Cloud Computing EE 655/755
Assignment #4

INTRODUCTION:

1. Azure Overview:

Microsoft Azure is a cloud computing platform that offers a wide range of services for building, deploying, and managing applications and services through Microsoft's global network of data centers. Azure provides services such as virtual machines, databases, storage, networking, and more, enabling businesses to scale their operations and infrastructure based on demand.

2. PuTTY:

PuTTY is a free and open-source SSH and telnet client used to access remote servers securely. It allows users to connect to a remote machine or server over a network and perform various tasks, such as running commands, transferring files, and managing configurations.

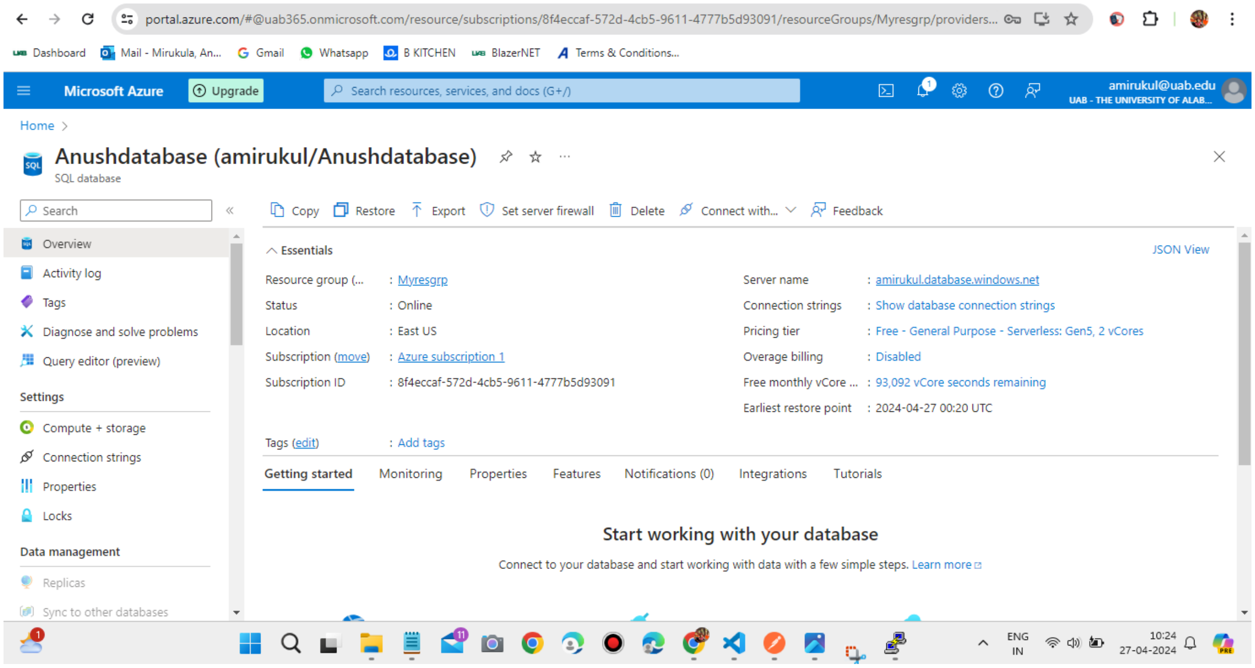
3. Postman:

Postman is a popular API development tool used for testing, documenting, and debugging APIs. It provides a user-friendly interface for sending HTTP requests to APIs, inspecting responses, and automating API testing workflows.

Using Azure, PuTTY, and Postman Together:

1. Creating Azure Resources:

- Azure SQL Database:
  - Log in to the Azure portal (portal.azure.com).
  - Navigate to "SQL databases" and create a new database instance.



Anushdatabase (amirukul/Anushdatabase)

SQL database

Resource group (... : Myresgrp

Status : Online

Location : East US

Subscription (move) : Azure subscription 1

Subscription ID : 8f4eccaf-572d-4cb5-9611-4777b5d93091

Server name : amirukul.database.windows.net

Connection strings : Show database connection strings

Pricing tier : Free - General Purpose - Serverless: Gen5, 2 vCores

Overage billing : Disabled

Free monthly vCore ... : 93,092 vCore seconds remaining

Earliest restore point : 2024-04-27 00:20 UTC

Tags (edit) : Add tags

Start working with your database

Connect to your database and start working with data with a few simple steps. Learn more

- Configure database settings, such as server name, resource group, and pricing tier.

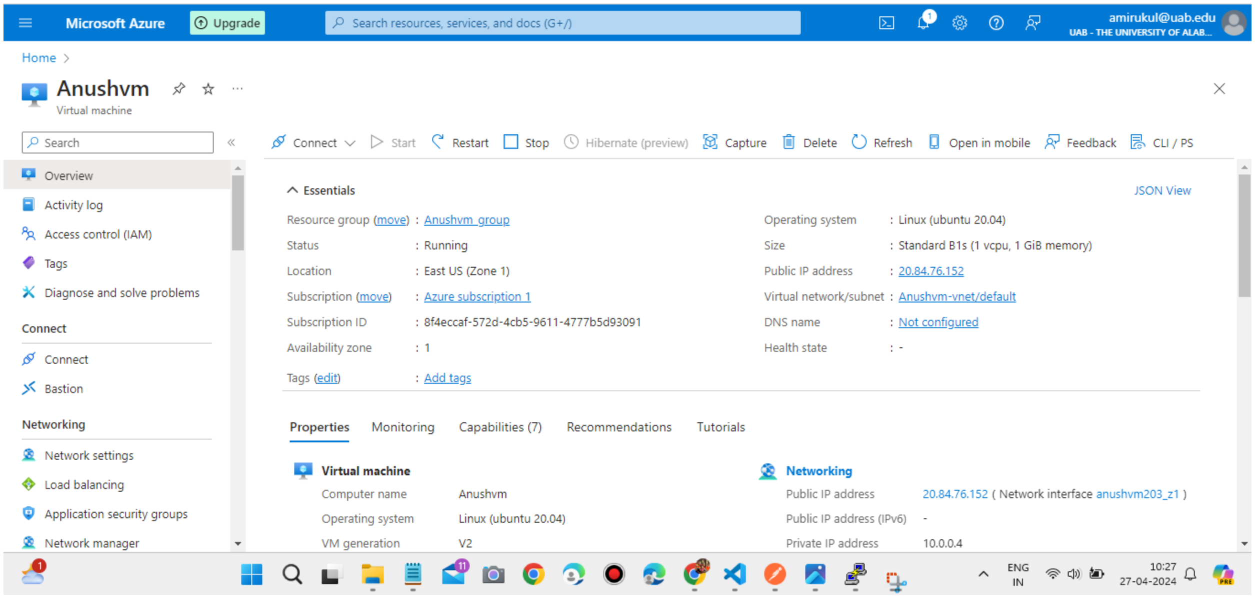
- Obtain connection credentials (server name, username, password) for accessing the database.

- Azure Virtual Machine (VM):

- Navigate to "Virtual machines" in the Azure portal.

- Create a new VM instance, specifying details like OS type, size, disk, and networking settings.

- Connect to the VM using SSH (for Linux VMs) or RDP (for Windows VMs) to access the remote server environment.



Anushvm
Virtual machine

Essentials

Resource group (move) : Anushvm_group
Status : Running
Location : East US (Zone 1)
Subscription (move) : Azure subscription 1
Subscription ID : 8f4eccaf-572d-4cb5-9611-4777b5d93091
Availability zone : 1
Tags (edit) : Add tags

Operating system : Linux (ubuntu 20.04)
Size : Standard B1s (1 vcpu, 1 GiB memory)
Public IP address : 20.84.76.152
Virtual network/subnet : Anushvm-vnet/default
DNS name : Not configured
Health state : -

Virtual machine
Computer name Anushvm
Operating system Linux (ubuntu 20.04)
VM generation V2

Networking
Public IP address 20.84.76.152 (Network interface anushvm203_z1)
Public IP address (IPv6) -
Private IP address 10.0.0.4

2. Accessing Azure VM Using PuTTY:

1. Download PuTTY:

If you haven't already, download PuTTY from the official website: [PuTTY Download Page](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html).

2. Launch PuTTY:

After installing PuTTY, launch the PuTTY application.

3. Enter VM Details:

In the PuTTY configuration window:

- Enter the VM's public IP address or DNS name in the "Host Name (or IP address)" field.

- Specify the SSH port (default is 22).

- Ensure that the connection type is set to "SSH".

4. Establish SSH Connection:

Click the "Open" button to initiate the SSH connection to the Azure VM.

5. Log in to the VM:

After clicking "Open", PuTTY will open a terminal window prompting you to log in:

- Enter your username for the Azure VM when prompted.
- Enter your password for the Azure VM when prompted (if you're using password-based authentication).

6. Verify Connection:

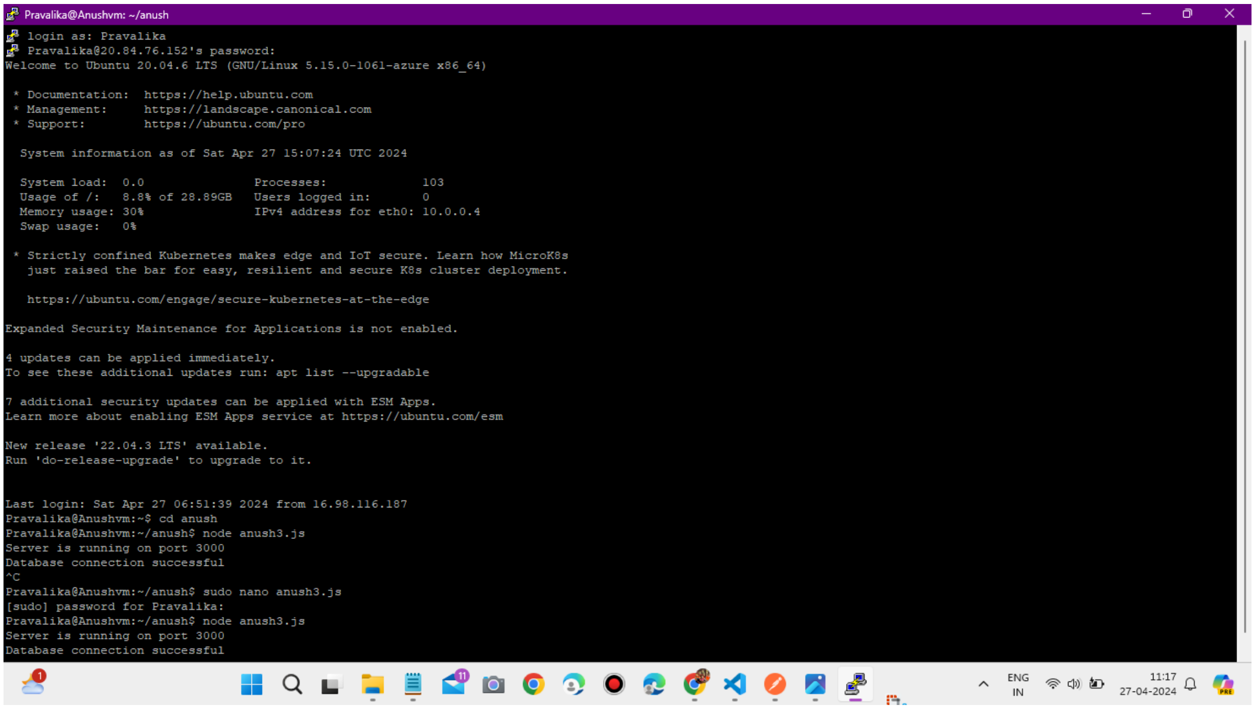
Once you've entered your credentials (username/password or SSH key), PuTTY will attempt to connect to the Azure VM. If the connection is successful, you'll see the terminal window with a command prompt for the VM.

7. Using PuTTY:

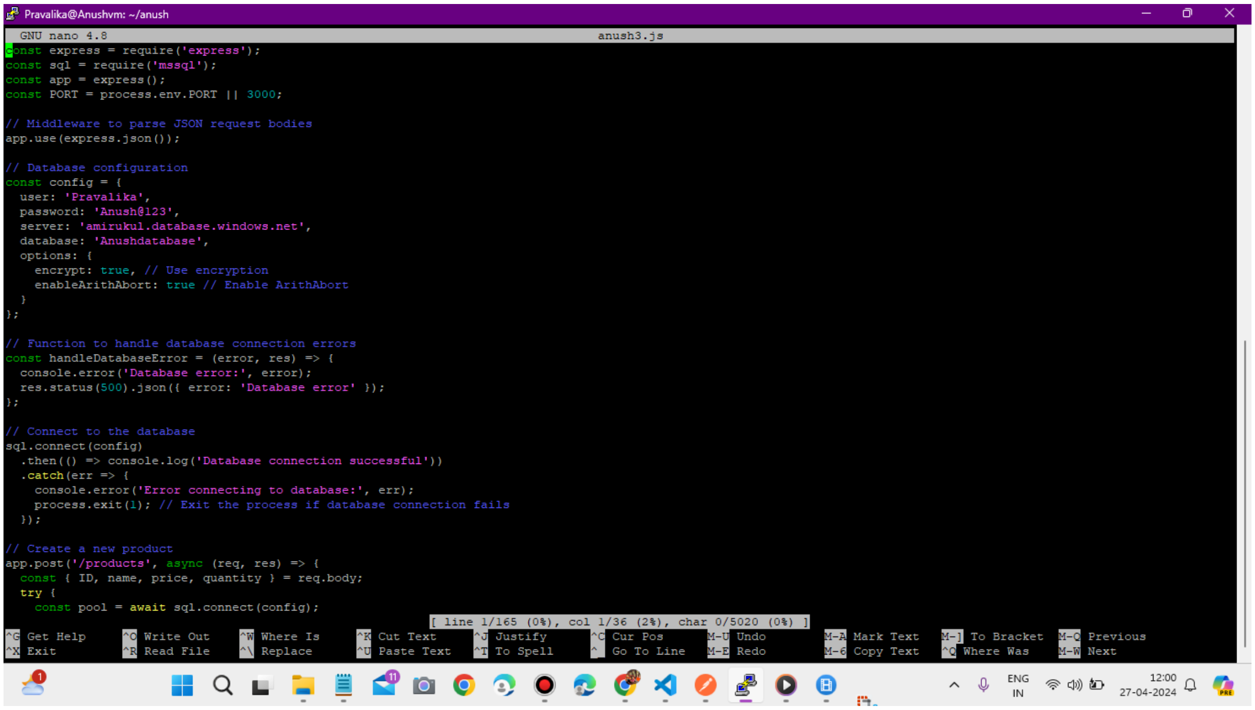
You can now use PuTTY to run commands, manage files, and perform administrative tasks on the Azure VM through the SSH connection.

Make sure you have the necessary credentials (username/password or SSH key) for logging in to your Azure VM. Adjust the SSH port and other settings as per your VM configuration.

-



```
login as: Pravalika
Pravalika@20.84.76.152's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1061-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Sat Apr 27 15:07:24 UTC 2024

  System load:  0.0               Processes:             103
  Usage of /:   8.8% of 28.89GB   Users logged in:       0
  Memory usage: 30%               IPv4 address for eth0: 10.0.0.4
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

4 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

7 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Sat Apr 27 06:51:39 2024 from 16.98.116.187
Pravalika@Anushvm:~$ cd anush
Pravalika@Anushvm:~/anush$ node anush3.js
Server is running on port 3000
Database connection successful
^C
Pravalika@Anushvm:~/anush$ sudo nano anush3.js
[sudo] password for Pravalika:
Pravalika@Anushvm:~/anush$ node anush3.js
Server is running on port 3000
Database connection successful
```



```
  GNU nano 4.8                                                   anush3.js
const express = require('express');
const sql = require('mssql');
const app = express();
const PORT = process.env.PORT || 3000;

// Middleware to parse JSON request bodies
app.use(express.json());

// Database configuration
const config = {
  user: 'Pravalika',
  password: 'Anush@123',
  server: 'amirukul.database.windows.net',
  database: 'Anushdatabase',
  options: {
    encrypt: true, // Use encryption
    enableArithAbort: true // Enable ArithAbort
  }
};

// Function to handle database connection errors
const handleDatabaseError = (error, res) => {
  console.error('Database error:', error);
  res.status(500).json({ error: 'Database error' });
};

// Connect to the database
sql.connect(config)
  .then(() => console.log('Database connection successful'))
  .catch(err => {
    console.error('Error connecting to database:', err);
    process.exit(1); // Exit the process if database connection fails
  });

// Create a new product
app.post('/products', async (req, res) => {
  const { ID, name, price, quantity } = req.body;
  try {
    const pool = await sql.connect(config);
                            [ line 1/165 (0%), col 1/36 (2%), char 0/5020 (0%) ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Undo       M-A Mark Text  M-] To Bracket M-Q Previous
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell    ^_ Go To Line  M-E Redo       M-6 Copy Text  ^Q Where Was   M-W Next
```

3. Testing APIs with Postman:

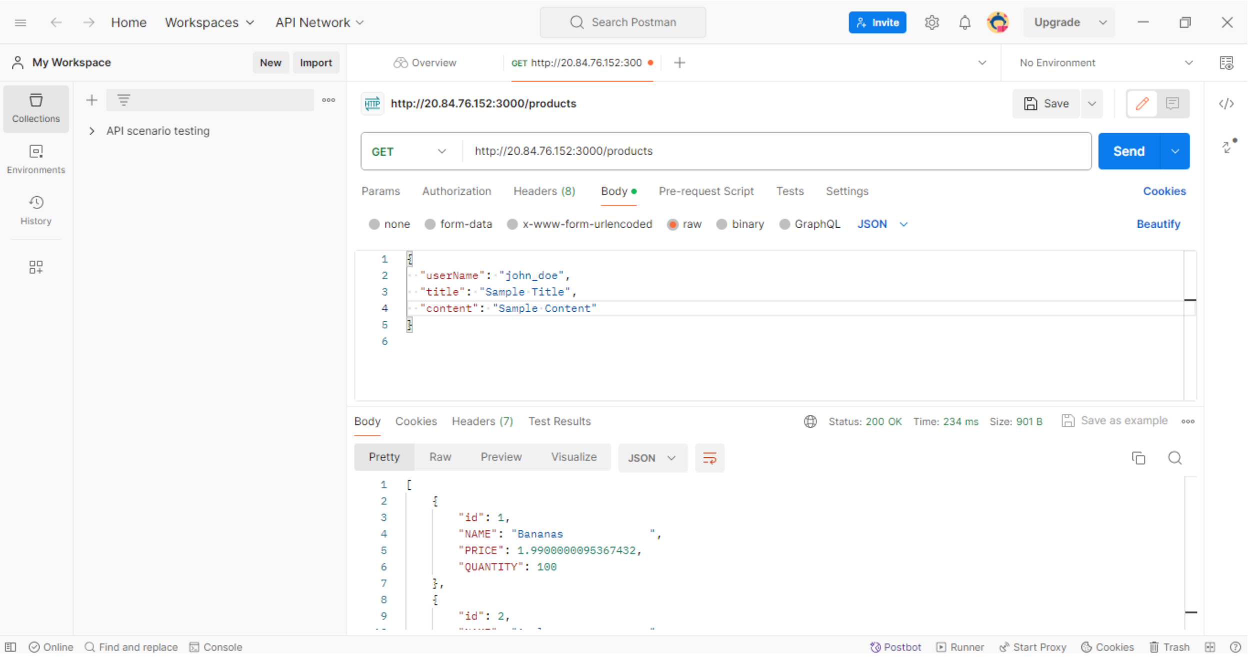
- Install Postman on your local machine.

- Create a new request in Postman and enter the API endpoint URL.

- Choose the HTTP method (GET, POST, PUT, DELETE) and add any required headers or parameters.

- Send the request to the API and view the response, including status codes, headers, and JSON data (if applicable).

- Use Postman's features for organizing requests, saving collections, and running automated tests.



http://20.84.76.152:3000/products

GET http://20.84.76.152:3000/products

```
{
  "userName": "john_doe",
  "title": "Sample Title",
  "content": "Sample Content"
}
```

Status: 200 OK Time: 234 ms Size: 901 B

```
[
    {
        "id": 1,
        "NAME": "Bananas              ",
        "PRICE": 1.9900000095367432,
        "QUANTITY": 100
    },
    {
        "id": 2,
```

Understanding Access and Security:

- Azure resources like databases and VMs have specific access controls and security measures.

- Use Azure's IAM (Identity and Access Management) to manage user roles and permissions for accessing resources.

- Ensure secure communication by using HTTPS for APIs and SSH/RDP for remote server access.

- Implement firewall rules, network security groups (NSGs), and encryption protocols to protect data and network traffic.

Best Practices:

- Regularly update software and services to the latest versions for security patches and performance improvements.

- Backup critical data and configurations to prevent data loss in case of failures or disasters.

- Monitor Azure resources using Azure Monitor and set up alerts for proactive monitoring and troubleshooting.

Conclusion:

Azure, PuTTY, and Postman are powerful tools for creating, managing, and testing cloud-based applications and infrastructure. By understanding how to use them together, developers and administrators can streamline development workflows, ensure system reliability, and deliver scalable solutions on the Azure platform.