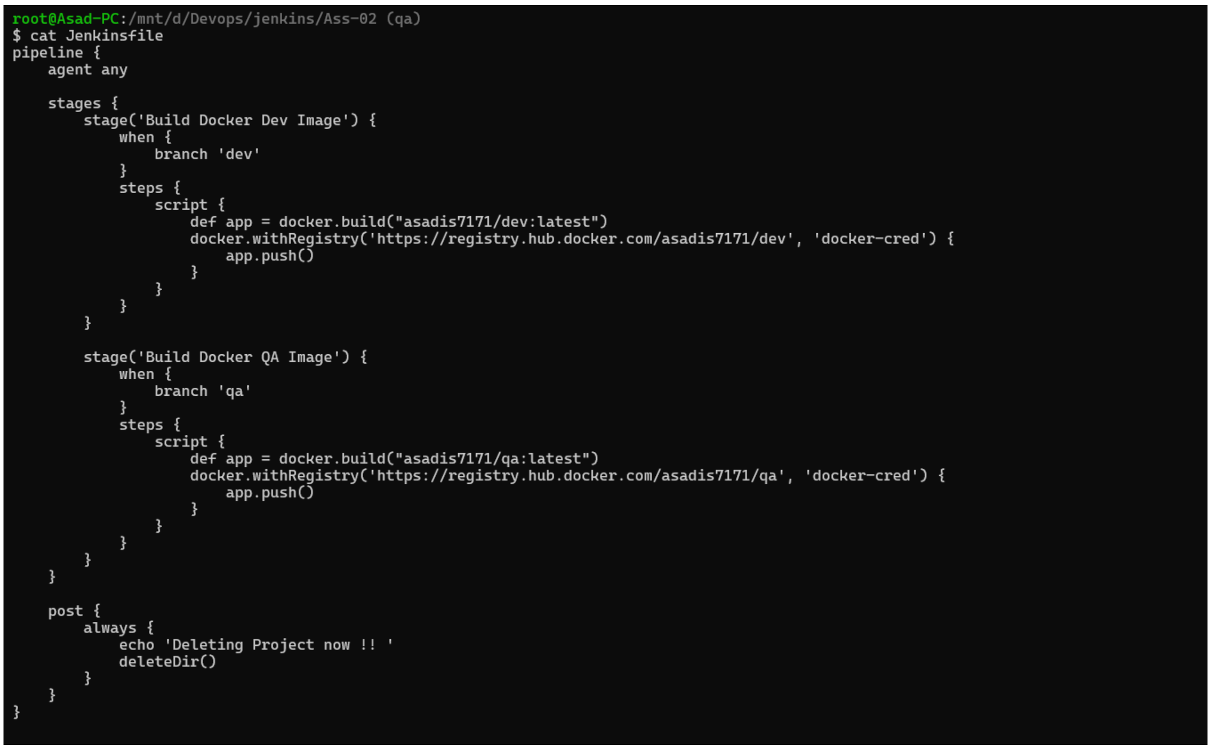
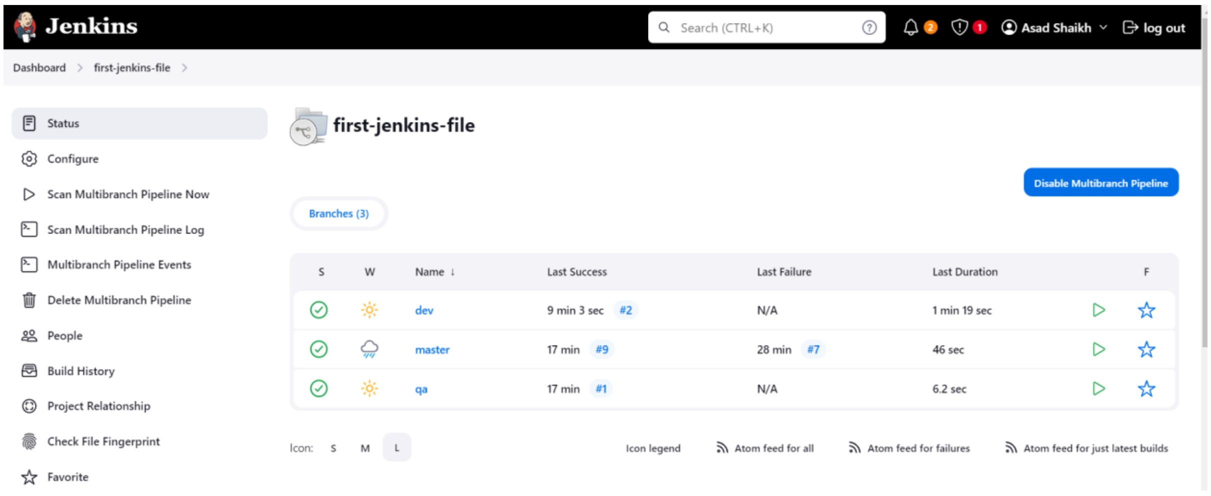
1. Create a Jenkins file to construct and upload Docker images to your Docker-hub registries. Ensure that when the branch is 'dev', the image is constructed and uploaded to the DEV Docker-hub registry. Similarly, when the branch is 'QA', it should be sent to the QA Docker-hub registry.

→



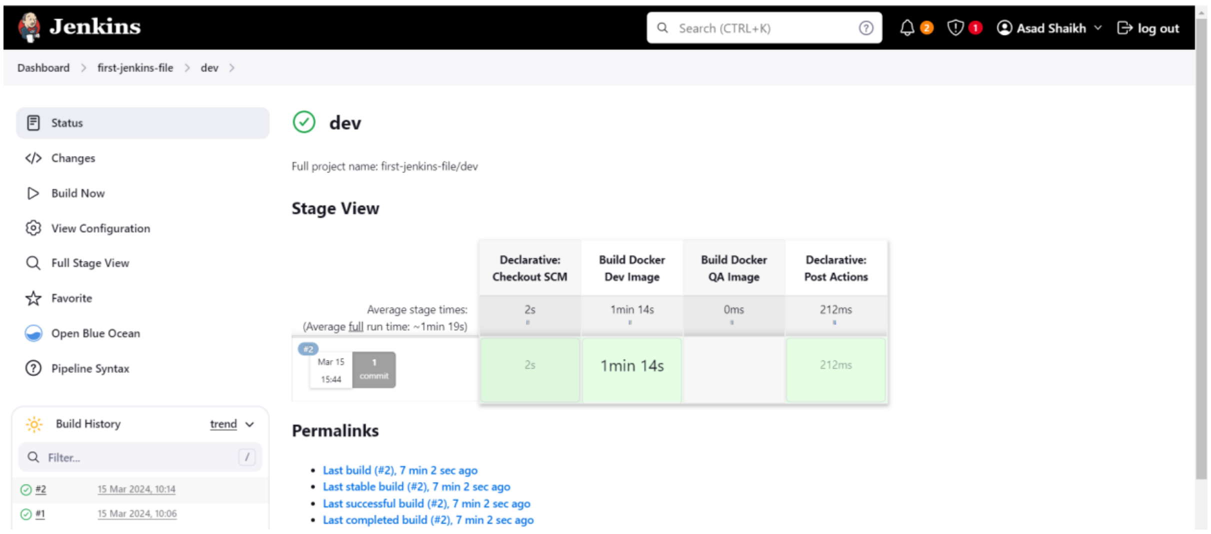
```
root@Asad-PC:/mnt/d/Devops/jenkins/Ass-02 (qa)
$ cat Jenkinsfile
pipeline {
    agent any

    stages {
        stage('Build Docker Dev Image') {
            when {
                branch 'dev'
            }
            steps {
                script {
                    def app = docker.build("asadis7171/dev:latest")
                    docker.withRegistry('https://registry.hub.docker.com/asadis7171/dev', 'docker-cred') {
                        app.push()
                    }
                }
            }
        }

        stage('Build Docker QA Image') {
            when {
                branch 'qa'
            }
            steps {
                script {
                    def app = docker.build("asadis7171/qa:latest")
                    docker.withRegistry('https://registry.hub.docker.com/asadis7171/qa', 'docker-cred') {
                        app.push()
                    }
                }
            }
        }
    }

    post {
        always {
            echo 'Deleting Project now !! '
            deleteDir()
        }
    }
}
```

Multi branch pipeline



S	W	Name ↓	Last Success	Last Failure	Last Duration	F
✓	☀	dev	9 min 3 sec #2	N/A	1 min 19 sec	☆
✓	☁	master	17 min #9	28 min #7	46 sec	☆
✓	☀	qa	17 min #1	N/A	6.2 sec	☆

Running dev branch pipeline



dev

Full project name: first-jenkins-file/dev

Stage View

	Declarative: Checkout SCM	Build Docker Dev Image	Build Docker QA Image	Declarative: Post Actions
Average stage times: (Average full run time: ~1min 19s)	2s	1min 14s	0ms	212ms
#2 Mar 15 15:44 1 commit	2s	1min 14s		212ms

Permalinks

- Last build (#2), 7 min 2 sec ago
- Last stable build (#2), 7 min 2 sec ago
- Last successful build (#2), 7 min 2 sec ago
- Last completed build (#2), 7 min 2 sec ago

#2	15 Mar 2024, 10:14
#1	15 Mar 2024, 10:06

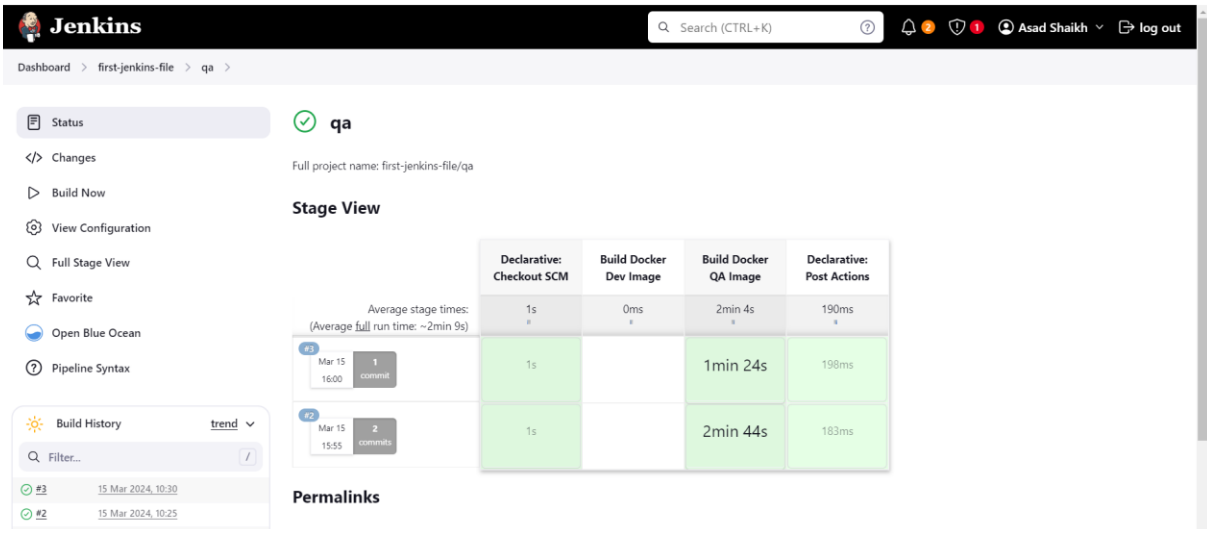
Below are the logs

```
Started by user Asad Shaikh
 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-f85fc5c1fab0c581be53f123d6c0a4e0/.git
# timeout=10
Setting origin to https://github.com/asadis7171/jenkins-ass-02.git
 > git config remote.origin.url https://github.com/asadis7171/jenkins-ass-02.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist,
please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host
key verification.
 > git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/master
Seen branch in repository origin/qa
Seen 3 remote branches
Obtained Ass-02/Jenkinsfile from 26c2632996ecef819c4dc74ab055e33781bb8ba2
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/first-jenkins-file_dev
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential asad.pem
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/asadis7171/jenkins-ass-02.git
 > git init /var/jenkins_home/workspace/first-jenkins-file_dev # timeout=10
Fetching upstream changes from https://github.com/asadis7171/jenkins-ass-02.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials my private key
```

```
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
 > git fetch --no-tags --force --progress -- https://github.com/asadis7171/jenkins-ass-02.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/asadis7171/jenkins-ass-02.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 26c2632996ecef819c4dc74ab055e33781bb8ba2 (dev)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 26c2632996ecef819c4dc74ab055e33781bb8ba2 # timeout=10
Commit message: " added all files"
 > git rev-list --no-walk 9a22f44d853b377f3642a1200b8231e148e11f43 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build Docker Dev Image)
[Pipeline] script
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t asadis7171/dev:latest .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 603B done
#1 DONE 0.1s

#2 [internal] load metadata for docker.io/jenkins/jenkins:2.440.1-jdk17
#2 DONE 12.4s

#3 [internal] load .dockerignore
#3 transferring context: 2B done
#3 DONE 0.0s

#4 [1/6] FROM docker.io/jenkins/jenkins:2.440.1-
jdk17@sha256:01c0b0cf789fa24253090fccea264df223b5e09b14a0ea59f0847c70bdc0f31c
#4 DONE 0.0s

#5 [2/6] RUN apt-get update && apt-get install -y lsb-release
#5 CACHED

#6 [3/6] RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc
https://download.docker.com/linux/debian/gpg
#6 CACHED

#7 [4/6] RUN echo "deb [arch=$(dpkg --print-architecture)   signed-by=/usr/share/keyrings/docker-
archive-keyring.asc]   https://download.docker.com/linux/debian   $(lsb_release -cs) stable" >
/etc/apt/sources.list.d/docker.list
#7 CACHED

#8 [5/6] RUN apt-get update && apt-get install -y docker-ce-cli
#8 CACHED
```

```
#9 [6/6] RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image sha256:a4607c273ead09a6af164931502b3d879ef073654d77dff7f7979e2e81ce02fd
done
#10 naming to docker.io/asadis7171/dev:latest done
#10 DONE 0.0s
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u asadis7171 -p ******** https://registry.hub.docker.com/asadis7171/dev
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/jenkins_home/workspace/first-jenkins-
file_dev@tmp/6de95ba7-be37-4fb9-a97f-5320dc190e3b/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag asadis7171/dev:latest registry.hub.docker.com/asadis7171/dev:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push registry.hub.docker.com/asadis7171/dev:latest
The push refers to repository [registry.hub.docker.com/asadis7171/dev]
9e39961a09b0: Preparing
5bad87ece93a: Preparing
1682bb48f9d2: Preparing
46355de50845: Preparing
05406dc0f097: Preparing
b6731a0d0ccc: Preparing
245d1d0dbfe7: Preparing
72f8bf709949: Preparing
afc00755885d: Preparing
6fd428650ee5: Preparing
b9fe7e6eb007: Preparing
7a50905d05d3: Preparing
0dc9deda4ee6: Preparing
146f222a2562: Preparing
6bdac86cde7b: Preparing
094de07c3af0: Preparing
1a5fc1184c48: Preparing
b9fe7e6eb007: Waiting
7a50905d05d3: Waiting
0dc9deda4ee6: Waiting
245d1d0dbfe7: Waiting
146f222a2562: Waiting
6bdac86cde7b: Waiting
72f8bf709949: Waiting
b6731a0d0ccc: Waiting
094de07c3af0: Waiting
afc00755885d: Waiting
1a5fc1184c48: Waiting
6fd428650ee5: Waiting
46355de50845: Layer already exists
5bad87ece93a: Layer already exists
9e39961a09b0: Layer already exists
05406dc0f097: Layer already exists
1682bb48f9d2: Layer already exists
245d1d0dbfe7: Layer already exists
b6731a0d0ccc: Layer already exists
b9fe7e6eb007: Layer already exists
7a50905d05d3: Layer already exists
72f8bf709949: Layer already exists
afc00755885d: Layer already exists
6fd428650ee5: Layer already exists
094de07c3af0: Layer already exists
6bdac86cde7b: Layer already exists
1a5fc1184c48: Layer already exists
146f222a2562: Layer already exists
0dc9deda4ee6: Layer already exists
latest: digest: sha256:86a064bdcce9cdd21b8e1c27f77a23758a8d6a136a413a20fad82283ebc7d378
size: 3884
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker QA Image)
Stage "Build Docker QA Image" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deleting Project now !!
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Running QA branch pipeline



	Declarative: Checkout SCM	Build Docker Dev Image	Build Docker QA Image	Declarative: Post Actions
Average stage times: (Average full run time: ~2min 9s)	1s	0ms	2min 4s	190ms
#3 Mar 15 16:00 1 commit	1s		1min 24s	198ms
#2 Mar 15 15:55 2 commits	1s		2min 44s	183ms

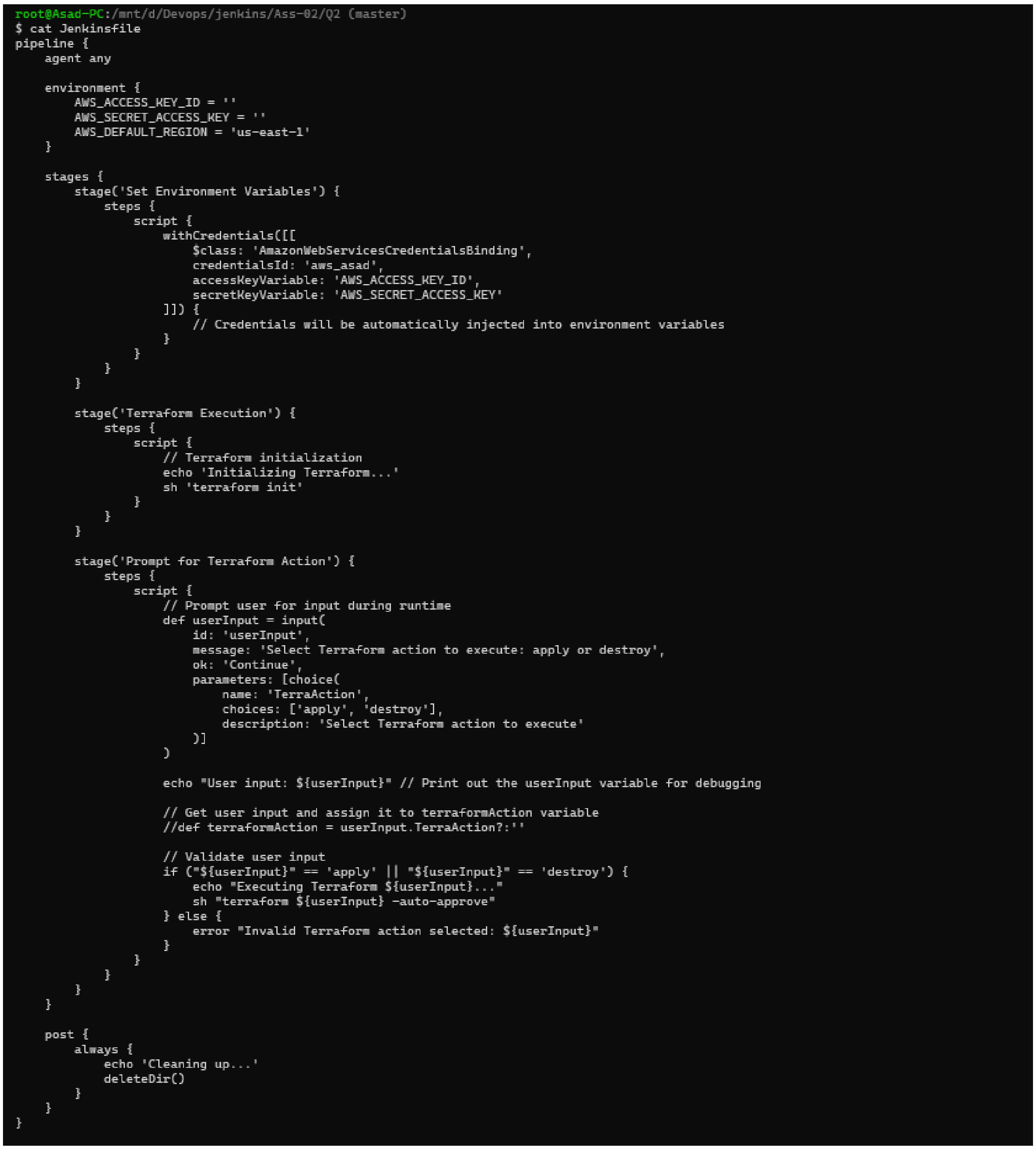
```
Logs
Started by user Asad Shaikh
 > git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-f85fc5c1fab0c581be53f123d6c0a4e0/.git
# timeout=10
Setting origin to https://github.com/asadis7171/jenkins-ass-02.git
 > git config remote.origin.url https://github.com/asadis7171/jenkins-ass-02.git # timeout=10
Fetching origin...
Fetching upstream changes from origin
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist,
please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host
key verification.
 > git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/dev
Seen branch in repository origin/master
Seen branch in repository origin/qa
Seen 3 remote branches
Obtained Ass-02/Jenkinsfile from 182c1bfac960f144f2b37cf5aa83b73cd876a68b
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/first-jenkins-file_qa
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential asad.pem
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/asadis7171/jenkins-ass-02.git
 > git init /var/jenkins_home/workspace/first-jenkins-file_qa # timeout=10
Fetching upstream changes from https://github.com/asadis7171/jenkins-ass-02.git
 > git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials my private key
```

```
Verifying host key using known hosts file
You're using 'Known hosts file' strategy to verify ssh host keys, but your known_hosts file does not exist, please go to 'Manage Jenkins' -> 'Security' -> 'Git Host Key Verification Configuration' and configure host key verification.
 > git fetch --no-tags --force --progress -- https://github.com/asadis7171/jenkins-ass-02.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/asadis7171/jenkins-ass-02.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 182c1bfac960f144f2b37cf5aa83b73cd876a68b (qa)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 182c1bfac960f144f2b37cf5aa83b73cd876a68b # timeout=10
Commit message: " added all files"
 > git rev-list --no-walk 69d0f13cc1c0ce3b944d262047ac62324600e841 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build Docker Dev Image)
Stage "Build Docker Dev Image" skipped due to when conditional
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker QA Image)
[Pipeline] script
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t asadis7171/qa:latest .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 603B done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/jenkins/jenkins:2.440.1-jdk17
#2 DONE 1.5s

#3 [internal] load .dockerignore
#3 transferring context: 2B done
#3 DONE 0.0s

#4 [1/6] FROM docker.io/jenkins/jenkins:2.440.1-
jdk17@sha256:01c0b0cf789fa24253090fccea264df223b5e09b14a0ea59f0847c70bdc0f31c
#4 DONE 0.0s

#5 [2/6] RUN apt-get update && apt-get install -y lsb-release
#5 CACHED

#6 [3/6] RUN curl -fsSLo /usr/share/keyrings/docker-archive-keyring.asc
https://download.docker.com/linux/debian/gpg
#6 CACHED

#7 [4/6] RUN echo "deb [arch=$(dpkg --print-architecture)   signed-by=/usr/share/keyrings/docker-
archive-keyring.asc]   https://download.docker.com/linux/debian   $(lsb_release -cs) stable" >
/etc/apt/sources.list.d/docker.list
```

```
#7 CACHED

#8 [5/6] RUN apt-get update && apt-get install -y docker-ce-cli
#8 CACHED

#9 [6/6] RUN jenkins-plugin-cli --plugins "blueocean docker-workflow"
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image sha256:a4607c273ead09a6af164931502b3d879ef073654d77dff7f7979e2e81ce02fd
done
#10 naming to docker.io/asadis7171/qa:latest done
#10 DONE 0.0s
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withDockerRegistry
$ docker login -u asadis7171 -p ******** https://registry.hub.docker.com/asadis7171/qa
WARNING! Using --password via the CLI is insecure. Use --password-stdin.
WARNING! Your password will be stored unencrypted in /var/jenkins_home/workspace/first-jenkins-
file_qa@tmp/e6dccfca-2589-430e-99cd-74d9e98ee3e7/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] {
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker tag asadis7171/qa:latest registry.hub.docker.com/asadis7171/qa:latest
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] isUnix
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker push registry.hub.docker.com/asadis7171/qa:latest
The push refers to repository [registry.hub.docker.com/asadis7171/qa]
9e39961a09b0: Preparing
5bad87ece93a: Preparing
1682bb48f9d2: Preparing
46355de50845: Preparing
05406dc0f097: Preparing
b6731a0d0ccc: Preparing
245d1d0dbfe7: Preparing
72f8bf709949: Preparing
afc00755885d: Preparing
6fd428650ee5: Preparing
b9fe7e6eb007: Preparing
7a50905d05d3: Preparing
0dc9deda4ee6: Preparing
146f222a2562: Preparing
6bdac86cde7b: Preparing
094de07c3af0: Preparing
1a5fc1184c48: Preparing
6fd428650ee5: Waiting
b9fe7e6eb007: Waiting
7a50905d05d3: Waiting
0dc9deda4ee6: Waiting
146f222a2562: Waiting
6bdac86cde7b: Waiting
094de07c3af0: Waiting
1a5fc1184c48: Waiting
245d1d0dbfe7: Waiting
72f8bf709949: Waiting
b6731a0d0ccc: Waiting
afc00755885d: Waiting
1682bb48f9d2: Layer already exists
5bad87ece93a: Layer already exists
46355de50845: Layer already exists
9e39961a09b0: Layer already exists
05406dc0f097: Layer already exists
b6731a0d0ccc: Layer already exists
72f8bf709949: Layer already exists
245d1d0dbfe7: Layer already exists
6fd428650ee5: Layer already exists
b9fe7e6eb007: Layer already exists
6bdac86cde7b: Layer already exists
afc00755885d: Layer already exists
094de07c3af0: Layer already exists
7a50905d05d3: Layer already exists
0dc9deda4ee6: Layer already exists
1a5fc1184c48: Layer already exists
146f222a2562: Layer already exists
latest: digest: sha256:86a064bdcce9cdd21b8e1c27f77a23758a8d6a136a413a20fad82283ebc7d378
size: 3884
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deleting Project now !!
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

2. Design a Jenkins file to execute any Terraform code, prompting the user for two inputs: Terraform apply and Terraform destroy. Depending on the provided inputs, execute the corresponding Terraform command accordingly.

→



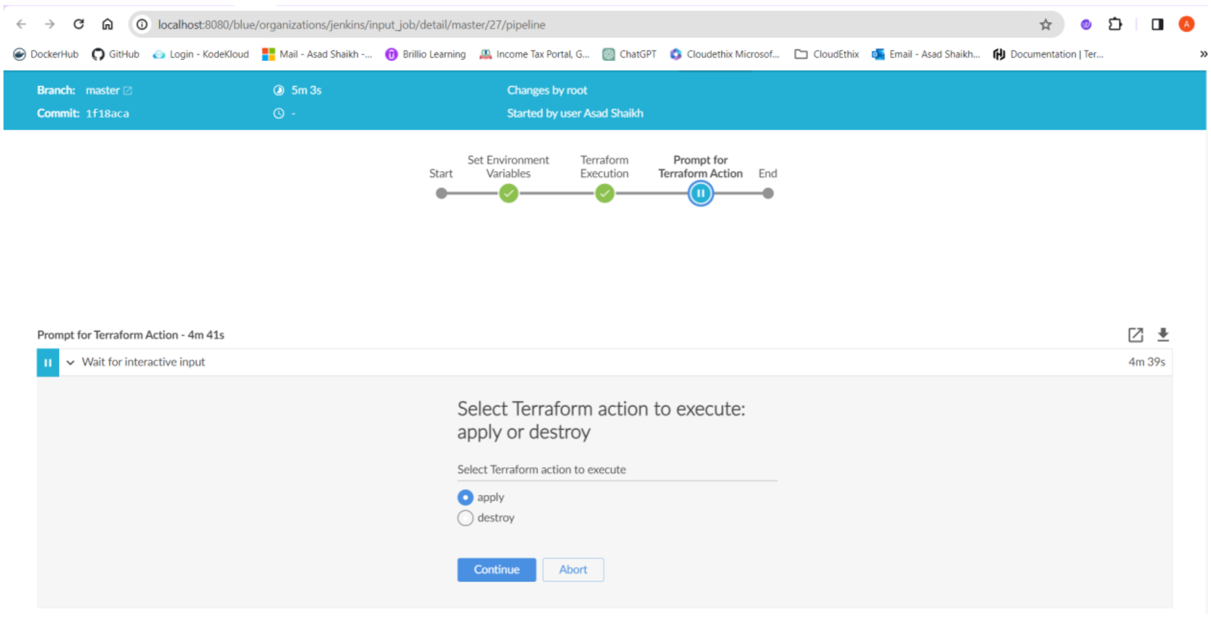
```
root@Asad-PC:/mnt/d/Devops/jenkins/Ass-02/Q2 (master)
$ cat Jenkinsfile
pipeline {
    agent any

    environment {
        AWS_ACCESS_KEY_ID = ''
        AWS_SECRET_ACCESS_KEY = ''
        AWS_DEFAULT_REGION = 'us-east-1'
    }

    stages {
        stage('Set Environment Variables') {
            steps {
                script {
                    withCredentials([[
                        $class: 'AmazonWebServicesCredentialsBinding',
                        credentialsId: 'aws_asad',
                        accessKeyVariable: 'AWS_ACCESS_KEY_ID',
                        secretKeyVariable: 'AWS_SECRET_ACCESS_KEY'
                    ]]) {
                        // Credentials will be automatically injected into environment variables
                    }
                }
            }
        }

        stage('Terraform Execution') {
            steps {
                script {
                    // Terraform initialization
                    echo 'Initializing Terraform...'
                    sh 'terraform init'
                }
            }
        }

        stage('Prompt for Terraform Action') {
            steps {
                script {
                    // Prompt user for input during runtime
                    def userInput = input(
                        id: 'userInput',
                        message: 'Select Terraform action to execute: apply or destroy',
                        ok: 'Continue',
                        parameters: [choice(
                            name: 'TerraAction',
                            choices: ['apply', 'destroy'],
                            description: 'Select Terraform action to execute'
                        )]
                    )

                    echo "User input: ${userInput}" // Print out the userInput variable for debugging

                    // Get user input and assign it to terraformAction variable
                    //def terraformAction = userInput.TerraAction?:''

                    // Validate user input
                    if ("${userInput}" == 'apply' || "${userInput}" == 'destroy') {
                        echo "Executing Terraform ${userInput}..."
                        sh "terraform ${userInput} -auto-approve"
                    } else {
                        error "Invalid Terraform action selected: ${userInput}"
                    }
                }
            }
        }
    }

    post {
        always {
            echo 'Cleaning up...'
            deleteDir()
        }
    }
}
```

Executing Jenkinsfile

logs

```
Started by user Asad Shaikh
 > /usr/bin/git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-
92285cb47f5ebf5c30f191b60844ee58/.git # timeout=10
Setting origin to https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git #
timeout=10
Fetching origin...
Fetching upstream changes from origin
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > /usr/bin/git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
 > /usr/bin/git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/master
Seen 1 remote branch
Obtained Jenkinsfile from 1f18aca07085464167b417434a970af6b51a3ff6
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/input_job_master
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential asad.pem
 > /usr/bin/git rev-parse --resolve-git-dir /var/jenkins_home/workspace/input_job_master/.git #
timeout=10
Fetching changes from the remote Git repository
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git #
timeout=10
Fetching without tags
Fetching upstream changes from https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
 > /usr/bin/git fetch --no-tags --force --progress -- https://github.com/asadis7171/jenkins-input-
variable.git +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking out Revision 1f18aca07085464167b417434a970af6b51a3ff6 (master)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f 1f18aca07085464167b417434a970af6b51a3ff6 # timeout=10
Commit message: " added all files"
 > /usr/bin/git rev-list --no-walk aed73a9096de40127a8b9ef856387327b5256919 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Set Environment Variables)
[Pipeline] script
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $AWS_ACCESS_KEY_ID or $AWS_SECRET_ACCESS_KEY
[Pipeline] {
```

```
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Terraform Execution)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Initializing Terraform...
[Pipeline] sh
+ terraform init

 [0m [1mInitializing the backend... [0m

 [0m [1mInitializing provider plugins... [0m
- Finding hashicorp/aws versions matching "5.41.0"...
- Installing hashicorp/aws v5.41.0...
- Installed hashicorp/aws v5.41.0 (signed by HashiCorp)

Terraform has created a lock file  [1m.terraform.lock.hcl  [0m to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future. [0m

 [0m [1m [32mTerraform has been successfully initialized! [0m [32m [0m
 [0m [32m
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary. [0m
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Prompt for Terraform Action)
[Pipeline] input
Input requested
```



Branch: master
Commit: 1f18aca
5m 3s
-
Changes by root
Started by user Asad Shaikh

Start — Set Environment Variables — Terraform Execution — Prompt for Terraform Action — End

Prompt for Terraform Action - 4m 41s

Wait for interactive input 4m 39s

Select Terraform action to execute: apply or destroy

Select Terraform action to execute
[✓ apply]
destroy

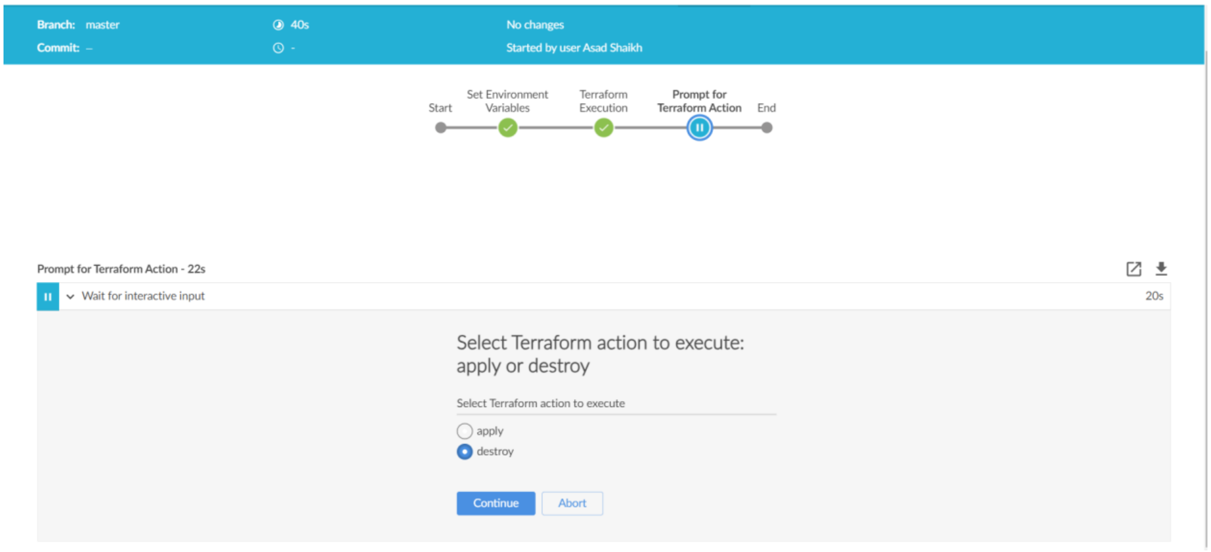
```
Started by user Asad Shaikh
 > /usr/bin/git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-
92285cb47f5ebf5c30f191b60844ee58/.git # timeout=10
Setting origin to https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git #
timeout=10
Fetching origin...
Fetching upstream changes from origin
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > /usr/bin/git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
 > /usr/bin/git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/master
Seen 1 remote branch
Obtained Jenkinsfile from 62214b71307ff974f7753d186692a77733d8e30a
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/input_job_master
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential asad.pem
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git init /var/jenkins_home/workspace/input_job_master # timeout=10
Fetching upstream changes from https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
```

```
 > /usr/bin/git fetch --no-tags --force --progress -- https://github.com/asadis7171/jenkins-input-variable.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git # timeout=10
 > /usr/bin/git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 62214b71307ff974f7753d186692a77733d8e30a (master)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f 62214b71307ff974f7753d186692a77733d8e30a # timeout=10
Commit message: " added all files"
 > /usr/bin/git rev-list --no-walk f685f4f980e120d6775f53970e4a0176d852cc57 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Set Environment Variables)
[Pipeline] script
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $AWS_ACCESS_KEY_ID or $AWS_SECRET_ACCESS_KEY
[Pipeline] {
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Terraform Execution)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Initializing Terraform...
[Pipeline] sh
+ terraform init

  [0m  [1mInitializing the backend...  [0m

  [0m  [1mInitializing provider plugins...  [0m
- Finding hashicorp/aws versions matching "5.41.0"...
- Installing hashicorp/aws v5.41.0...
- Installed hashicorp/aws v5.41.0 (signed by HashiCorp)

Terraform has created a lock file   [1m.terraform.lock.hcl   [0m to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.  [0m

  [0m  [1m  [32mTerraform has been successfully initialized!  [0m  [32m  [0m
  [0m  [32m
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
```

```
commands will detect it and remind you to do so if necessary.   [0m
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Prompt for Terraform Action)
[Pipeline] script
[Pipeline] {
[Pipeline] input
Input requested
Approved by Asad Shaikh
[Pipeline] echo
User input: apply
[Pipeline] echo
Executing Terraform apply...
[Pipeline] sh
+ terraform apply -auto-approve

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
    [32m+   [0m create   [0m

Terraform will perform the following actions:

  [1m  # aws_instance.this_ec2   [0m will be created
  [0m    [32m+   [0m   [0m resource "aws_instance" "this_ec2" {
       [32m+   [0m   [0m ami                                  = "ami-0d7a109bf30624c99"
       [32m+   [0m   [0m arn                                  = (known after apply)
       [32m+   [0m   [0m associate_public_ip_address          = (known after apply)
       [32m+   [0m   [0m availability_zone                    = (known after apply)
       [32m+   [0m   [0m cpu_core_count                       = (known after apply)
       [32m+   [0m   [0m cpu_threads_per_core                 = (known after apply)
       [32m+   [0m   [0m disable_api_stop                     = (known after apply)
       [32m+   [0m   [0m disable_api_termination              = (known after apply)
       [32m+   [0m   [0m ebs_optimized                        = (known after apply)
       [32m+   [0m   [0m get_password_data                    = false
       [32m+   [0m   [0m host_id                              = (known after apply)
       [32m+   [0m   [0m host_resource_group_arn              = (known after apply)
       [32m+   [0m   [0m iam_instance_profile                 = (known after apply)
       [32m+   [0m   [0m id                                   = (known after apply)
       [32m+   [0m   [0m instance_initiated_shutdown_behavior = (known after apply)
       [32m+   [0m   [0m instance_lifecycle                   = (known after apply)
       [32m+   [0m   [0m instance_state                       = (known after apply)
       [32m+   [0m   [0m instance_type                        = "t2.micro"
       [32m+   [0m   [0m ipv6_address_count                   = (known after apply)
       [32m+   [0m   [0m ipv6_addresses                       = (known after apply)
       [32m+   [0m   [0m key_name                             = "asad_practice"
       [32m+   [0m   [0m monitoring                           = (known after apply)
       [32m+   [0m   [0m outpost_arn                          = (known after apply)
       [32m+   [0m   [0m password_data                        = (known after apply)
       [32m+   [0m   [0m placement_group                      = (known after apply)
       [32m+   [0m   [0m placement_partition_number           = (known after apply)
       [32m+   [0m   [0m primary_network_interface_id         = (known after apply)
       [32m+   [0m   [0m private_dns                          = (known after apply)
       [32m+   [0m   [0m private_ip                           = (known after apply)
       [32m+   [0m   [0m public_dns                           = (known after apply)
       [32m+   [0m   [0m public_ip                            = (known after apply)
       [32m+   [0m   [0m secondary_private_ips                = (known after apply)
```

```
      [32m+   [0m   [0m security_groups                 = (known after apply)
      [32m+   [0m   [0m source_dest_check                = true
      [32m+   [0m   [0m spot_instance_request_id          = (known after apply)
      [32m+   [0m   [0m subnet_id                        = (known after apply)
      [32m+   [0m   [0m tags_all                         = (known after apply)
      [32m+   [0m   [0m tenancy                          = (known after apply)
      [32m+   [0m   [0m user_data                        = (known after apply)
      [32m+   [0m   [0m user_data_base64                 = (known after apply)
      [32m+   [0m   [0m user_data_replace_on_change       = false
      [32m+   [0m   [0m vpc_security_group_ids           = (known after apply)
    }

  [1mPlan:   [0m 1 to add, 0 to change, 0 to destroy.
  [0m   [0m   [1maws_instance.this_ec2: Creating...   [0m   [0m
  [0m   [1maws_instance.this_ec2: Still creating... [10s elapsed]   [0m   [0m
  [0m   [1maws_instance.this_ec2: Still creating... [20s elapsed]   [0m   [0m
  [0m   [1maws_instance.this_ec2: Creation complete after 27s [id=i-0ba26a08eaeeb7be9]   [0m
  [0m   [1m   [32m
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
  [0m
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Cleaning up...
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

`now executing for destroy`


Branch: master
Commit: –
40s
No changes
Started by user Asad Shaikh

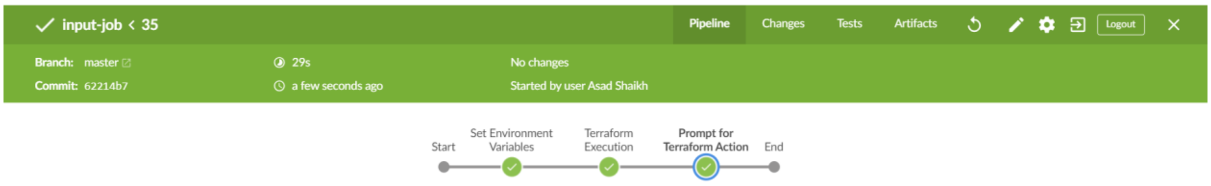
Prompt for Terraform Action - 22s

Wait for interactive input 20s

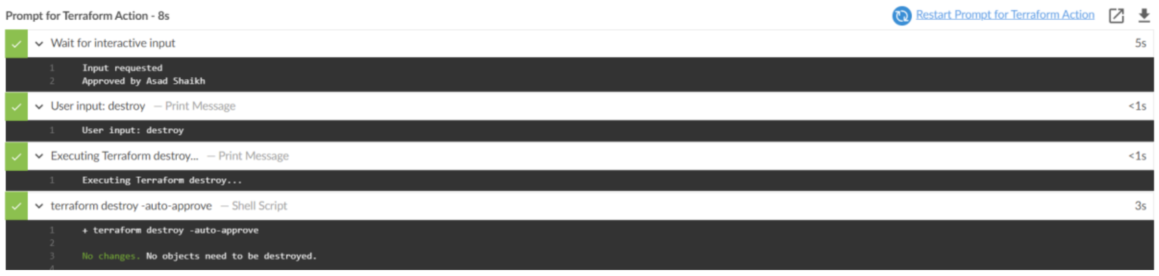
Select Terraform action to execute:
apply or destroy

Select Terraform action to execute

apply [✓ destroy]



Branch: master
Commit: 62214b7
29s
a few seconds ago
No changes
Started by user Asad Shaikh



Prompt for Terraform Action - 8s

Wait for interactive input 5s

```
Input requested
Approved by Asad Shaikh
```

User input: destroy — Print Message <1s

```
User input: destroy
```

Executing Terraform destroy... — Print Message <1s

```
Executing Terraform destroy...
```

terraform destroy -auto-approve — Shell Script 3s

```
+ terraform destroy -auto-approve

No changes. No objects need to be destroyed.
```

```
Started by user Asad Shaikh
 > /usr/bin/git rev-parse --resolve-git-dir /var/jenkins_home/caches/git-
92285cb47f5ebf5c30f191b60844ee58/.git # timeout=10
Setting origin to https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git #
timeout=10
Fetching origin...
Fetching upstream changes from origin
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
 > /usr/bin/git config --get remote.origin.url # timeout=10
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
 > /usr/bin/git fetch --tags --force --progress -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Seen branch in repository origin/master
Seen 1 remote branch
Obtained Jenkinsfile from 62214b71307ff974f7753d186692a77733d8e30a
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/input_job_master
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential asad.pem
Cloning the remote Git repository
Cloning with configured refspecs honoured and without tags
Cloning repository https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git init /var/jenkins_home/workspace/input_job_master # timeout=10
Fetching upstream changes from https://github.com/asadis7171/jenkins-input-variable.git
 > /usr/bin/git --version # timeout=10
 > git --version # 'git version 2.39.2'
using GIT_SSH to set credentials my private key
Verifying host key using known hosts file
 > /usr/bin/git fetch --no-tags --force --progress -- https://github.com/asadis7171/jenkins-input-
variable.git +refs/heads/*:refs/remotes/origin/* # timeout=10
```

```
 > /usr/bin/git config remote.origin.url https://github.com/asadis7171/jenkins-input-variable.git # timeout=10
 > /usr/bin/git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
Checking out Revision 62214b71307ff974f7753d186692a77733d8e30a (master)
 > /usr/bin/git config core.sparsecheckout # timeout=10
 > /usr/bin/git checkout -f 62214b71307ff974f7753d186692a77733d8e30a # timeout=10
Commit message: " added all files"
 > /usr/bin/git rev-list --no-walk 62214b71307ff974f7753d186692a77733d8e30a # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Set Environment Variables)
[Pipeline] script
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $AWS_ACCESS_KEY_ID or $AWS_SECRET_ACCESS_KEY
[Pipeline] {
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Terraform Execution)
[Pipeline] script
[Pipeline] {
[Pipeline] echo
Initializing Terraform...
[Pipeline] sh
+ terraform init

  [0m  [1mInitializing the backend...  [0m

  [0m  [1mInitializing provider plugins...  [0m
- Finding hashicorp/aws versions matching "5.41.0"...
- Installing hashicorp/aws v5.41.0...
- Installed hashicorp/aws v5.41.0 (signed by HashiCorp)

Terraform has created a lock file  [1m.terraform.lock.hcl  [0m to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.  [0m

  [0m  [1m  [32mTerraform has been successfully initialized!  [0m  [32m  [0m
  [0m  [32m
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.  [0m
[Pipeline] }
```

```
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Prompt for Terraform Action)
[Pipeline] script
[Pipeline] {
[Pipeline] input
Input requested
Approved by Asad Shaikh
[Pipeline] echo
User input: destroy
[Pipeline] echo
Executing Terraform destroy...
[Pipeline] sh
+ terraform destroy -auto-approve

  [0m  [1m  [32mNo changes.  [0m  [1m No objects need to be destroyed.  [0m

  [0mEither you have not created any objects yet or the existing objects were
already deleted outside of Terraform.
  [0m  [1m  [32m
Destroy complete! Resources: 0 destroyed.
  [0m
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Cleaning up...
[Pipeline] deleteDir
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

The destruction didn't occur because the .tfstate file couldn't be located to manage the resources. To address this, we need to set up an S3 backend and ensure that Jenkins has the necessary access permissions for S3. However, the task at hand was solely focused on obtaining user input without executing any actual apply or destroy actions. We'll tackle the S3 backend setup and permissions in a separate session

Thanks   Asad Shaikh