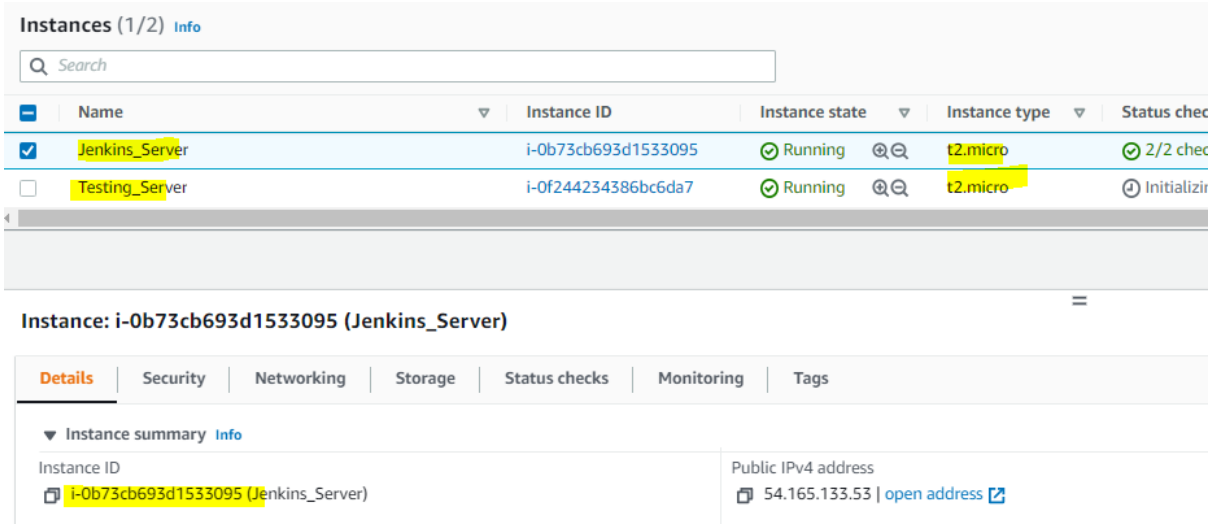
Task Requirements:

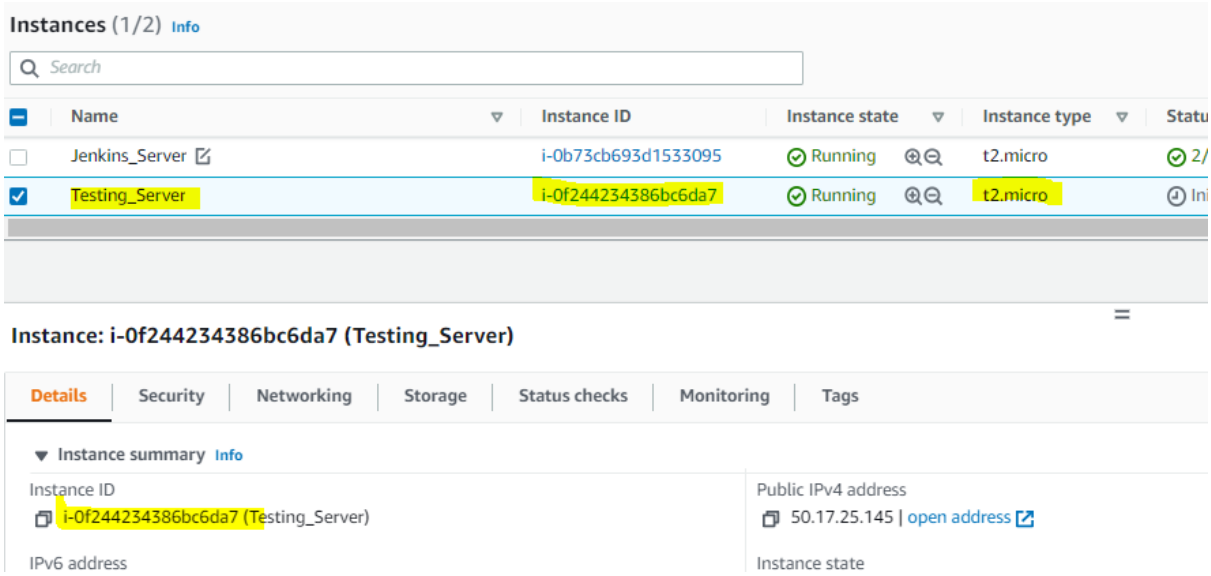
```
# Opstree-task
## Create a Jenkins Job that will take two inputs:
  1) Instance Id
  2) Instance Type

## The Jenkins Job should perform 3 operations:
  1) Change the instance type of instance with the provided instance type.
  2) Perform an operation using ansible on the provided instance
  3) Scale back the instance to the previous instance type.
```

Here I slightly changed the requirements:

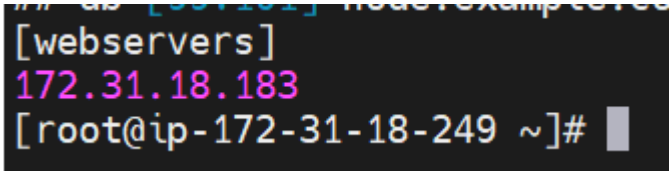
```
# Opstree-task
## Create a Jenkins Job that will take one input:
  1) Instance Type

## The Jenkins Job should perform 3 operations:
  1) Get the instance id from the tag and change the instance type of instance
with the provided instance type.
  2) Perform an operation using ansible on the provided instance
  3) Scale back instances to the previous instance type.
```

Created two instances – Jenkins_server and Testing_server





In the Jenkins_Server –

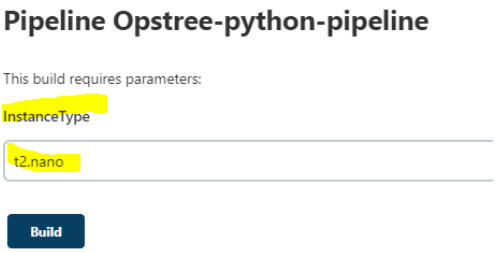
1. Jenkins pre-requisites (jdk and other dependencies)
2. Installed Jenkins
3. Ansible from amazon-linux-extras
    a. Created ansadmin user in both Jenkins_server and Testing_Server and created ssh-keys in Jenkins_server and copied public key and pasted in Testing_server – ansadmin -> .ssh –> authorized_keys (give permissions to .ssh folder and authorized_keys)
    b. Visudo – added ansadmin entry
    c. Changed Passwordlessauthentication yes in /etc/sshd/sshd_config then reloaded the sshd
    d. Verified ssh connectivity from ansible control node (jenkins_server) to target node (testing_server)
    e. In Jenkins server where ansible is installed under /etc/ansible/hosts – added webserver host group and given testing_server private ip
    i. 

4. In Jenkins installed Ansible plugin (configured ansible path in global tool configuration and configured ansadmin credentails as global tool configuration), Cloudbess aws plugin (configured aws-keys as username and password), configured git credentails

Below is the pipeline – https://github.com/madhusudhan7492/Opstree-task-python/blob/master/Jenkinsfile

<u>This pipeline code is all about:</u>

1. Taking the code from the Github repository
2. Verifying the instance which is having tags as Testing_Server (python script)
3. Verifying the instance type (in this case it is t2.micro)
4. Stopping the instance
5. Change the instance type (configured the parameterized input)



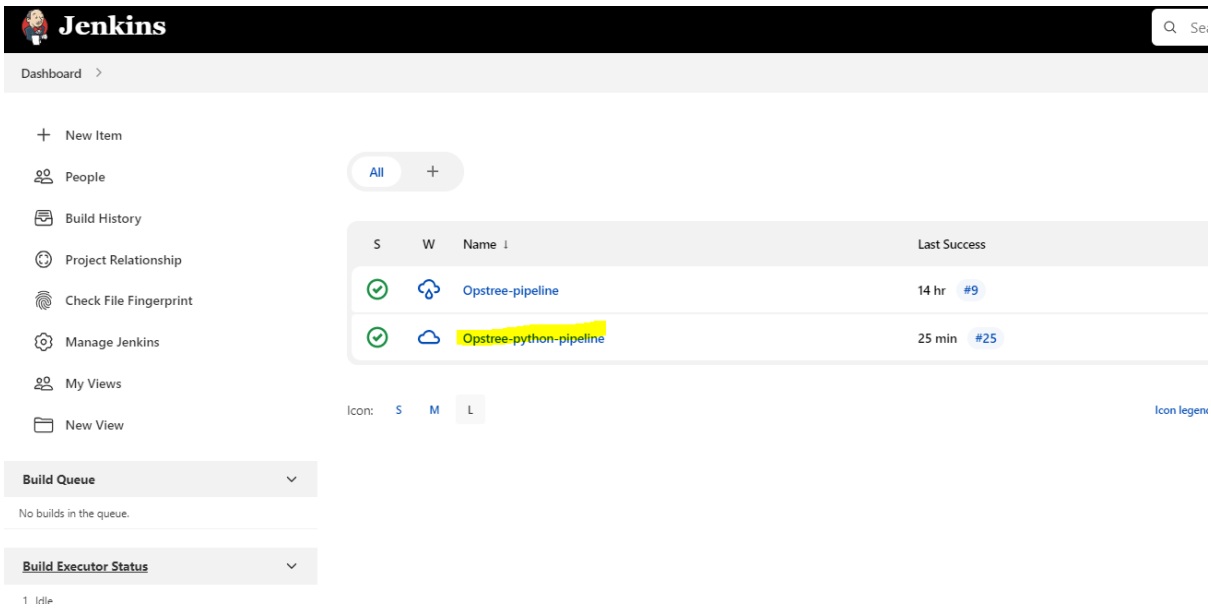
6. Start the instance
7. Run the ansible-playbook (install httpd, start the service and verify the service is running)
8. Stop the instance
9. Change the instance to its original (t2.micro)
10. Start the instance

```
pipeline {
  agent any
  stages {
    stage('Checkout SCM') {
      steps {
        checkout([$class: 'GitSCM', branches: [
          [name: '*/master']
        ], doGenerateSubmoduleConfigurations: false, extensions: [],
submoduleCfg: [], userRemoteConfigs: [
          [credentialsId: 'Github_creds', url:
'https://github.com/madhusudhan7492/Opstree-task-python.git']
        ]])
      }
    }

    stage('Run Start and Change Instance Type script') {
      steps {

        withCredentials([
          [$class: 'UsernamePasswordMultiBinding', credentialsId: 'aws-creds',
usernameVariable: 'AWS_ACCESS_KEY_ID', passwordVariable:
'AWS_SECRET_ACCESS_KEY']
        ]) {
          sh '/usr/bin/python3 -u startAndChangeInstance.py'
        }

      }
    }

    stage('Run ansible playbook') {
      steps {
        //here main.yml file is in the cloned repository
        ansiblePlaybook credentialsId: 'ansadmin', disableHostKeyChecking:
true, installation: 'ansible', playbook: 'main.yml'
      }
    }
```

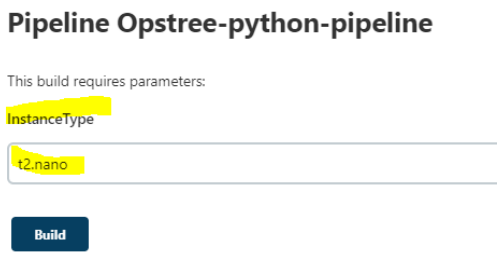
```
    stage('Run flipback and start instance script') {
     steps {

       withCredentials([
         [$class: 'UsernamePasswordMultiBinding', credentialsId: 'aws-creds',
usernameVariable: 'AWS_ACCESS_KEY_ID', passwordVariable:
'AWS_SECRET_ACCESS_KEY']
       ]) {
         sh '/usr/bin/python3 -u flipbackAndStartInstance.py'
       }

     }
   }

  }
}
```

Demo:

Created an Opstree-python-pipeline



Build with parameters



After running the job

	Declarative: Checkout SCM	Checkout SCM	Run Start and Change Instance Type script	Run ansible playbook	Run flipback and start instance script
Average stage times: (Average full run time: ~3min 1s)	221ms	194ms	1min 26s	7s	1min 3s
#25 Jul 03 12:41 — 1 commit	229ms	198ms	1min 26s	7s	1min 26s

BUILD_OUTPUT.txt

Build output:

Verifying the output – Httpd is installed and running successfully