

You've been invited to try an early, beta iteration of the new launch instance wizard. We will continue to improve the experience over the next few months. We're asking customers for their feedback on this early release. To exit the new launch instance wizard at any time, choose the **Cancel** button.

Try it now!

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

# Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Red Hat

Free tier eligible

Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

SUSE Linux

Free tier eligible

**SUSE Linux Enterprise Server 15 SP3 (HVM), SSD Volume Type** - ami-08895422b5f3aa64a (64-bit x86) / ami-08f182b25f271ef79 (64-bit Arm)

SUSE Linux Enterprise Server 15 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.5, PHP 5.3, and Ruby 1.8.7 available.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Ubuntu

Free tier eligible

**Ubuntu Server 20.04 LTS (HVM), SSD Volume Type** - ami-04505e74c0741db8d (64-bit x86) / ami-0b49a4a6e8e22fa16 (64-bit Arm)

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Amazon RDS

**Are you launching a database instance? Try Amazon RDS.**

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy **Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server** databases on AWS. [Aurora](#) is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. [Learn more about RDS](#)

Launch a database using RDS

☒ 64-bit (x86)

☐ 64-bit (Arm)

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)

Select

☒ 64-bit (x86)

☐ 64-bit (Arm)

Select

1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

## Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances ⓘ	Volumes ⓘ	Network Interfaces ⓘ	
<input type="text" value="Name"/>	<input type="text" value="Demo"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Add another tag

(Up to 50 tags maximum)

Cancel

Previous

Review and Launch

Next: Configure Security Group

## Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a **new** security group  
☐ Select an **existing** security group

Security group name:

Description:

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>	Description <small>i</small>	
All traffic ▼	All	0 - 65535	Anywhere ▼ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕

Add Rule



### Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel

Previous

Review and Launch

aws

Services

Q

Search for services, features, blogs, docs, and more

[Alt+S]

N. Virginia

Poluru Anitha

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 7: Review Instance Launch

Free tier eligible

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-04505e74c0741db8d

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>)

Root Device Type: ebs Virtualization type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory
t2.micro	-	1	1 GiB

Security Groups

Security group name	Description
launch-wizard-1	launch-wizard-1 created 2022-08-01 10:00

Type	Protocol
All traffic	All
All traffic	All

Instance Details

Edit instance type

Edit security groups

Edit instance details

Cancel

Previous

Launch

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance. Amazon EC2 supports ED25519 and RSA key pair types.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair type

☒ RSA

☐ ED25519

Key pair name

virginia

Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

Launch Instances

Feedback

English (US)

© 2022, Amazon Internet Services Private Ltd. or its affiliates.

Privacy

Terms

Cookie preferences


 New EC2 Experience

Tell us what you think


X





- EC2 Dashboard
- EC2 Global View
- Events
- Tags
- Limits
- ▼ Instances
  - Instances New
  - Instance Types
  - Launch Templates
  - Spot Requests
  - Savings Plans
  - Reserved Instances New
  - Dedicated Hosts
  - Scheduled Instances
  - Capacity Reservations
- ▼ Images
  - AMIs New
  - AMI Catalog

Instances (1) [Info](#)

 [Connect](#) [Instance state ▼](#) [Actions ▼](#) [Launch instances](#) ▼

< 1 >



<input type="checkbox"/>	Name ▼	Instance ID ▲	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS
<input type="checkbox"/>	Demo	<a href="#">i-01a8d6a768bfff9ff</a>	 Running 	t2.micro	 Initializing	No alarms 	us-east-1a	ec2-54-84-13-48

Select an instance

X







instance type

2.micro

en address

(IPv4 only)

.0.ec2.internal

Status check

Alarm status

Availability Zone

Public IP

ec2-54-8

Putty Configuration

?

×

Category:

Logging

Terminal

Keyboard

Bell

Features

Window

Appearance

Behaviour

Translation

Selection

Colours

Connection

Data

Proxy

SSH

Kex

Host keys

Cipher

Auth

TTY

X11

Tunnels

Bugs

More bugs

Options controlling SSH authentication

☒ Display pre-authentication banner (SSH-2 only)

☐ Bypass authentication entirely (SSH-2 only)

☐ Disconnect if authentication succeeds trivially

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:

C:\Users\Aravind\Downloads\virginia.ppk

Browse...

About

Help

Open

Cancel

Answer private resource DNS name

IPv4 (A)



ubuntu@ip-172-31-90-240: ~

```
Preparing to unpack .../08-libxpm4_1%3a3.5.12-1_amd64.deb ...
Unpacking libxpm4:amd64 (1:3.5.12-1) ...
Selecting previously unselected package libgd3:amd64.
Preparing to unpack .../09-libgd3_2.2.5-5.2ubuntu2.1_amd64.deb ...
Unpacking libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Selecting previously unselected package nginx-common.
Preparing to unpack .../10-nginx-common_1.18.0-0ubuntu1.2_all.deb ...
Unpacking nginx-common (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-http-image-filter.
Preparing to unpack .../11-libnginx-mod-http-image-filter_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-http-image-filter (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-http-xslt-filter.
Preparing to unpack .../12-libnginx-mod-http-xslt-filter_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-mail.
Preparing to unpack .../13-libnginx-mod-mail_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-mail (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package libnginx-mod-stream.
Preparing to unpack .../14-libnginx-mod-stream_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking libnginx-mod-stream (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package nginx-core.
Preparing to unpack .../15-nginx-core_1.18.0-0ubuntu1.2_amd64.deb ...
Unpacking nginx-core (1.18.0-0ubuntu1.2) ...
Selecting previously unselected package nginx.
Preparing to unpack .../16-nginx_1.18.0-0ubuntu1.2_all.deb ...
Unpacking nginx (1.18.0-0ubuntu1.2) ...
Setting up libxpm4:amd64 (1:3.5.12-1) ...
Setting up nginx-common (1.18.0-0ubuntu1.2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /lib/systemd/system/nginx.service.
Setting up libjpeg8:amd64 (2.1-3.1build1) ...
Setting up libnginx-mod-http-xslt-filter (1.18.0-0ubuntu1.2) ...
Setting up libwebp6:amd64 (0.6.1-2ubuntu0.20.04.1) ...
Setting up fonts-dejavu-core (2.37-1) ...
Setting up libjpeg-turbo8:amd64 (2.0.3-0ubuntu1.20.04.1) ...
Setting up libjpeg8:amd64 (8c-2ubuntu8) ...
Setting up libnginx-mod-mail (1.18.0-0ubuntu1.2) ...
Setting up fontconfig-config (2.13.1-2ubuntu3) ...
Setting up libnginx-mod-stream (1.18.0-0ubuntu1.2) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.2) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Setting up libnginx-mod-http-image-filter (1.18.0-0ubuntu1.2) ...
Setting up nginx-core (1.18.0-0ubuntu1.2) ...
Setting up nginx (1.18.0-0ubuntu1.2) ...
Processing triggers for ufw (0.36-6ubuntu1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@ip-172-31-90-240:~$
```



ubuntu

## Apache2 Ubuntu Default Page

### It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the

```
<html>
<head>
Ubuntu Server
</head>
<body bgcolor="violet">
<p><h2>Hello INTELLIPAAT</h2></p>
</body>
</html>□
```

Ubuntu Server

Hello INTELLIPAAT