Case Study Problem Description 1:

Secure Web Application Deployment

Company Background:

ABC Corporation is a leading e-commerce company that wants to deploy a new web application to handle customer orders and payments. The web application is built using Python and uses a MySQL database. The company wants to ensure that the web application is secure, scalable, and highly available.

Requirements:

1. Security: The web application must use HTTPS to encrypt data in transit. The company wants to obtain a trusted SSL certificate and private key to secure the web application.
2. Scalability: The web application must be able to handle a large number of concurrent requests without a significant decrease in performance.
3. High Availability: The web application must be available 24/7 with minimal downtime.
4. Automated Deployment: The company wants to automate the deployment process using a CI/CD pipeline to reduce manual errors and improve efficiency.

Current Infrastructure:

1. Source Code: The source code for the web application is stored in a GitHub repository.
2. Docker Image: The web application is packaged as a Docker image and stored in a Docker Hub repository.
3. Azure Subscription: The company has an Azure subscription with access to Azure Kubernetes Service (AKS) and other Azure services.

Challenge:

The company's IT team is responsible for deploying the web application to production. However, they face challenges in ensuring the security, scalability, and high availability of the web application. They want to automate the deployment process using a CI/CD pipeline and ensure that the web application is secure, scalable, and highly available.

Your Task:

Design and implement a solution that meets the requirements and challenges outlined above. The solution should include the following components:

1. CI/CD Pipeline: Implement a CI/CD pipeline using GitHub Actions to automate the deployment process.
2. Docker Image: Use Docker to package the web application and store it in a Docker Hub repository.
3. AKS Cluster: Create an AKS cluster to deploy the web application.
4. Ingress Controller: Configure an ingress controller to expose the web application to the internet using HTTPS.
5. SSL Certificate: Obtain a trusted SSL certificate and private key to secure the web application.

Deliverables:

1. A detailed design document outlining the solution architecture.
2. A GitHub Actions workflow file that automates the deployment process.
3. A Dockerfile that packages the web application.
4. An AKS cluster configuration file that deploys the web application.
5. An ingress controller configuration file that exposes the web application to the internet using HTTPS.
6. A report that summarizes the solution and its benefits.