Date: 7/4/2025

Infrastructure Scanning and Vault Security

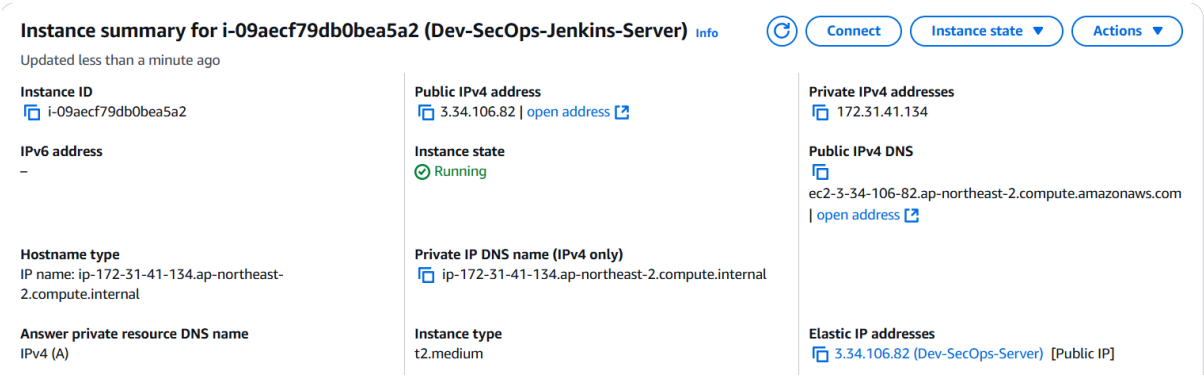
Objectives: To implement two pipelines –

1. Infrastructure scanning i.e scanning of terraform/IAC code files for vulnerabilities.
2. Incorporate Hashicorp Vault to secure server credentials which are to be retrieved only during deployment.

Target repositories:

1. https://github.com/Msocial123/EverNorth-Terraform-Project
2. https://github.com/Msocial123/fss-Retail-App_kubernetes

1.The repositories were forked and cloned. An EC2 server instance was set up on AWS.



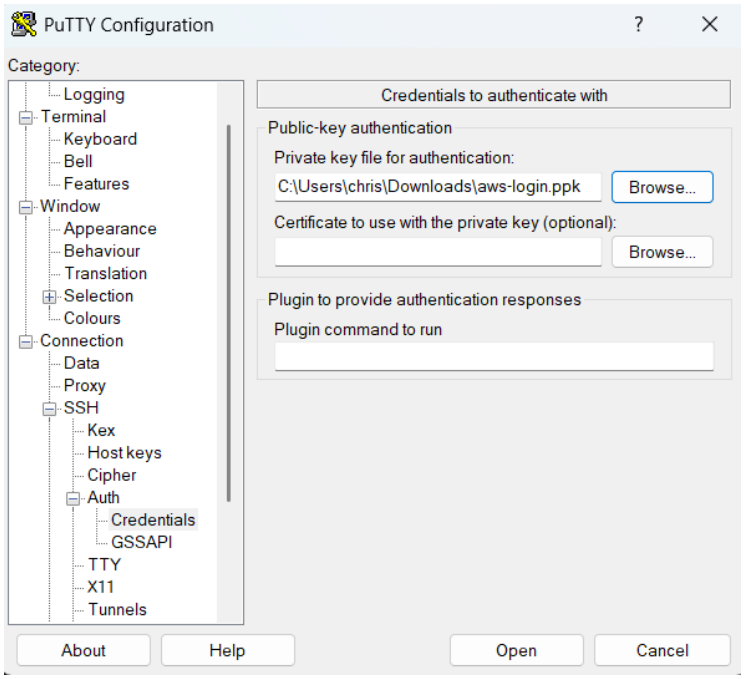
2.An SSH connection was established to the instance via PuTTy

-Download the ppk file from AWS

-Set host as <username>@<instance-ip>

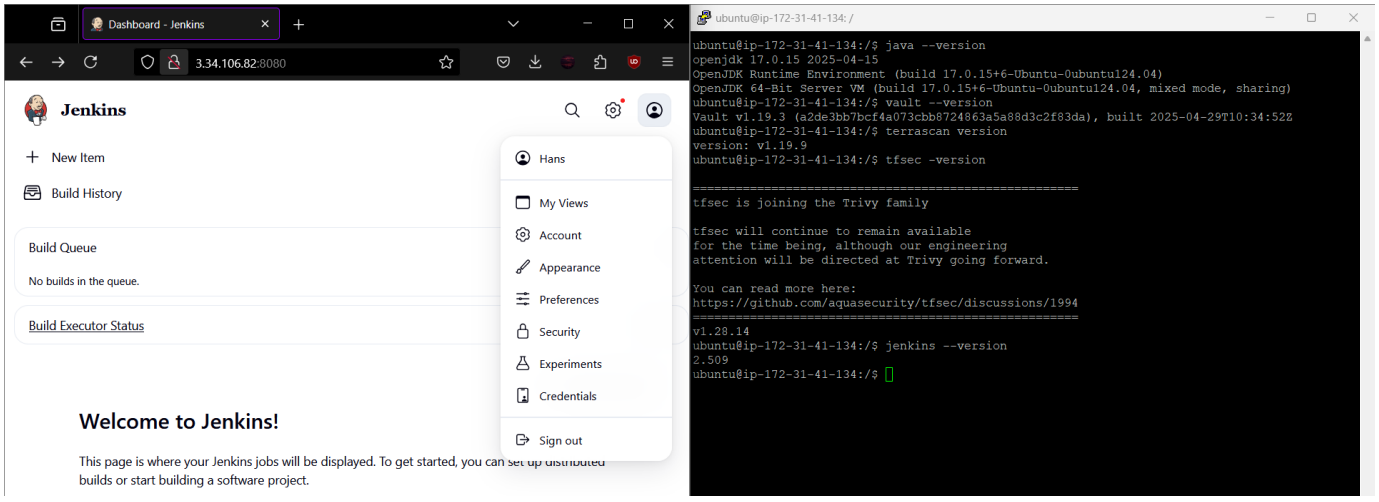
Date: 7/4/2025

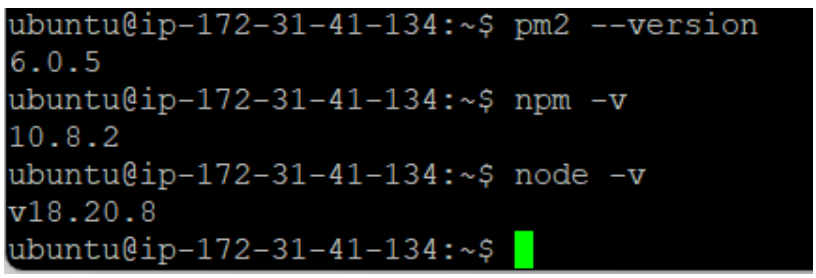
-Add the ppk file in the PuTTY configuration terminal under credentials



3.Downloaded the required tools and software on the server.

-Jenkins, Java, Vault, Terrascan, Tfsec, NodeJS, PM2





Date: 7/4/2025

4.Export vault credentials and set-up Vault server

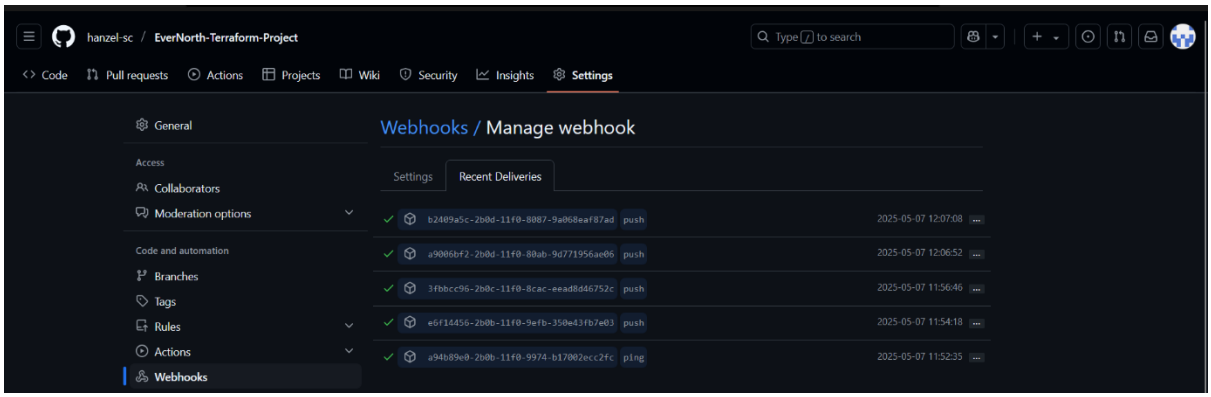
```
ubuntu@ip-172-31-41-134:/$ export VAULT_ADDR='http://127.0.0.1:8200'
export VAULT_TOKEN='hvs.U1udKY9MQEPsgkFu6XvGppGs'
ubuntu@ip-172-31-41-134:/$ vault status
Key             Value
---             -----
Seal Type       shamir
Initialized     true
Sealed          false
Total Shares    1
Threshold       1
Version         1.19.3
Build Date      2025-04-29T10:34:52Z
Storage Type    inmem
Cluster Name    vault-cluster-6320fcb5
Cluster ID      542997fb-b13d-395e-d24b-187915e0481b
HA Enabled      false
ubuntu@ip-172-31-41-134:/$
```

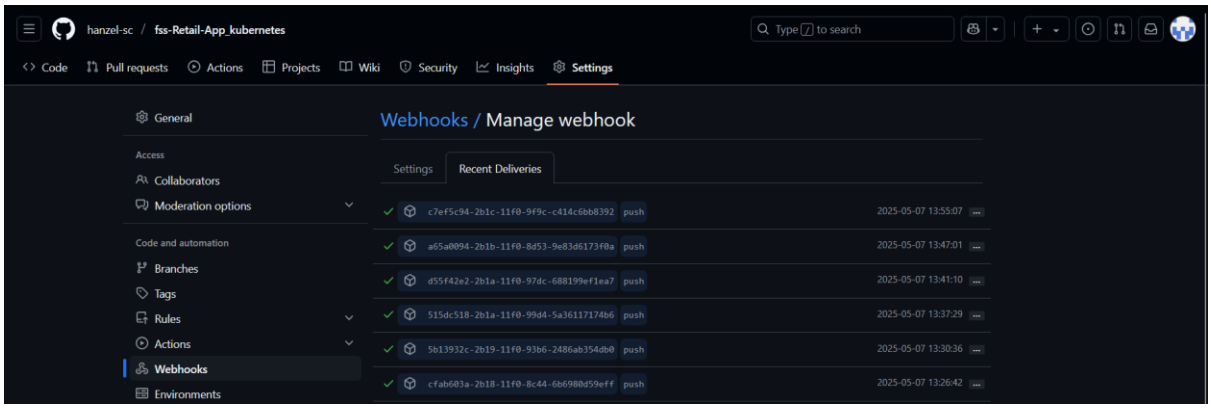
5. Store application server credentials in the vault

```
ubuntu@ip-172-31-41-134:~$ export VAULT_ADDR='http://127.0.0.1:8200'
export VAULT_TOKEN='hvs.U1udKY9MQEPsgkFu6XvGppGs'
ubuntu@ip-172-31-41-134:~$ vault kv put secret/fss-retail-app EMAIL_USER="chaga
ntyteja2502@gmail.com" EMAIL_PASS="yxoq bjuk rdnt alzp" PORT="3130" SESSION_SEC
RET="1234" MONGO_URI="mongodb://localhost:27017/myDatabase"
======= Secret Path =======
secret/data/fss-retail-app

======= Metadata =======
Key                Value
---                -----
created_time       2025-05-07T07:07:16.946078739Z
custom_metadata    <nil>
deletion_time      n/a
destroyed          false
version            1
ubuntu@ip-172-31-41-134:~$ vault kv get secret/fss-retail-app
======= Secret Path =======
secret/data/fss-retail-app

======= Metadata =======
Key                Value
---                -----
created_time       2025-05-07T07:07:16.946078739Z
custom_metadata    <nil>
deletion_time      n/a
destroyed          false
version            1

========= Data =========
Key              Value
---              -----
EMAIL_PASS       yxoq bjuk rdnt alzp
EMAIL_USER       chagantyteja2502@gmail.com
MONGO_URI        mongodb://localhost:27017/myDatabase
PORT             3130
SESSION_SECRET   1234
ubuntu@ip-172-31-41-134:~$
```

Date: 7/4/2025

6.Set up two GitHub webhooks – one for each repository

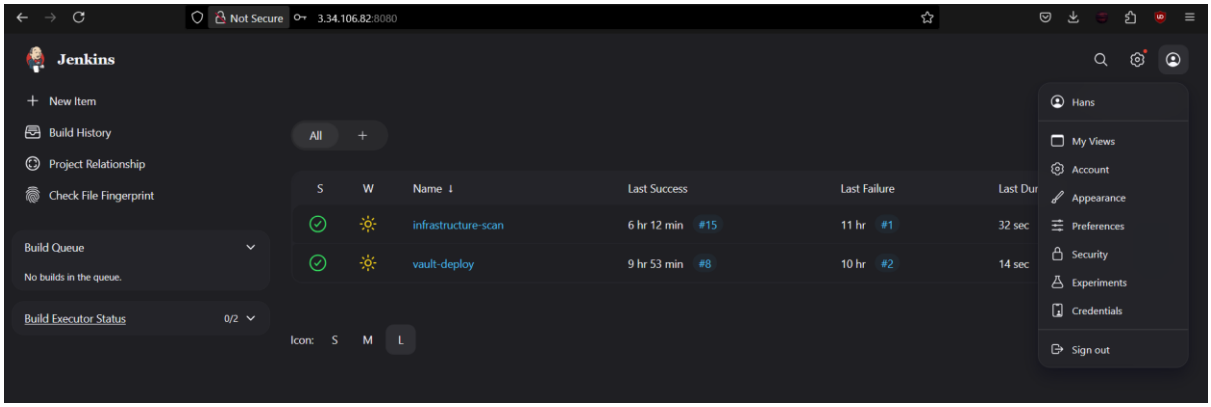




7. After installing the pre-requisites start the Jenkins Server using :

- sudo systemctl start jenkins

Configure Jenkins with the necessary installations and create two pipeline jobs, to be triggered by Git SCM polling



Date: 7/4/2025

8. Implement pipeline scripts on both repositories.

a. Infrastructure scanning pipeline

```
pipeline {
  agent any

  environment {
    TERRASCAN = '/usr/local/bin/terrascan'
    TFSEC = '/usr/local/bin/tfsec'
  }

  stages {
    stage('Checkout') {
      steps {
        git branch: 'test', url: 'https://github.com/hanzel-sc/EverNorth-Terraform-Project'
      }
    }

    stage('Run Pentest Scan') {
          steps {
            sh 'chmod +x scan.sh'
            sh './scan.sh'
          }
        }
    stage('Publish Reports') {
      steps {
        archiveArtifacts artifacts: 'reports/*.json', fingerprint: true
        archiveArtifacts artifacts: 'reports/*.txt', fingerprint: true
        archiveArtifacts artifacts: 'logfile/*.txt', fingerprint: true
```

```
        }
      }
    }
}
```

b. Vault security pipeline

```
pipeline {
    agent any

    environment {

        EMAIL_PASS=''
        EMAIL_USER=''
        PORT=''
        MONGODB_URI=''
        SESSION_SECRET=''

    }

    tools {

        nodejs "NodeJS"

    }

    stages {
```

```
    stage('Git checkout') {
      steps {
        git url: 'https://github.com/hanzel-sc/fss-Retail-App_kubernetes.git', branch: 'test'
      }
    }

    stage('Fetch Secrets from Vault') {
      steps {
        withVault(
          configuration: [vaultUrl: 'http://127.0.0.1:8200', vaultCredentialId: 'vault-token'],
          vaultSecrets: [[
            path: 'secret/fss-retail-app',
            secretValues: [
              [envVar: 'EMAIL_PASS', vaultKey: 'EMAIL_PASS'],
              [envVar: 'EMAIL_USER', vaultKey: 'EMAIL_USER'],
              [envVar: 'PORT', vaultKey: 'PORT'],
              [envVar: 'MONGODB_URI', vaultKey: 'MONGODB_URI'],
              [envVar: 'SESSION_SECRET', vaultKey: 'SESSION_SECRET']
            ]
          ]]
        ) {
          echo "Secrets retrieved successfully."
          echo "Email password is: ${env.EMAIL_PASS.take(2)}***"
        }
      }
    }
```

```
        stage('Install Dependencies') {
            steps {
                script {
                    sh 'npm install'
                }
            }
        }

        stage('Deploy') {
            steps {
                script {
                    sh 'pm2 delete all || true'
                    sh "pm2 start server.js --name nodejs-backend"
                    echo "Server is up and running"
                }
            }
        }
    }

    post {
        failure {
            echo 'Build failed! Check Jenkins logs.'
        }
        success {
            echo 'Successfully deployed NodeJS application with secrets from Vault!!'
        }
    }
}
```

Date: 7/4/2025

9. Implement a simple shell script to run the pentest scans (scan.sh file). Optionally, using grep and awk – log critical errors into log files for review.

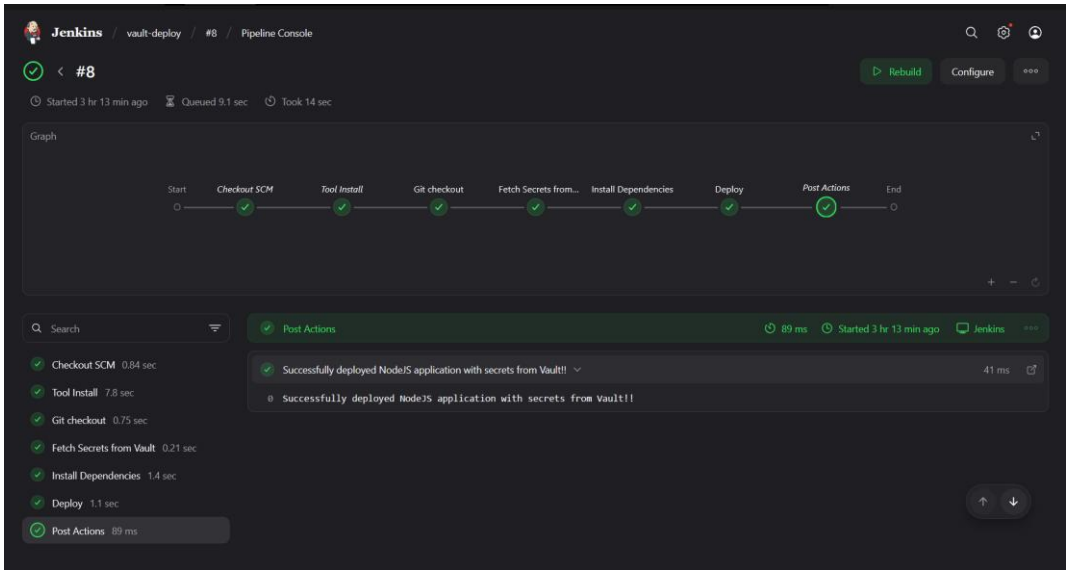
Jenkinsfile | $ scan.sh

EverNorth-Terraform-Project > $ scan.sh

```
#!/bin/bash

rm -rf reports
rm -rf logfile
mkdir -p reports
mkdir -p logfile

#timestamp=$(date +"%Y%m%d%H%M%S") //attaching timestamp if needed.

# Running the terrascan scan
/usr/local/bin/terrascan scan -t aws -d . -o json > reports/terrascan_report.json
/usr/local/bin/terrascan scan -t aws -d . -o human > reports/terrascan_report.txt

#Running the TFsec scan
/usr/local/bin/tfsec .  --format text > reports/tfsec_report.txt
/usr/local/bin/tfsec .  --format json > reports/tfsec_report.json

#Logging the critical and severe errors
awk '
  BEGIN { RS=""; FS="\n" }
  /Severity[[:space:]]*:[[:space:]]*HIGH/ { print "\n---\n" $0 }
' reports/terrascan_report.txt > logfile/terrascan_issues.txt

grep -i -E 'critical|high|severe' reports/tfsec_report.txt > logfile/tfsec_issues.txt


echo "Terrascan and tfsec reports saved to $(pwd)/reports/"
echo "Critical issues are logged in $(pwd)/logfile/"

```

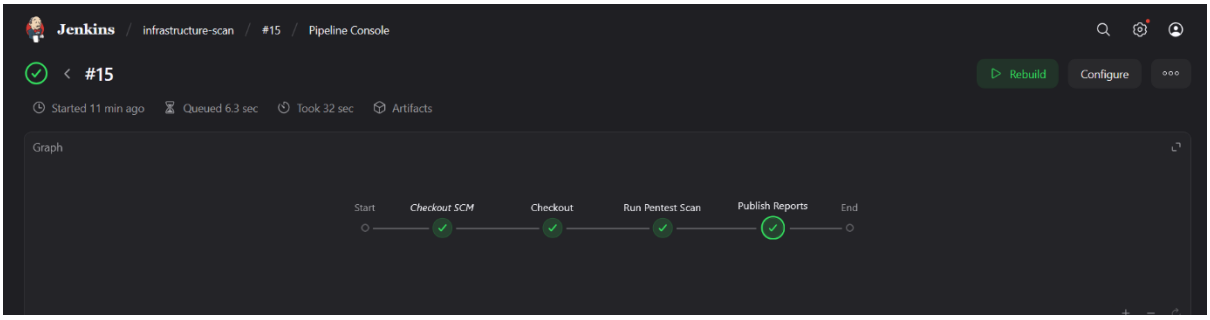
9. Push the code changes onto GitHub to trigger the pipelines. Ensure all stages of the pipeline are completed successfully.

- Vault Security pipeline with application deployment

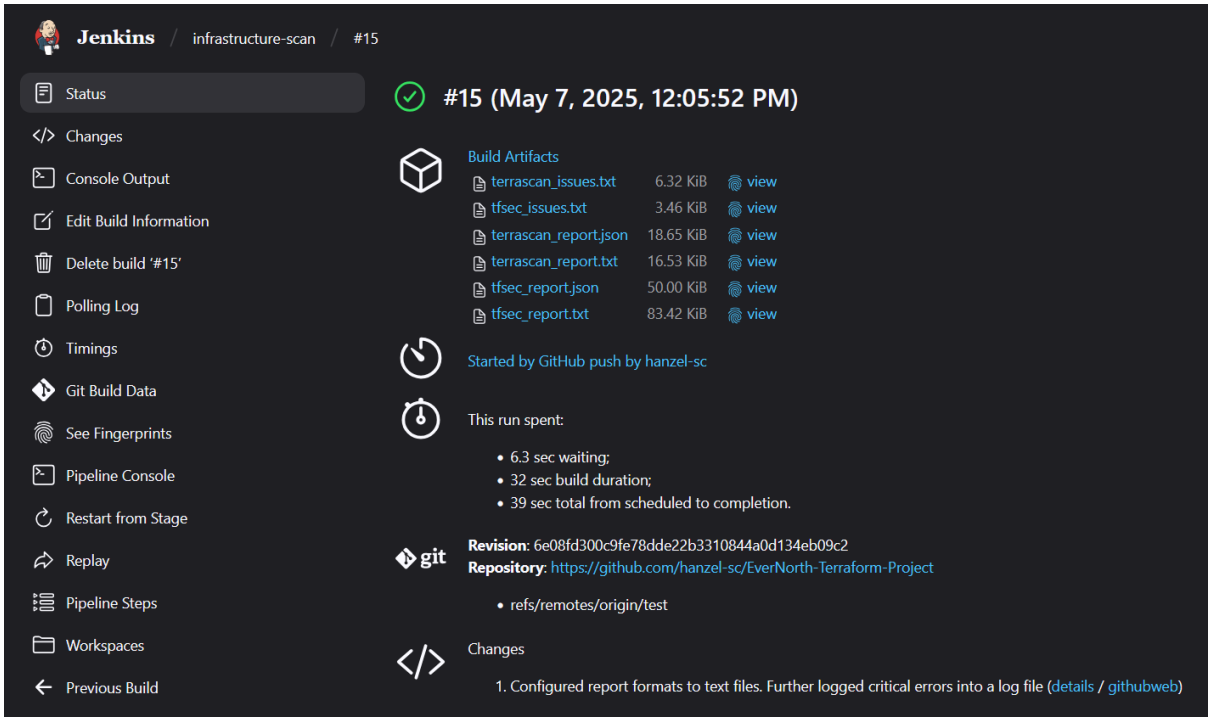


Date: 7/4/2025

- Infrastructure code scanning

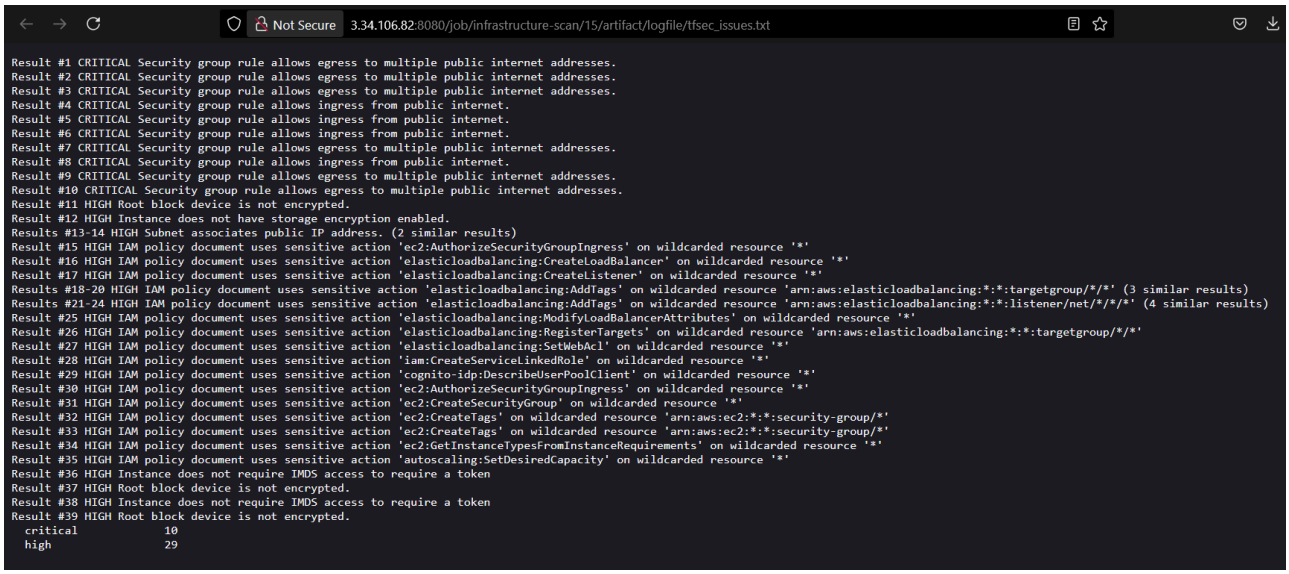


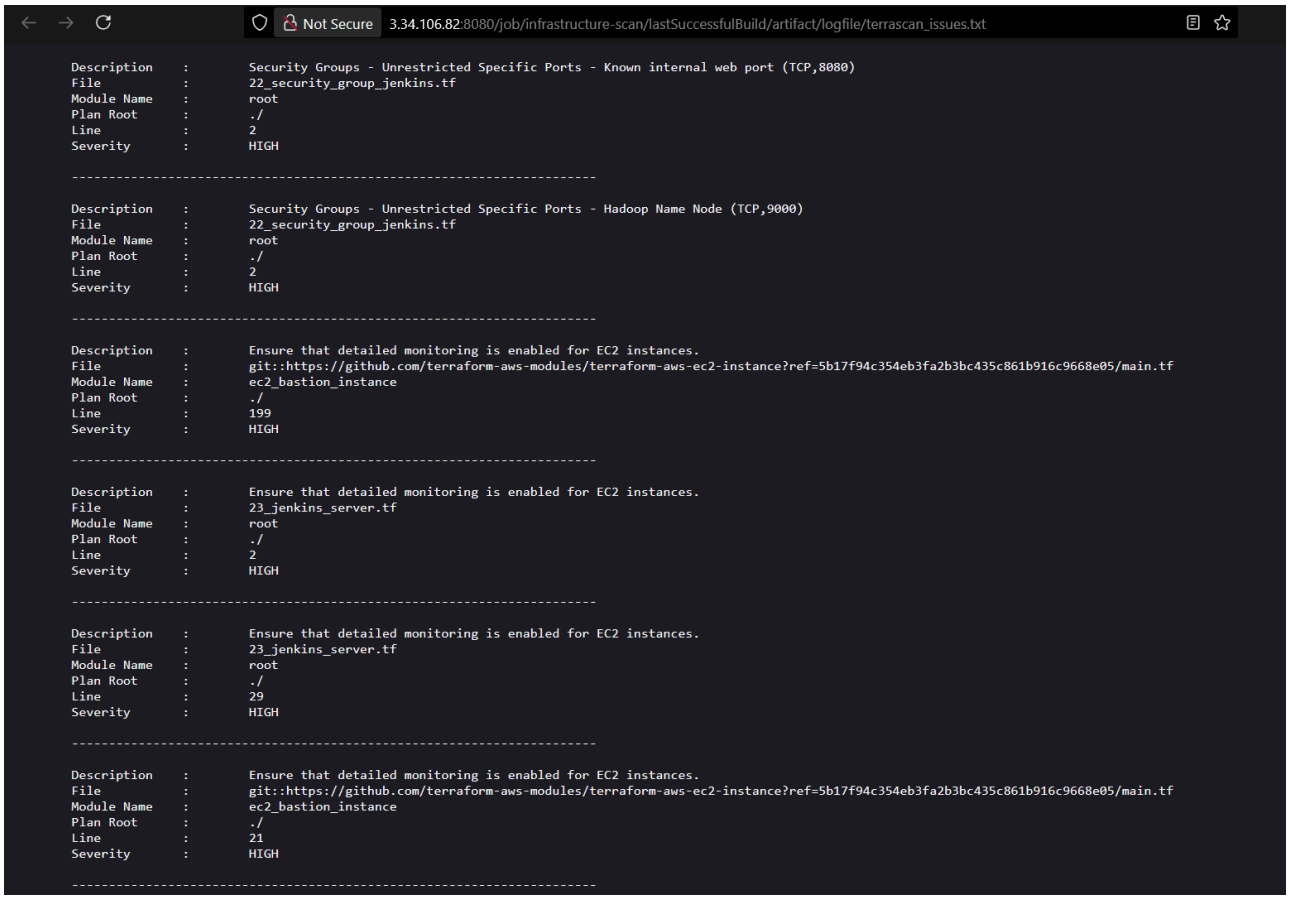
10. Observer the Build artifacts which contain the scan reports and logfiles containing the critical issues



Date: 7/4/2025

- Analyse the custom log files to review the critical issues





Links to the forked, local repositories:

1. https://github.com/hanzel-sc/EverNorth-Terraform-Project
2. https://github.com/hanzel-sc/fss-Retail-App_kubernetes