CA -3

DevOps Automation

Submitted by:

Nandana S

12317635

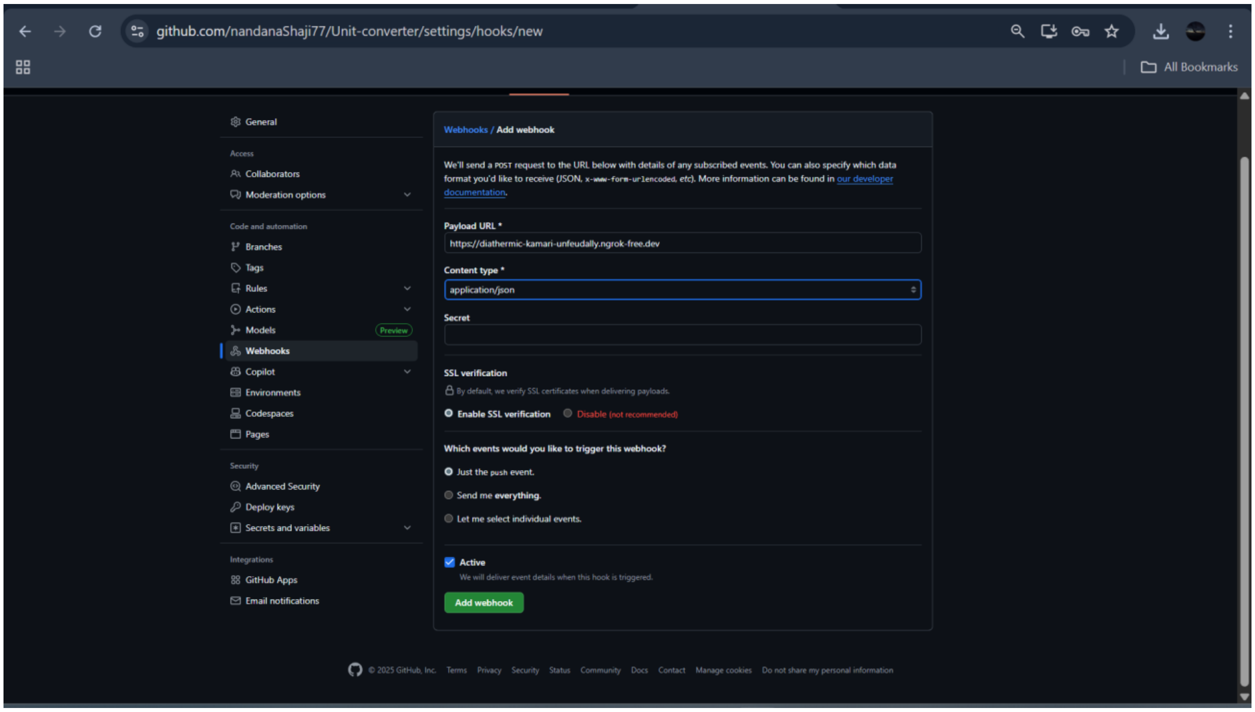
K23DV - A - 07

B-Tech CSE DevOps

GitHub Link: https://github.com/nandanaShaji77/CA-3-DevOps-Automation.git

LOVELY PROFESSIONAL UNIVERSITY

Task 1: Automated Triggering via ngrok



Webhooks / Add webhook

We'll send a `POST` request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

Payload URL *
https://diathermic-kamari-unfeudally.ngrok-free.dev

Content type *
application/json

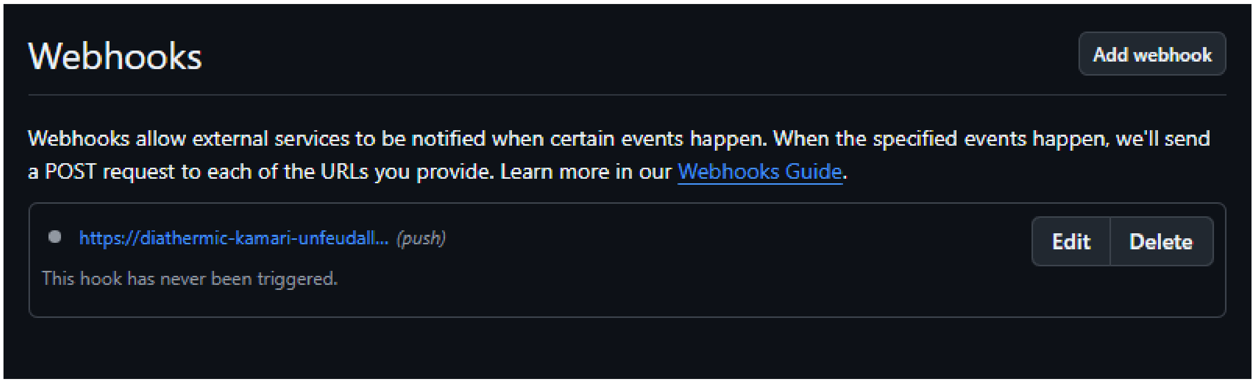
Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
● Enable SSL verification ○ Disable (not recommended)

Which events would you like to trigger this webhook?
● Just the `push` event.
○ Send me **everything**.
○ Let me select individual events.

☑ Active
We will deliver event details when this hook is triggered.

Add webhook


Webhooks

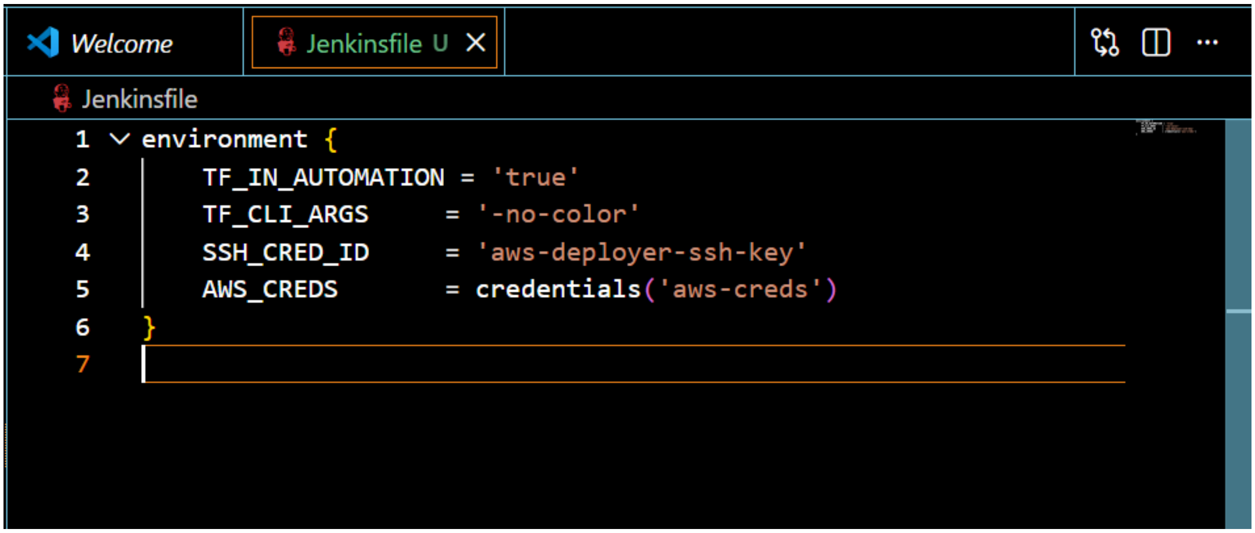
Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

https://diathermic-kamari-unfeudall... *(push)*
This hook has never been triggered.

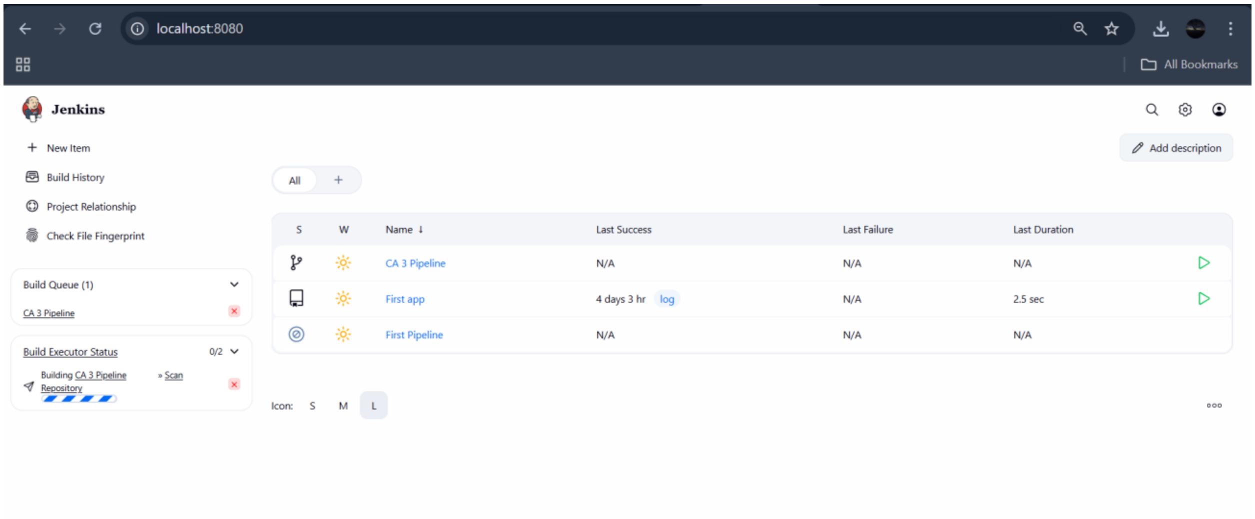
Webhook successfully created with the help of ngrok

“*ngrok http 8080*”

Task 2: Pipeline Environment & Credentials



```
environment {
    TF_IN_AUTOMATION = 'true'
    TF_CLI_ARGS     = '-no-color'
    SSH_CRED_ID     = 'aws-deployer-ssh-key'
    AWS_CREDS       = credentials('aws-creds')
}
```

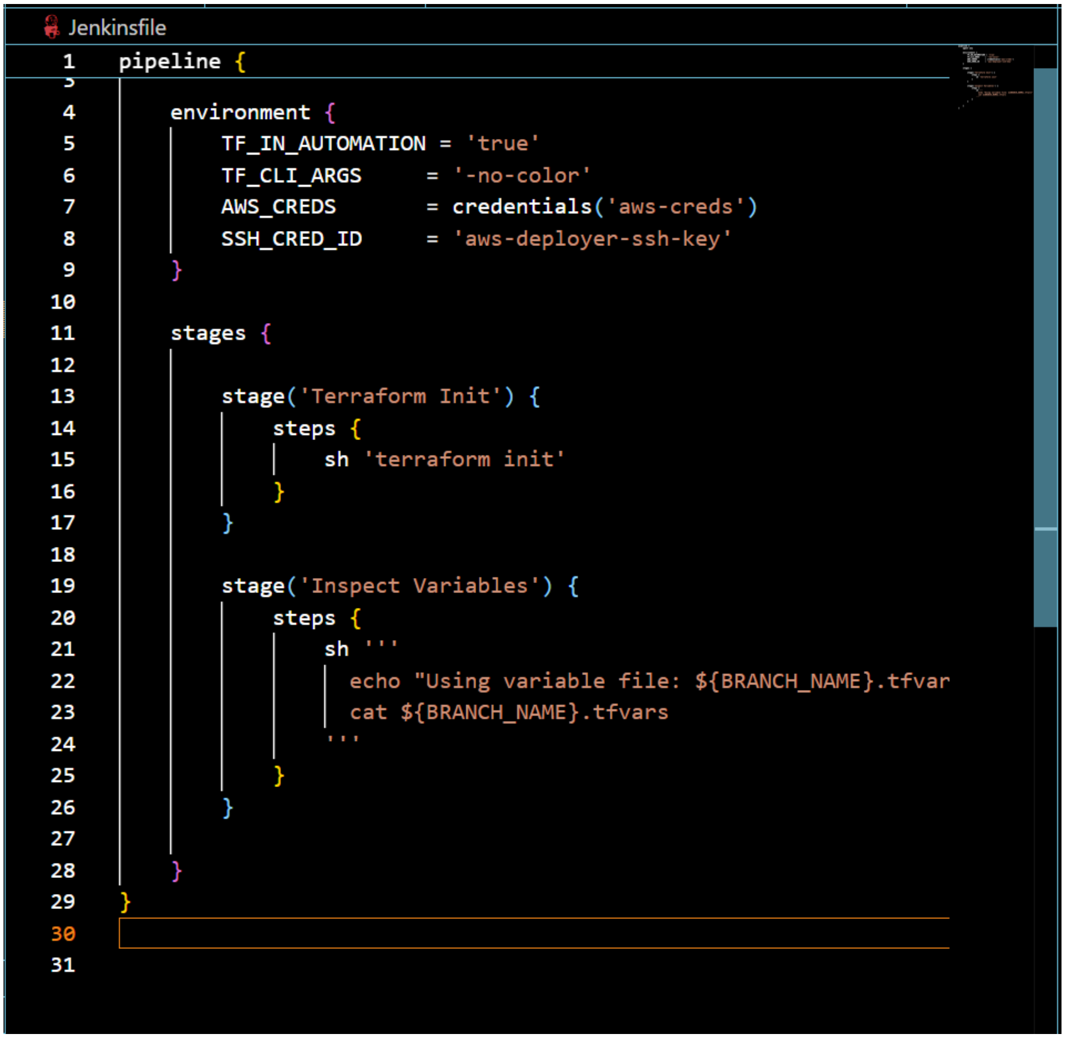


S	W	Name ↓	Last Success	Last Failure	Last Duration
	☀	CA 3 Pipeline	N/A	N/A	N/A
	☀	First app	4 days 3 hr log	N/A	2.5 sec
	☀	First Pipeline	N/A	N/A	N/A

Jenkins running on '*http://localhost:8080/*'

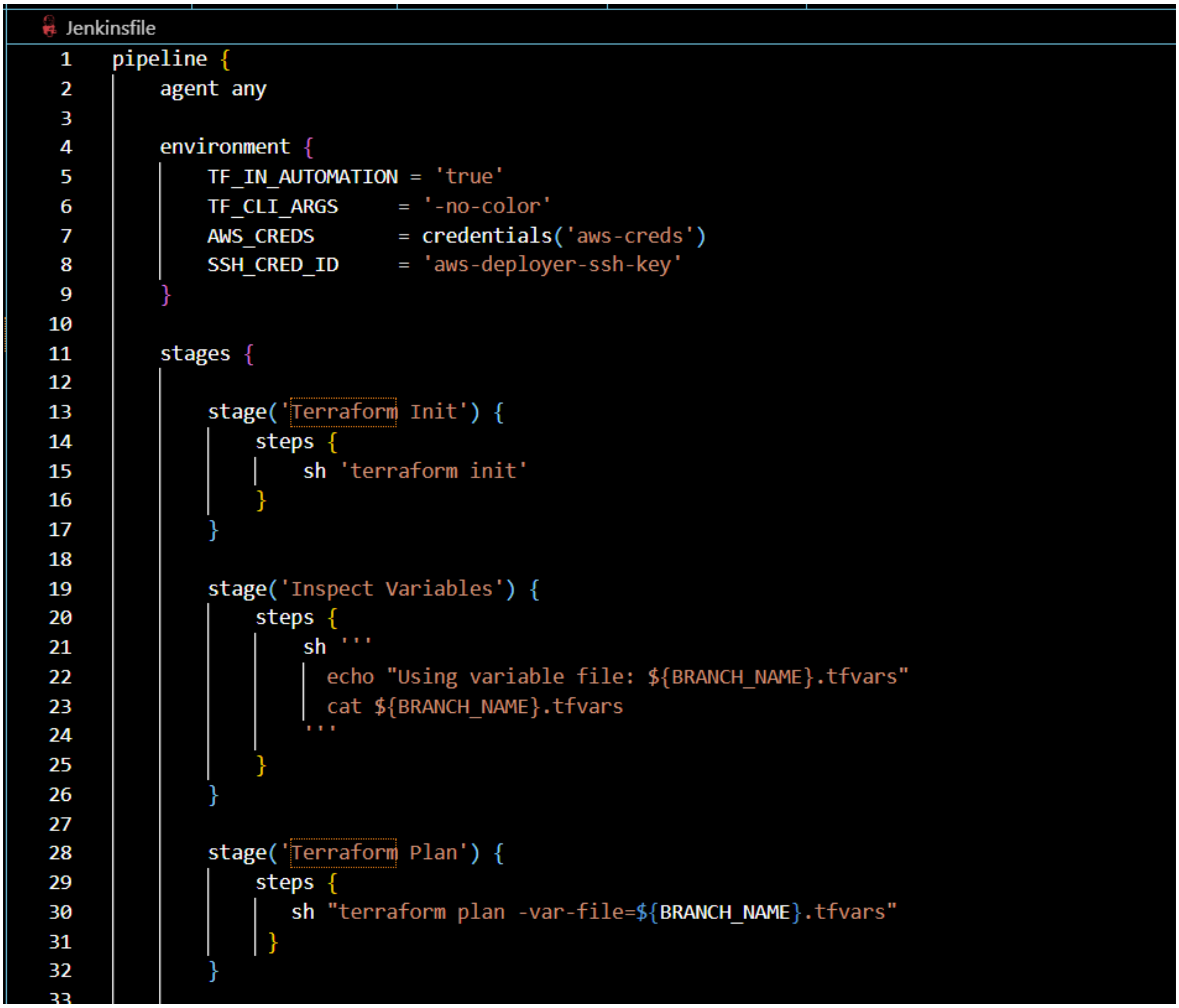
'CA 3 Pipeline' is the new multibranch pipeline created for the BYOD

Task 3: Initialization and variable inspection



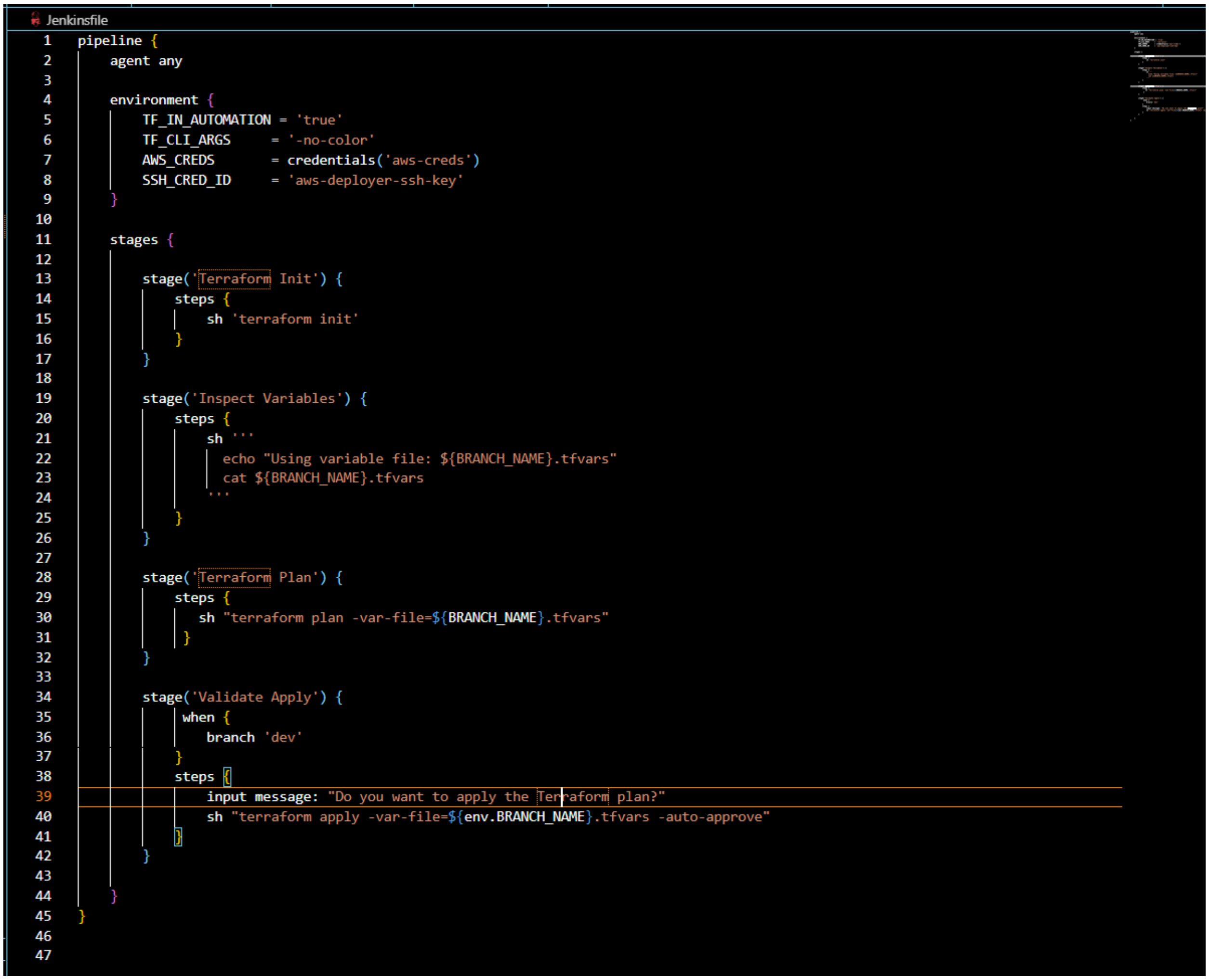
```
pipeline {
    environment {
        TF_IN_AUTOMATION = 'true'
        TF_CLI_ARGS     = '-no-color'
        AWS_CREDS       = credentials('aws-creds')
        SSH_CRED_ID     = 'aws-deployer-ssh-key'
    }

    stages {

        stage('Terraform Init') {
            steps {
                sh 'terraform init'
            }
        }

        stage('Inspect Variables') {
            steps {
                sh '''
                  echo "Using variable file: ${BRANCH_NAME}.tfvar
                  cat ${BRANCH_NAME}.tfvars
                '''
            }
        }

    }
}
```

Task 4: Branch-Specific Terraform Planning



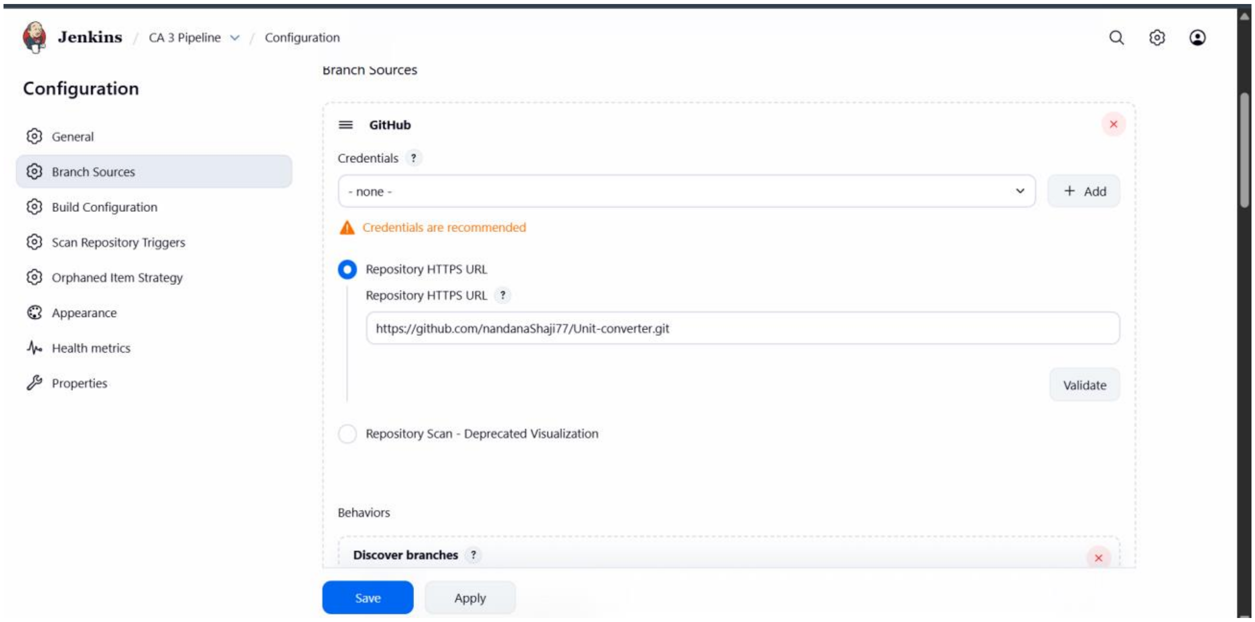
```
pipeline {
    agent any

    environment {
        TF_IN_AUTOMATION = 'true'
        TF_CLI_ARGS     = '-no-color'
        AWS_CREDS       = credentials('aws-creds')
        SSH_CRED_ID     = 'aws-deployer-ssh-key'
    }

    stages {

        stage('Terraform Init') {
            steps {
                sh 'terraform init'
            }
        }

        stage('Inspect Variables') {
            steps {
                sh '''
                  echo "Using variable file: ${BRANCH_NAME}.tfvars"
                  cat ${BRANCH_NAME}.tfvars
                '''
            }
        }

        stage('Terraform Plan') {
            steps {
               sh "terraform plan -var-file=${BRANCH_NAME}.tfvars"
             }
        }
```

Task 5: Conditional Manual Approval Gate



Jenkinsfile

```
pipeline {
    agent any

    environment {
        TF_IN_AUTOMATION = 'true'
        TF_CLI_ARGS      = '-no-color'
        AWS_CREDS        = credentials('aws-creds')
        SSH_CRED_ID      = 'aws-deployer-ssh-key'
    }

    stages {

        stage('Terraform Init') {
            steps {
                sh 'terraform init'
            }
        }

        stage('Inspect Variables') {
            steps {
                sh '''
                  echo "Using variable file: ${BRANCH_NAME}.tfvars"
                  cat ${BRANCH_NAME}.tfvars
                '''
            }
        }

        stage('Terraform Plan') {
            steps {
               sh "terraform plan -var-file=${BRANCH_NAME}.tfvars"
             }
        }

        stage('Validate Apply') {
             when {
                branch 'dev'
            }
            steps {
                input message: "Do you want to apply the Terraform plan?"
                sh "terraform apply -var-file=${env.BRANCH_NAME}.tfvars -auto-approve"
            }
        }

    }
}
```



Branch Sources

GitHub

Credentials ?

- none -

Credentials are recommended

● Repository HTTPS URL

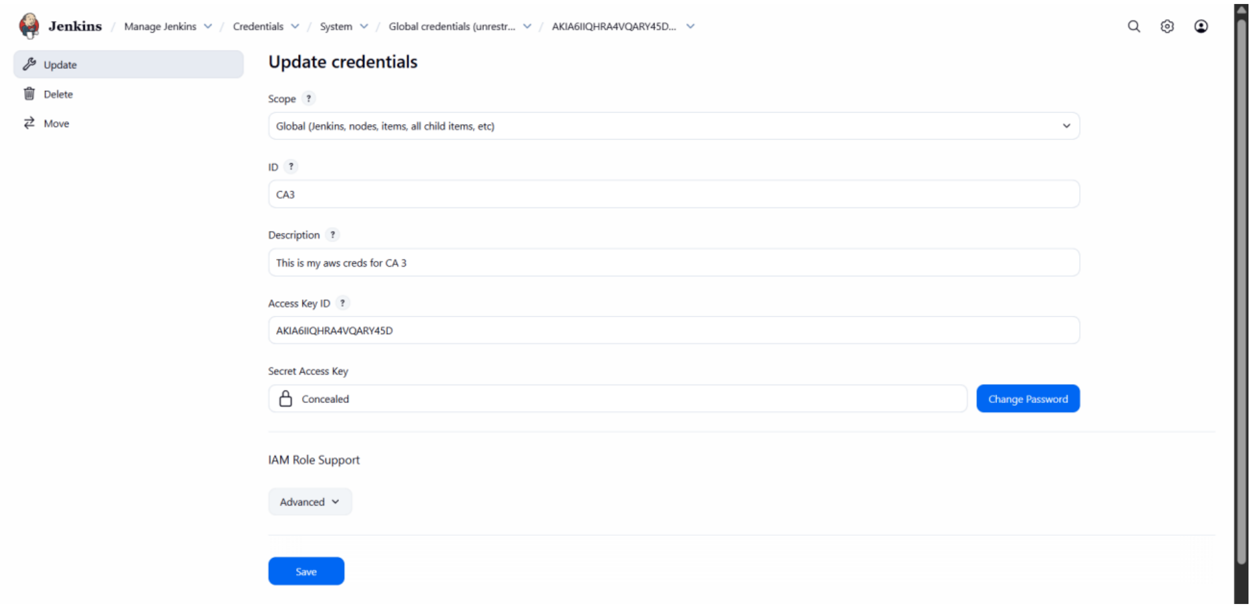
Repository HTTPS URL ?

https://github.com/nandanaShaji77/Unit-converter.git

○ Repository Scan - Deprecated Visualization

Behaviors

Discover branches ?



Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr... / AKIA6IIQHRA4VQARY45D...
Update
Delete
Move
Change Password
Advanced
Save

Update credentials

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

CA3

Description ?

This is my aws creds for CA 3

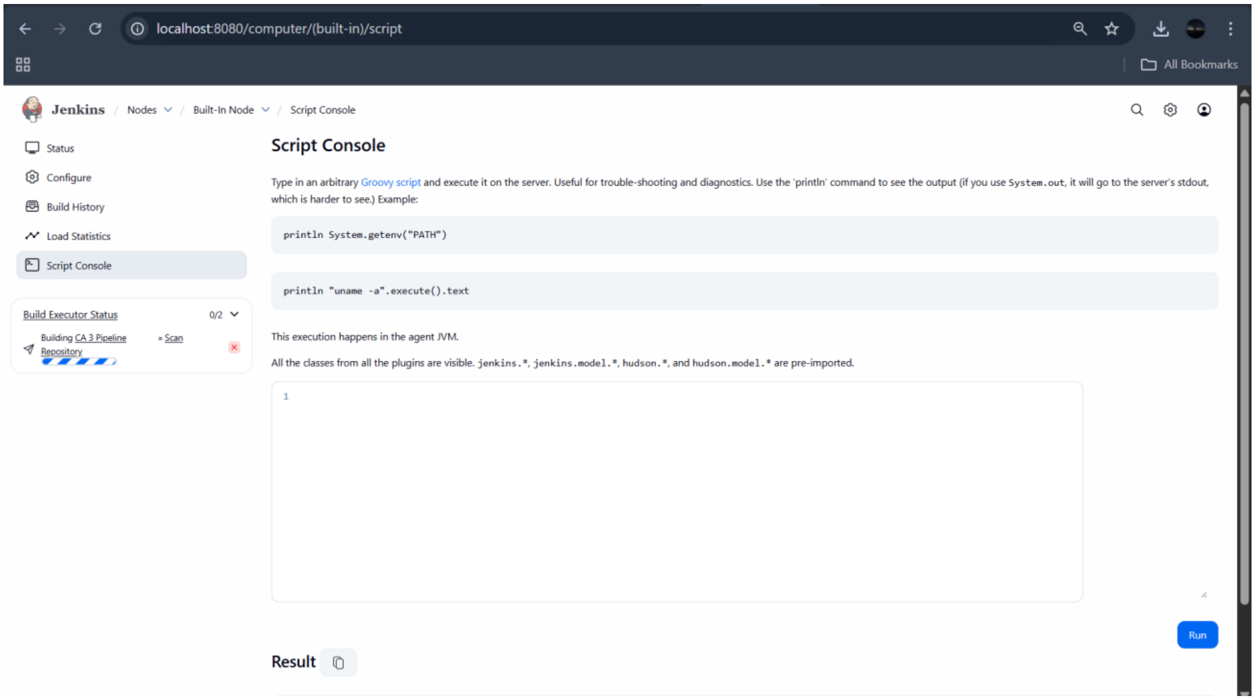
Access Key ID ?

AKIA6IIQHRA4VQARY45D

Secret Access Key

Concealed

IAM Role Support


localhost:8080/computer/(built-in)/script
All Bookmarks
Jenkins / Nodes / Built-In Node / Script Console
Status
Configure
Build History
Load Statistics
Script Console
Build Executor Status 0/2
Building CA 3 Pipeline » Scan Repository
Run
Result

Script Console

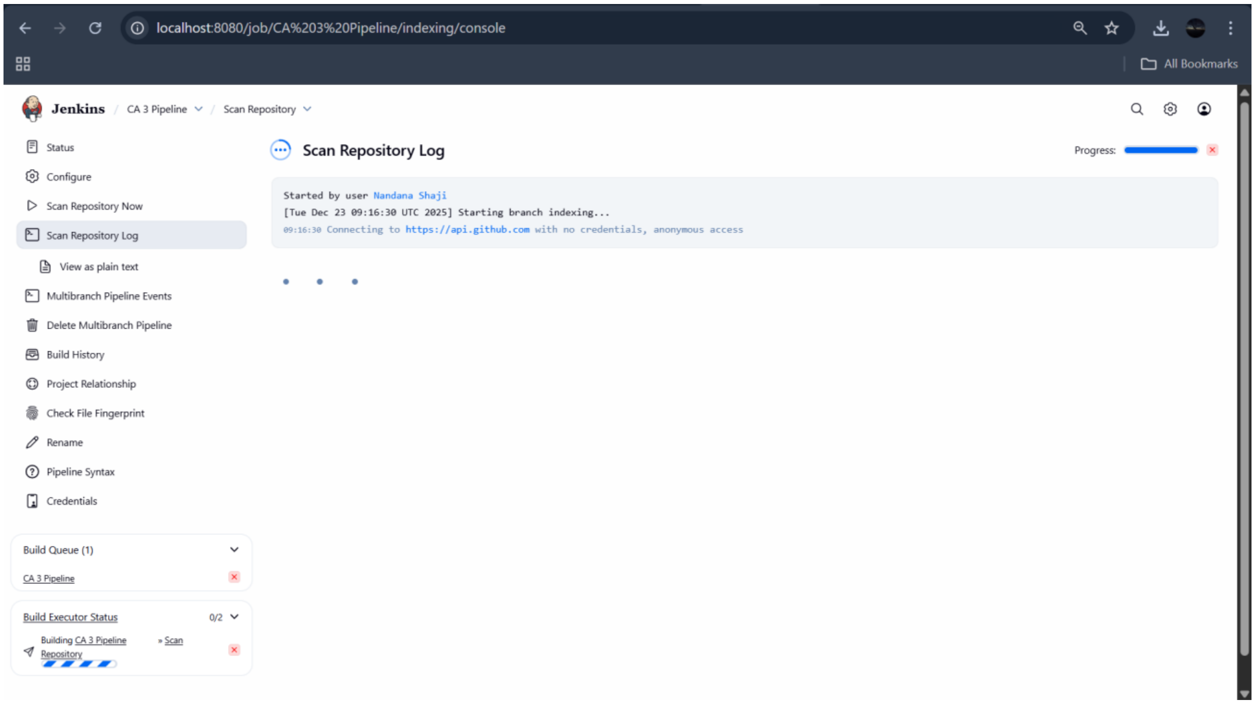
Type in an arbitrary Groovy script and execute it on the server. Useful for trouble-shooting and diagnostics. Use the 'println' command to see the output (if you use `System.out`, it will go to the server's stdout, which is harder to see.) Example:

```
println System.getenv("PATH")
```

```
println "uname -a".execute().text
```

This execution happens in the agent JVM.

All the classes from all the plugins are visible. `jenkins.*`, `jenkins.model.*`, `hudson.*`, and `hudson.model.*` are pre-imported.


localhost:8080/job/CA%203%20Pipeline/indexing/console
All Bookmarks
Jenkins / CA 3 Pipeline / Scan Repository
Status
Configure
Scan Repository Now
Scan Repository Log
View as plain text
Multibranch Pipeline Events
Delete Multibranch Pipeline
Build History
Project Relationship
Check File Fingerprint
Rename
Pipeline Syntax
Credentials
Build Queue (1)
CA 3 Pipeline
Build Executor Status 0/2
Building CA 3 Pipeline » Scan Repository
Progress:

Scan Repository Log

```
Started by user Nandana Shaji
[Tue Dec 23 09:16:30 UTC 2025] Starting branch indexing...
09:16:30 Connecting to https://api.github.com with no credentials, anonymous access
```