**The steps you followed on the Django VM to take a MySQL database backup and upload it to Azure Storage:**

**1. Install MySQL Client**

* **Command**:

bash

sudo apt install mysql-client-core-5.7

* This ensures you have the tools required to connect to the MySQL server.

**2. Connect to the MySQL Server**

* **Command**:

bash

mysql -h 10.0.3.4 -u django\_user -p

* You logged into the MySQL server hosted on the private VM at 10.0.3.4.

**3. Verify Database**

* Inside the MySQL prompt, list all databases to confirm the target database exists.

sql

SHOW DATABASES;

* Exit the MySQL prompt using:

sql

EXIT;

**4. Take the Database Backup**

* **Command**:

bash

mysqldump -h 10.0.3.4 -u django\_user -p my\_django\_db > my\_django\_db\_backup.sql

* This created a backup file named my\_django\_db\_backup.sql in the current directory.
* mysqldump is a utility provided by MySQL to create logical backups of databases. It exports database contents as SQL statements or a binary dump, which can be used to recreate the database.

**5. Install Azure CLI**

* To upload the backup to Azure Storage, you installed Azure CLI:

bash

sudo apt update

sudo apt install azure-cli -y

**6. Log in to Azure**

* Logged in to Azure CLI to authenticate:

bash

az login

**7. Upload the Backup File to Azure Storage**

* You uploaded the backup to your Azure Storage account (webstoring) and container (database):

bash

az storage blob upload --account-name webstoring --container-name database --name my\_django\_db\_backup.sql --file my\_django\_db\_backup.sql

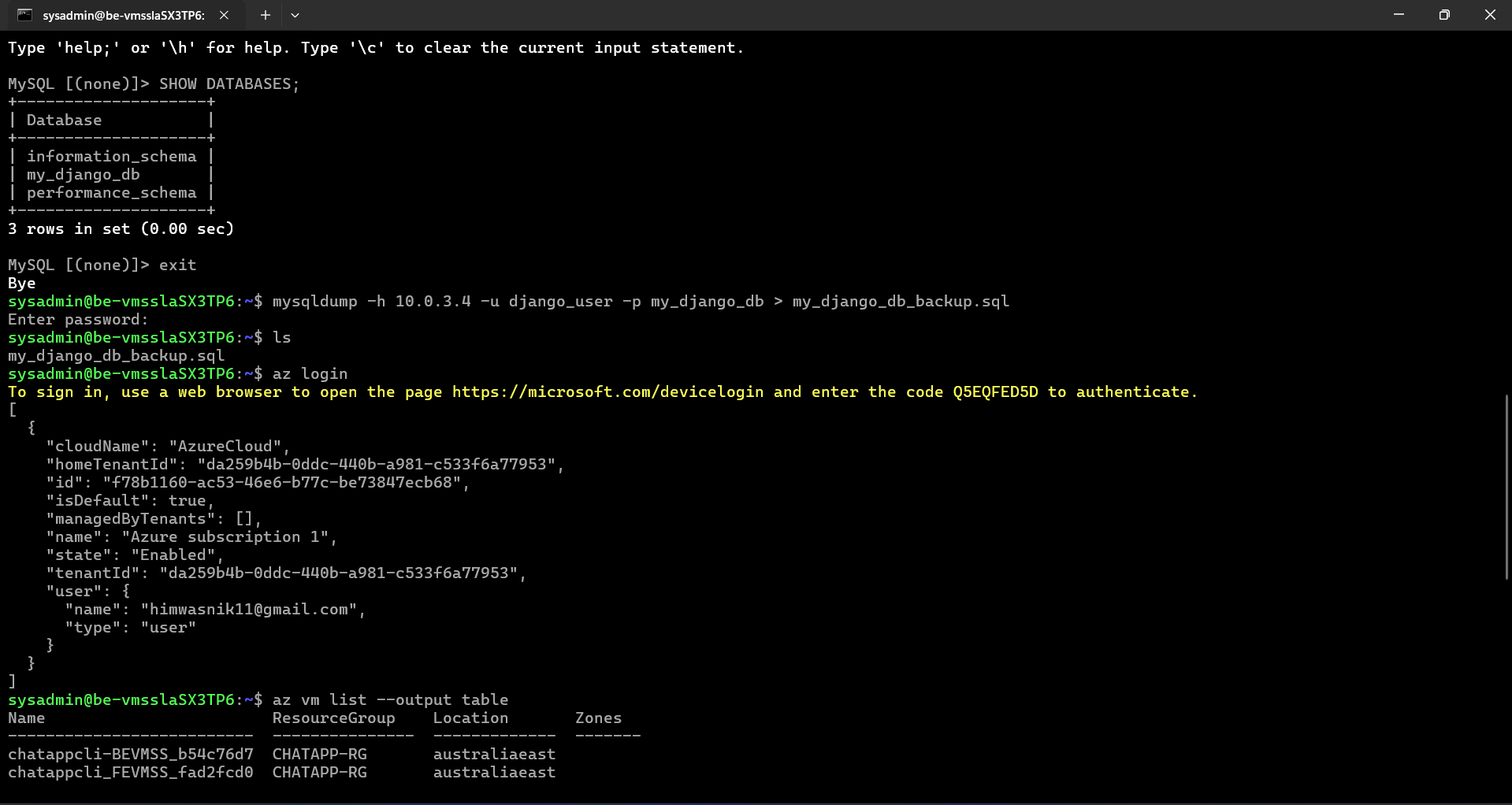
**8. Verify the Uploaded Backup**

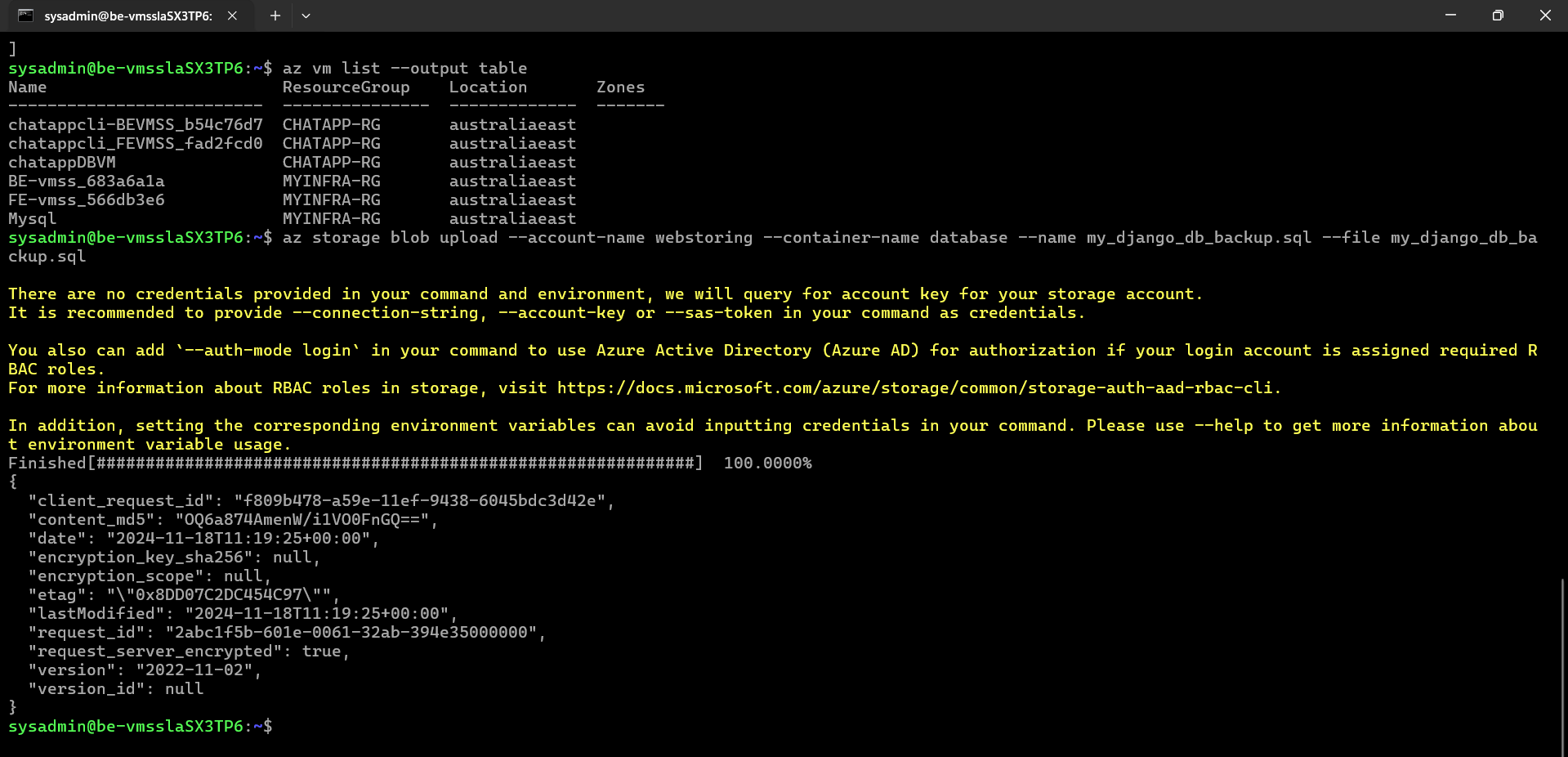
* To confirm the backup was successfully uploaded:

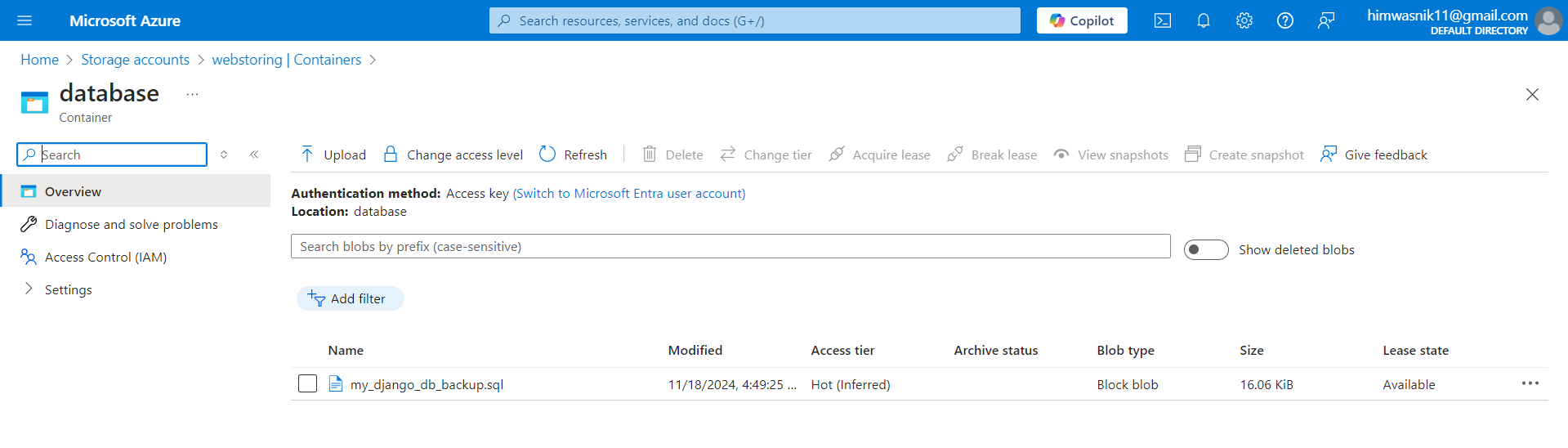
bash

az storage blob list --account-name webstoring --container-name database --output table

* You verified that my\_django\_db\_backup.sql was listed in the container.







***To schedule the MySQL backup and upload it to Azure Storage using cron, you need to create a script and then add a cron job for the desired schedule. Follow these steps:***

**Step 1: Create the Backup Script**

1. Create a shell script for the backup and upload process:

bash

nano /home/sysadmin/mysql\_backup.sh

1. Add the following content to the script:

bash

#!/bin/bash

# Variables

MYSQL\_HOST="10.0.3.4"

MYSQL\_USER="django\_user"

MYSQL\_PASSWORD="your\_mysql\_password"

DATABASE\_NAME="my\_django\_db"

BACKUP\_FILE="/home/sysadmin/my\_django\_db\_backup.sql"

AZURE\_STORAGE\_ACCOUNT="webstoring"

CONTAINER\_NAME="database"

BLOB\_NAME="my\_django\_db\_backup.sql"

# Take MySQL backup

mysqldump -h $MYSQL\_HOST -u $MYSQL\_USER -p$MYSQL\_PASSWORD $DATABASE\_NAME > $BACKUP\_FILE

# Upload to Azure Blob Storage

az storage blob upload --account-name $AZURE\_STORAGE\_ACCOUNT --container-name $CONTAINER\_NAME --name $BLOB\_NAME --file $BACKUP\_FILE --overwrite

# Optional: Remove the local backup file to save space

rm -f $BACKUP\_FILE

1. Save and exit (Ctrl + O, Enter, Ctrl + X).

**Step 2: Make the Script Executable**

bash

sudo chmod +x /home/sysadmin/mysql\_backup.sh

**Step 3: Test the Script**

Run the script manually to ensure it works:

bash

/home/sysadmin/mysql\_backup.sh

**Step 4: Schedule the Script with cron**

1. Open the crontab editor:

bash

crontab -e

1. Add an entry to schedule the script:
   * **Daily Backup at Midnight**:

bash

0 0 \* \* \* /home/sysadmin/mysql\_backup.sh

* + **Weekly Backup on Sundays at Midnight**:

bash

0 0 \* \* 0 /home/sysadmin/mysql\_backup.sh

1. Save and exit the editor.

**Step 5: Verify the Cron Job**

1. List all cron jobs to ensure your entry is saved:

crontab -l

1. Check logs to confirm the job runs as expected:

tail -f /var/log/syslog | grep CRON

