

Creating a key pair

Creating a key pair helps ensure that the correct form of authentication is used when you install Jenkins.

To create your key pair:

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/> and sign in.
2. In the navigation pane, under **NETWORK & SECURITY**, select **Key Pairs**.
3. Select **Create key pair**.
4. For **Name**, enter a descriptive name for the key pair. Amazon EC2 associates the public key with the name that you specify as the **key name**. A key name can include up to 255 ASCII characters. It cannot include leading or trailing spaces.
5. For **File format**, select the format in which to save the private key.
 - For OpenSSH compatibility, select **pem**.
 - For PuTTY compatibility, select **ppk**.
6. Select **Create key pair**.
7. The private key file downloads automatically. The base file name is the name you specified as the name of your key pair, and the file name extension is determined by the file format you chose. Save the private key file in a safe place.

This is the only chance for you to save the private key file.

8. If you use an SSH client on a macOS or Linux computer to connect to your Linux instance, run the following command to set the permissions of your private key file so that only you can read it.

```
$ chmod 400 <key_pair_name>.pem
```

If you do not set these permissions, you cannot connect to your instance using this key pair. For more information, refer to [Error: Unprotected private key file](#).

Creating a security group

A security group acts as a firewall that controls the traffic allowed to reach one or more EC2 instances. When you launch an instance, you can assign it one or more security groups. You add rules that control the traffic allowed to reach the instances in each security group. You can modify a security group's rules any time, and the new rules take effect immediately.

For this tutorial, you will create a security group and add the following rules:

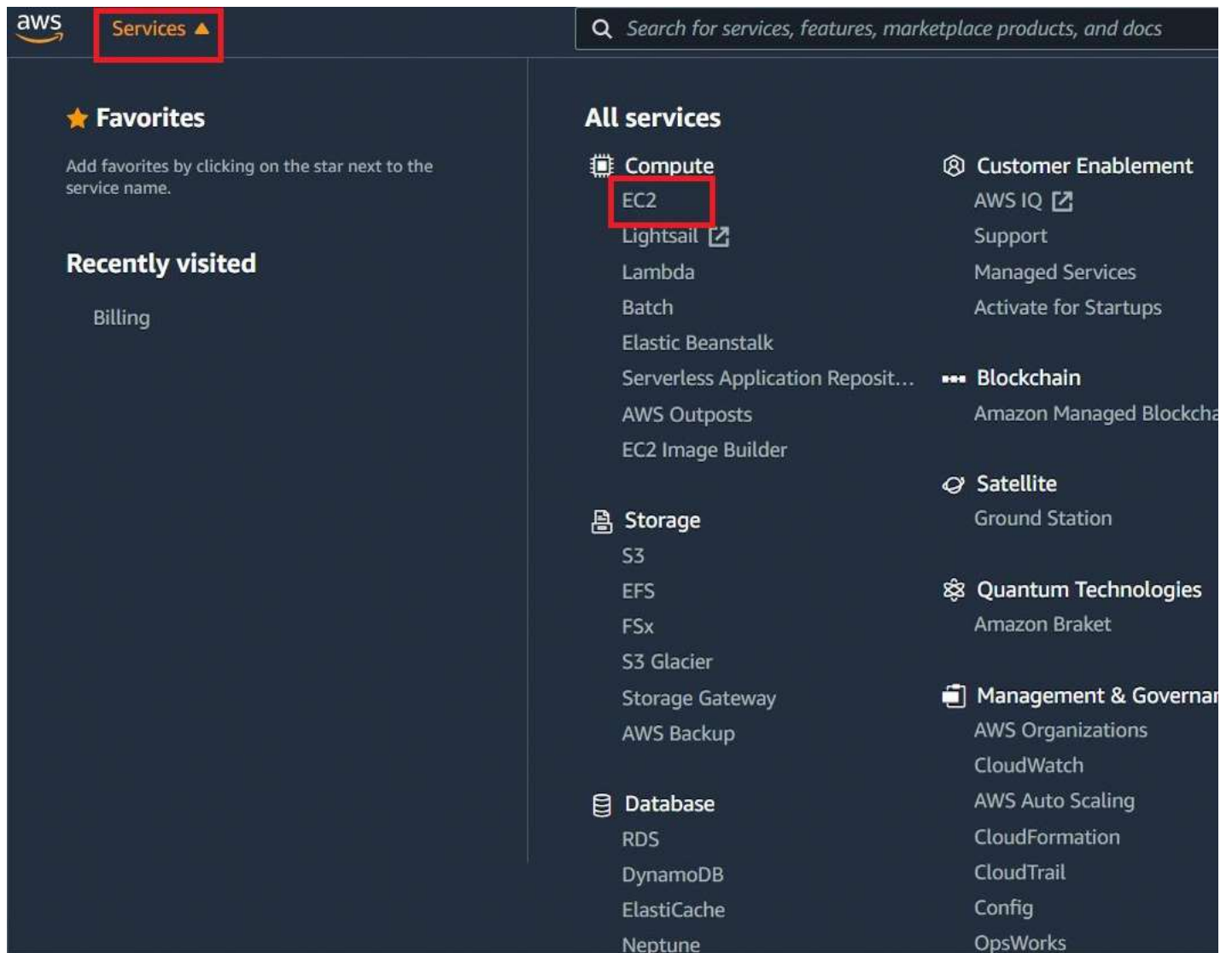
- Allow inbound HTTP access from anywhere.
- Allow inbound SSH traffic from your computer's public IP address so you can connect to your instance.

To create and configure your security group:

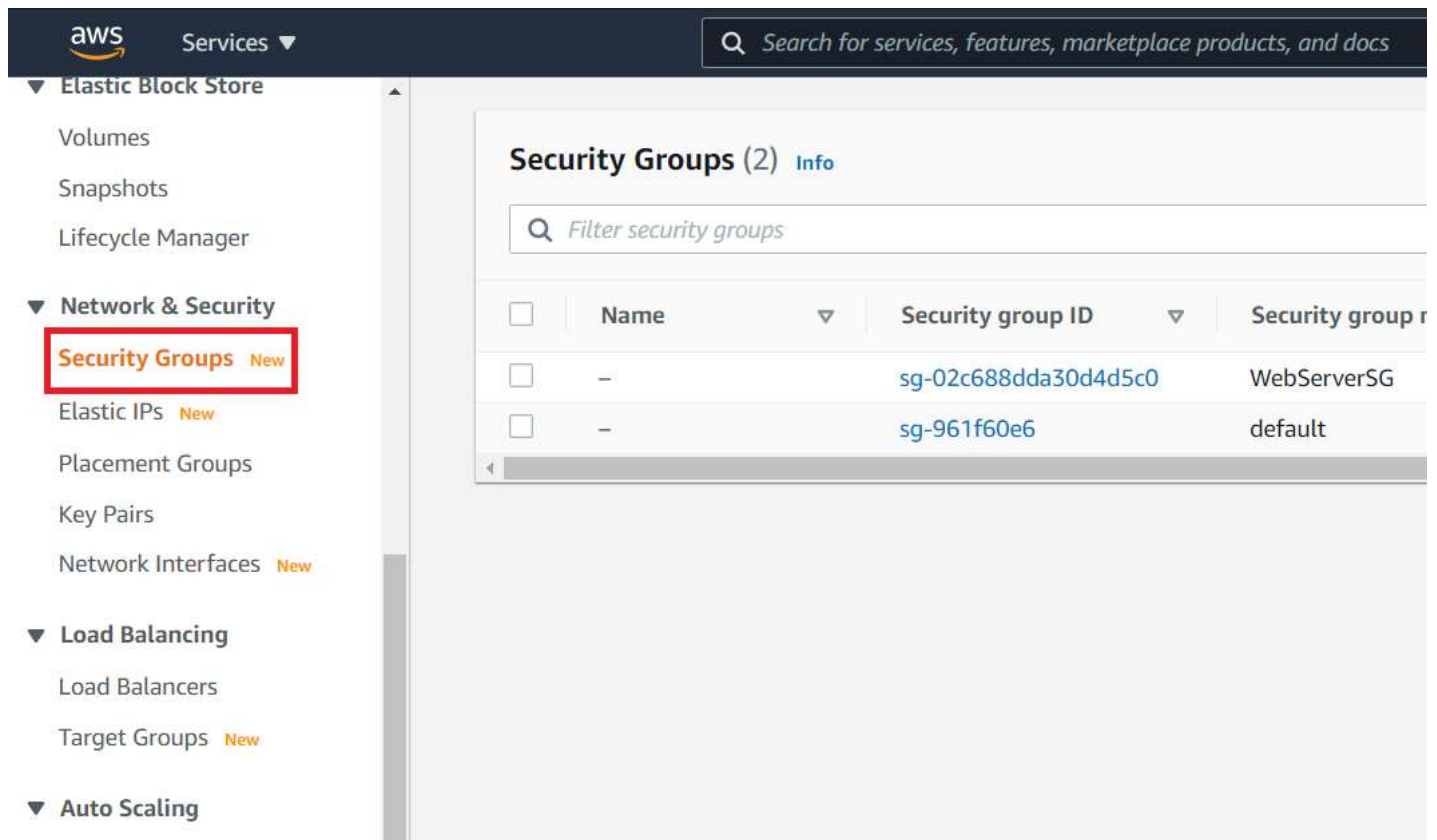
1. Decide who may access your instance. For example, a single computer or all trusted computers on a network. For this tutorial, you can use the public IP address of your computer.
 - To find your IP address, use the [check IP service tool](#) from AWS3 or search for the phrase "what is my IP address" in any search engine.
 - If you connect through an ISP or from behind your firewall without a static IP address, you will need the range of IP addresses used by client computers. If you don't know this address range, you can use 0.0.0.0/0 for this tutorial.

This is unsafe for production environments because it allows everyone to access your instance using SSH.

2. Sign in to the [AWS Management Console](#).
3. Open the Amazon EC2 console by selecting **EC2** under **Compute**.



4. In the left-hand navigation bar, select **Security Groups**, and then select **Create Security Group**.



5. In **Security group name**, enter **WebServerSG** or any preferred name of your choice, and provide a description.
6. Select your VPC from the list. You can use the default VPC.
7. On the **Inbound tab**, add the rules as follows:
 - a. Select **Add Rule**, and then select **SSH** from the Type list.
 - b. Under **Source**, select **Custom**, and in the text box, enter [the IP address from step 1](#), followed by /32 indicating a single IP Address. For example, 104.34.241.123/32 is a single IP address, while 198.51.100.2/24 results in a range of 256 IP addresses.
 - c. Select **Add Rule**, and then select **HTTP** from the Type list.
 - d. Select **Add Rule**, and then select **Custom TCP Rule** from the Type list.
 - e. Under **Port Range**, enter **8080**.
8. Select **Create**.

For more information, refer to [Security Groups](#) in the Amazon EC2 User Guide for Linux Instances.

Launching an Amazon EC2 instance

Now that you have configured a key pair and security group, you can launch an EC2 instance.

To launch an EC2 instance:

1. Sign in to the the [AWS Management Console](#).
2. Open the Amazon EC2 console by selecting EC2 under **Compute**.
3. From the Amazon EC2 dashboard, select **Launch Instance**.

The screenshot displays the AWS Management Console interface for the EC2 service in the US East (Ohio) Region. The left-hand navigation pane includes links to the EC2 Dashboard, Events, Tags, Limits, Instances, Images, and Elastic Block Store. The main content area is titled 'Resources' and lists various EC2 resources and their counts: Instances (running) 0, Elastic IPs 0, Key pairs 0, Placement groups 0, Snapshots 0, Dedicated Hosts, Instances (all states), Load balancers, Security groups, and Volumes. A 'Launch instance' button is highlighted with a red box. The right-hand sidebar shows 'Service health' and 'Region' information, indicating the region is US East (Ohio).

Resources

You are using the following Amazon EC2 resources in the US East (Ohio) Region:

Instances (running)	0	Dedicated Hosts
Elastic IPs	0	Instances (all states)
Key pairs	0	Load balancers
Placement groups	0	Security groups
Snapshots	0	Volumes

Easily size, configure, and deploy Microsoft SQL Server Always On availability group using the Launch Wizard for SQL Server. [Learn more](#)

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch instance

Note: Your instances will launch in the US East (Ohio) Region

Service health

Region: US East (Ohio)

4. The **Choose an Amazon Machine Image (AMI)** page displays a list of basic configurations called Amazon Machine Images (AMIs) that serve as templates for your instance. Select the HVM edition of the **Amazon Linux AMI**.

This configuration is marked **Free tier eligible**.

▼ **Application and OS Images (Amazon Machine Image)** [Info](#)

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

Recents Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S

aws Mac ubuntu Microsoft Red Hat

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 Free tier eligible ▼

ami-09d3b3274b6c5d4aa (64-bit (x86)) / ami-081dc0707789c2daf (64-bit (Arm))

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

Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221004.0 x86_64 HVM gp2

Architecture AMI ID

64-bit (x86) ami-09d3b3274b6c5d4aa Verified provider

5. Scroll down and select the key pair you created in the [creating a key pair](#) section above or any existing key pair you intend to use.

- Select **Select an existing security group**.
- Select the **WebServerSG** security group that you created.
- Select **Launch Instance**.

▼ 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

Select ▼

Create new key pair

▼ Network settings Info

Edit

Network Info

vpc-5d7a3227

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

Common security groups Info

Select security groups ▼

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Summary

Number of Instances Info

1

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI
ami-09d3b3274b6c5d4aa

Virtual server type (instance type)

t2.micro

Firewall (security group)

-

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year, you can run up to 750 hours of t2.micro (or t3.micro) instances on free tier. Regions in which t2.micro is available: US East (N. Virginia), US West (Oregon), Europe (Ireland), Asia Pacific (Singapore), Asia Pacific (Sydney), South America (Sao Paulo), Africa (Cape Town), and Australia (Sydney). For more information, see the Free tier page. 30 GiB of EBS storage, 1 GB of snapshots, and bandwidth to the internet.

Cancel

6. In the left-hand navigation bar, choose **Instances** to view the status of your instance. Initially, the status of your instance is pending. After the status changes to running, your instance is ready for use.

aws Services ▼

Search for services, features, marketplace products, and docs

New EC2 Experience Tell us what you think X

EC2 Dashboard New

Events

Tags

Limits

▼ Instances

Instances New

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts New

Capacity Reservations

Instances (1/1) Info

Filter instances

✓	Name ▼	Instance ID	Instance state ▼	Instance type
✓	-	i-05bb8d08747b18bad	Running	t2.micro

Installing and configuring Jenkins

Now that the Amazon EC2 instance has been launched, Jenkins can be installed properly.

In this step you will deploy Jenkins on your EC2 instance by completing the following tasks:

1. [Connecting to your Linux instance](#)
2. [Downloading and installing Jenkins](#)
3. [Configuring Jenkins](#)

Connecting to your Linux instance

After you launch your instance, you can connect to it and use it the same way as your local machine.

Before you connect to your instance, get the **public DNS** name of the instance using the Amazon EC2 console.

1. Select the instance and locate Public DNS.

The screenshot shows the AWS Management Console interface. On the left is a navigation sidebar with options like EC2 Dashboard, Events, Tags, Limits, and a dropdown for Instances. The main content area shows the 'Instance summary for i-05bb8d08747b18bad'. Key details include: Instance ID (i-05bb8d08747b18bad), Instance state (Running), Instance type (t2.micro), and Public IPv4 address (3.137.170.32). The 'Public IPv4 DNS' field is highlighted with a red box and contains the value 'ec2-3-137-170-32.u2.compute.amazonaws.com'.

If your instance doesn't have a public DNS name, open the VPC console, select the VPC, and check the **Summary** tab. If either DNS resolution or DNS hostnames is **no**, select **Edit** and change the value to **yes**.

Prerequisites

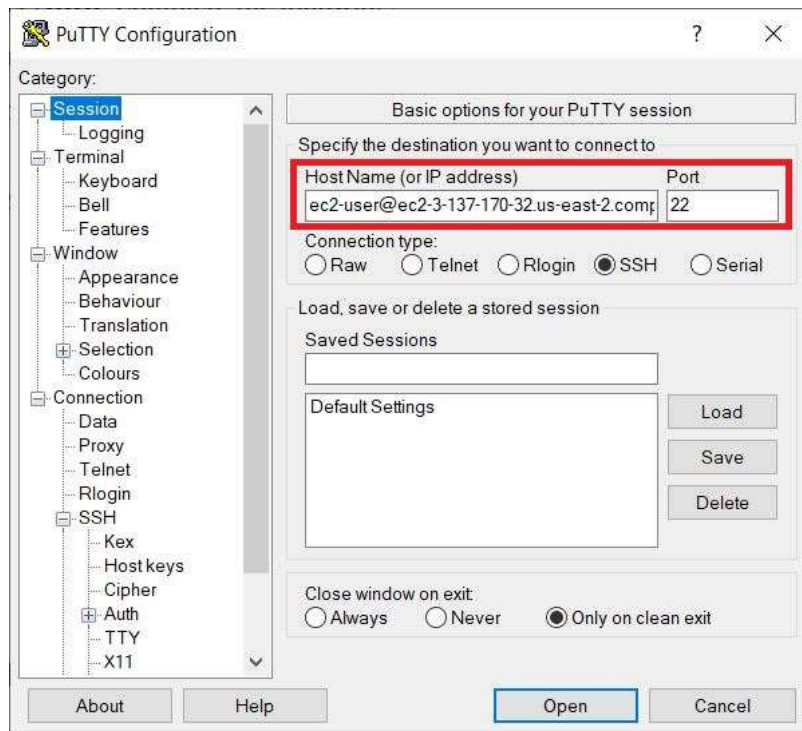
The tool that you use to connect to your Linux instance depends on your operating system.

- If your computer runs Windows, you will connect using PuTTY.
- If your computer runs Linux or Mac OS X, you will connect using the SSH client.

These tools require the use of your key pair. Be sure that you have created your key pair as described in [Creating a key pair](#).

Using PuTTY to connect to your instance

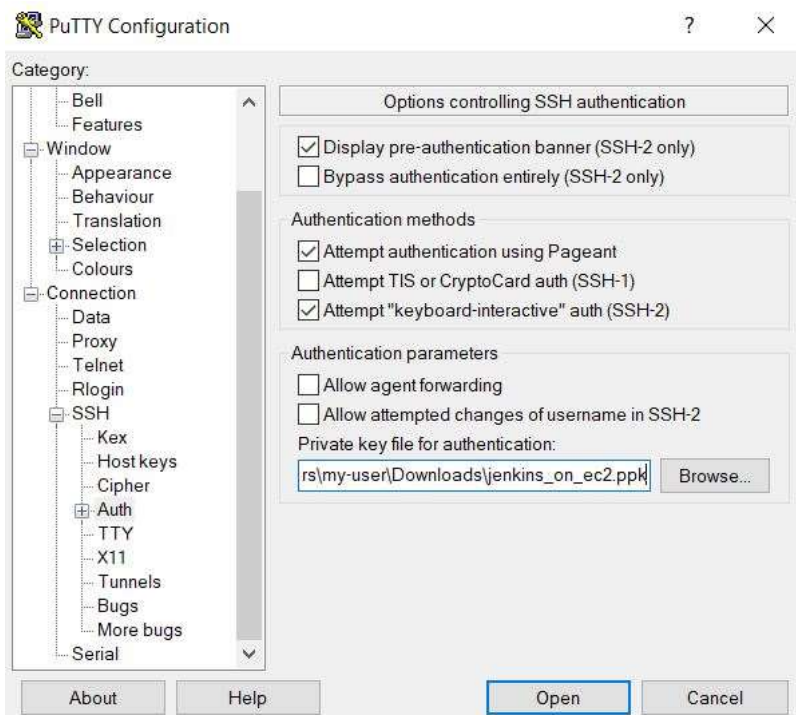
1. From the **Start** menu, select **All Programs > PuTTY > PuTTY**.
2. In the **Category** pane, select **Session**, and complete the following fields:
 - a. In **Host Name**, enter `ec2-user@public_dns_name`.
 - b. Ensure that **Port** is 22.



3. In the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**. Complete the following:

- Select **Browse**.
- Select the .ppk file that you generated for your key pair, as described in [Creating a key pair](#) and then select **Open**.

4. Select **Open** to start the PuTTY session.



Using SSH to connect to your instance

1. Use the ssh command to connect to the instance. You will specify the private key (.pem) file and ec2-user@public_dns_name.

```
$ ssh -i /path/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

You will receive a response like the following:

```
The authenticity of host 'ec2-198-51-100-1.compute1.amazonaws.com (10.254.142.33)' cant be established.
```



```
RSA key fingerprint is 1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
```

```
Are you sure you want to continue connecting  
(yes/no)?
```

2. Enter yes.

You will receive a response like the following:

```
Warning: Permanently added 'ec2-198-51-100-1.compute1.amazonaws.com' (RSA) to the list of known hosts.
```

Downloading and installing Jenkins

Completing the previous steps enables you to download and install Jenkins on AWS. To download and install Jenkins:

1. Ensure that your software packages are up to date on your instance by using the following command to perform a quick software update:

```
[ec2-user ~]$ sudo yum update -y
```

2. Add the Jenkins repo using the following command:

```
[ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \  
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

3. Import a key file from Jenkins-CI to enable installation from the package:

```
[ec2-user ~]$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key  
[ec2-user ~]$ sudo yum upgrade
```

4. Install Java (Amazon Linux 2):

```
[ec2-user ~]$ sudo amazon-linux-extras install java-openjdk11 -y
```

5. Install Java (Amazon Linux 2023):

```
[ec2-user ~]$ sudo dnf install java-11-amazon-corretto -y
```

6. Install Jenkins:

```
[ec2-user ~]$ sudo yum install jenkins -y
```

7. Enable the Jenkins service to start at boot:

```
[ec2-user ~]$ sudo systemctl enable jenkins
```

8. Start Jenkins as a service:

```
[ec2-user ~]$ sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

```
[ec2-user ~]$ sudo systemctl status jenkins
```

Configuring Jenkins

Jenkins is now installed and running on your EC2 instance. To configure Jenkins:

1. Connect to `http://<your_server_public_DNS>:8080` from your browser. You will be able to access Jenkins through its management interface:

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:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

2. As prompted, enter the password found in **/var/lib/jenkins/secrets/initialAdminPassword**.

a. Use the following command to display this password:

```
[ec2-user ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

3. The Jenkins installation script directs you to the **Customize Jenkins page**. Click **Install suggested plugins**.

4. Once the installation is complete, the **Create First Admin User** will open. Enter your information, and then select **Save and Continue**.

Getting Started

Create First Admin User

Username:	<input type="text" value="admin"/>
Password:	<input type="password" value="....."/>
Confirm password:	<input type="password" value="....."/>
Full name:	<input type="text"/>
E-mail address:	<input type="text"/>

Jenkins 2.263.1

[Skip and continue as admin](#)

5. On the left-hand side, select **Manage Jenkins**, and then select **Manage Plugins**.
6. Select the **Available** tab, and then enter **Amazon EC2 plugin** at the top right.
7. Select the checkbox next to **Amazon EC2 plugin**, and then select **Install without restart**.

Plugin Manager

Updates

Available

Installed

Advanced

Install	Name ↓	Re
<input checked="" type="checkbox"/>	Amazon EC2 1.68 Cloud Providers Cluster Management Agent Management spotinst aws This plugin integrates Jenkins with Amazon EC2 or anything implementing the EC2 API's such as an Ubuntu.	31
<input type="checkbox"/>	Amazon Elastic Container Service (ECS) / Fargate 1.41 Cluster Management Agent Management aws Use Amazon EC2 Container Service to provide elastic agents. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	31
<input type="checkbox"/>	Amazon EC2 Container Service plugin with autoscaling capabilities 1.0 Cluster Management Agent Management Use Amazon EC2 Container Service to provide elastic slaves.	61

Install without restart

Download now and install after restart

Update information obtained: 1 hr 51 min ago

Check now

8. Once the installation is done, select **Back to Dashboard**.

9. Select **Configure a cloud** if there are no existing nodes or clouds.

+ New Item

[Add description](#)

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Job Config History

Open Blue Ocean

Lockable Resources

New View

Build Queue

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job



Set up a distributed build

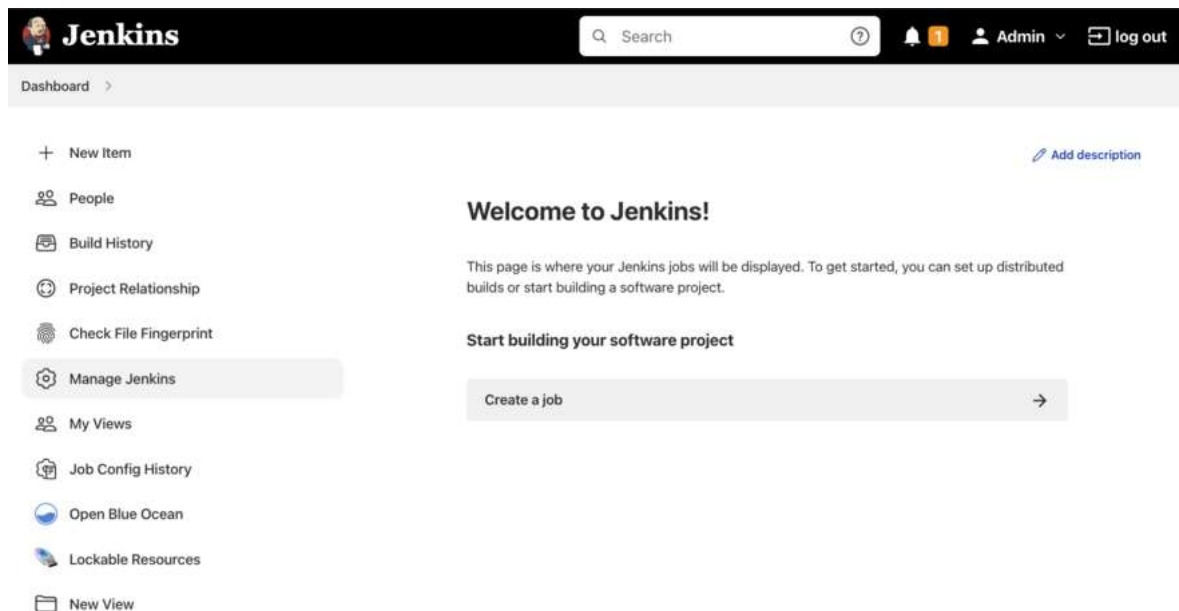
Set up an agent


[Configure a cloud](#)


Learn more about distributed builds

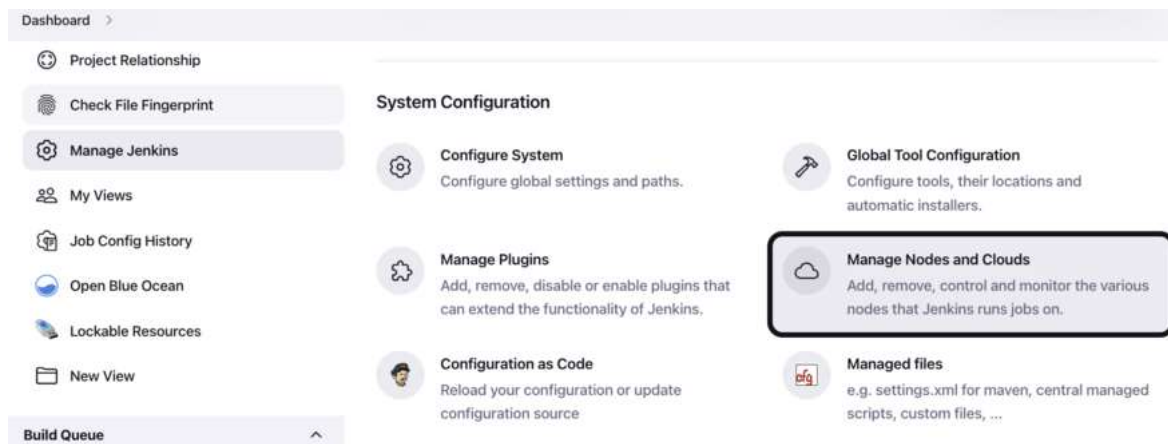


10. If you already have other nodes or clouds set up, select **Manage Jenkins**.



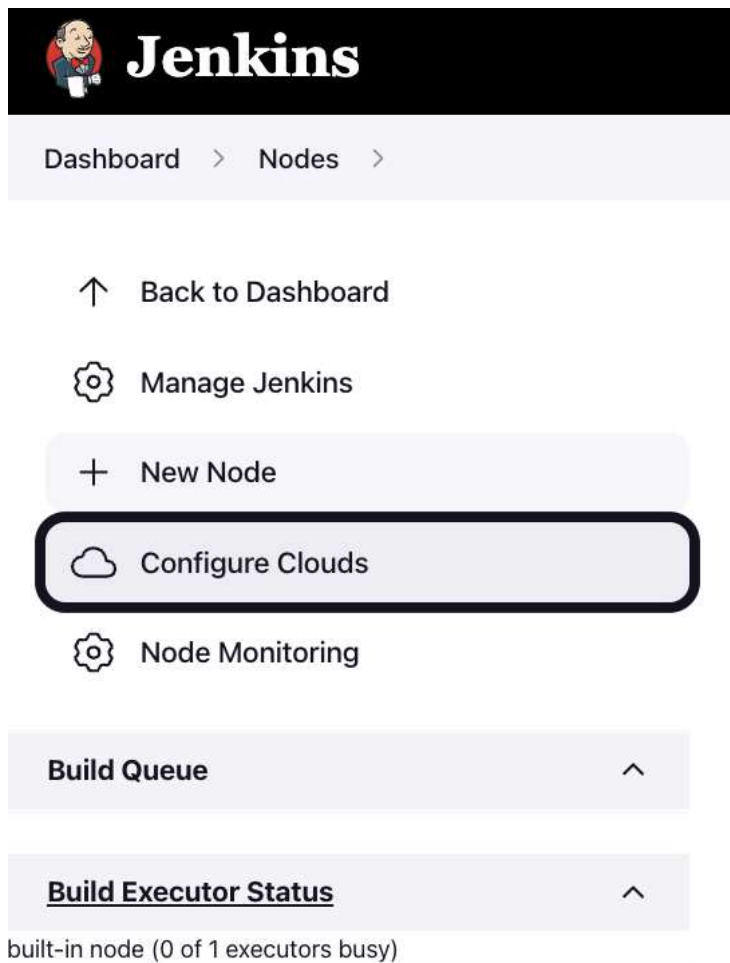
The screenshot shows the Jenkins Dashboard. At the top, there's a header with the Jenkins logo, a search bar, and user information (Admin) with a log out button. Below the header, the main content area is divided into a left sidebar and a main panel. The sidebar contains links to various features: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins (highlighted), My Views, Job Config History, Open Blue Ocean, Lockable Resources, and New View. The main panel displays a 'Welcome to Jenkins!' message, explaining that this is where Jenkins jobs are displayed and providing instructions on how to get started. It also includes a 'Start building your software project' section with a 'Create a job' button.

a. After navigating to **Manage Jenkins**, select **Configure Nodes and Clouds** from the left hand side of the page.



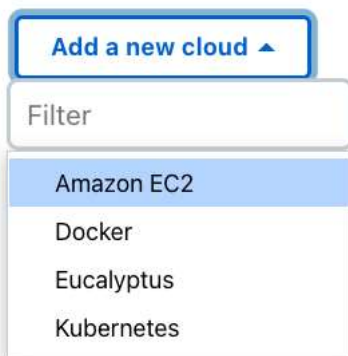
The screenshot shows the 'Manage Jenkins' page, specifically the 'System Configuration' section. The left sidebar is visible, with 'Manage Jenkins' highlighted. The main content area is divided into two columns. The left column contains 'Configure System' (Configure global settings and paths), 'Manage Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), and 'Configuration as Code' (Reload your configuration or update configuration source). The right column contains 'Global Tool Configuration' (Configure tools, their locations and automatic installers), 'Manage Nodes and Clouds' (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), and 'Managed files' (e.g. settings.xml for maven, central managed scripts, custom files, ...). The 'Manage Nodes and Clouds' option is highlighted with a red box.

b. From here, select **Clouds**.




11. Select **Add a new cloud**, and select **Amazon EC2**. A collection of new fields appears.

Configure Clouds



12. Click **Add** under Amazon EC2 Credentials



Jenkins

Dashboard > Configure Clouds

↑ Back to Dashboard

⚙️ Manage Nodes

Configure Clouds

Amazon EC2

Name

! No name is specified

Amazon EC2 Credentials ?

AWS IAM Access Key used to connect to EC2. If not specified, implicit authentication mechanisms are used (l

- none -

+ Add

☐ Use EC2 instance profile to obtain credentials ?

- a. From the Jenkins Credentials Provider, select AWS Credentials as the **Kind**.

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

AWS Credentials

Scope ?

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

ID ?

- b. Scroll down and enter in the IAM User programmatic access keys with permissions to launch EC2 instances and select **Add**.

ID ?

Description ?

Access Key ID ?

Secret Access Key

IAM Role Support

Advanced...

Add Cancel

☐ Use EC2 instance profile to obtain credentials ?

Alternate EC2 Endpoint

Used to populate the available regions dropdown. Only set this if you're using a different EC2 endpoint (i.e. operating in govcloud).

The regions will be populated once the keys above are entered.

Region ?

us-east-1

EC2 Key Pair's Private Key ?

- none -

+ Add

No ssh credentials selected

c. Scroll down to select your region using the drop-down, and select **Add** for the EC2 Key Pair's Private Key.

d. From the Jenkins Credentials Provider, select SSH Username with private key as the Kind and set the Username to ec2-user.

Add Credentials

Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

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

ID ?

Description ?

Username

ec2-user

c. Scroll down and select **Enter Directly** under Private Key, then select **Add**.

Private Key

☒ Enter directly

Key

No Stored Value

Passphrase

Add Cancel

f. Open the private key pair you created in the [creating a key pair](#) step and paste in the contents from "-----BEGIN RSA PRIVATE KEY-----" to "-----END RSA PRIVATE KEY-----". Select **Add** when completed.

Private Key

☒ Enter directly

Key

Passphrase

Add Cancel

g. Scroll down to "Test Connection" and ensure it states "Success". Select **Save** when done

Success

AMIs

List of AMIs to be launched as agents

Add

Add a new cloud ▾

Save Apply

You are now ready to use EC2 instances as Jenkins agents.

Cleaning up

After completing this tutorial, be sure to delete the AWS resources that you created so you do not continue to accrue charges.

Deleting your EC2 instance

1. In the left-hand navigation bar of the Amazon EC2 console, select **Instances**.
2. Right-click on the instance you created earlier, and select **Terminate**.

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo and a search bar. Below this, the left sidebar contains a menu with categories like 'New EC2 Experience', 'EC2 Dashboard', 'Events', 'Tags', 'Limits', 'Instances', 'Images', and 'Elastic Block Store'. The 'Instances' category is expanded, showing sub-links like 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', and 'Capacity Reservations'. The main content area is titled 'Instances (1/1)' and includes a search bar and a filter for 'Instance state: running'. A table lists instances, with one instance selected. A context menu is open for this instance, showing options like 'Launch instances', 'Launch instance from template', 'Connect', 'Stop instance', 'Start instance', 'Reboot instance', 'Hibernate instance', 'Terminate instance' (highlighted), 'Instance settings', 'Networking', 'Security', 'Image and templates', and 'Monitor and troubleshoot'.

Improve this page Report page issue



The content driving this site is licensed under the Creative Commons Attribution-ShareAlike 4.0 license.

Resources

[Downloads](#)
[Blog](#)
[Documentation](#)
[Plugins](#)
[Security](#)
[Contributing](#)

Project

[Structure and governance](#)
[Issue tracker](#)
[Roadmap](#)
[GitHub](#)
[Jenkins on Jenkins](#)

Community