

Azure Project 2

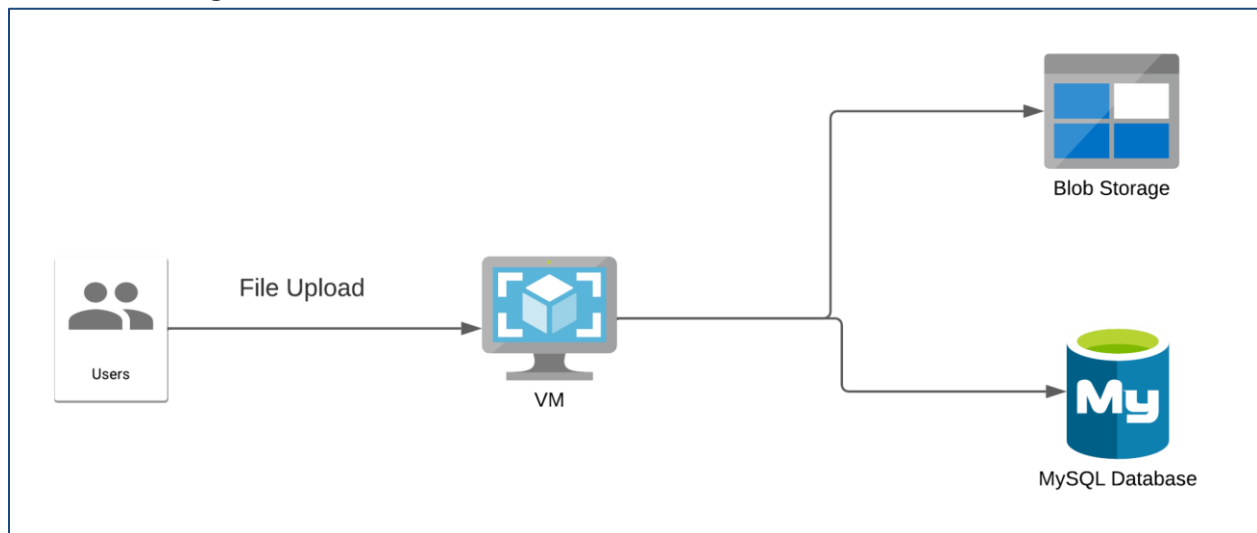
Building an Automated Business Process using Managed Services on a Public Cloud

In the connected world, it is imperative that the organizations be interlinked with the customers and vendors. This process has been very sluggish, manual, batch-based and prone to failures. Such Integration design has lead to impaired decision making and delays in the detection of fraudulent actions. This project created an automated, event-based real-time process using managed cloud services that do not have these limitations.

Skills and Tools

Azure VM, Python, Blob Storage, Azure SQL, Azure SDK for Python

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: Creation of resources

Step number	a
Step name	Creation of Resource group and blob storage
Instructions	<ol style="list-style-type: none">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>

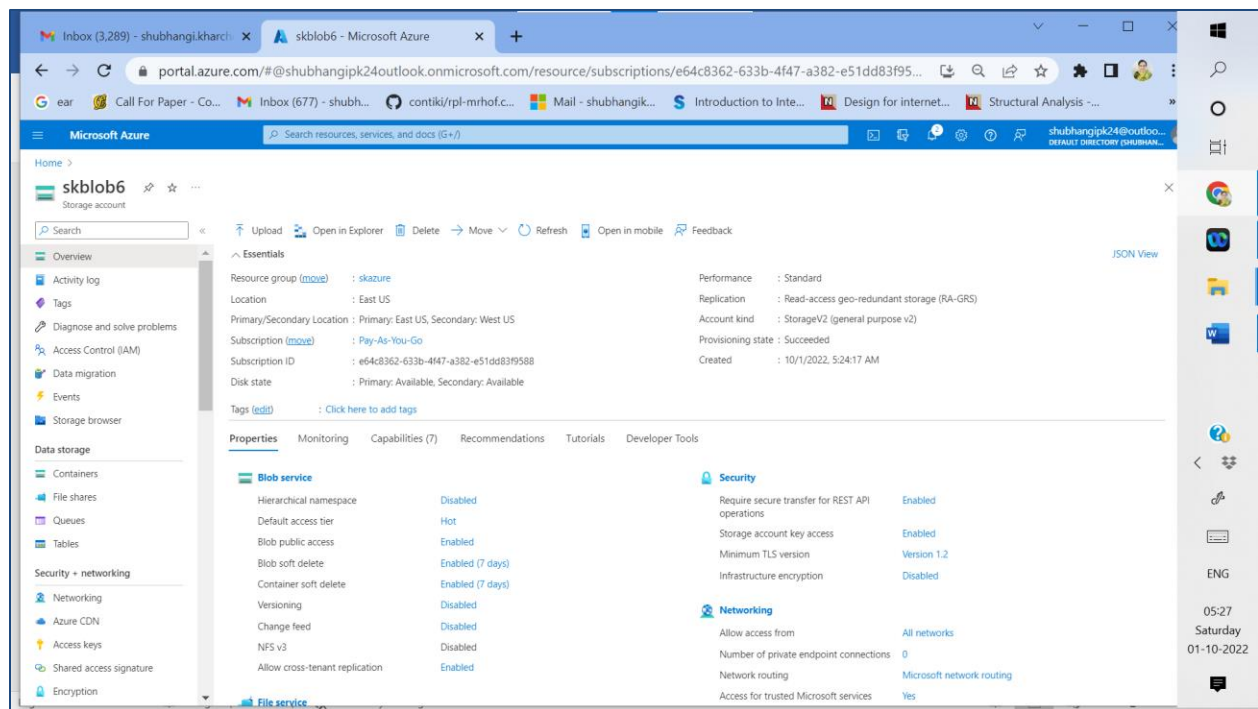


Fig 1: Screen showing created storage account

Step number	b
-------------	---

Step name	Creation of VM
Instructions	<ol style="list-style-type: none"> 1) Navigate to Virtual Machines 2) Create a VM using the Ubuntu 18.04 image. 3) Make sure that port 22 is enabled in inbound ports for the VM during creation. 4) Authentication type needs to be SSH public key. Make sure you note down the value you enter in the Username field. 5) 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>

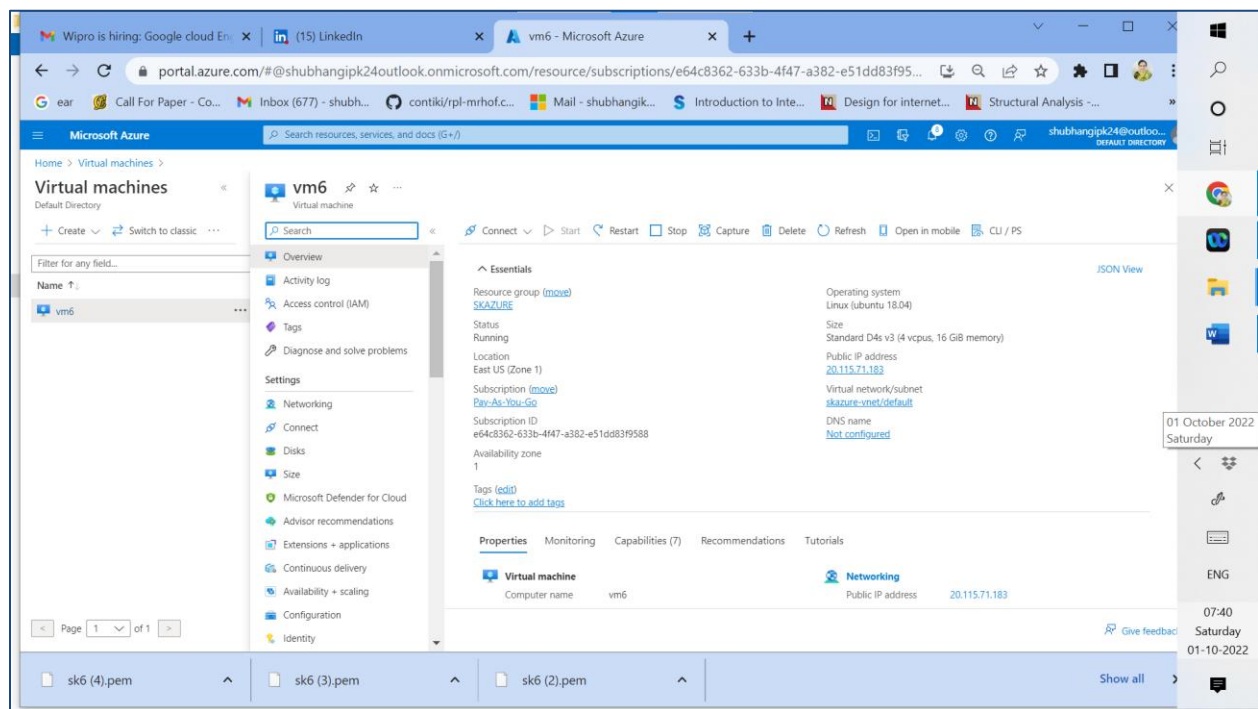


Fig 2: Created VM

Step number	c
-------------	---

Step name	Creation of MySQL server
Instructions	<ol style="list-style-type: none"> 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 Single 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) The rest of the fields can be left to their default values. Click on Create. 5) 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>

Server details

Enter required settings for this server, including picking a location and configuring the compute and storage resources.

Server name * ⓘ

mysql24 ✓

Data source * ⓘ

None Backup

Location * ⓘ

(US) East US ▼

Version * ⓘ

5.7 ▼

Compute + storage ⓘ

General Purpose

4 vCores, 100 GB storage

[Configure server](#)

Administrator account

Admin username * ⓘ

shubhangi24 ✓

Password * ⓘ

..... ✓

Confirm password *

..... ✓

✓ Password and confirm password must match.

Server name: mysql24.mysql.database.azure.com

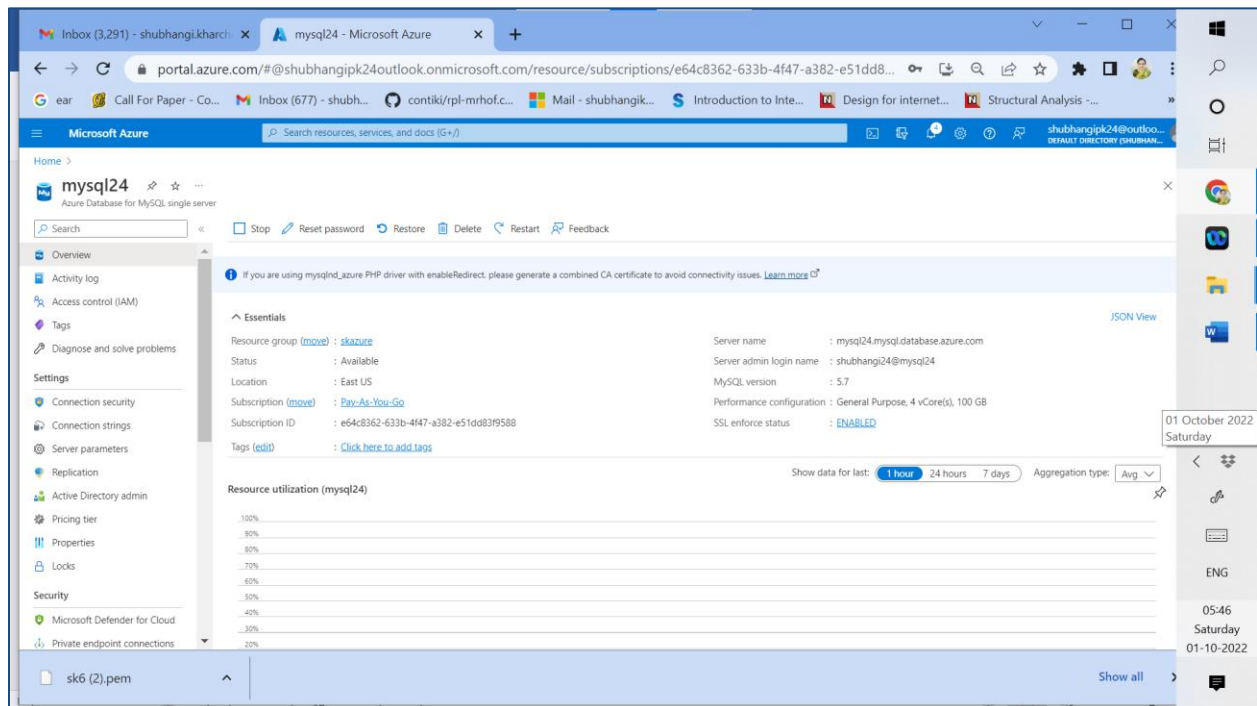


Fig 3 : Overview screen of the created database server.

Step 2: Run the custom program in the VM

Step number	a
Step name	Environment setup
Instructions	<ol style="list-style-type: none">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 choice3) 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. <code>scp -i <pem file> <file to be copied> <VM username>@<public IP of VM>:/home/ubuntu</code> 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

```

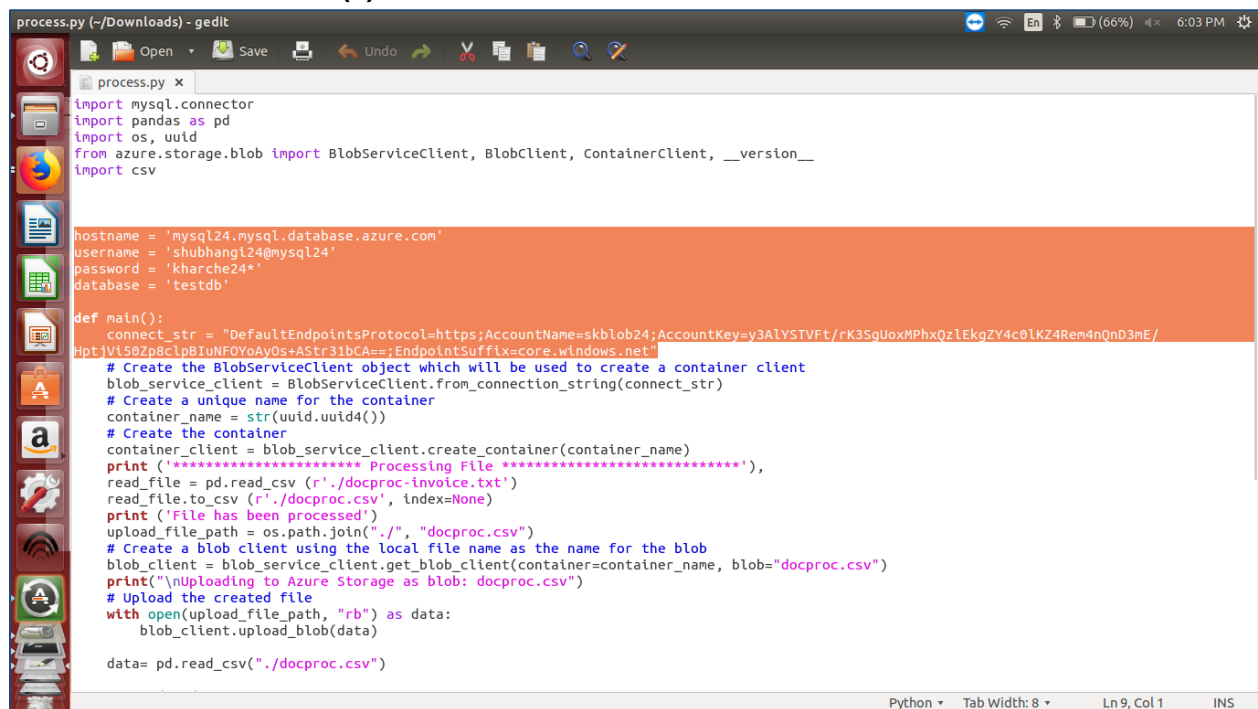
sudo apt install python3
sudo apt install python3-pip
sudo pip3 install pandas
sudo pip3 install azure-storage-blob
sudo pip3 install mysql-connector-python
sudo apt install mysql-client-core-5.7

```

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

hostname = 'mysql24.mysql.database.azure.com'
username = 'shubhangi24@mysql24'
password = 'kharche24*'
database = 'testdb'

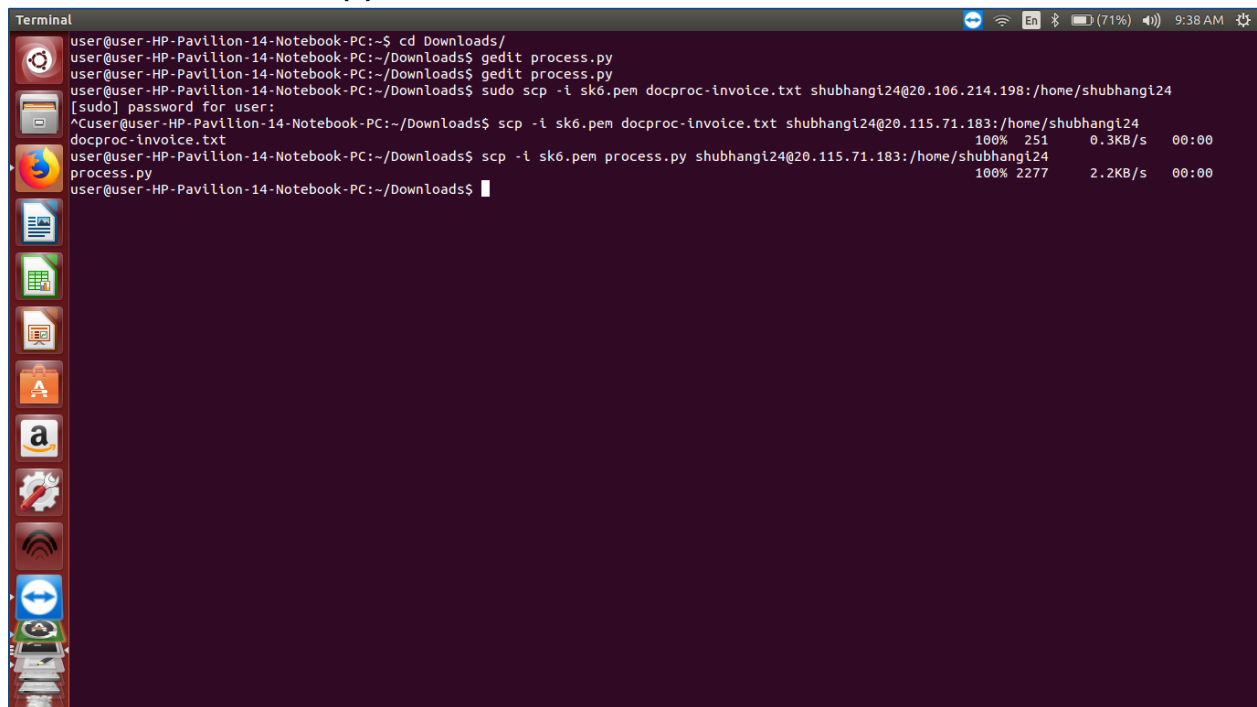
def main():
    connect_str = "DefaultEndpointsProtocol=https;AccountName=skblob24;AccountKey=y3ALYSTVft/rK3SgUoxMPHxQz1EkqZY4c0LKZ4Rem4nQnD3nE/
Hpt1Vi50Zp8clpBIuNFOYoAyOs+AStr31bCA==;EndpointSuffix=core.windows.net"
    # Create the BlobServiceClient object which will be used to create a container client
    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")

```

Fig 4: Screenshot of the process.py file after completing Step3 above (process.py file edited)

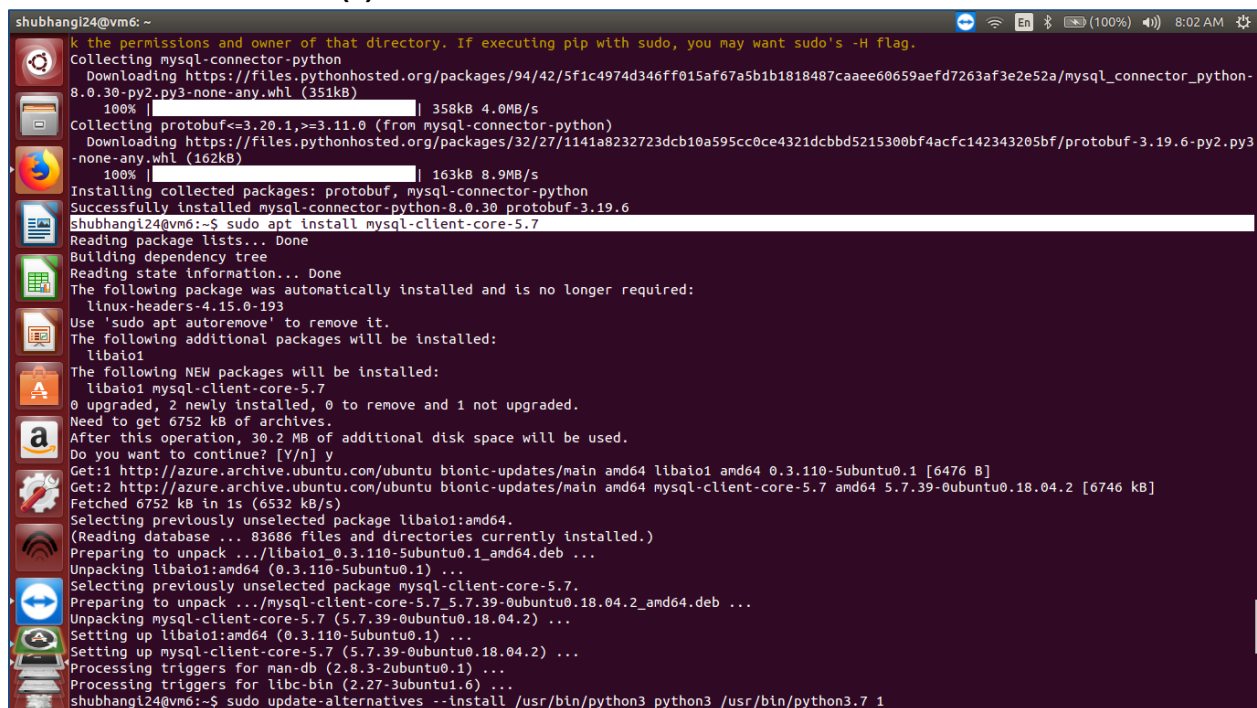
<Insert screenshot for a(2) here>

A terminal window on a host named 'user@user-HP-Pavilion-14-Notebook-PC'. The user navigates to the 'Downloads' directory and uses 'gedit' to edit 'process.py'. Then, they use 'sudo scp' to copy 'docproc-invoice.txt' and 'process.py' to a remote VM named 'shubhangi24' at IP '20.106.214.198'. The first copy of 'docproc-invoice.txt' (251 bytes) is successful. The second copy of 'process.py' (2277 bytes) is also successful. The terminal shows progress bars and transfer speeds for both files.

```
user@user-HP-Pavilion-14-Notebook-PC:~$ cd Downloads/
user@user-HP-Pavilion-14-Notebook-PC:~/Downloads$ gedit process.py
user@user-HP-Pavilion-14-Notebook-PC:~/Downloads$ sudo scp -i sk6.pem docproc-invoice.txt shubhangi24@20.106.214.198:/home/shubhangi24
[sudo] password for user:
^Cuser@user-HP-Pavilion-14-Notebook-PC:~/Downloads$ scp -i sk6.pem docproc-invoice.txt shubhangi24@20.115.71.183:/home/shubhangi24
docproc-invoice.txt                                100% 251      0.3KB/s   00:00
user@user-HP-Pavilion-14-Notebook-PC:~/Downloads$ scp -i sk6.pem process.py shubhangi24@20.115.71.183:/home/shubhangi24
process.py                                          100% 2277     2.2KB/s   00:00
user@user-HP-Pavilion-14-Notebook-PC:~/Downloads$
```

Fig 5: Copying the files using scp (securely copied process.py and docproc-invoice.txt to VM)

<Insert screenshot for a(3) here>

A terminal window on a VM named 'shubhangi24@vm6'. The user is prompted to collect permissions and owner of a directory. They then download a package from pythonhosted.org. The terminal shows progress bars and transfer speeds for the downloaded packages. The user then runs 'sudo apt install mysql-client-core-5.7'. The terminal shows the package lists, dependency tree, and the installation of the requested package. The user then runs 'sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 1'.

```
shubhangi24@vm6: ~
k the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting mysql-connector-python
  Downloading https://files.pythonhosted.org/packages/94/42/5f1c4974d346ff015af67a5b1b1818487caee60659aefd7263af3e2e52a/mysql_connector_python-8.0.30-py2.py3-none-any.whl (351kB)
    100% |#####| 358kB 4.0MB/s
Collecting protobuf<=3.20.1,>=3.11.0 (from mysql-connector-python)
  Downloading https://files.pythonhosted.org/packages/32/27/1141a8232723dcb10a595cc0ce4321dcbdd5215300bf4acfc142343205bf/protobuf-3.19.6-py2.py3-none-any.whl (162kB)
    100% |#####| 163kB 8.9MB/s
Installing collected packages: protobuf, mysql-connector-python
Successfully installed mysql-connector-python-8.0.30 protobuf-3.19.6
shubhangi24@vm6:~$ sudo apt install mysql-client-core-5.7
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  linux-headers-4.15.0-193
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libaio1
The following NEW packages will be installed:
  libaio1 mysql-client-core-5.7
0 upgraded, 2 newly installed, 0 to remove and 1 not upgraded.
Need to get 6752 kB of archives.
After this operation, 30.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libaio1 amd64 0.3.110-5ubuntu0.1 [6476 B]
Get:2 http://azure.archive.ubuntu.com/ubuntu bionic-updates/main amd64 mysql-client-core-5.7 amd64 5.7.39-0ubuntu0.18.04.2 [6746 kB]
Fetched 6752 kB in 1s (6532 kB/s)
Selecting previously unselected package libaio1:amd64.
(Reading database ... 83686 files and directories currently installed.)
Preparing to unpack .../libaio1_0.3.110-5ubuntu0.1_amd64.deb ...
Unpacking libaio1:amd64 (0.3.110-5ubuntu0.1) ...
Selecting previously unselected package mysql-client-core-5.7.
Preparing to unpack .../mysql-client-core-5.7_5.7.39-0ubuntu0.18.04.2_amd64.deb ...
Unpacking mysql-client-core-5.7 (5.7.39-0ubuntu0.18.04.2) ...
Setting up libaio1:amd64 (0.3.110-5ubuntu0.1) ...
Setting up mysql-client-core-5.7 (5.7.39-0ubuntu0.18.04.2) ...
Processing triggers for man-db (2.8.3-2ubuntu1.6) ...
Processing triggers for libc-bin (2.27-3ubuntu1.6) ...
shubhangi24@vm6:~$ sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 1
```

Fig 6: Screenshot after completing Step 5 above.(install mysql-client-core-5.7)

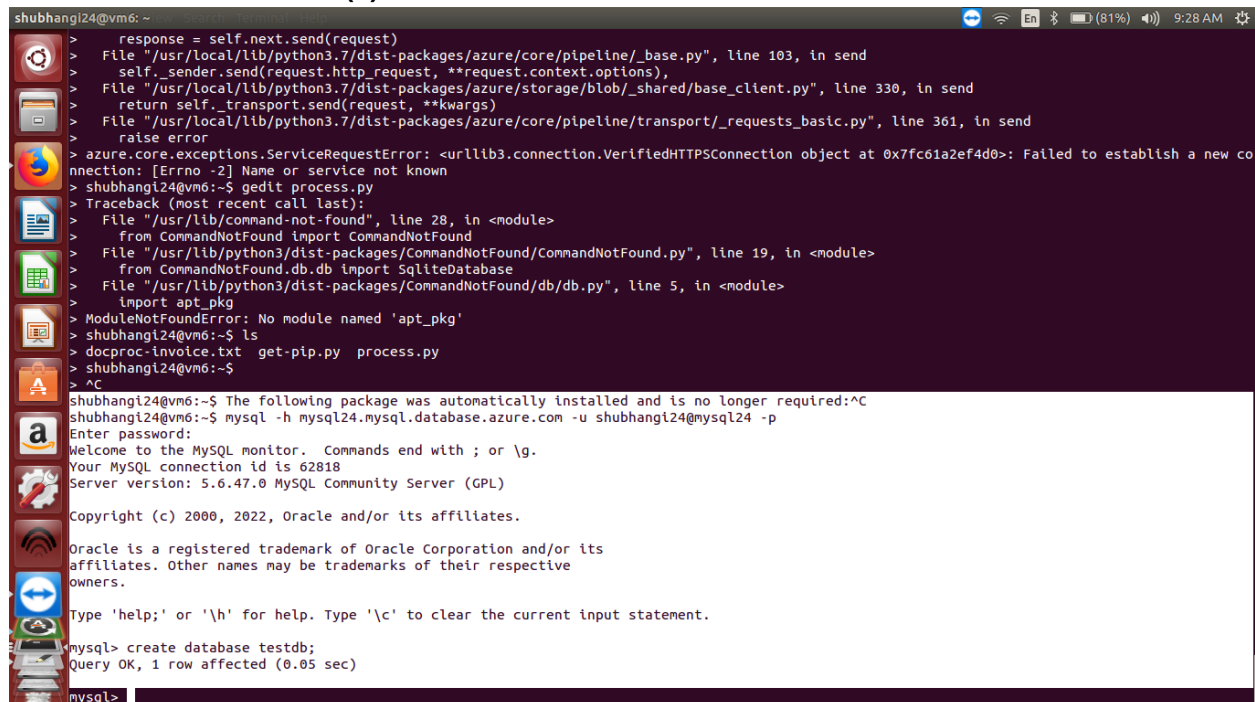
Step number b

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
 - 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>



```
shubhangi24@vm6: ~  
> response = self.next.send(request)  
> File "/usr/local/lib/python3.7/dist-packages/azure/core/pipeline/_base.py", line 103, in send  
> self._sender.send(request.http_request, **request.context.options),  
> File "/usr/local/lib/python3.7/dist-packages/azure/storage/blob/_shared/base_client.py", line 330, in send  
> return self._transport.send(request, **kwargs)  
> File "/usr/local/lib/python3.7/dist-packages/azure/core/pipeline/transport/_requests_basic.py", line 361, in send  
> raise error  
> azure.core.exceptions.ServiceRequestError: <urllib3.connection.VerifiedHTTPSConnection object at 0x7fc61a2ef4d0>: Failed to establish a new co  
nnection: [Errno -2] Name or service not known  
> shubhangi24@vm6:~$ gedit process.py  
> Traceback (most recent call last):  
> File "/usr/lib/command-not-found", line 28, in <module>  
> from CommandNotFound import CommandNotFound  
> File "/usr/lib/python3/dist-packages/CommandNotFound/CommandNotFound.py", line 19, in <module>  
> from CommandNotFound.db.db import SqliteDatabase  
> File "/usr/lib/python3/dist-packages/CommandNotFound/db/db.py", line 5, in <module>  
> import apt_pkg  
> ModuleNotFoundError: No module named 'apt_pkg'  
> shubhangi24@vm6:~$ ls  
> docproc-invoice.txt get-pip.py process.py  
> shubhangi24@vm6:~$  
> ^C  
shubhangi24@vm6:~$ The following package was automatically installed and is no longer required: ^C  
shubhangi24@vm6:~$ mysql -h mysql24.mysql.database.azure.com -u shubhangi24@mysql24 -p  
Enter password:  
Welcome to the MySQL monitor. Commands end with ; or \g.  
Your MySQL connection id is 62818  
Server version: 5.6.47.0 MySQL Community Server (GPL)  
  
Copyright (c) 2000, 2022, 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 (0.05 sec)  
  
mysql>
```

Fig 7: Screenshot after completing Step 3 above (create testdb)

Step number	c	
Step name	Running the custom program	
Instructions	<ol style="list-style-type: none"> 1) Run the program using the command <i>python3 process.py</i> 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. <i>mysql -h <database server name> -u <database_username> -p</i> 4) Enter the password when prompted. 5) Run the following commands to verify that the data has been entered into the database <i>use test db;</i> <i>select * from invoice;</i> 6) Enter <i>exit</i> 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>

```
shubhangi24@vm6: ~
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62740
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, 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
-> exit
->
-> exit
-> ^C
^C
mysql> exit
Bye
shubhangi24@vm6:~$ python3 process.py
File "process.py", line 15
  connect_str = "DefaultEndpointsProtocol=https;AccountName=skblob6;AccountKey=66D6EHMfQrt1B8txZY1W0efwNgH68y0s0LLhIyMFNSBUYgx1BKVJzABW6PT24Ax
s+HSJm0XBxUKQ+ASTnnk6aA==;EndpointSuffix=core.windows.net
                  ^
SyntaxError: EOL while scanning string literal
shubhangi24@vm6:~$ python3 process.py
***** Processing File *****
File has been processed

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

*****
Creating table invoice
Inserting data into database
shubhangi24@vm6:~$
```

Fig 8: Running the custom Python program

<Insert screenshot for c(2) here>

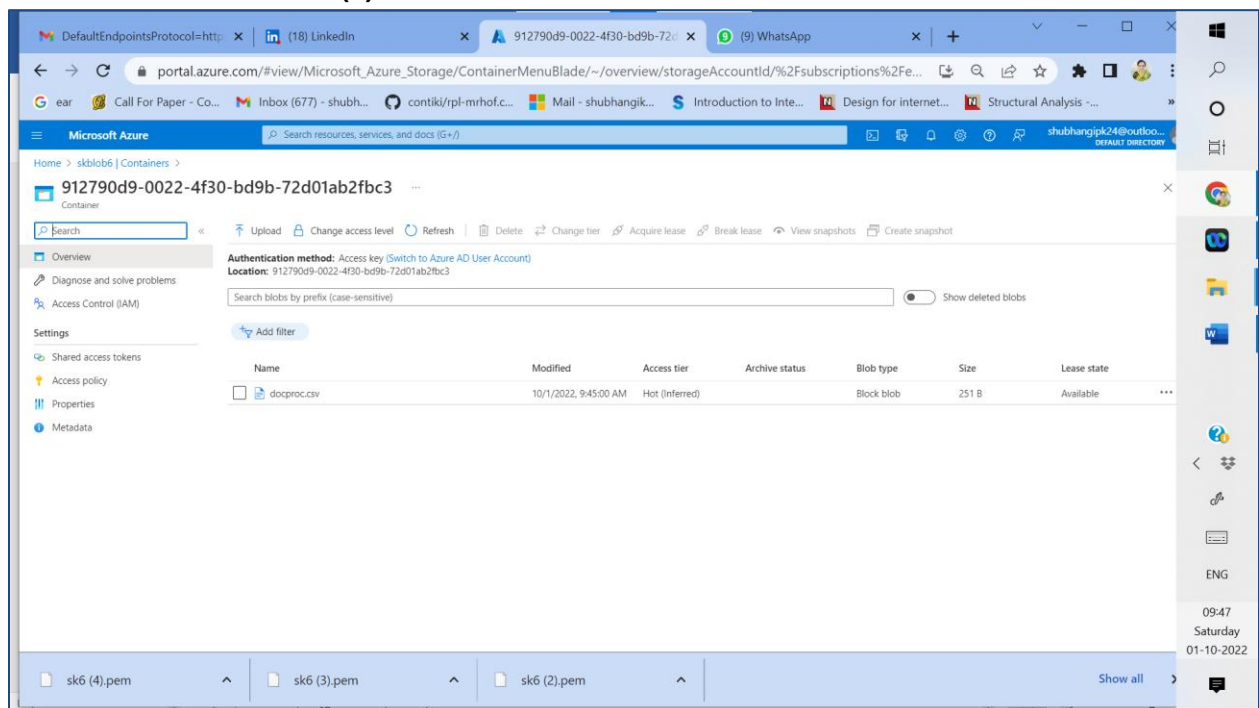


Fig 9: Created CSV file in Blob Storage

<Insert screenshot for c(3) here>

```
shubhangi24@vm6: ~
Uploading to Azure Storage as blob: docproc.csv
inv-00001
Mar 31 2018

*****
Creating table invoice
Inserting data into database
shubhangi24@vm6:~$ mysql -h mysql24.mysql.database.azure.com -u shubhangi24@mysql24 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 62853
Server version: 5.6.47.0 MySQL Community Server (GPL)

Copyright (c) 2000, 2022, 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> select * from invoice;
ERROR 1046 (3D000): No database selected
mysql> use test db;
ERROR 1049 (42000): Unknown database 'test'
mysql> select * from invoice;
ERROR 1046 (3D000): No database selected
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>
```

Fig 10: Screenshot after running step 5 above (select from invoice)

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

b) Blob 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

a) 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

b) 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

c) 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

c) Port 3306 has no bearing on this exercise.

Grades distribution	
MCQs	10 (2 mark each)
Implementation screenshots	10 marks (1 marks each)
Total	20 marks