

Senior SRE Hiring Exam Instructions

Objective: Assess your ability to containerize a Golang application, create a CI/CD pipeline, and set up infrastructure as code for deployment on AWS using Terraform.

Duration: ~4 hours

Exam Overview

You will be provided with a simple Golang application. Your tasks are to:

1. **Containerization:** Write a Dockerfile to containerize the application.
 2. **CI/CD Workflow:** Create a CI/CD workflow to build and push the Docker image to a container registry.
 3. **Infrastructure as Code:** Write Terraform configuration files to deploy the containerized application on AWS, including setting up a Postgres database either in Docker or on RDS.
 4. **Documentation:** Document the entire process clearly and thoroughly.
-

Detailed Instructions

1. **Setup:**
 - This GitHub repository contains a simple Golang application that reads IPFS CIDs from a CSV file and saves the CID, name and image location to a postgres database and then provides an API to retrieve this data.
 - <https://github.com/MoFoLuWaSo/ipfs-metadata>
 - In order for this to run properly make sure to provide the database connection information in a `.env` file.
 - Ensure you have access to the following:
 - A Docker Hub account (or another container registry of your choice)
 - An AWS account for infrastructure testing
 - Fork the provided GitHub repository to your own GitHub account. All your work should be committed to this forked repository.
2. **Containerization:**
 - **Task:** Write a Dockerfile to containerize the provided Golang application.
 - **Requirements:**
 - The Dockerfile should be optimized for efficiency (e.g., minimal layers, appropriate base image).
 - Ensure the application runs correctly inside the container.

3. CI/CD Workflow:

- **Task:** Create a CI/CD workflow using GitHub Actions (or another CI/CD tool) to build and push the Docker image to a container registry.
- **Requirements:**
 - The workflow should build the Docker image and push it to your chosen container registry.
 - Handle secrets securely (e.g., using GitHub Secrets for credentials).

4. Infrastructure as Code (Terraform):

- **Task:** Write Terraform configuration files to deploy the containerized application on AWS.
- **Components to Include:**
 - VPC, Subnets, Security Groups
 - ECS Cluster and Task Definition (or EKS if more complex)
 - Load Balancer (ALB/NLB)
 - IAM Roles and Policies
 - Postgres Database setup options:
 1. **Option 1:** Set up the Postgres database in a Docker container within the ECS cluster.
 2. **Option 2:** Set up the Postgres database on AWS RDS.
- **Requirements:**
 - The Terraform code should correctly provision the required AWS resources.
 - Use modules where appropriate for reusability and clarity.
 - Follow Terraform and AWS best practices.

5. Documentation:

- **Task:** Provide clear and thorough documentation for the entire process, including:
 - How to build and run the Docker container.
 - How to trigger the CI/CD pipeline.
 - How to deploy the application using Terraform.
 - Any assumptions and prerequisites.
- **Requirements:**
 - Documentation should be easy to follow and complete.
 - Include any relevant commands, configurations, and explanations.

6. Submission:

- Once you have completed all tasks, commit your work to your forked GitHub repository.
- Submit the link to your forked repository.

Evaluation Criteria

● Technical Skills:

- Proficiency in Docker, CI/CD tools, Terraform, and AWS.

- Code quality and adherence to best practices.
- **Problem Solving:**
 - Ability to troubleshoot and resolve issues.
 - Creativity in optimizing workflows and infrastructure.
- **Communication:**
 - Clarity and completeness of documentation.
 - Ability to explain technical choices and processes.

Note: You are not required to deploy the application to AWS as part of this exam. We will test the deployment using our environment based on the provided Terraform files and documentation. Please ensure your configurations and instructions are accurate and complete.

Good luck! If you have any questions during the exam, feel free to reach out for clarification.