Jenkins Pipeline AWS build 2/1/25

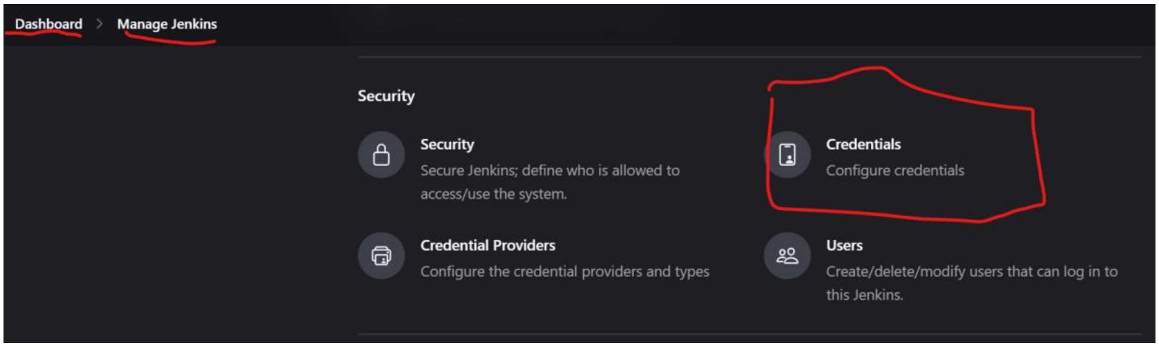
1) Create a new user IAM user in AWS console. Give that user admin and CLI rights.

2) Create AWS Access Key and Secret Key to be used in Jenkins for access to AWS

3) Start Jenkins container in docker desktop

4) Select port 8080:808 to open Jenkins

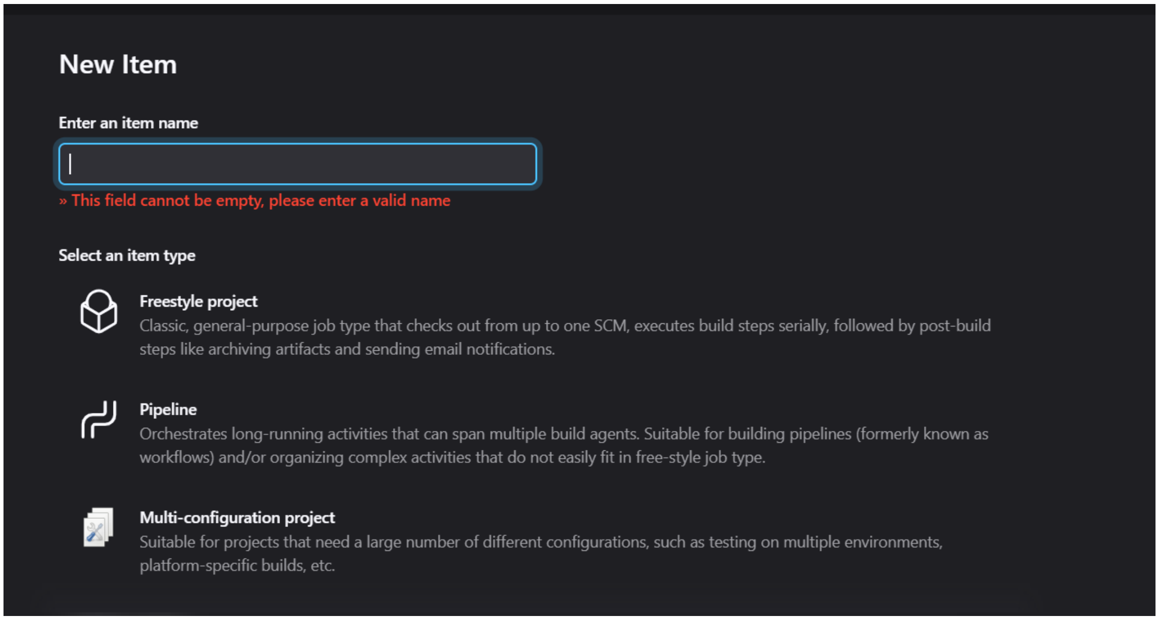
5) Login to Jenkins. Create credentials by navigating from the dashboard, manage Jenkins and then select credntials


Dashboard
Manage Jenkins
Security
Security
Secure Jenkins; define who is allowed to access/use the system.
Credentials
Configure credentials
Credential Providers
Configure the credential providers and types
Users
Create/delete/modify users that can log in to this Jenkins.

6) Add AWS Access and Secret Key to Jenkins under Global domain

7) Go Back to the Jennkins Dashboard and Select New item

8) Name the New Item and select Pipeline



New Item

Enter an item name

» This field cannot be empty, please enter a valid name

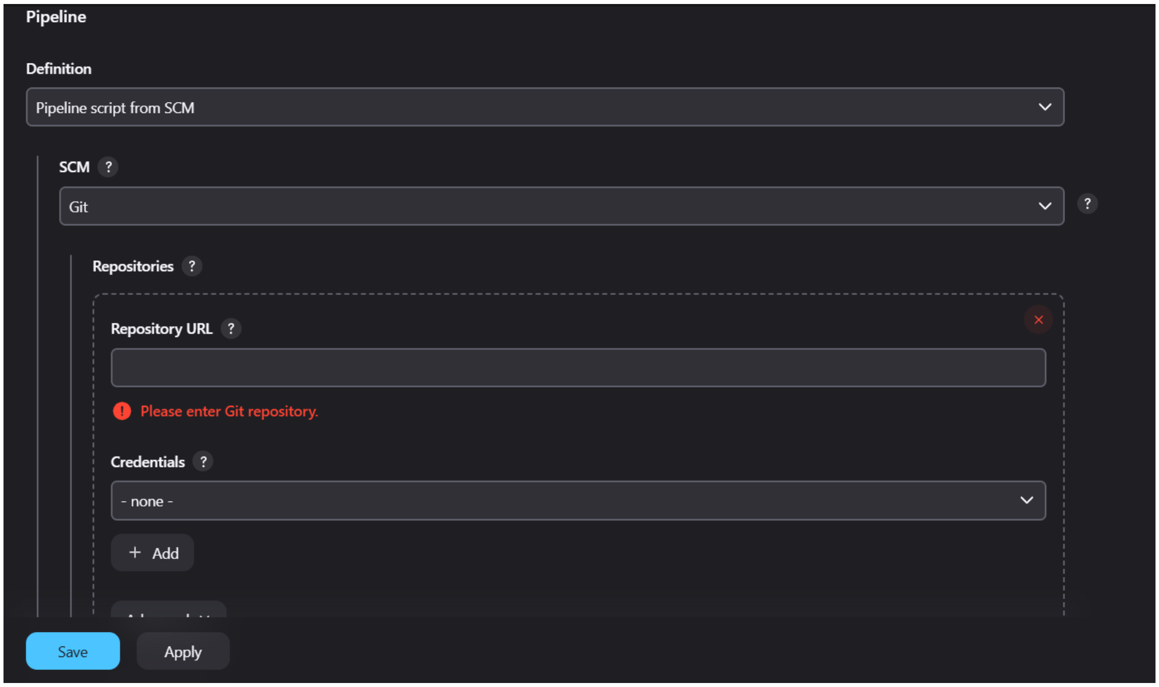
Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

9) Scroll down t o Pipeline and change the Definition to Pipeline script from SCM. Change SCM to Git. Add the URL from the Github repository you intend to build.



Pipeline

Definition: Pipeline script from SCM

SCM: Git

Repositories

Repository URL

Please enter Git repository.

Credentials: - none -

10) Under Branch Specifier, delete master and add Main. Select Save.

11) Make sure to add the Jenkinsfile to your Repo.

12) Open Git bash and run the following commands to add terraform to the container for your build.

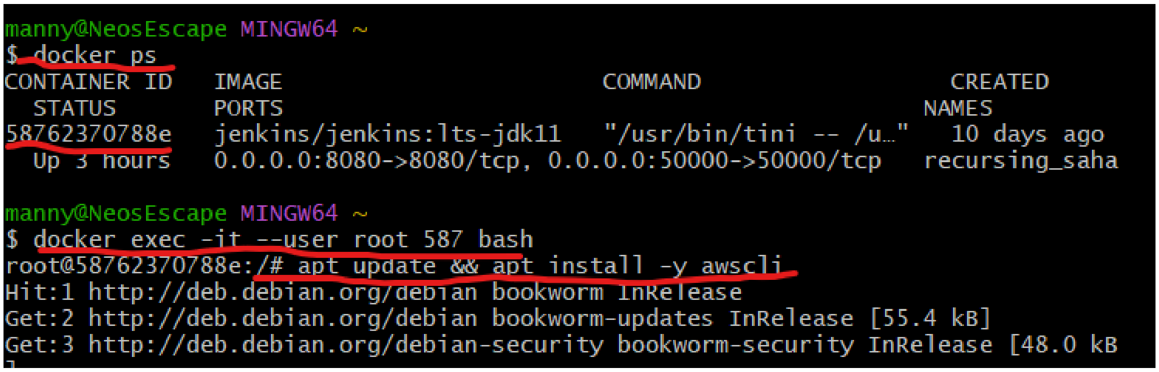
install aws and terraform into your jenkins instance you inted on using.

1.

docker exec -it --user root <YOUR-CONTAINER> bash

2. Install AWScli:

apt update && apt install -y awscli



```
manny@NeosEscape MINGW64 ~
$ docker ps
CONTAINER ID   IMAGE                        COMMAND                  CREATED
  STATUS       PORTS                                                   NAMES
58762370788e   jenkins/jenkins:lts-jdk11   "/usr/bin/tini -- /u…"   10 days ago
  Up 3 hours   0.0.0.0:8080->8080/tcp, 0.0.0.0:50000->50000/tcp   recursing_saha

manny@NeosEscape MINGW64 ~
$ docker exec -it --user root 587 bash
root@58762370788e:/# apt update && apt install -y awscli
Hit:1 http://deb.debian.org/debian bookworm InRelease
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB
]
```

3. Make a new directory for install:

mkdir -p /home/jenkins/bin

4. Curl and install Terraform:

curl -fsSL https://releases.hashicorp.com/terraform/1.5.7/terraform_1.5.7_linux_amd64.zip -o /home/jenkins/terraform.zip

5. Unzip:

unzip /home/jenkins/terraform.zip -d /home/jenkins/bin

6. Cleaning Up Zips:

rm /home/jenkins/terraform.zip

export PATH="/home/jenkins/bin:$PATH"

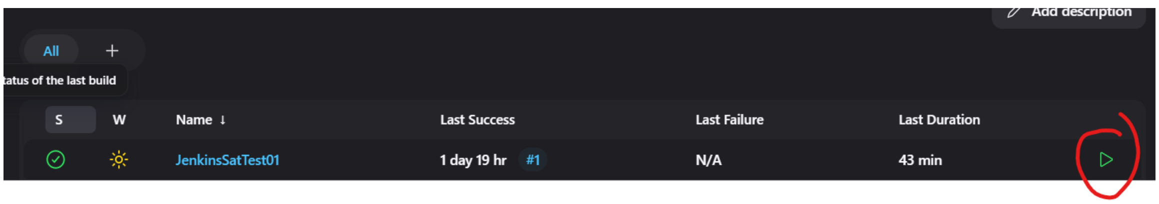
7. Move terraform to /usr/local/bin:

mv /home/jenkins/bin/terraform /usr/local/bin

8. Check the version to verify

terraform version

13) Head back to Jenkins and press play on your Pipeline



S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☼	JenkinsSatTest01	1 day 19 hr #1	N/A	43 min	▷

14) Select Console Output and watch the build for errors

15) Celebrate because the build completed successfully!

16)

- To view the terraform files deployed by jenkins follow the path below:

/var/jenkins_home/workspace/<PIPELINENAME>

How to tear Down the resources Jenkins deployed using Terraform:

- 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>
```

- 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"
```

- Run Terraform Destroy

```
terraform destroy
```