kura labs

Deployment #2

Welcome to Deployment 2!! Time to step it up a bit for this deployment. You will need to follow the steps below to set up a CI/CD pipeline from start to finish. We will still use Elastic Beanstalk and you will be in charge of customizing the pipeline!!

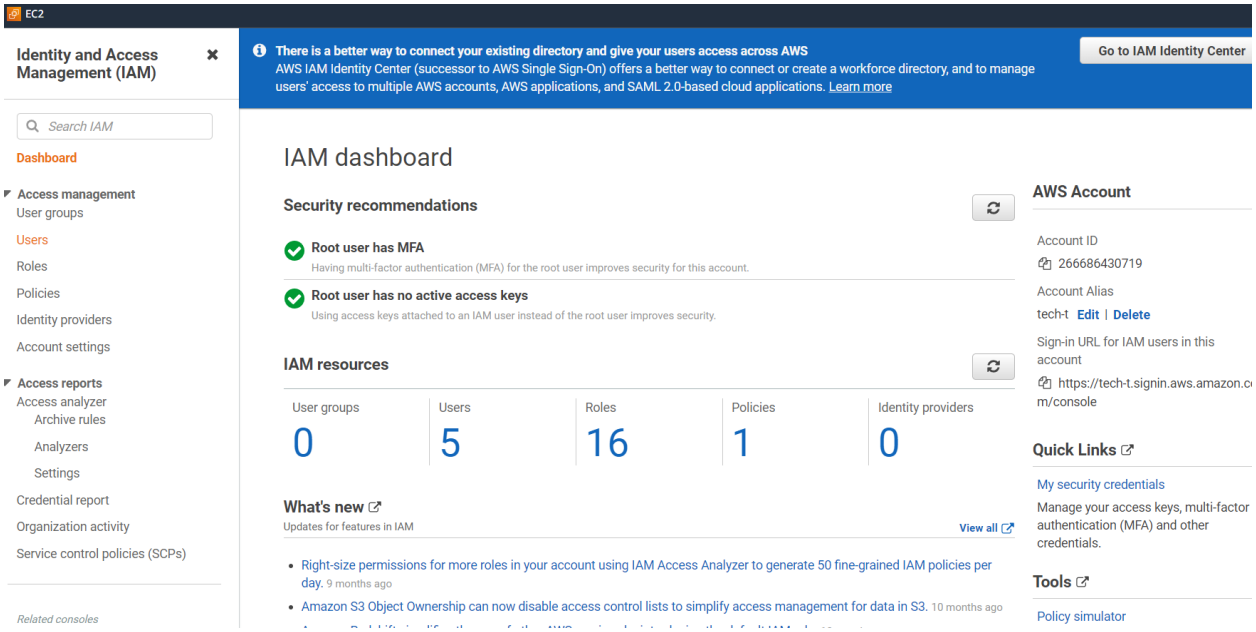
Install Jenkins on an EC2 if you haven't already:

Activate the Jenkins user on the EC2:

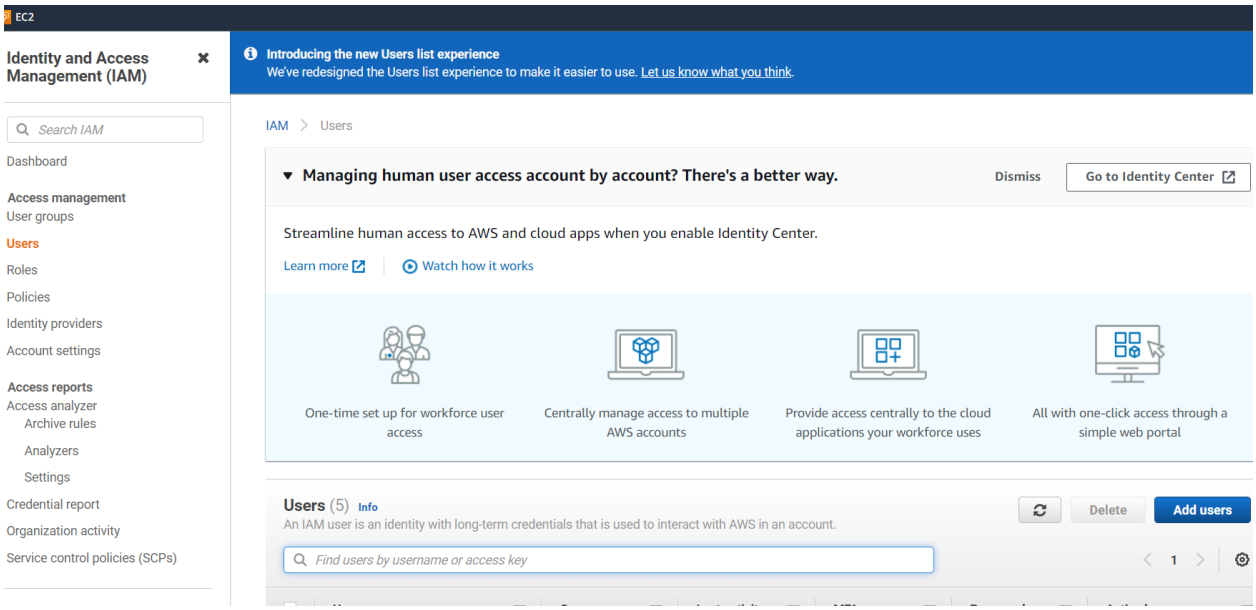
```
$sudo passwd jenkins
$sudo su - jenkins -s /bin/bash
```

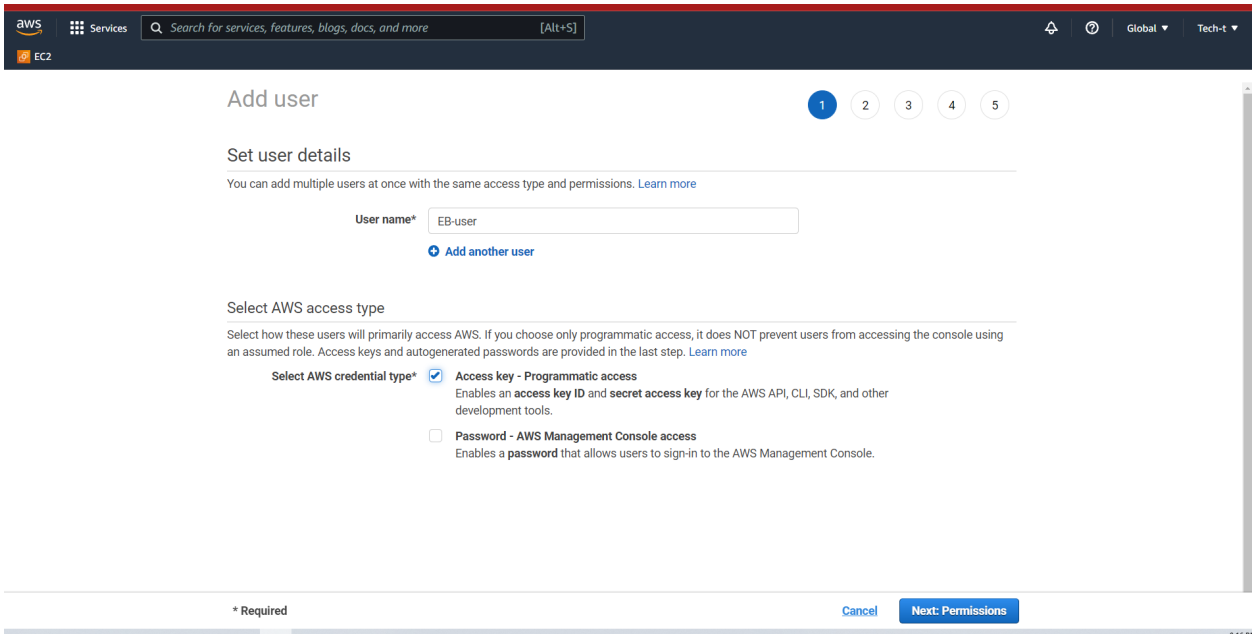
Create a Jenkins user in your aws account:

- Navigate to IAM in the AWS console. Next click on the Users option in the Access management.

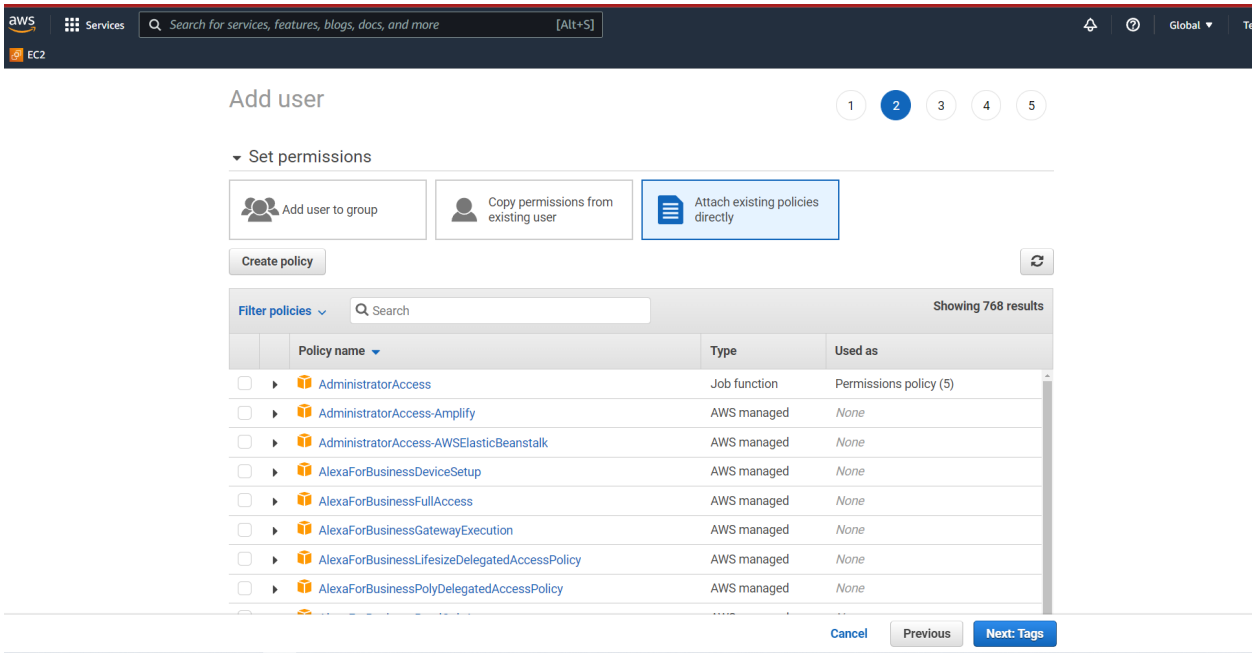


- Select add user. Next, the username can be EB-user. Then select Programmatic access and then next.

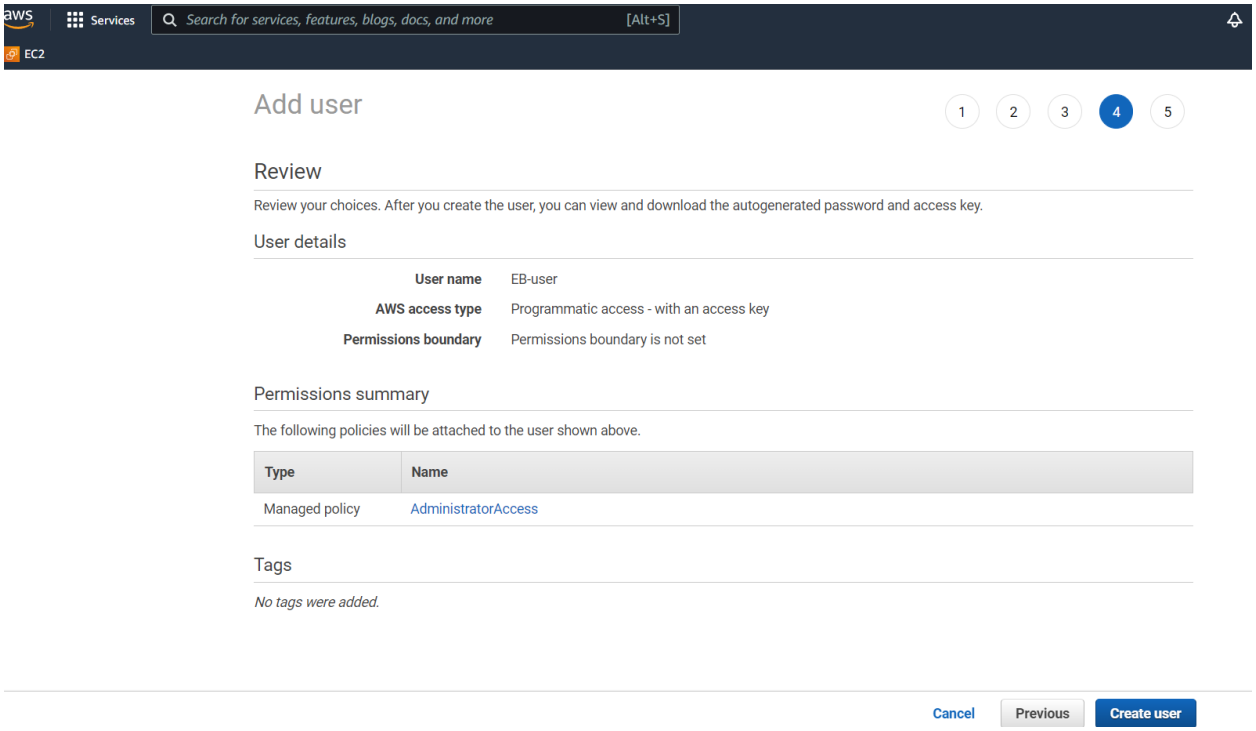


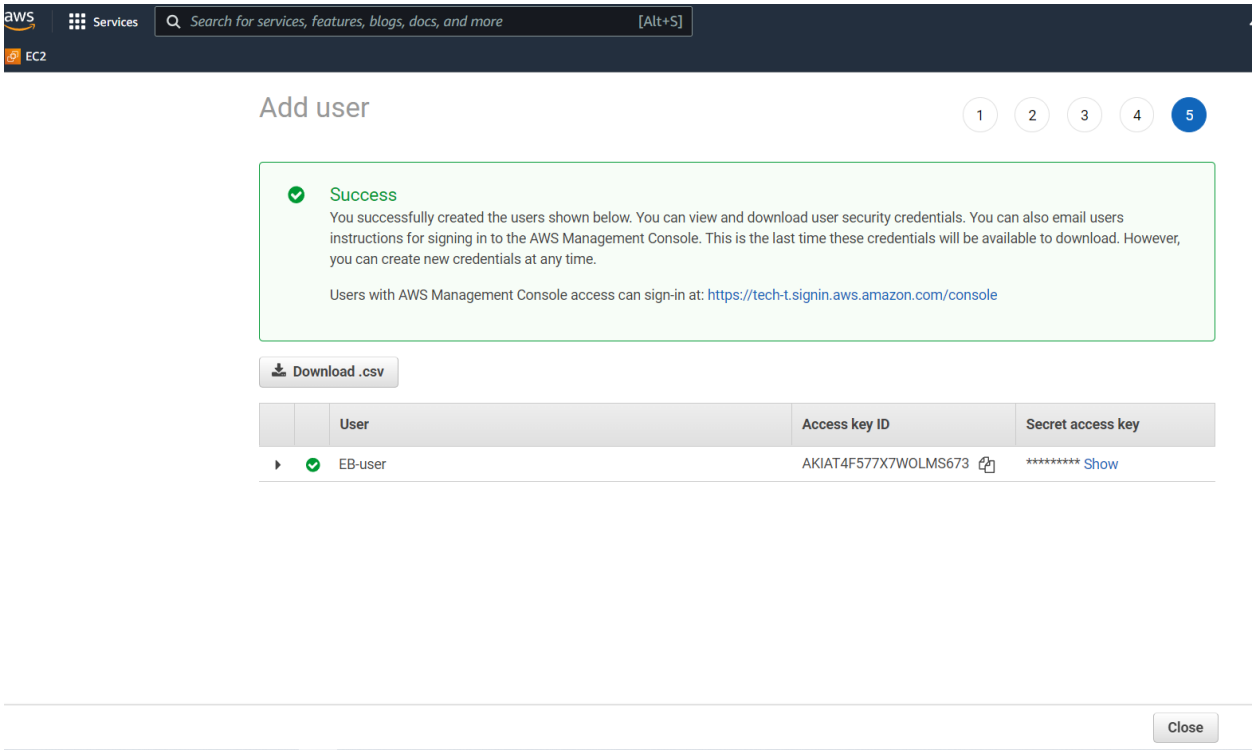


- Select “Attach existing policies directly” and select administrator access. Now select Next for this page and the next page.



- Finally, create the user and then copy and save the “access key ID” and the “secret access key”.





Install AWS CLI on the Jenkins EC2 and configure:

```
$curl
"https://awscli.amazonaws.com/awscli-exe-linux-x86_6
4.zip" -o "awscliv2.zip"

$unzip awscliv2.zip

$sudo ./aws/install
```

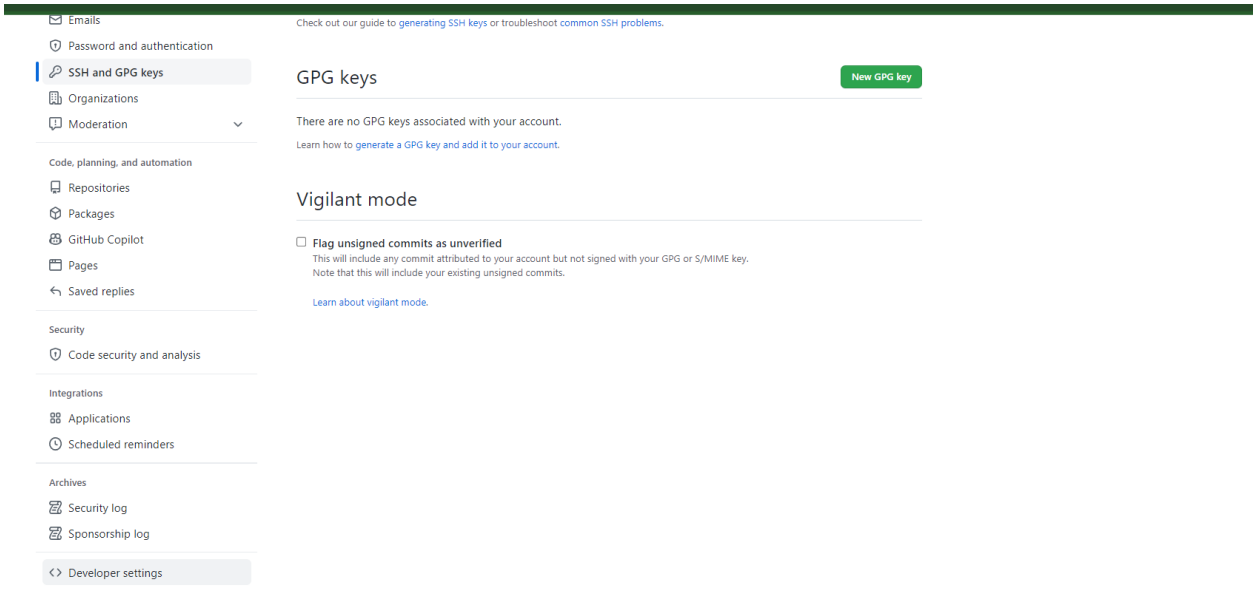
```
$aws --version

$sudo su - jenkins -s /bin/bash

$aws configure
  - Set Access Key ID
  - Set Secret Access Key
  - Set region to: us-east-1
  - Set Output format: json
```

Install EB CLI in the jenkins EC2 user:

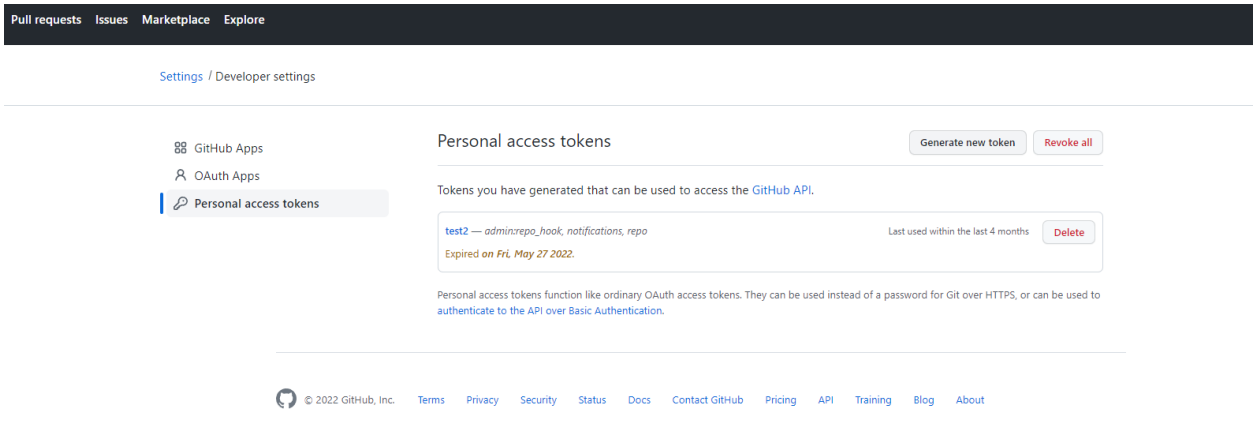
```
$pip install awsebcli --upgrade --user
$eb –-version
```

Connect GitHub to Jenkins Server:

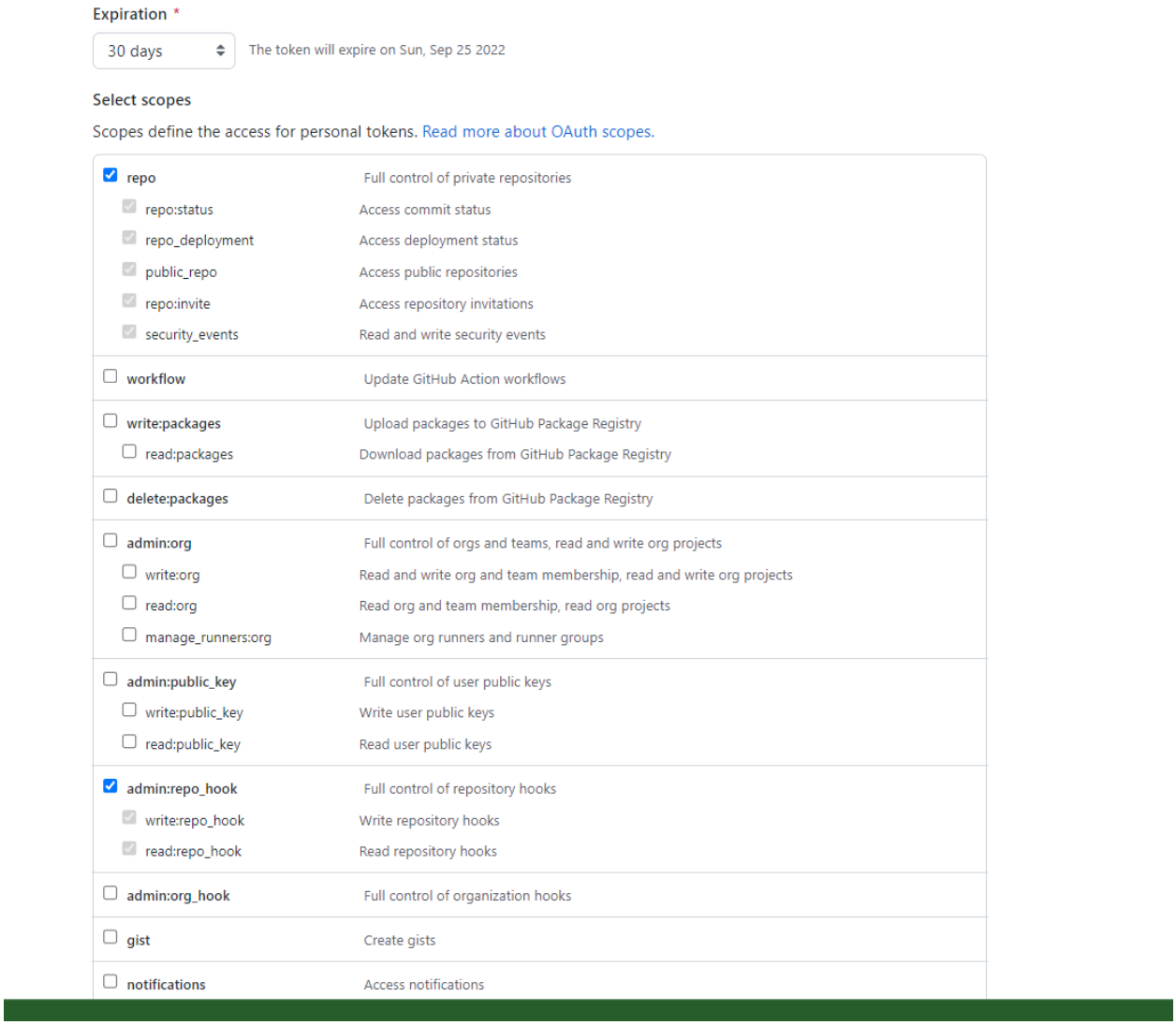
- First Fork the Deployment repo: https://github.com/kura-labs-org/kuralabs_deployment_2.git
- Next, create an access token from GitHub:
    - Navigate to your GitHub settings, select developer settings



- Select personal access token and create a new token.

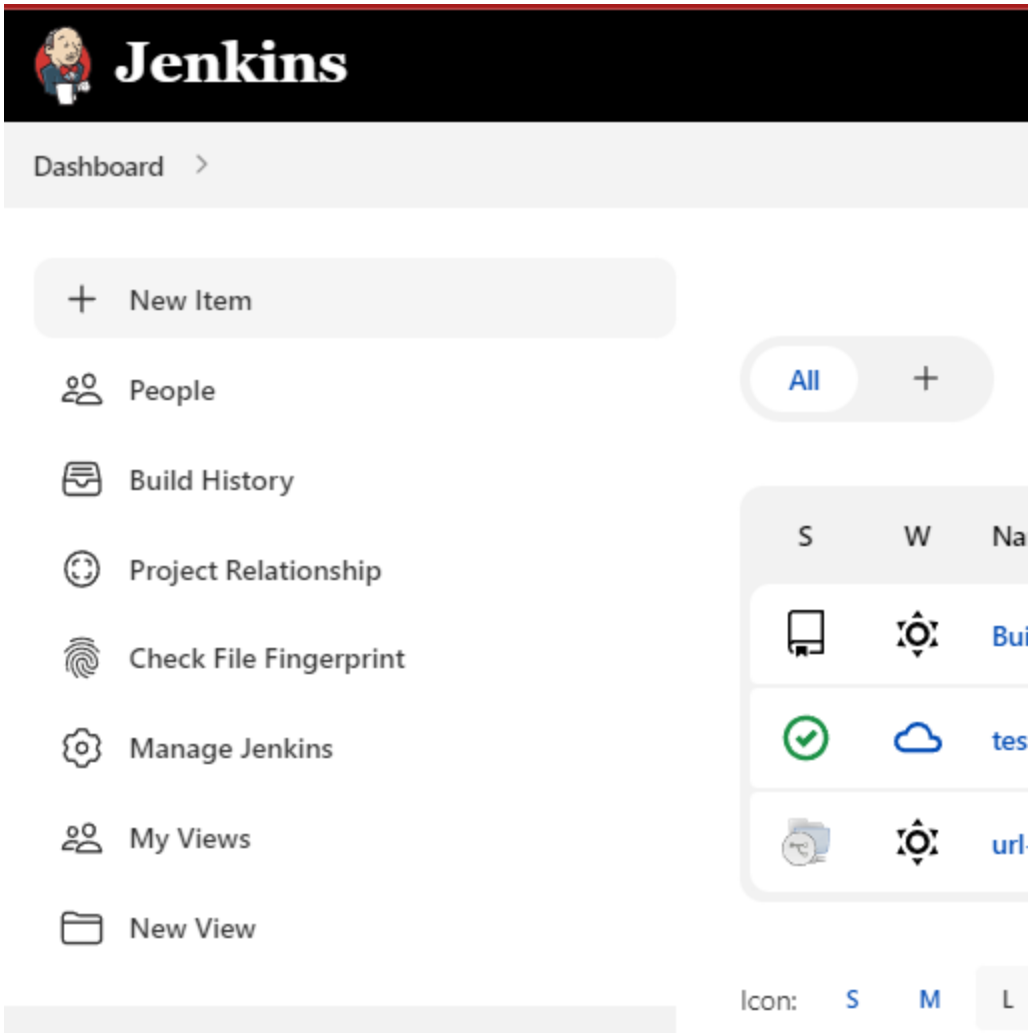


- Select the settings you see below for access token permissions.

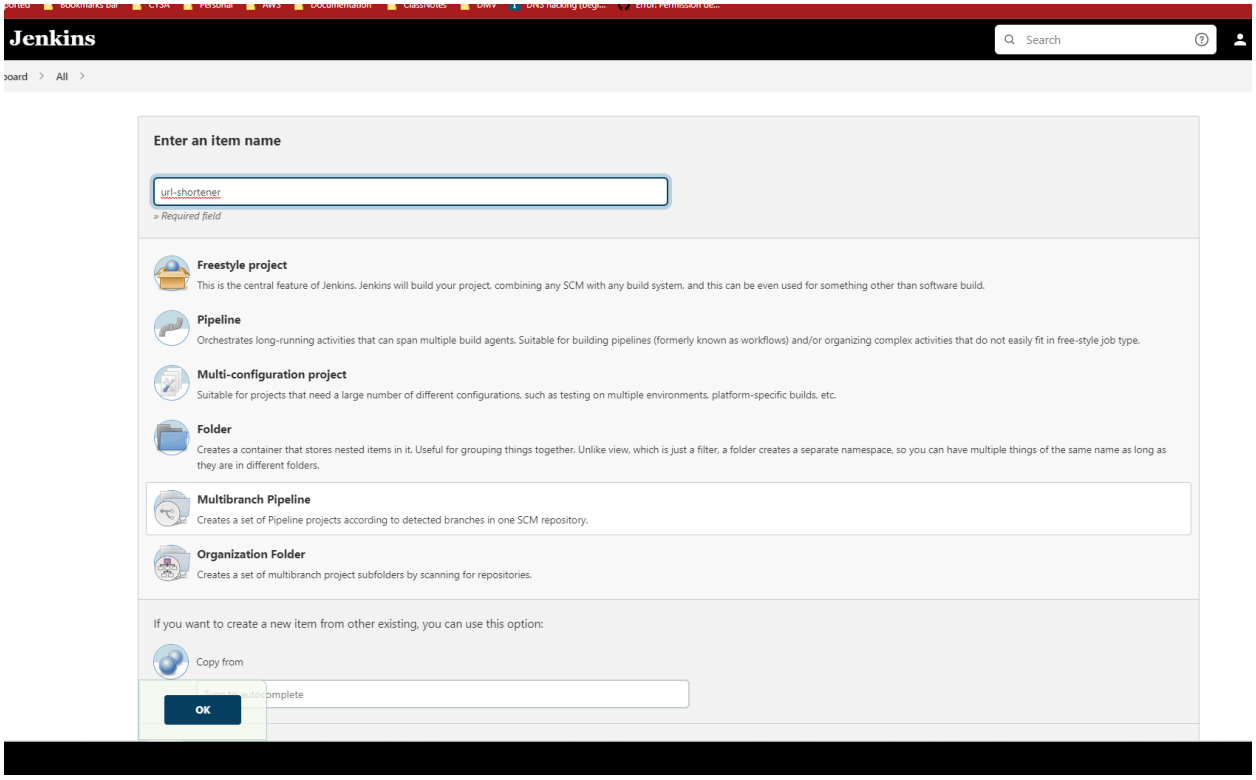


Create a multibranch build:

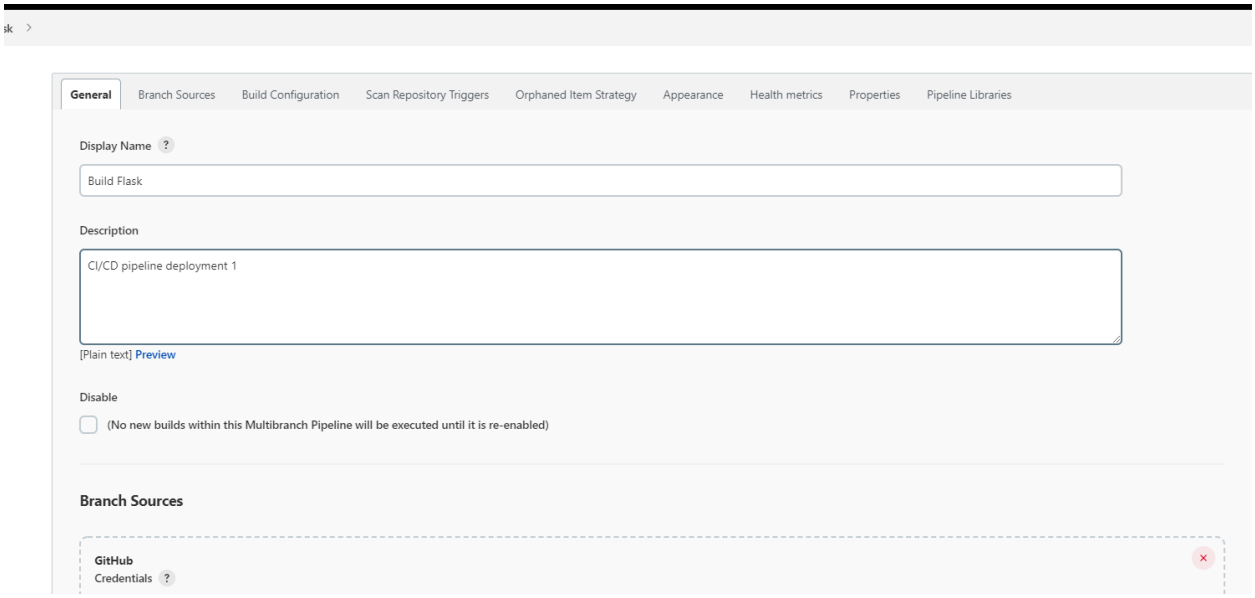
- Log back into Jenkins and select “New item”


Jenkins
Dashboard
New Item
People
Build History
Project Relationship
Check File Fingerprint
Manage Jenkins
My Views
New View
All
S
W
Na
Bui
tes
url
Icon:
S
M
L
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle

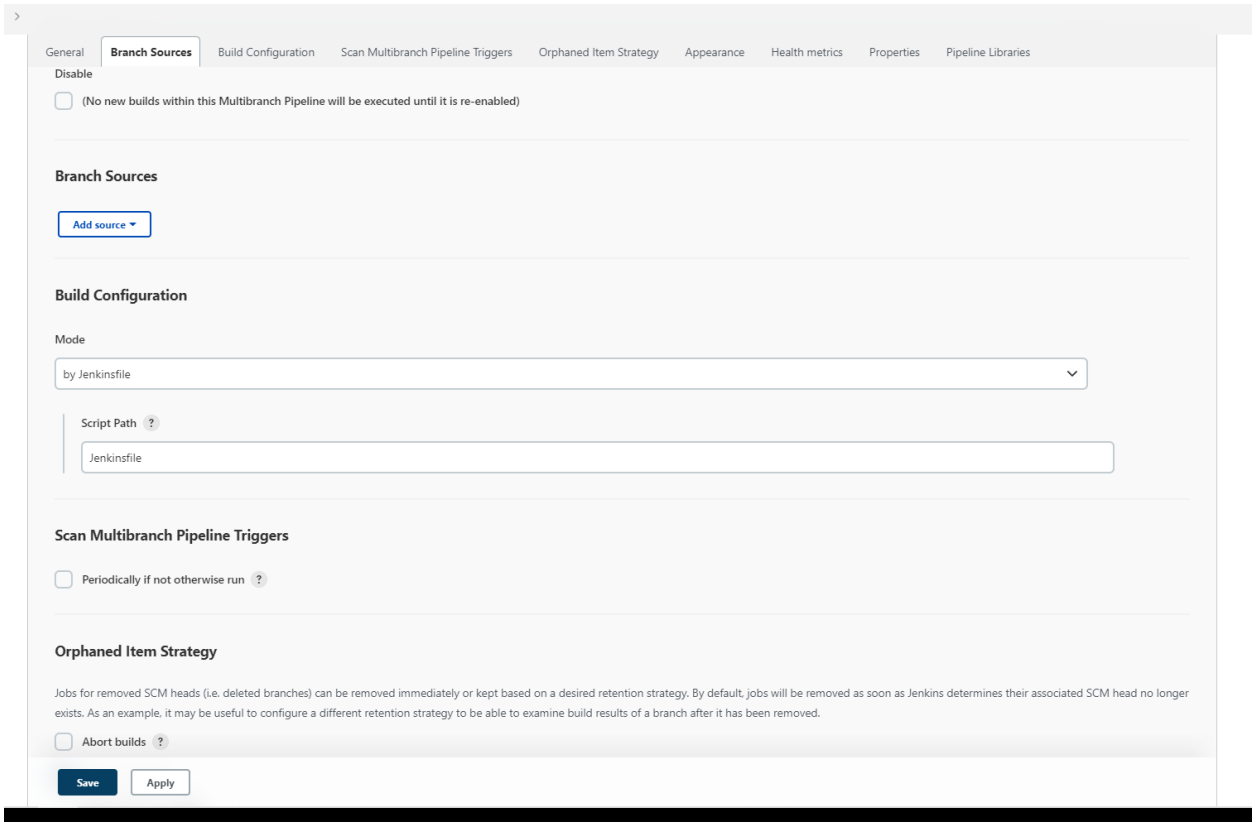
- Select multibranch pipeline



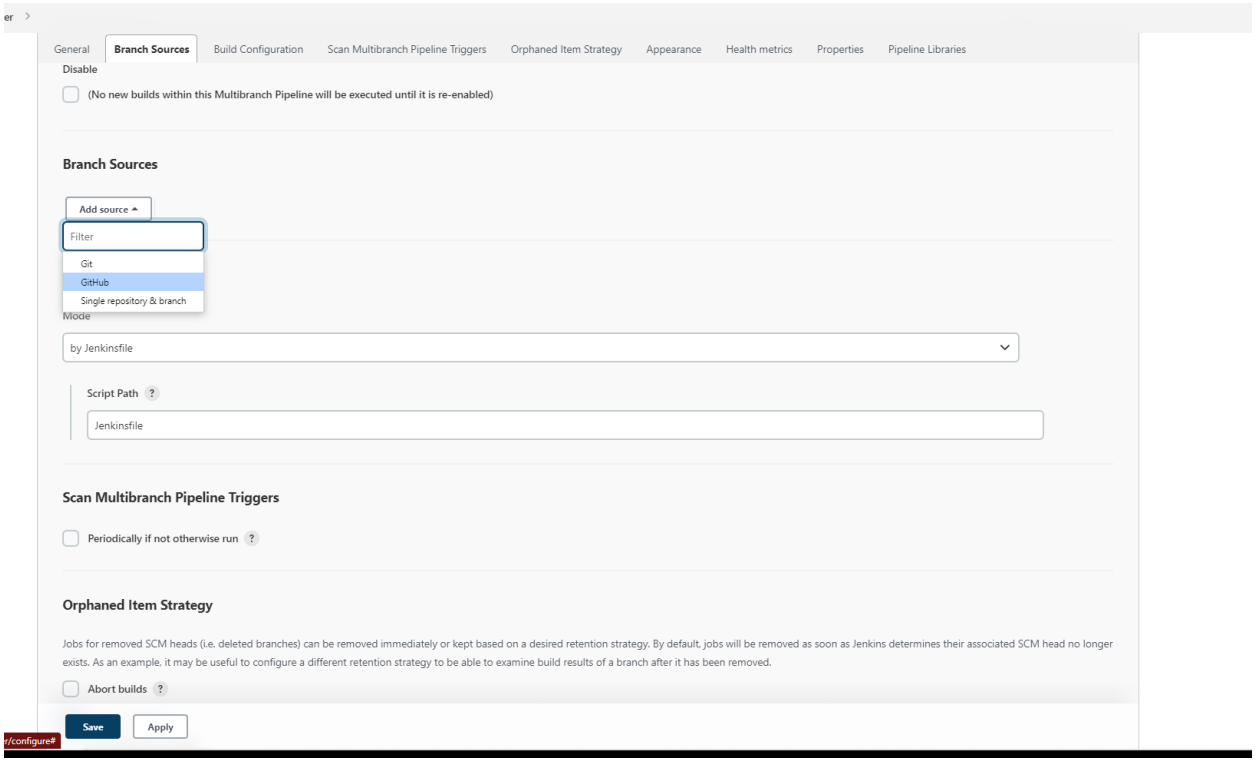
- Enter a display name and brief description



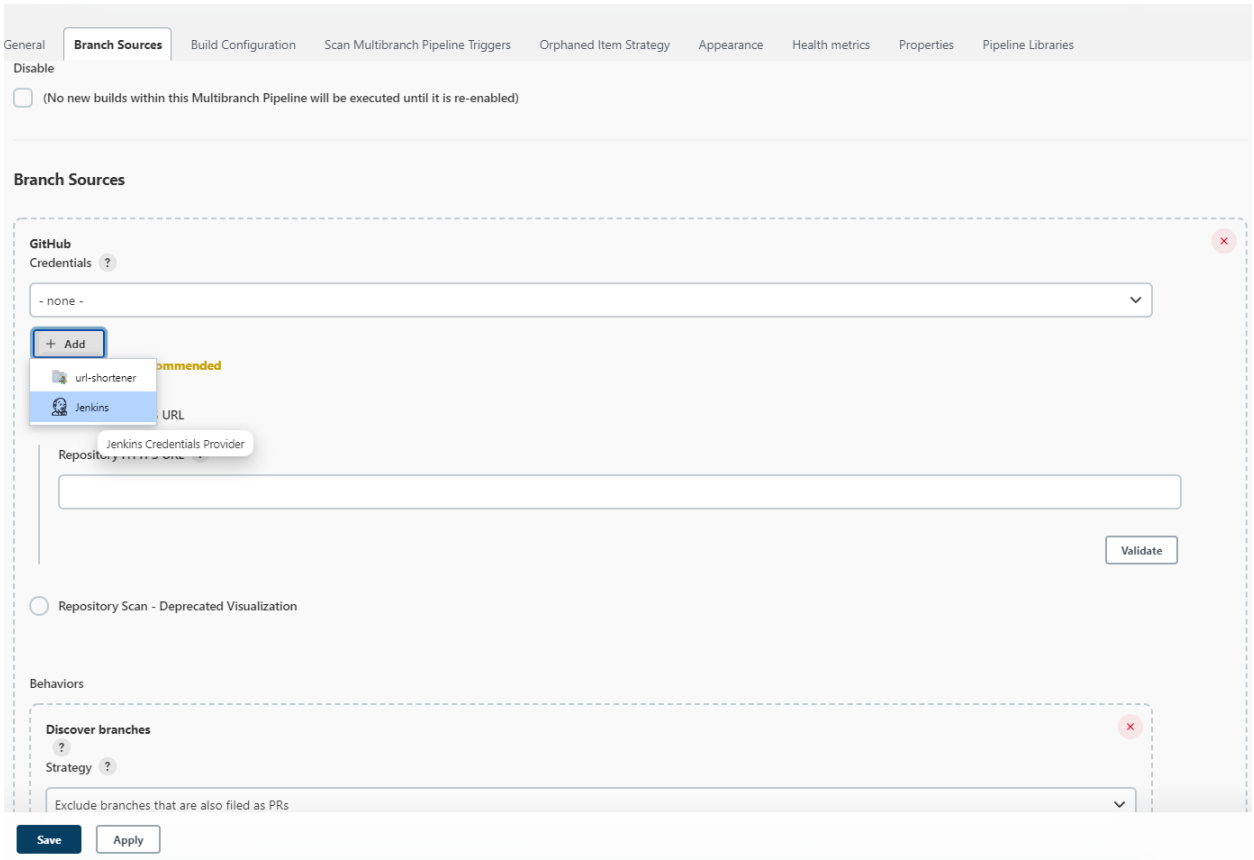
- Add a Branch source by selecting Add source and select GitHub



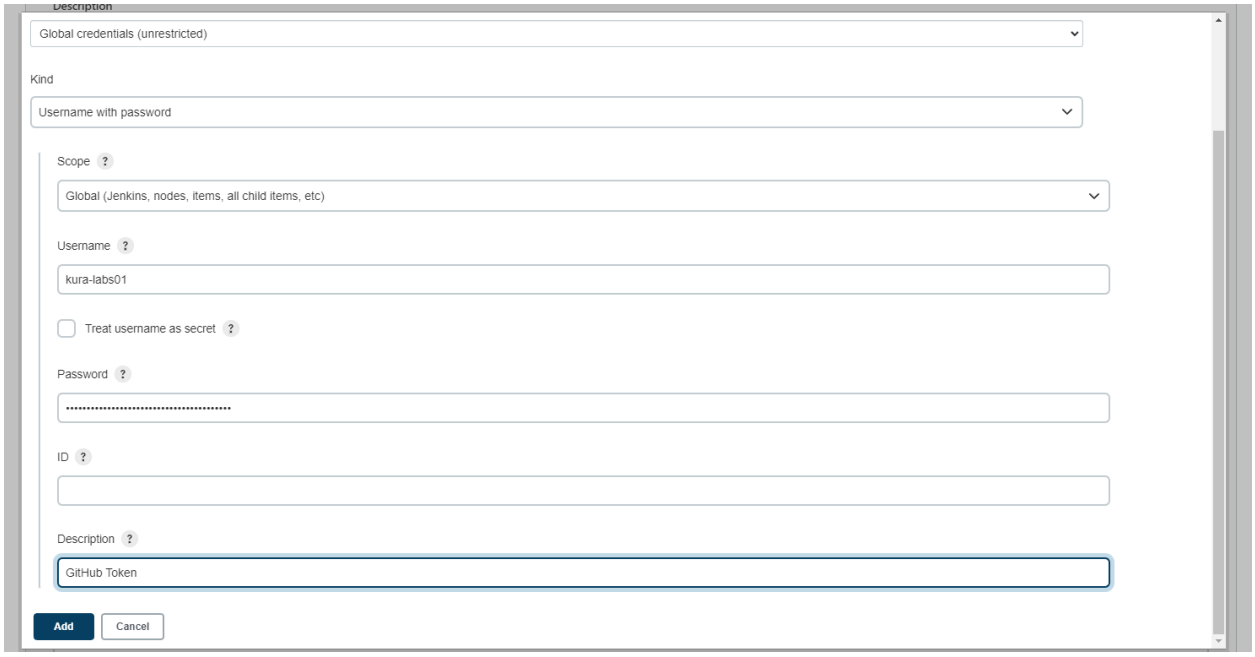
- Select the Add button and select GitHub



- Click on Add and then select Jenkins

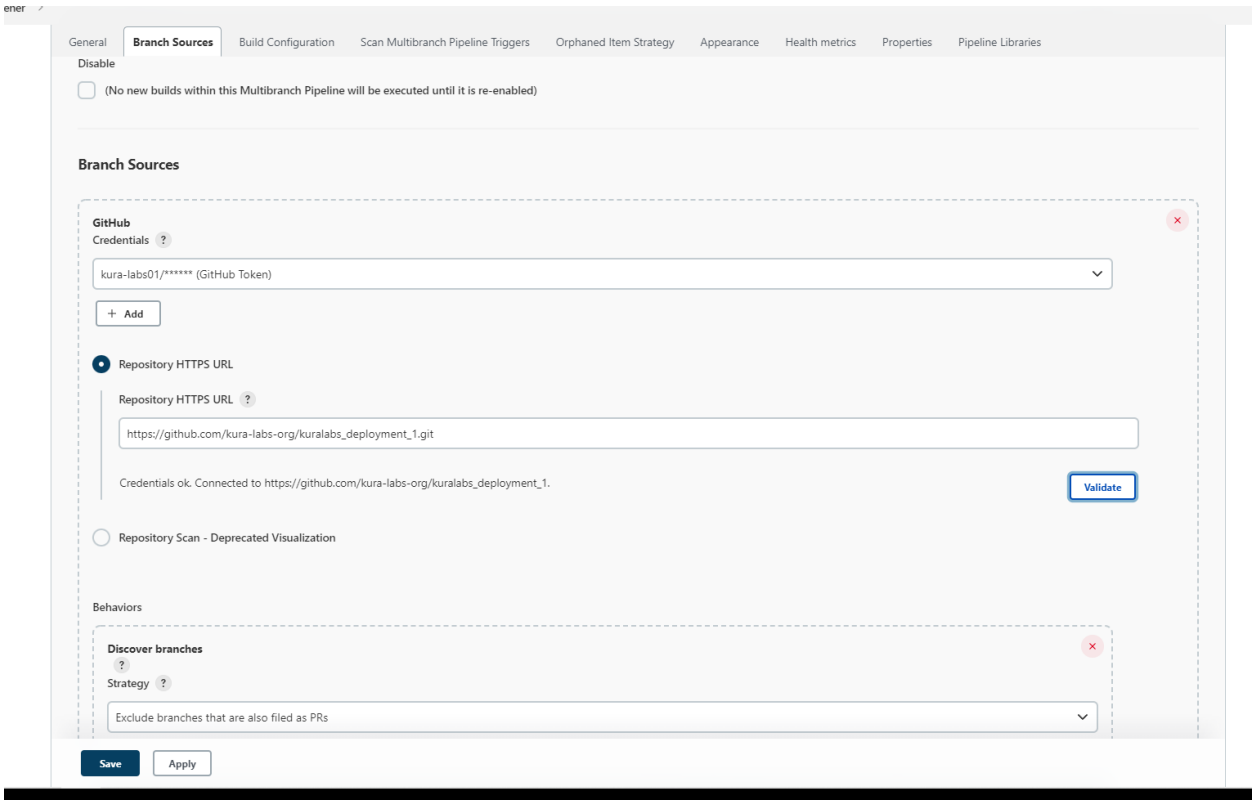


- Under username enter your GitHub username
- Under password enter your token

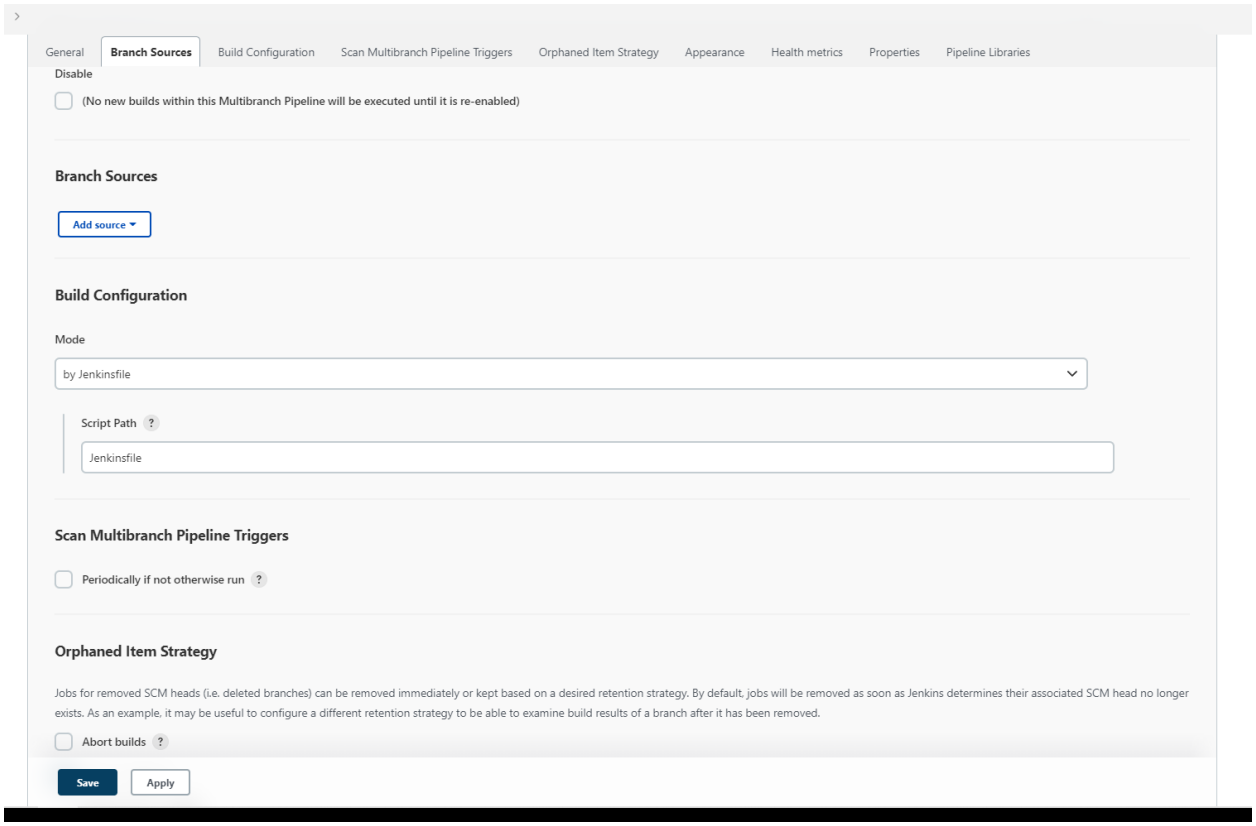


- (Optional) under ID and Description enter GitHub repo

- Enter your URL to the repository and you can validate by selecting validate.

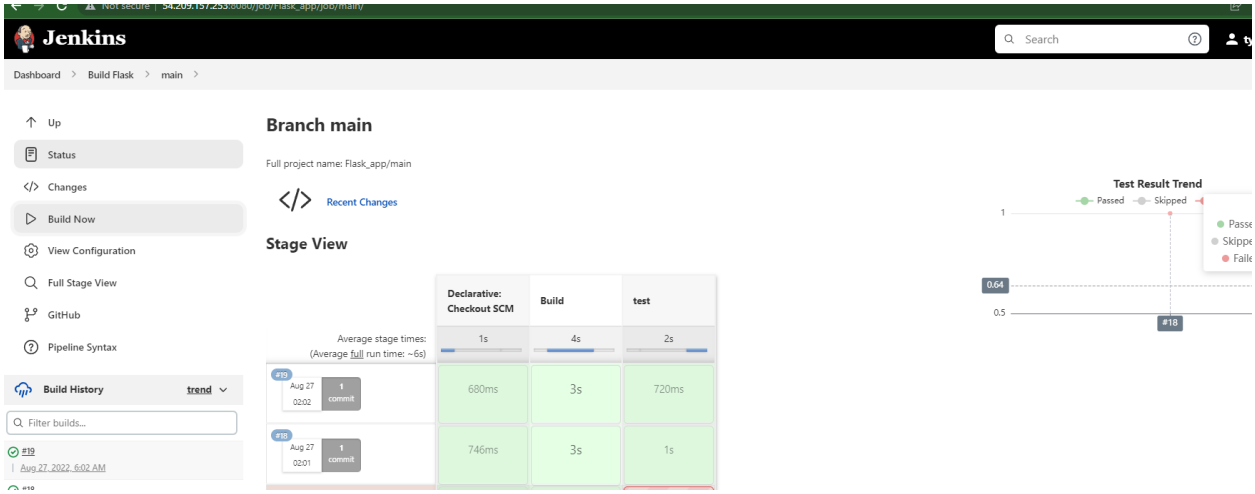


- Make sure this says Jenkinsfile



- Select Apply and then Save

- You should see a build happening. If you don’t, select Scan Repository.



Now deploy application from Elastic Beanstalk CLI:

```
$sudo su - jenkins -s /bin/bash
$cd /var/workspace/{{The name of your project}}/
$eb init
```

- Select: us-east-1
- Press enter
- Select: Python
- Select: (The latest version of python available)
- Select: N (for CodeCommit)

```
$eb create
```

- Take the default for the next 3 questions by hitting enter (remember the environment name)

- Spot Fleet: No
- Wait for the environment to be made!! And then check it

Now add a deployment stage to the pipeline in your Jenkinsfile:

```
pipeline {
  agent any
   stages {
    stage ('Build') {
      steps {
        sh '''#!/bin/bash
        python3 -m venv test3
        source test3/bin/activate
        pip install pip --upgrade
        pip install -r requirements.txt
        export FLASK_APP=application
        flask run &
        '''
     }
   }
    stage ('test') {
      steps {
        sh '''#!/bin/bash
        source test3/bin/activate
        py.test --verbose --junit-xml test-reports/results.xml
        '''
      }

      post{
        always {
```

```
          junit 'test-reports/results.xml'
        }

      }
    }
    stage ('Deploy') {
      steps {
        sh '/var/lib/jenkins/.local/bin/eb deploy {{Your environment
name}}'
      }
    }
  }
}
```

Now modify or add to the pipeline:

- You may add another test
- Add a way to notify you
- Use Cypress for testing
- Add a linter
- Change something on the application front

Once you have done that, apply your changes by building the pipeline again

Diagram the new pipeline!!!

Create documentation!!!

Note: Please submit your work by uploading your work to a repo or the forked repo. Then submit the link to the repo via LMS.