**Terraform CI/CD and Testing on AWS**

[Terraform Testing Framework]

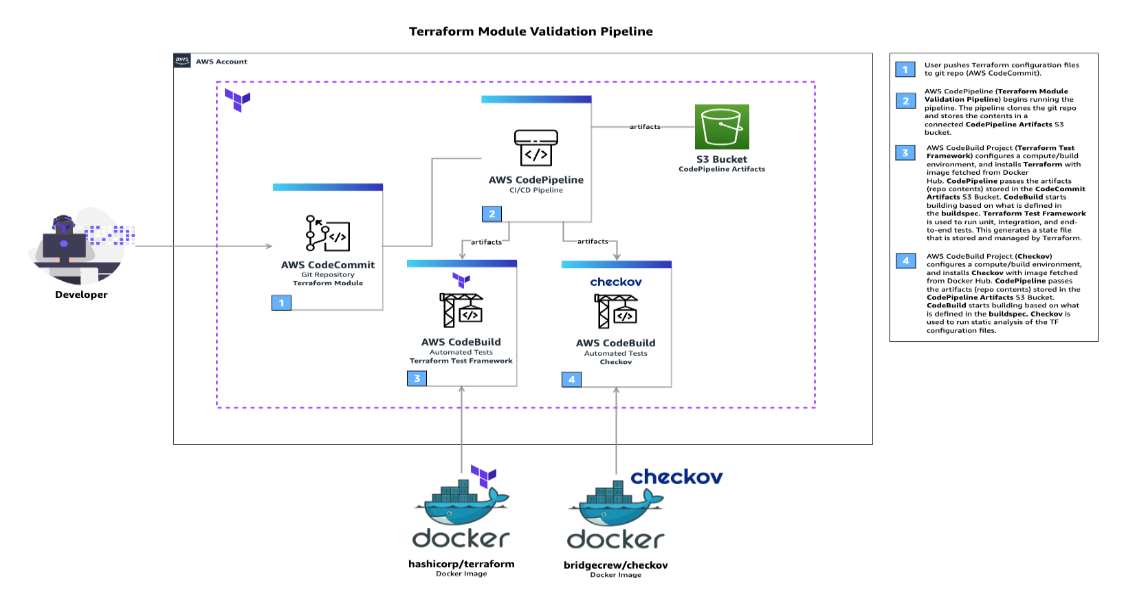
**Introduction**

Terraform Testing Framework allows users to perform manual unit and integration tests for terraform projects.

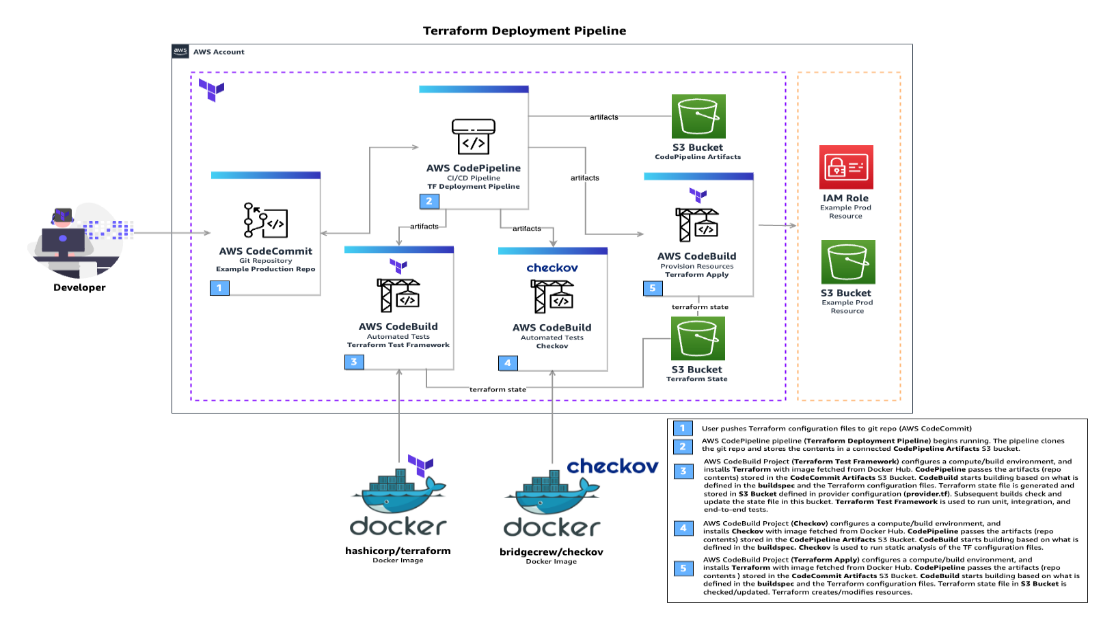
**Project Description - Scenario**

In this project I’m building a basic Terraform CI/CD pipeline on AWS to automate tests and deploy infrastructure safely and securely and validate the functionality and security of my configuration files with local testing, as well as with automated testing and deployment in CI/CD pipelines to deploy production workload to my AWS account.

**Architecture -Terraform Module Validation Pipeline - Overview**



**Architecture – Terraform Deployment Pipeline - Overview**



**Local Terraform Module Development and Testing.**

* Create a custom Terraform Module:
* module-aws-tf-cicd
* Test the security and functionality of the custom Terraform Module locally with the Terraform Test Framework and Checkov

**Creating input variables, providers, local values and data sources**

create\_codepipeline\_artifacts\_bucket, create\_codepipeline\_service\_role, create\_codebuild\_service\_role, create\_s3\_remote\_backend, s3\_public\_access\_block, codecommit\_repos, codebuild\_projects, codebuild\_service\_role\_arn, codepipeline\_pipelines.

**Data Sources**

The variable.tf file contains all the input variables.

A screen shot of a computer program

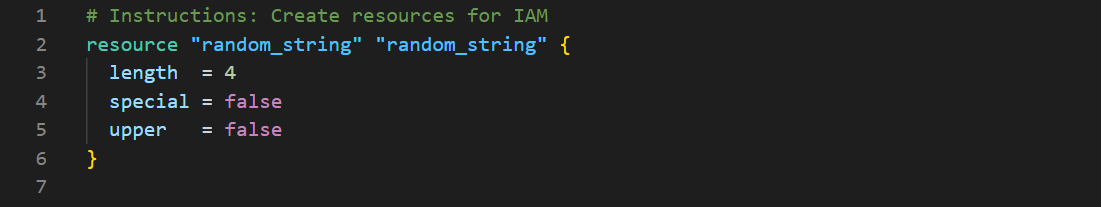
Description automatically generated

The data.tf file uses a Data Source to fetch the current AWS region.

A screenshot of a computer program

Description automatically generated

The iam.tf file uses a combination of Data Sources, Resources and Input Variables to dynamically create resources for IAM.



The s3.tf files are where the main S3 are defined e.g. The Artifact S3 Bucket that CodePipeline will store the artifacts in for each pipeline.

A screen shot of a computer code

Description automatically generated

The eventbridge.tf file is where EventBridge rules control when events are sent and what those events cause to happen.

A screen shot of a computer program

Description automatically generated

The codecommit.tf file dynamically creates AWS CodeCommit Repositories

A screen shot of a computer code

Description automatically generated

The codebuild.tf file dynamically creates AWS CodeBuild Projects.

A screen shot of a computer program

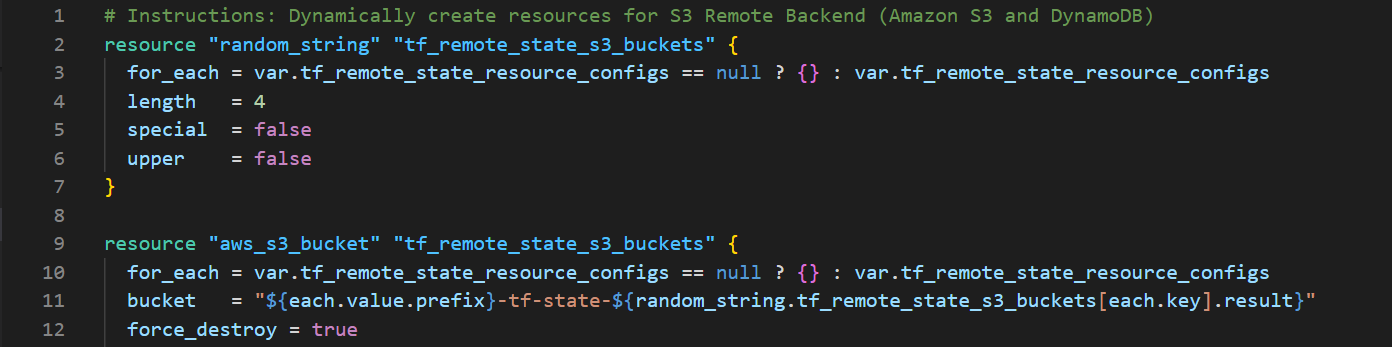
Description automatically generated

The codepipeline.tf file dynamically creates AWS CodePipeline Pipelines,

A screen shot of a computer program

Description automatically generated

The backend.tf file dynamically creates resources for storing Terraform State [S3 and DynamoDB]



The outputs.tf file defines the output values for the module.

A screenshot of a computer program

Description automatically generated

Module Configured.

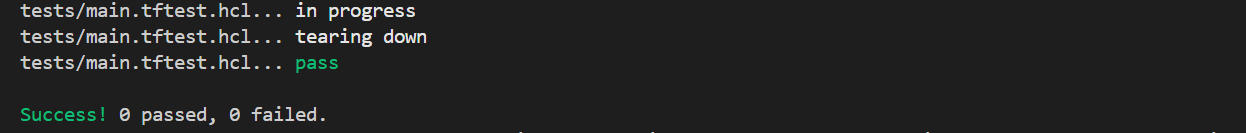
**Test Terraform Module [Terraform Test Framework]**

The main.tftest.hcl file is configured for testing using the Terraform Test Framework

A screen shot of a computer program

Description automatically generated

Testing the module with terraform test, this will run all the tests in the main.tftest.hcl file and show if it passed.



**Test Terraform Module [ Checkov] - Validated that you module is secure.**

A Policy-as-code tool that scans cloud infrastructure configurations to help identify misconfigurations before they are deployed, it can scan results across platforms such as Terraform, CloudFormation, Kubernetes, Helm, ARM Templates and Serverless framework.

A screen shot of a computer

Description automatically generated

**Deployment – Terraform CI/CD Pipelines**

Deploy Terraform CI/CD Pipelines (Terraform Module Validation Pipeline and Terraform Deployment Pipeline). Then use the pipelines to run automated tests when pushing code to git repos (Amazon S3 Buckets-Simulated).

**Deploying Resources - Creating Terraform Configuration Files**

The provider.tf file defines the provider.

**A black rectangle with white text

Description automatically generated**

The locals.tf defines multiple local values that can be used throughout the module.

A computer screen with text

Description automatically generated

Similar to the local.tf file to assign values that can be used in the configuration files.

A screenshot of a computer

Description automatically generated

The outputs.tf file

A computer screen shot of a code

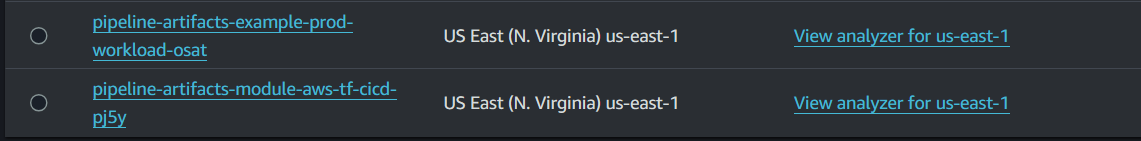
Description automatically generated

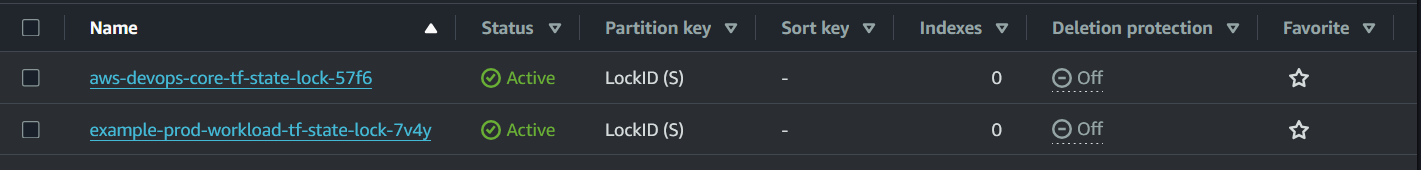
The main.tf file define the core terraform configuration

A screen shot of a computer program

Description automatically generated

Core configured!



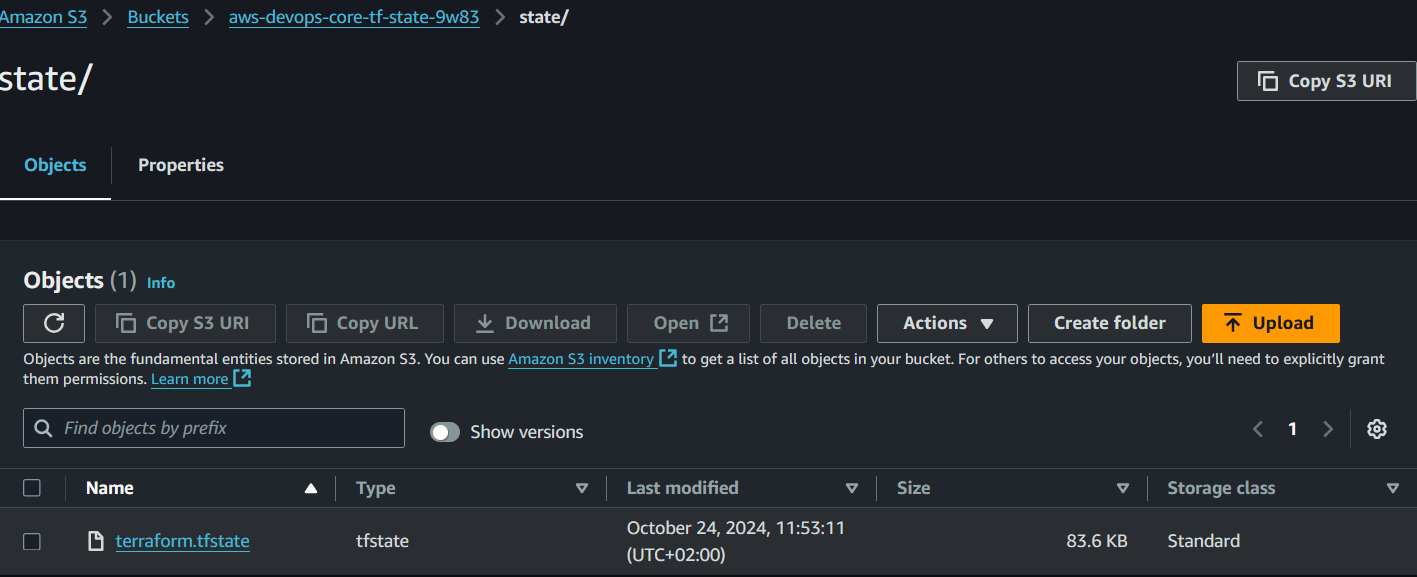


Successfully deployed resources!

Setting up Terraform Remote Backend to move the state file to the S3 bucket.

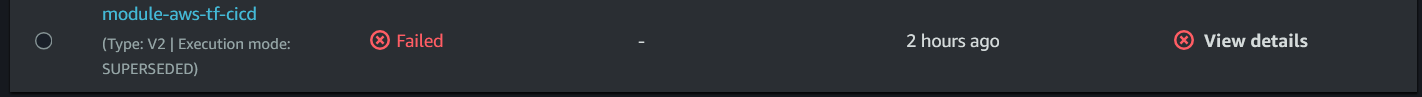
A computer screen with text

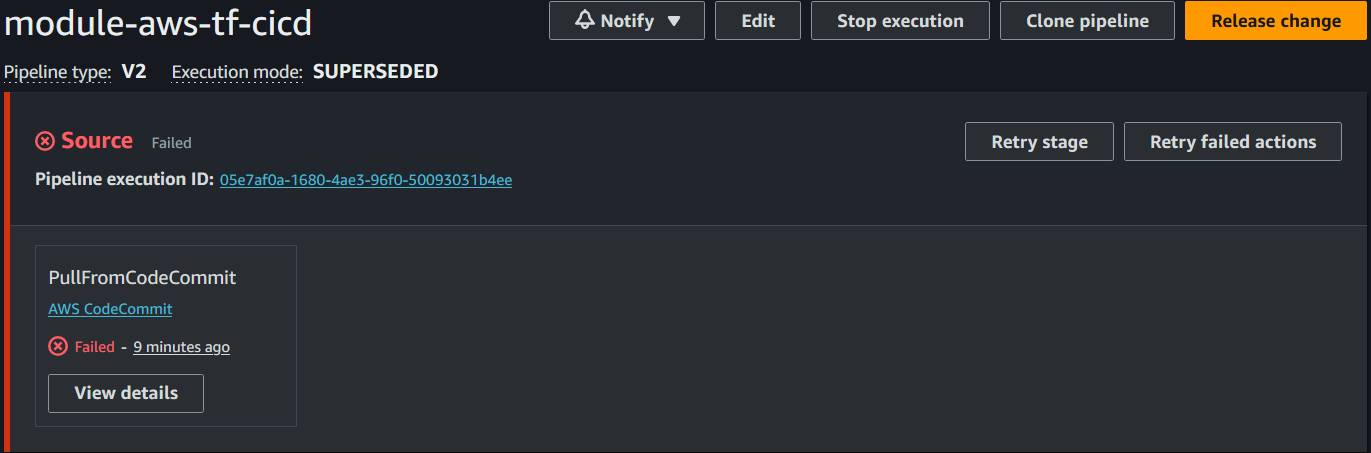
Description automatically generated



**Trigger Terraform Module Validation Pipeline**

Upload the Terraform Module code to the corresponding Amazon S3 bucket to test functionality of the Terraform Module Validation Pipeline. module-aws-tf-cicd

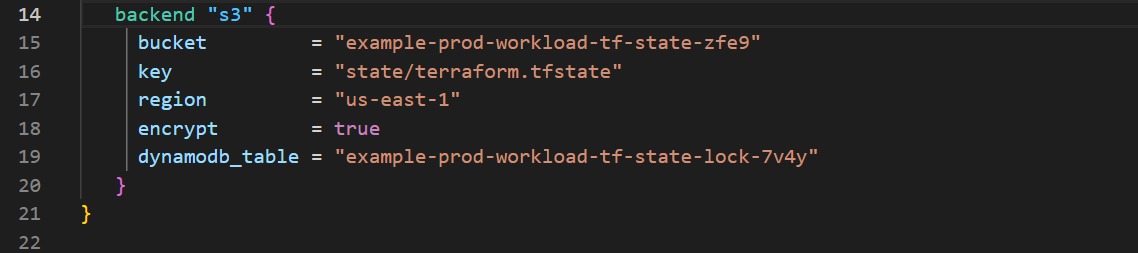




**Note:** Terraform Module Validation pipeline is **NOT** functioning correctly.

**Trigger Terraform Deployment Pipeline**

Updated.

****

A black screen with text

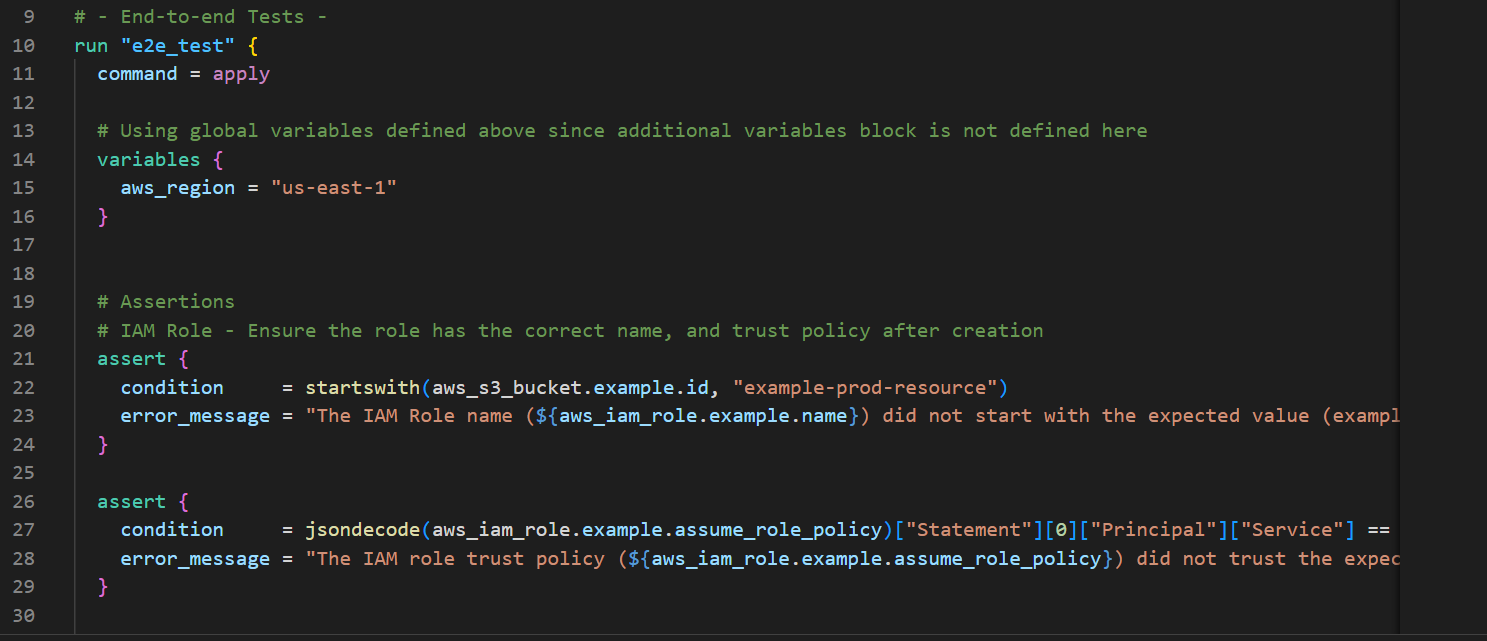
Description automatically generated

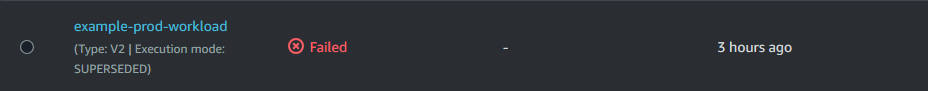
A screen shot of a computer program

Description automatically generated

Basic test for Production workload

main.tftest.hcl file





A screenshot of a computer

Description automatically generated

**Note:** The Pipeline is **NOT** functioning correctly.