

AWS EC2 Jenkins Server and Agent Registration

The purpose of this walkthrough is to demonstrate how a person can install a Jenkins server and register agents on different nodes to be used by Jenkins. There is not a hard requirement of using AWS as the steps for installing the server and agent registration are the same, but there will be some security group firewall issues that will arise with AWS that I will show how to solve.

This isn't a clean cut walkthrough because I wanted to inform you of *why* certain don't work in this area. This is very helpful because these concepts will apply in many different areas.

Create an EC2 instance

This tutorial assumes that you already have an AWS account that can use the EC2 service provided by AWS.

Create an Amazon-Linux EC2 instance.

Ensure that the Amazon Linux OS is installed.

EC2 > Instances > Launch an instance

Launch an instance

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name

Jenkins Server

Add additional tags

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

RecentsQuick Start

Amazon Linux

macOS

Ubuntu

Windows

Red Hat

S

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type

ami-05bfbece1ed5beb54 (64-bit (x86)) / ami-0652d397074720eb1 (64-bit (Arm))

Virtualization: hvm ENA enabled: true Root device type: ebs

Free tier eligible

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20230119.1 x86_64 HVM gp2

Architecture

AMI ID

64-bit (x86)

ami-05bfbece1ed5beb54

Verified provider

Instance type

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0116 USD per Hour

On-Demand SUSE pricing: 0.0116 USD per Hour

Free tier eligible

Compare instance types

Summary

Number of instances

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...read more

ami-05bfbece1ed5beb54

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel

Launch instance

You can keep a majority of the defaults for the machine.

*Note: You'll need to have a public private key pair for use with SSH. You can also set a rule for SSH traffic to allow **only** from your IP.*


2 / 16

▼ **Key pair (login)** [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

AWS-EC2-Key-Pair ▼

 [Create new key pair](#)

▼ **Network settings** [Info](#)

Edit

Network [Info](#)

vpc-fd284b96

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group


☐ Select existing security group

We'll create a new security group called '**launch-wizard-5**' with the following rules:

☒ Allow SSH traffic from

Helps you connect to your instance

My IP



▼

☐ Allow HTTPS traffic from the internet

To set up an endpoint, for example when creating a web server

☐ Allow HTTP traffic from the internet

To set up an endpoint, for example when creating a web server

Now that your instance is set up, you can navigate back to the instances page and gather basic information about the machine.

Instance ID	Public IPv4 address	Private IPv4 addresses
Jenkins Server i-073497aef1b167856	Running 18.117.231.51 open address	2/2 checks passed No alarms us-i
Jenkins Agent 2 i-0dfc774673c4a6166	Running t2.micro	Initializing No alarms us-i

Instance: i-073497aef1b167856 (Jenkins Server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

▼ Instance summary Info

Instance ID i-073497aef1b167856 (Jenkins Server)	Public IPv4 address 18.117.231.51 open address	Private IPv4 addresses 172.31.10.94
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-18-117-231-51.us-east-2.compute.amazonaws.com open address
Hostname type IP name: ip-172-31-10-94.us-east-2.compute.internal	Private IP DNS name (IPv4 only) ip-172-31-10-94.us-east-2.compute.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

Connect to your EC2 jenkins server.

You can see these commands when you first spin up the instance.

In this screenshot, I had made a SSH alias for the machine.

```
C:\Users\Chris Morales>ssh aws-j-server
The authenticity of host 'ec2-18-117-231-51.us-east-2.compute.amazonaws.com (18.117.231.51)' can't be established.
ECDSA key fingerprint is SHA256:+NdyaKgN46tZ4N6W6h3WInko0rZrUHmIEVm+RvzMQTAdiFVIU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-117-231-51.us-east-2.compute.amazonaws.com,18.117.231.51' (ECDSA) to the list of known hosts.

 _ _ _ _ _
| | | | |
|_|_|_|_|_| Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-10-94 ~]$
```

Installing the Jenkins Server

Now that you have a shell, we can now begin installing the jenkins server.

The Jenkins documentation that is provided by Jenkins is what I followed for this guide. The link is [here](#).

Installing the Jenkins Generic Java Package

You can find it [here](#). Just make sure to choose the right distribution; in this case "CentOS/Fedora/Red Hat" will be the correct choice.

You will have to run the following commands

```
sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io.key
```

As noted in the link from above, the command might fail if you have an existing machine with these installed at one point. With this new EC2 instance, you won't have this problem.

```
sudo yum install fontconfig java-11-openjdk
```

You will run into an issue here with the java-11-openjdk file that you wanted to install. You'll have an output that should be the same as the one in the next screenshot. The output will tell you the right command. But I've put it here for your convenience.

```
Installed:
  fontconfig.x86_64 0:2.13.0-4.3.amzn2

Dependency Installed:
  dejavu-fonts-common.noarch 0:2.33-6.amzn2      dejavu-sans-fonts.noarch 0:2.33-6.amzn2
  fontpackages-filesystem.noarch 0:1.44-8.amzn2

Complete!

java-11-openjdk is available in Amazon Linux Extra topic "java-openjdk11"
To use, run
# sudo amazon-linux-extras install java-openjdk11
Learn more at
https://aws.amazon.com/amazon-linux-2/faqs/#Amazon_Linux_Extras
```

```
sudo amazon-linux-extras install java-openjdk11
```

Now that you have this, you can install the jenkins server onto the machine.

```
sudo yum install jenkins
```

Running the Jenkins Server

Now that you have the Jenkins server package installed, you now need to run the server. To find the package, you can use this find command with grep.

```
sudo find / | grep jenkins.war
```

```
[ec2-user@ip-172-31-10-94 ~]$ sudo find / | grep jenkins.war
/usr/share/java/jenkins.war
[ec2-user@ip-172-31-10-94 ~]$
```

Now, you just need to run the the following command to start the server in the background. Replace the path of the .war file with wherever the file is.

```
[ec2-user@ip-172-31-10-94 ~]$ java -jar /usr/share/java/jenkins.war --httpPort=8080&
[1] 4989
```

```
java -jar /usr/share/java/jenkins.war --httpPort=8080&
```

Access the Jenkins Server

All you need to access this is either the public IP or the public DNS of the instance and input this into a web browser using the corresponding port.

Question: Will this work?

Answer: No. Why? Because the security group (essentially firewall) prevents all traffic other than SSH connections from our own IP. So, we need to add a specific rule to allow this HTTP traffic through the port specified.

Creating a HTTP rule in security groups

There are multiple ways that you can find the seecurity group, but the easiest way to find which group is applied to a machine is by selecting the machines from the instances page and then clicking the "Security" tab.

Instance ID
i-073497aef1b167856 (Jenkins Server)

IPv6 address
-

Hostname type
IP name: ip-172-31-10-94.us-east-2.compute.internal

Answer private resource DNS name
IPv4 (A)

Auto-assigned IP address
18.117.231.51 [Public IP]

IAM Role
-

Public IPv4 address
18.117.231.51 | [open address](#)

Instance state
Running

Private IP DNS name (IPv4 only)
ip-172-31-10-94.us-east-2.compute.internal

Instance type
t2.micro

VPC ID
vpc-fd284b96

Subnet ID
subnet-7efd7615

Privat
1

Publi
e

Elast
-

AWS
O

Auto
-

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Security details

IAM Role
-

Owner ID
182790868251

Laun
Thu f

Security groups
sg-0d7571fddcb5dea5f (launch-wizard-3)

▼ Inbound rules

Q Filter rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
-	sgr-0ffdaa622853dd784	22	TCP		launch-wizard-3	-

▼ Outbound rules

Q Filter rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
-	sgr-0b7a6524ec13c418d	All	All	0.0.0.0/0	launch-wizard-3	-

Here, you'll be getting a summary overview of the security rule. Now you can go further in and look at a summarized table of the group.

Security group name: launch-wizard-3

Clear filters

☒

Name

☒

Security group ID

☒

Security group name

☒

VPC ID

☒

Description

☒

Owner

☒

Inbound rules count

☒

Outbound rules count

-	sg-0d7571fddcb5dea5f	launch-wizard-3	vpc-fd284b96	launch-wizard-3 create...	182790868251	1 Permission entry	1 Permission entry
---	----------------------	-----------------	--------------	---------------------------	--------------	--------------------	--------------------

sg-0d7571fddcb5dea5f - launch-wizard-3

Details

Inbound rules

Outbound rules

Tags

Q Filter security group rules

Run Reachability Analyzer

X

Manage tags

Edit inbound rules

< 1 >

<input checked="" type="checkbox"/>	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
<input checked="" type="checkbox"/>	-	sgr-0ffdaa622853dd784	IPv4	SSH	TCP	22		-

Now, you can add a rule.

EC2 > Security Groups > sg-0d7571fddcb5dea5f - launch-wizard-3 > Edit inbound rules

Edit inbound rules

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules

Security group rule ID
sgr-0ffdaa622853dd784

Type
SSH

Protocol
TCP

Port range
22

Source
Custom

Description - optional

Add rule

Cancel

Preview changes

Save rules

7 / 16

You need to add the rule to allow HTTP connections



Press save and then you now have a new rule.

Now, refresh the webpage and you'll be brought to the initial "Unlock Jenkins" page. You can find the password based on what the page tells you.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/home/ec2-user/.jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Set up Jenkins Server Post-Fix

Having gotten the initial configuration page, you can set up your Jenkins server with basic stuff to get started.

1. You can install the recommended default plugins.
2. Set up a super user of choice.
3. Finally, you can update the link, but for now, it's fine to keep it normal.

Now, you have a working Jenkins instance.

Registering Agents

Now you need to register an agent to do the work. You'll be shown the general menu on the left hand side of the screen.

1. Press the "Manage Jenkins" button.

Dashboard >

+ New Item

👤 People

📁 Build History

⚙️ Manage Jenkins

📅 My Views

Build Queue ▼

No builds in the queue.

Build Executor Status ▼

1 Idle

2 Idle

2. Choose "Manage Nodes and Clouds"

System Configuration

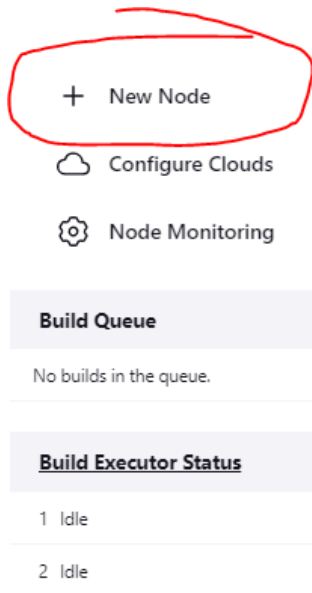
⚙️ **Configure System**
Configure global settings and paths.

🔧 **Global Tool Configuration**
Configure tools, their locations and automatic installers.

🔗 **Manage Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

☁️ **Manage Nodes and Clouds**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

3. Choose "New Node"



Manage nodes and clouds

S	Name ↓	Architecture	Clock Difference	Free Disk Space
	Built-In Node	Linux (amd64)	In sync	6.01 GB
	Data obtained	12 min	12 min	12 min

4. Give it a name and choose "permanent agent for our use case".

New node

Node name





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.

5. Create a new public-private key pair on a machine. Have the private key ready.
6. Back on Jenkins, go back into "Manage Jenkins" and then choose the "Manage Credentials" option.

System Configuration

- | | |
|---|---|
|  Configure System
Configure global settings and paths. |  Global Tool Configuration
Configure tools, their locations and automatic installers. |
|  Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. |  Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on. |
-

Security

- | | |
|---|--|
|  Configure Global Security
Secure Jenkins; define who is allowed to access/use the system. |  Manage Credentials
Configure credentials |
|  Configure Credential Providers
Configure the credential providers and types |  Manage Users
Create/delete/modify users that can log in to this Jenkins. |
-

7. You'll be able to create some new credentials under the dropdown under the "global" option. Choose "add credential" and then you will be brought to another screen. Manually change the type of credential to "SSH Username with Private Key"

New credentials

Kind

SSH Username with private key



Scope ?

Global (Jenkins, nodes, items, all child items, etc)



ID ?

Description ?

Username

☐

Treat username as secret ?

Private Key

☐

Enter directly

Passphrase

Create

8. Fill out the fields with the correct information. Give it any ID. And then make sure that the username is an **actual** user on the target machine. Copy and paste the private key inside of the private key field.

This picture showcases a complete form after the credential was made.

Scope ?

System (Jenkins and nodes only) ▼

ID ?

ssh-agent-key

Description ?

Key used to use nodes as workers.

Username


ec2-user

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

 Concealed for Confidentiality

Replace

Passphrase



Save

9. You can go back to the credetials tab to find your new entry.

Global credentials (unrestricted)

+ Add Credentials

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
 ssh-agent-key	jenkins (Key used to use nodes as workers.)	SSH Username with private key	Key used to use nodes as workers. 

Icon:

S

M

L

10. Go back to the agent creation, now we can fill in the later information.
1. Fill in the information according to how you want to organize and use the agent. Use the hostname or the IP address. In this case, it's another EC2 instance.

Name ?

Agent1

Description ?

Number of executors ?

1

Remote root directory ?

/home/ec2-user

Labels ?

agent1

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agents via SSH

Host ?

ec2-18-219-99-104.us-east-2.compute.amazonaws.com

Credentials ?

ec2-user (Key used to use nodes as workers.)

+ Add

Host Key Verification Strategy ?

Manually trusted key Verification Strategy



Require manual verification of initial connection ?



Advanced...

Save

You'll have to place the appropriate hostname in the "Host" field.

You'll also have to specify the credential key that you want to use.

11. Once you save the agent, the agent launching process will begin. However, the manual verification option in the figure above will require your input. Once you go back, you can press "Status" and a new option that will allow you trust the host key will be there.

Question: Does this work?

Answer: No. You haven't copied the public key to the target machine. And so, you need to update the `authorized_keys` file for the user that you're targeting. Then, you need to make sure that the SSH rule allows your Jenkins server to SSH to your agents.

12. Once you solve the SSH problem, you will run into another problem where the agent doesn't have java installed. You will have to simply run the same command from before on the agent to get java installed.
13. Now you can relaunch the agent and you should be see a connected message.

```
bash: /home/ec2-user/jdk/bin/java: No such file or directory

[02/03/23 05:02:59] [SSH] Checking java version of java
[02/03/23 05:02:59] [SSH] java -version returned 11.0.16.
[02/03/23 05:02:59] [SSH] Starting sftp client.
[02/03/23 05:02:59] [SSH] Copying latest remoting.jar...
[02/03/23 05:02:59] [SSH] Copied 1,368,830 bytes.
Expanded the channel window size to 4MB
[02/03/23 05:02:59] [SSH] Starting agent process: cd "/home/ec2-user" && java -jar remoting.jar -workDir /home/ec2-user -jar-cache /home/ec2-user/remoting/jarCache
Feb 03, 2023 5:03:00 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/remoting as a remoting work directory
Feb 03, 2023 5:03:00 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/remoting
<===[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3077.vd69cf116da_6f
Launcher: SSHLauncher
Communication Protocol: Standard in/out
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
```