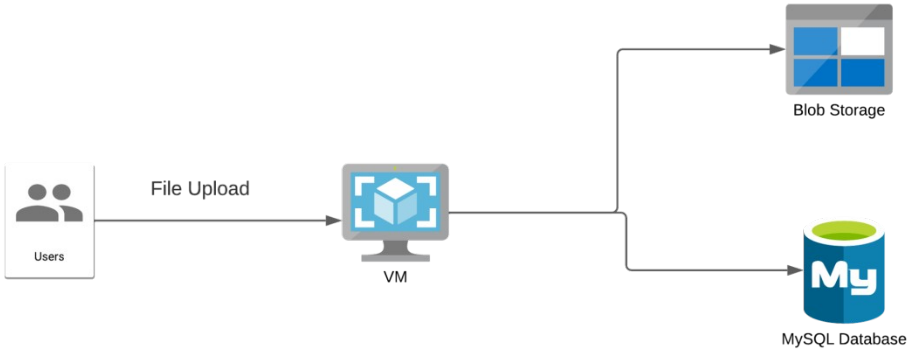
Declaration	
Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real-life situations where customers seldom give crystal clear requirements and ask unambiguous questions.	
I have read the above statement and agree to these conditions	
I AGREE	Nikita Sanjay Agarwal
	<Enter your name above this line to indicate that you are in agreement>

Instructions
Every screenshot requested in this workbook is compulsory and carries 1 marks
Your Azure account ID must be clearly visible in every screenshot using the Azure portal; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed.
All screenshots must be in the order mentioned under "Expected Screenshots" for every step
DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.
The file should be renamed in the format BATCH_FIRSTNAME_LASTNAME_PROJECT1. For example: PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf

Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you’ve completed your lab or project is always necessary. This is true whether you’re doing a lab or implementing a project at your workplace.
After completing the lab, make sure to delete each resource created in reverse chronological order.

Architecture diagram

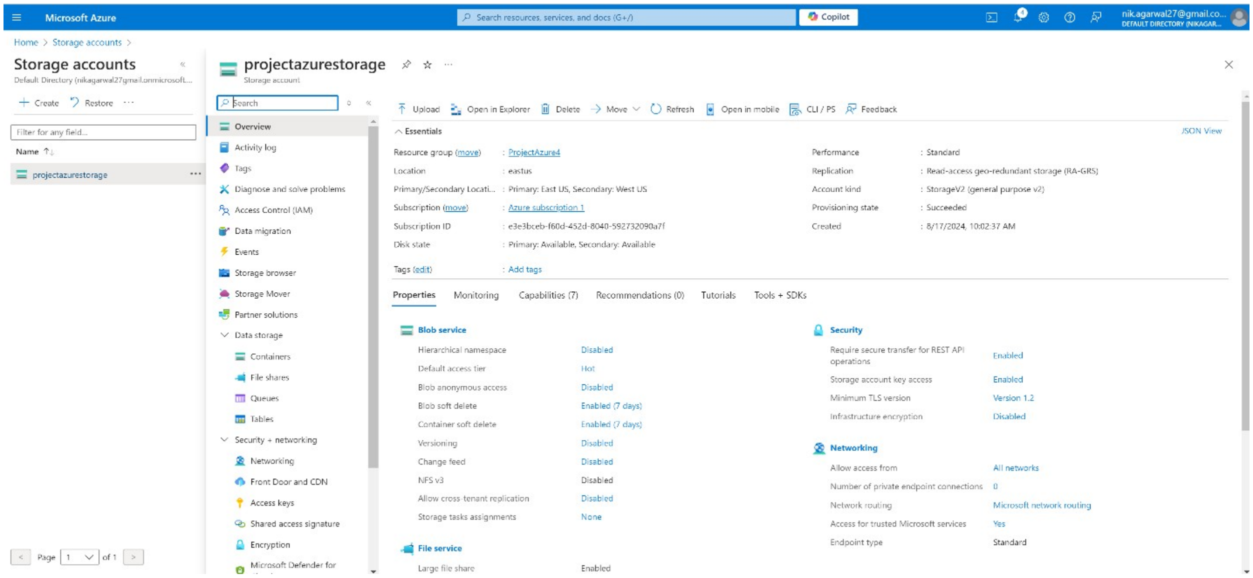


Architecture Implementation	
1	Upload the custom program and provided text file to a VM created using Ubuntu
2	Create a MySQL server using Azure Database service
3	Create a database inside the MySQL server created above
4	Running the custom program will convert the text file into a CSV file, upload it to blob storage and send the data to the MySQL server.

Step 1: Create resources

Step number	a
Step name	Creation of Resource group and blob storage
Instructions	1) Create a resource group using any region. Use the same resource group for all resources created in this exercise. 2) Navigate to Storage Accounts and Click on Create. 3) Enter a name and region for the Storage Account. The rest of the fields can be left to their default values. 4) Once the storage account has been created, navigate to the resource. 5) Using the menu on the left, navigate to Access Keys and note down the Connection String value for key 1. You may have to click on the Show keys button at the top of the screen to make the values visible.
Expected screenshots	1) Screen showing created storage account

<Insert screenshot for a(1) here>



Essentials

Resource group (move)	: ProjectAzure4
Location	: eastus
Primary/Secondary Locati...	: Primary: East US, Secondary: West US
Subscription (move)	: Azure subscription 1
Subscription ID	: e3e3bceb-f60d-452d-8040-592732090a7f
Disk state	: Primary: Available, Secondary: Available
Tags (edit)	: Add tags
Performance	: Standard
Replication	: Read-access geo-redundant storage (RA-GRS)
Account kind	: StorageV2 (general purpose v2)
Provisioning state	: Succeeded
Created	: 8/17/2024, 10:02:37 AM

Blob service

Hierarchical namespace	Disabled
Default access tier	Hot
Blob anonymous access	Disabled
Blob soft delete	Enabled (7 days)
Container soft delete	Enabled (7 days)
Versioning	Disabled
Change feed	Disabled
NFS v3	Disabled
Allow cross-tenant replication	Disabled
Storage tasks assignments	None

File service

Large file share	Enabled

Security

Require secure transfer for REST API operations	Enabled
Storage account key access	Enabled
Minimum TLS version	Version 1.2
Infrastructure encryption	Disabled

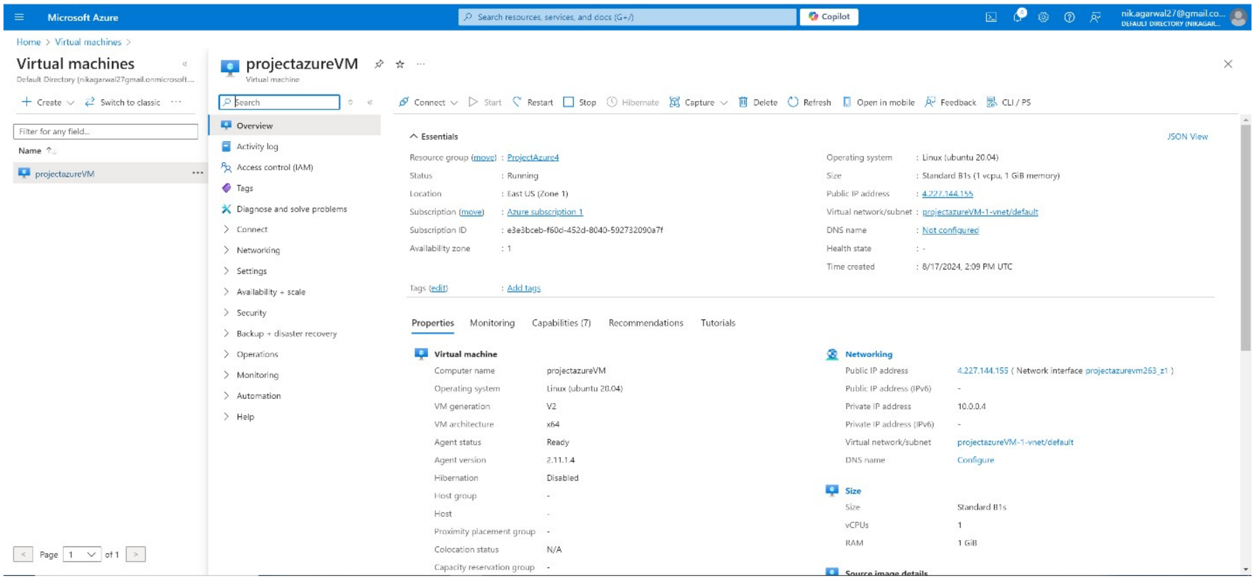
Networking

Allow access from	All networks
Number of private endpoint connections	0
Network routing	Microsoft network routing
Access for trusted Microsoft services	Yes
Endpoint type	Standard

Step number	b
Step name	Creation of VM

Instructions	1) Navigate to Virtual Machines 2) Create a VM using the Ubuntu 20.04 image. 3) Make sure that port 22 is enabled in inbound ports for the VM during creation. 4) User name - ubuntu 5) Authentication type needs to be SSH public key. 6) The rest of the fields can be left to their default values. Click on Create.
Expected screenshots	1) Created VM

<Insert screenshot for b(1) here>



projectazureVM

Essentials

Resource group (move) : ProjectAzure4
Status : Running
Location : East US (Zone 1)
Subscription (move) : Azure subscription 1
Subscription ID : e3e3bceb-f60d-452d-8040-592732090a7f
Availability zone : 1

Operating system : Linux (ubuntu 20.04)
Size : Standard B1s (1 vcpu, 1 GiB memory)
Public IP address : 4.227.144.155
Virtual network/subnet : projectazureVM-1-vnet/default
DNS name : Not configured
Health state : -
Time created : 8/17/2024, 2:09 PM UTC

Tags (edit) : Add tags

Virtual machine
Computer name projectazureVM
Operating system Linux (ubuntu 20.04)
VM generation V2
VM architecture x64
Agent status Ready
Agent version 2.11.1.4
Hibernation Disabled
Host group -
Host -
Proximity placement group -
Colocation status N/A
Capacity reservation group -

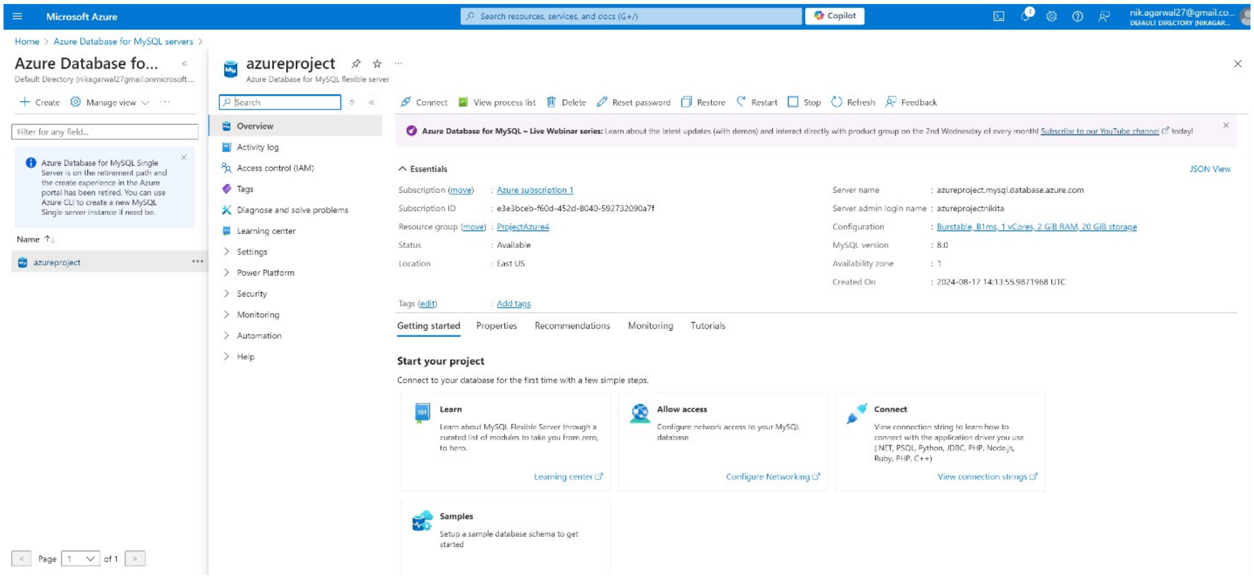
Networking
Public IP address 4.227.144.155 (Network interface projectazurevm263_z1)
Public IP address (IPv6) -
Private IP address 10.0.0.4
Private IP address (IPv6) -
Virtual network/subnet projectazureVM-1-vnet/default
DNS name Configure

Size
Size Standard B1s
vCPUs 1
RAM 1 GiB

Source image details

Step number	c
Step name	Creation of MySQL server
Instructions	1) Navigate to Azure Database for MySQL servers using the search bar at the top of the Azure portal and click on Create 2) Select the Flexible Server option 3) Enter the server name of choice and the username and password. Make sure to note down the username and password you have entered. 4) Under networking, ensure public access is allowed and check the box “Allow public access from any Azure service within Azure to this server” 5) The rest of the fields can be left to their default values. Click on Create. 6) Once the server has been created, navigate to the resource and note down the Server Name field present in the Overview section.
Expected screenshots	1) Overview screen of the created database server.

<Insert screenshot for c(1) here>



Azure Database for MySQL Single Server is on the retirement path and the create experience in the Azure portal has been retired. You can use Azure CLI to create a new MySQL Single server instance if need be.

Azure Database for MySQL – Live Webinar series: Learn about the latest updates (with demos) and interact directly with product group on the 2nd Wednesday of every month! Subscribe to our YouTube channel today!

Subscription (move) : Azure subscription 1
Subscription ID : e3e3bceb-f60d-452d-8040-592732090a7f
Resource group (move) : ProjectAzure4
Status : Available
Location : East US
Tags (edit) : Add tags

Server name : azureproject.mysql.database.azure.com
Server admin login name : azureprojectnikita
Configuration : Burstable, B1ms, 1 vCores, 2 GiB RAM, 20 GiB storage
MySQL version : 8.0
Availability zone : 1
Created On : 2024-08-17 14:13:55.9871968 UTC

Start your project

Connect to your database for the first time with a few simple steps.

Learn
Learn about MySQL Flexible Server through a curated list of modules to take you from zero, to hero.
Learning center

Allow access
Configure network access to your MySQL database
Configure Networking

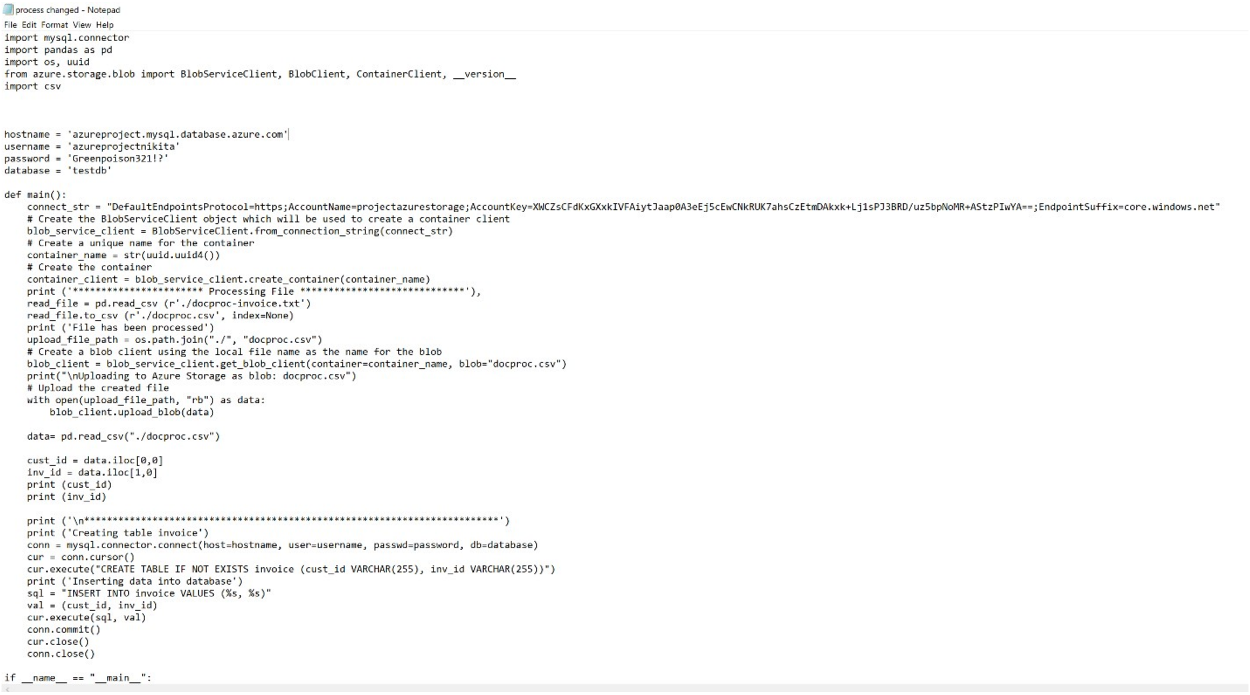
Connect
View connection string to learn how to connect with the application driver you use (.NET, PSQL, Python, JDBC, PHP, Node.js, Ruby, PHP, C++)
View connection strings

Samples
Setup a sample database schema to get started

Step 2: Run the custom program in the VM

Step number	a
Step name	Environment setup
Instructions	1) Download the invoice file and python script provided with this workbook. 2) Open the Python script using your text editor or code editor of choice 3) Replace the values in lines 9,10,11, and 15 with the database server name, username, password, and storage account connection string(received in step 1(a)(5)) respectively. Save the file. 4) Copy both the files to the VM using the scp command. *scp -i <pem file> <file to be copied> ubuntu@<public IP of VM>:/home/ubuntu* You will need to run the scp command twice, once for each file. 5) SSH into the VM using your SSH client of choice and run the below commands to set up the environment *sudo apt update* *sudo apt install python3* *sudo apt install python3-pip* *sudo pip3 install pandas* *sudo pip3 install azure-storage-blob* *sudo pip3 install mysql-connector-python* *sudo apt install mysql-client-core-8.0*
Expected screenshots	1) Screenshot of the process.py file after completing Step3 above 2) Copying the files using scp 3) Screenshot after completing Step 5 above.

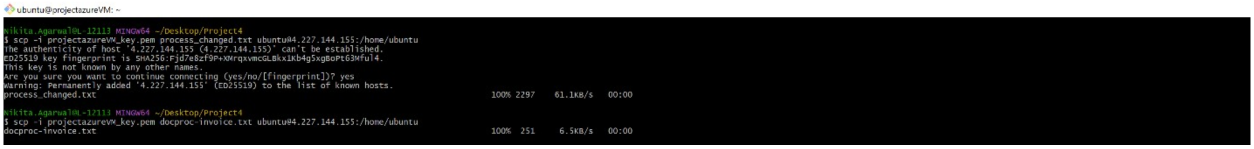
<Insert screenshot for a(1) here>



process changed - Notepad

```
import mysql.connector
import pandas as pd
import os, uuid
from azure.storage.blob import BlobServiceClient, BlobClient, ContainerClient, __version__
import csv



hostname = 'azureproject.mysql.database.azure.com'
username = 'azureprojectnikita'
password = 'Greenpoison321!?'
database = 'testdb'

def main():
    connect_str = "DefaultEndpointsProtocol=https;AccountName=projectazurestorage;AccountKey=XWCZsCFdKxGXxkIVFAiytJaap0A3eEj5cEwCNkRUK7ahsCzEtmDAkxk+Lj1sPJ3BRD/uz5bpNoMR+AStzPIwYA==;EndpointSuffix=core.windows.net"
    # Create the BlobServiceClient object which will be used to create a container client
    blob_service_client = BlobServiceClient.from_connection_string(connect_str)
    # Create a unique name for the container
    container_name = str(uuid.uuid4())
    # Create the container
    container_client = blob_service_client.create_container(container_name)
    print ('********************** Processing File ****************************'),
    read_file = pd.read_csv (r'./docproc-invoice.txt')
    read_file.to_csv (r'./docproc.csv', index=None)
    print ('File has been processed')
    upload_file_path = os.path.join("./", "docproc.csv")
    # Create a blob client using the local file name as the name for the blob
    blob_client = blob_service_client.get_blob_client(container=container_name, blob="docproc.csv")
    print("\nUploading to Azure Storage as blob: docproc.csv")
    # Upload the created file
    with open(upload_file_path, "rb") as data:
        blob_client.upload_blob(data)

    data= pd.read_csv("./docproc.csv")

    cust_id = data.iloc[0,0]
    inv_id = data.iloc[1,0]
    print (cust_id)
    print (inv_id)

    print ('\n****************************************************************************')
    print ('Creating table invoice')
    conn = mysql.connector.connect(host=hostname, user=username, passwd=password, db=database)
    cur = conn.cursor()
    cur.execute("CREATE TABLE IF NOT EXISTS invoice (cust_id VARCHAR(255), inv_id VARCHAR(255))")
    print ('Inserting data into database')
    sql = "INSERT INTO invoice VALUES (%s, %s)"
    val = (cust_id, inv_id)
    cur.execute(sql, val)
    conn.commit()
    cur.close()
    conn.close()

if __name__ == "__main__":
```

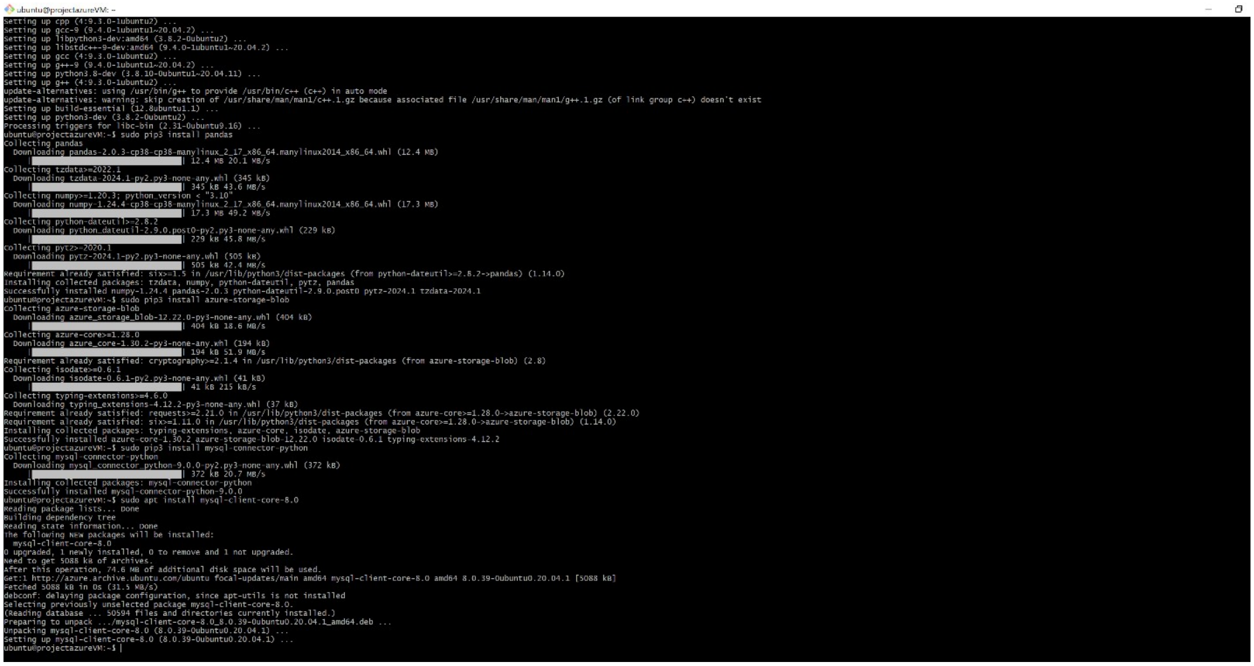
<Insert screenshot for a(2) here>



ubuntu@projectazureVM: ~

```
Nikita.Agarwal@L-12113 MINGW64 ~/Desktop/Project4
$ scp -i projectazureVM_key.pem process_changed.txt ubuntu@4.227.144.155:/home/ubuntu
The authenticity of host '4.227.144.155 (4.227.144.155)' can't be established.
ED25519 key fingerprint is SHA256:Fjd7e8zf9P+XMrqxvmcGLBkx1Kb4g5xgBoPt63Mful4.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '4.227.144.155' (ED25519) to the list of known hosts.
process_changed.txt                                                                100% 2297    61.1KB/s   00:00

Nikita.Agarwal@L-12113 MINGW64 ~/Desktop/Project4
$ scp -i projectazureVM_key.pem docproc-invoice.txt ubuntu@4.227.144.155:/home/ubuntu
docproc-invoice.txt                                                                100%  251     6.5KB/s   00:00
```

<Insert screenshot for a(3) here>

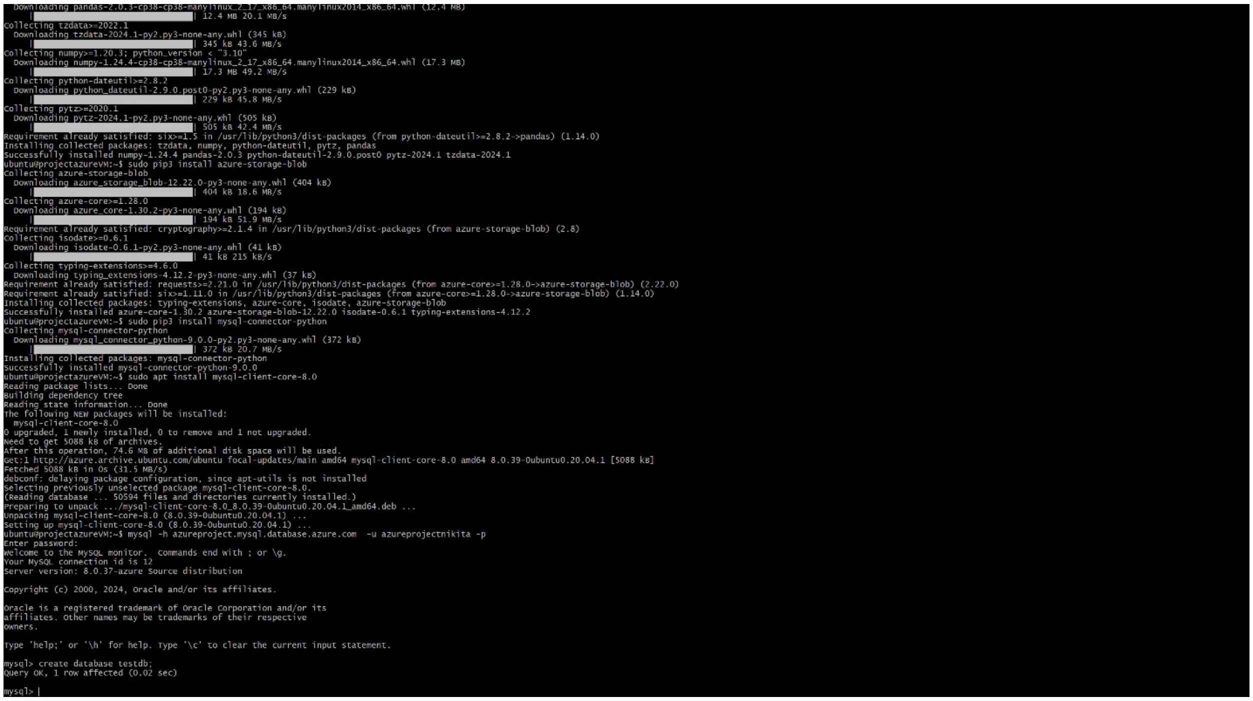


ubuntu@projectazureVM: ~

```
Setting up cpp (4:9.3.0-1ubuntu2) ...
Setting up gcc-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up libpython3-dev:amd64 (3.8.2-0ubuntu2) ...
Setting up libstdc++-9-dev:amd64 (9.4.0-1ubuntu1~20.04.2) ...
Setting up gcc (4:9.3.0-1ubuntu2) ...
Setting up g++-9 (9.4.0-1ubuntu1~20.04.2) ...
Setting up python3.8-dev (3.8.10-0ubuntu1~20.04.11) ...
Setting up g++ (4:9.3.0-1ubuntu2) ...
update-alternatives: using /usr/bin/g++ to provide /usr/bin/c++ (c++) in auto mode
update-alternatives: warning: skip creation of /usr/share/man/man1/c++.1.gz because associated file /usr/share/man/man1/g++.1.gz (of link group c++) doesn't exist
Setting up build-essential (12.8ubuntu1.1) ...
Setting up python3-dev (3.8.2-0ubuntu2) ...
Processing triggers for libc-bin (2.31-0ubuntu9.16) ...
ubuntu@projectazureVM:~$ sudo pip3 install pandas
Collecting pandas
  Downloading pandas-2.0.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.4 MB)
     |████████████████████████████████| 12.4 MB 20.1 MB/s
Collecting tzdata>=2022.1
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
     |████████████████████████████████| 345 kB 43.6 MB/s
Collecting numpy>=1.20.3; python_version < "3.10"
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
     |████████████████████████████████| 17.3 MB 49.2 MB/s
Collecting python-dateutil>=2.8.2
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     |████████████████████████████████| 229 kB 45.8 MB/s
Collecting pytz>=2020.1
  Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
     |████████████████████████████████| 505 kB 42.4 MB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2->pandas) (1.14.0)
Installing collected packages: tzdata, numpy, python-dateutil, pytz, pandas
Successfully installed numpy-1.24.4 pandas-2.0.3 python-dateutil-2.9.0.post0 pytz-2024.1 tzdata-2024.1
ubuntu@projectazureVM:~$ sudo pip3 install azure-storage-blob
Collecting azure-storage-blob
  Downloading azure_storage_blob-12.22.0-py3-none-any.whl (404 kB)
     |████████████████████████████████| 404 kB 18.6 MB/s
Collecting azure-core>=1.28.0
  Downloading azure_core-1.30.2-py3-none-any.whl (194 kB)
     |████████████████████████████████| 194 kB 51.9 MB/s
Requirement already satisfied: cryptography>=2.1.4 in /usr/lib/python3/dist-packages (from azure-storage-blob) (2.8)
Collecting isodate>=0.6.1
  Downloading isodate-0.6.1-py2.py3-none-any.whl (41 kB)
     |████████████████████████████████| 41 kB 215 kB/s
Collecting typing-extensions>=4.6.0
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Requirement already satisfied: requests>=2.21.0 in /usr/lib/python3/dist-packages (from azure-core>=1.28.0->azure-storage-blob) (2.22.0)
Requirement already satisfied: six>=1.11.0 in /usr/lib/python3/dist-packages (from azure-core>=1.28.0->azure-storage-blob) (1.14.0)
Installing collected packages: typing-extensions, azure-core, isodate, azure-storage-blob
Successfully installed azure-core-1.30.2 azure-storage-blob-12.22.0 isodate-0.6.1 typing-extensions-4.12.2
ubuntu@projectazureVM:~$ sudo pip3 install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.0.0-py2.py3-none-any.whl (372 kB)
     |████████████████████████████████| 372 kB 20.7 MB/s
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.0.0
ubuntu@projectazureVM:~$ sudo apt install mysql-client-core-8.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  mysql-client-core-8.0
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 5088 kB of archives.
After this operation, 74.6 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.20.04.1 [5088 kB]
Fetched 5088 kB in 0s (31.5 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 50594 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.39-0ubuntu0.20.04.1_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.39-0ubuntu0.20.04.1) ...
Setting up mysql-client-core-8.0 (8.0.39-0ubuntu0.20.04.1) ...
ubuntu@projectazureVM:~$ |
```

Step number	b
Step name	Configure the database
Instructions	1) Run the following command in the SSH terminal after substituting the database server name and username. *mysql -h <database server name> -u <database_username> -p* *Note : In case of a database server connection error , ensure that the firewall is configured correctly using the below link* https://learn.microsoft.com/en-us/azure/mysql/single-server/how-to-manage-firewall-using-portal 2) Enter the password when prompted. 3) Enter the following command *create database testdb;* 4) Enter *exit* to exit out of the MySQL environment.
Expected screenshots	1) Screenshot after completing Step 3 above

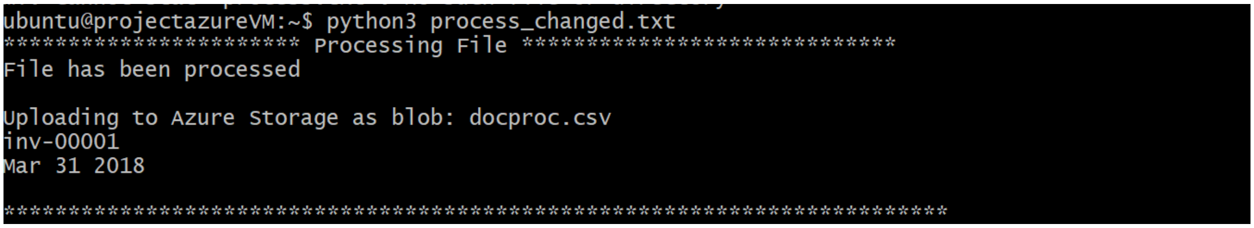
<Insert screenshot for b(1) here>



```
  Downloading pandas-2.0.3-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.4 MB)
     |████████████████████████████████| 12.4 MB 20.1 MB/s
Collecting tzdata>=2022.1
  Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
     |████████████████████████████████| 345 kB 43.6 MB/s
Collecting numpy>=1.20.3; python_version < "3.10"
  Downloading numpy-1.24.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
     |████████████████████████████████| 17.3 MB 49.2 MB/s
Collecting python-dateutil>=2.8.2
  Downloading python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
     |████████████████████████████████| 229 kB 45.8 MB/s
Collecting pytz>=2020.1
  Downloading pytz-2024.1-py2.py3-none-any.whl (505 kB)
     |████████████████████████████████| 505 kB 42.4 MB/s
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2->pandas) (1.14.0)
Installing collected packages: tzdata, numpy, python-dateutil, pytz, pandas
Successfully installed numpy-1.24.4 pandas-2.0.3 python-dateutil-2.9.0.post0 pytz-2024.1 tzdata-2024.1
ubuntu@projectazureVM:~$ sudo pip3 install azure-storage-blob
Collecting azure-storage-blob
  Downloading azure_storage_blob-12.22.0-py3-none-any.whl (404 kB)
     |████████████████████████████████| 404 kB 18.6 MB/s
Collecting azure-core>=1.28.0
  Downloading azure_core-1.30.2-py3-none-any.whl (194 kB)
     |████████████████████████████████| 194 kB 51.9 MB/s
Requirement already satisfied: cryptography>=2.1.4 in /usr/lib/python3/dist-packages (from azure-storage-blob) (2.8)
Collecting isodate>=0.6.1
  Downloading isodate-0.6.1-py2.py3-none-any.whl (41 kB)
     |████████████████████████████████| 41 kB 215 kB/s
Collecting typing-extensions>=4.6.0
  Downloading typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Requirement already satisfied: requests>=2.21.0 in /usr/lib/python3/dist-packages (from azure-core>=1.28.0->azure-storage-blob) (2.22.0)
Requirement already satisfied: six>=1.11.0 in /usr/lib/python3/dist-packages (from azure-core>=1.28.0->azure-storage-blob) (1.14.0)
Installing collected packages: typing-extensions, azure-core, isodate, azure-storage-blob
Successfully installed azure-core-1.30.2 azure-storage-blob-12.22.0 isodate-0.6.1 typing-extensions-4.12.2
ubuntu@projectazureVM:~$ sudo pip3 install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.0.0-py2.py3-none-any.whl (372 kB)
     |████████████████████████████████| 372 kB 20.7 MB/s
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.0.0
ubuntu@projectazureVM:~$ sudo apt install mysql-client-core-8.0
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  mysql-client-core-8.0
0 upgraded, 1 newly installed, 0 to remove and 1 not upgraded.
Need to get 5088 kB of archives.
After this operation, 74.6 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu focal-updates/main amd64 mysql-client-core-8.0 amd64 8.0.39-0ubuntu0.20.04.1 [5088 kB]
Fetched 5088 kB in 0s (31.5 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 50594 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.39-0ubuntu0.20.04.1_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.39-0ubuntu0.20.04.1) ...
Setting up mysql-client-core-8.0 (8.0.39-0ubuntu0.20.04.1) ...
ubuntu@projectazureVM:~$ mysql -h azureproject.mysql.database.azure.com  -u azureprojectnikita -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.37-azure Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database testdb;
Query OK, 1 row affected (0.02 sec)

mysql> |
```

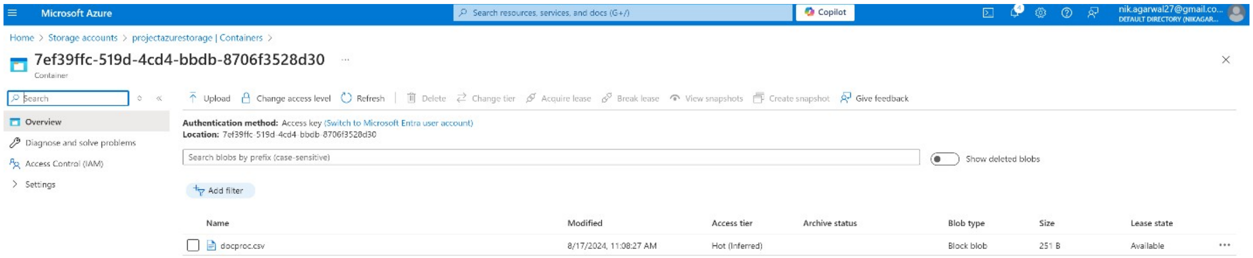
Step number	c
Step name	Running the custom program
Instructions	1) Run the program using the command *python3 process.py* 2) Navigate to the storage account using the Azure portal. Select the Containers option from the menu on the left and select the created container. Verify that it contains a generated CSV file 3) Run the following command in the SSH terminal after substituting the database server name and username. *mysql -h <database server name> -u <database_username> -p* 4) Enter the password when prompted. 5) Run the following commands to verify that the data has been entered into the database *use testdb;* *select * from invoice;* 6) Enter *exit* to exit out of the MySQL environment.
Expected screenshots	1) Running the custom Python program 2) Created CSV file in Blob Storage 3) Screenshot after running step 5 above

<Insert screenshot for c(1) here>



```
ubuntu@projectazureVM:~$ python3 process_changed.txt
*********************** Processing File *****************************
File has been processed

Uploading to Azure Storage as blob: docproc.csv
inv-00001
Mar 31 2018

*************************************************************************
```

<Insert screenshot for c(2) here>



Home > Storage accounts > projectazurestorage | Containers >

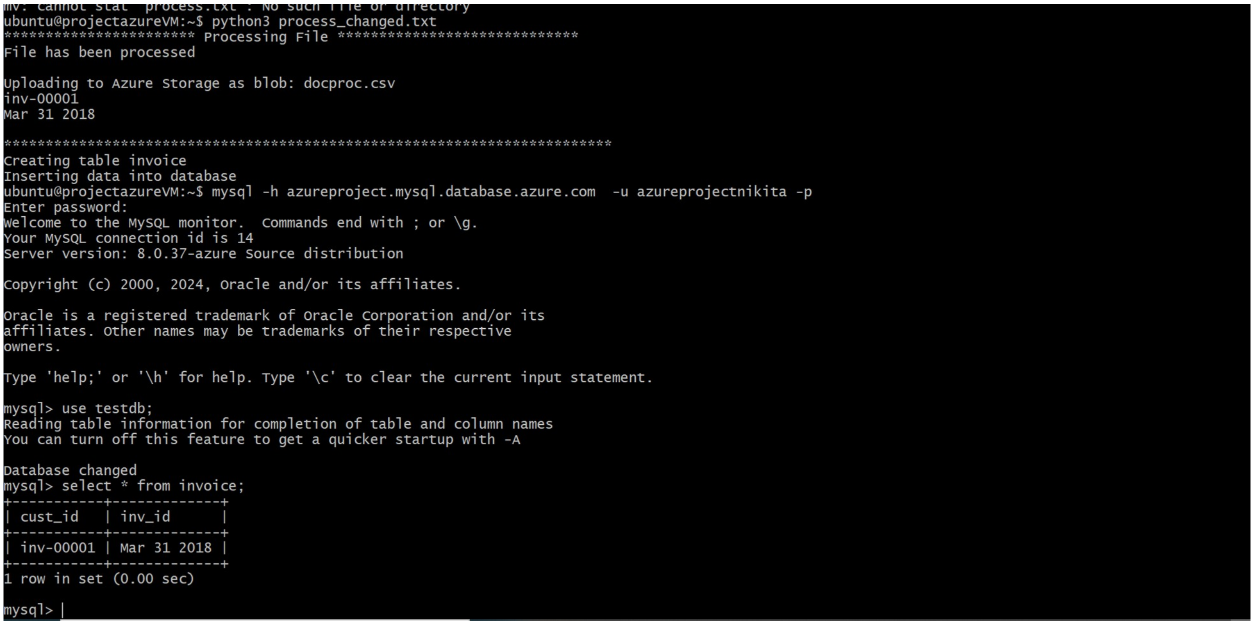
7ef39ffc-519d-4cd4-bbdb-8706f3528d30

Container

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: 7ef39ffc 519d 4cd4 bbdb 8706f3528d30

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
□ docproc.csv	8/17/2024, 11:08:27 AM	Hot (Inferred)		Block blob	251 B	Available

<Insert screenshot for c(3) here>



```
mv: cannot stat 'process.txt': No such file or directory
ubuntu@projectazureVM:~$ python3 process_changed.txt
*********************** Processing File ****************************
File has been processed

Uploading to Azure Storage as blob: docproc.csv
inv-00001
Mar 31 2018

*************************************************************************
Creating table invoice
Inserting data into database
ubuntu@projectazureVM:~$ mysql -h azureproject.mysql.database.azure.com  -u azureprojectnikita -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.37-azure Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use testdb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from invoice;
+-----------+-------------+
| cust_id   | inv_id      |
+-----------+-------------+
| inv-00001 | Mar 31 2018 |
+-----------+-------------+
1 row in set (0.00 sec)

mysql> |
```

Answer the following questions

Q1 At which level are lifecycle management rules for Blob storage applied?

a) File Level

b) Blob Level

c) Storage account level

d) Subscription level

Enter your answer here

Storage Account level

Q2 Which of the following is not true about the Premium performance storage tier in Azure?

a) Only Hot and Cool storage tiers are available

b) Supports only LRS and ZRS

c) Data is stored on SSDs

d) Geo-redundancy is not possible.

Enter your answer here

Only Hot and Cool storage tiers are available

Q3 Which of the following Azure SQL deployment options should you use when the number of databases to be created is variable.

a) On-premises deployment of Azure SQL

b) Azure SQL Database

c) Managed DB instance

d) None of these

Enter your answer here

Azure SQL Database

Q4 Which of the following Azure SQL purchasing models would be more beneficial for BYOL (Bring-Your-Own-License) use-cases?

a) Depends on the license type

b) Does not matter

c) vCore based

d) DTU based

Enter your answer here

vCore based

Q5 Why was port 3306 not enabled for incoming connections in the VM in this exercise?

a) The port is only required to be enabled on the database server

b) Azure MySQL uses a different port

c) Port 3306 has no bearing on this exercise.

d) None of these

Enter your answer here

The port is only required to be enabled on the database server

Grades distribution	
MCQs	10 (2 points each)
Implementation screenshots	10 points (1 point each)
Total	20 points