**Case Study on Azure Development**

**Problem Statement**

Stratacent is planning to scale it’s business operations and plans to migrate it’s solutions to cloud . They chose Azure as the choice of platform and wants to use it’s Platform as a Service offerings. They have the following architecture:

**Application Architecture**

Stratacent has a full stack application developed in Python using Flask for it’s backend. It hosts a list of APIs in the backends which store the data in Mongo DB

**Planned Migrations**

For deploying the application to Azure, Stratacent team chooses to use Azure App Service which can be used for direct deployment from Visual Studio Code.

The team will be deploying the application initially on a test environment which can be swapped with the production environment once the app is ready for production. This process should be automated and every time the app is deployed to the test slot the configuration of the test slot should appear in the production slot after the swap

**Traffic Management**

They also want to use Canary Deployment pattern and want to route 20% of the traffic to the test environment to check if some of the users who are redirected to the test website are reacting positively to the new features deployed.

Also, in case of high demand they want their application should be able to scale out to handle the load automatically. The app should add another instance as the CPU% goes above the average value and should be calculated every 5 minutes for the last 5 minutes. It should revert to single instance once the value falls below average. Failsafe values should be added to handle any conflicts and notification should be sent to the admins once the scaling happens

**Performance Optimization**

To optimize performance, the app content should be cached using a service which will allow the cache to be refreshed after a set expiration time

The app needs to deployed in 5 different regions: Central India, South India, East US, West US and Central US and should be available from all those regions which needs to be checked at an interval of every 15 minutes for a 200 OK response. In case the app is not available from any region the admin should be notified via mail or SMS with the severity level as Critical

As soon as the deployment succeeds, a notification related to the swapping of the environments should be sent as a message to a storage service using an automated solution

**Security Concerns**

Stratacent wants to prevent any unauthorized access to the app and wants only their users to be able to access the app only through their Microsoft accounts either using personal accounts or accounts which are registered with Azure Active Directory.

All API requests should be handled and to prevent any DDoS attacks the API calls should be limited to not more than 3 calls in a minute. API calls made to the endpoints should have access to read data from the web app using a secure solution

To secure access for solutions writing data to storage, a secure solution should be used with appropriate permissions for accessing the storage

**Cost Requirements**

The App Service Plan should offer support for dedicated infrastructure with features for scaling and testing deployment but offer economical solution without any premium features

The service used to handle the APIs should minimize cost and does not require to offer any SLA.

The storage service does not need to offer any features for backing up the data in availability zones or a different region

The caching solution can be created using basic features offered by Microsoft