Assignment : Azure Cloud Exercises

**Exercise 1**: Create and Configure a Virtual Machine

Objective:

Create and configure Ubuntu and Windows Virtual Machines on Azure Portal.

Steps:

1. Create an Ubuntu VM:

- Log in to the Azure Portal.

- Navigate to:

- Click on ‘Virtual Machines’ in the left sidebar.

- Click on ‘Create’.

- Choose Ubuntu Server:

- Select ‘Ubuntu Server 20.04 LTS’ from the list.

- Configure the VM:

- Size: Select ‘Standard\_B1s’.

- Authentication Type: Choose ‘SSH’ (generate a key pair if not available).

- Inbound Port: Allow SSH (port 22).

- Deploy and Connect:

- Click on Review + Create and then Create.

- Connect to the VM using SSH from your terminal.

2. Create a Windows VM:

- Follow similar steps:

- Navigate to Virtual Machines > Create.

- Select Windows Server 2022.

- Configure the VM:

- Size: Select Standard\_B1s (or similar).

- Authentication Type: Choose Username and Password.

- Inbound Port: Allow RDP (port 3389).

- Deploy and Connect:

- Click on Review + Create and then Create.

- Connect to the VM using RDP.

3. Task:

- Install Apache or IIS:

- For Ubuntu, use the command: `sudo apt update && sudo apt install apache2`.

- For Windows, use the Server Manager to install IIS.

- Verify Installation:

- Access the default web page from your local browser using the public IP of the VMs.

---

**Exercise 2**: Deploy a Static Web Application

Objective:

Host a static website using Azure App Service.

Steps:

1. Navigate to App Services:

- Click on App Services in the left sidebar.

- Click on Create.

2. Choose Configuration:

- Runtime Stack: Select Python 3.10 (or latest).

- Operating System: Choose Linux.

- Region: Select the region closest to your location.

3. Deploy the Application:

- Click on Review + Create and then Create.

4. Upload Static Website:

- Use FTP or the Kudu console to upload your static website files (e.g., `index.html` and CSS files).

5. Task:

- Verify Deployment:

- Access the site via its public URL.

- Modify HTML:

- Update the HTML to include a message like: "Welcome to Azure Static Web Apps!"

---

**Exercise 3**: Deploy a Flask Application (Dynamic Web App)

Objective:

Deploy a Python Flask application using Azure App Service.

Steps:

1. Create a Flask App:

- Write the following code in a file named `app.py`:

**python**

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return "Hello, Azure Flask App!"

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

```

2. Push Code to GitHub:

- Create a new repository on GitHub and push your `app.py` file to it.

3. In Azure Portal:

- Navigate to App Services > Create.

4. Configure the App:

- Runtime Stack: Select Python 3.10 (or latest).

- Deployment Source: Connect your GitHub repository.

5. Deploy the Flask App:

- Click on Review + Create and then Create.

- Verify the deployment by accessing the public URL.

---

**Exercise 4**: Set Up and Use an Azure SQL Database

Objective:

Create an Azure SQL Database and connect to it from your local machine.

Steps:

1. Navigate to SQL Databases:

- Click on SQL Databases in the left sidebar.

- Click on Create.

2. Configure the Database:

- Database Name: Enter `StudentDB`.

- Server: Create a new server with a username and password.

- Compute + Storage: Use the free tier.

3. Deploy the Database:

- Click on Review + Create and then Create.

4. Connect to the Database:

- Use Azure Data Studio or SQL Server Management Studio (SSMS) to connect.

5. Task:

- Create a Table:

- Execute the SQL command to create a table named `Students` with columns `ID`, `Name`, and `Age`.

- Insert Sample Data:

- Insert some sample data into the `Students` table and query it.

---

**Exercise 5**: Integrate Flask App with Azure SQL Database

Objective:

Connect a Flask app to Azure SQL Database and perform CRUD operations.

Steps:

1. Use the Flask App from Exercise 3.

2. Install Required Libraries:

- Run the command: `pip install flask pyodbc`.

3. Modify the App to Connect to SQL Database:

- Update your `app.py` to include the following code:

```python

import pyodbc

conn = py