

# 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>*



**L**OVELY  
**P**ROFESSIONAL  
**U**NIVERSITY

## Task 1: Automated Triggering via ngrok

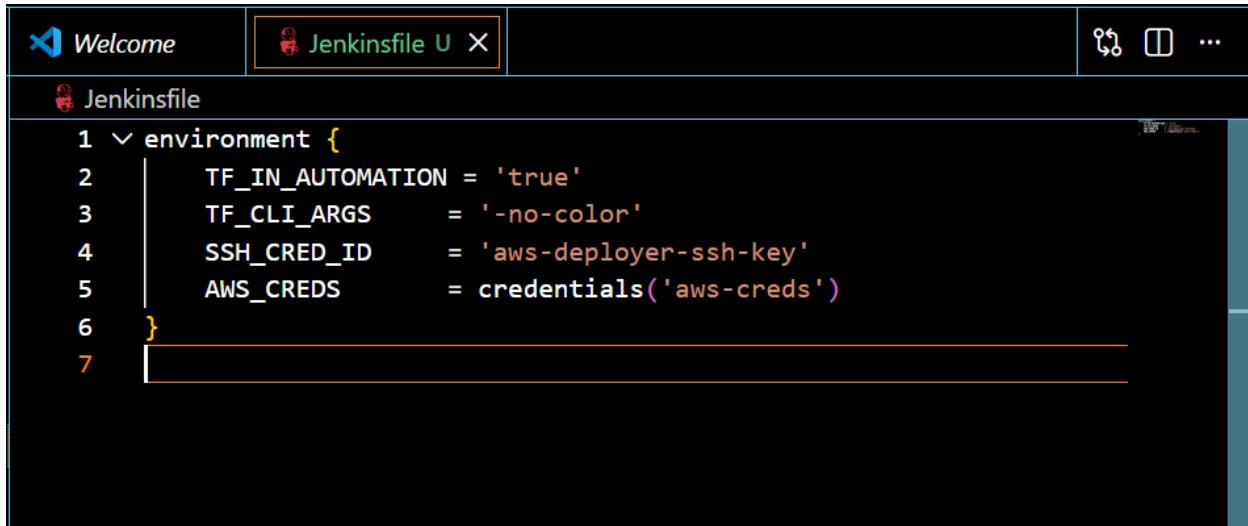
The screenshot shows the GitHub settings interface for a repository. On the left, there's a sidebar with various options like General, Access, Code and automation, Security, and Integrations. The 'Webhooks' section is selected. On the right, a modal window titled 'Webhooks / Add webhook' is open. It contains fields for 'Payload URL' (set to 'http://diathermic-kamari-unfeudally.ngrok-free.dev'), 'Content type' (set to 'application/json'), and a 'Secret' field. Under 'SSL verification', the 'Enable SSL verification' option is selected. In the 'Which events would you like to trigger this webhook?' section, the 'Just the push event.' radio button is selected. The 'Active' checkbox is checked. At the bottom of the modal is a green 'Add webhook' button.

The screenshot shows the 'Webhooks' page on GitHub. The title 'Webhooks' is at the top, along with an 'Add webhook' button. Below it, a paragraph explains what webhooks are and points to a 'Webhooks Guide'. A list of existing webhooks is shown, with one entry highlighted: 'https://diathermic-kamari-unfeudall... (push)'. To the right of this entry are 'Edit' and 'Delete' buttons. A note below the list says 'This hook has never been triggered.'

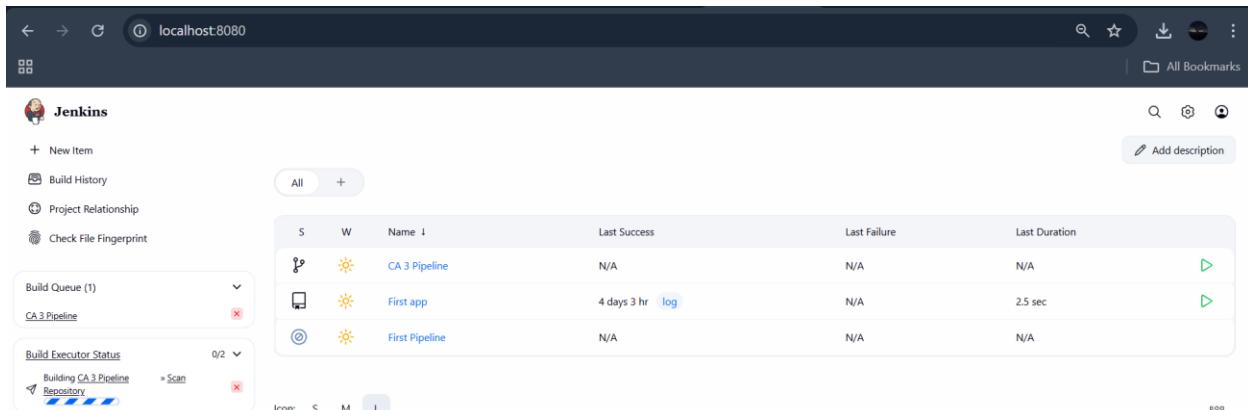
Webhook successfully created with the help of ngrok

*“ngrok http 8080”*

## Task 2: Pipeline Environment & Credentials



```
>Welcome Jenkinsfile U ...
Jenkinsfile
1 ✓ environment {
2     TF_IN_AUTOMATION = 'true'
3     TF_CLI_ARGS      = '-no-color'
4     SSH_CRED_ID      = 'aws-deployer-ssh-key'
5     AWS_CREDS         = credentials('aws-creds')
6 }
7
```



The screenshot shows the Jenkins dashboard at [localhost:8080](http://localhost:8080). The main area displays a table of pipelines:

S	W	Name	Last Success	Last Failure	Last Duration
⌚	☀️	CA 3 Pipeline	N/A	N/A	N/A
⌚	☀️	First app	4 days 3 hr <a href="#">log</a>	N/A	2.5 sec
⌚	☀️	First Pipeline	N/A	N/A	N/A

Below the table, there are sections for 'Build Queue (1)', 'Build Executor Status', and 'Check File Fingerprint'. A sidebar on the left includes links for 'New Item', 'Build History', 'Project Relationship', and 'Check File Fingerprint'.

Jenkins running on '<http://localhost:8080/>'

**‘CA 3 Pipeline’** is the new multibranch pipeline created for the BYOD

## Task 3: Initialization and variable inspection

```
 Jenkinsfile
 1 pipeline {
 2     environment {
 3         TF_IN_AUTOMATION = 'true'
 4         TF_CLI_ARGS      = '-no-color'
 5         AWS_CREDS        = credentials('aws-creds')
 6         SSH_CRED_ID      = 'aws-deployer-ssh-key'
 7     }
 8
 9 }
10
11 stages {
12
13     stage('Terraform Init') {
14         steps {
15             sh 'terraform init'
16         }
17     }
18
19     stage('Inspect Variables') {
20         steps {
21             sh '''
22                 echo "Using variable file: ${BRANCH_NAME}.tfvar"
23                 cat ${BRANCH_NAME}.tfvars
24             '''
25         }
26     }
27
28 }
29
30
31 }
```

## Task 4: Branch-Specific Terraform Planning

```
Jenkinsfile
1 pipeline {
2     agent any
3
4     environment {
5         TF_IN_AUTOMATION = 'true'
6         TF_CLI_ARGS      = '-no-color'
7         AWS_CREDS        = credentials('aws-creds')
8         SSH_CRED_ID      = 'aws-deployer-ssh-key'
9     }
10
11    stages {
12
13        stage('Terraform Init') {
14            steps {
15                sh 'terraform init'
16            }
17        }
18
19        stage('Inspect Variables') {
20            steps {
21                sh '''
22                    echo "Using variable file: ${BRANCH_NAME}.tfvars"
23                    cat ${BRANCH_NAME}.tfvars
24                '''
25            }
26        }
27
28        stage('Terraform Plan') {
29            steps {
30                sh "terraform plan -var-file=${BRANCH_NAME}.tfvars"
31            }
32        }
33    }
}
```

## Task 5: Conditional Manual Approval Gate

```
④ Jenkinsfile
 1 pipeline {
 2     agent any
 3
 4     environment {
 5         TF_IN_AUTOMATION = 'true'
 6         TF_CLI_ARGS      = '-no-color'
 7         AWS_CREDS        = credentials('aws-creds')
 8         SSH_CRED_ID      = 'aws-deployer-ssh-key'
 9     }
10
11     stages {
12         stage('Terraform Init') {
13             steps {
14                 sh 'terraform init'
15             }
16         }
17
18         stage('Inspect Variables') {
19             steps {
20                 sh ...
21                 echo "Using variable file: ${BRANCH_NAME}.tfvars"
22                 cat ${BRANCH_NAME}.tfvars
23                 ...
24             }
25         }
26
27         stage('Terraform Plan') {
28             steps {
29                 sh "terraform plan -var-file=${BRANCH_NAME}.tfvars"
30             }
31         }
32
33         stage('Validate Apply') {
34             when {
35                 branch 'dev'
36             }
37             steps [
38                 input message: "Do you want to apply the Terraform plan?"
39                 sh "terraform apply -var-file=${env.BRANCH_NAME}.tfvars -auto-approve"
40             ]
41         }
42
43     }
44 }
45
46
47 }
```

The screenshot shows the Jenkins Pipeline Configuration page for the 'CA 3 Pipeline'. The left sidebar lists configuration sections: General, Branch Sources (selected), Build Configuration, Scan Repository Triggers, Orphaned Item Strategy, Appearance, Health metrics, and Properties. The main area is titled 'Branch Sources' and contains a 'GitHub' section. It shows a dropdown for 'Credentials' set to 'none', a note that 'Credentials are recommended', and a selected radio button for 'Repository HTTPS URL' with the value 'https://github.com/nandanaShaji77/Unit-converter.git'. There is also an option for 'Repository Scan - Deprecated Visualization'. At the bottom are 'Save' and 'Apply' buttons.

The screenshot shows the Jenkins 'Update credentials' page under 'Manage Jenkins > Credentials > Global credentials (unrestricted) > AKIA6I0QHRA4VQARY45D...'. The left sidebar has options for Update, Delete, and Move. The main form is titled 'Update credentials' and contains fields for 'Scope' (set to 'Global (Jenkins, nodes, items, all child items, etc)'), 'ID' (set to 'CA3'), 'Description' ('This is my aws creds for CA 3'), 'Access Key ID' (set to 'AKIA6I0QHRA4VQARY45D'), and 'Secret Access Key' (set to 'Concealed'). A 'Change Password' button is visible next to the secret key field. At the bottom are 'Advanced' and 'Save' buttons.

The screenshot shows the Jenkins Script Console interface at [localhost:8080/computer/\(built-in\)/script](http://localhost:8080/computer/(built-in)/script). The left sidebar includes links for Status, Configure, Build History, Load Statistics, and Script Console (which is selected). The main area is titled "Script Console" and contains a text input field with the following Groovy code:

```
println System.getenv("PATH")
println "uname -a".execute().text
```

A note below the code states: "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." A large text area shows the output of the script, which is just the number "1". At the bottom right is a blue "Run" button.

The screenshot shows the Jenkins Scan Repository Log interface at [localhost:8080/job/CA%203%20Pipeline/indexing/console](http://localhost:8080/job/CA%203%20Pipeline/indexing/console). The left sidebar includes links for Status, Configure, Scan Repository Now (which is selected), Scan Repository Log (selected), View as plain text, Multibranch Pipeline Events, Delete Multibranch Pipeline, Build History, Project Relationship, Check File Fingerprint, Rename, Pipeline Syntax, and Credentials. The main area is titled "Scan Repository Log" and displays log output for a GitHub repository scan. It shows the log started by user Nandana Shaji on Tuesday, December 23, 2025, at 09:16:30 UTC. The log indicates it is connecting to <https://api.github.com> with no credentials, anonymous access. A progress bar at the top right shows the progress of the scan. Below the log, there is a "Build Queue (1)" section showing "CA 3 Pipeline" and a "Build Executor Status" section showing "Building CA 3 Pipeline" with a "Scan" button.