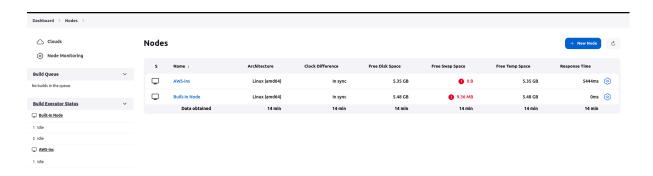
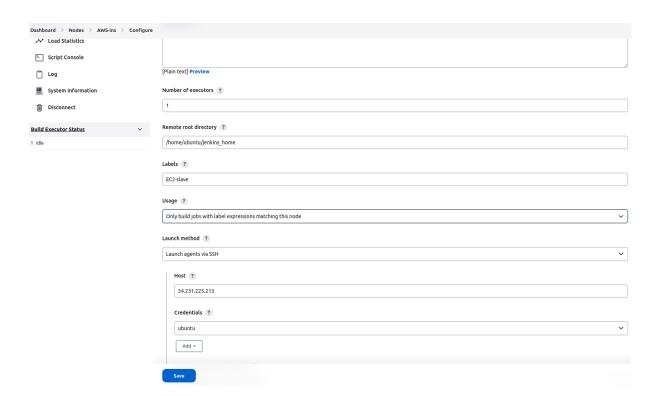
- 4- Try to make new slave as container or ec2 server and configure master to use it.
 - Create AWS Slave.



- Slave config, adding JNK private key



- Update the agent to AWS

```
Blame 35 lines (32 loc) · 897 Bytes
Code
        pipeline {
            agent {label 'AWS-ins'}
             environment {
                D = 'docker'
            parameters {
                 string(name: 'ENVIRONMENT', defaultValue: 'dev', description: 'Environment to deploy')
                 booleanParam(name: 'CLEAN_BUILD', defaultValue: true, description: 'Perform a clean build')
            stages {
                stage('Build') {
                     steps {
                         {\tt echo} \ {\tt "Build} \ {\tt for \ environment \ \$\{params. {\tt ENVIRONMENT}\} \ is \ successful"}
                  stage('Test') {
                     steps {
                         echo "Test for environment ${params.ENVIRONMENT} is successful"
                 stage('Deploy') {
```

Running process

Dashboard > PipeLine-app > #18

Status

Changes

Console Output

Edit Build Information

Delete build '#18'

Parameters

○ Restart from Stage⇒ Replay

Pipeline Steps

─ Workspaces← Previous Build

Onsole Output

Started by user Abdelfattah Ashraf [Pipeline] Start of Pipeline [Pipeline] node Still waiting to schedule task Waiting for next available executor on 'AWS-ins' Running on AWS-ins in /home/ubuntu/jenkins home/workspace/PipeLine-app@2 [Pipeline] { [Pipeline] withEnv (hide) [Pipeline] { [Pipeline] stage [Pipeline] { (Build) [Pipeline] echo Build for environment dev is successful [Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (Test) [Pipeline] echo Test for environment dev is successful [Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (Deploy) [Pipeline] echo Deployment for environment dev is successful [Pipeline] } [Pipeline] // stage [Pipeline] stage [Pipeline] { (Cleanup) [Pipeline] echo Cleanup for environment dev is successful [Pipeline] } [Pipeline] // stage [Pipeline] } [Pipeline] // withEnv