

FYI Every New Service NEEDS its OWN KEYS!!!

Generated from # AWS 6.0 - Jenkins Terraform Pipeline 1/29/2025 YouTube Video

A Container is its OWN Machine!! It also needs the Software you are looking to operate on it installed as well.

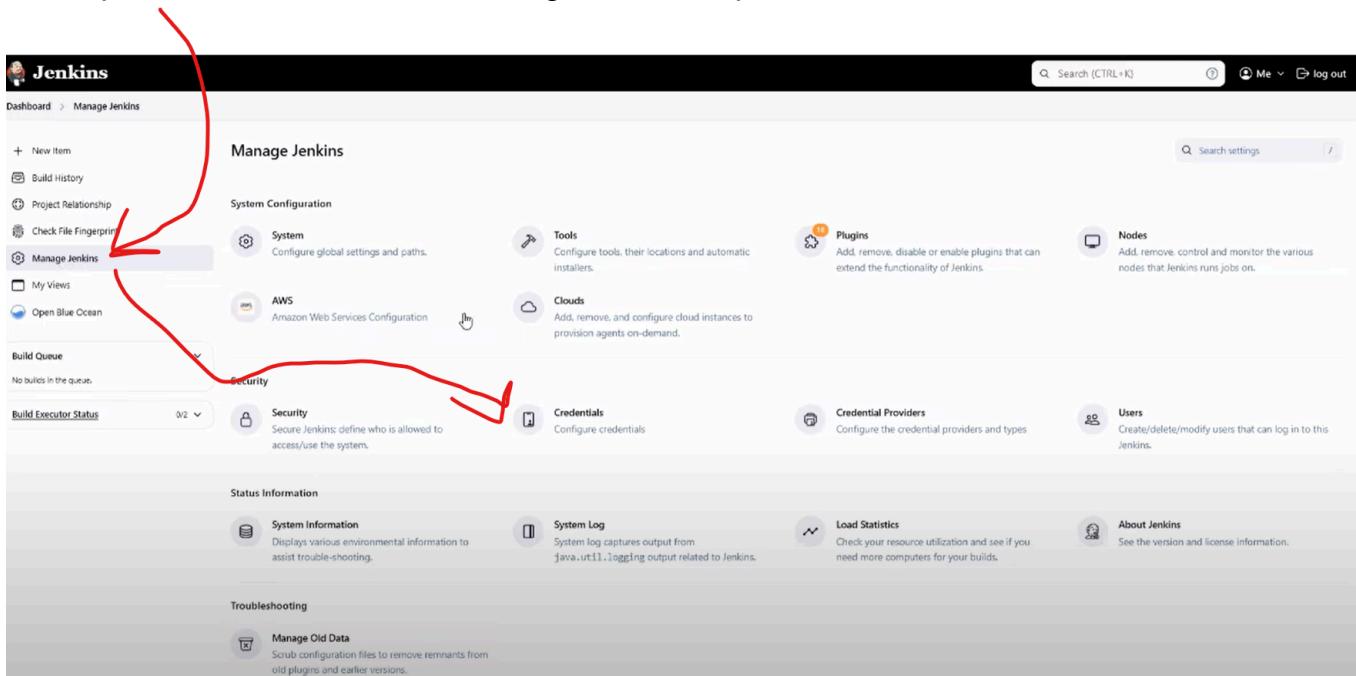
First I made a new IAM User account just for Jenkins to run through and saved the Access Key and Secret Access Key. These go into the "New Credentials" section further below.

~~this screenshot is just here so I don't lose it.~~

```
docker exec -it jenkins-terraform bash
docker exec -it --user root jenkins-terraform bash
jenkins@b013607bc1f9:/$ /etc/init.d/jenkins restart
docker commit <container_name> jenkins-terraform
apt update && apt install -y awscli
mkdir -p /home/jenkins/bin
curl -fsSL https://releases.hashicorp.com/terraform/1.5.7/terraform_1.5.7_linux_amd64.zip -o /home/jenkins/terraform.zip
unzip /home/jenkins/terraform.zip -d /home/jenkins/bin
rm /home/jenkins/terraform.zip
export PATH="/home/jenkins/bin:$PATH"
```

Once Jenkins has been installed, start it up within Docker Desktop. Be sure to Sign in to Docker (Desktop)

Start up Jenkins and select the "Manage Jenkins" option and select "Credentials"



Then click on "Global"

The screenshot shows the Jenkins 'Credentials' page under 'Manage Jenkins'. A red arrow points from the top navigation bar to the 'Stores scoped to Jenkins' section. Another red arrow points from the 'Domains' dropdown menu to the 'System' store entry.

T	P	Store	ID	Name
		System	(global)	AWS_SECRET_ACCESS_KEY

Stores scoped to Jenkins

P	Store	ID	Domains
	System	(global) ▾	

Icon: S M L

Select "Add Credentials"

The screenshot shows the Jenkins 'Global credentials (unrestricted)' page under 'Manage Jenkins > Credentials'. A red arrow points from the top navigation bar to the '+ Add Credentials' button. The page lists a single credential entry:

ID	Name	Kind	Description
AWS_SECRET_ACCESS_KEY	AKIASU566XGQQ5V6I6UP (AWS_SECRET_ACCESS_KEY)	AWS Credentials	AWS_SECRET_ACCESS_KEY

Icon: S M L

While in "New Credentials", select the following options,
 "Kind" = AWS Credentials
 "ID" = "Description" = can be what you want to name it.
 "Access Key ID" = Access key created for new IAM User account in AWS
 "Secret Access Key" = Secret Access key created for new IAM User account in AWS
 Then select Create.

The screenshot shows the Jenkins 'New credentials' page for 'AWS Credentials'. The form fields are filled as follows:

- Kind:** AWS Credentials
- Scope:** Global (Jenkins, nodes, items, all child items, etc.)
- ID:** AWS_SECRET_ACCESS_KEY
- Description:** AWS_SECRET_ACCESS_KEY
- Access Key ID:** (empty field)
- Secret Access Key:** (empty field)
- IAM Role Support:** Advanced

Create

Then go to "+ New Item"

The screenshot shows the Jenkins Manage Jenkins interface. On the left sidebar, there is a button labeled "+ New Item" with a red arrow pointing to it. The main area is titled "Manage Jenkins" and contains several configuration sections: "System Configuration" (with "System" and "AWS" sub-options), "Tools" (with "Clouds" sub-option), "Security" (with "Security" and "Credentials" sub-options), and "Status Information".

Once within, give your new pipeline a title and select "Pipeline" from the available options and select "Okay".

The screenshot shows the Jenkins "New Item" creation page. At the top, there is a search bar and a user profile icon. The main heading is "New Item". Below it, there is a field "Enter an item name" containing "vladPipeline1". Under "Select an item type", there are four options: "Freestyle project", "Pipeline" (which is highlighted and has a detailed description), "Multi-configuration project", and "Folder".

Once in the "Configure - General" section,

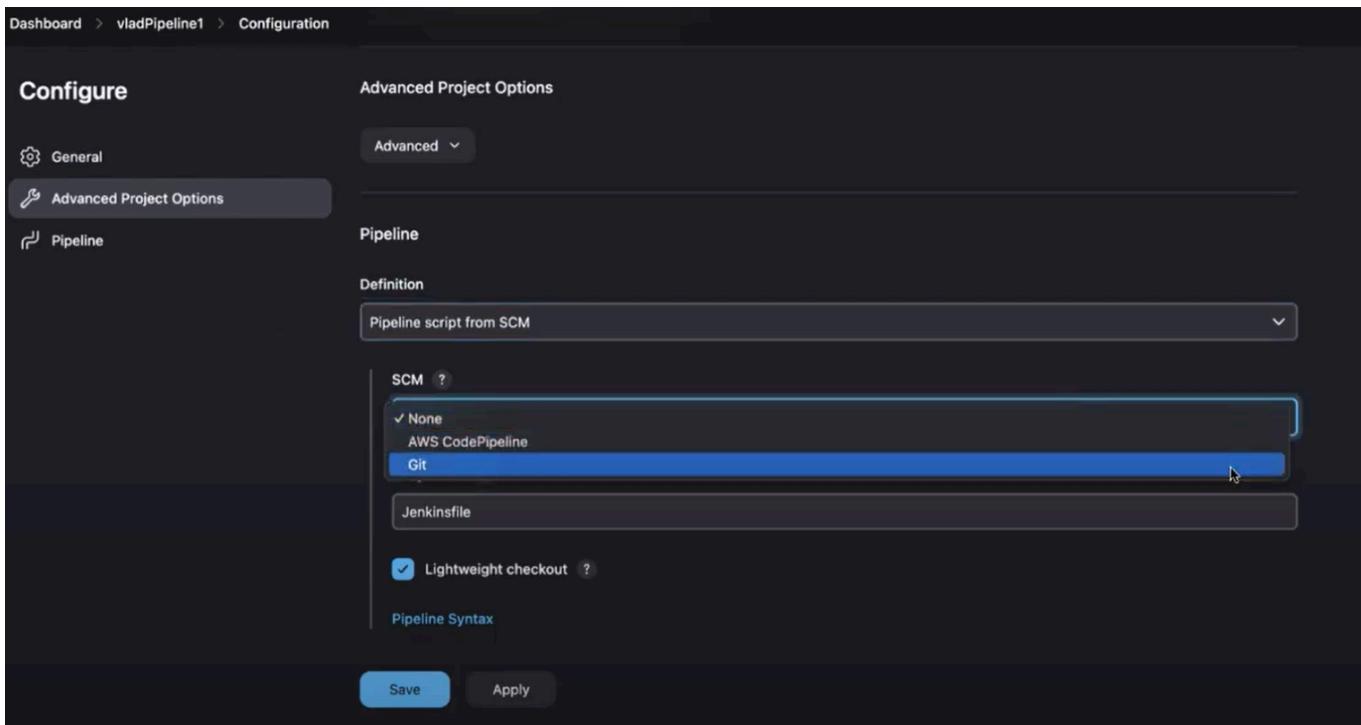
The screenshot shows the Jenkins configuration interface for a pipeline named 'vladPipeline1'. The 'General' tab is selected. At the top right, there is an 'Enabled' toggle switch which is turned on. Below it is a large, empty text area for 'Description'. Underneath this are several configuration options, each with a checkbox and a question mark icon for help:

- Discard old builds
- Do not allow concurrent builds
- Do not allow the pipeline to resume if the controller restarts
- GitHub project
- Pipeline speed/durability override
- Preserve stashes from completed builds
- This project is parameterized

scroll down to "Advanced Pipeline Options" and select the drop down for "Definition" and select "Pipeline script from SCM"

The screenshot shows the Jenkins configuration interface for the 'Pipeline' section. The 'Definition' dropdown is currently set to 'Pipeline script from SCM', which is highlighted with a blue selection bar. Below this, there is a 'Script' editor window containing a single line of code: '1'. To the right of the editor is a 'try sample Pipeline...' button. At the bottom of the editor, there is a checked checkbox for 'Use Groovy Sandbox'. Below the editor, there is a 'Pipeline Syntax' section. At the very bottom of the screen are 'Save' and 'Apply' buttons.

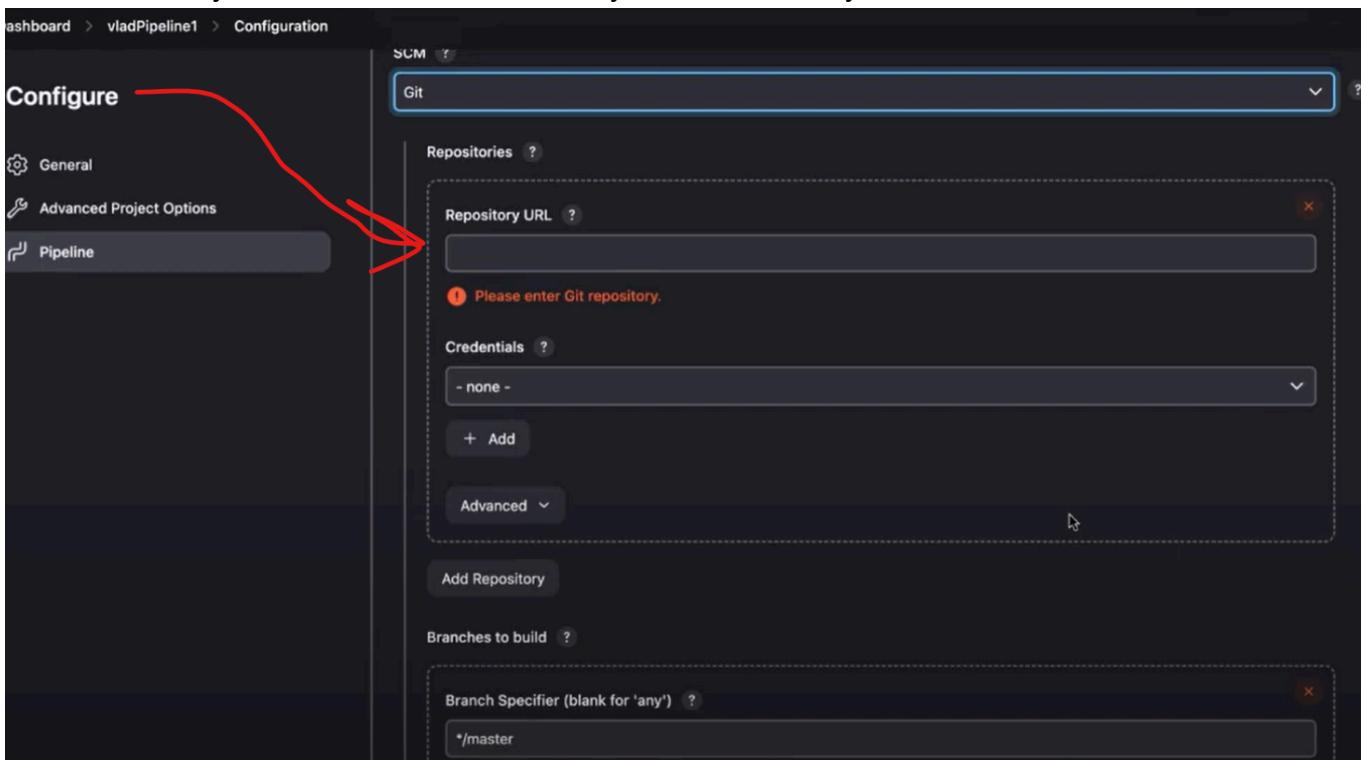
Then select from the "SCM" dropdown, "GIT"



Make sure your "Script Path" matches the file name you are actually using within your Repo for the Jenkins Pipeline.

After that, pull the Repository URL from GitHub you want to use for the Pipeline.

To make it easy, use a terraform code that you know Already WORKS!!



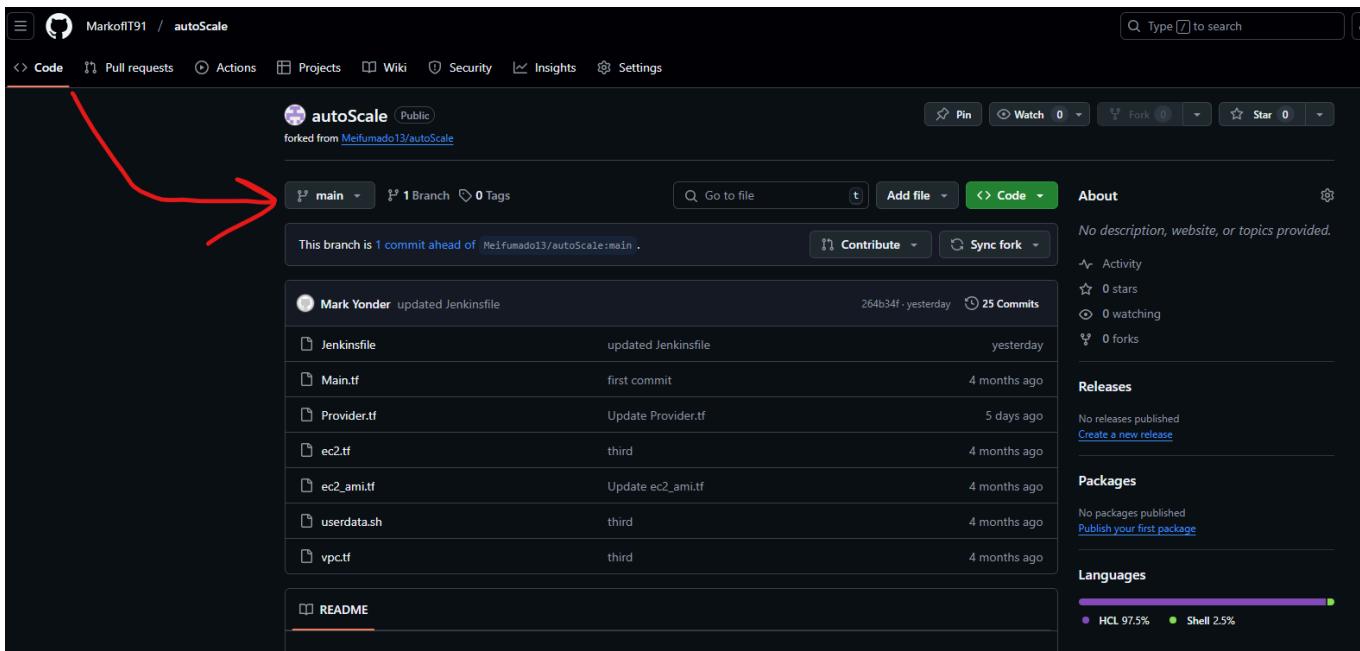
I used one from another groups project

A screenshot of a GitHub repository page for 'autoScale'. The repository is public and has 1 branch and 0 tags. The main branch is ahead by 1 commit. The commit history shows a single commit from 'Mark Yonder' updating 'Jenkinsfile'. The repository URL is listed as <https://github.com/MarkoffT91/autoScale.git>. A red arrow points from the URL field to the 'Branch Specifier' input field in the Jenkins configuration.

Back in Jenkins, under the "Branches to Build" and in its sub category "Branch Specifier", set it as "/main", Then select "Save".

A screenshot of the Jenkins Pipeline configuration screen for a job named 'test 2'. Under the 'Pipeline' tab, the SCM section is set to 'Git'. The 'Repository URL' is set to <https://github.com/derrickSh43/autoScale>. In the 'Branches to build' section, the 'Branch Specifier' field contains the value '/main'. A red arrow points from the 'Branch Specifier' field in the Jenkins interface to the corresponding field in the GitHub repository URL.

This is determined by the branch of GitHub you pulled the Repo URL from.



Verify Pipeline creation within the Dashboard.

The Jenkins file is similar to the YAML file in such that it is a staging File for the Pipeline. Gives the Jenkins Pipeline its Orders.

Be sure to update the following values within the Jenkins File with your own unique information,

```
pipeline {
    agent any
    environment {
        AWS_REGION = 'us-east-1'
    }
    stages {
        stage('Set AWS Credentials') {
            steps {
                withCredentials([
                    $class: 'AmazonWebServicesCredentialsBinding',
                    credentialsId: 'JenkinsUser'
                ]) {
                    sh ""
                    echo "AWS_ACCESS_KEY_ID: ${AWS_ACCESS_KEY_ID} aws sts get-caller-identity"
                    stage('Checkout Code') {
                        steps {
                            git branch: 'main'
                        }
                    }
                    AWS_ACCESS_KEY_ID
                    export AWS_SECRET_ACCESS_KEY=${AWS_SECRET_ACCESS_KEY}
                    stage('Apply Terraform') {
                        steps {
                            terraform plan -out=tfplan
                            terraform apply -auto-approve tfplan
                        }
                    }
                }
            }
        }
    }
}
```

```

}
}
}
}

post {
success {
echo 'Terraform deployment completed successfully!'
}
failure {
echo 'Terraform deployment failed!'
}
}
}
}
}

```

"CredentialsID" - pulled from your Jenkins Global credentials (unrestricted) menu.

Jenkins

Search (CTRL+K)

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

ID	Name	Kind	Description
JenkinsUser	AKIAZDZTBTYVOKGZFL5J (JenkinsUser)	AWS Credentials	JenkinsUser

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
JenkinsUser	AKIAZDZTBTYVOKGZFL5J (JenkinsUser)	AWS Credentials	JenkinsUser

"AWS_REGION" - pulled from AWS Region and AZ you're currently operating your Jenkins IAM User account from.

aws | Search [Alt+S]

Console Home Info

Recently visited

- Billing and Cost Management
- EC2
- S3
- IAM
- CloudTrail
- Route 53
- CloudFront
- Service Quotas

AWS Organizations

EC2 Global View

Amazon Bedrock

Applications (0) Info

Region: US East (N. Virginia)

us-east-1 (Current Region) Find applications

No applications

Create application

"git branch: 'main', url" - pulled from Github

The screenshot shows a GitHub repository page for 'autoScale'. The top navigation bar includes links for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main content area displays the repository's code structure, showing files like Jenkinsfile, Main.tf, Provider.tf, ec2.tf, ec2_ami.tf, userdata.sh, and vpc.tf. On the right side, there are sections for About, Releases, Packages, and Languages. The 'About' section notes 'No description, website, or topics provided.' and shows 0 stars, 0 forks, and 0 watching. The 'Languages' section indicates HCL at 97.5% and Shell at 2.5%.

Run "git clone (insert repo URL pulled from GitHub here)" to copy the repo URL from GitHub to the folder you will be accessing the file from within your command line tool, eg. GitBash, Powershell etc.

```
mjnic@LAPTOP-VQ5RN68H MINGW64 ~/desktop
$ git clone https://github.com/MarkofIT91/autoScale.git
Cloning into 'autoScale'...
remote: Enumerating objects: 93, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 93 (delta 35), reused 29 (delta 29), pack-reused 55 (from 1)
Receiving objects: 100% (93/93), 5.14 MiB | 22.32 MiB/s, done.
Resolving deltas: 100% (42/42), done.

mjnic@LAPTOP-VQ5RN68H MINGW64 ~/desktop
$
```

Install Terraform and AWS through the Docker CLI Terminal.

The screenshot shows the Docker Desktop application on a Windows system. The left sidebar contains links for Containers, Images, Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area shows a container named 'laughing_keldysh' running the Jenkins/Jenkins:its-jdk11 image. The container status is 'Running (3 hours ago)'. Below the container list is a terminal window titled 'Terminal' showing a PowerShell session. The terminal output includes Jenkins installation logs and a prompt 'PS C:\Users\mjnic>'. At the bottom of the screen, a status bar shows 'Engine running', RAM usage (1.79 GB), CPU usage (0.00%), and Disk usage (6.30 GB used / limit 1006.85 GB). The Docker version is v4.37.1.

Run "docker ps" to check for running Containers. The first three characters of the Container ID are enough to identify it when running some code.

Install AWS CLI and Terraform into Docker Container Image

In GitBash, Powershell, Docker Terminal (Windows Powershell) or some other command line, Run "docker exec -it --user root (your Container name here) bash" in order to open and operate and run code from the Docker command line

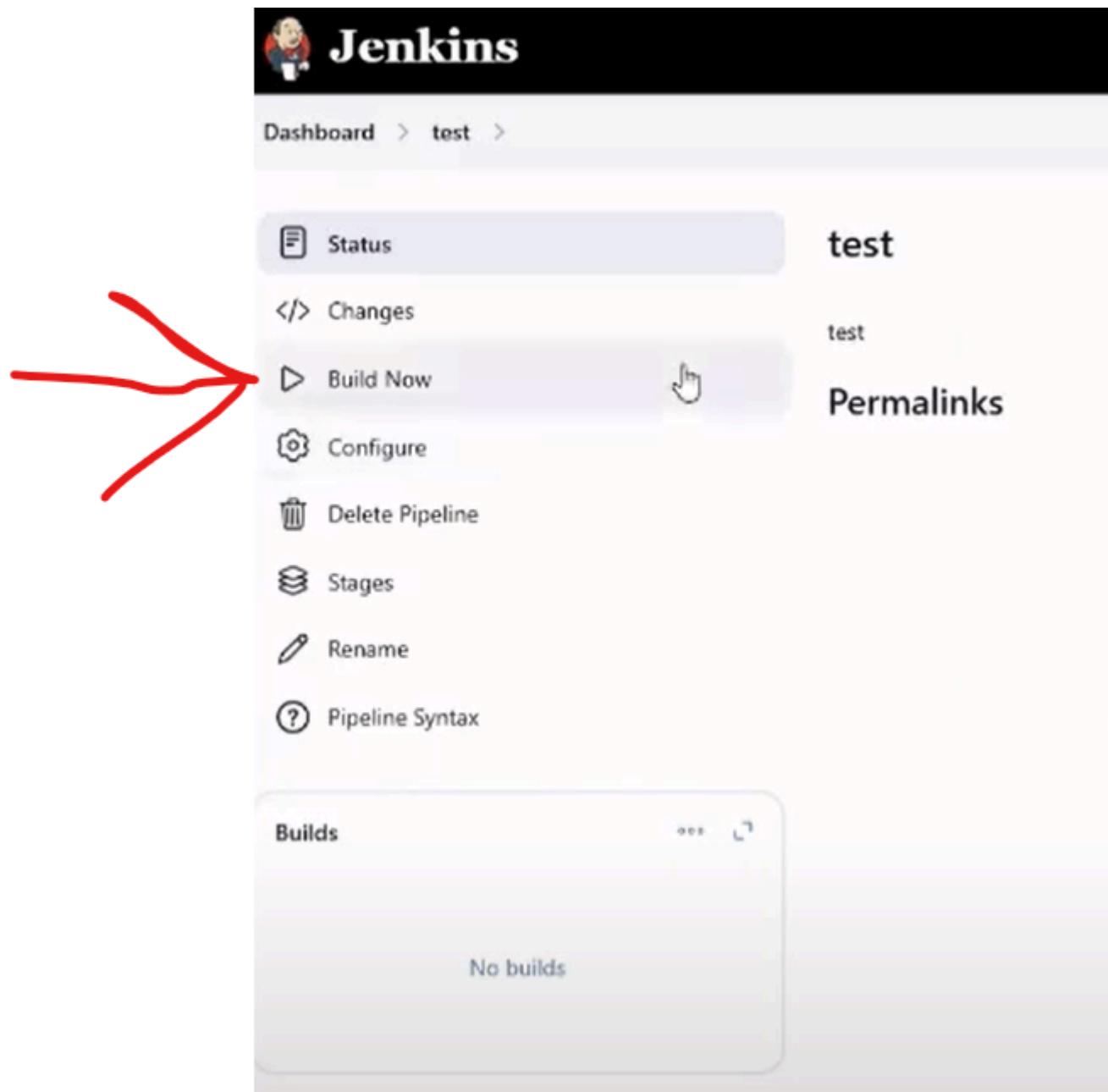
Run "apt update && apt install -y awscli" to install or update AWS CLI within the Docker Container.

Run the following line one after another or all at once to install Terraform in the docker container if you haven't already,

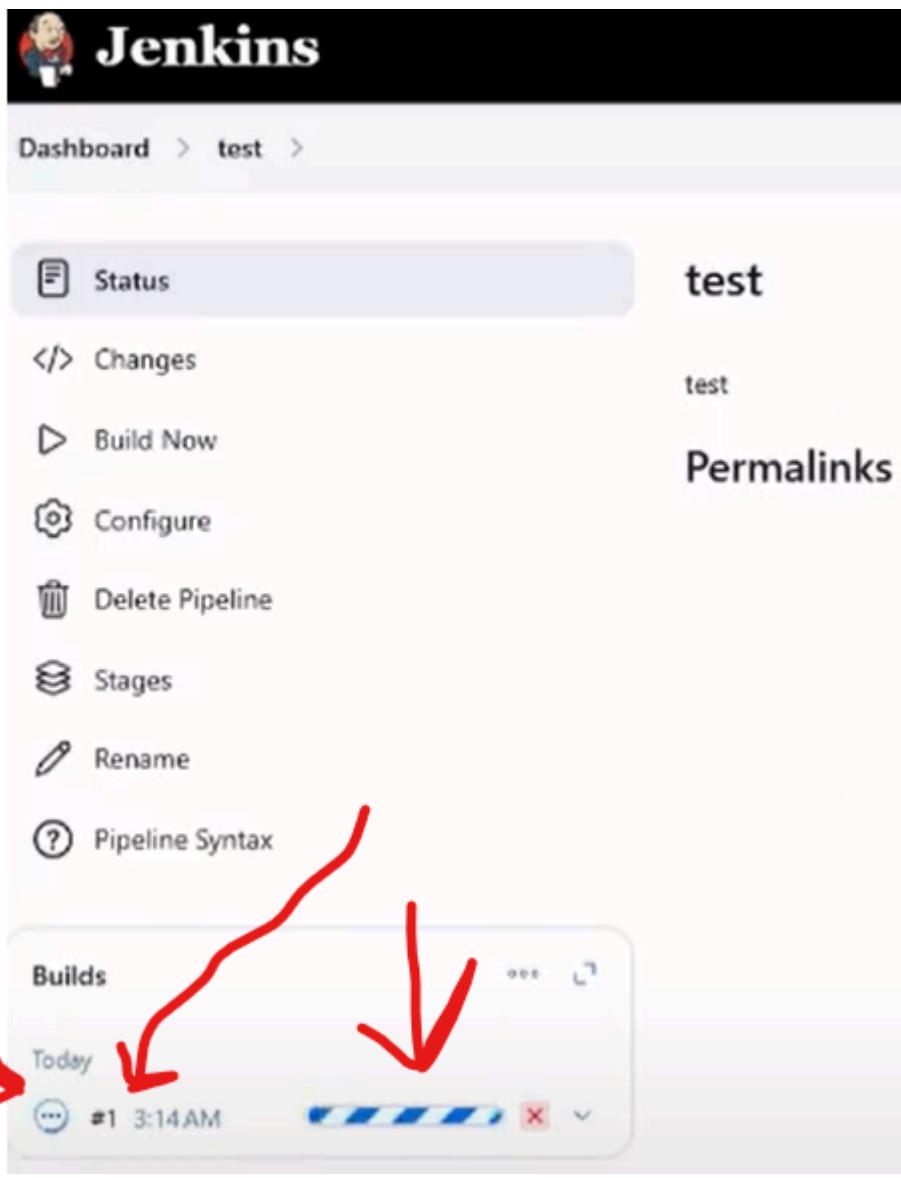
```
mkdir -p /home/jenkins/bin  
curl -fsSL https://releases.hashicorp.com/terraform/1.5.7/terraform_1.5.7_linux_amd64.zip -o /home/jenkins/terraform.zip  
unzip /home/jenkins/terraform.zip -d /home/jenkins/bin  
rm /home/jenkins/terraform.zip  
export PATH="/home/jenkins/bin:$PATH"
```

Run "terraform --version" to verify Terraform Installation.

Next go back to your Jenkins "Dashboard" and go into your created Pipeline and select "Build Now"

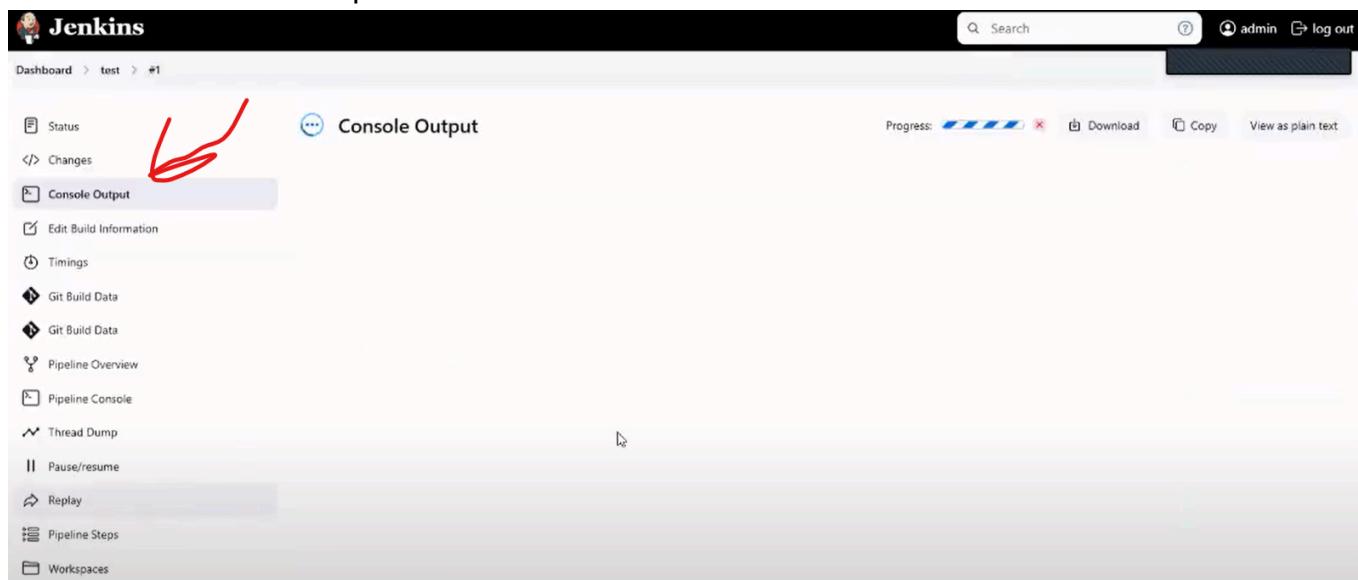


As the Pipeline is starting up select the Build that is being worked,



The screenshot shows the Jenkins Pipeline test status page. On the left, there's a sidebar with various options: Status, Changes, Build Now, Configure, Delete Pipeline, Stages, Rename, and Pipeline Syntax. Below these is a 'Builds' section for today, showing a single build labeled '#1 3:14 AM'. A red arrow points from the 'Pipeline Syntax' link in the sidebar down to the '#1' build entry. Another red arrow points from the 'Builds' section down to the '#1' build entry.

and select "Console Output"



The screenshot shows the Jenkins Pipeline test build #1 console output page. The top navigation bar includes links for Status, Changes, and Console Output, with 'Console Output' being the active tab. The main content area displays the build log, which is currently empty. There are download and copy buttons at the top right of the log area. A red arrow points from the 'Console Output' link in the sidebar down to the log area.

If it runs correctly you should see something like this,

Dashboard > JenkinsPipePause > #10

Status  **Console Output** Download Copy View as plain text

</> Changes

 **Console Output**

 Edit Build Information

 Delete build '#10'

 Timings

 Git Build Data

 Pipeline Overview

 Pipeline Console

 Replay

 Pipeline Steps

 Workspaces

 Previous Build

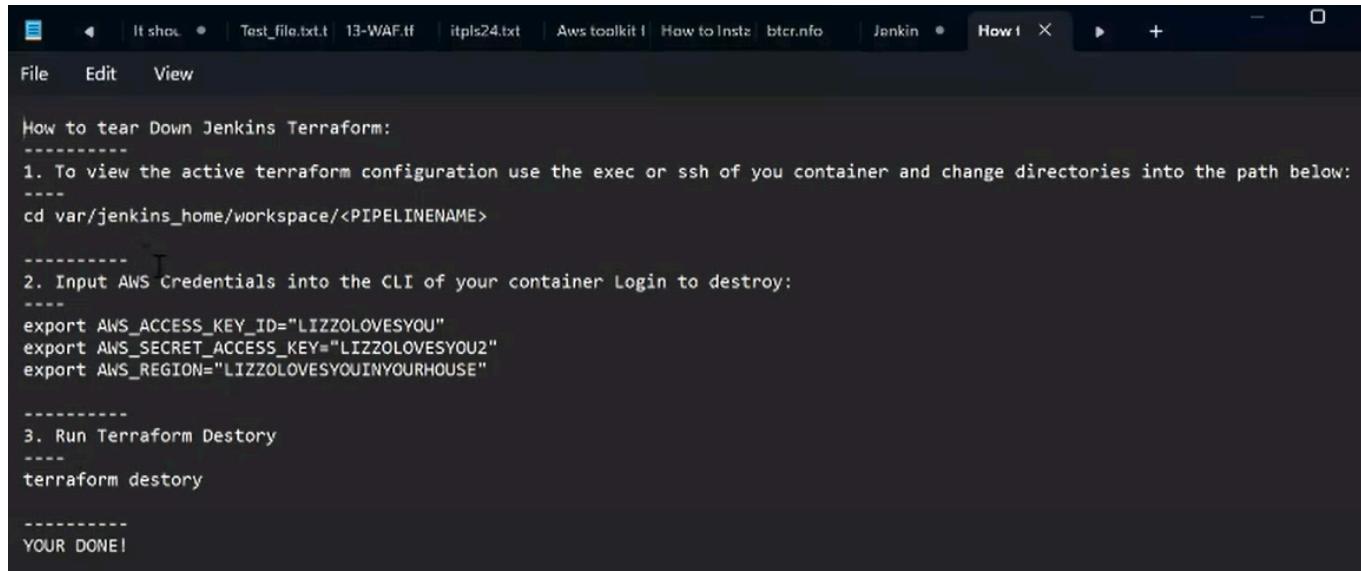
Started by user **MarkOfIT91**
Obtained Jenkinsfile from git <https://github.com/MarkofIT91/autoScale.git>
[Pipeline] Start of Pipeline
[Pipeline] node
Running on **Jenkins** in /var/jenkins_home/workspace/JenkinsPipePause
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/JenkinsPipePause/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url <https://github.com/MarkofIT91/autoScale.git> # timeout=10
Fetching upstream changes from <https://github.com/MarkofIT91/autoScale.git>
> git --version # timeout=10
> git -version # 'git version 2.39.5'
> git fetch --tags --force --progress -- <https://github.com/MarkofIT91/autoScale.git> +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 264b34f82879d7db9317b6a5273594fcda11ad3 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 264b34f82879d7db9317b6a5273594fcda11ad3 # timeout=10
Commit message: "updated Jenkinsfile"
> git rev-list --no-walk 49d376cd23c48e3046ece5a2ed4beabae0d1954d # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {

[0m@1mnull_resource.delete_instances: Creation complete after 1s [id=5041131486270232147]@0m
[0m@1maws_launch_template.ec2_launch_template: Creation complete after 5s [id=lt-012b5fbcae5bb48b1]@0m
[0m@1maws_autoscaling_group.ec2_asg: Creating...@0m@0m
[0m@1maws_autoscaling_group.ec2_asg: Still creating... [10s elapsed]@0m@0m
[0m@1maws_autoscaling_group.ec2_asg: Still creating... [20s elapsed]@0m@0m
[0m@1maws_autoscaling_group.ec2_asg: Creation complete after 26s [id=custom-web-server-asg]@0m
[0m@1m@32m
Apply complete! Resources: 28 added, 0 changed, 0 destroyed.
[0m@0m@1m@32m
Outputs:

[0minstance_id = "i-06f813698a6c5af4d4"
[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

Should it Fail, Time to Troubleshoot!!

The process for Tear Down is as follows,



File Edit View

```
How to tear Down Jenkins Terraform:
-----
1. To view the active terraform configuration use the exec or ssh of you container and change directories into the path below:
-----
cd var/jenkins_home/workspace/<PIPELINENAME>

-----
2. Input AWS Credentials into the CLI of your container Login to destroy:
-----
export AWS_ACCESS_KEY_ID="LIZZOLOVESYOU"
export AWS_SECRET_ACCESS_KEY="LIZZOLOVESYOU2"
export AWS_REGION="LIZZOLOVESYOUINYOURHOUSE"

-----
3. Run Terraform Destory
-----
terraform destroy

-----
YOUR DONE!
```

Enter your Docker Jenkins Pipeline by running "docker exec -it --user root (first 3 numbers of your Container ID) bash".

"cd" into your Jenkins Pipeline by running "cd \$JENKINS_HOME/workspace/(your Pipeline name here)"

```
mjnic@LAPTOP-VQ5RN68H MINGW64 ~/desktop
$ docker exec -it --user root 975 bash
root@9754673f8263:/# cd $JENKINS_HOME/workspace/JenkinsPipePause
root@9754673f8263:/var/jenkins_home/workspace/JenkinsPipePause# ...
```

run the following commands with your AWS Access key Information,

```
export AWS_ACCESS_KEY_ID="(AWS Access Key used for Your Jenkins Account)"
export AWS_SECRET_ACCESS_KEY="(AWS Secret Access Key used for Your Jenkins
Account)"
```

Then run, "terraform destroy"

```
jnicPLATOP-MQ5RN68H MINGW64 ~/OneDrive/Desktop/Jenkins/class6/autoscale (main)
$ docker exec -it --user root 975 bash
root@9754673f8263:/# cd $JENKINS_HOME/workspace/JenkinsPipePause
root@9754673f8263:/var/jenkins_home/workspace/JenkinsPipePause# export AWS_ACCESS_KEY_ID="AKIAZDZTBTYVOKG2FLS3"
export AWS_SECRET_ACCESS_KEY="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"
root@9754673f8263:/var/jenkins_home/workspace/JenkinsPipePause# terraform destroy
aws_vpc.custom_vpc: Refreshing state... [id=vpc-098b4422dea8c5ce8]
aws_subnet.private_subnet[1]: Refreshing state... [id=subnet-06300d8356f640224]
aws_subnet.public_subnet[0]: Refreshing state... [id=subnet-0e79370845d0b11f8]
aws_internet_gateway.internet_gateway: Refreshing state... [id=igw-0934ac800a52a82c7]
aws_subnet.private_subnet[0]: Refreshing state... [id=subnet-09046317c67b59788]
aws_subnet.private_subnet[2]: Refreshing state... [id=subnet-07e704b53525a9bbb]
aws_subnet.public_subnet[1]: Refreshing state... [id=subnet-052bdb891b5349918]
aws_security_group.alb_sg: Refreshing state... [id=sg-0b6f03c2c26591de8]
aws_subnet.public_subnet[2]: Refreshing state... [id=subnet-0431adb987201915e]
aws_lb_target_group.alb_ec2_tg: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:targetgroup/custom-web-server-tg/5c45ac8416074472]
aws_route_table.custom_route_table_public_subnet: Refreshing state... [id=rtb-01887c6577f465b96]
aws_eip.eip: Refreshing state... [id=eipalloc-03f7377ca328e0f8c]
aws_security_group.ec2_sg: Refreshing state... [id=sg-0fdeac63d7dab66d]
aws_lb.app_lb: Refreshing state... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:loadbalancer/app/custom-app-1b/cc24c450f94b815f]
aws_instance.example_instance: Refreshing state... [id=i-06f813698a6c5af04]
aws_route_table_association.public_subnet_association[1]: Refreshing state... [id=rtaassoc-01df5fa32705e16c]
aws_route_table_association.public_subnet_association[2]: Refreshing state... [id=rtaassoc-0e88bbcf1c56ea6e1]
aws_route_table_association.public_subnet_association[3]: Refreshing state... [id=rtaassoc-0f135333a256e16c]
```

```

aws_rpm_gateway_accel_ipv4_for_iprock = {
  - cidr_block = "10.230.0.0/16"
  - default_network_acl_id = "acl-0a917671d9de87bf6"
  - default_route_table_id = "rtb-dd0fefa2307839b20"
  - default_security_group_id = "sg-0f6061336272cde18"
  - dhcp_options_id = "dopt-0f94f40f1f475e7d3"
  - enable_dns_hostnames = true
  - enable_dns_support = true
  - enable_network_address_usage_metrics = false
  - id = "vpc-098b4422dea8c5ce8"
  - instance_tenancy = "default"
  - ipv6_netmask_length = 0
  - main_route_table_id = "rtb-0d0fefa2307839b20"
  - owner_id = "626635415082"
  - tags = [
    - "Name" = "Custom VPC"
  ]
  - tags_all = [
    - "Name" = "Custom VPC"
  ]
}

# null_resource.delete_instances will be destroyed
resource "null_resource" "delete_instances" {
  - id = "5041131486270232147"
  - triggers = [
    - "instance_id" = "i-06f813698a6c5af4"
  ]
}

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

null_resource.delete_instances: Destroying... [id=5041131486270232147]
null_resource.delete_instances: Destruction complete after 0s
ws_route_table_association.public_subnet_association[1]: Destroying... [id=rtbassoc-01df5fa32705e16c]
ws_route_table_association.private_subnet_association[2]: Destroying... [id=rtbassoc-02ab427c6c8fsec93]
ws_route_table_association.private_subnet_association[0]: Destroying... [id=rtbassoc-09e00e94ccb0884e9]
ws_route_table_association.private_subnet_association[1]: Destroying... [id=rtbassoc-06669770e9ee017b0]
ws_route_table_association.public_subnet_association[2]: Destroying... [id=rtbassoc-0e89bbcf1d56e6e1]
ws_route_table_association.public_subnet_association[0]: Destroying... [id=rtbassoc-0a343898296a3279c]
ws_lb_listener.alb_listener: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:listener/app/custom-app-1b/cc24c450f94b815f/8ed13537661722a]
ws_autoscaling_group.ec2_asg: Destroying... [id=custom-web-server-asg]
ws_route_table_association.private_subnet_association[0]: Destruction complete after 1s
ws_route_table_association.private_subnet_association[1]: Destruction complete after 1s
ws_route_table_association.private_subnet_association[2]: Destruction complete after 1s
ws_route_table_association.public_subnet_association[1]: Destruction complete after 1s
ws_route_table.custom_route_table_private_subnet: Destroying... [id=rtb-06f0de5a052556dc3]
ws_route_table_association.public_subnet_association[2]: Destruction complete after 1s
ws_route_table_association.public_subnet_association[0]: Destruction complete after 1s
ws_route_table.custom_route_table_public_subnet: Destroying... [id=rtb-01887c6577f465b96]
ws_lb_listener.alb_listener: Destruction complete after 1s
ws_lb.app_lb: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:loadbalancer/app/custom-app-1b/cc24c450f94b815f]
ws_route_table.custom_route_table_private_subnet: Destruction complete after 0s
ws_nat_gateway.custom_nat_gateway: Destroying... [id=nat-020b1fbcd3e604d5]
ws_route_table.custom_route_table_public_subnet: Destruction complete after 0s
ws_lb.app_lb: Destruction complete after 1s
ws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 10s elapsed]
ws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-020b1fbcd3e604d5, 10s elapsed]

```

```

aws_autoscaling_group.ec2_asg: Destroying... [id=custom-web-server-asg]
aws_route_table_association.private_subnet_association[0]: Destruction complete after 1s
aws_route_table_association.private_subnet_association[1]: Destruction complete after 1s
aws_route_table_association.private_subnet_association[2]: Destruction complete after 1s
aws_route_table_association.public_subnet_association[1]: Destruction complete after 1s
aws_route_table_association.public_subnet_association[2]: Destroying... [id=rtb-0ef0de5a05255edc3]
aws_route_table_association.public_subnet_association[2]: Destruction complete after 1s
aws_route_table_association.public_subnet_association[0]: Destruction complete after 1s
aws_route_table_association.custom_route_table_public_subnet: Destroying... [id=rtb-01887c6577f465b96]
aws_lb_listener.alb_listener: Destruction complete after 1s
aws_lb_app_lb: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:loadbalancer/app/custom-app-1b/cc24c450f94b815f]
aws_route_table_association.custom_route_table_private_subnet: Destruction complete after 0s
aws_nat_gateway.custom_nat_gateway: Destroying... [id=nat-02cb1fbcd3e604d5]
aws_route_table_association.custom_route_table_public_subnet: Destruction complete after 0s
aws_lb_app_lb: Destruction complete after 1s
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 10s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 10s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 20s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 20s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 30s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 30s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 40s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 40s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 50s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 50s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m5s elapsed]
aws_nat_gateway.custom_nat_gateway: Still destroying... [id=nat-02cb1fbcd3e604d5, 1m5s elapsed]
aws_nat_gateway.custom_nat_gateway: Destruction complete after 1m1s
aws_elb_elb: Destroying... [id=elb-alloc-03f7377ca328efcc]
aws_elb_elb: Destruction complete after 1s
aws_internet_gateway.internet_gateway: Destroying... [id=igw-0934ac80ca52a82c?]
aws_internet_gateway.internet_gateway: Destruction complete after 1s
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m10s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m20s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m30s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m40s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 1m50s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 2m0s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 2m10s elapsed]
aws_autoscaling_group.ec2_asg: Still destroying... [id=custom-web-server-asg, 2m20s elapsed]
aws_autoscaling_group.ec2_asg: Destruction complete after 2m25s
aws_subnet.private_subnet[0]: Destroying... [id=subnet-0904631c67b5978?]
aws_subnet.private_subnet[2]: Destroying... [id=subnet-07e704b53525a9bb?]
aws_subnet.private_subnet[1]: Destroying... [id=subnet-06300c8356f640224]
aws_lb_target_group.alb_ec2_tg: Destroying... [id=arn:aws:elasticloadbalancing:us-east-1:626635415082:targetgroup/custom-web-server-tg/5c45ac8416074472]

aws_launch_template.ec2_launch_template: Destroying... [id=lt-012b5fbcae5bb48b1]
aws_launch_template.ec2_launch_template: Destruction complete after 0s
aws_ami_from_instance.example_ami: Destroying... [id=ami-0479f4eacc4fab75a]
aws_security_group.ec2_sg: Destroying... [id=sq-0fde31c03c7dab66d]
aws_lb_target_group.alb_ec2_tg: Destruction complete after 0s
aws_subnet.private_subnet[1]: Destruction complete after 0s
aws_subnet.private_subnet[0]: Destruction complete after 0s
aws_subnet.private_subnet[2]: Destruction complete after 0s
aws_security_group.ec2_sg: Destruction complete after 1s
aws_security_group.alb_sg: Destroying... [id=sq-0b6f03c2c26591dd8]
aws_security_group.alb_sg: Destruction complete after 0s
aws_ami_from_instance.example_ami: Destruction complete after 6s
aws_subnet.public_subnet[1]: Destroying... [id=subnet-052bdb891b5349908]
aws_subnet.public_subnet[2]: Destroying... [id=subnet-0431ad987201915e]
aws_subnet.public_subnet[0]: Destroying... [id=subnet-0c79370845d0b11f8]
aws_subnet.public_subnet[0]: Destruction complete after 0s
aws_subnet.public_subnet[1]: Destruction complete after 0s
aws_subnet.public_subnet[2]: Destruction complete after 0s
aws_vpc.custom_vpc: Destroying... [id=vpc-098b4422dea8c5ce8]
aws_vpc.custom_vpc: Destruction complete after 1s

Destroy complete! Resources: 27 destroyed.

```

root@9754673F8263:/var/jenkins_home/workspace/JenkinsPipePause|