

# Using Jenkins agents – On Premise

Generating an SSH key pair

1. In a terminal window run the command: # `ssh-keygen -f ~/.ssh/jenkins_agent_key`
2. Provide a passphrase to use with the key

Just enter

3. Confirm the output looks something like this:

Generating public/private rsa key pair.

Will look very similar to this

```
-bash-4.2$ ssh-keygen -f ~/.ssh/jenkins_agent_key
Generating public/private rsa key pair.
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/jenkins_agent_key.
Your public key has been saved in /var/lib/jenkins/.ssh/jenkins_agent_key.pub.
The key fingerprint is:
SHA256:UR9hc06enSLrELpox1M06TT9Qx36pAPdKYEVxCjdzSc jenkins@ip-10-0-1-37.us-east-2.compute.internal
The key's randomart image is:
+---[RSA 2048]---+
|      .  *+0.o  |
|      . +o= E +. |
|      oo o.o*.. |
|      +.o B +   |
|      *S= * =   |
|      = * * =   |
|      o * . * . |
|      . . .  o   |
|                |
+---[SHA256]-----+
```

4. `cat .ssh/jenkins_agent_key` ( Private key)

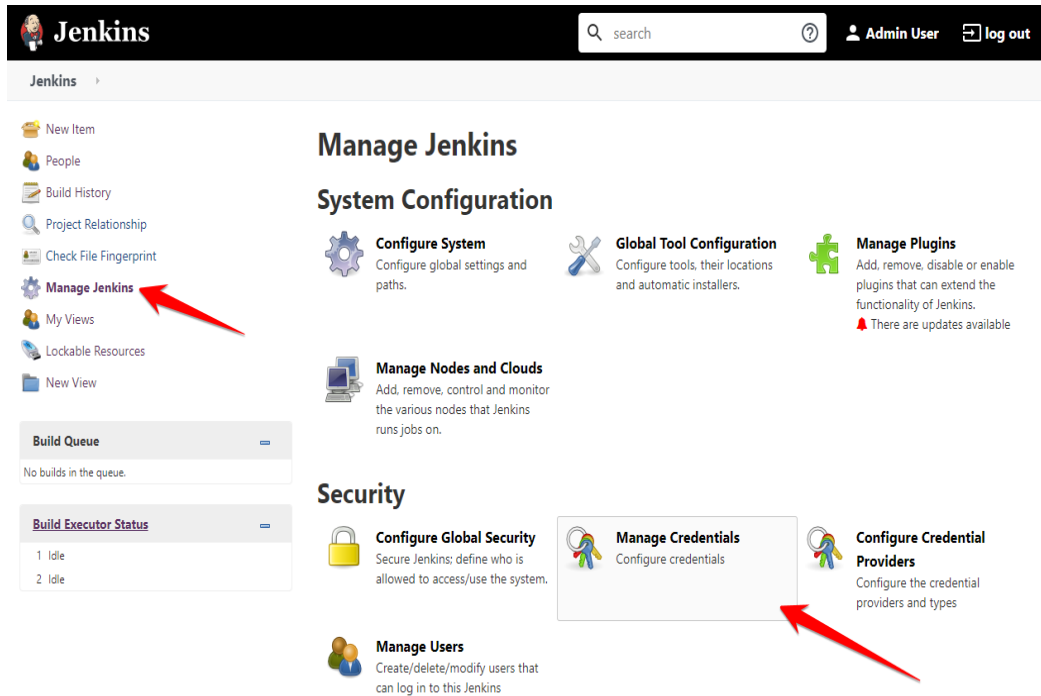
```
-bash-4.2$ cat .ssh/jenkins_agent_key
```

-----BEGIN RSA PRIVATE KEY-----

MIIEpAlBAAKCAQEA18OQ/jlqnFSCkQ471BtcyCbh229xJTnq5zZ2qWPehFUWQvQi  
r5xA7M8P83NmXzdV8LCqYkjm1tfkBeiZ6O5FCbsIFHP6eR2hLfFYc2EzDZG/lBiA  
jBHcFjbi470rNLDnHqGeUgzybsYGHjwFUuAE5QrTurEx8vvX8maF79RSHvzbwhyL  
5Ev+X+6H/0zLOWljUJAbwoBg25XumcTLHxMx3u8LHf9cxVNmOg9be8rdiW/oJSBR  
0l2WmyJMfwPC3gmKleEGU89TZHyEM8UfH4AEwAEWMokunpC8l6PmsoCaZ5/x0MHU  
fqUyMBoMwHoRtkK91zfVgk0sirIkMILsCEfh8wIDAQABAoIBAESKkL3h2Xct9Wc7  
aVAZHSFMEMWXolbUDkZC6/Elk6NwMDE2wuoMdieEI6oPx76JophiVs4AIKyxSnOC  
D5qKTo4nxwpQSYo+Glyw5YtJnmE6lqTWH8ffSI4aA3ubNWV4qQL3Doc8K6t65qim  
ln7QQV3NUbNI5Nsum8waO4QTa78BuOXpaBUUOyv2YuQNyoS/cqEZyIA1jdWF6xwa  
l3jf/OJqGxKPf6OaQIJ1C3+1lbaCJYDeHwoRLLtbLgLrCE+RcCOzuNGKqmgIY/50  
l3jf/OJqGxKPf6OaQIJ1C3+1lbaCJYDeHwoRLLtbLgLrCE+RcCOzuNGKqmgIY/50  
McEYrPSHhApjIzFZ996RKjzDaGI3TsR47KE6EKWu0zLZgieMTnzlNnkyV3qeb3G6  
8wej7eECgYEA+huWyMyZdfqVaSiYZG5kOj/rahLSLDhLWBzOICR5xqQFw6SMzGEO  
tDCerYnDpfkyFR1RaoNTjdtmA7OXTTeunLfLsnpkZxkKWfIbmeTWxZ0sVxGB+8A/  
PAVMNZeq2mKPPfeFil1g+gcafSy9t39clEC9mlogEkCvLWZUy0sEf1ECgYEA3NjZ  
ldSwrEZK7qGTWptbjdfXQlWdrbuBkeBu65EEJ7SPbEgk3DkCv4eyUuSmC5Puoo7E  
OBT127AueiB3J5jlcXoFbyHDmxEW589pdAwjaUxQQe16qumqf+DY94cmVQ64xU/v  
4pvuzCOLrhoSzMkmnqnfWSXSV9y939zI3lJAMCgYEApKVa2u9Nqvv+Jc7ym1LM  
qkf/6sFfGHsqpmO4EhLRE8JQhR2jDrxqGJbEERXa9T0aCipVuX+gloczHgaw+MVK  
LqMT9nd59yDV3ZSX1PR/qnGwidTds+Q3/IQkSjdvxUQStiFnB6DCkXXxqyuz7MNs  
J7CC4MSpzjpe1+dAtN+6ozECgYBP3+IQWVOYy0eYrTmVQLFJSPJiNLFFv5ZqQ0j6  
zBU7oxNob2NdfwUebp6xWI8vus5JXuda/UG6tirplsrVuG1uVpXyldrwy+Mobib  
LzSzhDVVMuX14NM4swLI66ySgOOVAYdaZ1lj3ctA0DNwV0v6zze6o2IQOIRIPWRQ  
RTv1LwKBgQDOheAqPORM+qlsPU+s7wuf40H2LxphCwdEFTRL/9uOatnl/yrSuiS3  
Ox9p/3DpVaVuSUvtQfn5MBtMXBqd8QbUtXg09JYAOFOiFsBgtcaNAbUG1WCTVXnc  
Q4qiN5YekP39IkFW2qSfepHJAq2WSUDuE/UQwCEXzDt1P1pkYCLshA==  
-----END RSA PRIVATE KEY-----

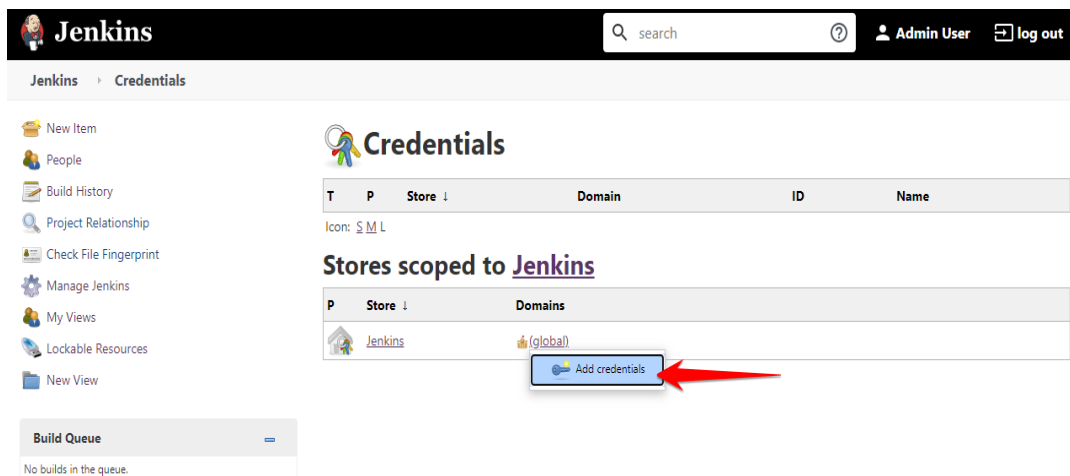
5.

1. Go to your Jenkins dashboard;
2. Go to Manage Jenkins option in main menu and click on credentials button;



The screenshot shows the Jenkins 'Manage Jenkins' page. The left sidebar contains a list of navigation items: New Item, People, Build History, Project Relationship, Check File Fingerprint, **Manage Jenkins** (highlighted with a red arrow), My Views, Lockable Resources, and New View. Below the sidebar are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing 2 idle executors). The main content area is titled 'Manage Jenkins' and 'System Configuration'. It includes several configuration options: 'Configure System' (gear icon), 'Global Tool Configuration' (wrench icon), 'Manage Plugins' (puzzle piece icon, with a note 'There are updates available'), 'Manage Nodes and Clouds' (server icon), 'Configure Global Security' (lock icon), 'Manage Credentials' (key icon, highlighted with a red arrow), 'Configure Credential Providers' (key icon), and 'Manage Users' (people icon). A search bar and user information ('Admin User', 'log out') are at the top.

6. Select the drop option **Add Credentials** from the global item;



The screenshot shows the Jenkins 'Credentials' page. The left sidebar is the same as the previous screenshot, with 'Manage Jenkins' highlighted. The main content area is titled 'Credentials'. It features a table with columns: T, P, Store, Domain, ID, and Name. Below the table, there's a section 'Stores scoped to Jenkins' with a table showing two entries: 'Jenkins' and '(global)'. A red arrow points to a blue button labeled 'Add credentials' located below the '(global)' entry. The top of the page includes the Jenkins logo, a search bar, and user information ('Admin User', 'log out').

- 7.
- Fill the form:
- Kind: SSH Username with private key;
  - id: jenkins
  - description: The jenkins ssh key
  - username: jenkins

- Private Key: select **Enter directly** and press the Add button to insert your **PRIVATE KEY** from `~/.ssh/jenkins_agent_key`
- Passphrase: fill your passphrase used to generate the SSH key pair and then press OK

The screenshot shows the Jenkins 'Add Credentials' form. The breadcrumb trail is 'Jenkins > Credentials > System > Global credentials (unrestricted)'. On the left, there are links for 'Back to credential domains' and 'Add Credentials'. The form fields are: 'Kind' (SSH Username with private key), 'Scope' (Global (Jenkins, nodes, items, all child items, etc.)), 'ID' (jenkins), 'Description' (The jenkins ssh key), 'Username' (jenkins), 'Private Key' (radio button selected for 'Enter directly'), 'Key' (a text area containing a long base64-encoded private key string), and 'Passphrase' (a text field with dots). An 'OK' button is at the bottom.

## Creating your Docker agent

On Linux

Here we will use the [docker-ssh-agent image](#) to create the agent containers.

run the command to start your first agent:

```
# docker run -d --rm --name=agent1 -p 22:22 \
-e "JENKINS_AGENT_SSH_PUBKEY=[your-public-key]" \
jenkins/ssh-agent:alpine
```

- Remember to replace the tag [your-public-key] for your own SSH **public** key.
- Your public key in this example is: `cat ~/.ssh/jenkins_agent_key.pub`

2. Now run the following command to update the container environment:

```
$ VARS1="HOME=|USER=|MAIL=|LC_ALL=|LS_COLORS=|LANG="
```

```
$ VARS2="HOSTNAME=|PWD=|TERM=|SHLVL=|LANGUAGE=|_="
```

```
$ VARS="${VARS1}|${VARS2}"
```

```
$ docker exec agent1 sh -c "env | egrep -v '^(${VARS})' >>  
/etc/environment"
```

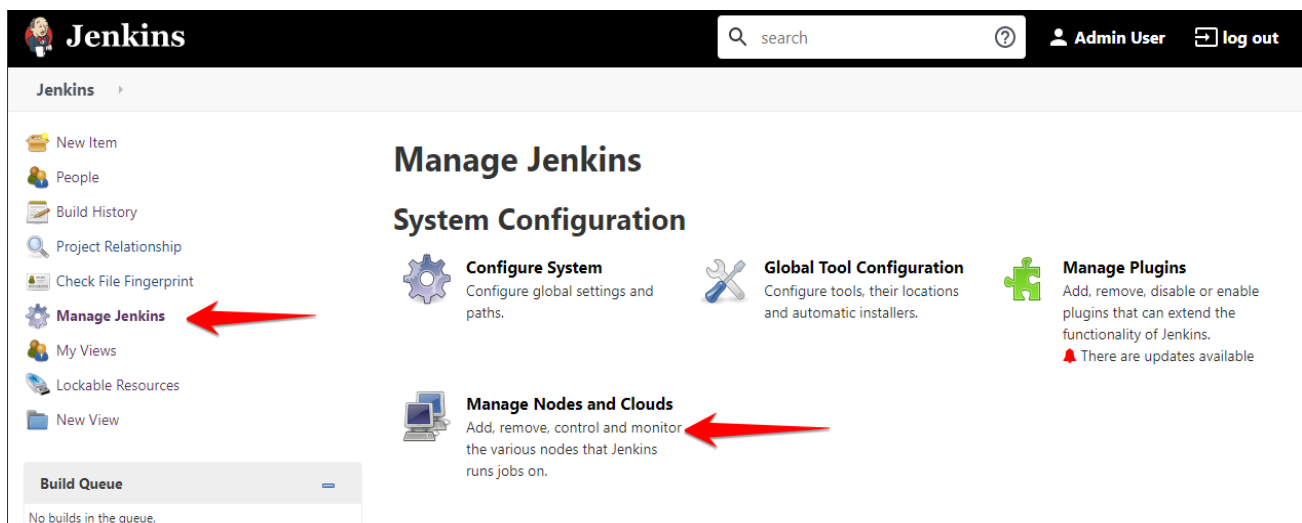
The step 2 is necessary because the image is configured to reject changes to environment variables. when the [issue #33](#) is fixed, we can ignore this step.

3. Now the container **agent1** is running.

Hint: the command **docker ps** can be used to check if the container is running as expected.

## Setup up the agent1 on jenkins.

1. Go to your Jenkins dashboard;
2. Go to **Manage Jenkins** option in main menu;
3. Go to **Manage Nodes and clouds** item;



1. Go to **New Node** option in side menu;

2. Fill the Node/agent name and select the type; (e.g. Name: agent1, Type: Permanent Agent)
3. Now fill the fields:
  - root directory; (e.g.: /home/jenkins ) – simply run **pwd** on your host
  - label; (e.g.: jenkins-agent1 ) – **can be anything**
  - usage; (e.g.: only build jobs with label expression...)
  - Launch method; (e.g.: **Launch agents by SSH** )
    - Host; (e.g.: localhost or your IP address ) – **put your local host ip address**
    - Credentials; (e.g.: jenkins )
    - Host Key Verification Strategy; (e.g.: **Manually trusted key verification ...** )

The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, and user information (Admin User, log out). Below the header is a breadcrumb trail: Jenkins > Nodes >. On the left sidebar, there are links: Back to Dashboard, Manage Jenkins, New Node, Configure Clouds, and Node Monitoring. Below these are two panels: 'Build Queue' showing 'No builds in the queue.' and 'Build Executor Status' showing two 'Idle' executors. The main content area is the 'New Node' configuration form for 'agent1'. The form is enclosed in a red border. It contains the following fields: Name (agent1), Description (empty), # of executors (1), Remote root directory (/home/jenkins), Labels (agent1), Usage (Only build jobs with label expressions matching this node), Launch method (Launch agents via SSH), Host (localhost), Credentials (jenkins (The jenkins ssh key) with an 'Add' button), Host Key Verification Strategy (Manually trusted key Verification Strategy), Require manual verification of initial connection (unchecked), Availability (Keep this agent online as much as possible), and Node Properties (Disable deferred wipeout on this node, Environment variables, Tool Locations, all unchecked). A 'Save' button is at the bottom left of the form, and an 'Advanced...' button is on the right.

Jenkins

Search

Admin User 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**

1 Idle

2 Idle

Name: agent1

Description:

# of executors: 1

Remote root directory: /home/jenkins

Labels: agent1

Usage: Only build jobs with label expressions matching this node

Launch method: Launch agents via SSH

Host: localhost

Credentials: jenkins (The jenkins ssh key) Add

Host Key Verification Strategy: Manually trusted key Verification Strategy

Require manual verification of initial connection: ☐

Advanced...

Availability: Keep this agent online as much as possible

**Node Properties**

☐ Disable deferred wipeout on this node

☐ Environment variables

☐ Tool Locations

Save

**Jenkins**

Admin User
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

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	agent1		N/A	N/A	N/A	N/A	N/A
	master	Linux (amd64)	In sync	234.39 GB	4.00 GB	234.39 GB	0ms
Data obtained			2 ms	2 ms	1 ms	0 ms	0 ms
							55 min

Refresh status

Now press the button **Launch agent** and go to logs, then you should receive the message: **Agent successfully connected and online** on the last log line.

Like this

Jenkins
Nodes
agent1

```

OPTERR=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/local/openjdk-11/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PIPESTATUS=([0]="0")
PPID=170
PS4='+ '
PWD=/home/jenkins
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
SHLVL=1
SSH_CLIENT='172.17.0.1 40120 22'
SSH_CONNECTION='172.17.0.1 40120 172.17.0.3 22'
TERM=dumb
UID=1000
USER=jenkins
_='']'
[07/07/20 23:18:53] [SSH] Checking java version of /home/jenkins/jdk/bin/java
Couldn't figure out the Java version of /home/jenkins/jdk/bin/java
bash: /home/jenkins/jdk/bin/java: No such file or directory

[07/07/20 23:18:53] [SSH] Checking java version of java
[07/07/20 23:18:53] [SSH] java -version returned 11.0.7.
[07/07/20 23:18:53] [SSH] Starting sftp client.
[07/07/20 23:18:53] [SSH] Copying latest remoting.jar...
[07/07/20 23:18:54] [SSH] Copied 1,522,833 bytes.
Expanded the channel window size to 4MB
[07/07/20 23:18:54] [SSH] Starting agent process: cd "/home/jenkins" && java -jar remoting.jar -workDir /home/jenkins -jar-cache /home/jenkins/remoting/jarCache
Jul 08, 2020 2:18:54 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/remoting as a remoting work directory
Jul 08, 2020 2:18:54 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins/remoting
<===[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 4.3
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online

```