Login as admin to http://127.0.0.1:8080/jenkins (Jenkins baseURL) or http://127.0.0.1/jenkins (Nginx reverse proxy).

Click ‘**Manage Jenkins**’ / ‘**Manage Nodes and Clouds**’ / ‘**New Node**’

To create Inbound Agent (aka JNLP4 or Java Web Start agent), for example enter:

Node: **jkagent01**

Click ‘**Permanent Agent**’

Click ‘**OK**’

Enter:

# of executors: **2**

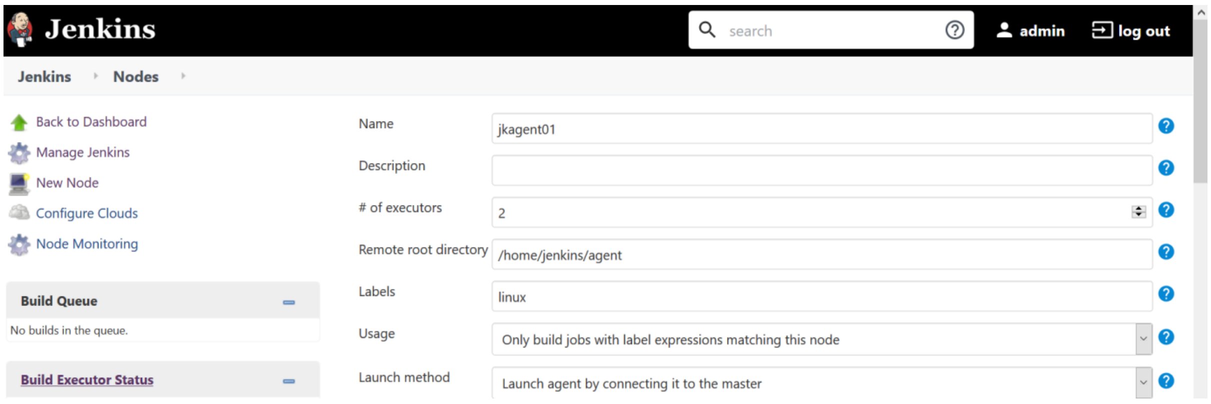
Remote root directory: **/home/Jenkins/agent**

Labels: **linux**

Usage: **Only build jobs with label expressions matching this node**

Launch Method: **Launch agent by connecting it to the master**

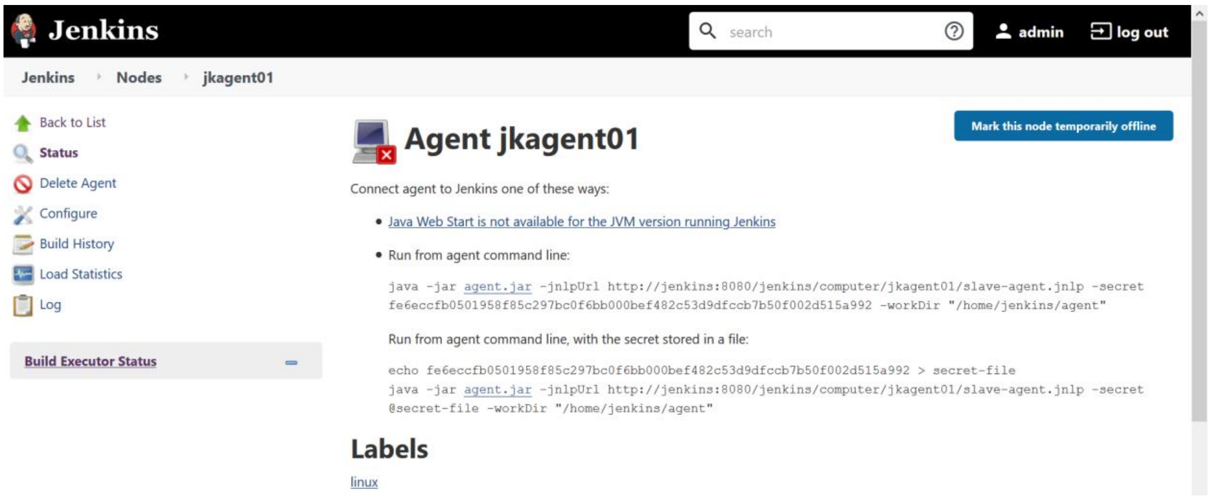
Click ‘**Save**’



No builds in the queue.

Name	jkagent01
Description	
# of executors	2
Remote root directory	/home/jenkins/agent
Labels	linux
Usage	Only build jobs with label expressions matching this node
Launch method	Launch agent by connecting it to the master

Click the created node ‘**jkagent01**’ to look at its details.



Agent jkagent01

Connect agent to Jenkins one of these ways:

- Java Web Start is not available for the JVM version running Jenkins
- Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://jenkins:8080/jenkins/computer/jkagent01/slave-agent.jnlp -secret
fe6eccfb0501958f85c297bc0f6bb000bef482c53d9dfccb7b50f002d515a992 -workDir "/home/jenkins/agent"
```

Run from agent command line, with the secret stored in a file:

```
echo fe6eccfb0501958f85c297bc0f6bb000bef482c53d9dfccb7b50f002d515a992 > secret-file
java -jar agent.jar -jnlpUrl http://jenkins:8080/jenkins/computer/jkagent01/slave-agent.jnlp -secret
@secret-file -workDir "/home/jenkins/agent"
```

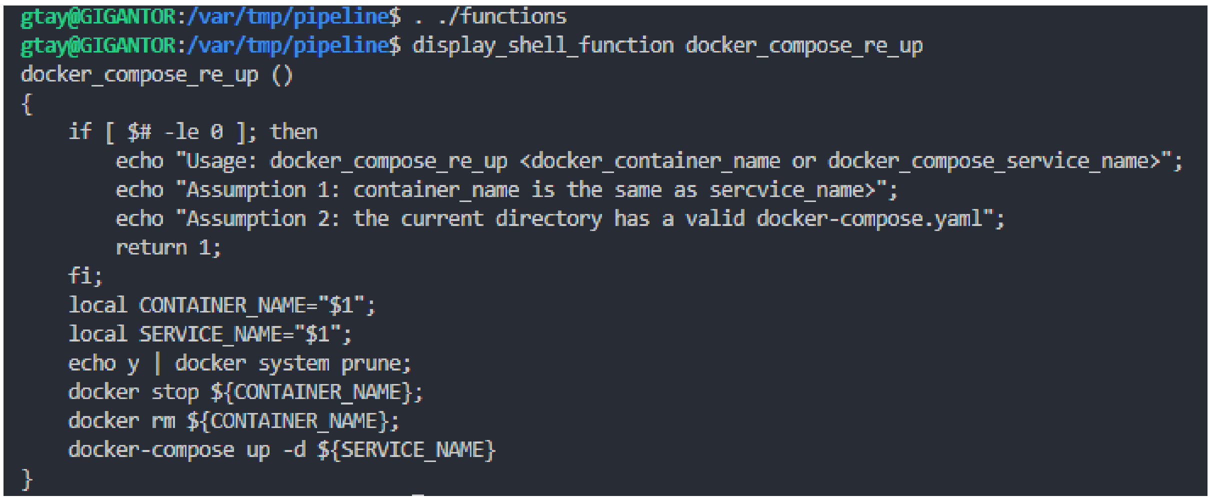
Labels

linux

Note down the secret and update it in docker-compose.yaml file.

To re-stop-rm-up jkagent01 container, you may run the following steps:

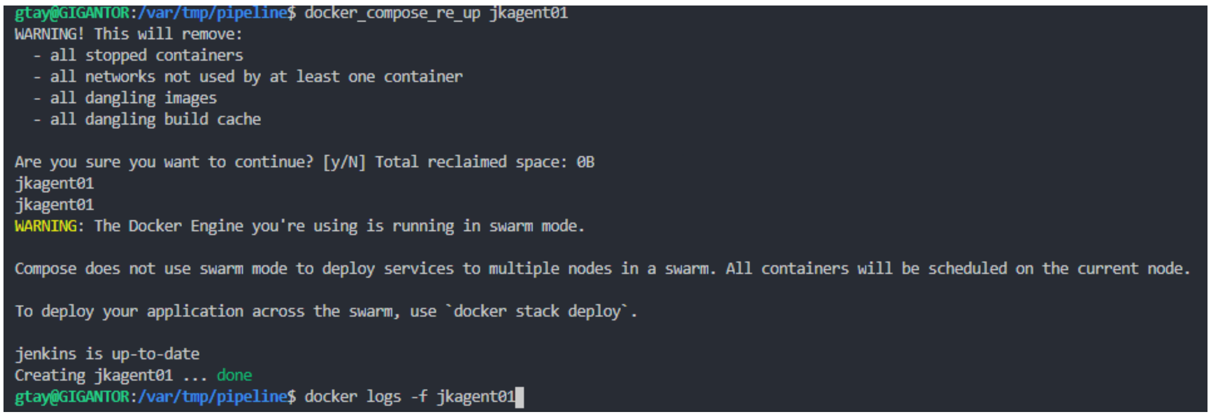
$ **. ./functions && display_shell_function docker_compose_re_up**



```
gtay@GIGANTOR:/var/tmp/pipeline$ . ./functions
gtay@GIGANTOR:/var/tmp/pipeline$ display_shell_function docker_compose_re_up
docker_compose_re_up ()
{
    if [ $# -le 0 ]; then
        echo "Usage: docker_compose_re_up <docker_container_name or docker_compose_service_name>";
        echo "Assumption 1: container_name is the same as sercvice_name>";
        echo "Assumption 2: the current directory has a valid docker-compose.yaml";
        return 1;
    fi;
    local CONTAINER_NAME="$1";
    local SERVICE_NAME="$1";
    echo y | docker system prune;
    docker stop ${CONTAINER_NAME};
    docker rm ${CONTAINER_NAME};
    docker-compose up -d ${SERVICE_NAME}
}
```

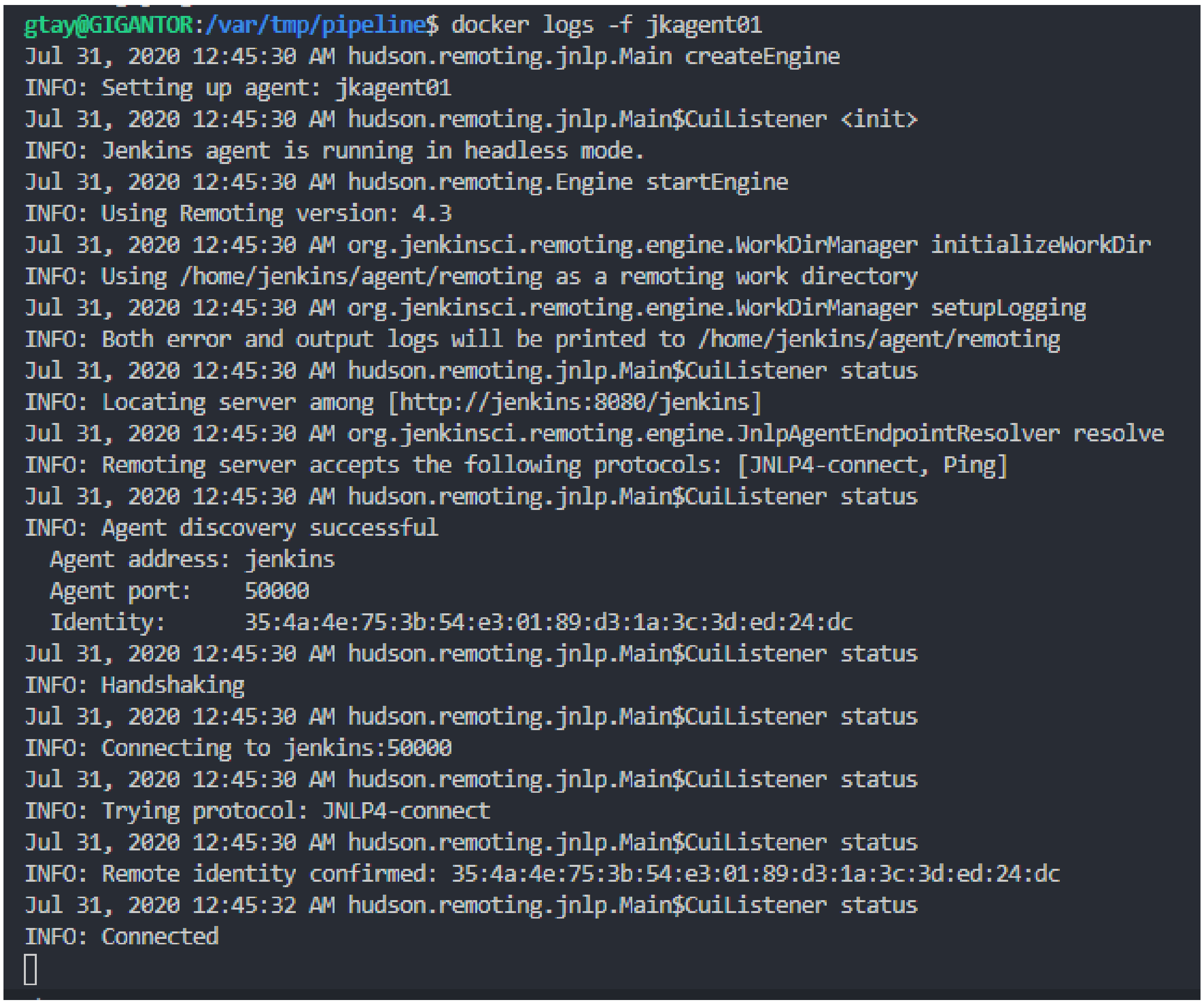
$ **sudo ./fix_jenkins_perms.sh**

$ **docker_compose_re_up jkagent01**



```
gtay@GIGANTOR:/var/tmp/pipeline$ docker_compose_re_up jkagent01
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all dangling images
  - all dangling build cache

Are you sure you want to continue? [y/N] Total reclaimed space: 0B
jkagent01
jkagent01
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

jenkins is up-to-date
Creating jkagent01 ... done
gtay@GIGANTOR:/var/tmp/pipeline$ docker logs -f jkagent01
```

Tail the container log file to check, last message should be ‘INFO: Connected’



```
gtay@GIGANTOR:/var/tmp/pipeline$ docker logs -f jkagent01
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: jkagent01
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Jul 31, 2020 12:45:30 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.3
Jul 31, 2020 12:45:30 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/agent/remoting as a remoting work directory
Jul 31, 2020 12:45:30 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins/agent/remoting
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://jenkins:8080/jenkins]
Jul 31, 2020 12:45:30 AM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
  Agent address: jenkins
  Agent port:    50000
  Identity:      35:4a:4e:75:3b:54:e3:01:89:d3:1a:3c:3d:ed:24:dc
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to jenkins:50000
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Jul 31, 2020 12:45:30 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 35:4a:4e:75:3b:54:e3:01:89:d3:1a:3c:3d:ed:24:dc
Jul 31, 2020 12:45:32 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

To create SSH Agent, first create a SSH-Key-Pair for example by '**ssh-keygen**' with null passphrase, note down its content of private key, assuming you name the keypair file using **jenkins@jkagent02**,

The private key will be jenkins@jkagent02 and the public ley Jenkins@jkagent02.pub

Copy the private key and public key to Notepad++ just in case to have their line feeds properly adjusted.

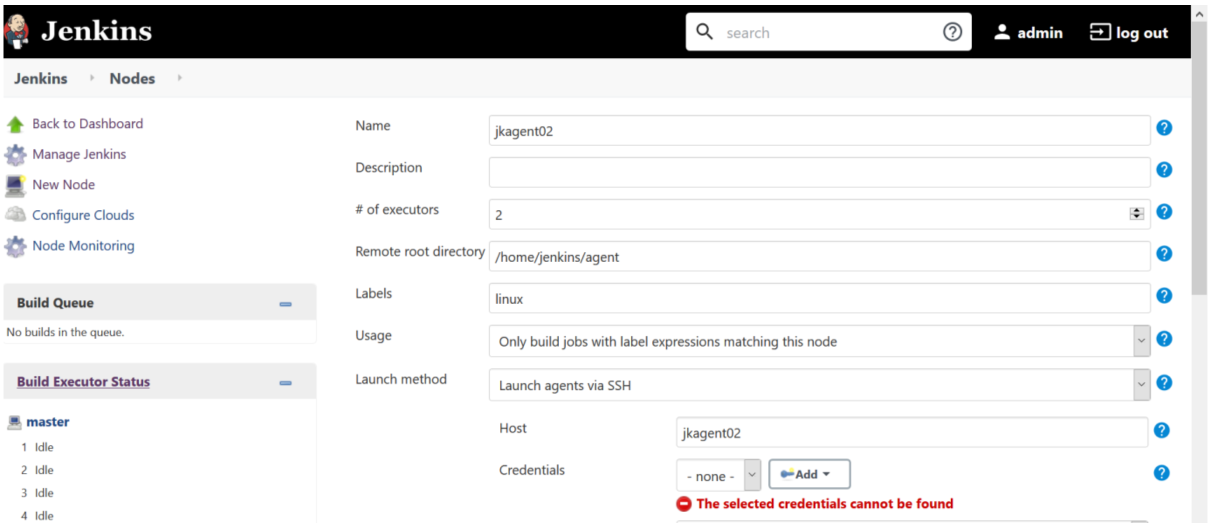
Then for example enter:

Node: **jkagent02**

Click '**Permanent Agent**'

Click '**OK**'

Enter the following parameters as shown:


Jenkins

search

admin

log out

Jenkins › Nodes ›

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

master

1 Idle

2 Idle

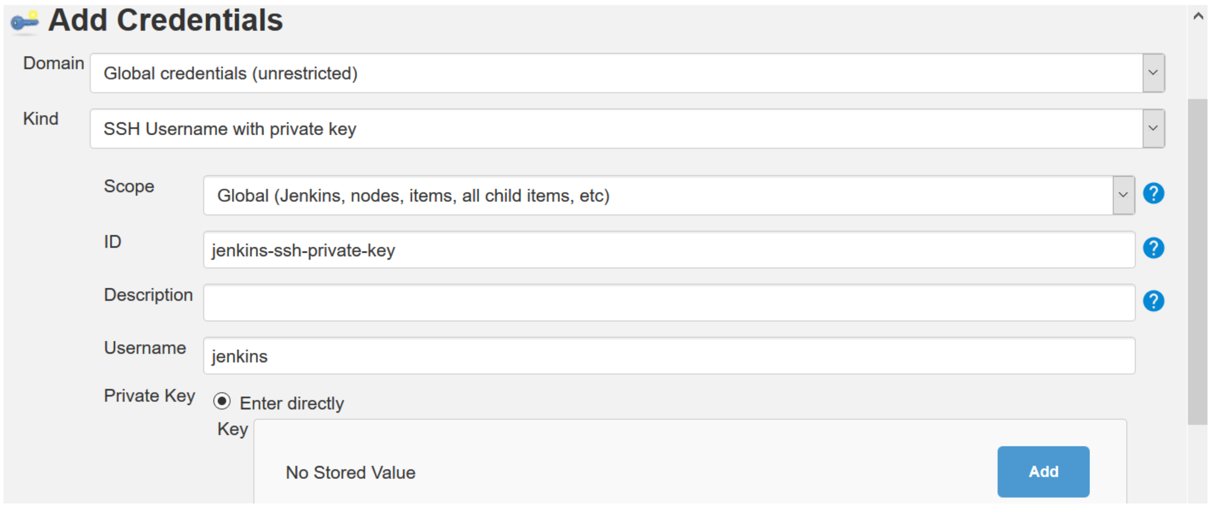
3 Idle

4 Idle

Name	jkagent02
Description	
# of executors	2
Remote root directory	/home/jenkins/agent
Labels	linux
Usage	Only build jobs with label expressions matching this node
Launch method	Launch agents via SSH
Host	jkagent02
Credentials	- none -

The selected credentials cannot be found

Click ‘**Add**’ / ‘**Jenkins**’ to create SSH Private Key Credentials in Jenkins, enter for example:

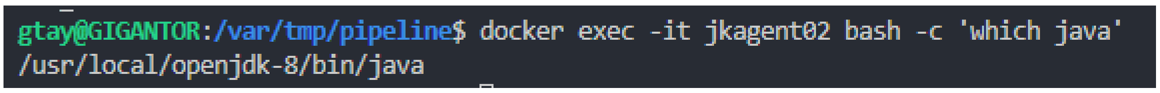

Add Credentials

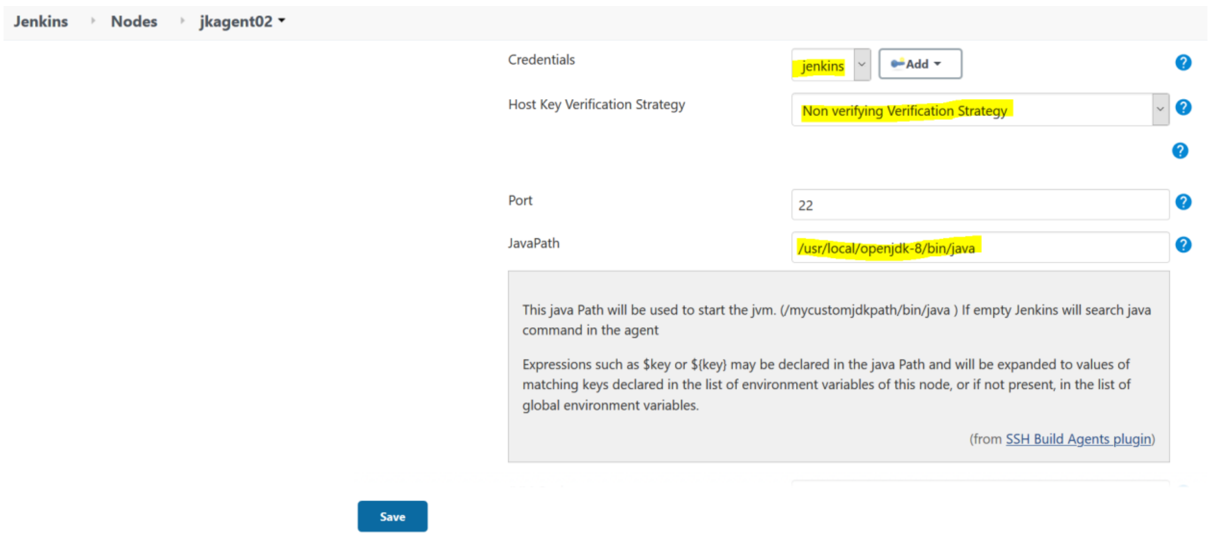
Domain	Global credentials (unrestricted)
Kind	SSH Username with private key
Scope	Global (Jenkins, nodes, items, all child items, etc)
ID	jenkins-ssh-private-key
Description	
Username	jenkins
Private Key	● Enter directly
Key	No Stored Value

Click ‘**Add**’ again, and paste the content of the SSH private key obtained previously.

Click ‘**Add**’ again, click the drop-down to the right of ‘Credentials’, and select ‘**jenkins**’ (Username in above screen), you may wish to select ‘**Non verifying Verification Strategy**’, i.e. not checking /home/jenkins/.ssh/known_hosts file, click ‘Advanced’ and enter JavaPath found via:

$ docker exec -it jkagent02 bash -c ‘which java’


```
gtay@GIGANTOR:/var/tmp/pipeline$ docker exec -it jkagent02 bash -c 'which java'
/usr/local/openjdk-8/bin/java
```


Jenkins › Nodes › jkagent02

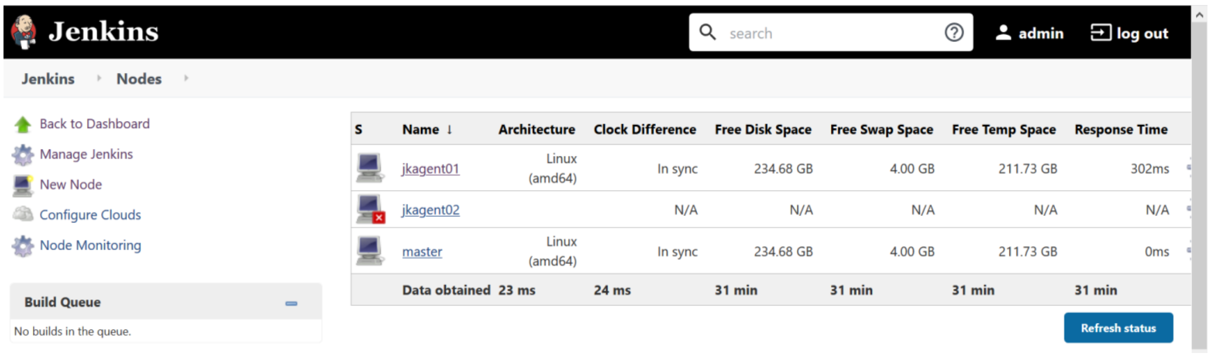
Credentials	jenkins
Host Key Verification Strategy	Non verifying Verification Strategy
Port	22
JavaPath	/usr/local/openjdk-8/bin/java

This java Path will be used to start the jvm. (/mycustomjdkpath/bin/java) If empty Jenkins will search java command in the agent

Expressions such as $key or ${key} may be declared in the java Path and will be expanded to values of matching keys declared in the list of environment variables of this node, or if not present, in the list of global environment variables.

(from SSH Build Agents plugin)

Click ‘**Save**’, you might notice the agent is not up.


Jenkins › Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	jkagent01	Linux (amd64)	In sync	234.68 GB	4.00 GB	211.73 GB	302ms
	jkagent02		N/A	N/A	N/A	N/A	N/A
	master	Linux (amd64)	In sync	234.68 GB	4.00 GB	211.73 GB	0ms
	Data obtained	23 ms	24 ms	31 min	31 min	31 min	31 min

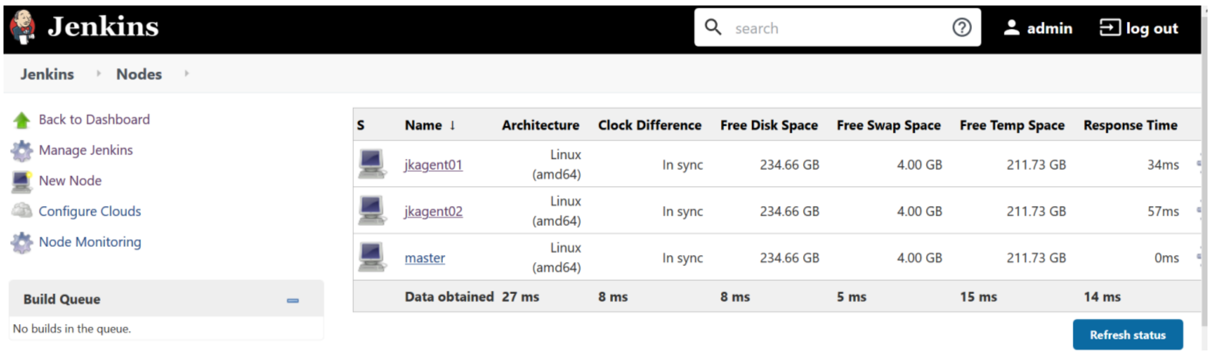
Next copy the public key to the container’s jenkins’ $HOME/.ssh/authorized_keys

$ cat ~/.ssh/jenkins@jkagent02.pub

$ **docker cp ~/.ssh/jenkins@jkagent02.pub jkagent02:/home/jenkins/.ssh/authorized_keys**

$ docker exec -it jkagent02 bash -c 'cat /home/jenkins/.ssh/authorized_keys’

Click the ‘**Launch Agent**’. Both agents should be up now.


Jenkins › Nodes

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	jkagent01	Linux (amd64)	In sync	234.66 GB	4.00 GB	211.73 GB	34ms
	jkagent02	Linux (amd64)	In sync	234.66 GB	4.00 GB	211.73 GB	57ms
	master	Linux (amd64)	In sync	234.66 GB	4.00 GB	211.73 GB	0ms
	Data obtained	27 ms	8 ms	8 ms	5 ms	15 ms	14 ms