Go Development -

Firefly Home Assignment

Overview

This assignment evaluates your Go programming skills through the development of a tool to detect infrastructure drift between AWS EC2 instances and their Terraform configurations. You'll build a program that compares an EC2 instance's actual configuration in AWS with its expected configuration in Terraform.

Requirements

Core Functionality

1. Create a Go application that:
    - Retrieves EC2 instance configuration from AWS
    - Parses a Terraform state file or HCL configuration for the same instance
    - The program should compare two configurations and detect drift across a list of specified attributes, including but not limited to **instance_type**. It should return whether a drift is detected for any attribute in the list and specify which attributes have changed - consider handling multiple type of fields and nested ones
    - Reports detected drift in a structured, easy-to-understand format

Technical Specifications

1. Use Go modules for dependency management
2. Implement proper error handling and logging
3. Include unit tests with at least 70% code coverage
4. Document your code following Go best practices
5. Use the AWS SDK for Go v2 for AWS interactions
6. Use the Terraform Go SDK or custom parsing for Terraform configuration

Deliverables

1. Complete source code, including:
    - Main application
    - Unit tests
    - Documentation
2. A README.md file containing:
    - Project overview
    - Setup and installation instructions
    - Usage examples
    - Design decisions and trade-offs
    - Ideas for future improvements
3. A sample input Terraform configuration
4. A sample AWS EC2 response (or mock data)

Project Setup Instructions

Environment Setup

1. Ensure Go 1.19+ is installed
2. Configure AWS credentials (for testing)

Sample Data

AWS EC2 Response

Provide a sample JSON structure for the AWS EC2 response, including at least the instance type.

Terraform Configuration

Provide a sample Terraform configuration (HCL or JSON) that defines an EC2 instance with a specific instance type.

Evaluation Criteria

1. Code quality and organization
2. Proper error handling
3. Test coverage and quality
4. Documentation clarity
5. Performance considerations
6. Understanding of Go concurrency patterns (if applicable)
7. Clean and idiomatic Go code

Bonus Question

Extend the application to handle drift detection for multiple EC2 instances and additional attributes (security groups, tags, etc.) concurrently using Go's concurrency primitives. Implement a command-line interface that allows users to specify which attributes to check for drift.

Submission Guidelines

1. Create a private GitHub repository
2. Upload your code and documentation
3. Ensure all tests pass
4. Add clear instructions for running your application
5. Add README with approach, decisions, and challenges faced

Good luck! We look forward to reviewing your solution.