Dr. Shubhangi Kharche

**Azure Project 2**

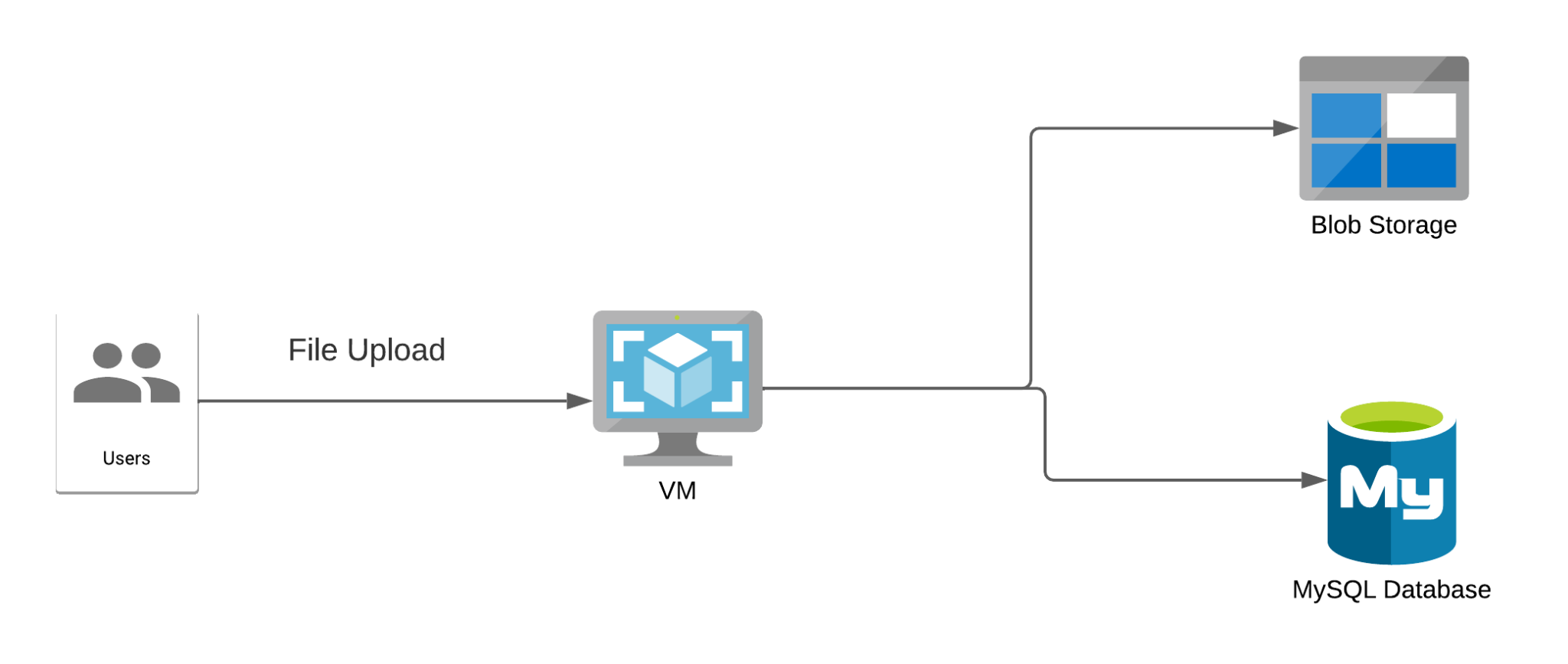
#### Building an Automated Business Process using Managed Services on a Public Cloud

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**

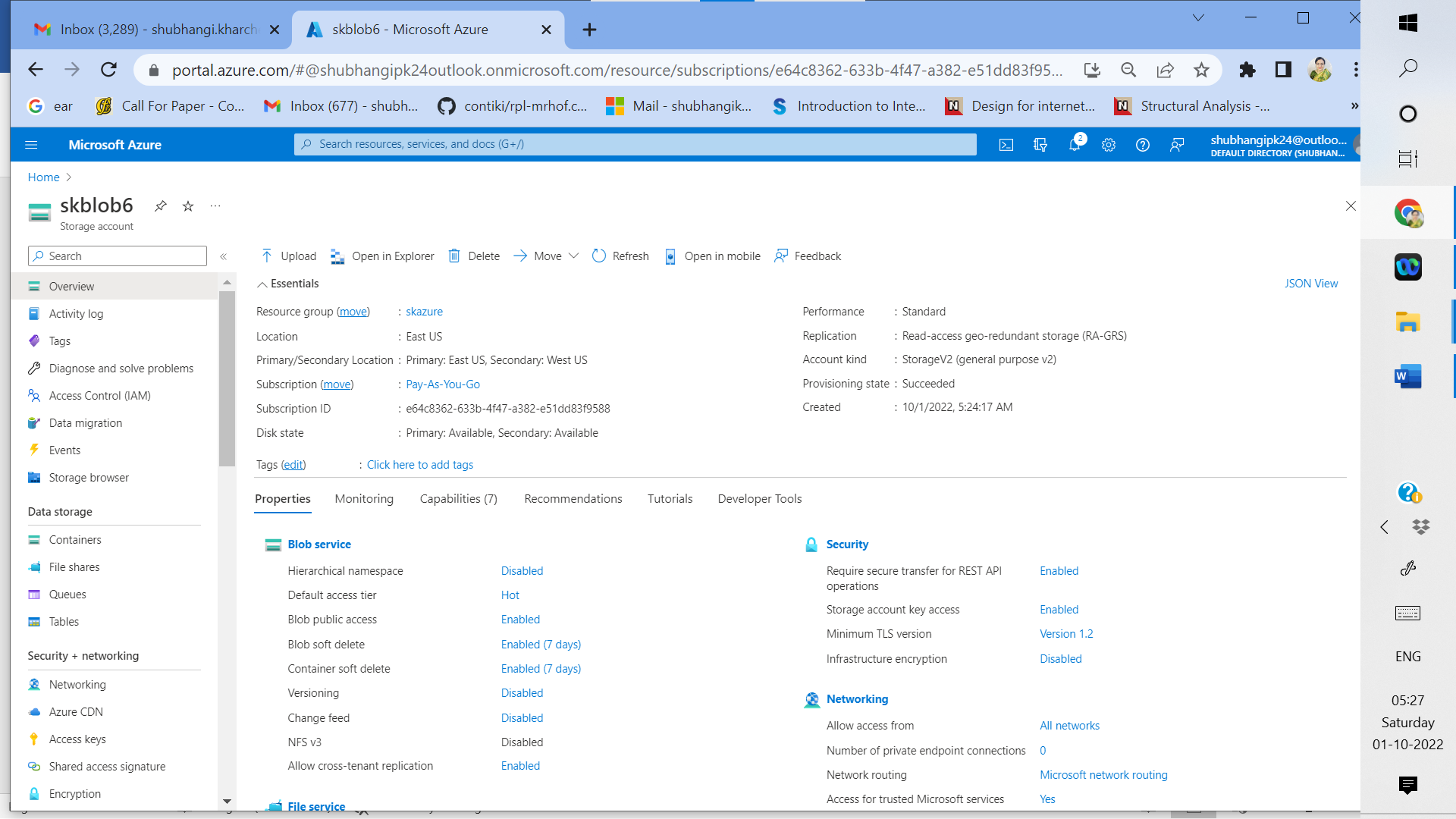
****

|  |  |
| --- | --- |
| **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 | 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 |

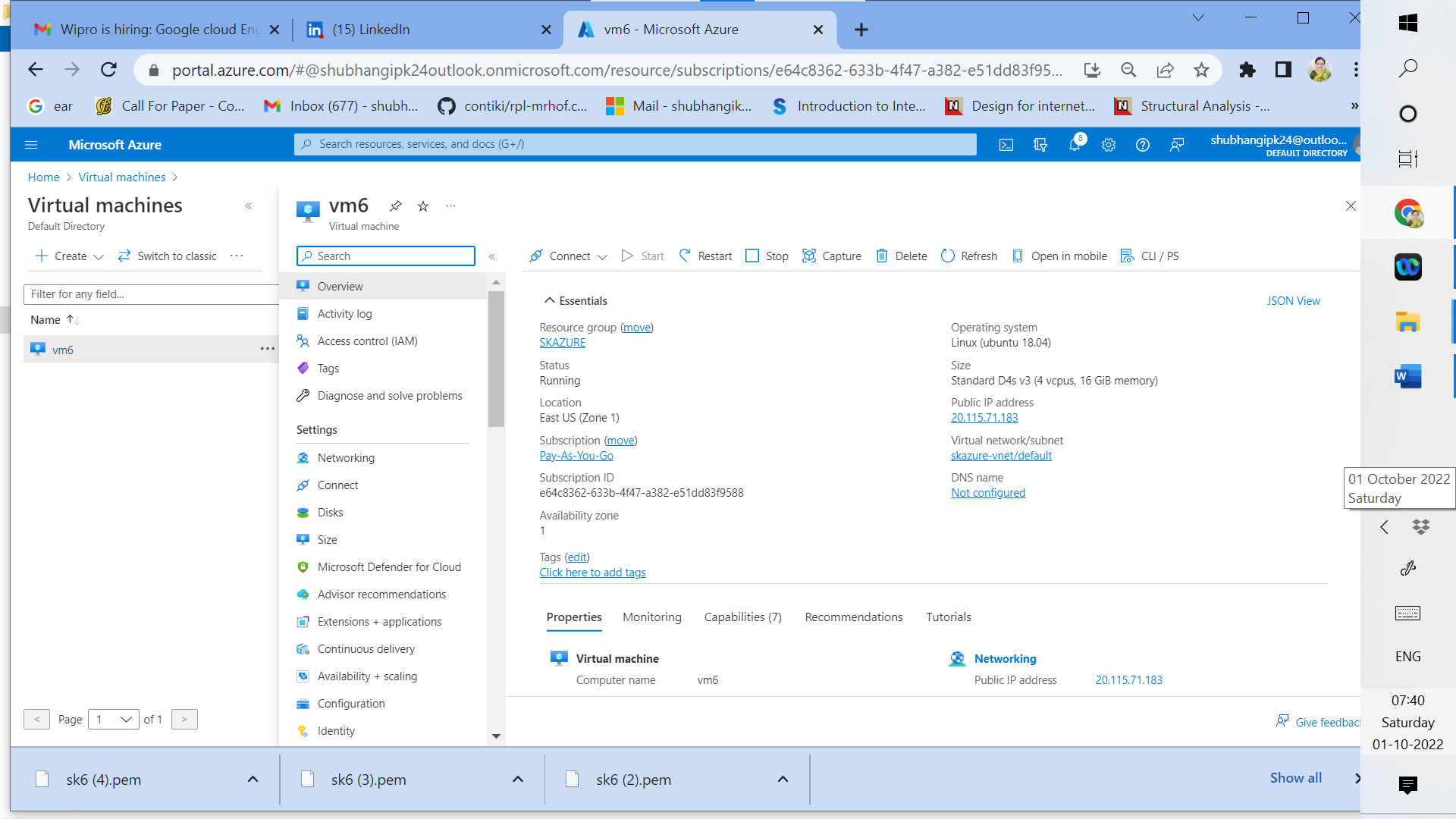
**<Insert screenshot for a(1) here>**

****

**Fig 1:** Screen showing created storage account

|  |  |  |
| --- | --- | --- |
| Step number | b |  |
| Step name | Creation of VM |  |
| Instructions | 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 |  |

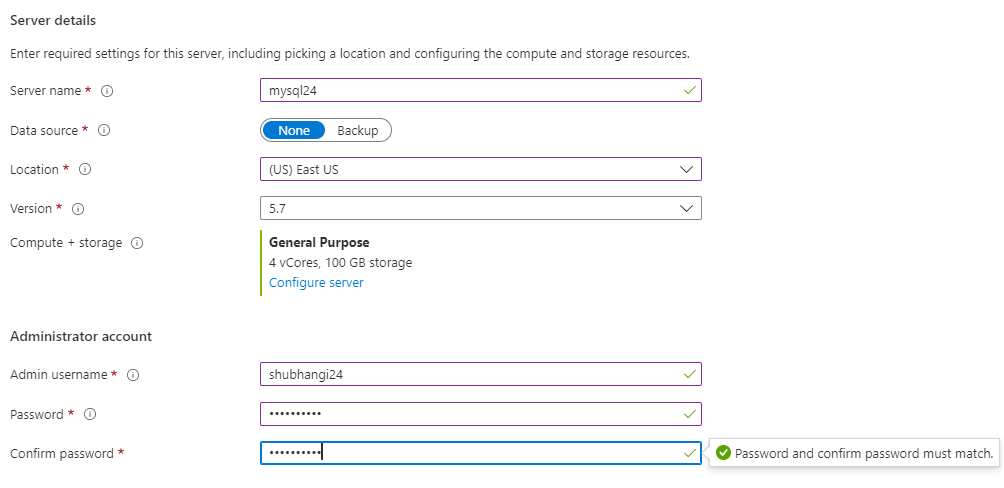
**<Insert screenshot for b(1) here>**



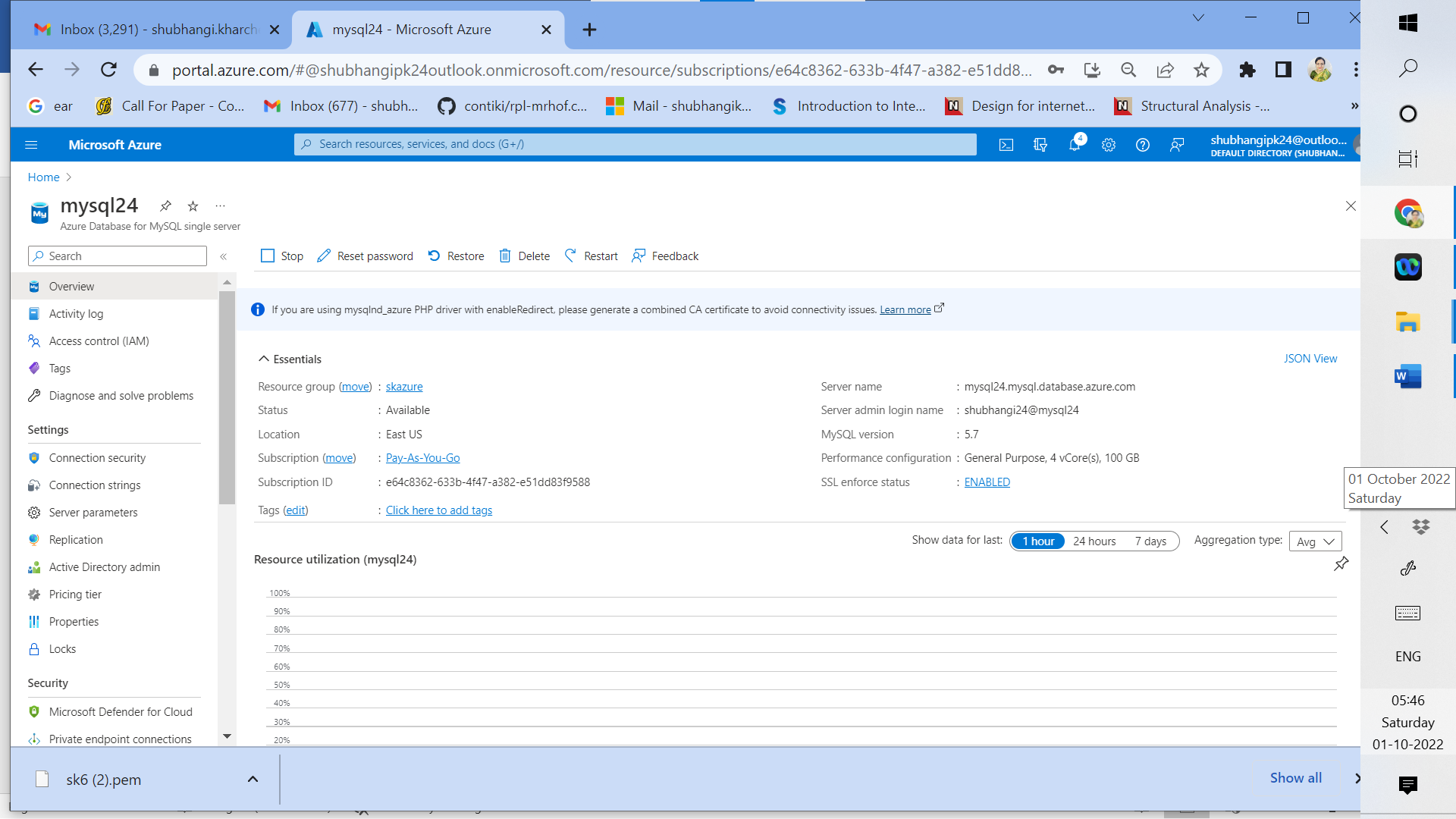
**Fig 2: Created VM**

|  |  |
| --- | --- |
| Step number | c |
| Step name | Creation of MySQL server |
| Instructions | 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 name: mysql24.mysql.database.azure.com

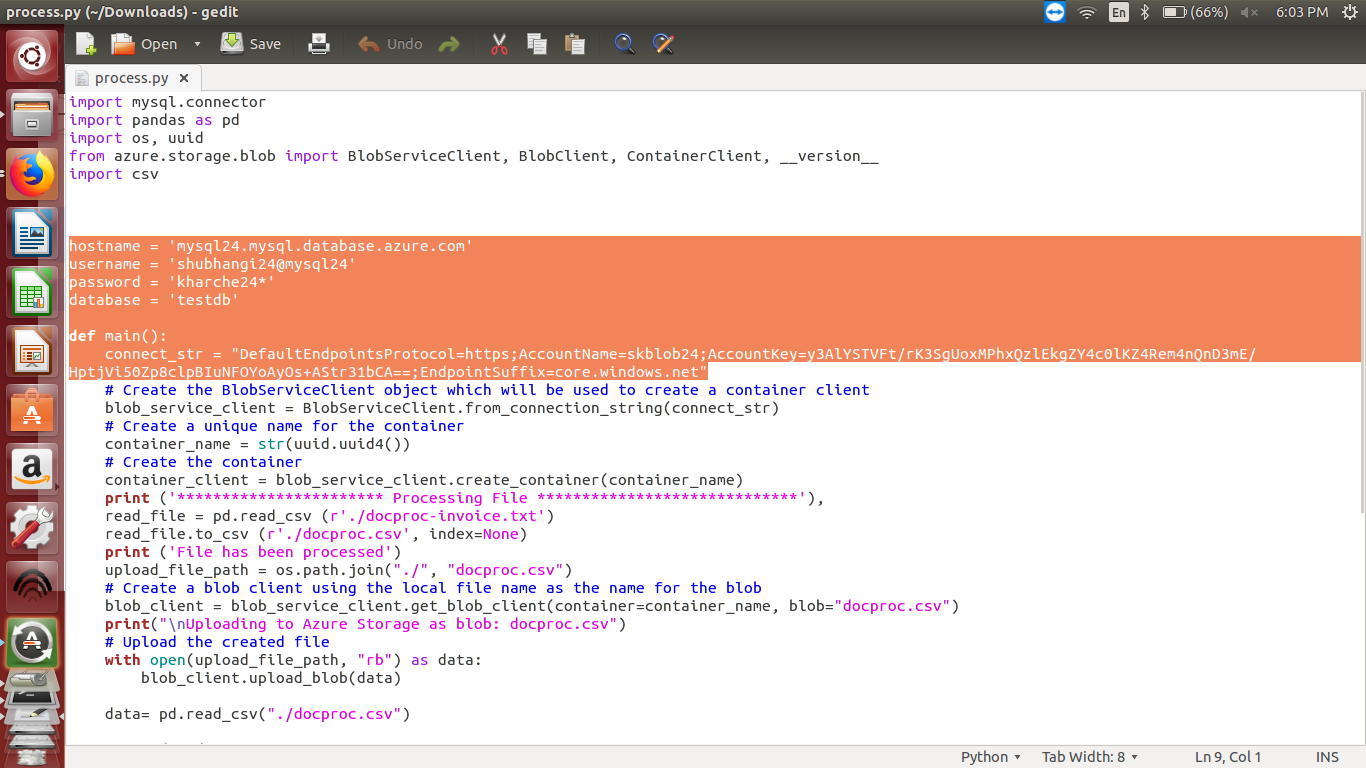


**Fig 3 : Overview screen of the created database server.**

Step 2: Run the custom program in the VM

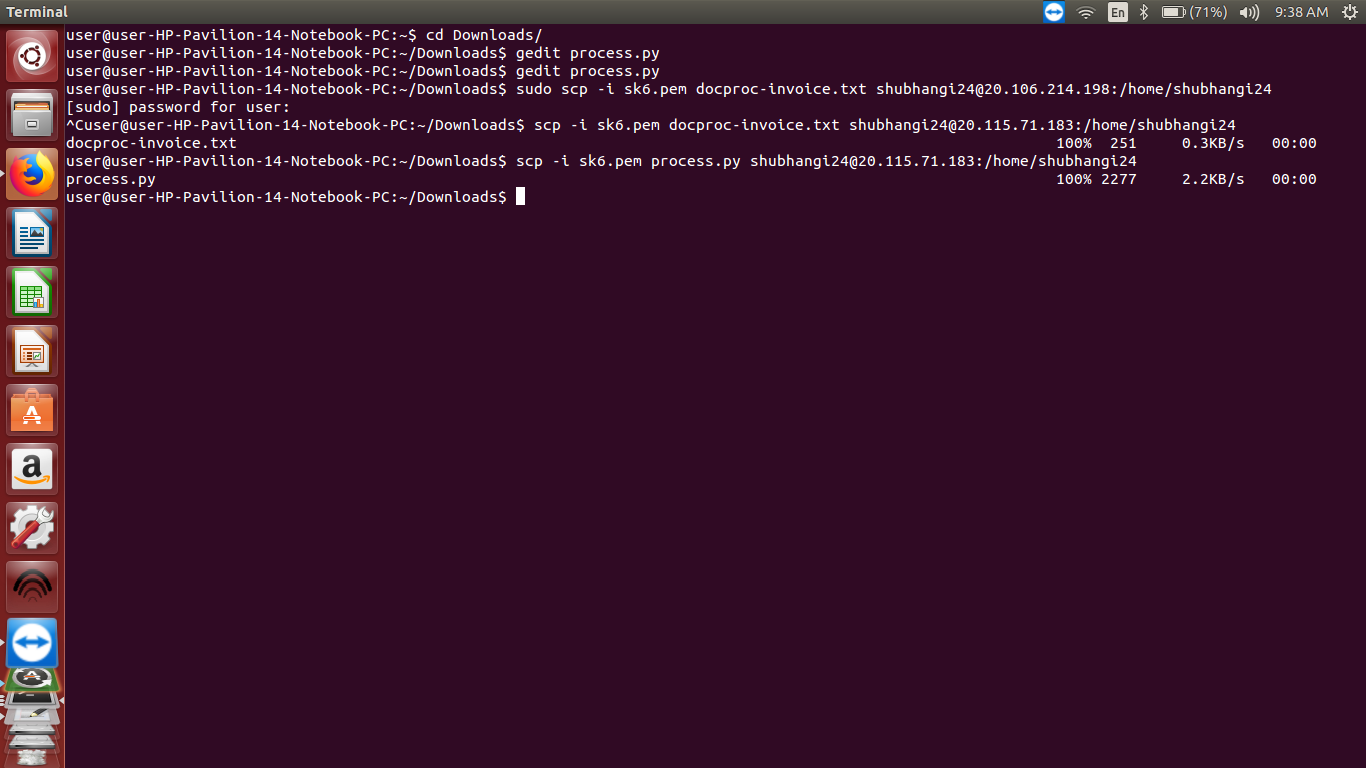
|  |  |  |
| --- | --- | --- |
| Step number | a |  |
| Step name | Environment setup |  |
| Instructions | 1. Download the invoice file and python script provided with this workbook. 2. Open the Python script using your text editor or code editor of choice 3. Replace the values in lines 9,10,11, and 15 with the database server name, username, password, and storage account connection string(recevied 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> <VM username>@<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 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>**

****

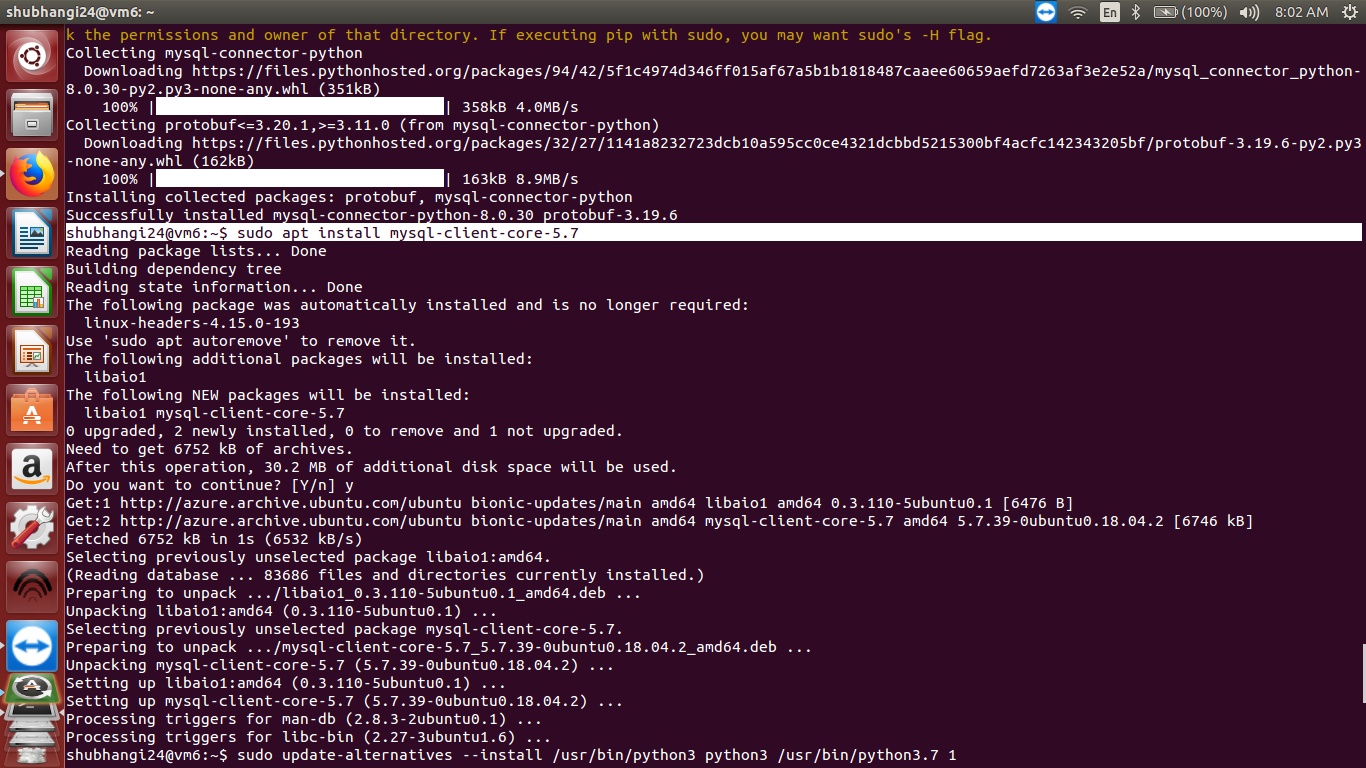
**Fig 4: Screenshot of the process.py file after completing Step3 above (process.py file edited)**

**<Insert screenshot for a(2) here>**

****

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

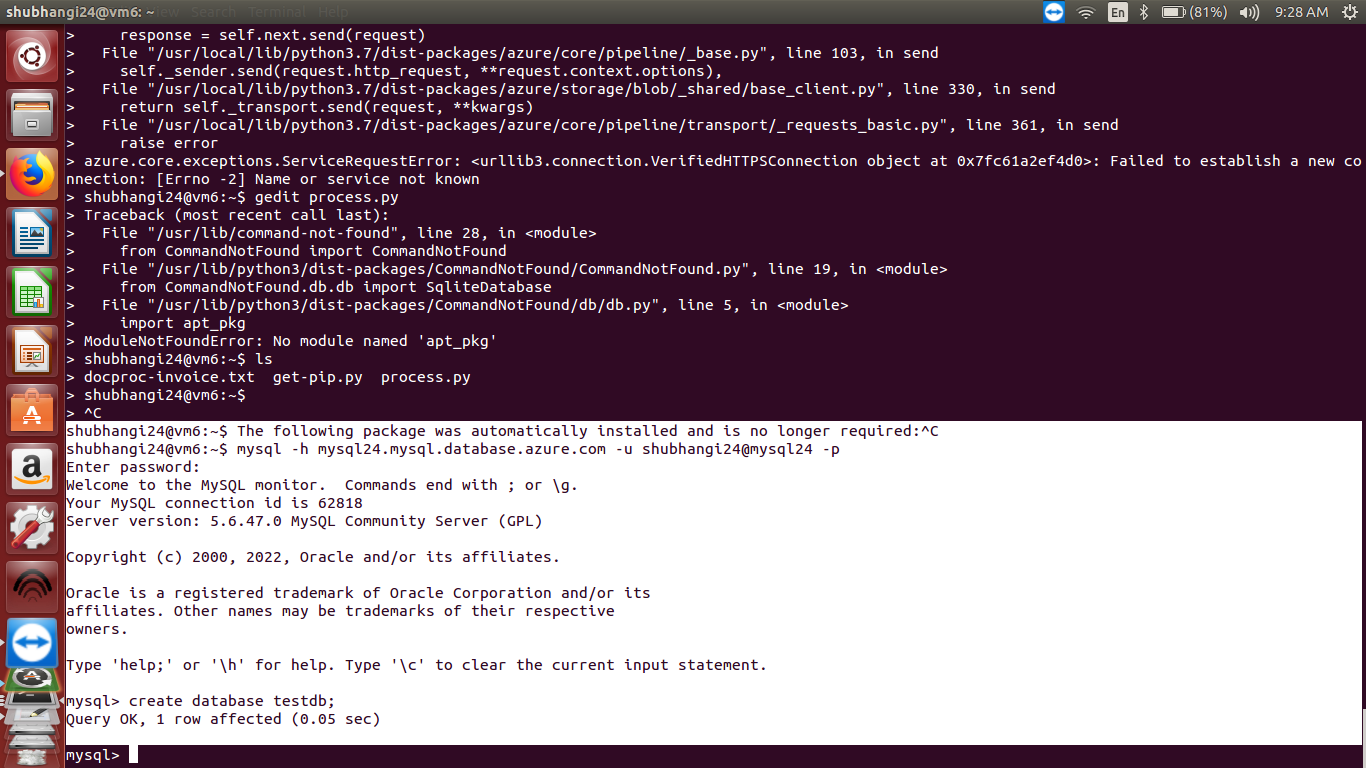
**<Insert screenshot for a(3) here>**

****

**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 |  |

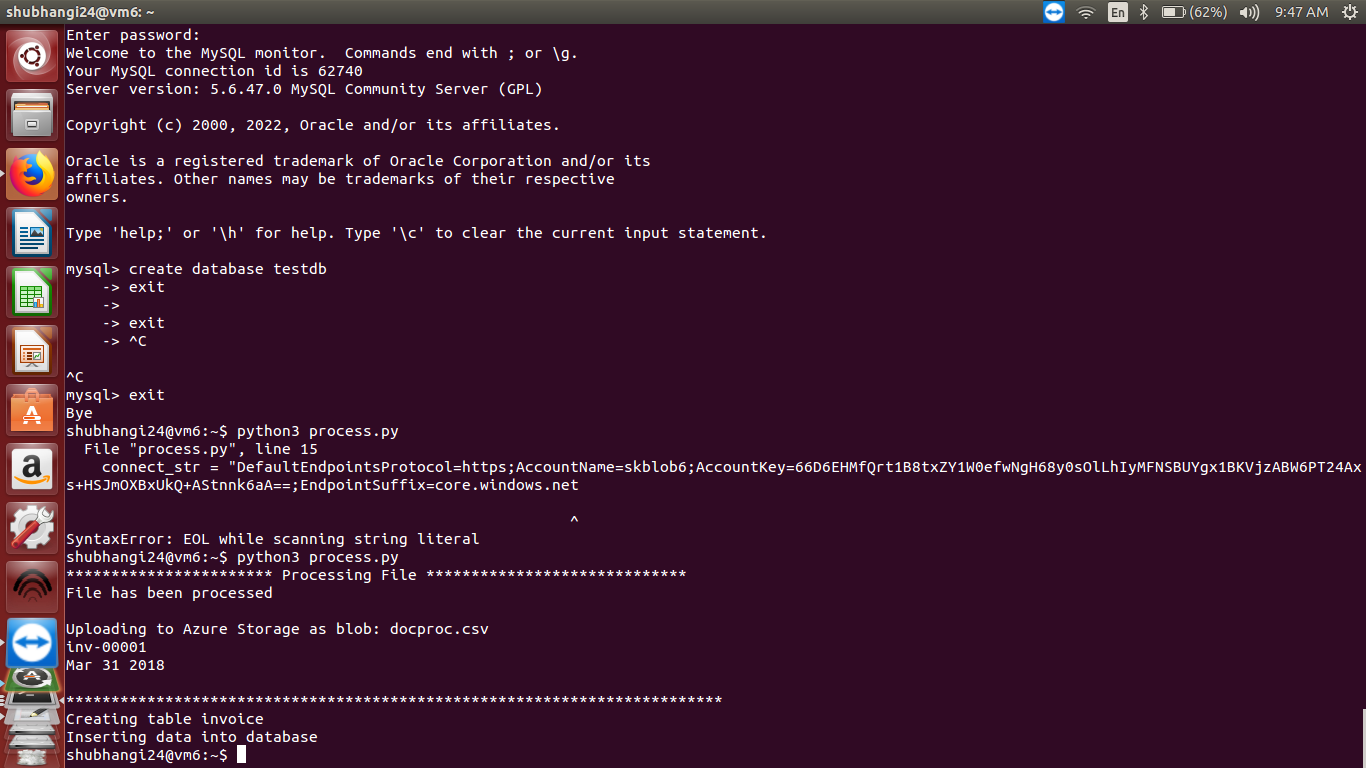
**<Insert screenshot for b(1) here>**

****

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

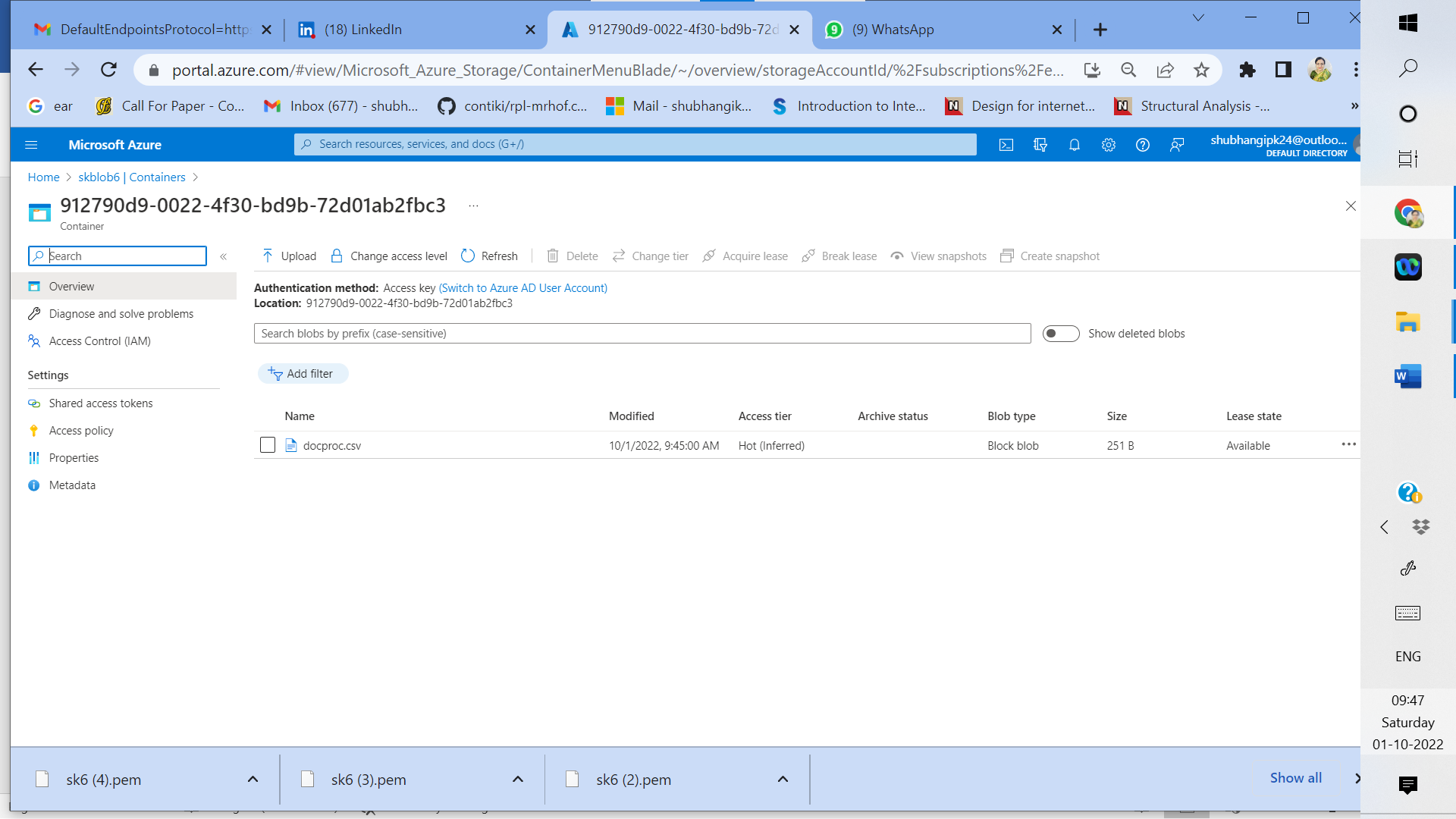
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step number | c |  |  |  |
| Step name | Running the custom program |  |  |  |
| Instructions | 1. Run the program using the command  *python3 process.py* 2. Navigate to the storage account using the Azure portal. Select the Containers option from the 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 folliowing commands to verify that the data has been entered into the database *use test db; 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>**

****

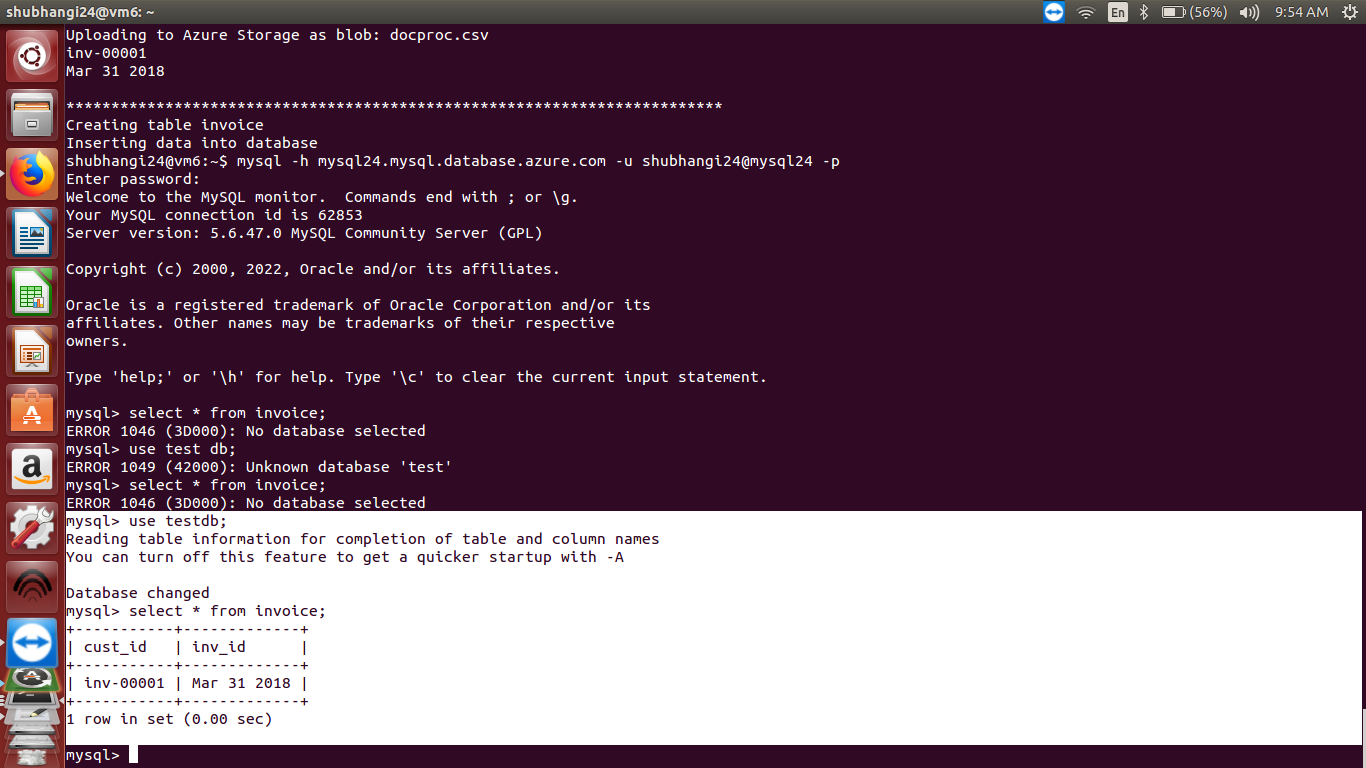
**Fig 8: Running the custom Python program**

**<Insert screenshot for c(2) here>**

****

**Fig 9: Created CSV file in Blob Storage**

**<Insert screenshot for c(3) here>**

****

**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 |