Tree Tier Application Deployment

(Date: 31-May & 1-June-2025)

We will deploy three tier application using Frontend, Backend & Database servers 

**Step-by-Step Deployment: 3-Tier Todo App on Azure**

### ✅ Step 1: Create Azure SQL Database (PaaS)

#### Provision Azure SQL Server & Database >>

## **SQL Server** >> todoapp-sqlserver-01 >> Use SQL authentication >> Networking > firewall rule: Yes >> additional setting >> Microsoft Defender SQL >> Not now >> review + create

## **SQL databases** >> create > rg-todo >> db\_name > todoappdb >> LRS >>Enable public IP address >> yes >> review + create

**Todoappdb >>** Query editor (to login database virtually) >> Login >> add below query and run and save

Create Table of employees

CREATE TABLE employees (

employee\_id INT PRIMARY KEY IDENTITY(1,1),

email VARCHAR(100) UNIQUE,

phone VARCHAR(15) UNIQUE,

location VARCHAR(100)

);

Add data:

INSERT INTO employees (email, phone, location) VALUES

('rahul.sharma@example.com', '9876543210', 'Delhi'),

('priya.verma@example.com', '9123456780', 'Mumbai'),

('amit.kumar@example.com', '9012345678', 'Bangalore'),

('sneha.singh@example.com', '9988776655', 'Hyderabad');

To check and select the employee details by the help of query

SELECT \* FROM employees

WHERE location = 'Mumbai';

### 🐍 Step 2: Deploy Backend Application (FastAPI on Azure VM)

#### 2.1 Create Backend Azure VM

Create backend VM by selecting given requirements from readme.md file

Git clone <https://github.com/devopsinsiders/PyTodoBackendMonolith.git> >> download inside the local repo

Open with VS code >> go to readme.md file and follow these steps >> update connection string >> ODBC

Go to database >> connection string >> ODBC >> copy and came to again on vs code and go to app.py >> connection string >> conection\_string = <paste the copied link> update pwd = write password of dB server >> also update the ODBC version from 18 to 17 which is written in readme file

Connect VM >> ssh username@publicip

Search on google (as per redme file) >> how to install python and pip in ubuntu 20.4 >> sudo apt update >> sudo apt install python3 python3-pip

In VS code >> scp -r C:\Devops\Terraform\_code\Todoapp\Backend\_Todo\pytodobackend [adminuser@4.247.162.13:/home/adminuser/](mailto:adminuser@4.247.162.13:/home/adminuser/)

adminuser@todobackedn-vm: ls >> below command need to be run

# Switch to root

sudo su

# Update and install required packages

apt-get update && apt-get install -y unixodbc unixodbc-dev

curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add -

curl https://packages.microsoft.com/config/debian/10/prod.list > /etc/apt/sources.list.d/mssql-release.list

apt-get update

ACCEPT\_EULA=Y apt-get install -y msodbcsql17

# Install Python packages

pip install -r requirements.txt

uvicorn app:app --host 0.0.0.0 --port 8000

Result will be appear like this

INFO: Started server process [5425]

INFO: Waiting for application startup.

INFO: Application startup complete.

INFO: Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)

INFO: 152.58.128.70:14611 - "GET /api HTTP/1.1" 200 OK

go to your local browser >> publicip:8000/api >> now something will be appear

Till now we have created a Database server & Backend VM and both got connected through connection string now we need to create a frontend VM and if we will provide ip of backend vm to frontend vm then frontend vm and backend vm will get connected with each other.

Backend

Frontend

Database

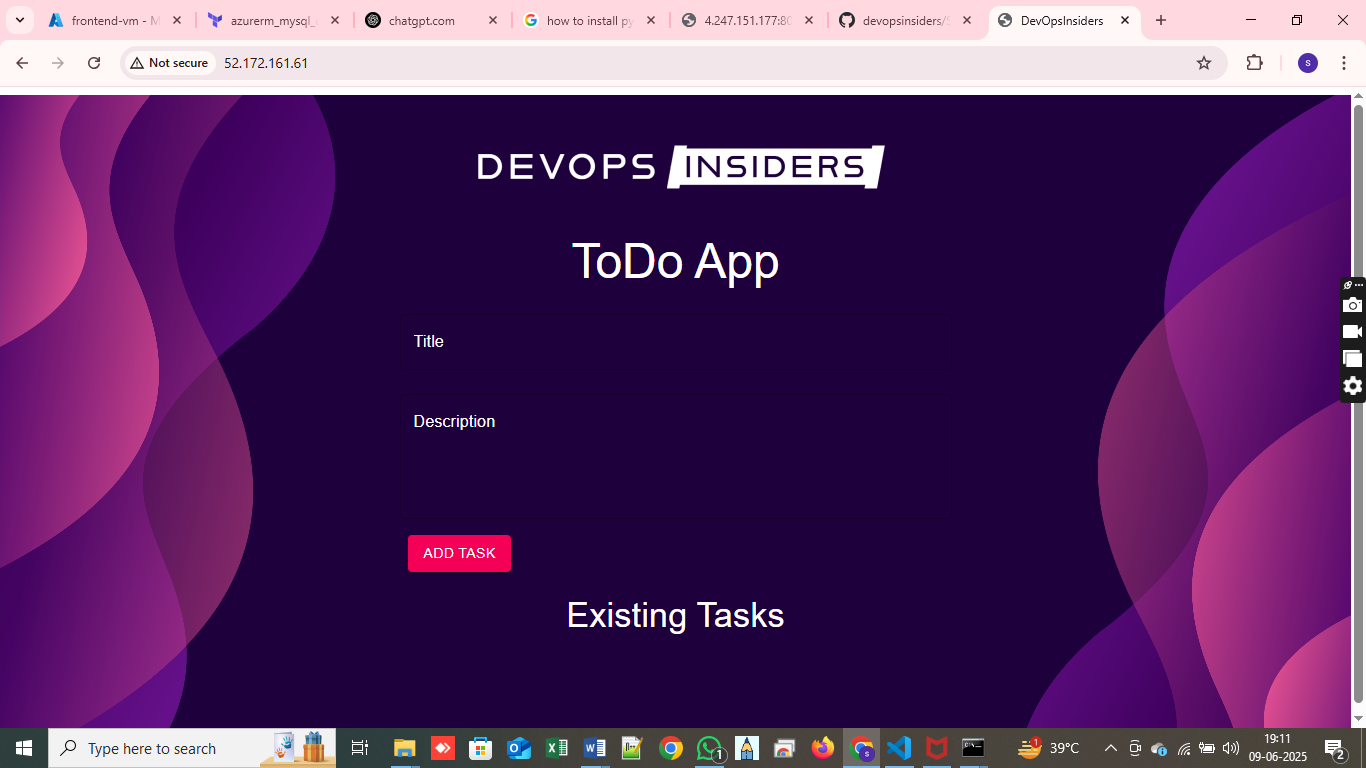
Backend VM IP Connection String

### ✅ Task 3: Frontend Deployment (React on VM)

1. Create a Linux VM (Ubuntu 20.04) on Azure and assign a Public IP.
2. Install Node.js, NPM, and Nginx.
3. Clone the React frontend repo. >> git clone <https://github.com/devopsinsiders/ReactTodoUIMonolith.git>
4. Go to that folder >> cd .\ReactTodoUIMonolith\
5. Update the src/TodoApp.js file to point to your backend public IP and port.
6. Build app using >> npm install >> npm run built
7. Deploy the build artifacts to /var/www/html/ using Nginx.
8. Access the app via: >> http://<frontend-public-ip>

Bhai agar website fir bhi na run kre to niche wala command bs frontend Vm me chala dena bs

1. sudo chown -R www-data:www-data /var/www/html
2. sudo chmod -R 755 /var/www/html
3. http://<your-public-ip>



But till now frontend application is not connected with the backend VM then:-

Azure Policy – Built-in, custom

Managed Identity –1. User assigned (Resource Independent, resource delete hone pr delete nhi hogi),

2. System assigned (Managed by Azure, Resource dependent, resource delete hone pr delete ho jayegi)

Endpoint

(Ek koi point jha pr jane pr service accessible ho)

For Example: - Agar hm mobile phone ki dukaan pr phone lene jate hai to hmare liye phone ki dukaan endpoint hai, jha pr hme kuch services mil rhi hai.

For Example: - [www.google.com](http://www.google.com) ye ek URL ho bhi hai aur ek endpoint bhi hai.

Agar koi server chal rha hai to server ki IP ya to uska URL endpoint hai.

Endpoint AAS service pr bhi hota hai aur PAAS service pr bhi work krta hai jaise storage account, key vault.

Endpoint hmesha across the region kaam bhi krta hai, bus condition hai ki vm aur subnet same region me hona chahiye storage account dusre region me so sakta hai

