Jenkins on AWS

Table of Contents

- Prerequisites
- Creating a key pair
- Creating a security group
- Launching an Amazon EC2 instance
- Installing and configuring Jenkins
  - Connecting to your Linux instance
    - Prerequisites
    - Using PuTTY to connect to your instance
    - Using SSH to connect to your instance
  - Downloading and installing Jenkins
  - Configuring Jenkins
- Cleaning up
  - Deleting your EC2 instance

Jenkins is an open-source automation server that integrates with a number of AWS Services, including: AWS CodeCommit, AWS CodeDeploy, Amazon EC2 Spot, and Amazon EC2 Fleet. You can use Amazon Elastic Compute Cloud (Amazon EC2) to deploy a Jenkins application on AWS.

This tutorial walks you through the process of deploying a Jenkins application. You will launch an EC2 instance, install Jenkins on that instance, and configure Jenkins to automatically spin up Jenkins agents if build abilities need to be augmented on the instance.

In this tutorial, you will perform the following steps:

1. Prerequisites.
2. Create a key pair using Amazon EC2. If you already have one, you can skip to step 3.
3. Create a security group for your Amazon EC2 instance. If you already have one, you can skip to step 4.
4. Launch an Amazon EC2 instance.
5. Install and configure Jenkins.
6. Clean up tutorial resources.

Prerequisites

1. An **AWS account**. If you don't have one, you can register here.
2. An Amazon EC2 key pair. If you don't have one, refer to Creating a key pair.
3. An AWS IAM User with programmatic key access and permissions to launch EC2 instances

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.

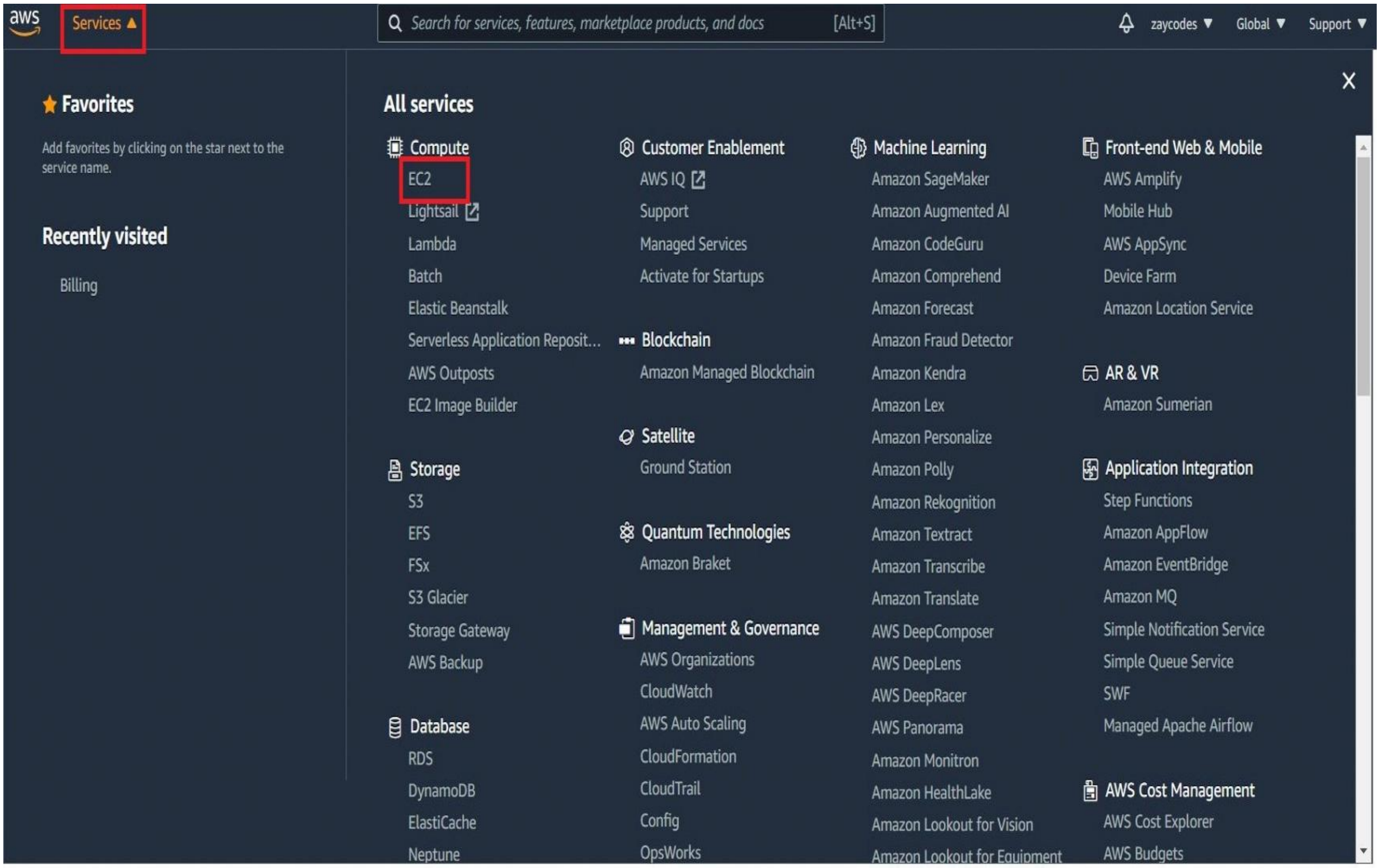
To create and configure your security group:

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.

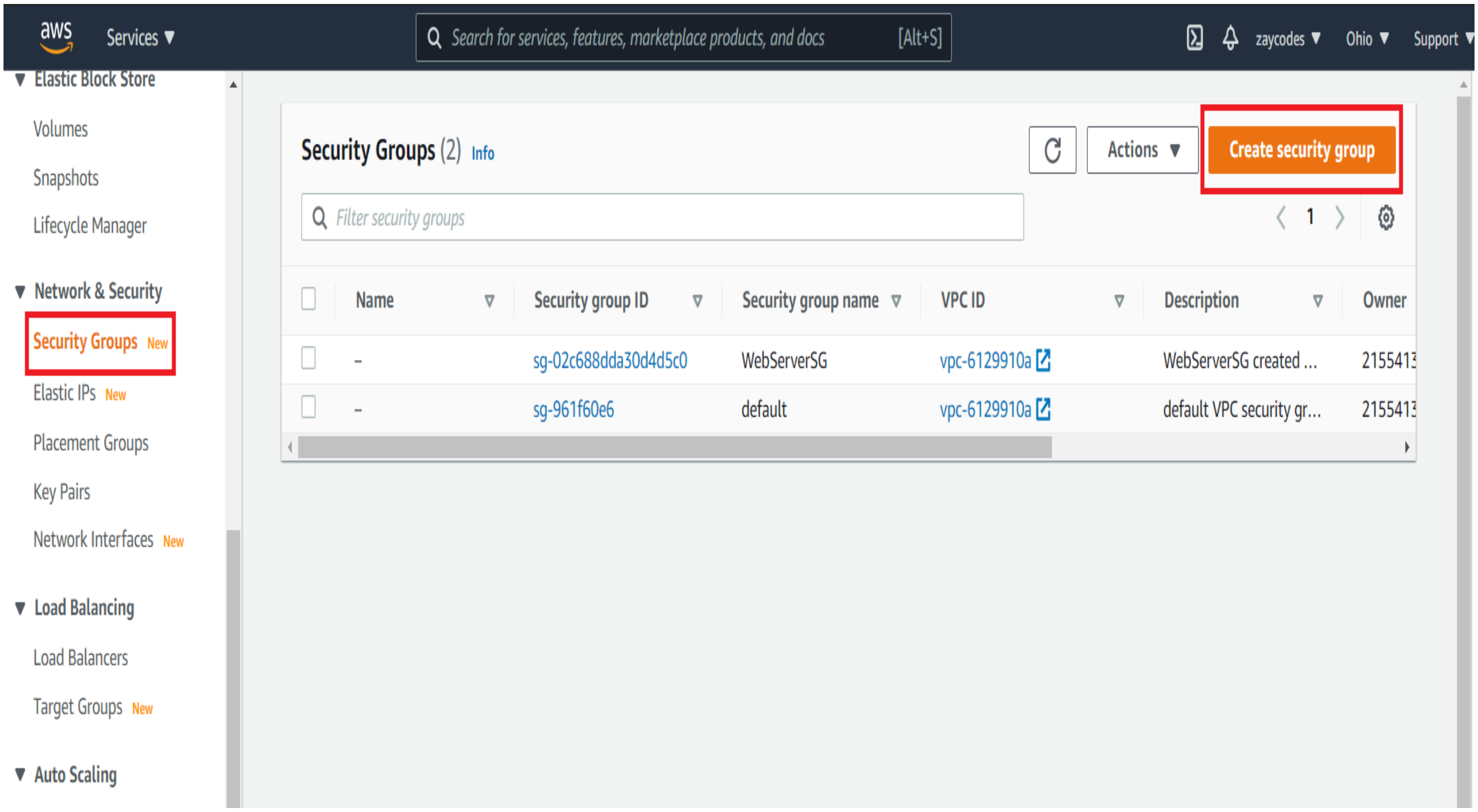
  - To find your IP address, use the check IP service tool from AWS3 or search for the phrase "what is my IP address" in any search engine.
  - If you connect through an ISP or from behind your firewall without a static IP address, you will need the range of IP addresses used by client computers. If you 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**.



□	Name	Security group ID	Security group name	VPC ID	Description	Owner
□	-	sg-02c688dda30d4d5c0	WebServerSG	vpc-6129910a	WebServerSG created ...	2155413
□	-	sg-961f60e6	default	vpc-6129910a	default VPC security gr...	2155413

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.

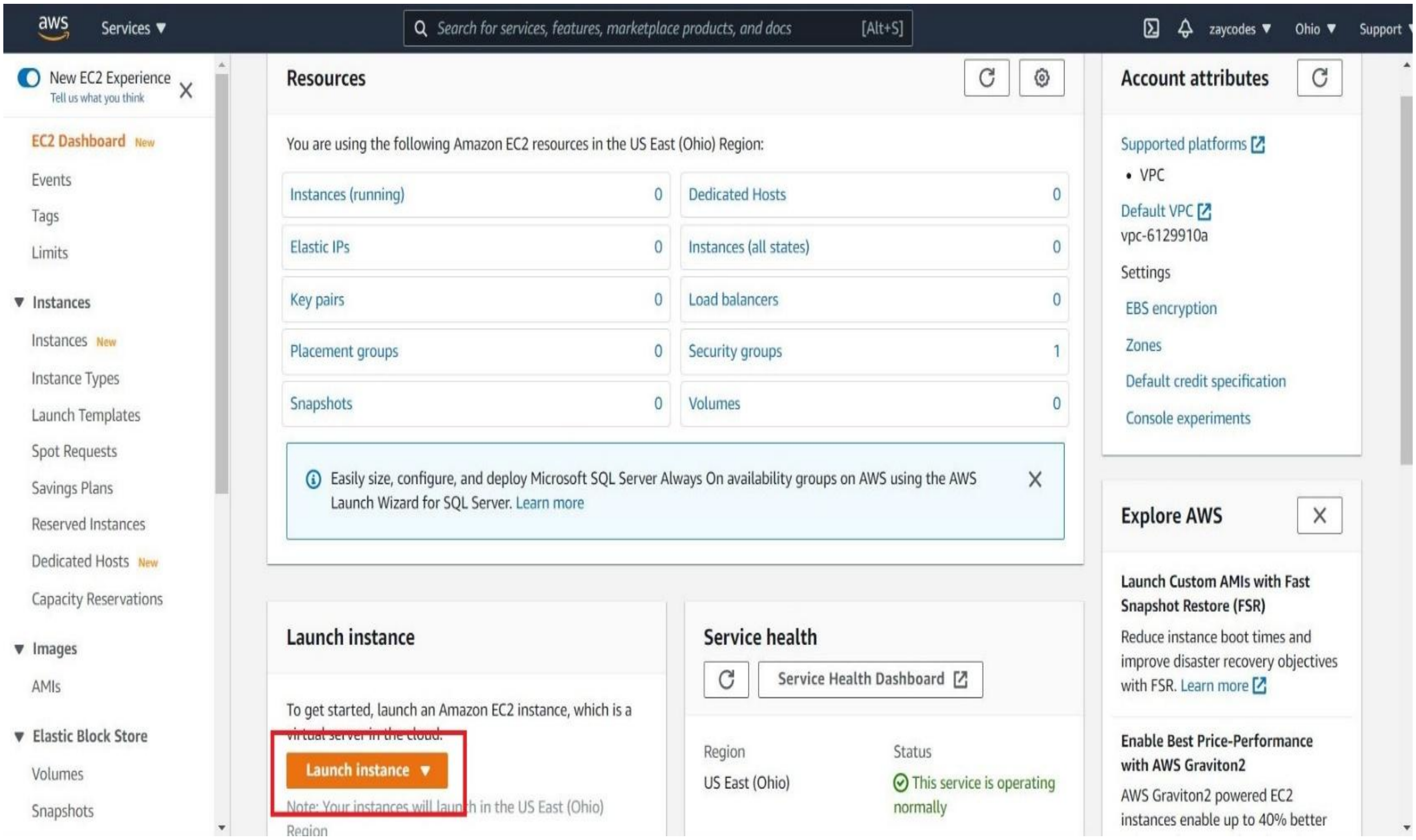
For more information, refer to Security Groups in the Amazon EC2 User Guide for Linux Instances.

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**.



Resources

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

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances (all states)	0
Key pairs	0	Load balancers	0
Placement groups	0	Security groups	1
Snapshots	0	Volumes	0

Easily size, configure, and deploy Microsoft SQL Server Always On availability groups on AWS using the AWS 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.

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

Service health

Region	Status
US East (Ohio)	This service is operating normally

Account attributes

Supported platforms
- VPC

Default VPC
vpc-6129910a

Explore AWS

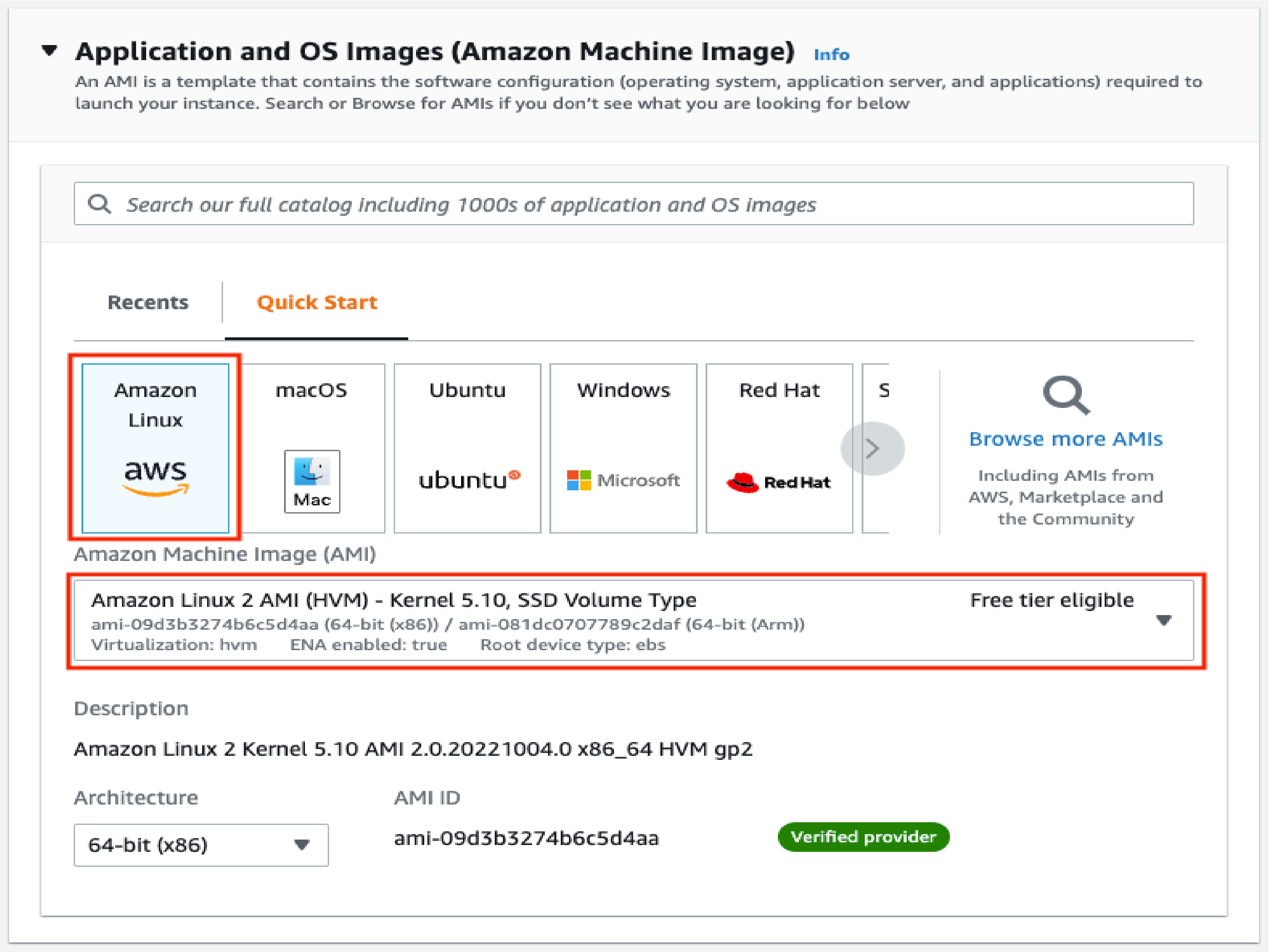
Launch Custom AMIs with Fast Snapshot Restore (FSR)
Reduce instance boot times and improve disaster recovery objectives with FSR. Learn more

Enable Best Price-Performance with AWS Graviton2
AWS Graviton2 powered EC2 instances enable up to 40% better

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**.

5.



▼ 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
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
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 — Free tier eligible
ami-09d3b3274b6c5d4aa (64-bit (x86)) / ami-081dc0707789c2daf (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

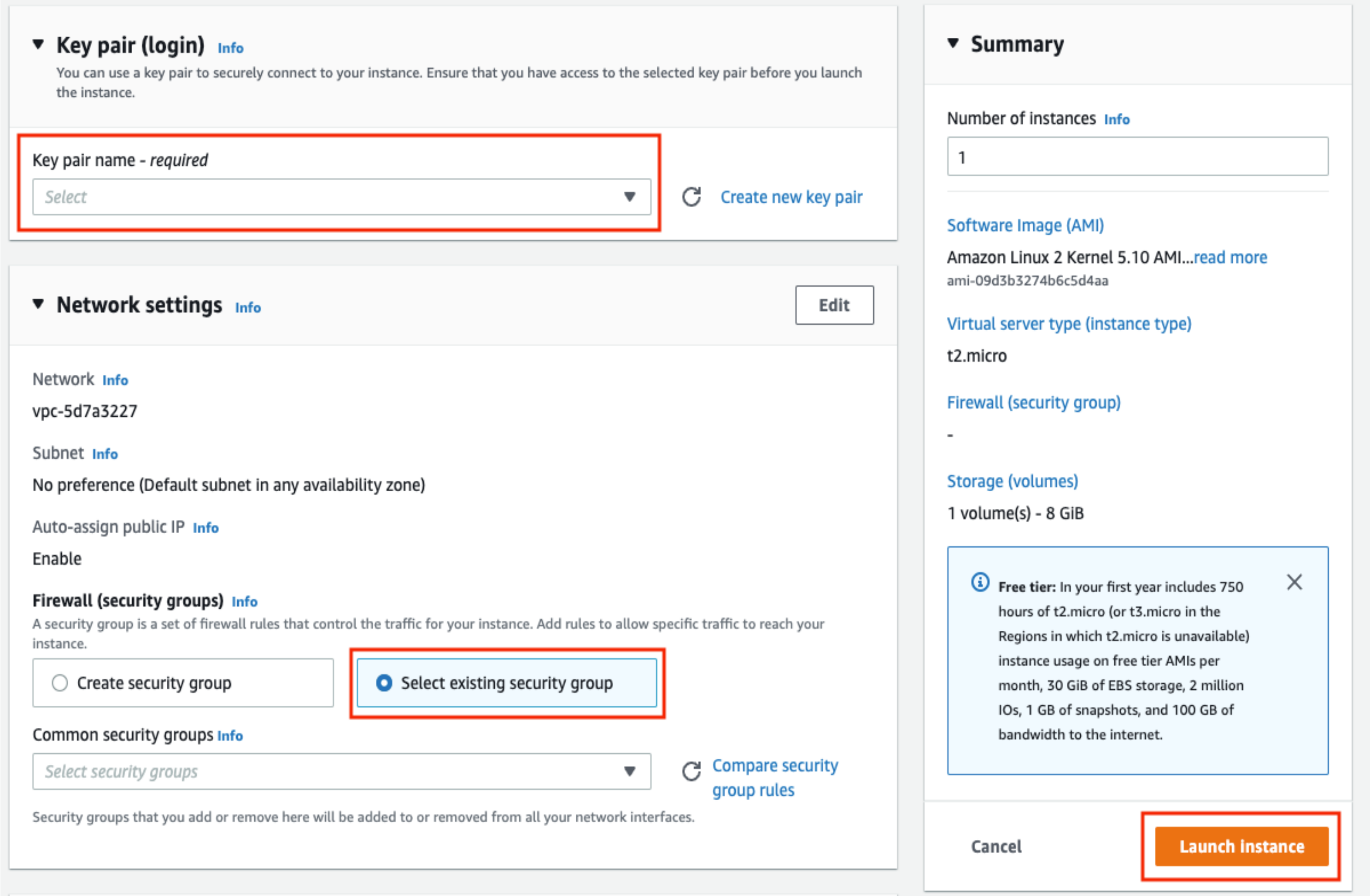
Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20221004.0 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-09d3b3274b6c5d4aa

6. 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.
   a. Select **Select an existing security group**.
   b. Select the **WebServerSG** security group that you created.
   c. 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...read more
ami-09d3b3274b6c5d4aa

Virtual server type (instance type)
t2.micro

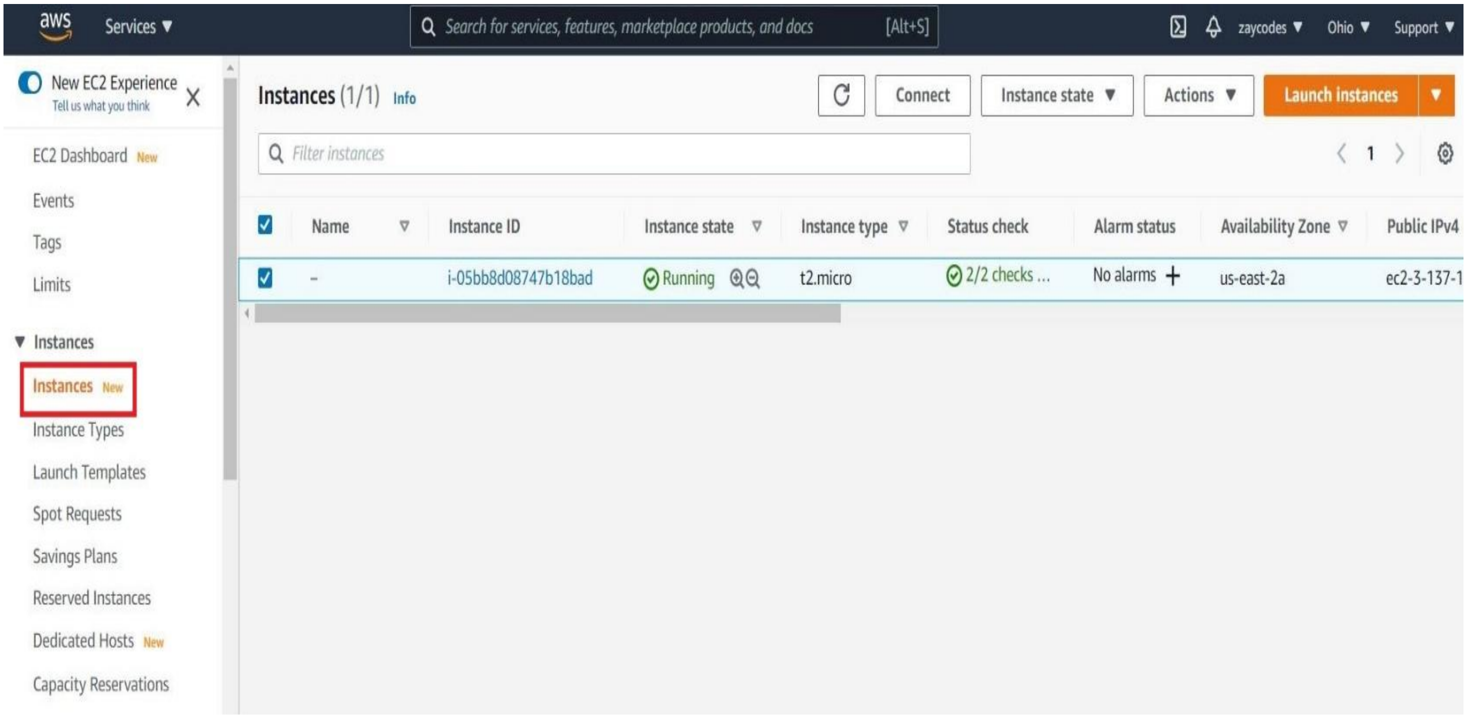
Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year 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

7. 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.



Instances (1/1) Info

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☑	–	i-05bb8d08747b18bad	Running	t2.micro	2/2 checks ...	No alarms +	us-east-2a	ec2-3-137-1

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:

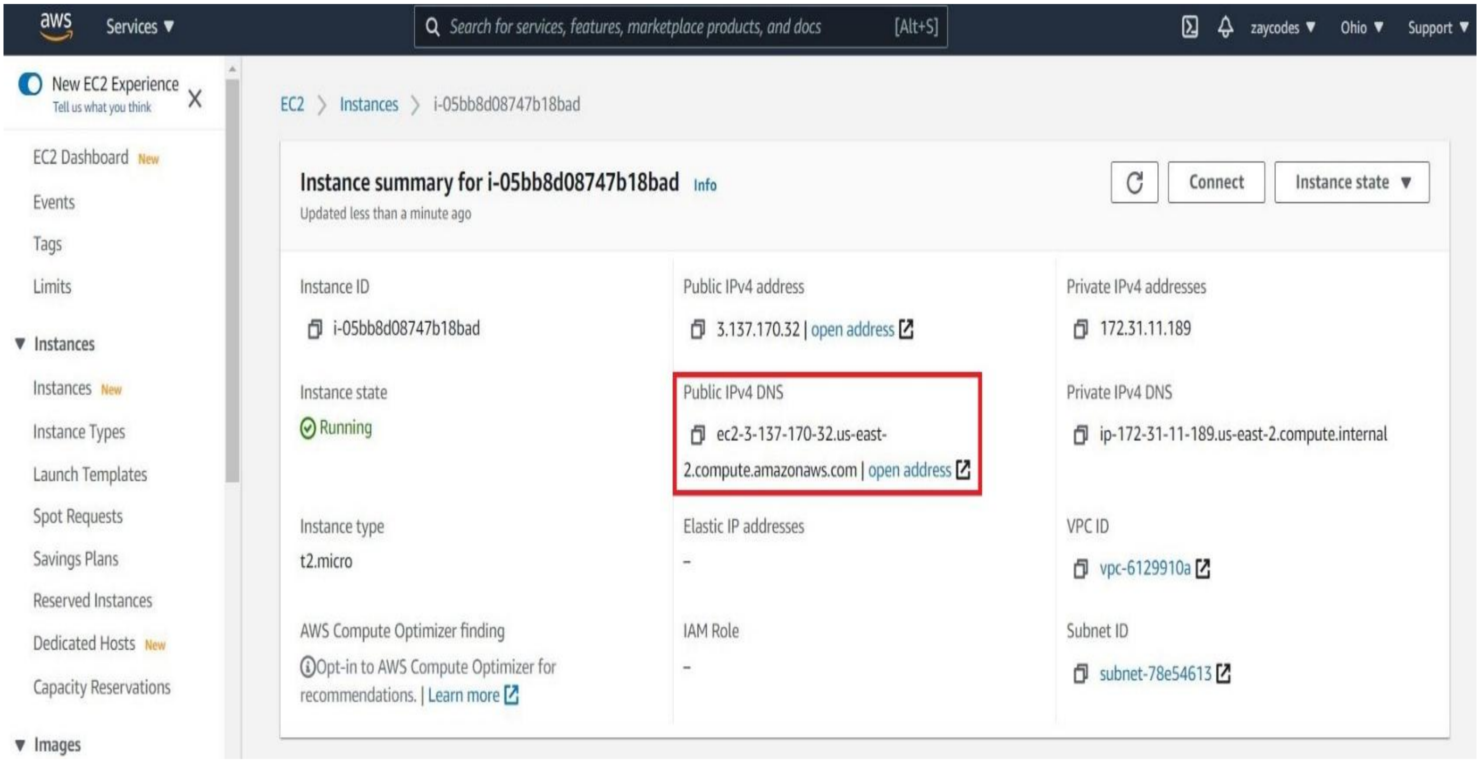
1. Connecting to your Linux instance
2. Downloading and installing Jenkins
3. Configuring Jenkins

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.



EC2 > Instances > i-05bb8d08747b18bad

Instance summary for i-05bb8d08747b18bad Info

Updated less than a minute ago

Instance ID: i-05bb8d08747b18bad

Public IPv4 address: 3.137.170.32 | open address

Private IPv4 addresses: 172.31.11.189

Instance state: Running

Public IPv4 DNS: ec2-3-137-170-32.us-east-2.compute.amazonaws.com | open address

Private IPv4 DNS: ip-172-31-11-189.us-east-2.compute.internal

Instance type: t2.micro

Elastic IP addresses: -

VPC ID: vpc-6129910a

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: -

Subnet ID: subnet-78e54613

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

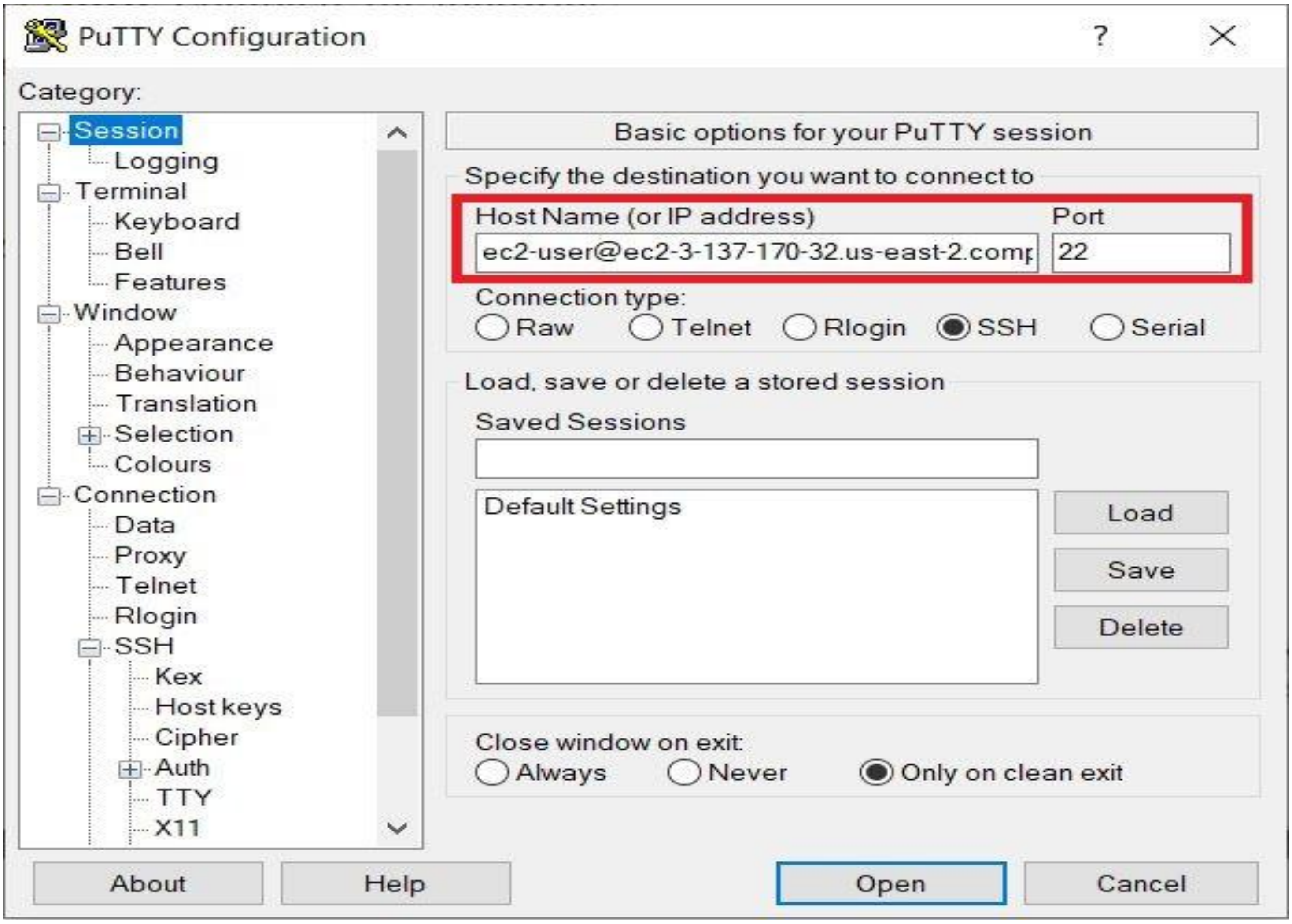
The tool that you use to connect to your Linux instance depends on your operating system.

- 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.



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ec2-user@ec2-3-137-170-32.us-east-2.comp

Port: 22

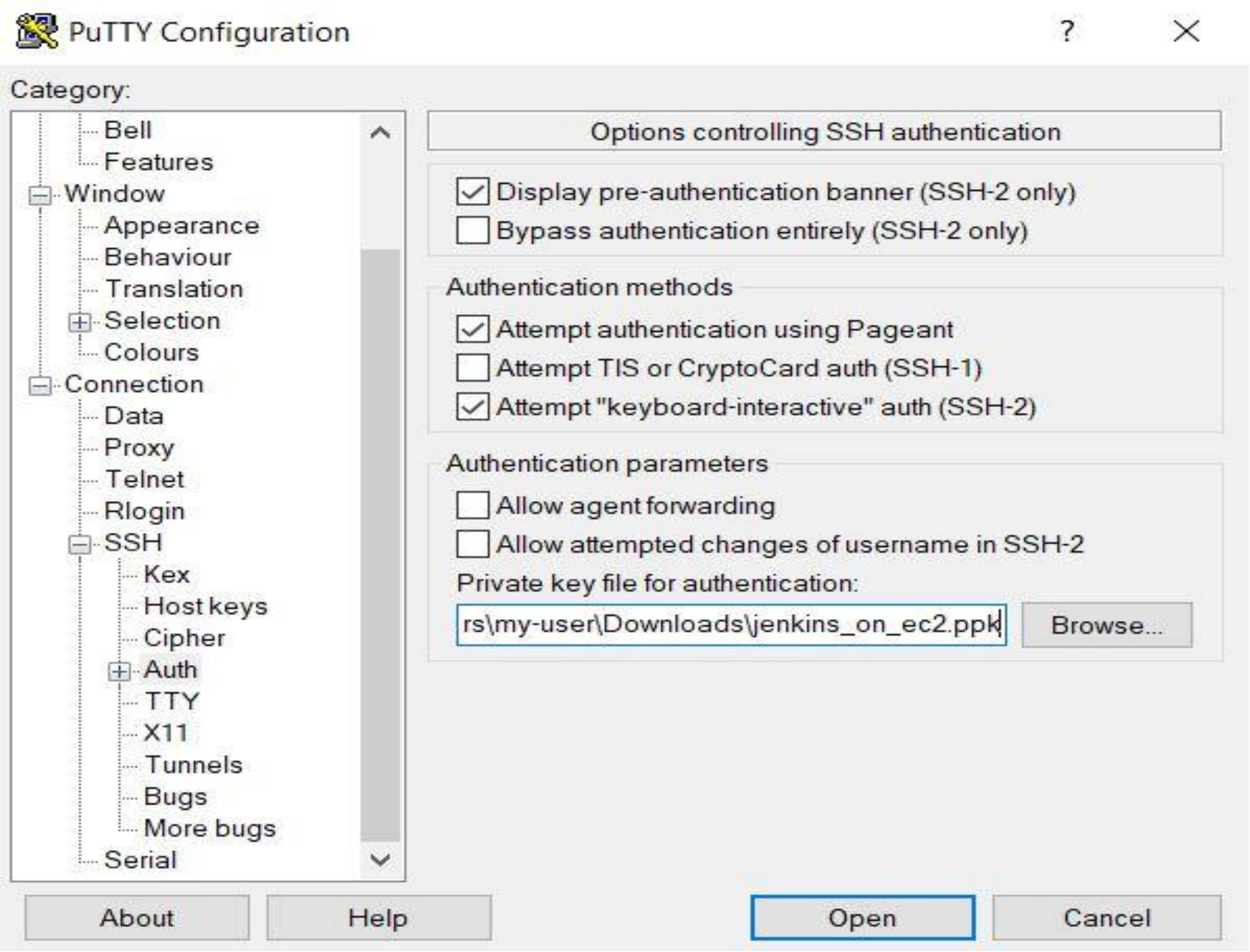
Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

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



Options controlling SSH authentication

☑ Display pre-authentication banner (SSH-2 only)
☐ Bypass authentication entirely (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant
☐ Attempt TIS or CryptoCard auth (SSH-1)
☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

☐ Allow agent forwarding
☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

rs\my-user\Downloads\jenkins_on_ec2.ppk Browse...

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.
2. $ 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)?
```

3. 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:
3. [ec2-user ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \

```
 https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

4. 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
```

5. Install Java (Amazon Linux 2023):

```
[ec2-user ~]$ sudo dnf install java-17-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
```

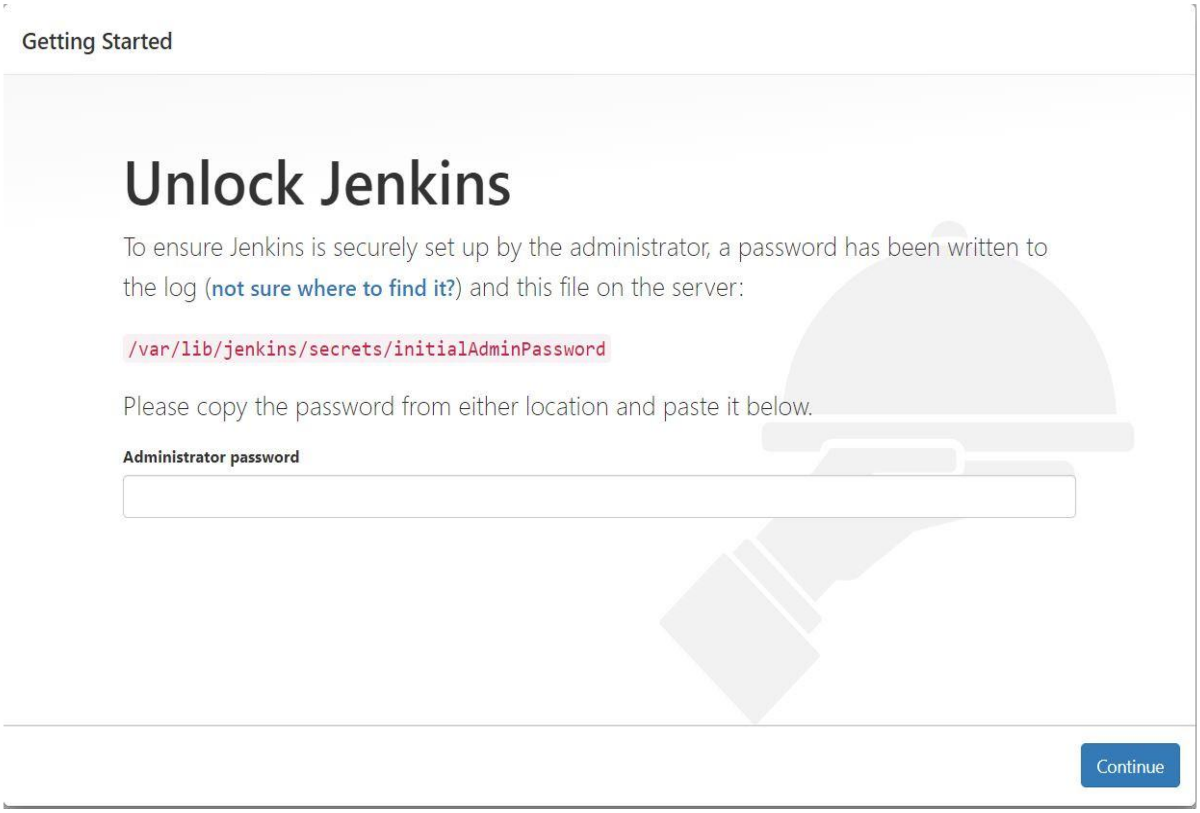
You can check the status of the Jenkins service using the command:

```
[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
```

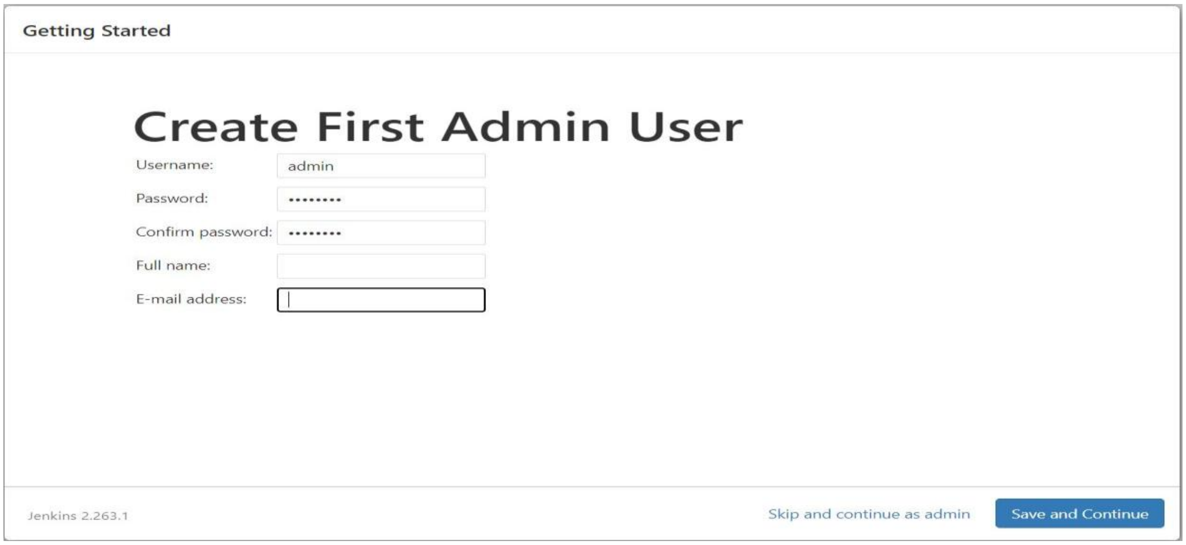
Please copy the password from either location and paste it below.

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: admin

Password: ••••••••

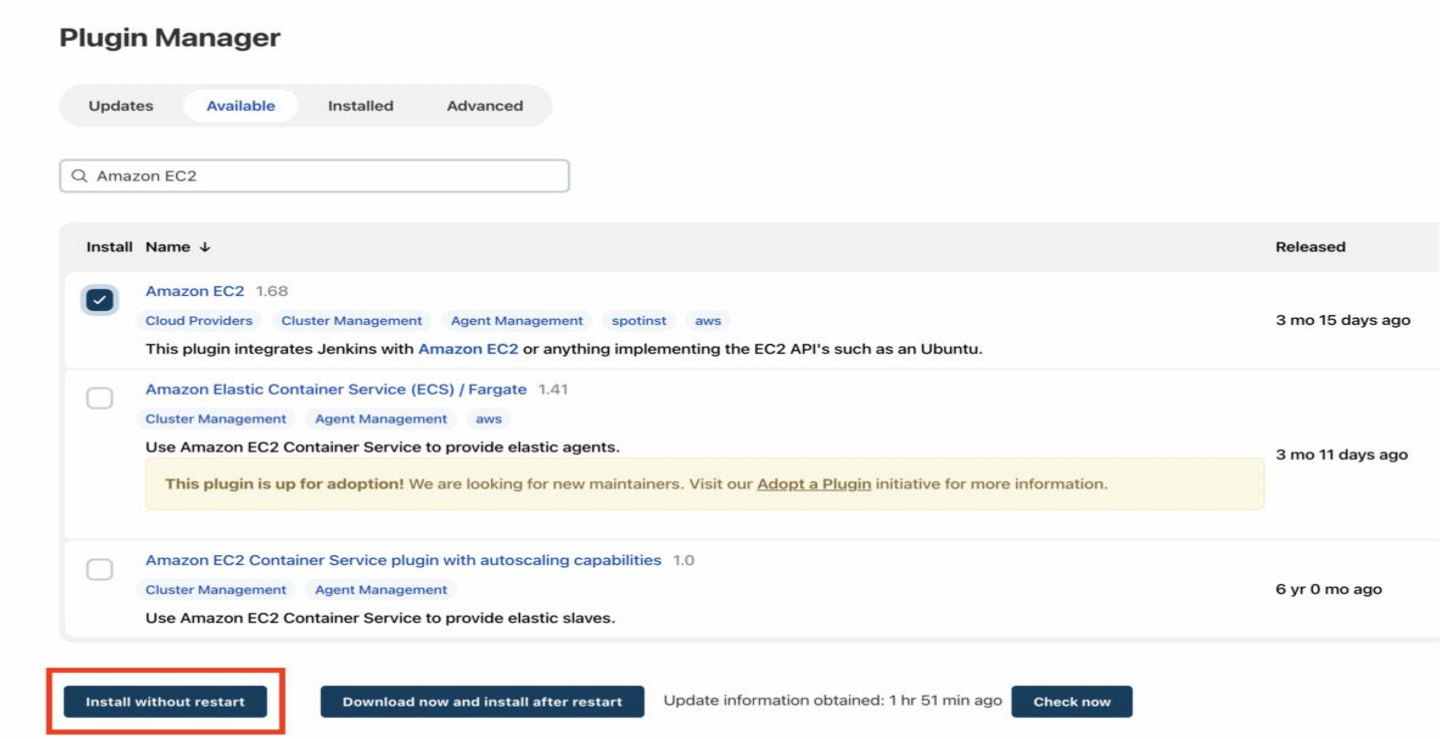
Confirm password: ••••••••

Full name:

E-mail address:

Jenkins 2.263.1

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

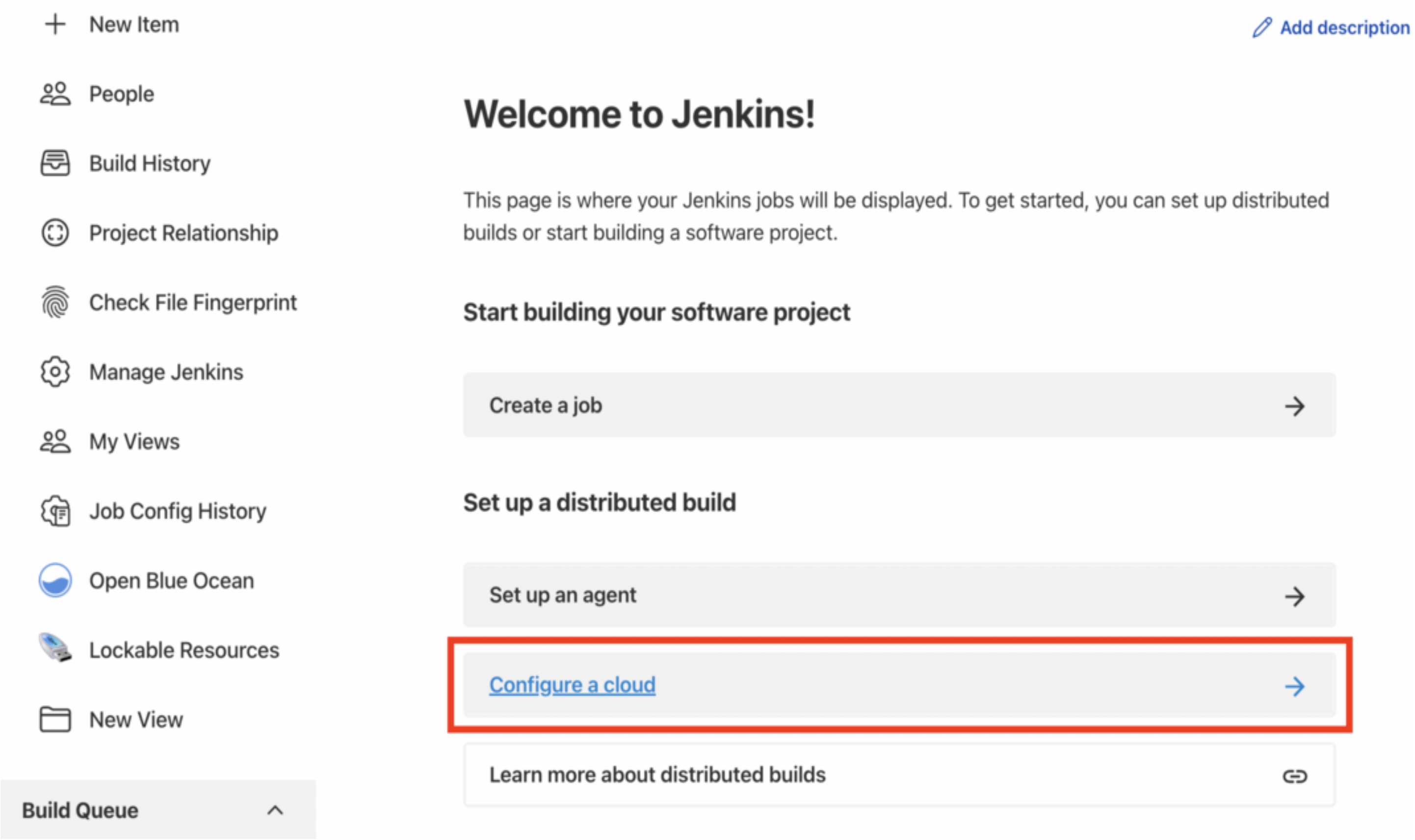
Amazon EC2

Install	Name ↓	Released
☑	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.	3 mo 15 days ago
□	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.	3 mo 11 days ago
□	Amazon EC2 Container Service plugin with autoscaling capabilities 1.0 Cluster Management · Agent Management Use Amazon EC2 Container Service to provide elastic slaves.	6 yr 0 mo ago

Update information obtained: 1 hr 51 min ago

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

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



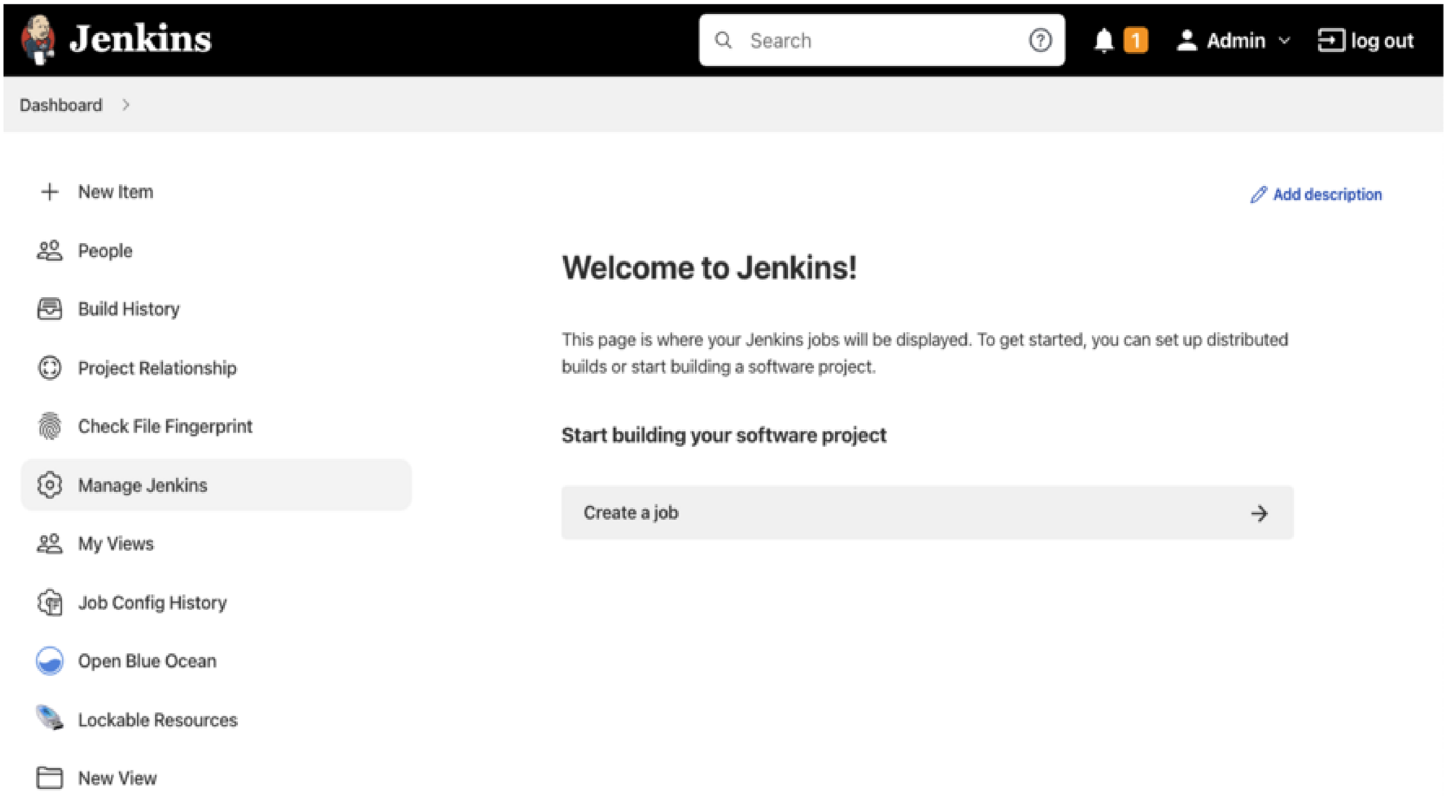
Welcome to Jenkins!

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

Set up a distributed build

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

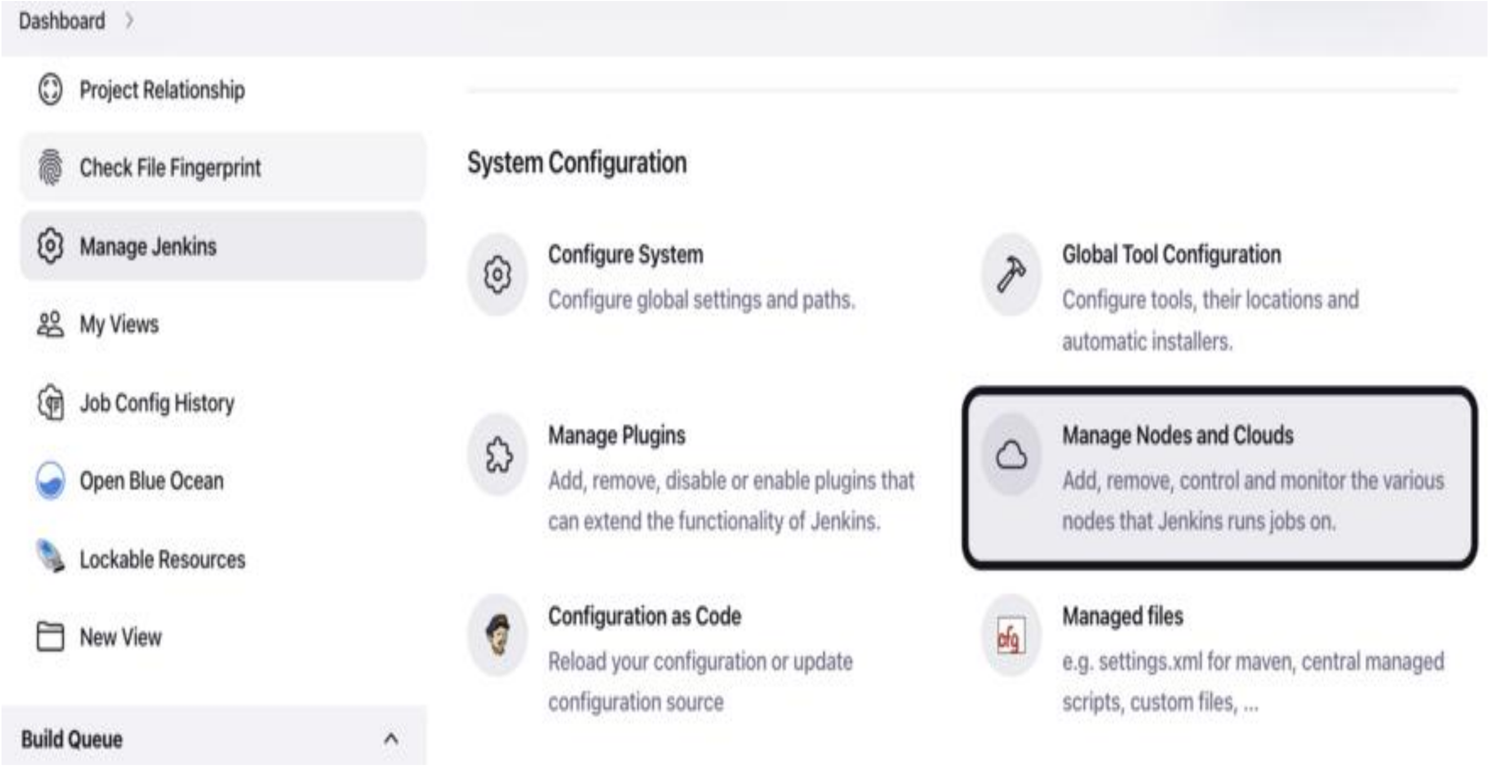


Welcome to Jenkins!

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

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



Dashboard >

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Job Config History

Open Blue Ocean

Lockable Resources

New View

Build Queue

System Configuration

Configure System
Configure global settings and paths.

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

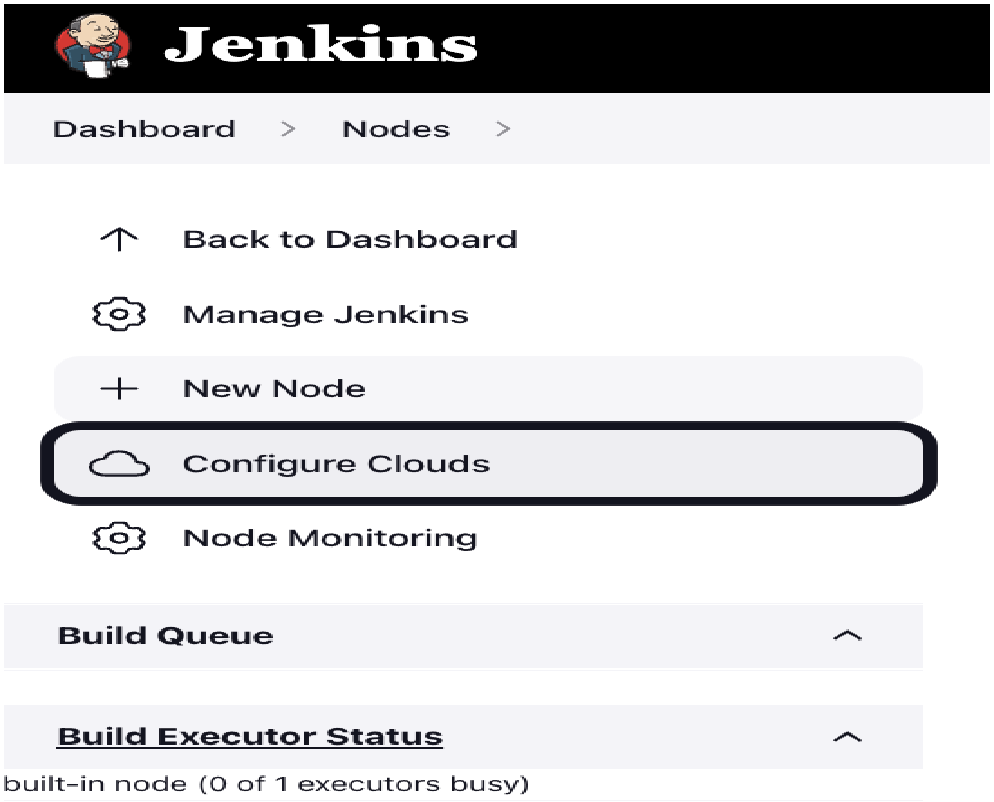
Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

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

Configuration as Code
Reload your configuration or update configuration source

Managed files
e.g. settings.xml for maven, central managed scripts, custom files, ...

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



Jenkins

Dashboard > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

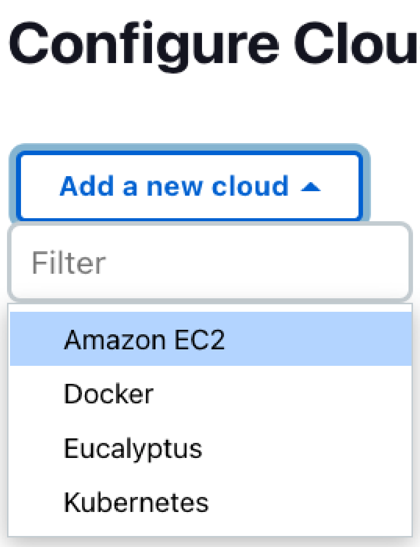
Node Monitoring

Build Queue

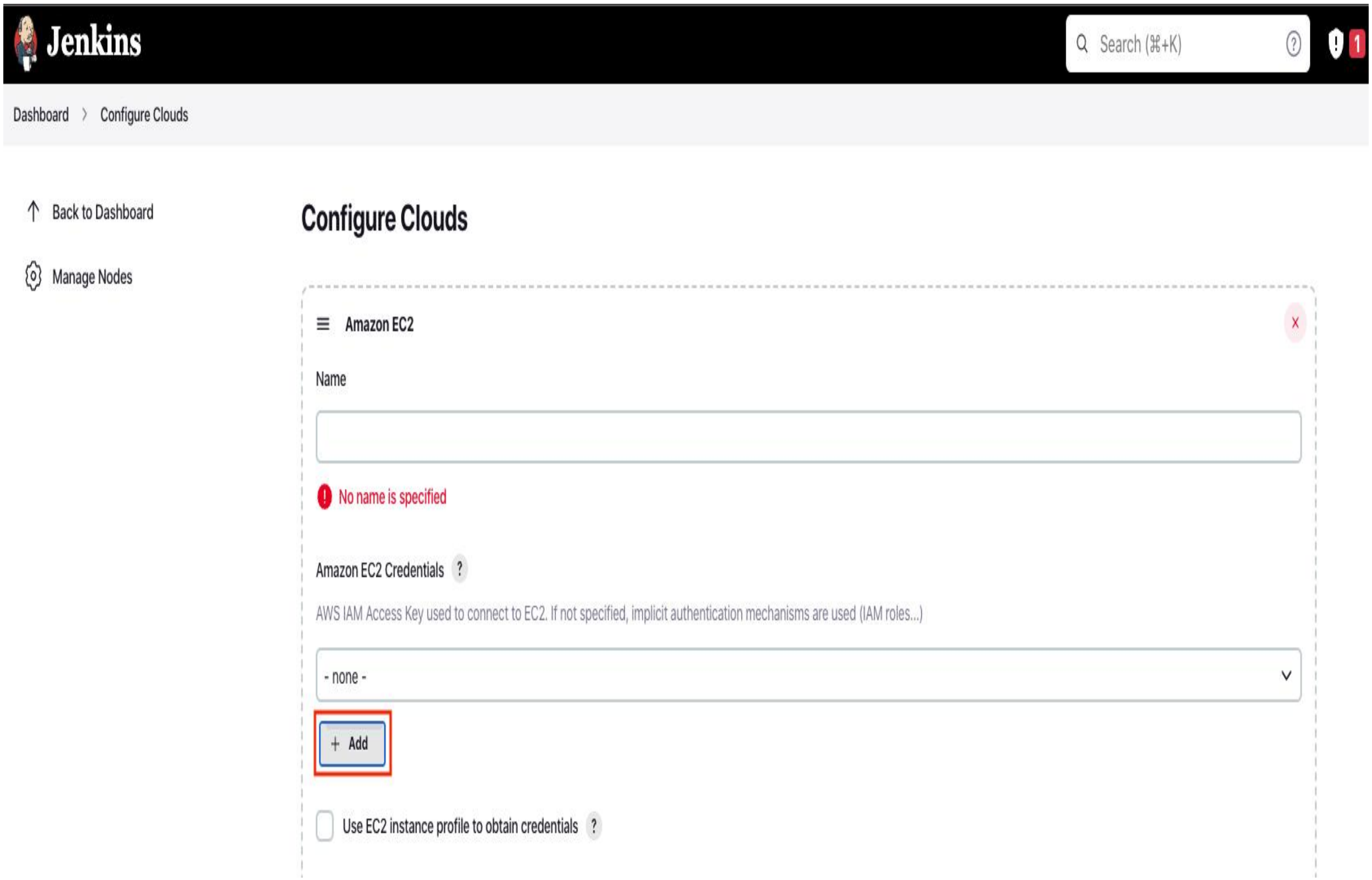
Build Executor Status

built-in node (0 of 1 executors busy)

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



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



Name

No name is specified

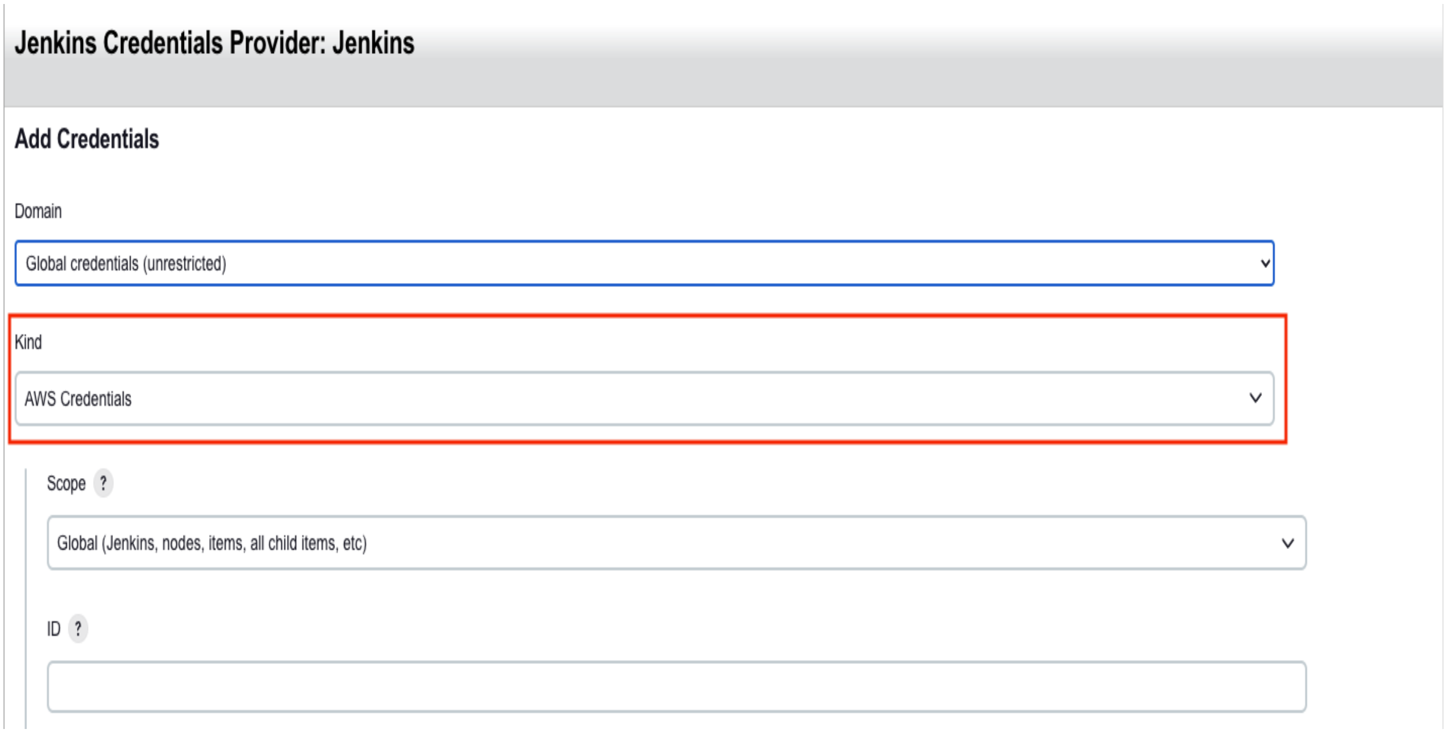
Amazon EC2 Credentials ?

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

- none -

□ Use EC2 instance profile to obtain credentials ?

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



Jenkins Credentials Provider: Jenkins

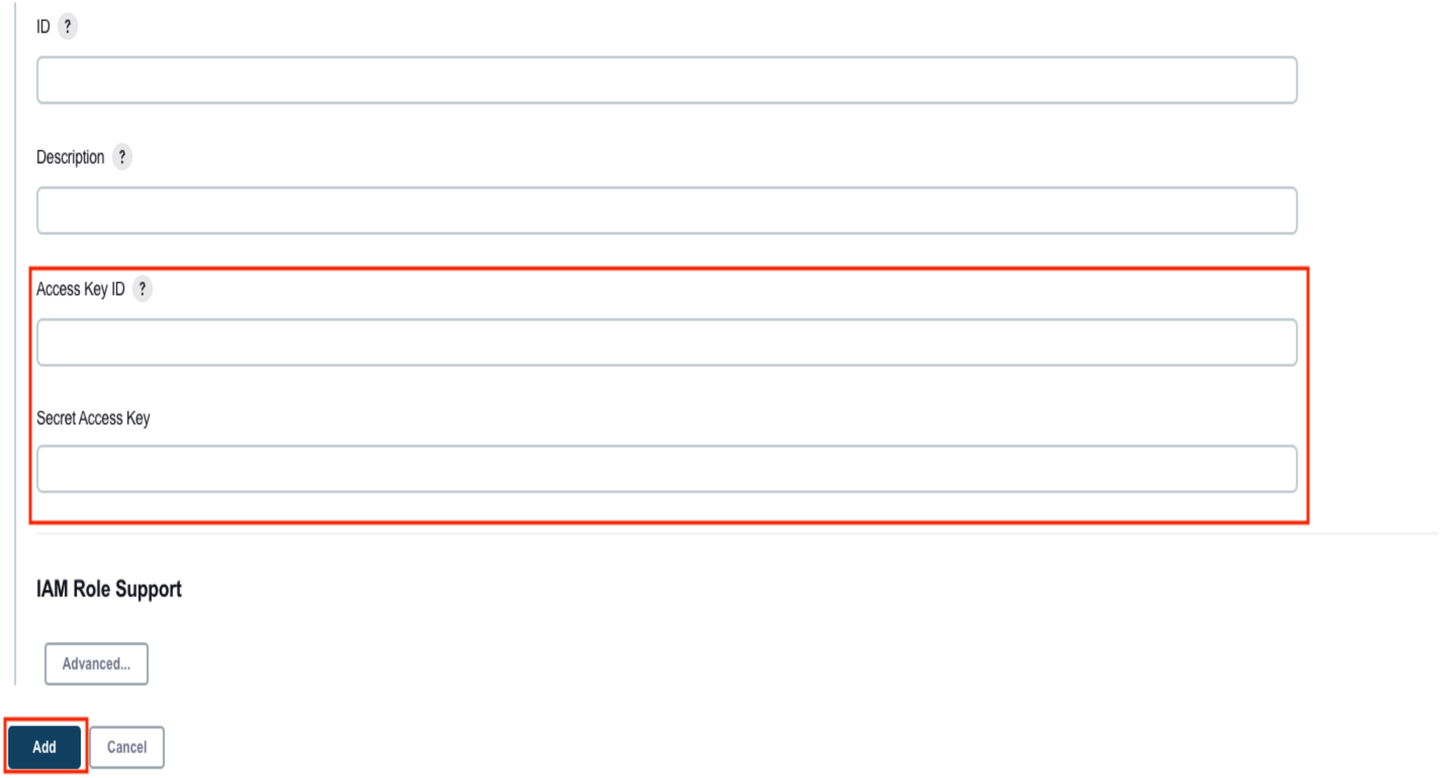
Add Credentials

Domain: Global credentials (unrestricted)

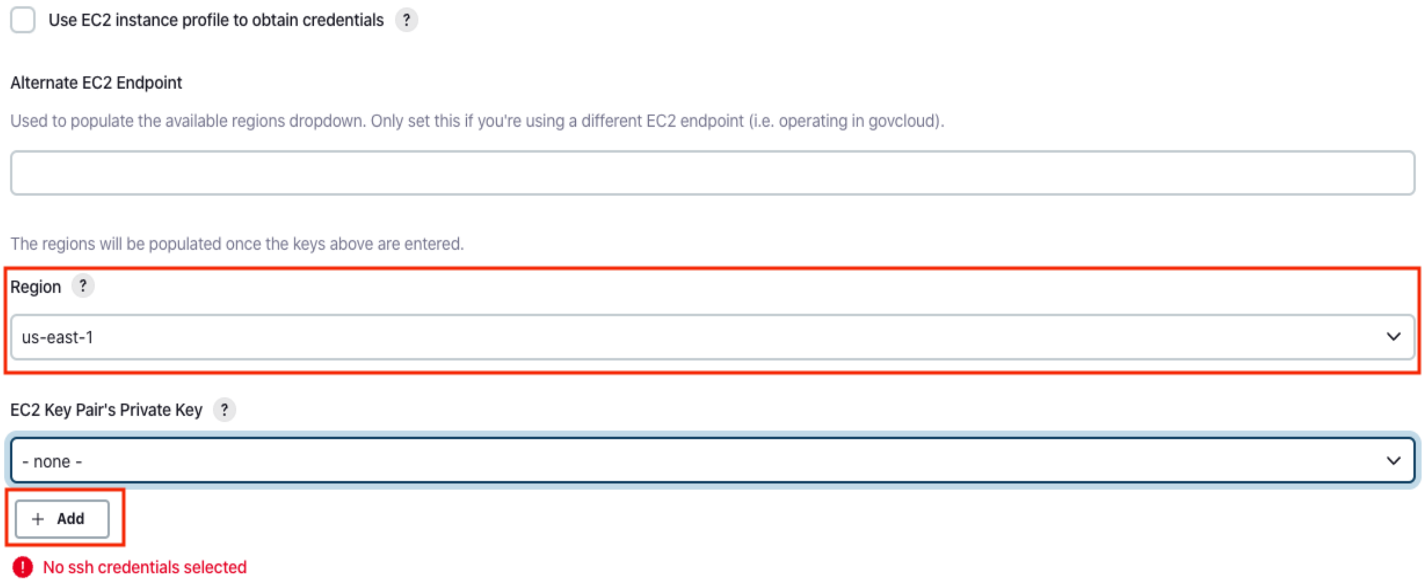
Kind: AWS Credentials

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

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



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



□ 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 ?

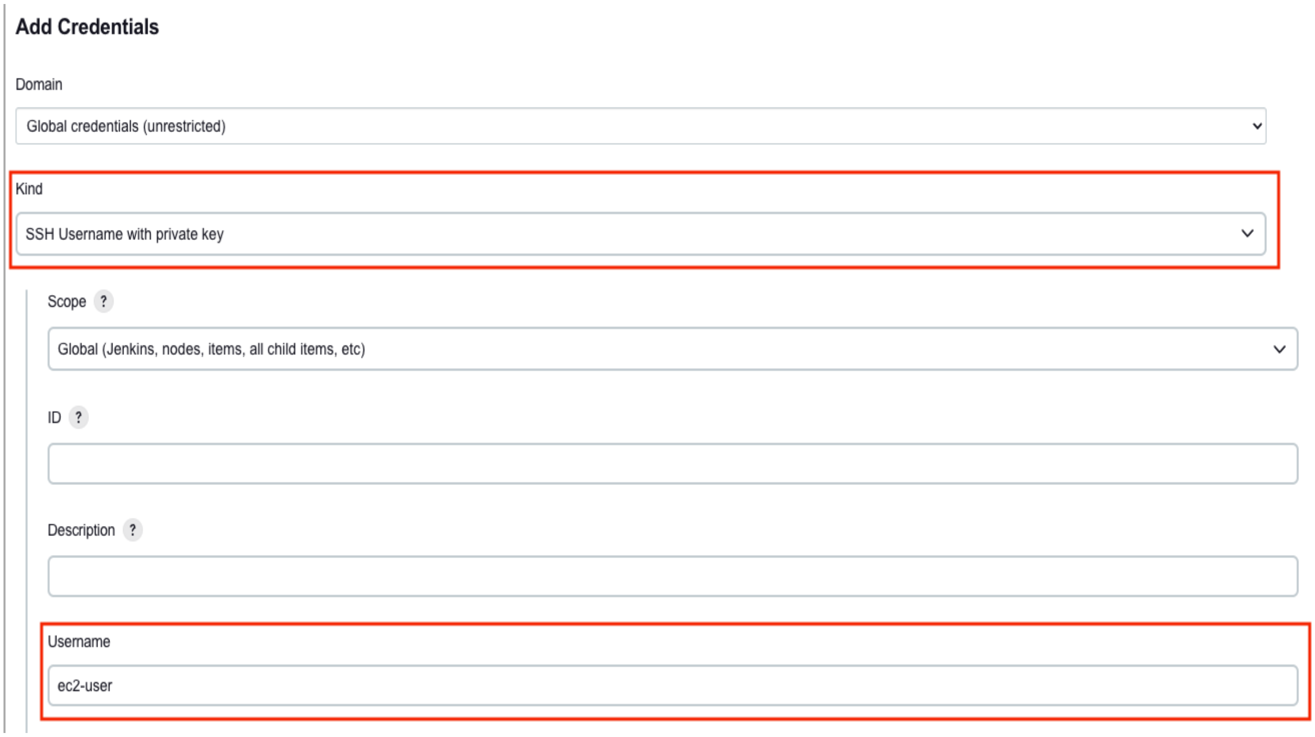
us-east-1

EC2 Key Pair's Private Key ?

- none -

No ssh credentials selected

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)

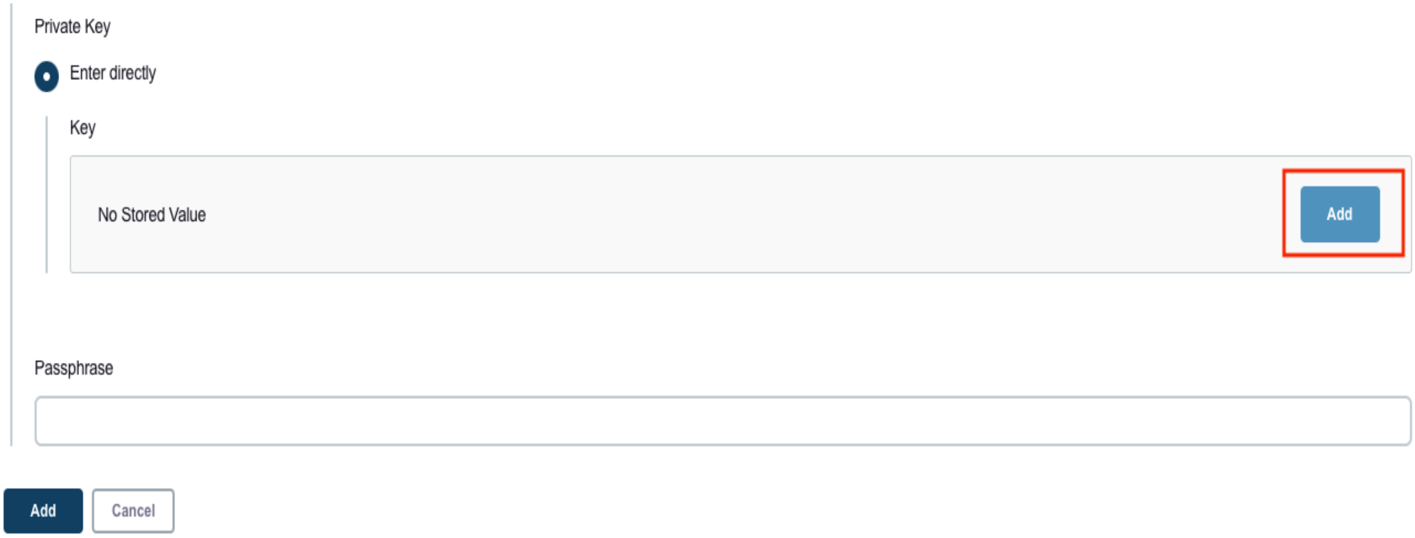
ID ?

Description ?

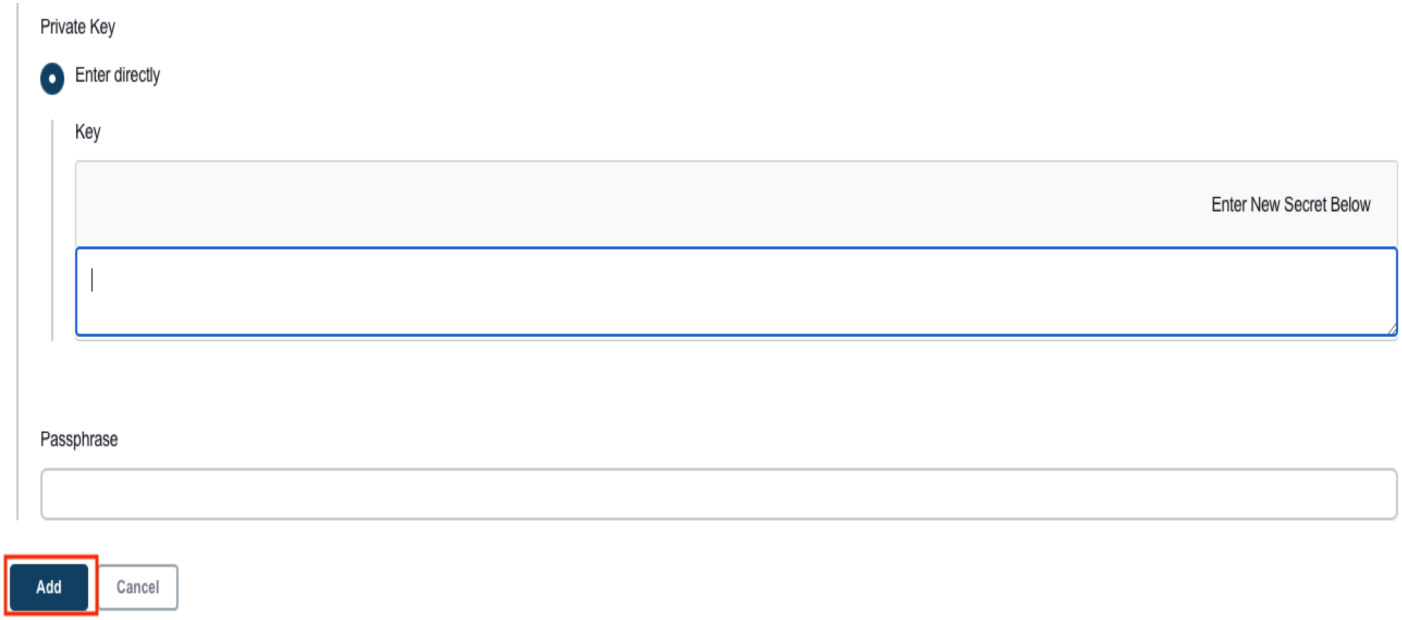
Username

ec2-user

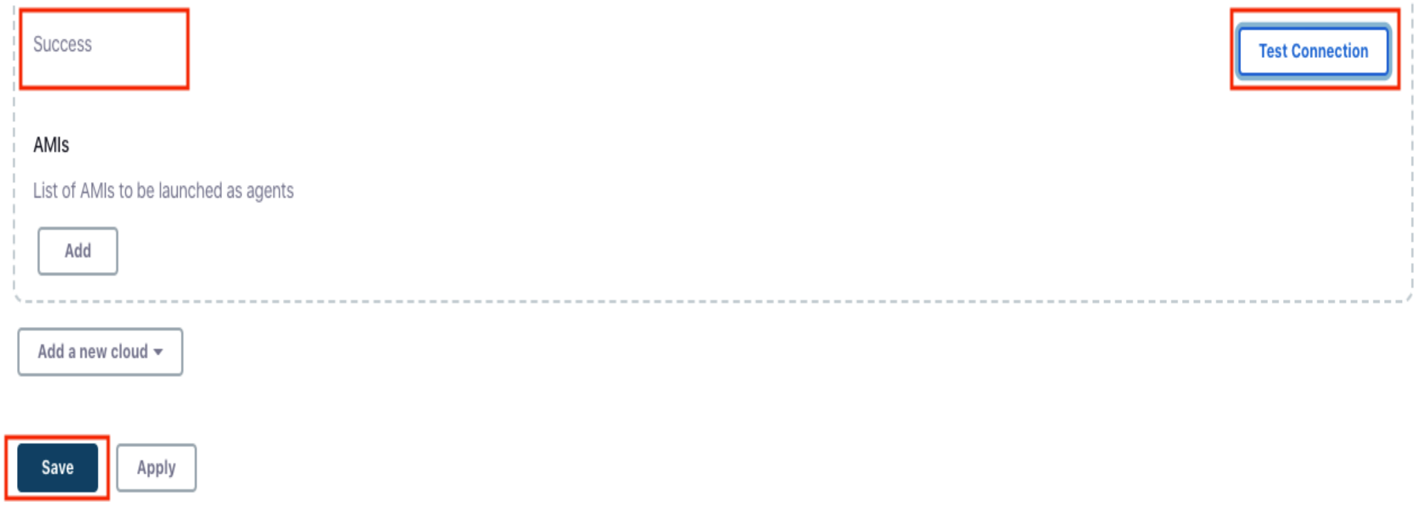
e. Scroll down and select **Enter Directly** under Private Key, then select **Add**.



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.



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



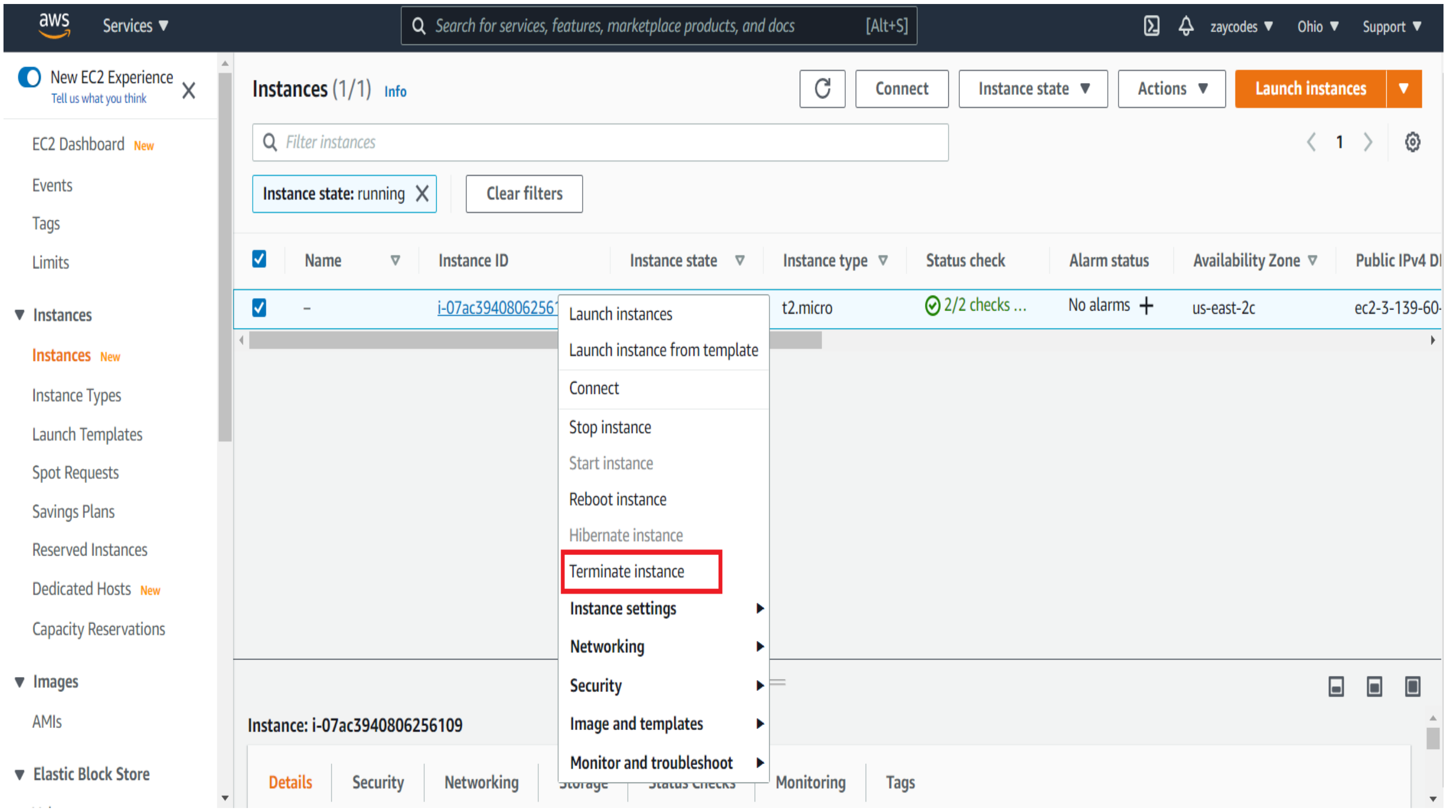
You are now ready to use EC2 instances as Jenkins agents.

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**.


aws
Services
Search for services, features, marketplace products, and docs [Alt+S]
zaycodes
Ohio
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Capacity Reservations
Images
AMIs
Elastic Block Store
Instances (1/1) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Instance state: running
Clear filters
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DI
–
i-07ac3940806256
t2.micro
2/2 checks ...
No alarms
us-east-2c
ec2-3-139-60-
Launch instances
Launch instance from template
Connect
Stop instance
Start instance
Reboot instance
Hibernate instance
Terminate instance
Instance settings
Networking
Security
Image and templates
Monitor and troubleshoot
Instance: i-07ac3940806256109
Details
Security
Networking
Storage
Status checks
Monitoring
Tags