

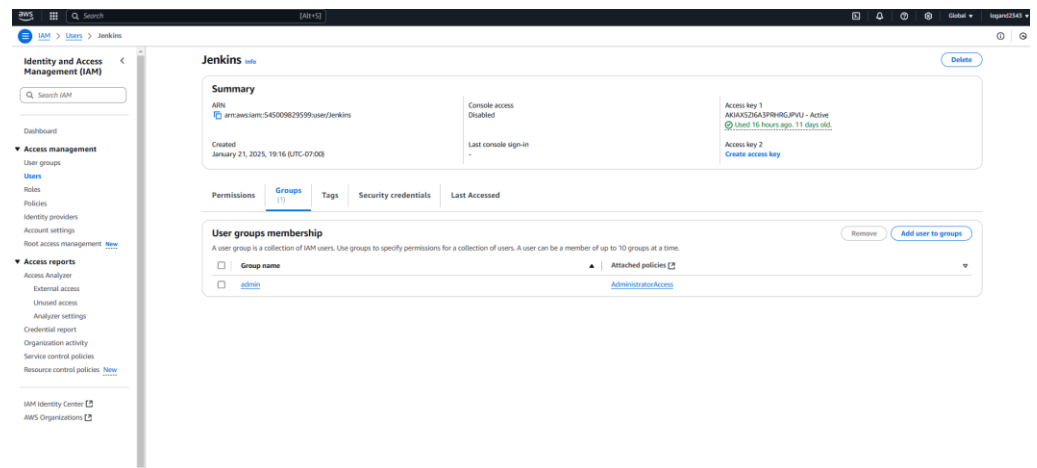
Jenkins Documentation

Overview Steps

1. For Jenkins service, create a dedicated AWS IAM user with Access and Secret Keys.
2. Turn on Jenkins container via the CLI.
3. Within the CLI, install both Terraform and AWSCLI.
4. Access Jenkins in web browser via the localhost80:80 url and login with Jenkins credentials.
5. Add our Jenkins AWS IAM user keys to our Jenkins credentials settings.
6. Create new Github repository with working Terraform project and ensure the Jenkins file is properly configured and included in repository.
7. Create a new Pipeline in Jenkins and configure it to utilize our Github repository.
8. Run our new Pipeline in Jenkins to fire off the build of our Terraform project.
9. Confirm the Pipeline build was successful.
10. Perform Terraform teardown within our container in the CLI.

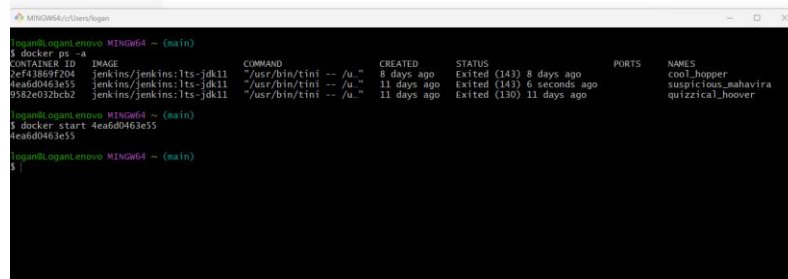
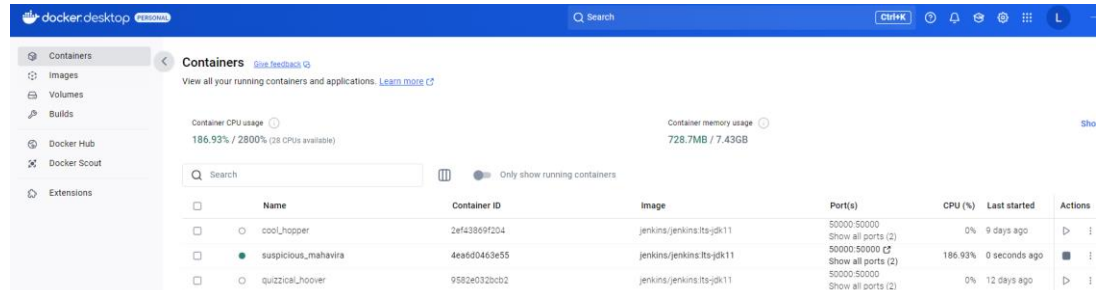
Steps Breakdown

1. For Jenkins service, create a dedicated AWS IAM user with Access and Secret Keys.
 - a. Log into AWS and navigate to the IAM section.
 - b. Create a new user for Jenkins.
 - c. Assign the user to a group with administrator permissions.
 - d. Create an access key and save the Access Key and Secret Key.
 - e. Screenshots

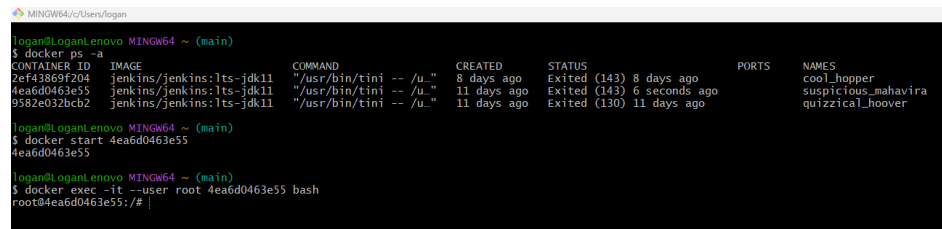


- i.
2. Turn on Jenkins container via the CLI.
 - a. Open Gitbash

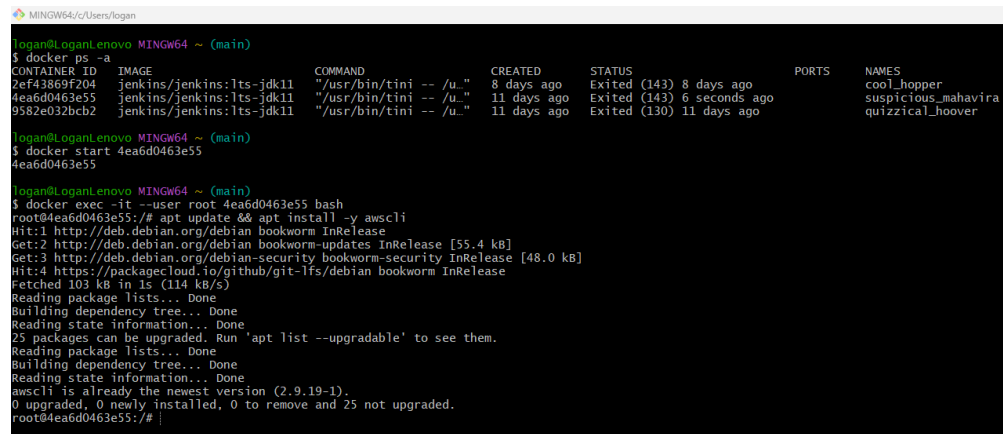
- b. Run the docker command “docker ps -a” to get a list of all containers on local machine so we can retrieve the container id.
- c. Run the docker command “docker start <container id>” to start the docker container.



- d.
 - a. Run this docker command “docker exec -it --user root <container id> bash” to ssh into this running container.



- i.
 - b. Run this docker command “apt update && apt install -y awscli” to update our container and install awscli.



- c. Run this docker command “mkdir -p /home/jenkins/bin” to make a directory for Jenkins.
- d. Run this docker command “curl -fsSL https://releases.hashicorp.com/terraform/1.5.7/terraform_1.5.7_linux_amd64.zip -o /home/jenkins/terraform.zip” to grab terraform from hashicorp and save the zip file inside our container.
- e. Run this docker command “unzip /home/jenkins/terraform.zip -d /home/jenkins/bin” to unzip our terraform zip and place it in the Jenkins directory.
- f. Run this docker command “rm /home/jenkins/terraform.zip” to remove the zip file since it is no longer necessary.
- g. Run this docker command “export PATH="/home/jenkins/bin:\$PATH" to let the container know where we installed terraform for future runs.
- h. Run the following commands “terraform --version” and “aws --version” to ensure both are installed.

```

MINGW64/c/Users/Logan
logan@LoganLenovo MINGW64 ~ (main)
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
2ef43869f204   jenkins/jenkins:lts-jdk11  "/usr/bin/tini -- /u..."  8 days ago    Exited (143) 8 days ago           cool_hopper
4ea6d0463e55   jenkins/jenkins:lts-jdk11  "/usr/bin/tini -- /u..."  11 days ago   Exited (143) 6 seconds ago        suspicious_mahavira
9582e032bcb2   jenkins/jenkins:lts-jdk11  "/usr/bin/tini -- /u..."  11 days ago   Exited (130) 11 days ago        quizzical_hoover

logan@LoganLenovo MINGW64 ~ (main)
$ docker start 4ea6d0463e55
4ea6d0463e55

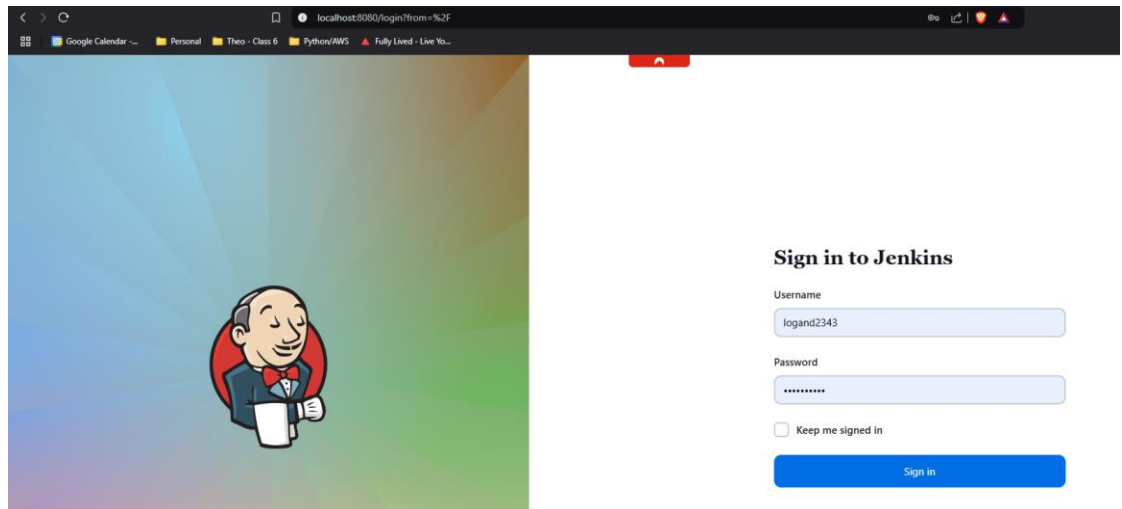
logan@LoganLenovo MINGW64 ~ (main)
$ docker exec -it --user root 4ea6d0463e55 bash
Building dependency tree... Done
Reading state information... Done
awscli is already the newest version (2.9.19-1).
0 upgraded, 0 newly installed, 0 to remove and 25 not upgraded.
root@4ea6d0463e55:/# mkdir -p /home/jenkins/bin
root@4ea6d0463e55:/# curl -fsSL https://releases.hashicorp.com/terraform/1.5.7/terraform_1.5.7_linux_amd64.zip -o /home/jenkins/terraform.zip
root@4ea6d0463e55:/# unzip /home/jenkins/terraform.zip -d /home/jenkins/bin
Archive: /home/jenkins/terraform.zip
  inflating: /home/jenkins/bin/terraform
root@4ea6d0463e55:/# rm /home/jenkins/terraform.zip
root@4ea6d0463e55:/# export PATH="/home/jenkins/bin:$PATH"
root@4ea6d0463e55:/# terraform --version
Terraform v1.5.7
on linux_amd64

Your version of Terraform is out of date! The latest version
is 1.10.5. You can update by downloading from https://www.terraform.io/downloads.html
root@4ea6d0463e55:/# aws --version
aws-cli/2.9.19 Python/3.11.2 Linux/5.15.167.4-microsoft-standard-WSL2 source/x86_64.debian.12 prompt/off
root@4ea6d0463e55:/#

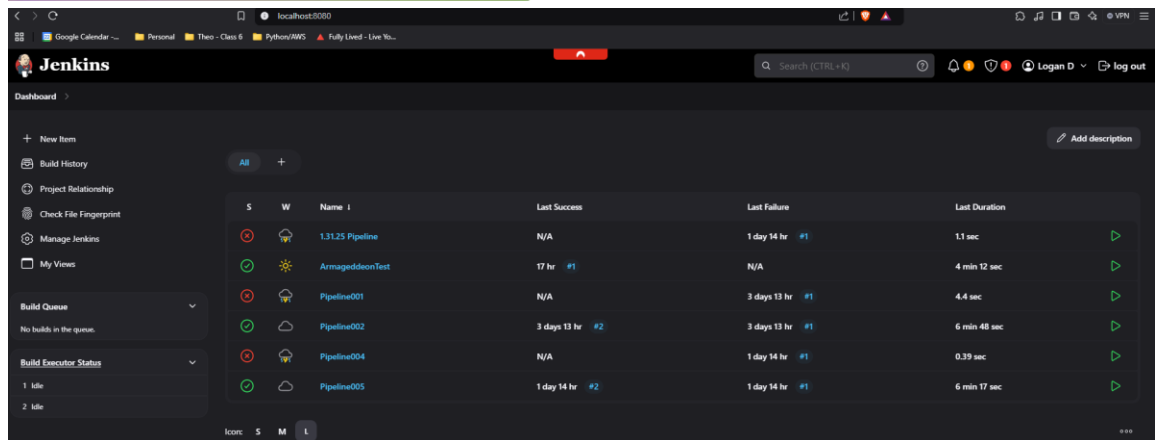
```

i.

4. Access Jenkins in web browser via the localhost80:80 url and login with Jenkins credentials.

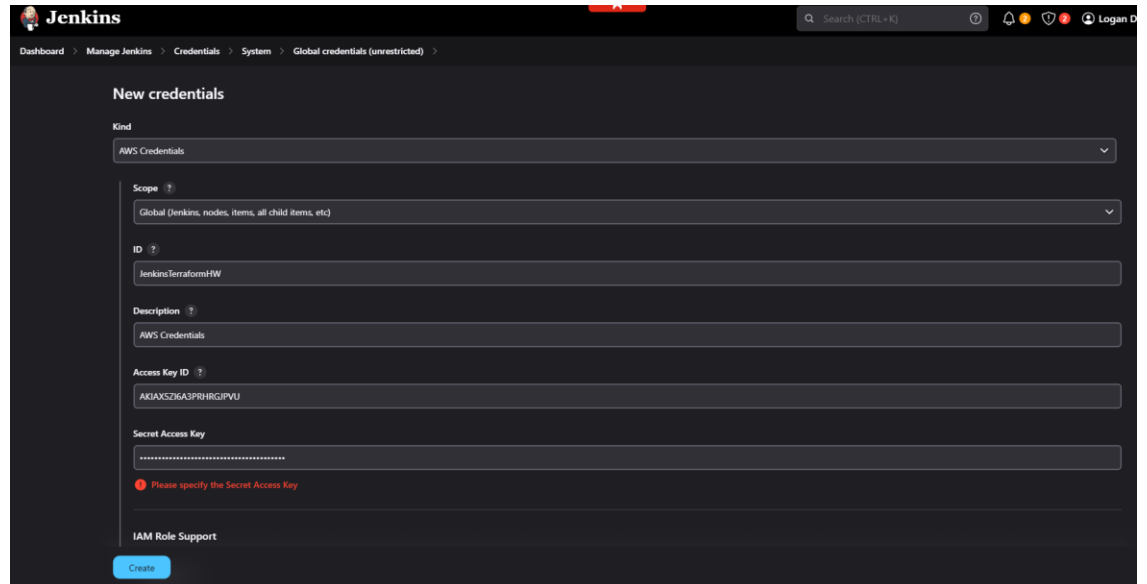


a.



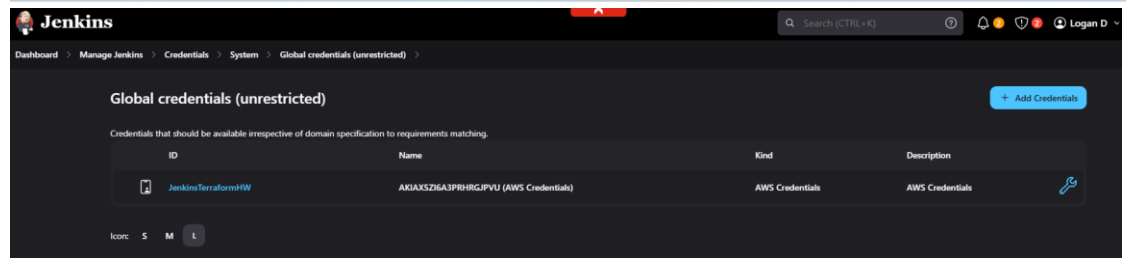
b.

5. Add our Jenkins AWS IAM user keys to our Jenkins credentials settings.
 - a. Inside Jenkins, go to Manage Jenkins and then Credentials.
 - b. Click on System, then Global credentials (unrestricted) and then click on the Add Credentials button.
 - c. For Kind select AWS Credentials. Add an ID label for this credential.
 - d. Add the Access Key ID & Secret Access Key.
 - e. Click Create to save credentials in Jenkins.



The screenshot shows the Jenkins 'New credentials' form. The 'Kind' is set to 'AWS Credentials'. The 'Scope' is 'Global (Jenkins, nodes, items, all child items, etc)'. The 'ID' is 'JenkinsTerraformHW'. The 'Description' is 'AWS Credentials'. The 'Access Key ID' is 'AKIA526A3PRHRCIPVU'. The 'Secret Access Key' field is empty, with a red error message 'Please specify the Secret Access Key'. The 'IAM Role Support' checkbox is unchecked. A 'Create' button is at the bottom.

f.

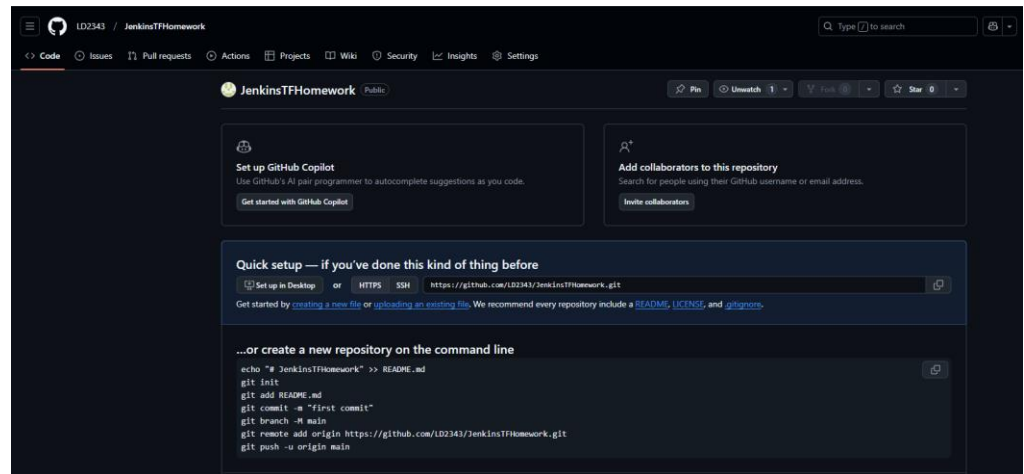


The screenshot shows the Jenkins 'Global credentials (unrestricted)' page. It contains a table with one credential entry.

| ID | Name | Kind | Description |
|--------------------|--------------------------------------|-----------------|-----------------|
| JenkinsTerraformHW | AKIA526A3PRHRCIPVU (AWS Credentials) | AWS Credentials | AWS Credentials |

g.

6. Create new Github repository with working Terraform project and ensure the Jenkins file is properly configured and included in repository.
 - a. Log into Github and create a new repository.



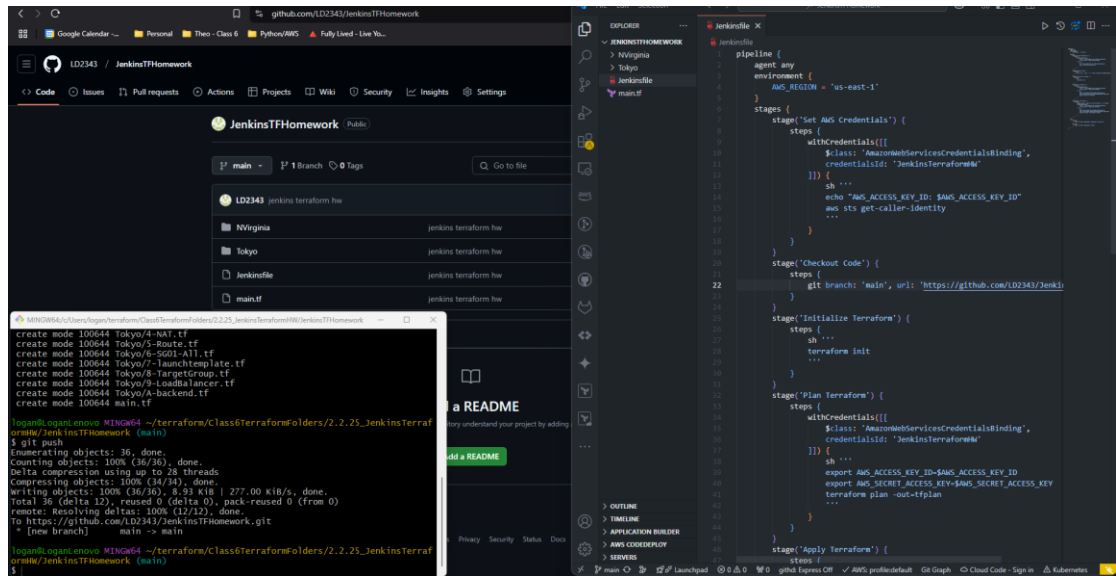
The screenshot shows the Github repository page for 'JenkinsTFHomework'. It includes a 'Quick setup' section with instructions for setting up the repository on a desktop or using HTTPS/SSH. Below this, there is a code block with the following commands:

```

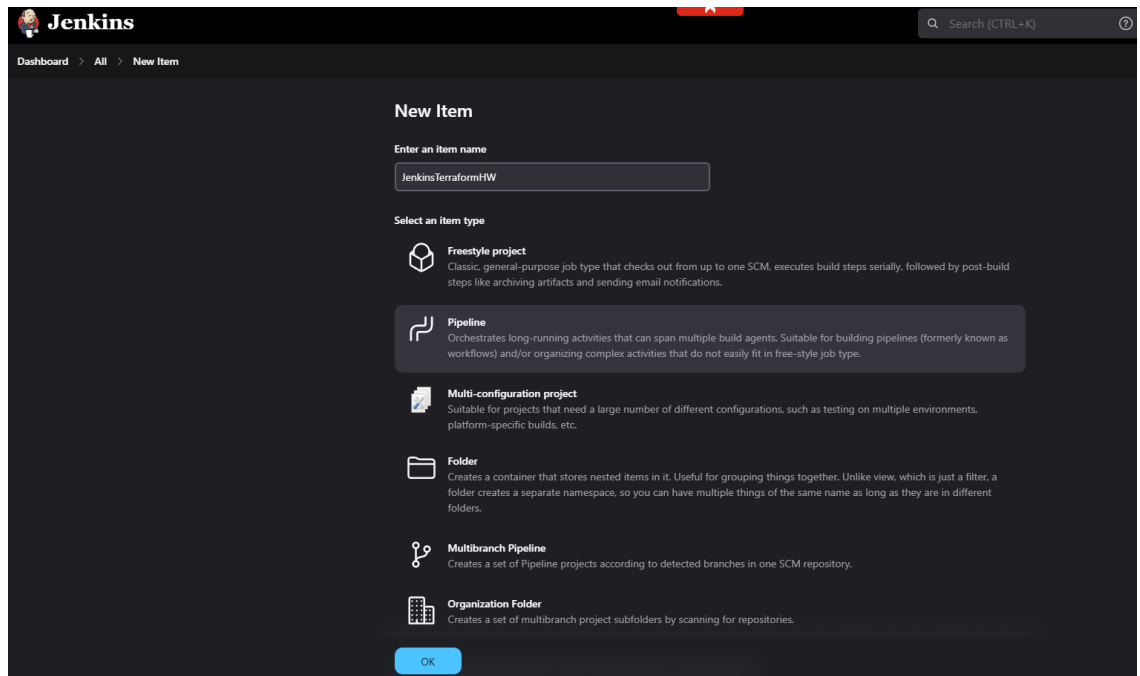
...or create a new repository on the command line
echo "# JenkinsTFHomework" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -m main
git remote add origin https://github.com/LD2343/JenkinsTFHomework.git
git push -u origin main
  
```

i.

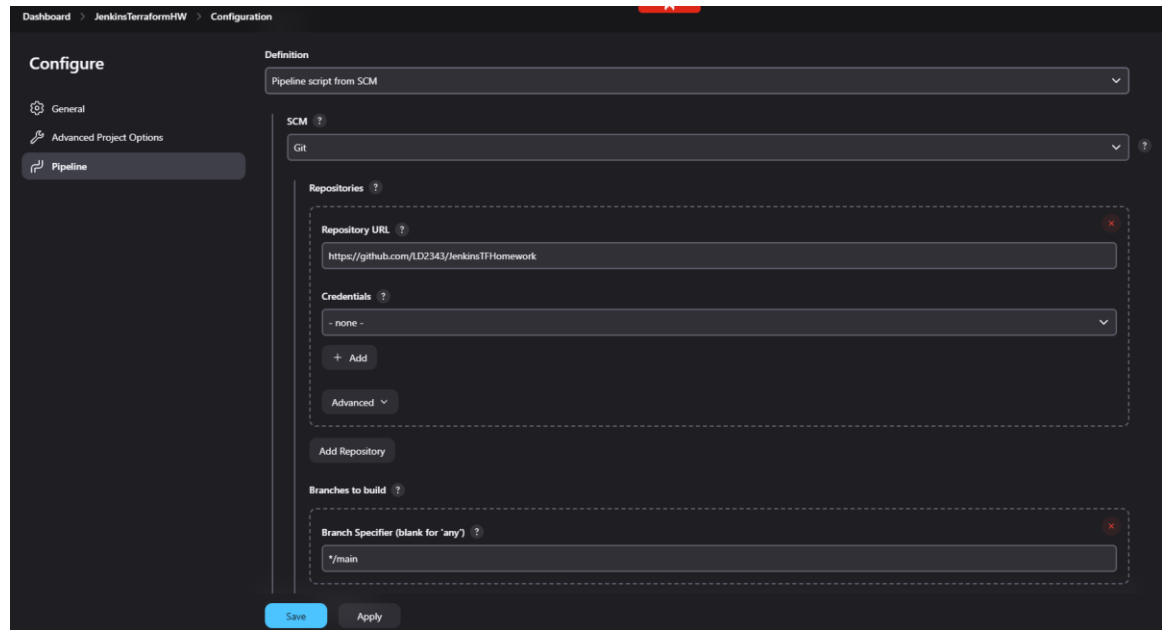
- b. Create a folder on pc to clone this empty repository to.
 - c. Add working Terraform project and Jenkins file to this folder.
 - d. In VS code update the Jenkins file with the correct
 - e. Perform Git commands to push the terraform project and Jenkins file to repository.




- f.
7. Create a new Pipeline in Jenkins and configure it to utilize our Github repository.
 - a. In Jenkins under Dashboard, click on New Item.
 - b. Name item and select Pipeline then click ok.
 - c. In the Pipeline section, for Definition select Pipeline script from SCM.
 - d. In SCM, select Git.
 - e. Paste the Github repository url and update the Branch Specifier to */main.
 - f. Click Save



g.



- h.
8. Run our new Pipeline in Jenkins to fire off the build of our Terraform project.
 - a. Inside the Pipeline, click on Build Now to run.
 - b. Click on the progress bar.
 - c. When prompted click on Deploy in the Console Output.
 - d. Wait for build to complete.
 - e. ...



Jenkins

Dashboard > JenkinsTerraformHW >

Status

Changes

Build Now

Configure

Delete Pipeline


Stages

Rename

Pipeline Syntax

JenkinsTerraformHW

Permalinks


Build History
trend

Filter...

#1

Feb 2, 2025, 5:14 PM

Atom feed for all
Atom feed for failures

f.

Dashboard > JenkinsTerraformHW > #1

```

resource "tls_private_key" "ToykoLinux" {
  algorithm = "RSA"
  ecdsa_curve = "P224"
  id = (known after apply)
  private_key_openssh = (sensitive value)
  private_key_pem = (sensitive value)
  private_key_pem_pkcs8 = (sensitive value)
  public_key_fingerprint_md5 = (known after apply)
  public_key_fingerprint_sha256 = (known after apply)
  public_key_openssh = (known after apply)
  public_key_pem = (known after apply)
  rsa_bits = 2048
}

[1mPlan:0m 56 to add, 0 to change, 0 to destroy.
[0m0[90m

Saved the plan to: tfplan

To perform exactly these actions, run the following command to apply:
  terraform apply "tfplan"

[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Apply Terraform)
[Pipeline] input
Approve Terraform Apply?
Deploy or Abort

```

g.

h. ...

- i. ...
 - j. ...
9. Confirm the Pipeline build was successful.
 - a. Verify the build was successful in Jenkins in the Console Output, Pipeline Overview and AWS.

```

Dashboard > JenkinsTerraformRW > #1

H[0m[[module.Wirginia.aws_ec2_transit_gateway_vpc_attachment.virginia_attachment: Still creating... [1m40s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Still creating... [2m20s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_ec2_transit_gateway_vpc_attachment.virginia_attachment: Creation complete after 1m41s [id=tgw-attach-0fe78c89f348175e]]]H[0m
H[0m[[module.tokyo.aws_ec2_transit_gateway_vpc_attachment.tokyovpc_attachment: Creation complete after 1m38s [id=tgw-attach-07bc172ed4de6523]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Still creating... [2m30s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Still creating... [2m40s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Still creating... [2m50s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Still creating... [3m00s elapsed]]]H[0m
H[0m[[module.Wirginia.aws_lb.virginia_alb: Creation complete after 3m0s [id=arn:aws:elasticloadbalancing:us-east-1:545009829599:loadbalancer/app/virginia-load-balancer/0675e572dae2b3e2]]]H[0m
H[0m[[module.Wirginia.aws_lb_listener.http: Creating...]]H[0m
H[0m[[module.Wirginia.aws_lb_listener.http: Creation complete after 1s [id=arn:aws:elasticloadbalancing:us-east-1:545009829599:listener/app/virginia-load-balancer/0675e572dae2b3e2/8ed4bc4c39948dc6]]]H[0m
H[0m[[main
Apply complete! Resources: 56 added, 0 changed, 0 destroyed.
H[0m
[Pipeline]
[Pipeline] // withCredentials
[Pipeline]
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { Declarative: Post Actions)
[Pipeline] echo
Terraform deployment completed successfully!
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

b.

c.

| Name | Resource ID | Resource Type | Region |
|---------------|---------------------------------------|---------------|----------------|
| tokyovpc | vpc-0b808bcceea66b0b0 | Vpc | ap-northeast-1 |
| nvirginiaavpc | vpc-020d7cded4bebecf6 | Vpc | us-east-1 |

d.

aws [Alt+S]

EC2 Global View

Region explorer

Global search

Settings

Region explorer

Global search

Updated 2 minutes ago

Global search (7)

Perform a global search to search for specific resources across all Regions for which your account is enabled

Find resources by attribute or tag

Resource Type = Instance

Clear filters

| Name | Resource ID | Resource Type | Region |
|--------------------|---------------------|---------------|----------------|
| nvirginia-instance | i-087cd8b802025aba3 | Instance | us-east-1 |
| nvirginia-instance | i-03add701c00581034 | Instance | us-east-1 |
| nvirginia-instance | i-05ff34bec16743192 | Instance | us-east-1 |
| tokyo-instance | i-04435315ff7e979f4 | Instance | ap-northeast-1 |
| tokyo-instance | i-0e05b63c7637bf33 | Instance | ap-northeast-1 |
| tokyo-instance | i-06d6d346c2242437e | Instance | ap-northeast-1 |
| tokyo-instance | i-0bb03432a1c55a54a | Instance | ap-northeast-1 |

e.

aws [Alt+S]

EC2 Global View

Region explorer

Global search

Settings

Region explorer

Global search

Updated 4 minutes ago

Global search (5)

Perform a global search to search for specific resources across all Regions for which your account is enabled

Find resources by attribute or tag

Resource Type = Security Group

Name : tokyo

Name : nvirginia

Clear filters

| Name | Resource ID | Resource Type | Region |
|---------------------|----------------------|----------------|----------------|
| TG01-SG01-nvirginia | sg-0c2e1b3d9697c44a3 | Security Group | us-east-1 |
| LB01-SG01-nvirginia | sg-0664a8676195f6afd | Security Group | us-east-1 |
| AuroraDB-SG-tokyo | sg-062f7e8595f4ac4d2 | Security Group | ap-northeast-1 |
| TG01-SG01-tokyo | sg-0d3a47734d40c94fe | Security Group | ap-northeast-1 |
| Syslog-SG-tokyo | sg-02652fa29bf1326e9 | Security Group | ap-northeast-1 |

f.

10. Perform Terraform teardown within our container in the CLI.

- In CLI inside our container, run to following command “cd var/jenkins_home/workspace/<PIPELINENAME>” to change directory to our pipeline.
- Run the following commands to input our AWS credentials and region in our container in order to run terraform destroy.
 - export AWS_ACCESS_KEY_ID="xxxxxxx"
 - export AWS_SECRET_ACCESS_KEY="xxxxxxx"
 - export AWS_REGION="xxxxxxx"
- Run the following command to teardown our terraform build.
 - terraform destroy

Plan: 0 to add, 0 to change, 56 to destroy.

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value:

d.

MINGW64/c/Users/logan

Plan: 0 to add, 0 to change, 8 to destroy.

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```
module.NVirginia.aws_autoscaling_group.nvirginia_asg: Destroying... [id=nvirginia-auto-scaling-group-20250202171723032100000003]
module.NVirginia.aws_autoscaling_group.nvirginia_asg: Destruction complete after 1s
module.NVirginia.aws_subnet.private-us-east-1c: Destroying... [id=subnet-01ad39ecc4d1e60cd]
module.NVirginia.aws_subnet.private-us-east-1a: Destroying... [id=subnet-0b7402f40121d3f0b]
module.NVirginia.aws_lb_target_group.nvirginia_tg: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:545009829599:targetgroup/nvirginia-
target-group/3aa163369f6f6b6d]
module.NVirginia.aws_subnet.private-us-east-1b: Destroying... [id=subnet-0e6778602602528cf]
module.NVirginia.aws_launch_template.nvirginia_lt: Destroying... [id=lt-0a941dabf7669b384]
module.NVirginia.aws_lb_target_group.nvirginia_tg: Destruction complete after 0s
module.NVirginia.aws_launch_template.nvirginia_lt: Destruction complete after 0s
module.NVirginia.aws_security_group.TG01-SG01-nvirginia: Destroying... [id=sg-0c2e1b3d9697c44a3]
module.NVirginia.aws_subnet.private-us-east-1b: Destruction complete after 0s
module.NVirginia.aws_subnet.private-us-east-1c: Destruction complete after 0s
module.NVirginia.aws_subnet.private-us-east-1a: Destruction complete after 0s
module.NVirginia.aws_security_group.TG01-SG01-nvirginia: Destruction complete after 1s
module.NVirginia.aws_vpc.nvirginia_vpc: Destroying... [id=vpc-020d7cded4bebecf6]
module.NVirginia.aws_vpc.nvirginia_vpc: Destruction complete after 0s
```

e.