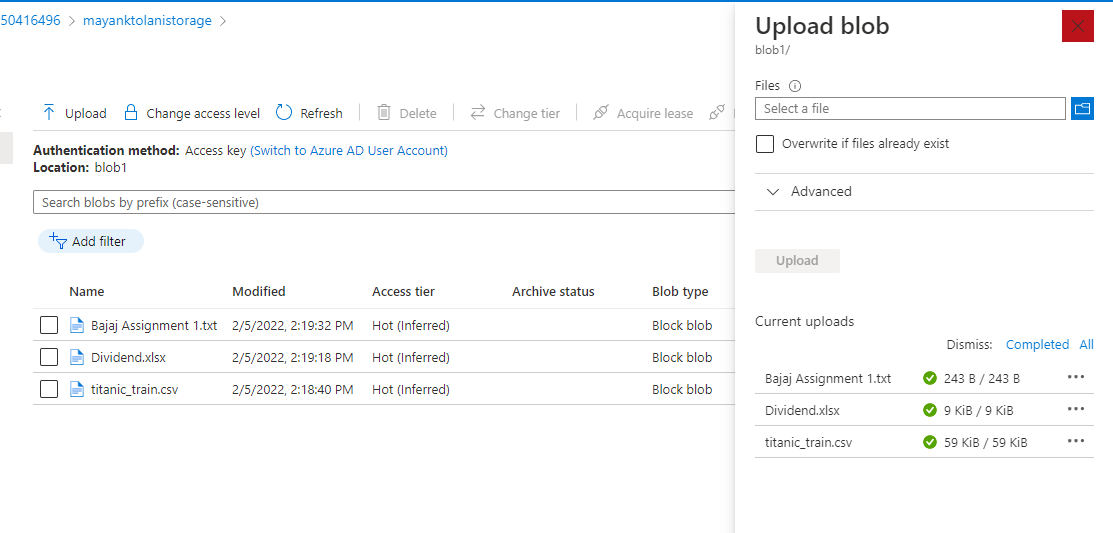
1. Go to create a new container and give a name to it. Then click on create.



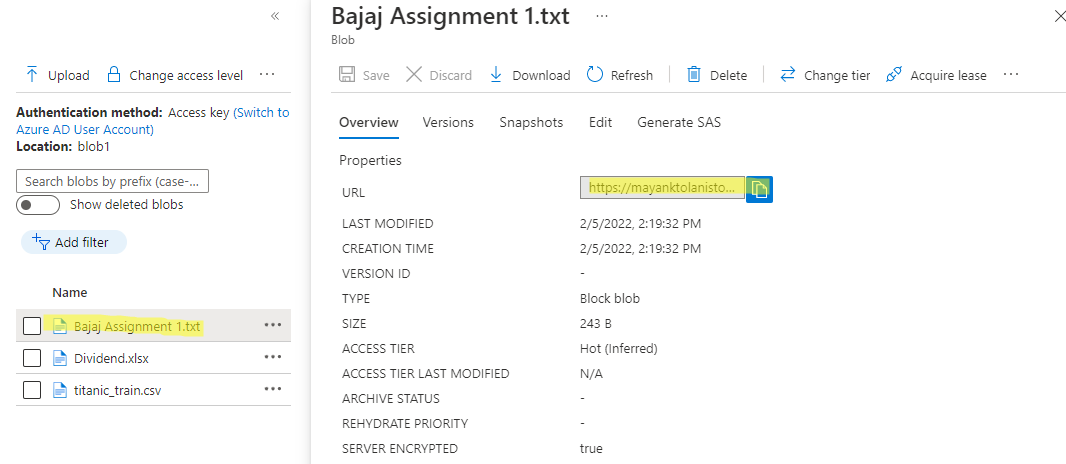
1. Upload some files and then go to the url to download them.



1. Uploaded the excel, csv and text file.



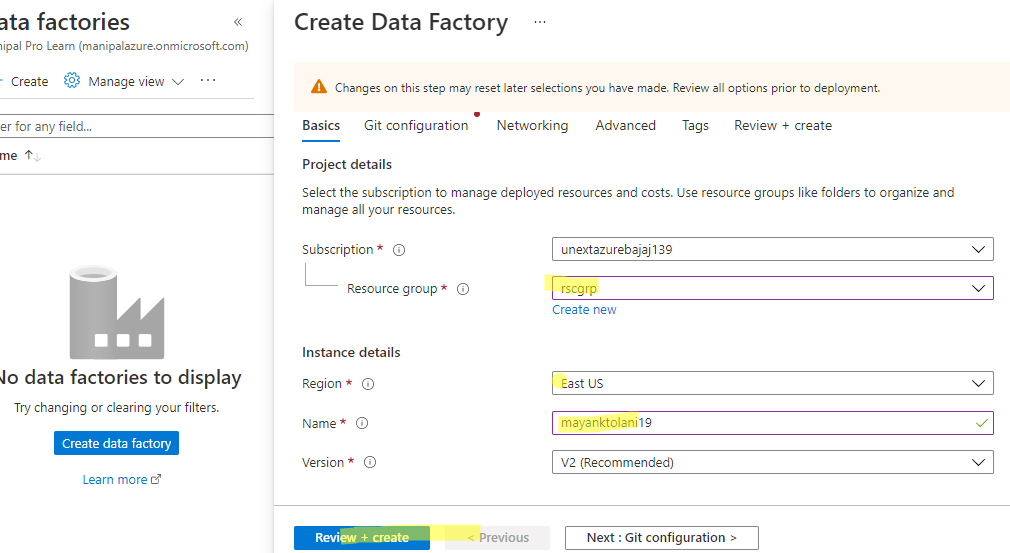
1. On clicking the file, you can get a download link for it.



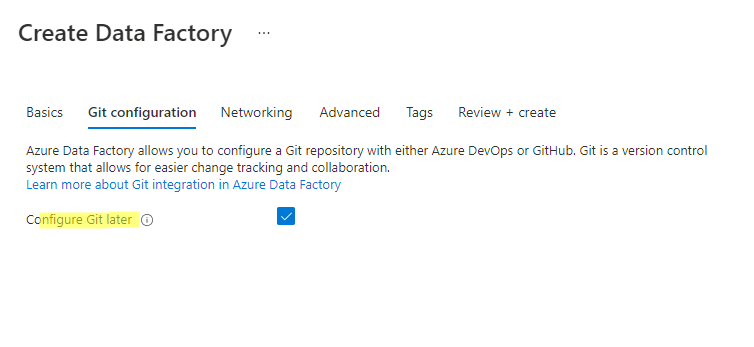
8. In Azure, please create Azure Data factory, create source & destination storage blobs,  
  
- try to copy a few files from Source blob to destination blob using Azure Data factory.

Steps:

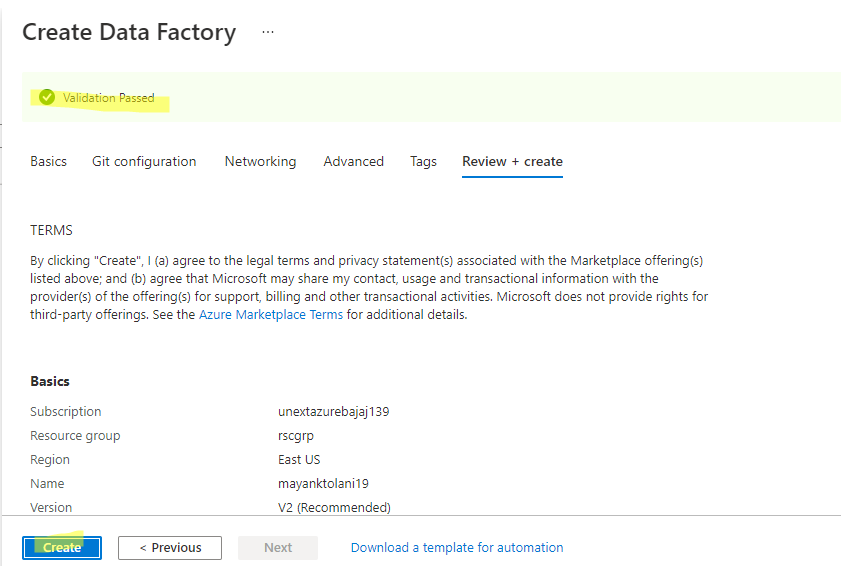
1. Create a data factory, give a resource group name, and the instance name.



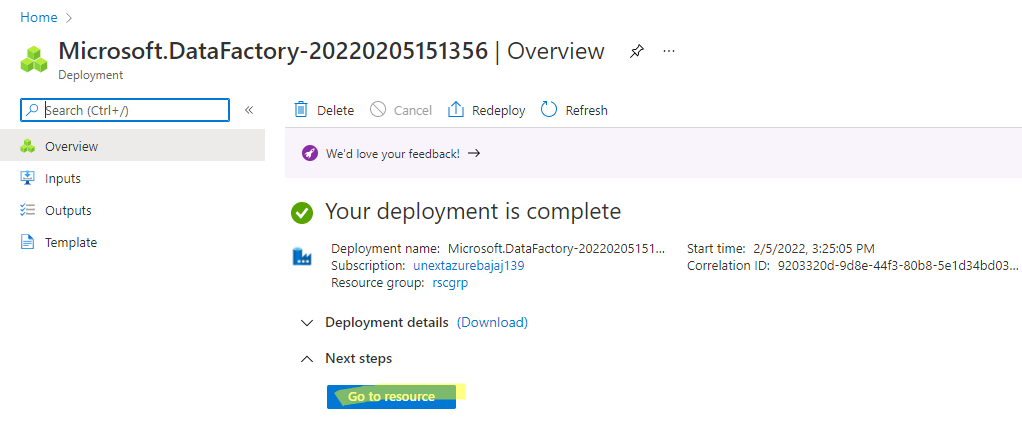
1. Under git configuration, select configure git later.



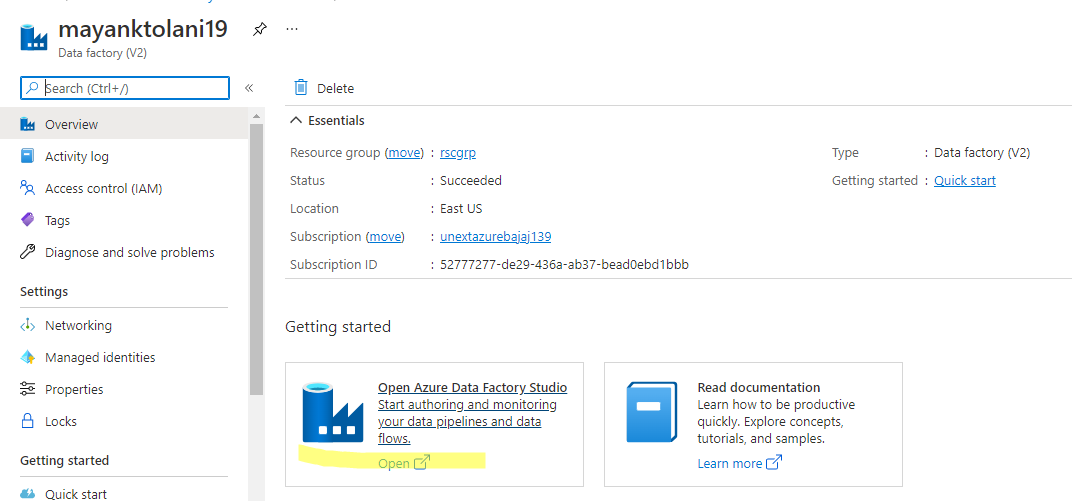
1. Once validation has been passed, click on create.



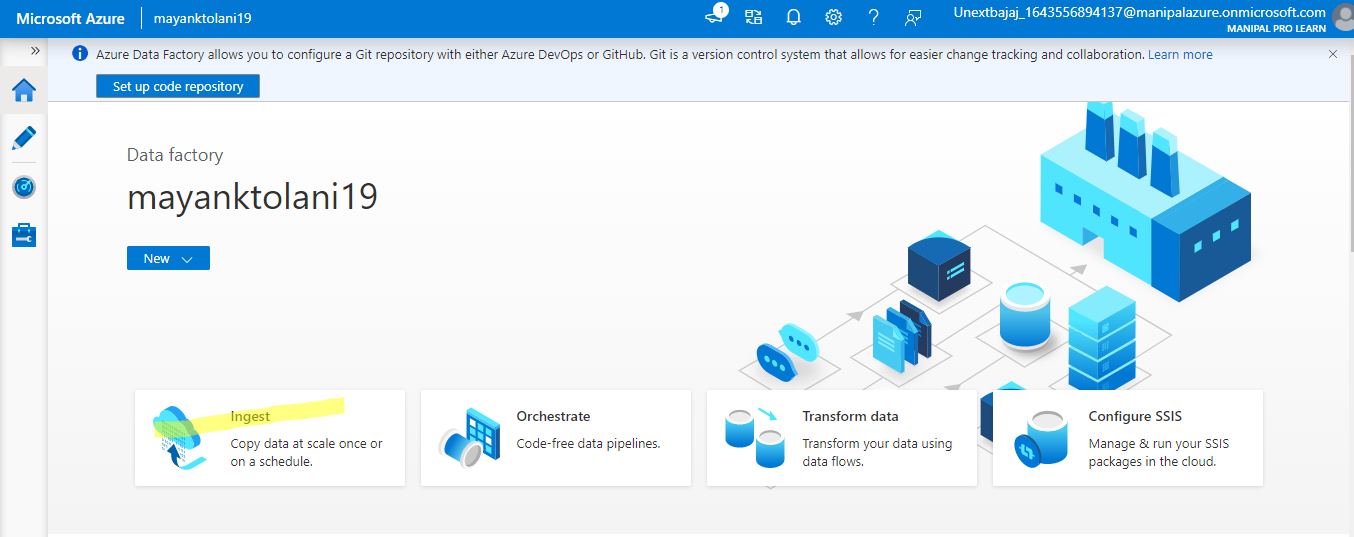
1. Once deployment is complete, go to resource.



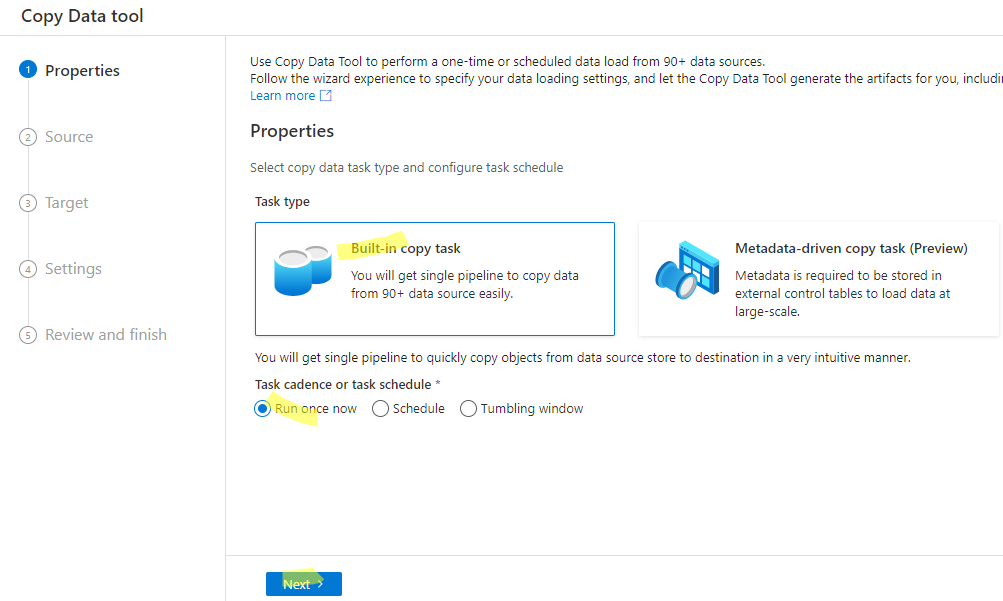
1. Now click on azure data factory studio.



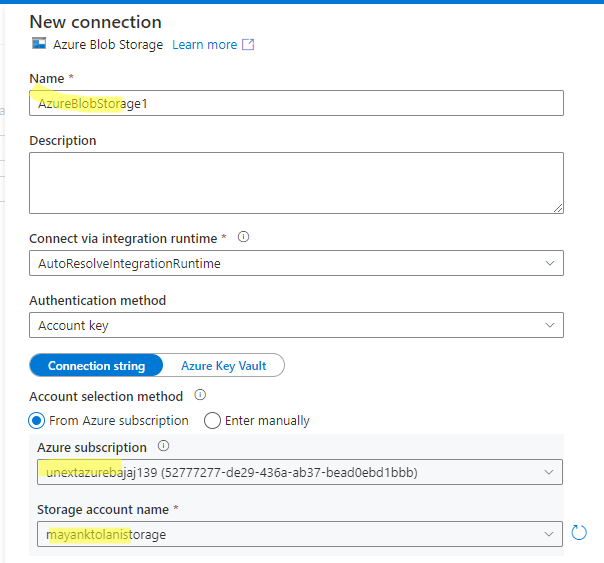
1. Once data factory is opened, click on ingest.



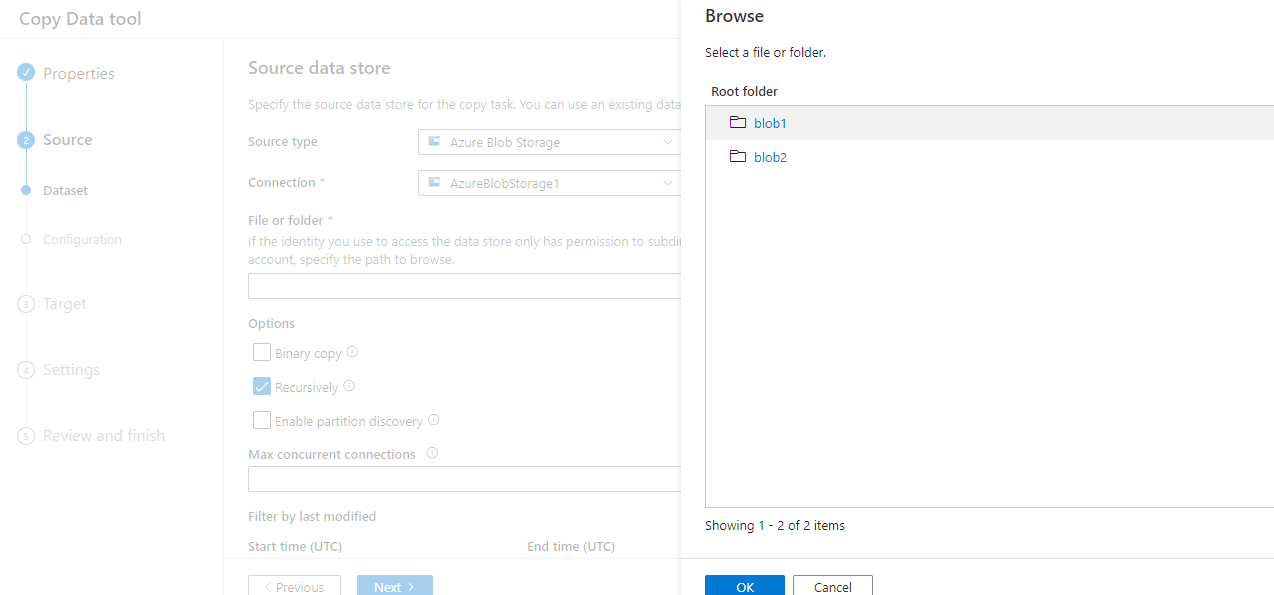
1. Select built-in copy task as the task type and run once now.



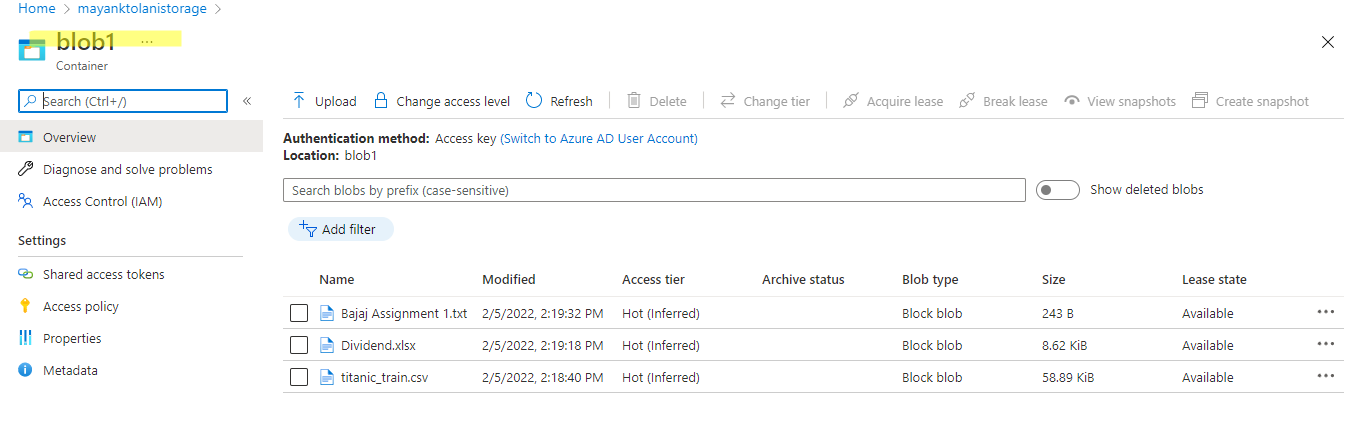
1. Add a new connection. Select your subscription and the storage account name.



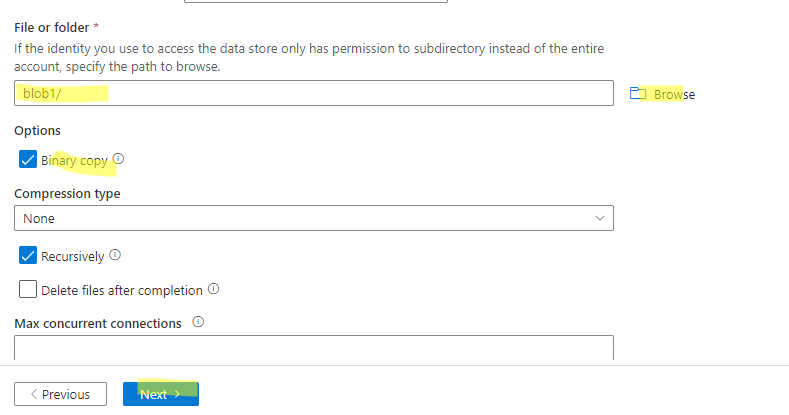
1. Select blob1 as the source folder which has 3 files.



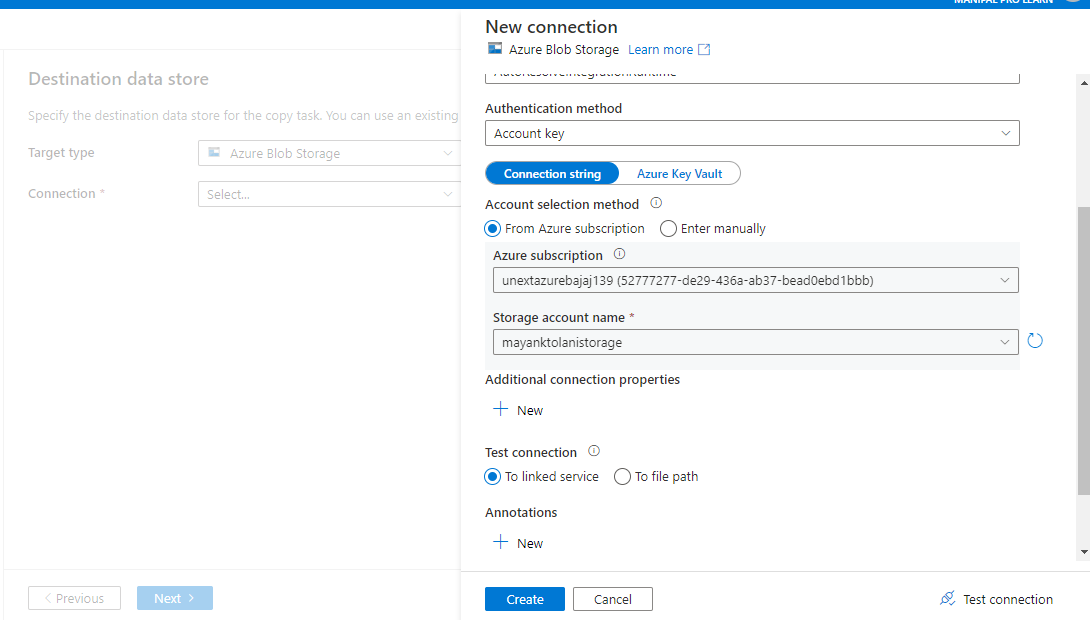
1. Blob1 has 3 files which will be copied to blob2.



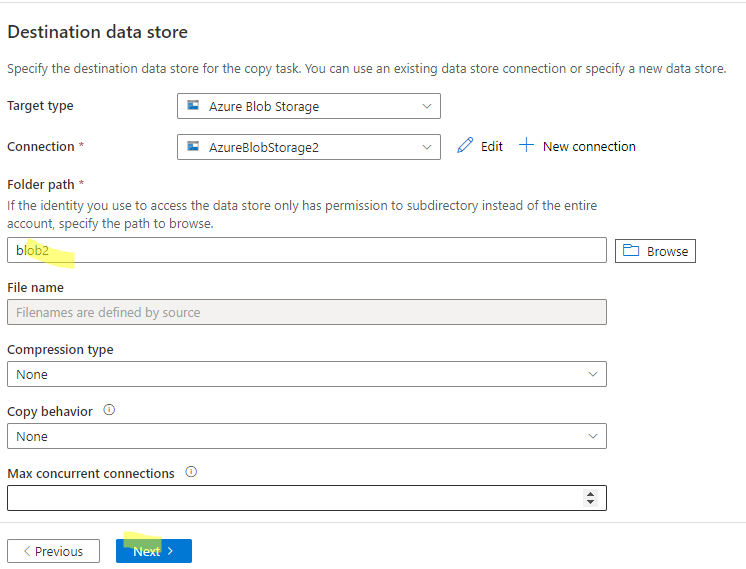
1. Click on binary copy and go to next



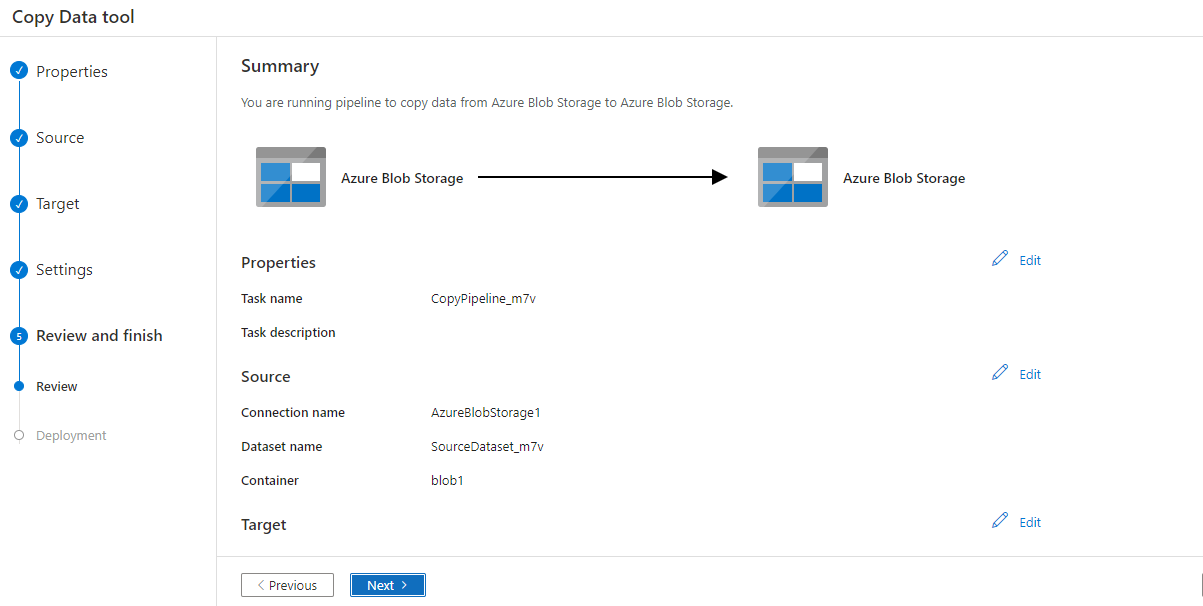
1. Select destination data store as blob2.



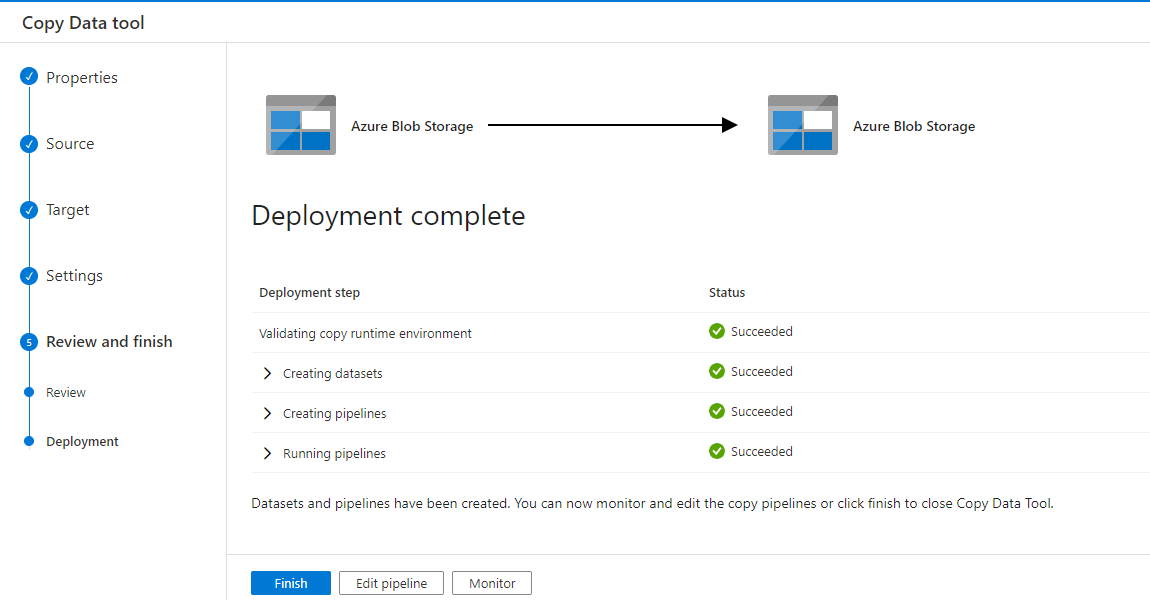
1. Now select blob2 here



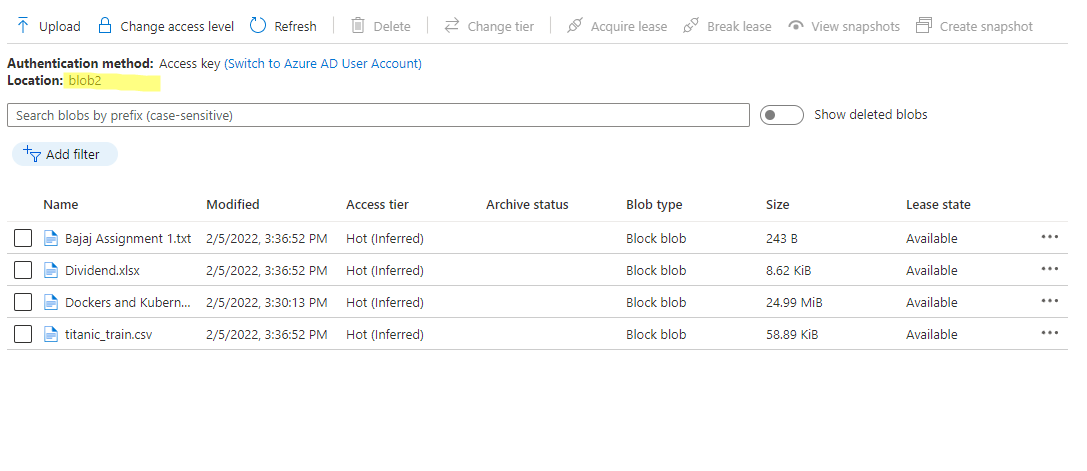
1. Summary is shown below.



1. Successfully created



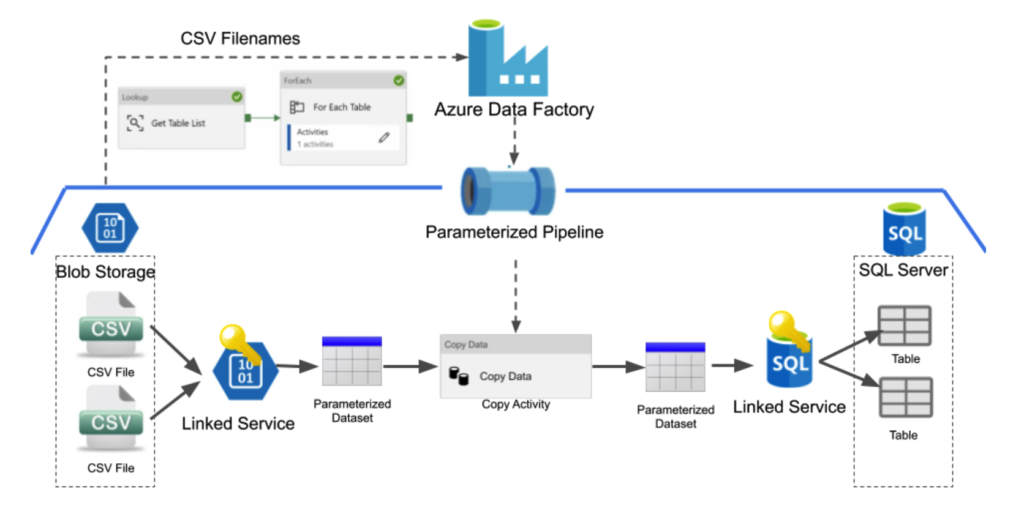
1. It can be seen that blob1 data is now available inside blob2



9. Explain Azure Data factory pipeline process with the diagram.

A data factory might have one or more pipelines. A pipeline is a logical grouping of activities that performs a unit of work. Together, the activities in a pipeline perform a task. For example, a pipeline can contain a group of activities that ingests data from an Azure blob, and then runs a Hive query on an HDInsight cluster to partition the data.

The benefit of this is that the pipeline allows you to manage the activities as a set instead of managing each one individually. The activities in a pipeline can be chained together to operate sequentially, or they can operate independently in parallel.



The diagram above is a simple example of an Azure Data Factory pipeline. In this example, we want to move a single CSV file from blob storage into a table stored in a SQL server database. This activity is done through an Azure Data Factory (ADF) pipeline. ADF pipelines consist of several parts and typically consist of linked services, datasets, and activities.

A Linked Service is used to connect a data store to ADF. It is very similar to a connection string in that it defines the connection information needed for ADF to connect to external resources, such as blob storage or a SQL Database.

A dataset is a named view of data that simply points to or references the data you want to use in your activities as inputs and outputs.

Activities in a pipeline define the actions you wish to perform on your data. Here is a link to find out about all the activities ADF offers, but for this example, we will focus on the copy activity which simply copies the data from one dataset to the other.

To create the example pipeline above a user would need to create all of the following:

A linked service to allow ADF access to blob storage

A dataset, which references the target CSV file in blob storage

A linked service to allow ADF access to the SQL server database

A dataset, which references the target table in the SQL server database

A copy activity that will copy the data from the first dataset to the second

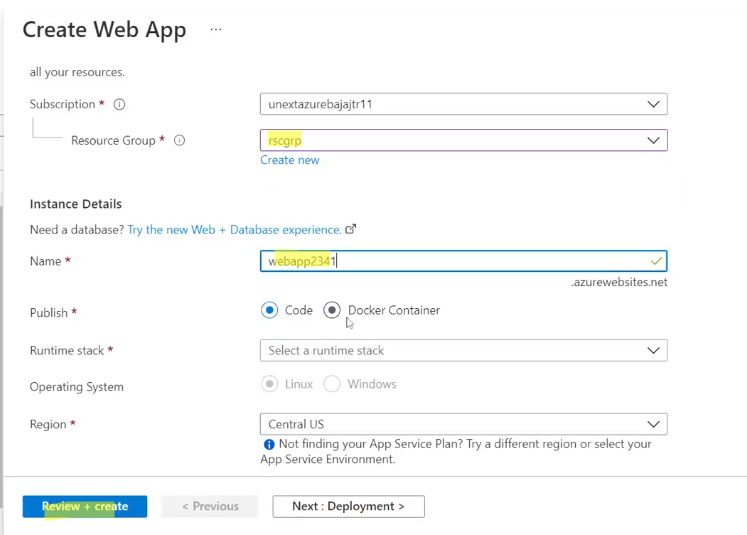
Once you have created all of the linked services, datasets, and the copy activity, this pipeline will then populate the table in the SQL server database with the data that was in the original CSV file.

This simple pipeline works wonderfully for copying a single CSV file into the SQL table. Now let’s take a look at a diagram where we want to copy over two CSV files into two separate tables into the SQL database.

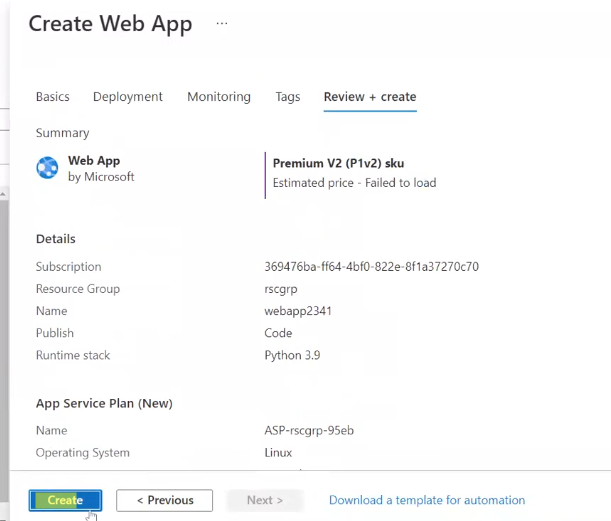
10. In Azure, please create a Python app service & explain all the steps with screenshots?

Steps:

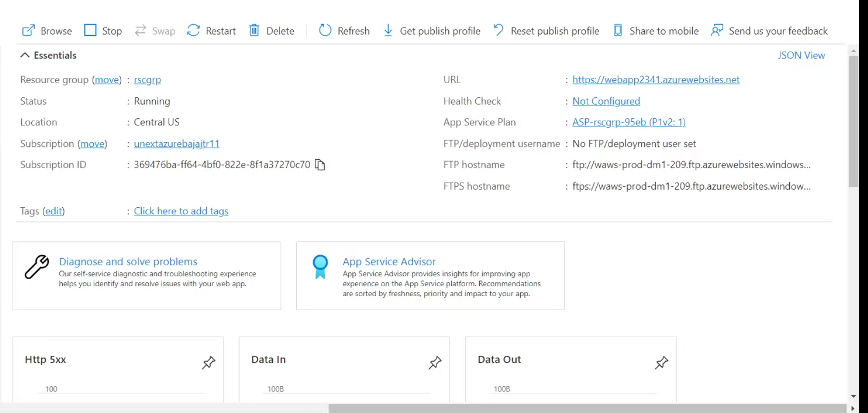
1. Create a web app – Select a resource group, give the name of the app and select a runtime stack.



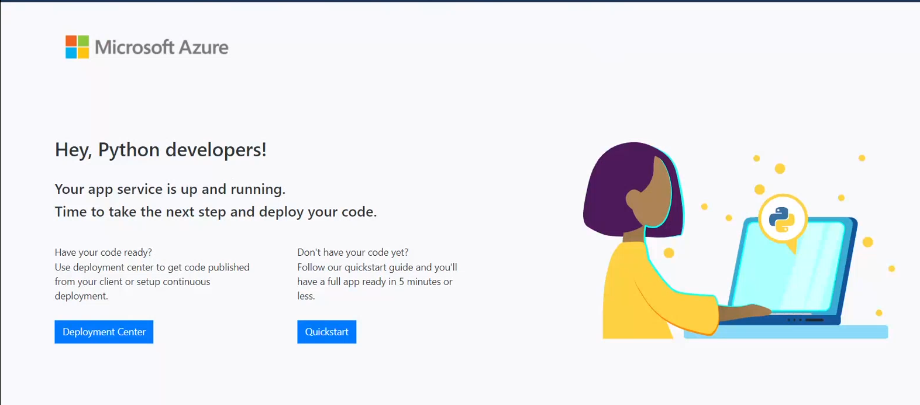
1. Go to review+create and click on create.



3. Once the deployment is complete go to resource and then go to the url to view your python application deployed.



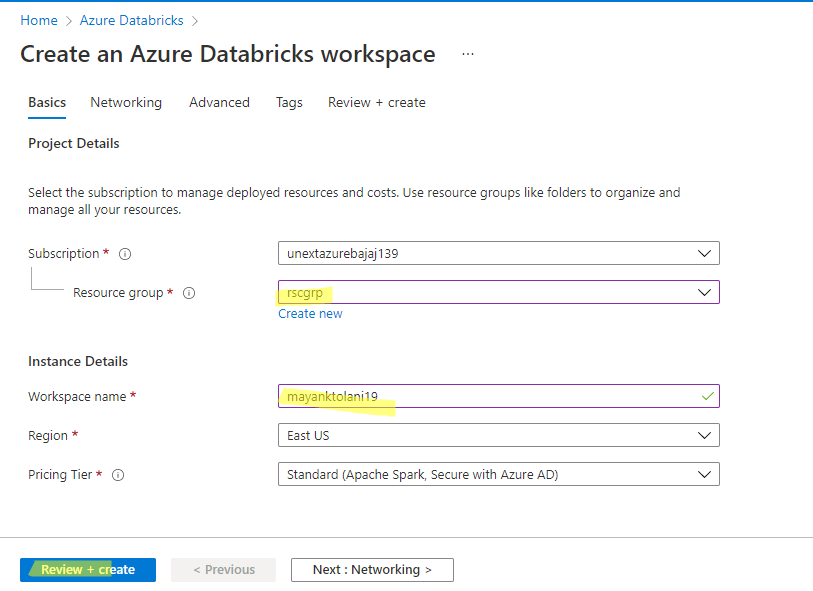
4.



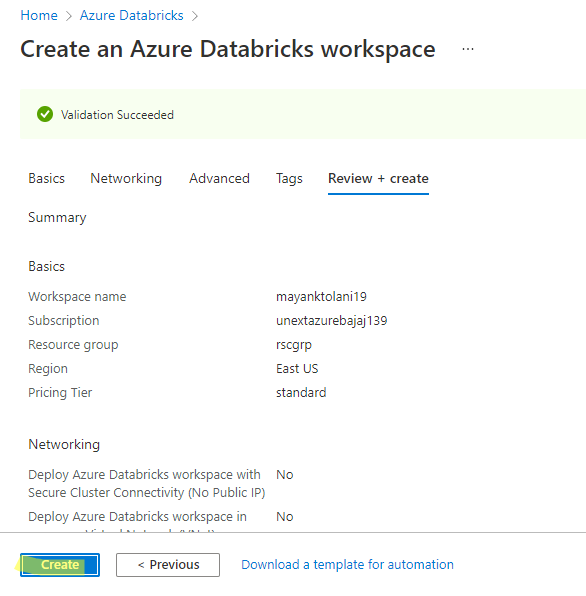
11. Please create Azure data bricks & explain with all steps & screenshots.

Steps:

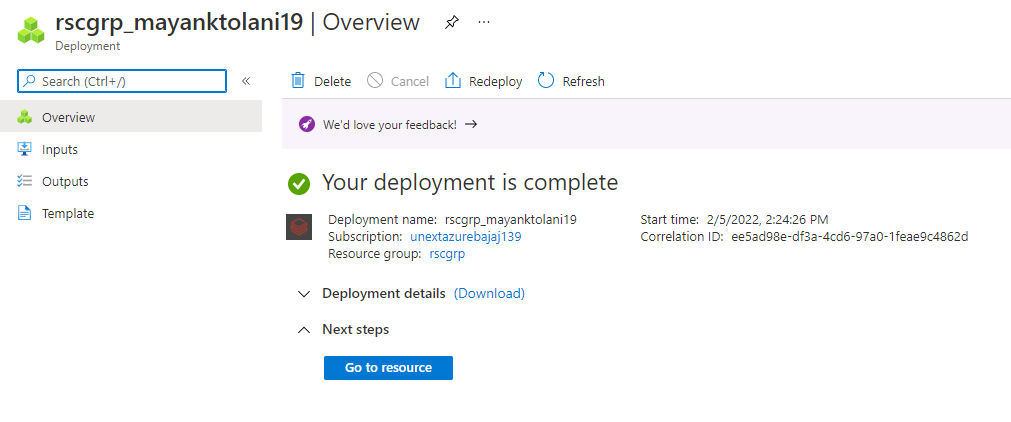
1. Creating a new azure databricks workspace



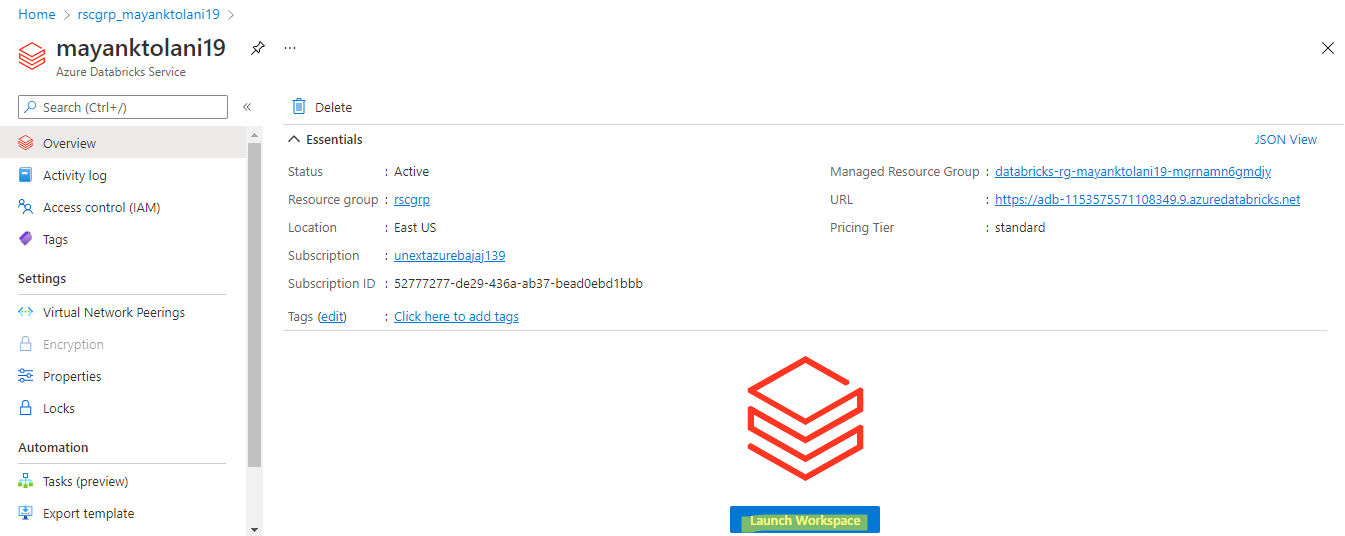
1. After validation is complete, click on create.



1. When deployment is complete, go to resource group.

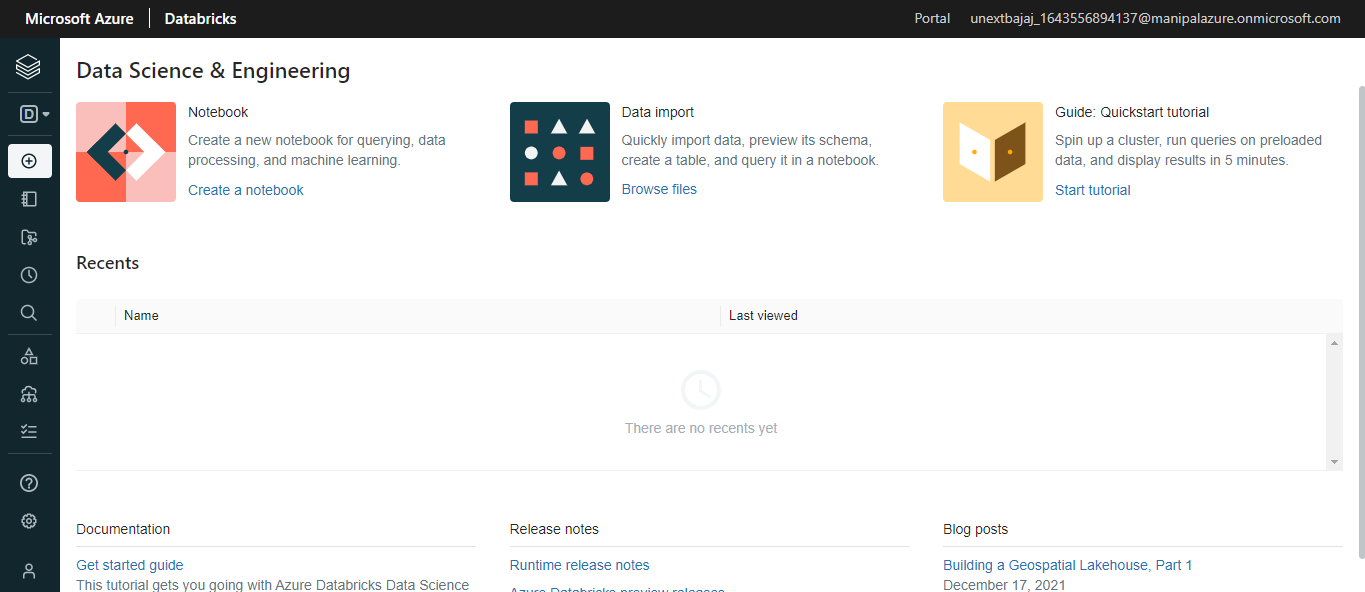


1. Click on launch workspace



Then sign in to databricks.

1. Thus we have successfully connected to databricks by creating azure data bricks.

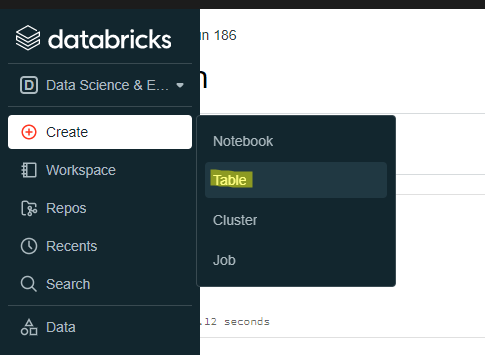


1. In Azure data bricks - please add data & create table

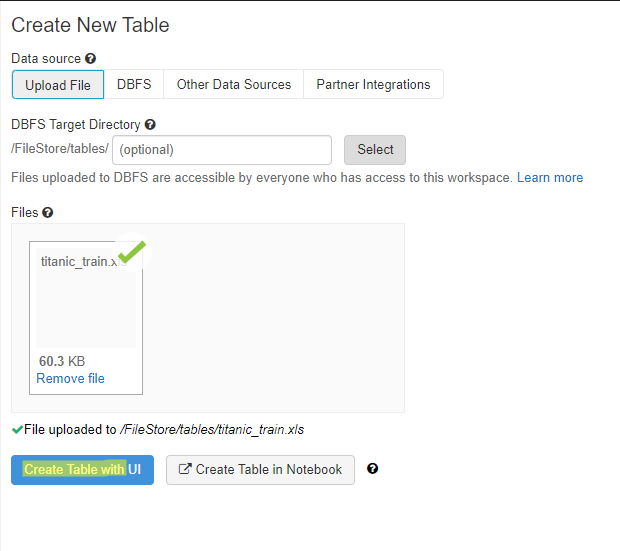
(1) Using**UI option**  
(2) create a table using the notebook option & explain all the steps.

Steps with UI option:

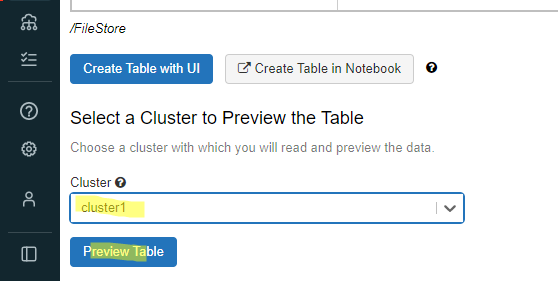
1. Go to create, then select table.



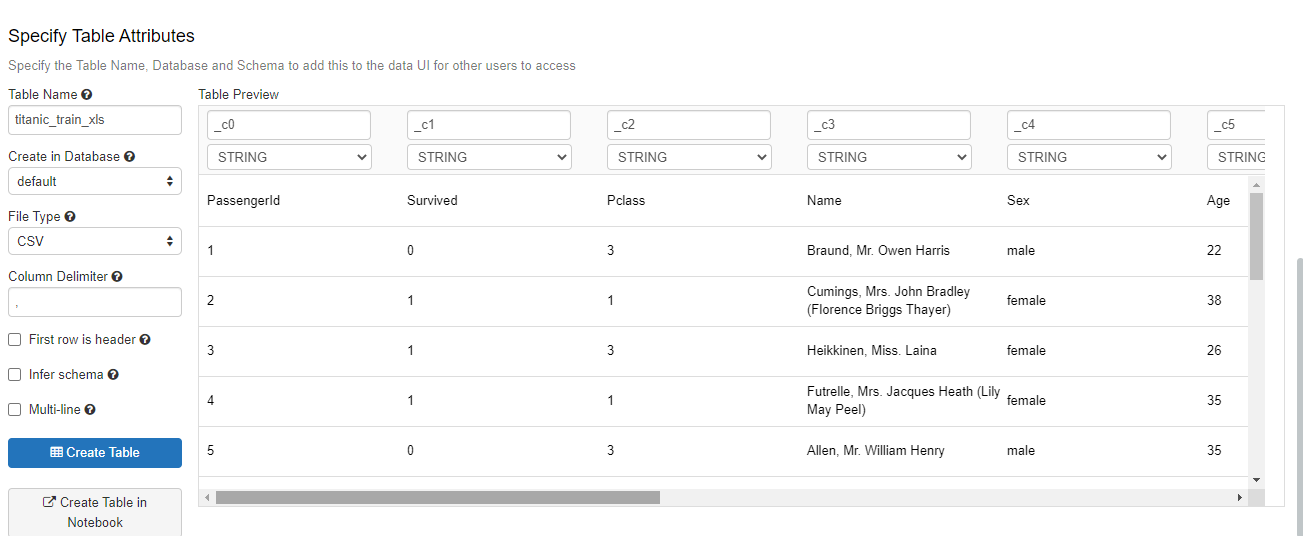
2. Upload a file and then click on create table with UI.



1. Select cluster and go to preview table.

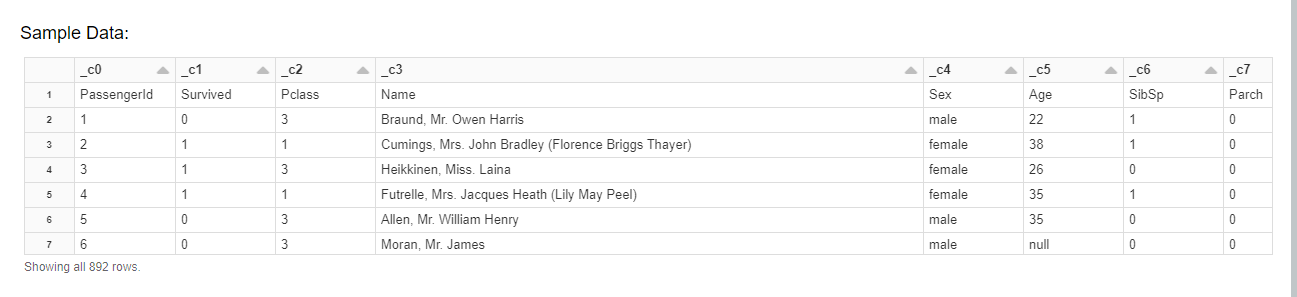


1. On previewing, the table is shown as such, click on create table.



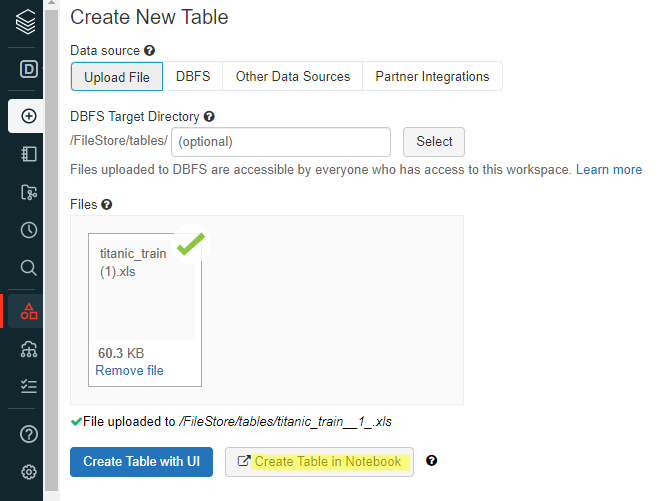
1. The table has been created successfully. The schema and sample data is shown below





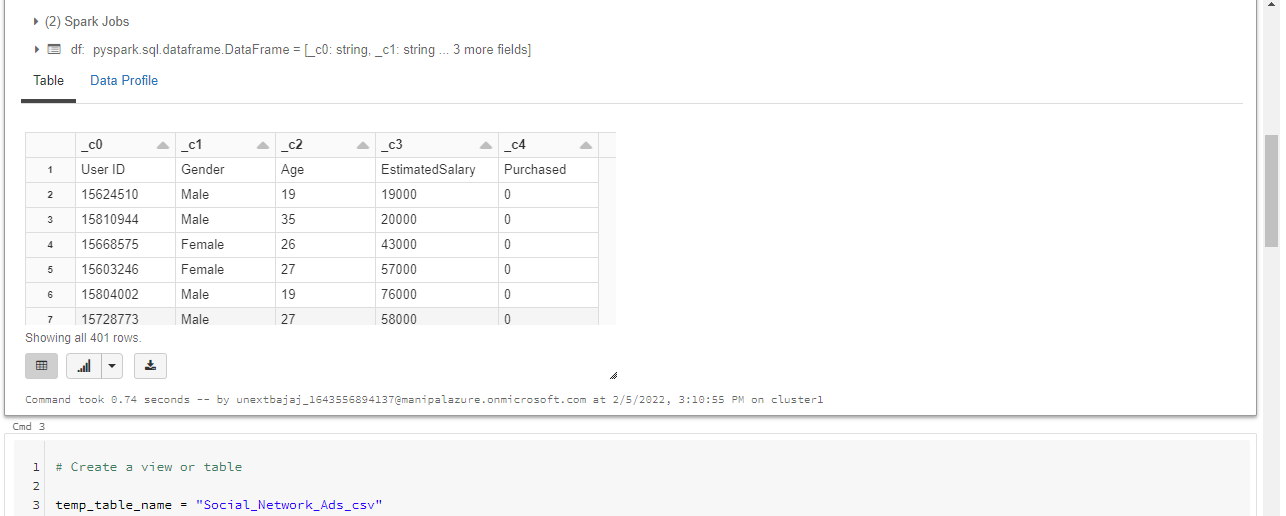
Steps with notebook option:

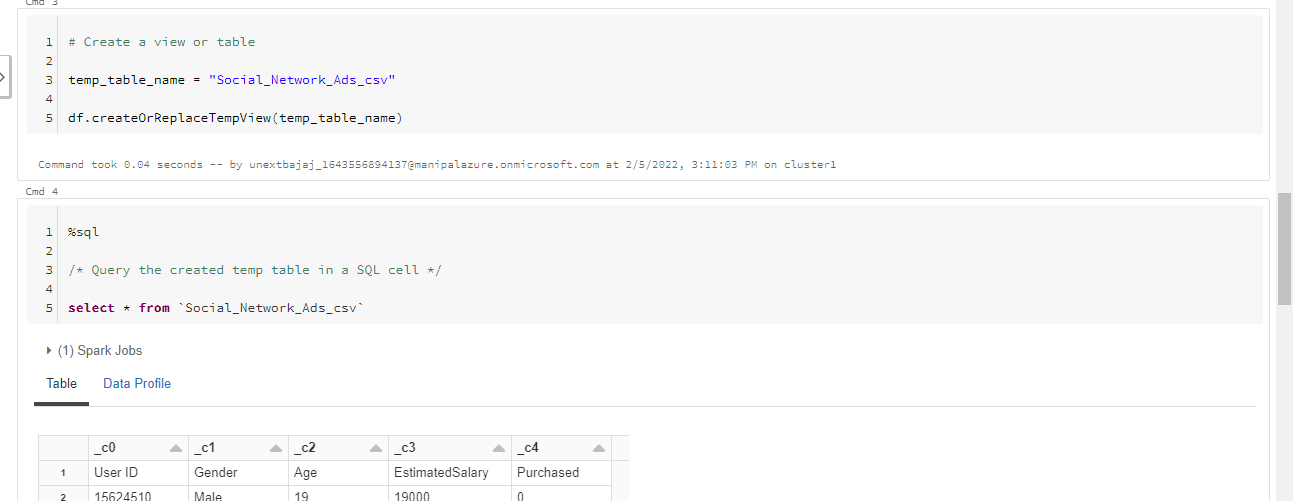
1. Upload the file then click on create table with notebook.

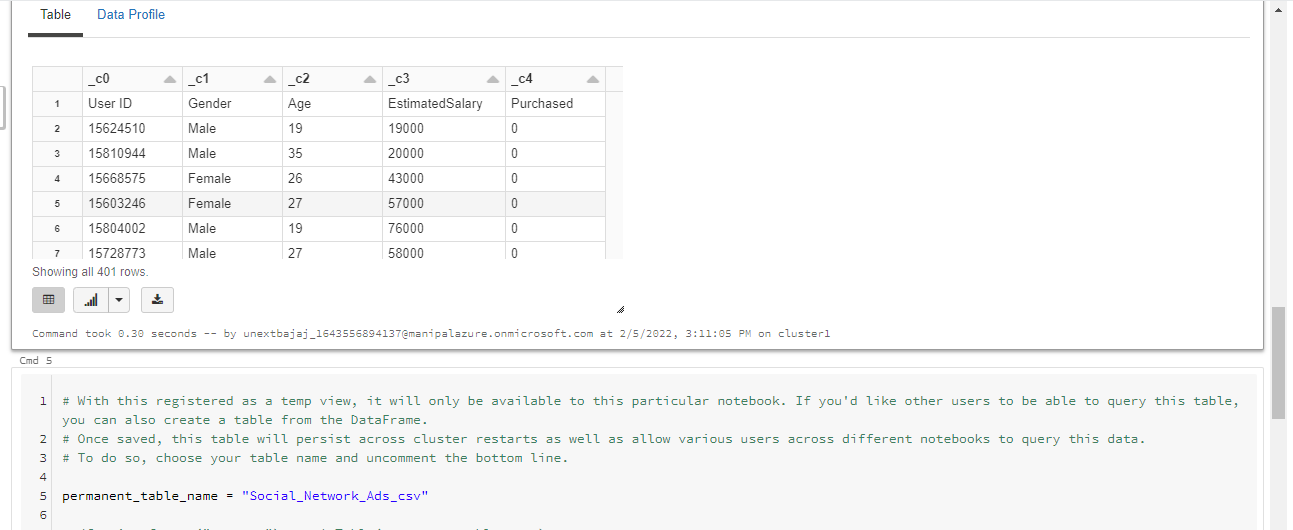


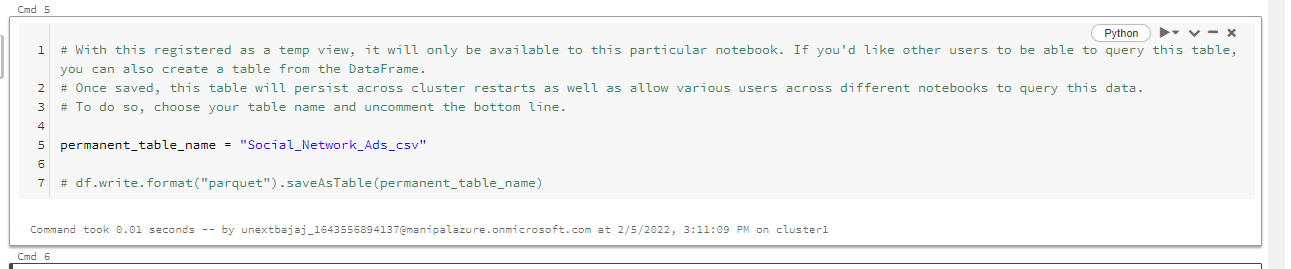
2. Attach the cluster. The code and its output is shown in the below screenshot.







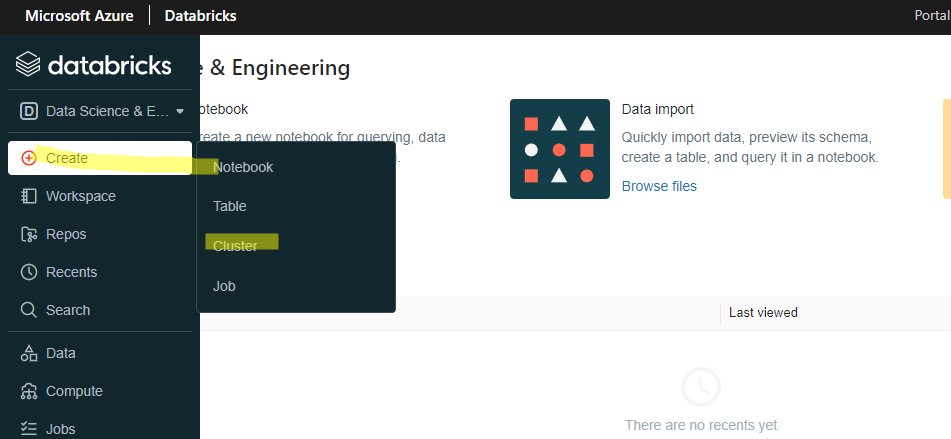




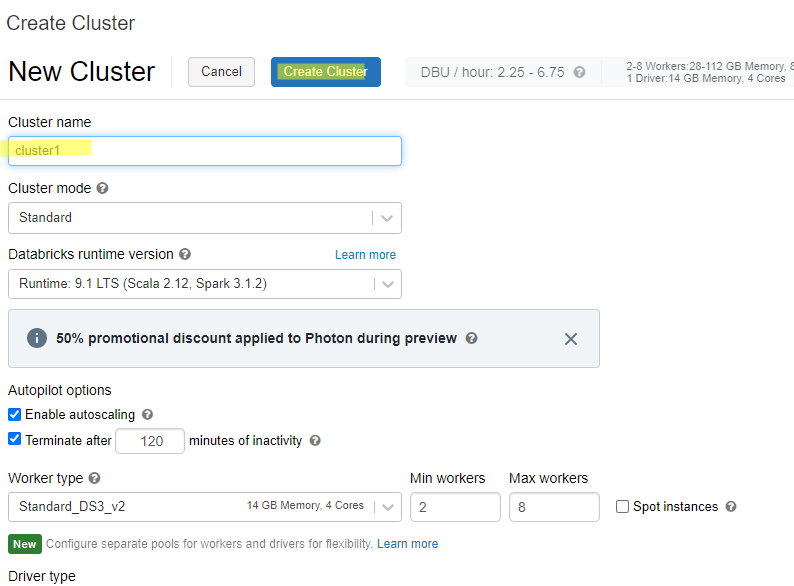
13. In Azure data bricks - please create Cluster & explain with all steps with screenshots.

Steps:

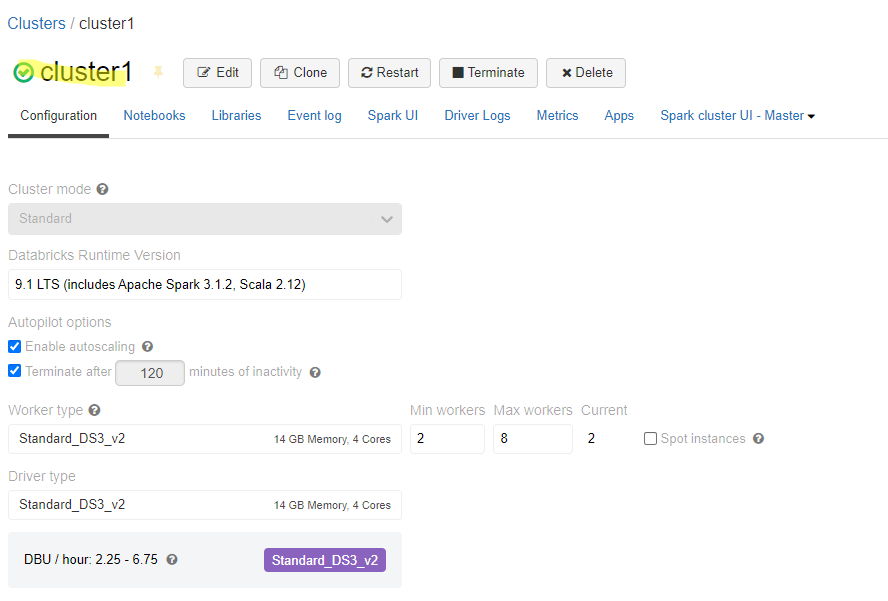
1. Click on the plus sign to create a new cluster.



1. Give a cluster name and click on create cluster.

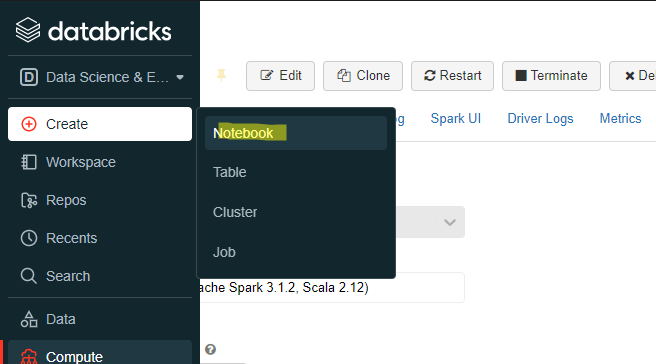


3. Cluster has been successfully created.

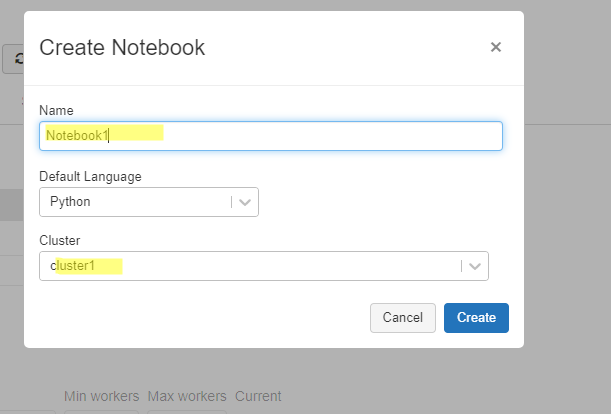


14. In Azure data bricks - please create sample Notebook & jobs, further assign sample notebook to jobs & run it, explain with all steps with screenshots?  
  
Steps:

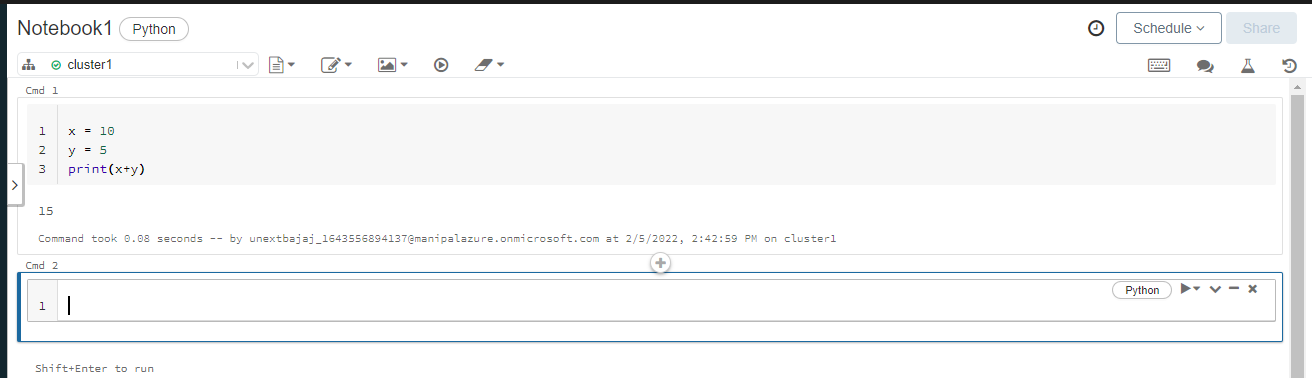
1. Go to create, then select notebook.



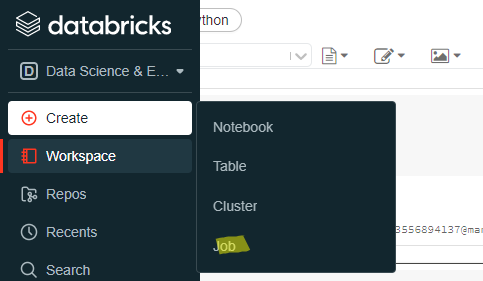
2. Give your notebook a name and associate it with the cluster we just created in the previous question.



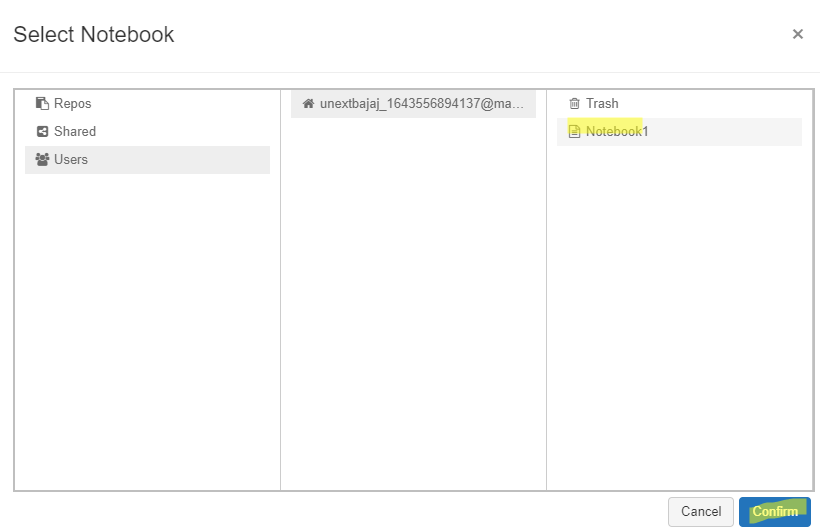
3. The notebook has been successfully created and tested by running a small python code.



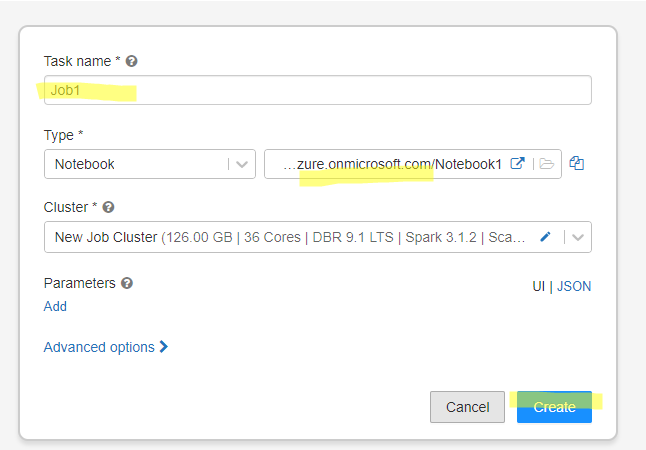
4. Now create a job.



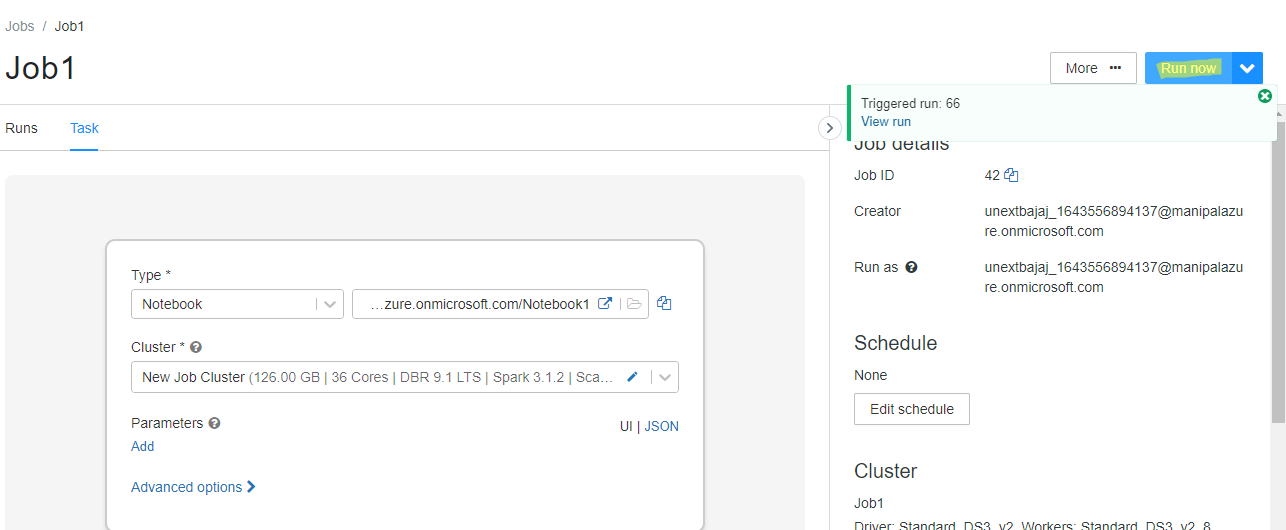
1. Now select a notebook which will be associated with this job.



1. Give your job a name and click on create.



1. Click on run now to run the current job.



1. Output of the job run

