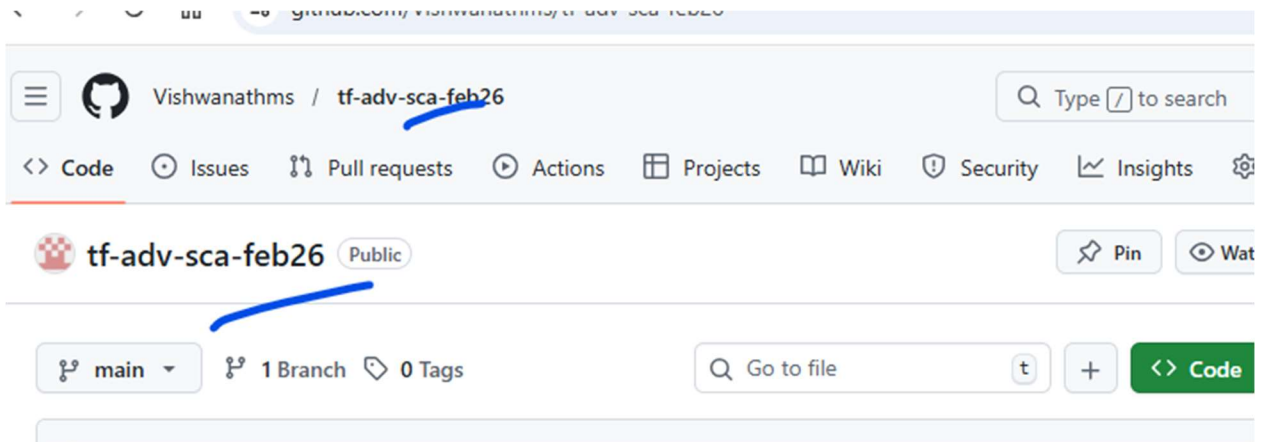


Tf-adv-lab1-Steps

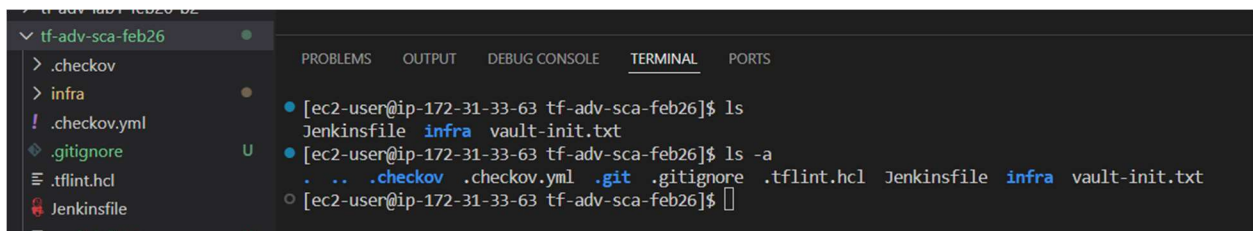
Steps:

1. Create github repo
2. Clone on the vs code
3. Copy the lab2 content to the new repo folder on vs code
4. Git commands to push to the github
5. Create an Jenkins pipeline (normal)
6. Run the Jenkins pipeline to check the output of “checkov”
7. Comment the checkov and uncomment the tflint , save the file, push the code
8. Run the Jenkins pipeline to check the output of “tflint”

1. Create github repo



2. Clone on the vs code and Copy the lab2 content to the new repo folder on vs code



3. Git commands to push to the github

tf-adv-sca-feb26

main 1 Branch 0 Tags

Go to file

EC2 Default User configured for secerts manager 6ea38c3 · yesterday 17 Commits

.checkov/custom_policies	initial sca repo	yesterday
infra	configured for secerts manager	yesterday
.checkov.yml	enabled checkov	yesterday
.tflint.hcl	added var that is not used	yesterday
Jenkinsfile	configured for secerts manager	yesterday

README

4. Create an Jenkins pipeline (normal)

Jenkins / vishwa-job2 / Configure

Configure

- General
- Triggers
- Pipeline
- Advanced

General

Description

normal pipeline job

Plain text Preview

- ☐ Discard old builds ?
- ☐ Do not allow concurrent builds



Jenkins

/ vishwa-job2



/ Configure

Configure



General



Triggers



Pipeline



Advanced

☐ Trigger builds remotely (e.g., from scripts) ?

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script from SCM

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/Vishwanathms/tf-adv-sca-feb26.git

Use the https, to avoid creds

Pipeline

Advanced

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

+ Add Branch

Repository browser ?

(Auto)

Additional Behaviours

+ Add

Script Path ?

Jenkinsfile

☒ Lightweight checkout ?

[Pipeline Syntax](#)

Save

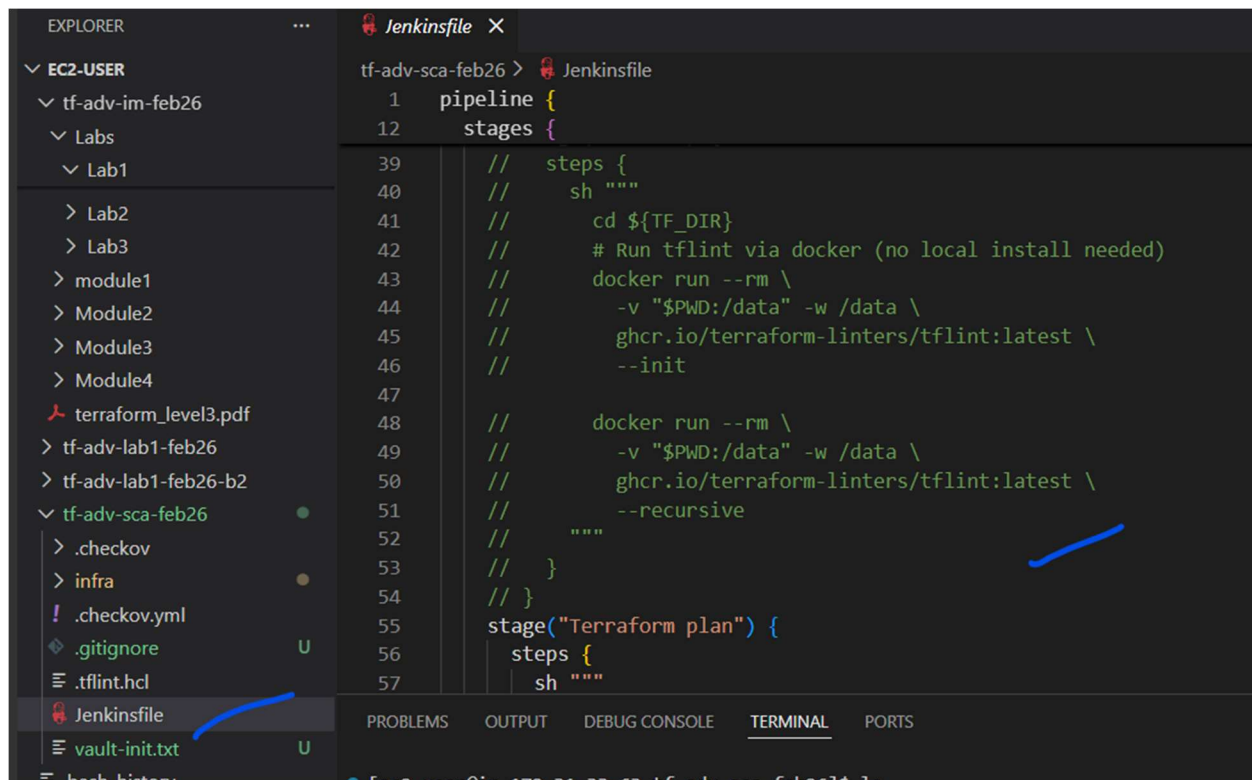
Apply

Change the branch to main and

Give the right jenkinsfile

5. Run the Jenkins pipeline

Note: for first time we will see the output of "Checkov", so the tflint is commented.



The screenshot shows a VS Code editor interface. On the left is the Explorer sidebar showing a project structure under 'EC2-USER'. The project includes folders like 'tf-adv-im-feb26', 'Labs', and 'Lab1'. A file named 'Jenkinsfile' is highlighted in the Explorer. The main editor area displays the content of the 'Jenkinsfile', which is a Jenkins pipeline script. The script defines a pipeline with a 'stages' block containing a 'Terraform plan' stage. This stage uses a Docker container to run 'tflint' for linting Terraform configurations. The script is as follows:

```
1 pipeline {
12  stages {
39    // steps {
40    //   sh """
41    //     cd ${TF_DIR}
42    //     # Run tflint via docker (no local install needed)
43    //     docker run --rm \
44    //       -v "$PWD:/data" -w /data \
45    //       ghcr.io/terraform-linters/tflint:latest \
46    //       --init
47
48    //     docker run --rm \
49    //       -v "$PWD:/data" -w /data \
50    //       ghcr.io/terraform-linters/tflint:latest \
51    //       --recursive
52    //   """
53    // }
54    // }
55    stage("Terraform plan") {
56      steps {
57        sh """
```

At the bottom of the editor, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is currently selected.

Once we run the job

**Jenkins**

/ vishwa-job2 / #15

Timestamps

[View as plain text](#)

- ☒ System clock time
☒ Use browser timezone
☐ Elapsed time
☐ None

```
[Pipeline] stage
[Pipeline] { (checkov)
[Pipeline] sh
00:50:07 + docker run --rm -v /home/ec2-user/workspace/vishwa-job2:/repo -w /repo bridgecrew/checkov:latest -d /repo/ir
/repo/.checkov/custom_policies
00:50:11 terraform scan results:
00:50:11
00:50:11 Passed checks: 8, Failed checks: 15, Skipped checks: 0
00:50:11
00:50:11 Check: CKV_AWS_23: "Ensure every security group and rule has a description"
00:50:11 FAILED for resource: aws_security_group.bad_sg
00:50:11 File: /main.tf:11-28
00:50:11 Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-p
00:50:11
00:50:11      11 | resource "aws_security_group" "bad_sg" {
00:50:11      12 |     name     = "lab11-open-sg"
00:50:11      13 |     description = "Open SG for lab"
00:50:11      14 |
00:50:11      15 |     ingress {
00:50:11      16 |         from_port = 22
00:50:11      17 |         to_port   = 22
00:50:11      18 |         protocol  = "tcp"
00:50:11      19 |         cidr_blocks = ["0.0.0.0/0"] # <-- should fail
00:50:11      20 |     }
00:50:11      21 |
00:50:11      22 |     egress {
00:50:11      23 |         from_port = 0
00:50:11      24 |         to_port   = 0
00:50:11      25 |         protocol  = "-1"
00:50:11      26 |         cidr_blocks = ["0.0.0.0/0"]
00:50:11      27 |     }
00:50:11      28 | }
00:50:11
00:50:11 Check: CKV_AWS_382: "Ensure no security groups allow egress from 0.0.0.0 to port -1"
00:50:11 FAILED for resource: aws_security_group.bad_sg
00:50:11 File: /main.tf:11-28
00:50:11 Guide: https://docs.prismacloud.io/en/enterprise-edition/policy-reference/aws-policies/aws-networking-p
```