National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Vikram Kariyavula
Student ID: 24112682

School of Computing
National College of Ireland

Supervisor: Shivani Jaswal

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Vikram Kariyavula……

Student ID: ……24112682…….……

Programme: ……Cloud Computing…… **Year:** …2025……

Module: ……MSc Research Project……

Lecturer: …… Shivani Jaswal ……

Submission Due Date: ……28-01-2026……

Project Title: …… Configuration Manual……

Word Count: ……1253…… **Page Count:** ……9……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Vikram kariyavula……

Date: ……28-01-2026……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

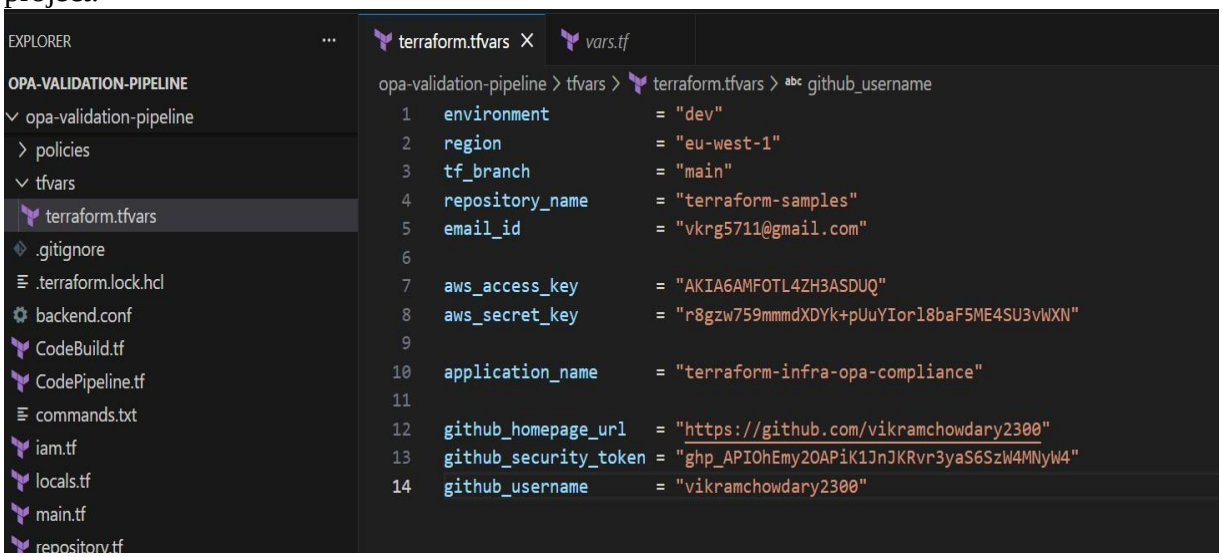
Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Vikram Kariyavula
Student ID: 24112682

1 Understanding the configuration file terraform.tfvars

The terraform.tfvars file provides the values of the user defined variables that are used in the terraform configuration. They often set above variable of tfvars allows a clear separation of concerns and enhances security and removes sensitive values in the hard-coded .tf files. They are listed below, and they are the keys that are deemed to be applicable in the present thesis project.



1.1 environment:

This refers to the implementation stage (dev, production etc..,). In Terraform, it represents a value that helps distinguish resources according to the specific lifecycle phase they belong to.

1.2 region:

Determines the region which is characterized by AWS of cloud resources (e.g., eu-west-1). This is geographical positioning influences the cost and compliance requirements and latency.

1.3 Tf branch:

It indicated the branch in the git hub for example main branch. From which source should be pulled.

1.4 Repository name:

The name of a GitHub repository of the Terraform configuration

1.5 Email id:

Subscribes of the SNS topic to the email address to receive the alert particularly where the OPA analysis of policy run is not successful. In addition, the user will be informed automatically when compliance rule is not adhered to.

1.6 Aws access-key / aws secret-key:

Terraform is used together with AWS programmatically authentication credentials. The cloud resources can be deployed using these keys, modified using these keys and states of the resources are managed.

1.7 Application name:

Name (application name). Name applied to the logical application (e.g. opa-compliance-pipeline). To achieve traceability, it is used in naming standards and tagging.

1.8 Github URL:

This refers to the repository url in the github. It can be utilized when setting up CI trigger.

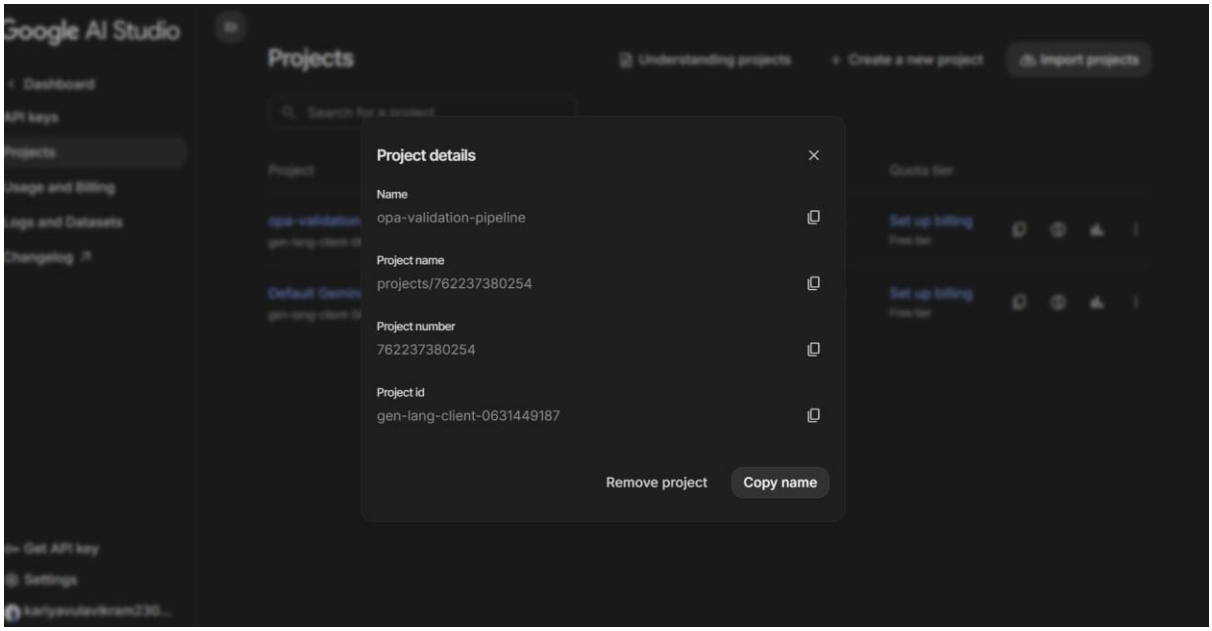
1.9 Github security token:

GitHub Personal Access Token (PAT) that would be used to communicate Terraform and CI/CD pipeline with GitHub API. This is used to clone the repository, push the changes and make events

1.10 Git hub username:

The attached GitHub account that serves to identify what has been done on the repository and CI triggers both.

2 How to Create and Set Up a Gemini API Key



Gemini API keys are obtained through Google AI Studio, which provides access to Gemini 1.5 models.
Steps to Generate Gemini API Key
1. Navigate to: https://aistudio.google.com
2. Sign in with your Google account.
3. Go to API Keys on the left sidebar.
4. Click Create API Key.
5. Copy the generated key.

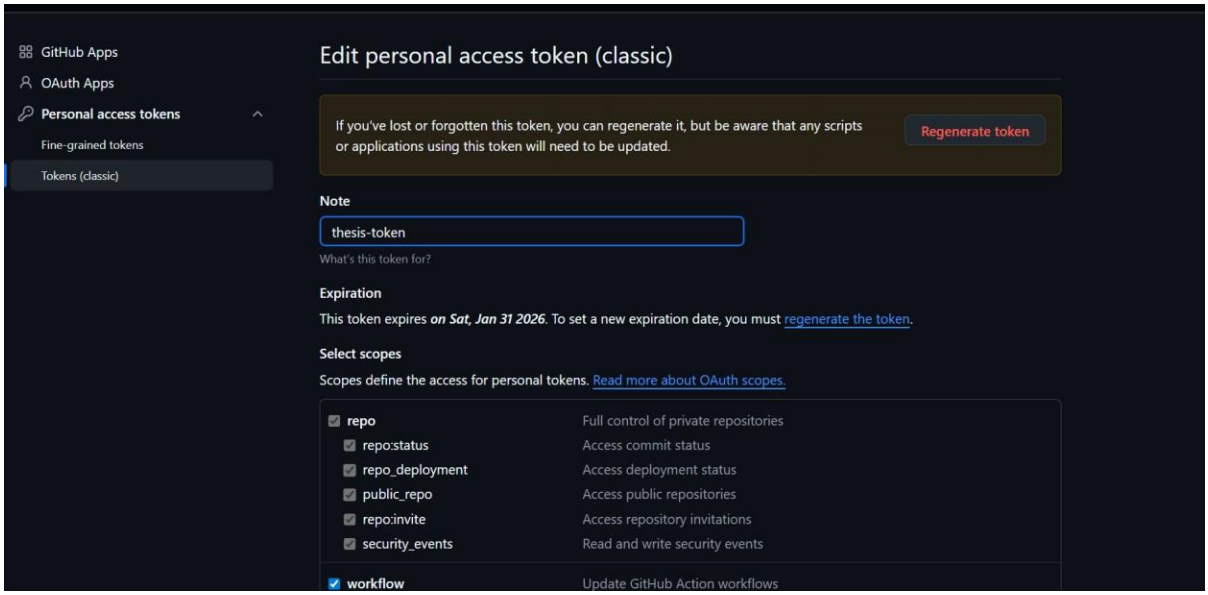
Usage in Project:

- Add the key to environment variables or configuration files.
- The key is referenced inside Terraform via .tfvars.

3. Creating and Setting up GIT HUB Personal Access Token

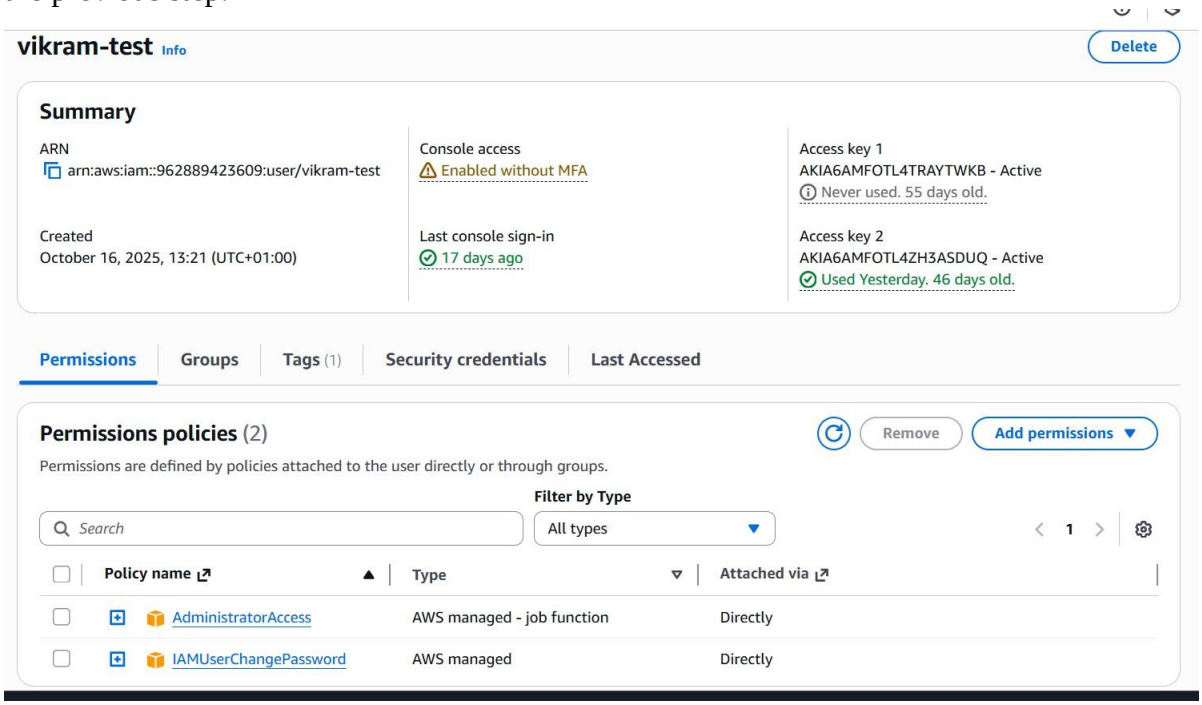
To authenticate Terraform and AWS Code Pipeline with GitHub repositories, a GitHub PAT (Personal Access Token) is needed.

Steps to create:
1) Go to GitHub
Visit https://github.com/settings/tokens.
2) Create New Token: Select "Generate New Token" (Classic) based on the parameters.
• Selecting a scope
• repo (full command over private repositories)
• proc (if relevant to GitHub Actions)
• To manage webhooks for triggers related to continuous integration
3) Establish Expiration and Take note: Include a description, like "thesis token," and choose a suitable expiration date, like 30 or 60 days.
4) Make a duplicate of the token
Crucial: Make a copy of this value and store it securely because it can only be viewed once.
5) Make use of Terraform. tfvars: The github_security_token is "your-generated-token".



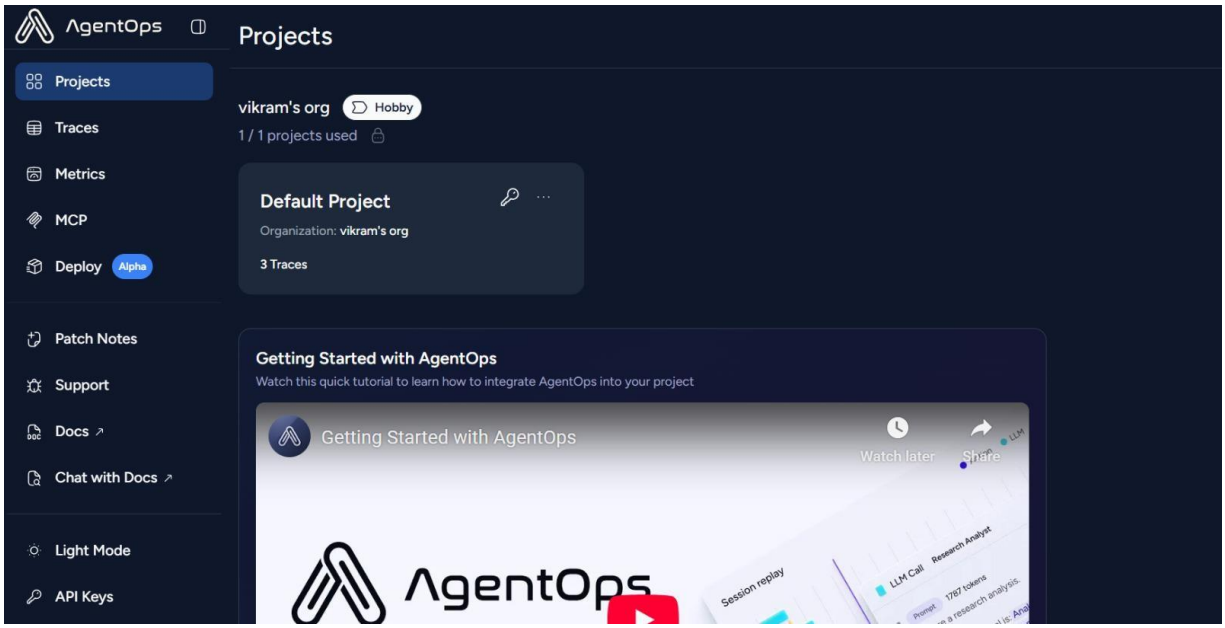
4 Setting Up Credentials for an AWS Account

Valid AWS credentials are necessary for infrastructure provisioning and application deployment utilizing Terraform and AWS Code services.
Method:
1) Open the AWS Console:
Visit https://console.aws.amazon.com.
2) Go to IAM > Users > Add User and choose Programmatic access to establish an IAM user.
Add AdministratorAccess or custom least-privilege policies.
3) Make Keys for Access:
From the IAM tab of the application menu, choose Users > [Your User] > Security credentials.
Select "Create access key." Select the use case for the Command Line Interface (CLI).
Get access to the credentials, which include the access key ID and secret access key.
Include the two credentials mentioned above in the terraform.tfvars file that was created in the previous step.



5 Creating AgentOps Key

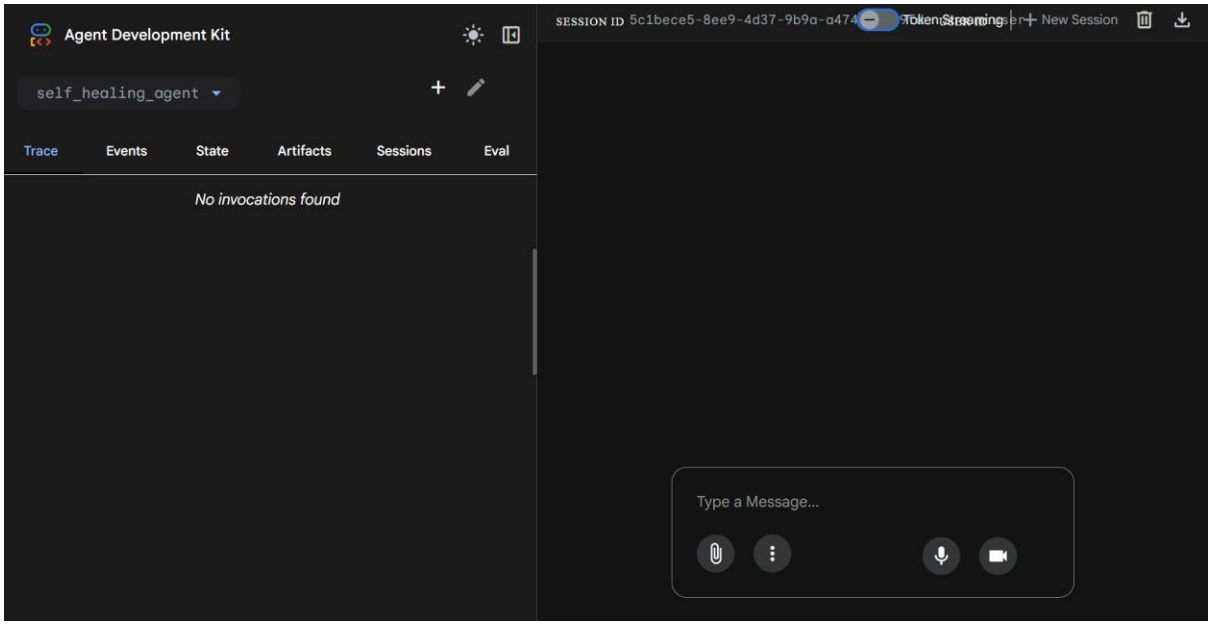
Agent behavior can be monitored and visualized with AgentOps.
In order to produce the API key:
Steps to create:
Go to https://www.agentops.ai, the AgentOps website.
Go to the API Keys area after logging in.
Make a fresh API key.
To the.env file, add the key



6. ADK Commands

Command line tools that enable local agent execution are included in the Agent Development Kit (ADK). “adk Web”

The ADK Web Interface, which displays workflows and enables the examination of agent states and logs, is launched by this command.



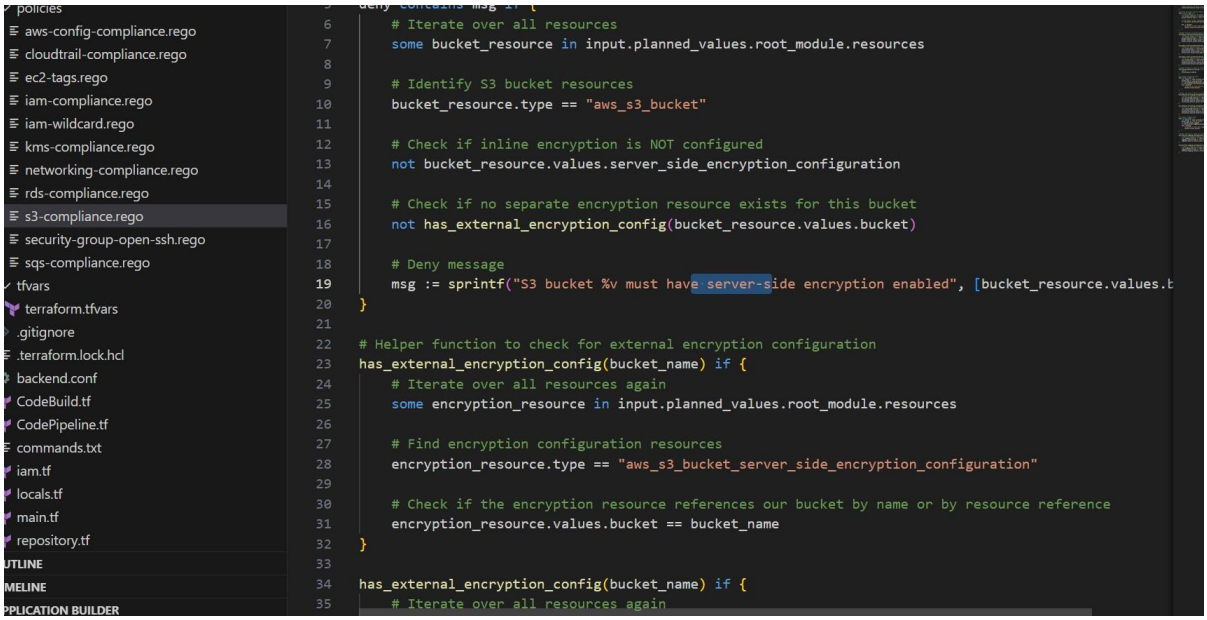
7. Looking at Agent Logs

- Logs are a crucial component in tracking the agent's and pipeline's behavior.
- AWS CodeBuild Logs: CloudWatch receives logs from each build action. Validation, planning, and compliance scan stages are among them.
- OPA Logs: OPA generates opa-results.json, which includes structured information on every policy infraction.

- ADK Agent Logs: The agent prints logs at each stage of execution, indicating which changes were tried and whether they were successful.
- SNS Notifications: An email is sent to the configured email address in the event that the pipeline fails or a violation is discovered.

8. OPA Policies

Rules for compliance are enforced by Open Policy Agent (OPA). Policies are kept in the policies/directory and authored in the Rego language. Terraform develops a plan for each pipeline run, and OPA compares it to the policies.



Key policies consist of:

Rego's s3-compliance
ensures that versioning is enabled, public access is blocked, and S3 buckets are encrypted.

iam-wildcard.rego
uses "." to identify IAM rules with excessively broad permissions.

open-ssh-security-group.rego
Security groups that permit SSH from any IP address are flagged.

Rego's kms-compliance
ensures that encryption at rest is done using KMS.

Compliance with cloudtrail.rego
makes sure CloudTrail is set up for auditing.

Database encryption, backup setup, and public accessibility are all checked by rds-compliance.rego.

9. Agent WorkFlow

ADK is used in the construction of the healing system, which has multiple agents and a controlled workflow.

The state:
The state contains shared data that is transferred between agents, including loop counters, file locations, Terraform output, and detected infractions.

Agent in Sequence:

The primary workflow is as follows:

- Create a repository
- Examine OPA findings.
- Perform a remediation loop
- Push and commit updates.

Concurrent Agent

For consistency, this project mostly employs sequential execution, even if ADK supports it. In subsequent development, independent tests can use parallel execution.

Loop Manager

Until the infractions are fixed, the Loop Agent resumes the remediation procedure. Every iteration completes:

- Use remediation
- Check syntax
- Verify using Terraform and OPA
- Choose whether to proceed.
- When the iteration limit is achieved or there are no infractions, the loop ends.

References

Spacelift (2025) Terraform.tfvars files: Variables management with examples. https://spacelift.io/blog/terraform-tfvars (Accessed: 24 April 2025)

GitHub (2025) Managing your personal access tokens. https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/managing-your-personal-access-tokens (Accessed: 24 April 2025).

Online Scientific Research (2023) Managing Identity and Access Management (IAM) in Amazon Web Services (AWS). https://onlinescientificresearch.com/articles/managing-identity-and-access-management-iam-in-amazon-web-services-awsnbsp.pdf (Accessed: 24 April 2025).

Open Policy Agent (2025) Rego policy language documentation. https://www.openpolicyagent.org/docs/latest/policy-language (Accessed: 24 April 2025).

Paul, A., Manoj, R. and S, U. (2024) 'Amazon Web Services cloud compliance automation with Open Policy Agent', *2024 International Conference on Expert Clouds and Applications (ICOECA)*, pp. 313–317. Available at: https://doi.org/10.1109/ICOECA62351.2024.00063 (Accessed: 24 April 2025).

Goldman Sachs Engineering (2024) Scaling OPA and managing OPA bundle signing. https://developer.gs.com/blog/posts/scaling-opa-for-oces (Accessed: 24 April 2025).

Conftest (2024) Conftest documentation. https://www.conftest.dev (Accessed: 24 April 2025).

Gitleaks (2024) Gitleaks documentation. https://github.com/gitleaks/gitleaks (Accessed: 24 April 2025).