kura labs

Deployment #3

Welcome to Deployment 3!! Time to deploy to your customized VPC. You will need to follow the steps below and then add to the pipeline.

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

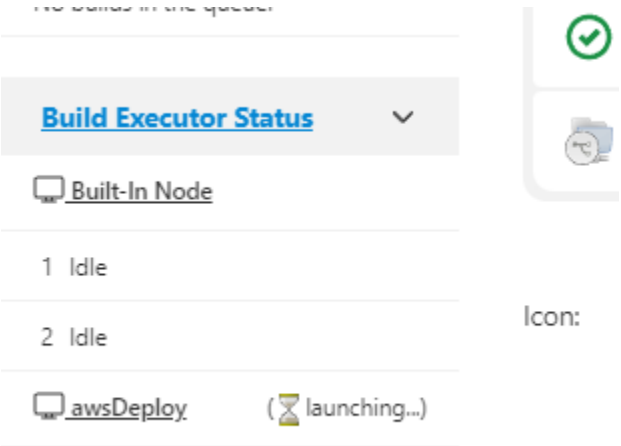
- You do not have to recreate a Jenkins server in your VPC. Highly recommended you use your Jenkins server from the default VPC!!

2. Create an EC2 in your Public Subnet of your VPC:

- The Ubuntu EC2 will need ports number: 22 and 5000 open.
- Install packages: **default-jre, python3-pip, python3.10-venv and nginx.**

3. Configure and connect a Jenkins agent to Jenkins:

- Enter your Jenkins server and Select the Build Executor Status:



- Next Select “+ New Node” to configure and add the agent. Enter the node name “awsDeploy” and select “Permanent Agent” and then create.

Jenkins

Dashboard > Nodes >

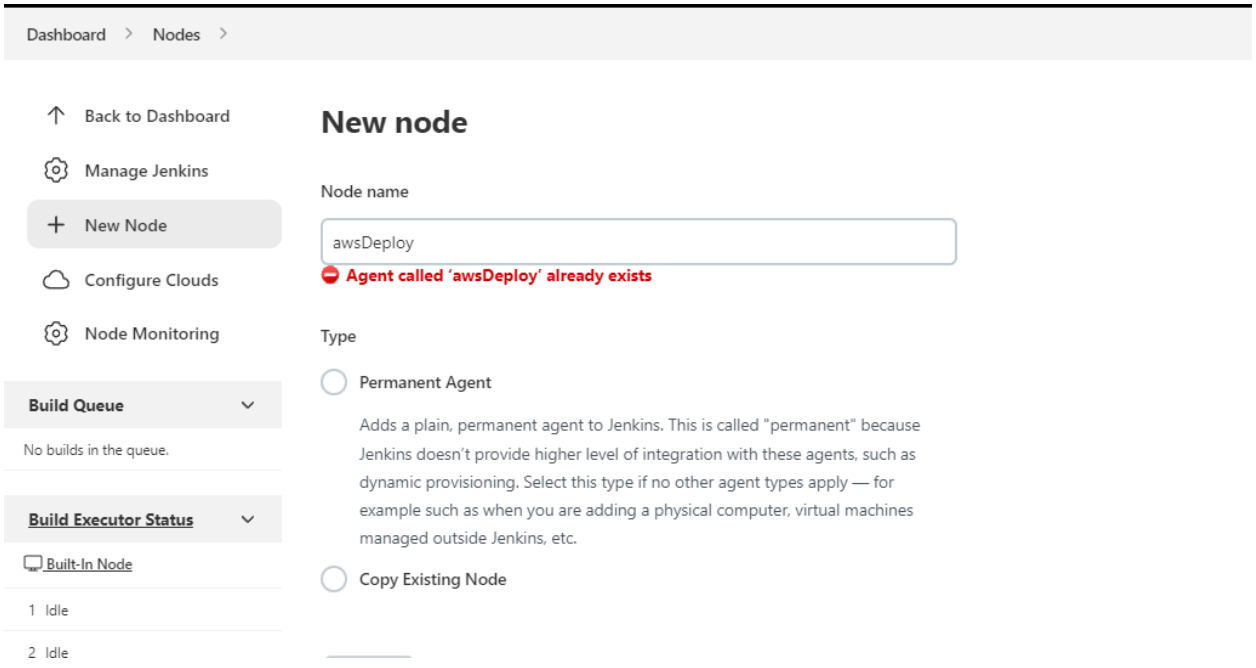
- Back to Dashboard
- Manage Jenkins
- New Node
- Configure Clouds
- Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node



- Now enter the configurations below:
  - Name: **awsDeploy**
  - Description: **Deployment server**
  - Number of executors: **1**
  - Remote root directory: **/home/ubuntu/agent**
  - Labels: **aweDeploy**
  - Usage: **only build jobs with label….**
  - Launch method: **launch agents via ssh**
  - Host: {Enter the public IP of your EC2 in the Public subnet and not this text}
  - **Credentials: see below**
  - Host key verification strategy: **non verifying verification strategy**

- Availability: **keep this agent online as much as possible**

awsDeploy

Name ?

awsDeploy

Description ?

Deployment server

Number of executors ?

1

Remote root directory ?

/home/ubuntu/agent

Labels ?

awsDeploy

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

54.163.30.187

Credentials ?

ubuntu (SSH-CALI)

+ Add

Host Key Verification Strategy ?

Non verifying Verification Strategy ?

Advanced...

- Credential steps:
  - Select “Add” => “Jenkins”=>Kind:”SSH username with private key”
  - Enter the ID, Description, username
  - To add the key, select “Enter Directly” => select “add” => paste the private key into the white box and save.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Username with password

- Username with password
- AWS Credentials
- GitHub App
- SSH Username with private key
- Secret file
- Secret text
- Certificate

Username ?

Treat username as secret ?

Password ?

ID ?

Description ?

Add Cancel

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

JenkinsAgent

Description ?

Deployment agent server

Username

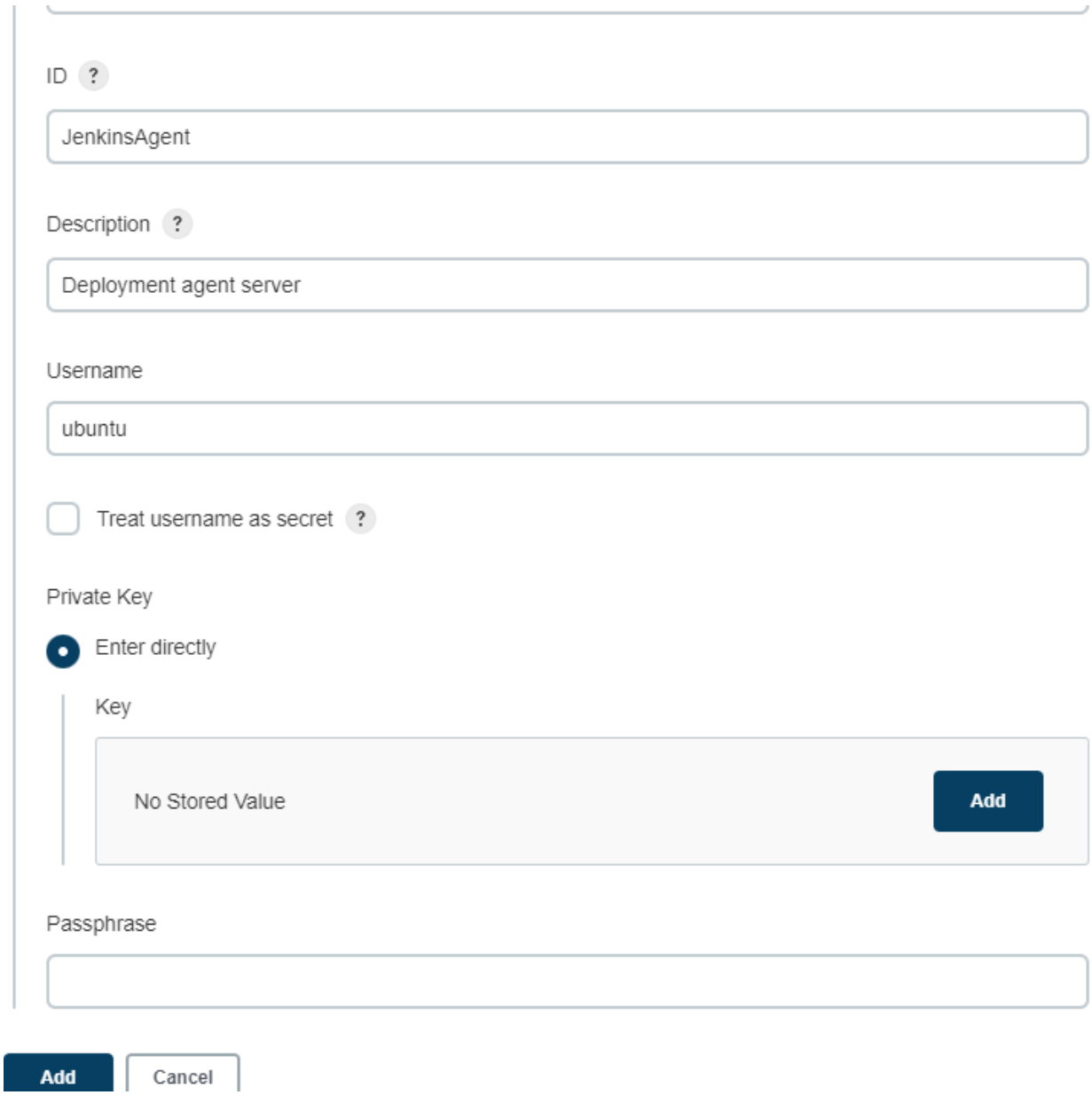
ubuntu

Treat username as secret ?

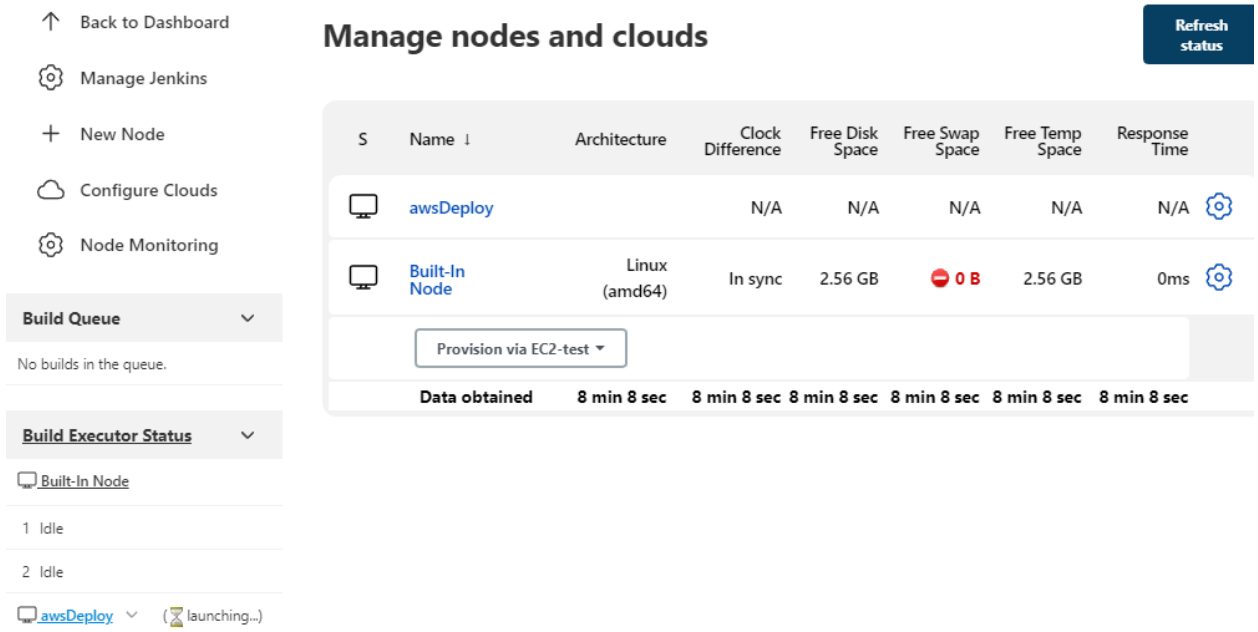
Private Key

Enter directly

Passphrase



- Save the configurations and wait for Jenkins to connect to the agent. It should look like what you see below:



4. Create a Pipeline build in Jenkins:

- Before you build your pipeline, SSH into the EC2 in your VPC and then nano into the "/etc/nginx/sites-enabled/default" file.
- First change the port from 80 to 5000:

```
server {
    listen 5000;
```

- Scroll down to where you see "location" and replace it with the text below:

```
location / {
```

```
        proxy_pass http://127.0.0.1:8000;
        proxy_set_header Host $host;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
```

- Now edit the Jenkinsfile in your repo to the script below:

```
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') {
      agent{label 'awsDeploy'}
      steps {
        sh '''#!/bin/bash
        git clone https://github.com/kura-labs-org/kuralabs_deployment_2.git
        cd ./kuralabs_deployment_2
        python3 -m venv test3
        source test3/bin/activate
        pip install -r requirements.txt
        pip install gunicorn
        gunicorn -w 4 application:app -b 0.0.0.0 --daemon
        '''
      }
    }
  }
}
```

- Log back into Jenkins and configure a multi branch pipeline or just a single pipeline build. Make sure you connect Jenkins to your GitHub Repo and then start your build!!

1. **Now add your additions from Deployment 2 to the Pipeline!!**

2. Diagram the new pipeline!!!

- Must have a diagram of pipeline and VPC
- Must also include the type of stack (rescrach software stacks)
- Must included any additions to the pipeline in the diagram

3. 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.