DevOps Internship - 2024

Infrastructure setup on AWS using DevOps best Practices!

Month 1: Setting Up Infrastructure and CI/CD Pipeline

1. Week 1: Planning and Setting Up AWS Resources
   - Set up an AWS free tier account.
   - Use Terraform to provision necessary AWS resources like EC2 instances, VPC, security groups, and RDS (for the database).
   - Create an IAM role for Jenkins.

2. Week 2: Configuring Jenkins for CI/CD
   - Install Jenkins on an EC2 instance.
   - Configure Jenkins with necessary plugins for Node.js, AWS, and Git.
   - Set up a Jenkins pipeline for the Node.js application that triggers on code commits.

3. Week 3: Developing the Node.js Application
   - Start building the Node.js application with Express.js.
   - Implement user authentication using libraries like Passport.js.
   - Set up routes for user registration, login, and logout.

4. Week 4: Integrating CI/CD Pipeline
   - Connect Jenkins to the Git repository of the Node.js application.
   - Create Jenkins jobs to build, test, and deploy the application.
   - Ensure the CI/CD pipeline is triggered automatically on code changes.

Month 2: Implementing Monitoring and Logging

1. Week 1: Setting Up Prometheus and Grafana
   - Deploy Prometheus for monitoring metrics.
   - Install Grafana and configure it to visualise Prometheus metrics.

2. Week 2: Instrumenting Application for Metrics
   - Integrate Prometheus client library into the Node.js application to expose custom metrics.
   - Define relevant metrics to monitor, such as HTTP request latency, error rates, etc.

3. Week 3: Configuring Loki and Promtail
   - Deploy Loki for logging.
   - Install Promtail on the application server to collect logs.

4. Week 4: Visualising Logs in Grafana
   - Configure Grafana to visualise logs from Loki.
   - Create dashboards to monitor application logs and performance metrics.

Month 3: Refining and Scaling

1. Week 1-2: Load Testing and Optimization
   - Use tools like Apache JMeter or Gatling to perform load testing on the application.
   - Identify performance bottlenecks and optimise the application code and infrastructure accordingly.

2. Week 3: Implementing High Availability
   - Configure auto-scaling for EC2 instances based on CPU or custom metrics.
   - Set up a multi-AZ RDS instance for high availability and failover.

3. Week 4: Documentation and Finalization
   - Document the entire setup, including infrastructure, CI/CD pipeline, monitoring, and scaling strategies.
   - Conduct a final review of the project and make any necessary adjustments.

Throughout the project, ensure to keep track of costs to stay within the AWS free tier limits. Additionally, prioritise security by implementing best practices for IAM, network security, and application security.