

# Create an EC2 instance with Amazon Linux

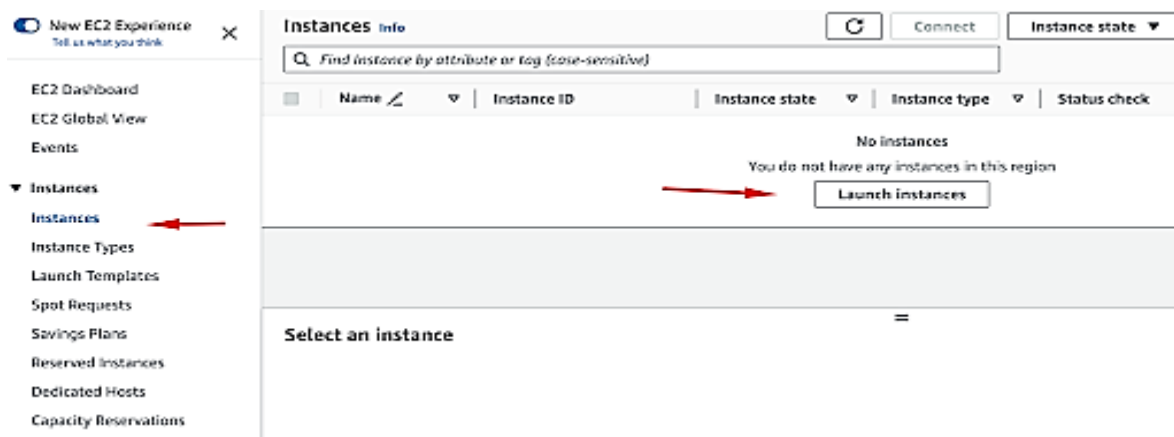
## Step 1: Sign in to your AWS account

Before you can set up an EC2 instance with Amazon Linux, you need an AWS account.

## Step 2: Launch an EC2 instance

Once you have logged into your AWS account, in the search bar you can type EC2 in order to access the main page for provisioning and management.

Next, under Instances, you can select the option “Launch Instances.”



We will land on the new page where we can select additional options, such as the Name of our instance, Application, and OS Images for our instance and we will select the default option — Amazon Linux.

Name

e.g. My Web Server
Add additional tags

▼ 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

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

Quick Start

Amazon Linux  
aws

macOS  
Mac

Ubuntu  
ubuntu

Windows  
Microsoft

Red Hat  
Red Hat

SUSE Li  
SUS

Browse more AMIs  
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI  
ami-01342111f883d5e4e (64-bit (x86)) / ami-0314747aba1360bc7 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible

When we scroll below we can choose the type of our instance. Amazon Linux has a minimum requirement of 512MB of RAM and 1 CPU core, so if we are testing things, we can select the t2.micro instance which fits into the free tier in AWS.

▼ Instance type Info

Instance type

t2.micro  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand SUSE base pricing: 0.018 USD per Hour  
On-Demand SUSE base pricing: 0.0134 USD per Hour  
On-Demand Linux base pricing: 0.0134 USD per Hour  
On-Demand RHEL base pricing: 0.0734 USD per Hour
Free tier eligible

All generations  
Compare instance types

Additional costs apply for AMIs with pre-installed software

When you have a project and instance that should be production-ready, you can select instances with more RAM and CPU power, by choosing from the dropdown list.

Next, we will make sure to create a new security group for our 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

Creating a new security group for your new EC2 instance is a fundamental security best practice in AWS. It allows you to define and enforce customized network access controls, adhere to the [least privilege principle](#), and maintain better isolation and security for your EC2 instances.

Create key pair

Key pair name

Key pairs allow you to connect to your instance securely.

jumpcloud\_amazon\_linux

The name can include upto 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type



☒ RSA  
RSA encrypted private and public key pair

☐ ED25519  
ED25519 encrypted private and public key pair

Private key file format

☒ .pem  
For use with OpenSSH

☐ .ppk  
For use with PuTTY

 When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#) 

Cancel

Create key pair

We can select Private key file format which can be either in .pem format or .ppk in case you use a Windows SSH client such as PuTTY.

In our case, we will select the .pem format since we will be using the terminal.

Finally, click on the “Create key pair” option in order to generate the SSH key and this will prompt the download in your browser. Also, there is a warning from AWS that we should store our private keys in a secure and accessible location on our computers.

Further in our setup process, we would need to select our Network Settings.



Our final step is confirming the selected options in the summary and we can proceed to launch the instance. Encrypt EBS snapshots and volumes.

▼ Summary

Number of instances [Info](#)

Amazon Linux 2023 AMI 2023.2.2...[read more](#)

ami-01342111f883d5e4e

Virtual server type (instance type)

t2.micro

Firewall (security group)


New security group

Storage (volumes)

1 volume(s) - 8 GiB

③ 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

[Review commands](#)

You will get the notification that the instance is successfully launched and you'll return to your Instances list.

EC2 > [Instances](#) > Launch an instance

✓ Success

Successfully initiated launch of instance [\(i-0a74bc2e7a94b4a7c\)](#)

After a short wait, typically with a fast provisioning time, you'll be able to locate your instance in the list.

<input checked="" type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check	Alarm status	Actions
<input checked="" type="checkbox"/>	jumpcloud-amazon-linux	i-0a74bc2e7a94b4a7c	Running	t2.micro	2/2 checks passed	No alarms	+

Instance: i-0a74bc2e7a94b4a7c (jumpcloud-amazon-linux)

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

▼ Instance summary info

Instance ID i-0a74bc2e7a94b4a7c (jumpcloud-amazon-linux)	Public IPv4 address 3.79.150.186 <a href="#">open address</a>	Private IPv4 addresses 172.31.27.20
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-3-79-150-186.eu-central-1.compute.amazonaws.com <a href="#">open address</a>

## Step 3: Connect to your Amazon Linux instance

Now the next step is to connect to the instance via SSH. We can do so by using our Instances menu and clicking Connect:

Instances (1/1) Info

Find instance by attribute or tag (case-sensitive)

Connect

Instance state ▼

Actions ▲

Connect

View details

Manage instance state

<input checked="" type="checkbox"/>	Name ↗	Instance ID	Instance state	Instance type	Status check
<input checked="" type="checkbox"/>	jumpcloud-amazon-linux	i-0a74bc2e7a94b4a7c	Running	t2.micro	2/2 checks passed

Here we will see a new menu where we will select the SSH client menu item:

Connect to instance Info

Connect to your instance i-0a74bc2e7a94b4a7c (jumpcloud-amazon-linux) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID  
i-0a74bc2e7a94b4a7c (jumpcloud-amazon-linux)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is jumpcloud\_amazon\_linux.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
chmod 400 jumpcloud\_amazon\_linux.pem
4. Connect to your instance using its Public DNS:  
ec2-3-79-150-186.eu-central-1.compute.amazonaws.com

Example:

```
ssh -i "jumpcloud_amazon_linux.pem" ec2-user@ec2-3-79-150-186.eu-central-1.compute.amazonaws.com
```

**Note:** In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

We will now follow the instructions provided by AWS in order to connect to our instance.

After opening your terminal and finding where you have downloaded the .pem key, it is important to change permissions to it, for security reasons, so that the key is not publicly viewable, and in our case, we will run the following command:

```
chmod 400 jumpcloud_amazon_linux.pem
```

Next, we will use the SSH key to connect to our instance:

```
ssh -i "jumpcloud_amazon_linux.pem" ec2-user@ec2-3-79-150-186.eu-central-1.compute.amazonaws.com
```

Here, the `-i` flag in SSH is used to specify the path to the private key file to be used for authentication when connecting to a remote server. It allows you to choose a specific key file when you have multiple key pairs or non-standard key file names and locations.

When we log into our Amazon Linux instance for the first time, we need to confirm the authenticity of the host. Here you can type yes and press Enter.

```
The authenticity of host 'ec2-3-79-150-186.eu-central-1.compute.amazonaws.com (3
.79.150.186)' can't be established.
ED25519 key fingerprint is SHA256:7+14e5rWSJ3DfH2UOJ5mDdG7LOK23hVked2L/I1Ta48.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

After this action, we will be logged into our instance:

```
2-3-79-150-186.eu-central-1.compute.amazonaws.com
#_
~\####_ Amazon Linux 2023
~~~\#####\
~~~~\###|
~~~~\#/ https://aws.amazon.com/linux/amazon-linux-2023
~~~~V~'-'->
~~~~
~~~~._./
~~~~/_m/'
```

[ec2-user@ip-172-31-27-20 ~]\$

You can always verify the version of your Amazon Linux by typing:

**cat /etc/os-release**

```
[ec2-user@ip-172-31-27-20 ~]$ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
SUPPORT_END="2028-03-01"
```

We can list updates to our system by running the following command:

**sudo dnf update**

This command requires higher system privileges so make sure you use the sudo.

Often the repositories are up to date once the instance has been provisioned, but it is always a good idea to check, mainly for security and compatibility reasons.

```
[ec2-user@ip-172-31-27-20 ~]$ sudo dnf update
Last metadata expiration check: 0:57:44 ago on Sat Sep 30 14:09:52 2023.
Dependencies resolved.
Nothing to do.
Complete!
```