Task 7: Disaster Recovery & High Availability

This task focuses on implementing **backup and disaster recovery (DR) strategies** to ensure **high availability** of the deployed application on **Azure App Service**.

📌 Overview

Disaster recovery and high availability are crucial for:
✅ **Minimizing downtime** during failures.
✅ **Protecting data** from accidental loss or cyberattacks.
✅ **Ensuring business continuity** with automated failover mechanisms.

This document covers:
- 🔹 **Disaster Recovery Strategy (Including RTO & RPO)**
- 🔹 **Automated Backups in Azure**
- 🔹 **High Availability Implementation**

📌 1. Disaster Recovery (DR) Strategy

🔹 What is Disaster Recovery?

A **Disaster Recovery Plan (DRP)** ensures that the system can **recover quickly from failures** such as:

- Server crashes
- Data loss
- Cyberattacks
- Network failures

🔹 Key Metrics for Disaster Recovery

Metric	Definition	Target
RTO (Recovery Time Objective)	Maximum time to restore service after failure	**≤ 15 minutes**

RPO (Recovery Point Objective)	Maximum data loss allowed during recovery	**≤ 5 minutes**

✅ **Lower RTO and RPO ensure faster recovery with minimal data loss.**

📌 2. Backup & Disaster Recovery Implementation in Azure

🔹 Step 1: Enable Automated Backups in Azure App Service

1. **Go to Azure Portal → App Services**.
2. Click on **devops-pythonwebapp → Backups**.
3. Click **Configure Backup**.
4. **Choose a Storage Account** to store backups.
5. **Set Backup Frequency** (e.g., every **6 hours**).
6. **Enable Retention Policy** (e.g., keep backups for **30 days**).
7. Click **Save**.

✅ **Azure will now automatically back up the application at regular intervals.**

🔹 Step 2: Implement Database Backups in Azure

1. **Go to Azure Portal → Azure Database for PostgreSQL (or MongoDB)**.
2. **Enable Point-in-Time Restore**:
    - Set **backup retention period** (e.g., **7 days**).
    - Configure **geo-redundant backups** (replication across regions).

Enable Read Replicas:
sh
CopyEdit

```
az postgres server replica create --name replica-db --source-server devops-db
```

3.

Test Recovery Process:
sh
CopyEdit

```
az postgres server restore --name devops-db-restore
--restore-point-in-time "2025-02-10T12:00:00Z"
```

4.

✅ **Database backups ensure data recovery with minimal loss (RPO ≤ 5 minutes).**

📌 3. High Availability Implementation

🔹 Step 3: Deploy Application in Multiple Regions (Geo-Redundancy)

1. **Go to Azure Portal → Traffic Manager**.
2. Click **Create Profile** → Select **Routing Method**:
   - **Priority** (Failover to backup region).
   - **Performance** (Route users to nearest region).
3. **Add Endpoints**:
   - **Primary Region** (e.g., East US).
   - **Secondary Region** (e.g., West US).
4. Click **Save**.

✅ **If the primary region fails, traffic is automatically redirected to the secondary region.**

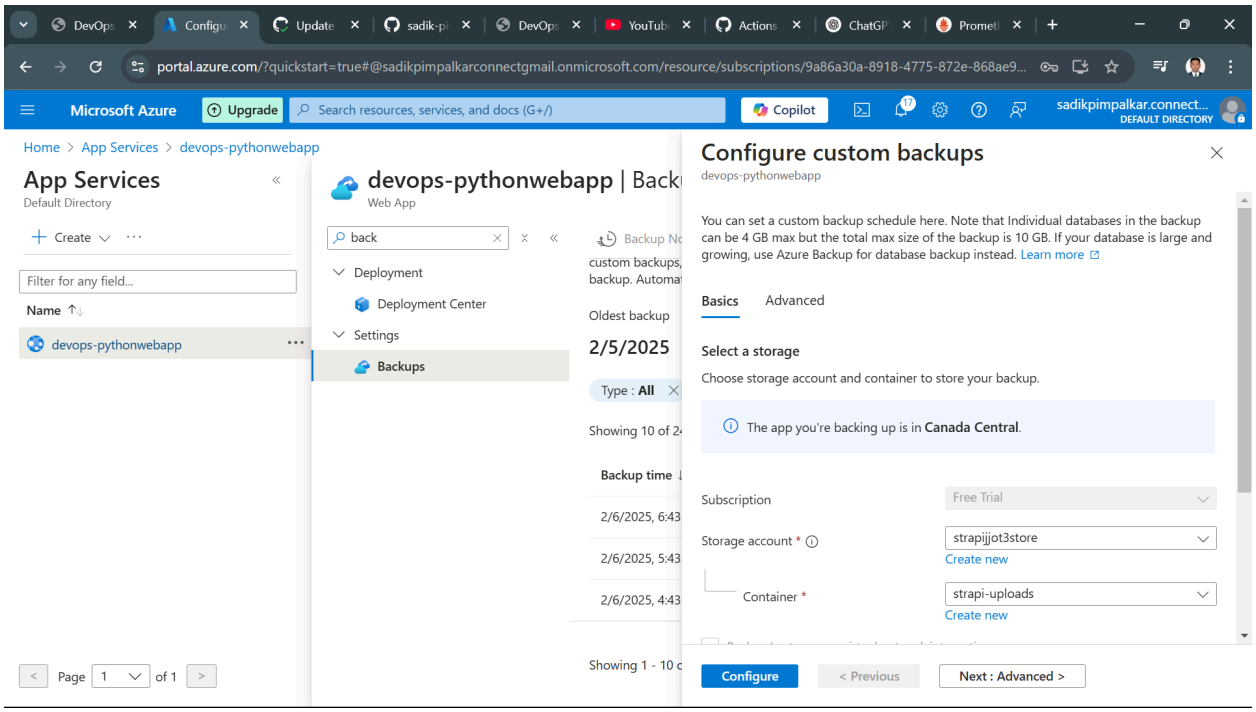
🔹 Step 4: Set Up Auto-Scaling for High Availability

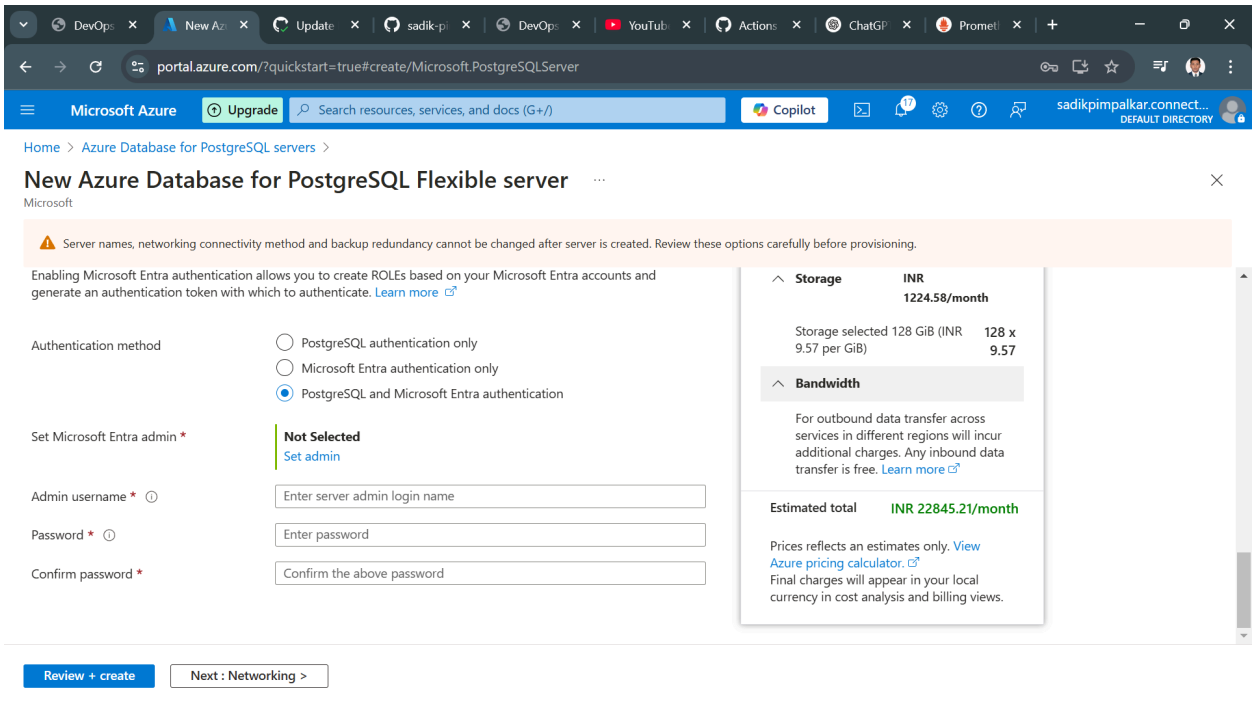
1. **Go to Azure Portal → App Services → Scaling**.
2. Enable **Autoscale** → Configure:
   - **Minimum Instances:** 2
   - **Maximum Instances:** 5
   - **CPU Threshold:** 70%
3. Click **Apply Changes**.

✅ **Auto-scaling ensures the application can handle traffic spikes automatically.**

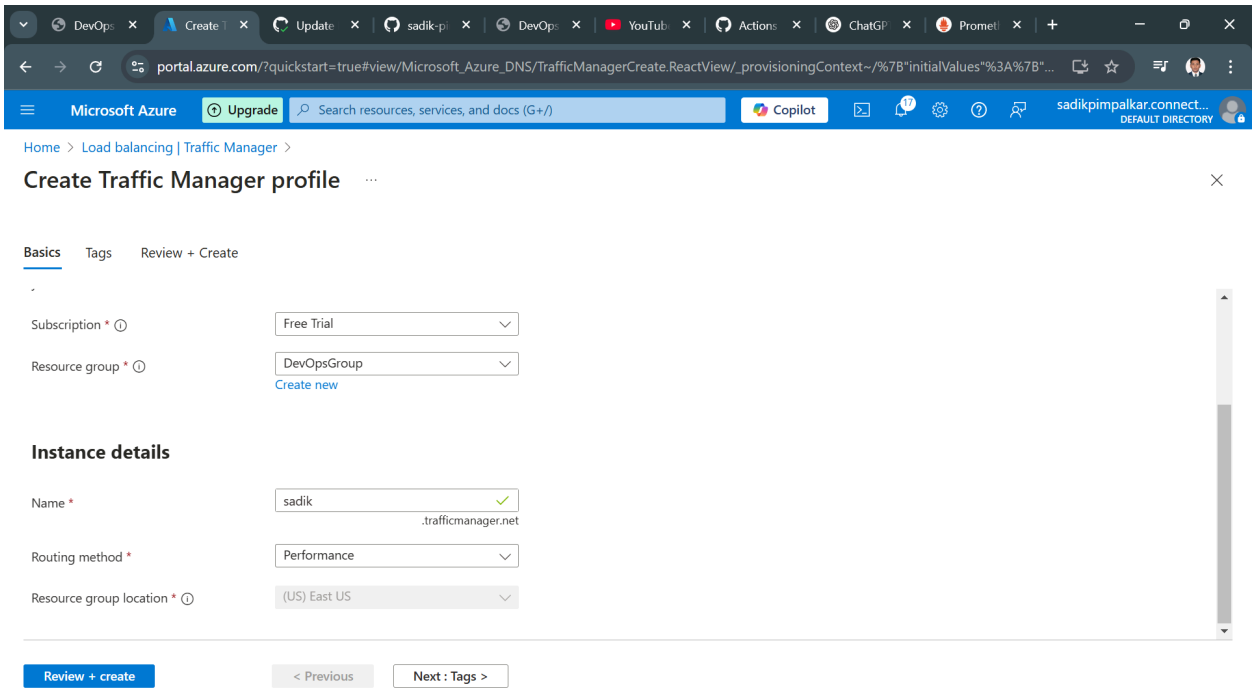
📌 4. Screenshots & Proof of Implementation

Azure Backups

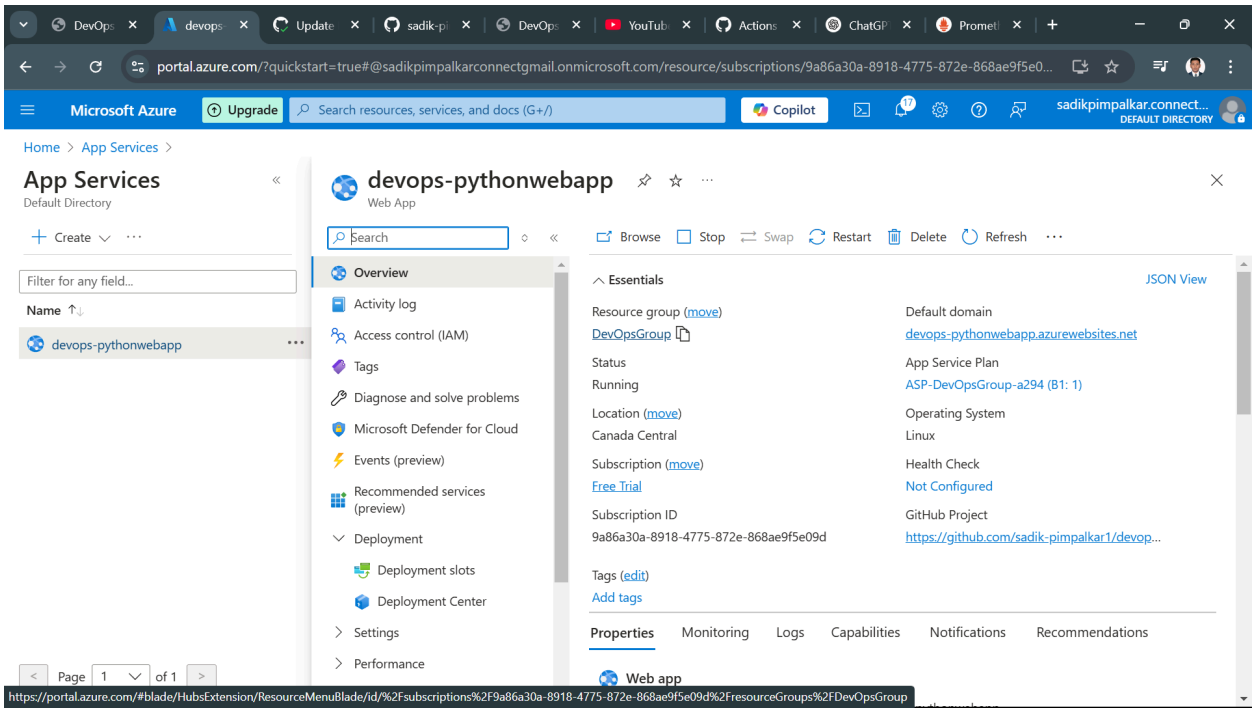




Traffic Manager.



Auto-Scaling



📌 5. Conclusion

✅ **Azure Backups ensure disaster recovery with minimal data loss (RPO ≤ 5 min).**
✅ **Geo-Redundancy & Auto-Scaling ensure high availability (RTO ≤ 15 min).**
✅ **Traffic Manager enables automatic failover in case of failure.**

🚀 **Now, the application is fully resilient to disasters and system failures!** 🎉