

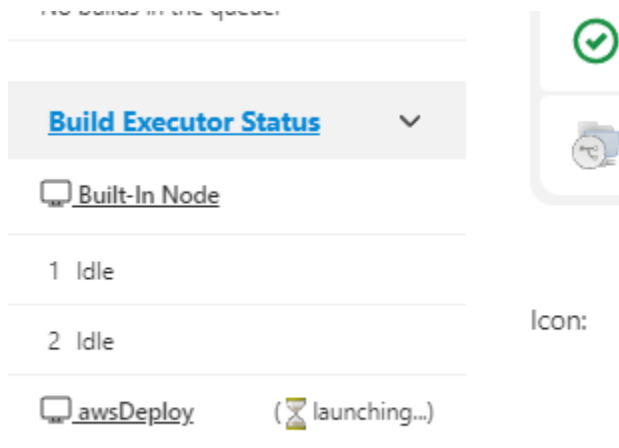
Deployment #3

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.

- **First I installed Jenkins on an EC2**
- **Next, I created an EC2 in my Public Subnet of the Kura VPC that we created in class last week**
 - In the security groups for the Ubuntu EC2, I added ports number: 22 and 5000 .
 - I attempted to use the command `sudo apt install` to install the following packages: **default-jre, python3-pip, python3.10-venv and nginx.**
 - **I could not install python3-pip at first and I received the error " Package 'python3-pip' has no installation candidate". To fix this error I ran `sudo apt update`. Then, I was able to run `sudo apt install python3-pip`.**
 - **I had to use `sudo apt-get install -y python3-venv` because `sudo apt install` did not work.**
- **I also was then able to install nginx.**

3. In order to configure and connect the Jenkins agents


- I entered my Jenkins server and selected the Build Executor status:




- Next, I selected “+ New Node” to configure and add the agent. I entered the node name “awsDeploy”, then selected “Permanent Agent”, and then I created the Node.




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](#)

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

1 Idle

2 Idle

New node

Node name

awsDeploy

🚫 Agent called 'awsDeploy' already exists

Type

☐ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

☐ Copy Existing Node

- Next I entered the configurations below:
 - 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**
 - I added the IP of my Public EC2 to the Host: 34.231.169.48
 - **These were the steps that I took to add the credentials.**
 - 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.

- Availability: **keep this agent online as much as possible**
- **Host key verification strategy: non verifying verification strategy**

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...

○

Credentials ?

ubuntu (SSH-CALI) ▼

+ Add



Jenkins

Host Key Verification Strategy ?

Non verifying Verification Strategy ▼

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



ID ?

JenkinsAgent

Description ?

Deployment agent server

Username

ubuntu

☐ Treat username as secret ?

Private Key

☒ Enter directly

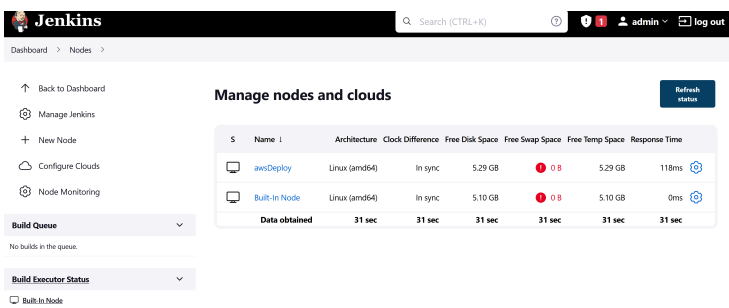
Key

No Stored Value Add

Passphrase

Add Cancel

- I saved the configurations but Jenkins agent failed to launch because the credentials failed to authenticate. It turns out that I was missing a dash from the pem file. When I added the dash, the credentials authenticated and the Jenkins agent launched. Please see what my agent looked like below:



4.To create a Pipeline build in Jenkins:

- I first SSH'ed into the EC2 in my public VPC and used sudo to nano'ed into the “/etc/nginx/sites-enabled/default” file.
- In the file, I changed the port from 80 to 5000:

```
server {  
    listen 5000;  
    listen [::]:5000;  
    # ...  
}
```

- I then scrolled down to “location” and replaced it with the text below:

```
location / {  
    # ...  
}
```

```
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;
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}
```

- Next I edited the Jenkinsfile in my 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
    '''
  }
}
}
}
}

```

- I configured a multi branch pipeline in Jenkins and connected Jenkins to my GitHub Repo but the build failed



Once I installed the python virtual environment on Jenkins, the build was successful but the url shortner website would not come up. Once I modified the Jenkinsfile with the `JENKINS_NODE_COOKIE=stayAlive`, I was able to have a successful build and I was able to deploy the url shortener website.

Also, I added the greeting modification from Deployment 2 to test file and I was able to have a successful build. Please see the screenshot below.

Stage View

