

**Project in AWS
Practice Lab**

Set EC2 Instance Bootstrapping

Andra-Diana Popescu

2025

ABOUT THIS LAB

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. In this lab, we configure a web server to corporate standards, noting how much effort it can take to walk through these steps manually each time. We then learn how to bootstrap an instance using a user data script, allowing us to automatically build servers. By the end of this lab, the user will understand what an EC2 bootstrap script is and how to use one to automate one of the most tedious parts of server building.

LEARNING OBJECTIVES

- Manually Install Software on webserver-01
- Use a Bootstrap Script to Build webserver-02 and Debug Issues
- Use a Fixed Bootstrap Script to Build webserver-03

AWS Documentation about EC2:

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html>

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/user-data.html>

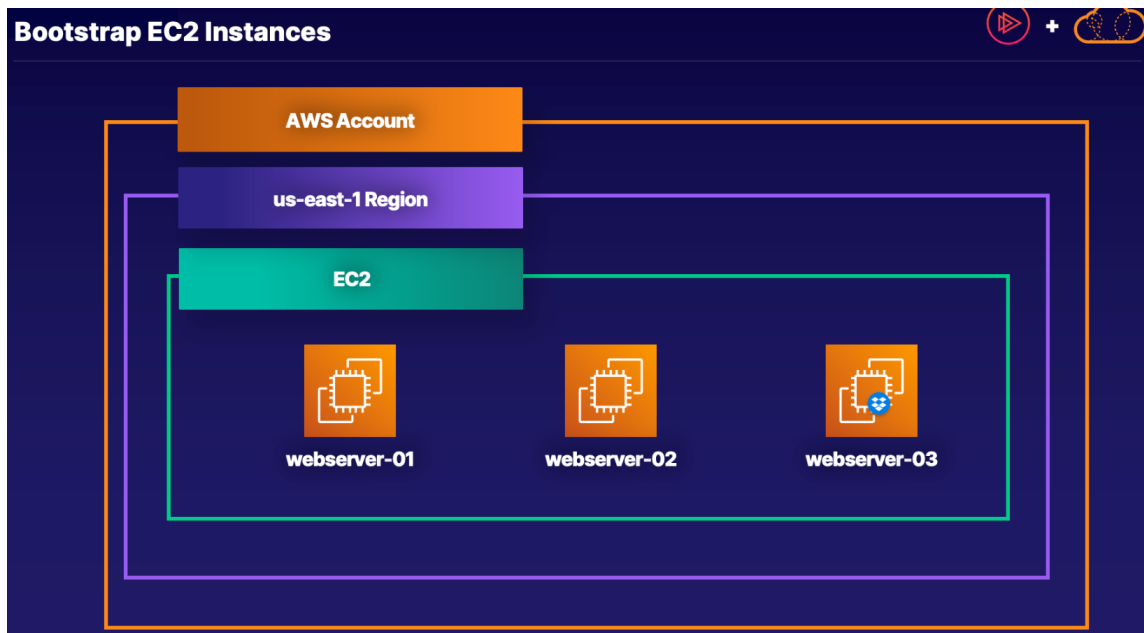
<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html>

Source: <https://learn.acloud.guru/course/certified-solutions-architect-associate/>

Table of Contents

Lab Diagram	4
Log in to your AWS account	5
1. Manually Install Software on webserver-01	5
1.1. Set Up apache2	5
1.2. Set Up the AWL CLI Tool & MySQL Client.....	8
2. Use a Bootstrap Script to Build webserver-02 and Debug Issues.....	11
2.1. Set Up the Script	11
2.2. Connect to and View webserver-02	13
3. Use a Fixed Bootstrap Script to Build webserver-03	15
3.1. Set Up the Script	15
3.2. Connect to and View webserver-03	18

Lab Diagram



Let's look at the configured infrastructure. We have our AWS account in the **us-east-1** North Virginia region, and we have an EC2 instance. Our scenario is that we've been tasked to set a group of 3 EC2 instances with Apache Web Server for testing, the Amazon Command Line tool, and a MySQL database.

We'll begin the lab by manually installing everything required to bring our first web server up to company standards – fully up to date (**webserver-01**). Then, we'll configure bootstrapping to quickly configure 2 more instances (**webserver-02**, **webserver-03**).

Log in to your AWS account



Sign in as IAM user

Account ID (12 digits) or account alias

Type Account ID

IAM user name

Type IAM user name

Password

☐ Remember this account

Sign in

Sign in using root user email

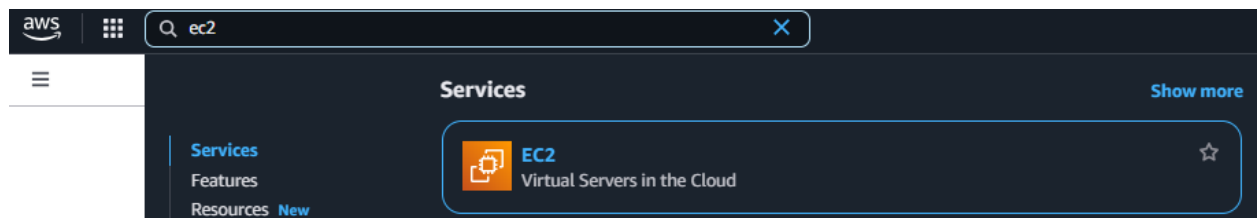
[Forgot password?](#)



1. Manually Install Software on **webserver-01**

1.1. Set Up **apache2**

1. Once you are logged in to the AWS Management Console, navigate to EC2.



2. In Resources at the top, click **Instances (running)**.

Resources

You are using the following Amazon EC2 resources in the United States (N. Virginia) Region:

Instances (running)	1	Auto Scaling Groups	0	Capacity Reservations	0
Dedicated Hosts	0	Elastic IPs	0	Instances	1
Key pairs	0	Load balancers	0	Placement groups	0
Security groups	2	Snapshots	0	Volumes	1

Launch instance

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

[Launch instance](#) [Migrate a server](#)

Note: Your instances will launch in the United States (N. Virginia) Region

Service health

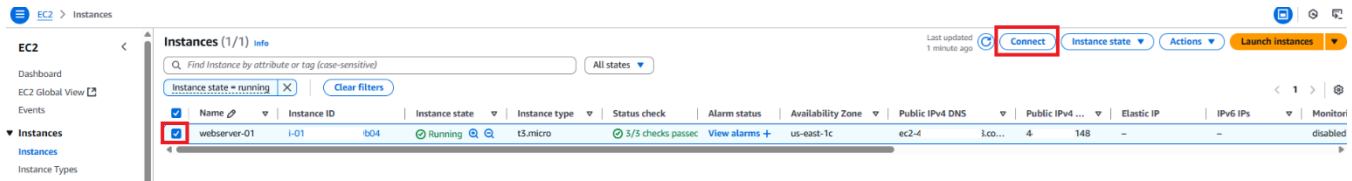
[AWS Health Dashboard](#)

Region
United States (N. Virginia)

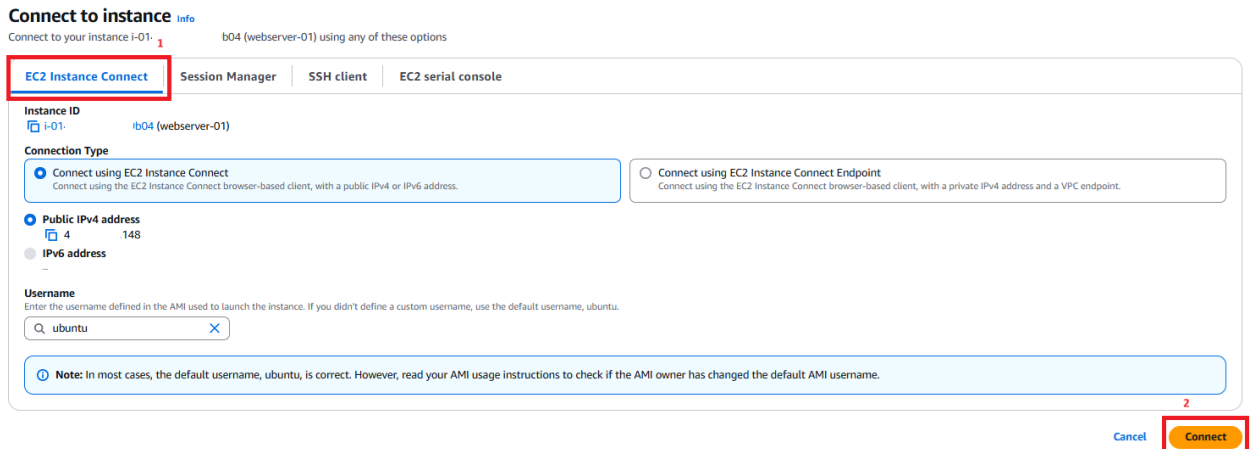
Status
✔ This service is operating normally.

Zones

3. Select the **webserver-01** instance and click **Connect**.



4. Select **EC2 Instance Connect** and click **Connect** to open a terminal window.



5. To meet company standards, we need to make sure that our server is fully up to date. We need to install an Apache web server, the AWS CLI tool and MySQL client (but we don't need to configure a database). Also, we need to set up an *index.html* file that will display some details about this server.
6. Let's get started by updating our server. Update and install the packages using the same password as before. Type the command:

sudo apt-get update && sudo apt-get upgrade -y

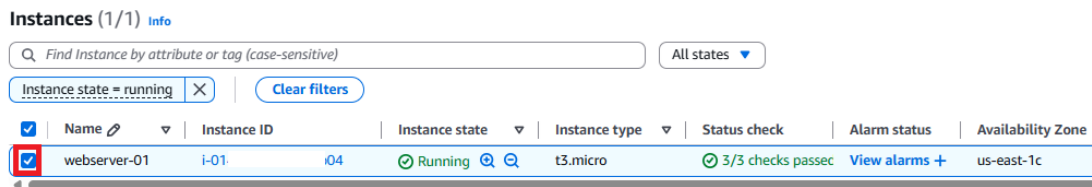
```
cloud user@cloud:~$ sudo apt-get update && sudo apt-get upgrade -y
[sudo] password for cloud user:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [128 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3865 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [588 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [18.0 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3724 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [521 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [604 B]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1265 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [304 kB]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [29.3 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.7 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [8316 B]
Get:16 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3474 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 c-n-f Metadata [688 B]
71% [6 Translation-en store 0 B]
```

7. Note: It may take a few minutes to complete.
8. Install the **apache2** web server: **sudo apt-get install apache2 -y**

```
cloud-user@:~$ sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser openssl-blacklist
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libjansson4 liblua5.2-0 ssl-cert
0 upgraded, 11 newly installed, 0 to remove and 3 not upgraded.
Need to get 1874 kB of archives.
After this operation, 8121 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libapr1 amd64 1.6.5-1ubuntu1.1 [91.5 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1 amd64 1.6.1-4ubuntu2.2 [85.1 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-4ubuntu2.2 [10.5 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-ldap amd64 1.6.1-4ubuntu2.2 [8752 B]
```

9. Once installed, return to the AWS Management Console to confirm the apache2 install was successful:

- Click the checkbox next to webserver-01.
- Scroll down to the Details section of the page and copy the Public IPv4 address.



- Paste the IP address in the address bar of a new browser tab. The page is displayed, which means that we successfully installed Apache.



Note: If using the open address link, you may receive an error that the site can't be reached. This is because the link defaults to HTTPS instead of HTTP. In the address URL, change HTTPS to HTTP to load the Apache2 default welcome page.

1.2. Set Up the AWL CLI Tool & MySQL Client

1. Now, we need to install the AWS CLI tool. To do this, we're going to need to download a file, and this will be a ZIP file. In order for us to use that file, we need to unzip it. So, let's install the unzip tool.
2. Return to the terminal window and install the unzip tool: **sudo apt-get install unzip -y**

```
cloud_user@ip-10-0-1-10:~$ sudo apt-get install unzip -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  zip
The following NEW packages will be installed:
  unzip
0 upgraded, 1 newly installed, 0 to remove and 3 not upgraded.
Need to get 169 kB of archives.
After this operation, 593 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 unzip amd64 6.0-25ubuntu1.2 [169 kB]
Fetched 169 kB in 0s (6761 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 126183 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-25ubuntu1.2_amd64.deb ...
Unpacking unzip (6.0-25ubuntu1.2) ...
Setting up unzip (6.0-25ubuntu1.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
cloud_user@ip-10-0-1-10:~$
```

3. Download the AWS CLI tool: **curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"**

```
cloud_user@ip-10-0-1-10:~$ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 67.1M  100 67.1M    0     0  247M      0 --:--:-- --:--:-- --:--:-- 247M
cloud_user@ip-10-0-1-10:~$
```

4. Unzip the file: **unzip awscliv2.zip**

```
cloud_user@ip-10-0-1-10:~$ unzip awscliv2.zip
```

5. Install the AWS CLI tool (we'll use the script included in that .zip file): **sudo ./aws/install**

```
cloud_user@ip-10-0-1-10:~$ sudo ./aws/install
You can now run: /usr/local/bin/aws --version
cloud_user@ip-10-0-1-10:~$
```

6. Verify AWS CLI version 2 has been installed: **aws - -version**

```
cloud_user@ip-10-0-1-10:~$ aws --version
aws-cli/2.26.1 Python/3.13.2 Linux/5.15.0-1075-aws exe/x86_64.ubuntu.20
cloud_user@ip-10-0-1-10:~$
```

7. To edit the web page's *index.html* file, you'll need to grant user access to the file: **sudo chmod 777 /var/www/html/index.html**

```
cloud_user@ip-10-0-1-10:~$ sudo chmod 777 /var/www/html/index.html
```


Note: In a production environment, you would never want to give full permissions to a file like this. But since this is just a demo environment, we can go ahead and break this rule so that we can edit this file.

8. To get instance metadata about the server's Availability Zone, enter the following command. Observe the Availability Zone is listed at the front of the username in the result: **curl http://169.254.169.254/latest/meta-data/placement/availability-zone**

```
cloud user@ip-100-100-100-100:~$ curl http://169.254.169.254/latest/meta-data/placement/availability-zone
us-east-1c
cloud user@ip-100-100-100-100:~$
```

Note: If you don't see any output, wait a moment and retry the command. So, **us-east-1c** is our availability zone (that's the kind of information that we want to put in our index file).

169.254.169.254 is an IP address from the reserved [IPv4 Link Local Address](#) space 169.254.0.0/16 (169.254.0.0 through 169.254.255.255). Similar to the private address ranges in [RFC-1918](#) (10.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16) in the sense that this block also can't be used on the Internet, Link Local is further restricted to being unreachable via any router - by design, they only exist on the directly connected network.

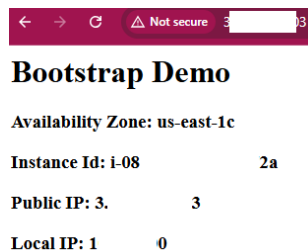
AWS needed to create a service endpoint accessible from any system and the selection of an address in this block allows it to avoid conflict with the commonly used IP address space. Clever choice. Presumably this specific address within the block was chosen for its aesthetic appeal or being easy to remember.

9. Add the Availability Zone, instance ID, public IP, and local IP instance metadata to the index.html file (paste all of this into the terminal):

```
echo '<html><h1>Bootstrap Demo</h1><h3>Availability Zone: ' >
/var/www/html/index.html curl http://169.254.169.254/latest/meta-
data/placement/availability-zone >> /var/www/html/index.html echo '</h3>
<h3>Instance Id: ' >> /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html echo
'</h3> <h3>Public IP: ' >> /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html echo
'</h3> <h3>Local IP: ' >> /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html echo
'</h3></html>' >> /var/www/html/index.html
```

```
cloud_user@ip-10.0.0.10:~$ echo '<html><h1>Bootstrap Demo</h1><h3>Availability Zone: ' > /var/www/html/index.html
cloud_user@ip-10.0.0.10:~$ curl http://169.254.169.254/latest/meta-data/placement/availability-zone >> /var/www/html/index.html
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
00 10 100 10 0 0 10000 0 ----- --:--:-- 10000
cloud_user@ip-10.0.0.10:~$ echo '</h3> <h3>Instance Id: ' >> /var/www/html/index.html
cloud_user@ip-10.0.0.10:~$ curl http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
00 19 100 19 0 0 19000 0 ----- --:--:-- 19000
cloud_user@ip-10.0.0.10:~$ echo '</h3> <h3>Public IP: ' >> /var/www/html/index.html
cloud_user@ip-10.0.0.10:~$ curl http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
00 13 100 13 0 0 6500 0 ----- --:--:-- 6500
cloud_user@ip-10.0.0.10:~$ echo '</h3> <h3>Local IP: ' >> /var/www/html/index.html
cloud_user@ip-10.0.0.10:~$ curl http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
00 10 100 10 0 0 3333 0 ----- --:--:-- 3333
cloud_user@ip-10.0.0.10:~$ echo '</h3></html> ' >> /var/www/html/index.html
cloud_user@ip-10.0.0.10:~$
```

10. Navigate back to the Apache web page, and refresh it to view the results of the changes you made.



11. Return to the terminal and install **mysql**: **sudo apt-get install mysql-server -y**

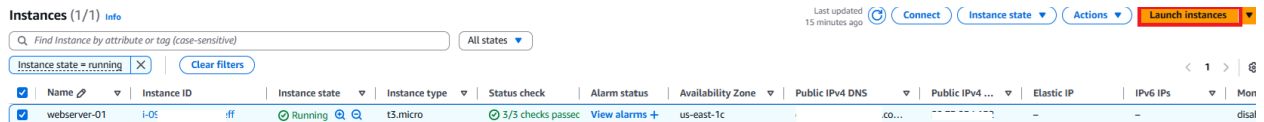
```
cloud user@ip-10-0-1-17 ~ % sudo apt-get install mysql-server -y
[sudo] password for cloud user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libio-fast-perl libio-poll-perl libipc-run-perl libjson-xs-perl liblibevent-core-2.1-7 libevent-pthreads-2.1-7 libfogi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblua-mediatypes-perl libmccab2 libtimedate-perl liburi-perl mcab-ipadic mcab-ipadic-utf8 mcab-util mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinycsa
The following NEW packages will be installed:
  libio-fast-perl libio-poll-perl libipc-run-perl libjson-xs-perl liblibevent-core-2.1-7 libevent-pthreads-2.1-7 libfogi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblua-mediatypes-perl libmccab2 libtimedate-perl liburi-perl mcab-ipadic mcab-ipadic-utf8 mcab-util mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 25 newly installed, 0 to remove and 3 not upgraded.
Need to get 37.0 MB of archives.
After this operation, 318 MB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 mysql-common all 5.8+1.0-Subunit2 [7496 B]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/updates/main amd64 mysql-client-core-8.0 amd64 8.0.41-0ubuntu0.20.04.1 [5085 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/updates/main amd64 mysql-client-8.0 amd64 8.0.41-0ubuntu0.20.04.1 [22.0 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libevent-core-2.1-7 amd64 2.1.11-stable-1 [89.1 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libevent-pthreads-2.1-7 amd64 2.1.11-stable-1 [7372 B]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 libmccab2 amd64 0.996-10build1 [233 kb]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/updates/main amd64 mysql-server-core-8.0 amd64 8.0.41-0ubuntu0.20.04.1 [22.8 MB]
```

Note: It may take a few minutes for MySQL to install.

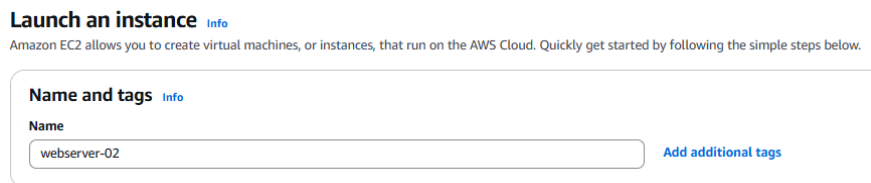
2. Use a Bootstrap Script to Build **webserver-02** and Debug Issues

2.1. Set Up the Script

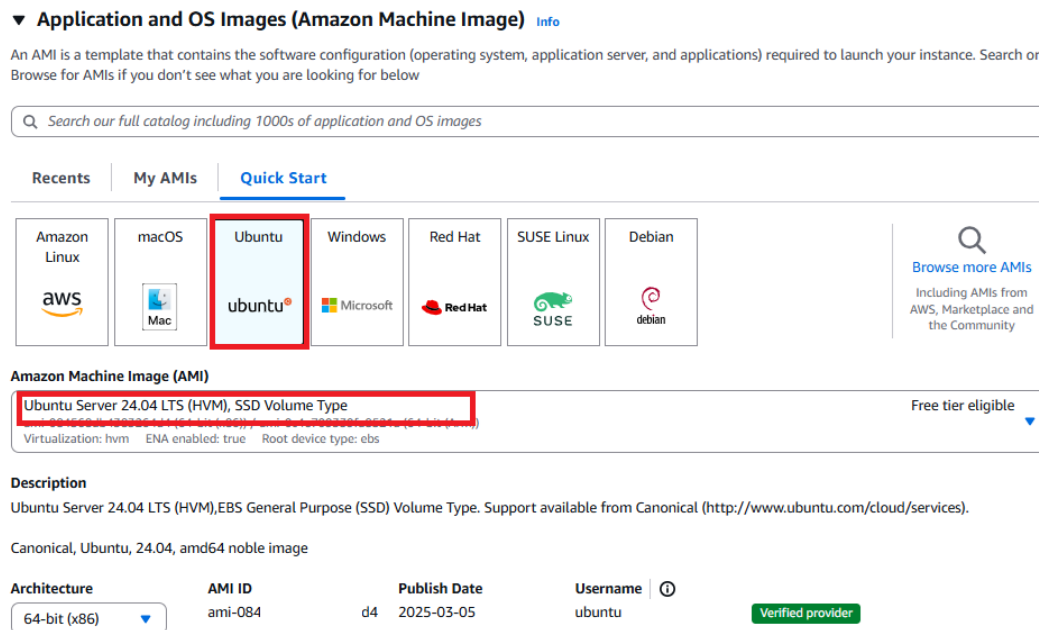
1. Return to the AWS Management Console and navigate to **EC2**.
2. On the EC2 dashboard, click **Launch instances**.



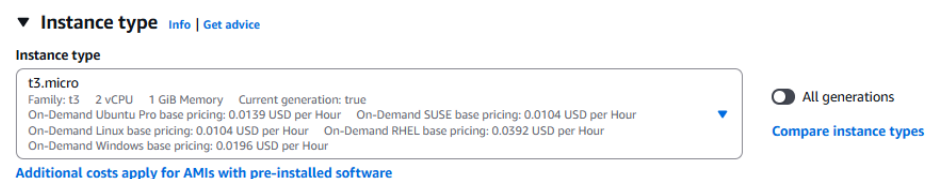
3. In the **Launch an instance** section, under **Name and tags** type **webserver-02**.



4. Scroll down to the **Application and OS Images** (Amazon Machine Image) to select the Ubuntu logo, click the dropdown menu to select **Ubuntu Server 24.04 LTS (HVM), SSD Volume Type**.



5. Scroll down to the **Instance type**, and click the dropdown menu to select **t3.micro**.



6. Under **Key pair (login)**, click the dropdown and select **Proceed without a key pair (Not recommended) Default value**.

▼ **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*

Proceed without a key pair (Not recommended)

Default value ▼

🔄 [Create new key pair](#)

7. Under **Network settings**, click **Edit** and enter the following information:

- **Auto-assign public IP:** Select **Enable** from the dropdown menu.
- **Firewall (security groups):** Choose **Select existing security group**.
- **Common security groups:** Select **EC2SecurityGroup** from the dropdown menu.

▼ **Network settings** [Info](#)

VPC - *required* | [Info](#)

vpc-02b4
10.0.0.0/16

id8

▼



Subnet | [Info](#)

subnet-079

a9

VPC: vpc-02

18

Owner: 9

7

Availability Zone: us-east-1b

Zone type: Availability Zone

IP addresses available: 251

CIDR: 1

.0/24)

▼



[Create new subnet](#)

Auto-assign public IP | [Info](#)

Enable

▼

[Additional charges apply when outside of free tier allowance](#)

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

▼

cfst-3356-d953f51f6

4b834ba-

EC2SecurityGroup

xRd

×



[Compare security group rules](#)

VPC: vpc-0

d8

sg-03

if

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

8. Under **Advanced details**, click the dropdown arrow to expand.

► **Advanced details** [Info](#)

9. Scroll down to **Metadata version** and select **V1 and V2 (token optional)**.

Metadata version | [Info](#)

V1 and V2 (token optional)

▼

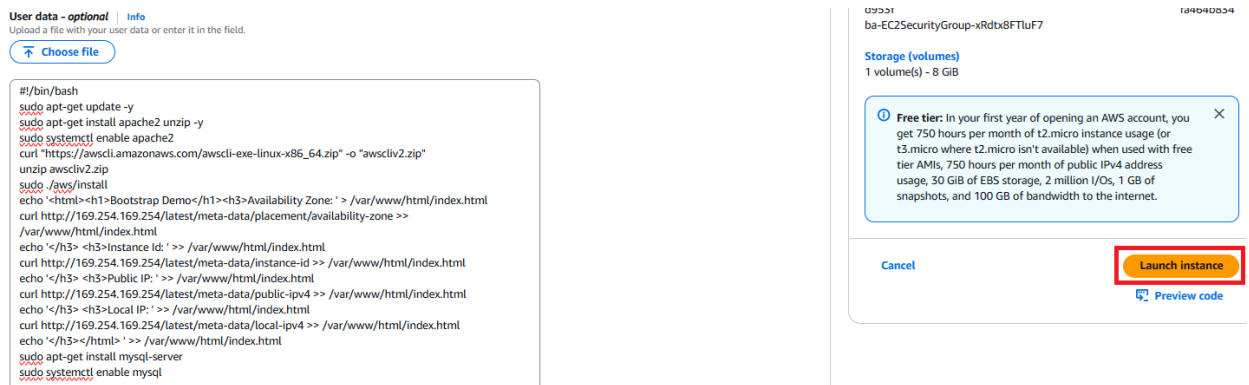


EC2 recommends using metadata version 2 unless you explicitly require metadata version 1.

10. Scroll down to **User data** and paste in the following bootstrap script:

```
#!/bin/bash sudo apt-get update -y sudo apt-get install apache2 unzip -y sudo systemctl enable apache2 curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o
```

```
"awscli2.zip" unzip awscli2.zip sudo ./aws/install echo '<html><h1>Bootstrap
Demo</h1><h3>Availability Zone: ' > /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/placement/availability-zone >>
/var/www/html/index.html echo '</h3> <h3>Instance Id: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html echo
'</h3> <h3>Public IP: ' >> /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html echo
'</h3> <h3>Local IP: ' >> /var/www/html/index.html curl
http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html echo
'</h3></html>' >> /var/www/html/index.html sudo apt-get install mysql-server sudo
systemctl enable mysql
```



Note: This script is going to do everything we just did manually to **webserver-01**, but it's going to do it while the instance is being created.

We're getting the server up to date, we're installing apache2 and unzip, and we're going to enable apache2. We're going to pull down our AWS CLI tool and unzip it and then install it. Then, we're creating our *index.html* file with all the data that we provided previously. Finally, we'll install our **mysql** server and enable that.

11. Click **Launch instance**.

2.2. Connect to and View **webserver-02**

- Once the instance launch has been successfully initiated, click **View all instances**.
- Click the refresh button, if webserver-02 is not displayed. It may take a few minutes for **webserver-02** to complete its configuration.
- Once the **webserver-02** instance has passed status checks, select this instance, and click **Connect**.

Instances (1/2) [Info](#) Last updated less than a minute ago [Refresh](#) [Connect](#)

Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Pub
<input type="checkbox"/>	webserver-01	i-0...	ff	Running	t3.micro	3/3 checks passed	us-east-1c	ec	132.co... 52.7
<input checked="" type="checkbox"/>	webserver-02	i-0...	i2f	Running	t3.micro	3/3 checks passed	us-east-1b	ec	225.co... 52.9

4. Select **EC2 Instance Connect** and click **Connect**.

Connect to instance info
Connect to your instance i-02: f (webserver-02) using any of these options

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

Instance ID: i-4 (webserver-02)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

☒ Public IPv4 address
5 25
☐ IPv6 address
-

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

Q ubuntu X

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel **Connect**

5. In the terminal, check if Apache was installed correctly. The output should display **Active: active (running)**: **sudo systemctl status apache2**

```
ubuntu@ip-10-0-1-10:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-14 16:54:15 UTC; 16min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 1943 (apache2)
      Tasks: 55 (limit: 1077)
     Memory: 5.8M (peak: 6.1M)
        CPU: 92ms
    CGroup: /system.slice/apache2.service
            └─1943 /usr/sbin/apache2 -k start
              └─1945 /usr/sbin/apache2 -k start
                └─1946 /usr/sbin/apache2 -k start

Apr 14 16:54:15 ip-10-0-1-10 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Apr 14 16:54:15 ip-10-0-1-10 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-10-0-1-10:~$
```

6. Verify **apache2** is running. The output should display a few **apache2** processes: **ps aux | grep apache**

```
ubuntu@ip-10-0-1-10:~$ ps aux | grep apache
root        1943  0.0  0.5 6876 5020 ?        Ss   16:54   0:00 /usr/sbin/apache2 -k start
www-data    1945  0.0  0.6 1213024 6448 ?        Sl   16:54   0:00 /usr/sbin/apache2 -k start
www-data    1946  0.0  0.6 1212968 5680 ?        Sl   16:54   0:00 /usr/sbin/apache2 -k start
www-data    2056  0.0  0.1 3616 1544 ?        Ss   16:54   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod_cache_disk -l 300M -n
ubuntu      2953  0.0  0.2 7076 2176 pts/0    S+   17:12   0:00 grep --color=auto apache
ubuntu@ip-10-0-1-10:~$
```

7. Verify **mysql** is running: **sudo systemctl status mysql**
8. Try using **mysqld**: **sudo systemctl status mysqld**
9. Check for any running **mysql** processes: **ps aux | grep mysql**
10. Try to start the mysql service: **sudo systemctl start mysql**

Note: These commands should all return an error that the mysql service was not found.

```

ubuntu@ip-10-0-1-10:~$ sudo systemctl status mysql
Unit mysql.service could not be found.
ubuntu@ip-10-0-1-10:~$ sudo systemctl status mysqld
Unit mysqld.service could not be found.
ubuntu@ip-10-0-1-10:~$ ps aux | grep mysql
ubuntu 2973 0.0 0.2 7076 2176 pts/0 S+ 17:16 0:00 grep --color=auto mysql
ubuntu@ip-10-0-1-10:~$ sudo systemctl start mysql
Failed to start mysql.service: Unit mysql.service not found.
ubuntu@ip-10-0-1-10:~$

```

11. Use **curl** to retrieve the **user-data**: **curl http://169.254.169.254/latest/user-data**
12. At the bottom of the script, notice the following code: “**sudo apt-get install mysql-server**”
13. Observe the code is missing the **-y** flag needed for mysql to automatically install without a user prompt.

```

ubuntu@ip-10-0-1-10:~$ curl http://169.254.169.254/latest/user-data
#!/bin/bash
sudo apt-get update -y
sudo apt-get install apache2 unzip -y
sudo systemctl enable apache2
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
echo '<html><h1>Bootstrap Demo</h1><h3>Availability Zone: ' > /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/placement/availability-zone >> /var/www/html/index.html
echo '</h3> <h3>Instance Id: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html
echo '</h3> <h3>Public IP: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html
echo '</h3> <h3>Local IP: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html
echo '</h3></html>' >> /var/www/html/index.html
sudo apt-get install mysql-server
sudo systemctl enable mysqlubuntu@ip-10-0-1-10:~$

```

14. Install **mysql-server** manually: **sudo apt-get install mysql-server -y**
15. Enable the **mysql** service: **sudo systemctl enable mysql**

```

ubuntu@ip-10-0-1-10:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
ubuntu@ip-10-0-1-10:~$

```

3. Use a Fixed Bootstrap Script to Build **webserver-03**

3.1. Set Up the Script

1. Return to the AWS Management Console and navigate to **EC2**.
2. On the EC2 dashboard, click **Launch instances**.

Instances (1/1) [info](#)

Find Instance by attribute or tag (case-sensitive) All states ▾

Instance state: **running**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Mon
webserver-01	i-0f...	Running	t3.micro	3/3 checks passed	View alarms	us-east-1c	co...				disal

3. In the **Launch an instance** section, under **Name and tags** type **webserver-03**.

Launch an instance [Info](#)

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 [Info](#)

Name

webserver-03

[Add additional tags](#)

4. Scroll down to the **Application and OS Images** (Amazon Machine Image) to select the Ubuntu logo, click the dropdown menu to select **Ubuntu Server 24.04 LTS (HVM), SSD Volume Type**.

▼ 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

My AMIs

Quick Start



[Browse more AMIs](#)
Including AMIs from
AWS, Marketplace and
the Community

Amazon Machine Image (AMI)

Ubuntu Server 24.04 LTS (HVM), SSD Volume Type

Free tier eligible

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

Description

Ubuntu Server 24.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Canonical, Ubuntu, 24.04, amd64 noble image

Architecture

64-bit (x86)

AMI ID

ami-084

Publish Date

d4 2025-03-05

Username

ubuntu

Verified provider

5. Scroll down to the **Instance type**, and click the dropdown menu to select **t3.micro**.

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t3.micro

Family: t3 2 vCPU 1 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0139 USD per Hour On-Demand SUSE base pricing: 0.0104 USD per Hour
On-Demand Linux base pricing: 0.0104 USD per Hour On-Demand RHEL base pricing: 0.0392 USD per Hour
On-Demand Windows base pricing: 0.0196 USD per Hour

☐ All generations

[Compare instance types](#)

[Additional costs apply for AMIs with pre-installed software](#)

6. Under **Key pair (login)**, click the dropdown and select **Proceed without a key pair (Not recommended)** Default value.

▼ 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

Proceed without a key pair (Not recommended)

Default value

[Create new key pair](#)

7. Under **Network settings**, click **Edit** and enter the following information:

- **Auto-assign public IP:** Select **Enable** from the dropdown menu.
- **Firewall (security groups):** Choose **Select existing security group**.
- **Common security groups:** Select **EC2SecurityGroup** from the dropdown menu.

▼ Network settings Info

VPC - required Info

vpc-02b4
10.0.0.0/16

id8

Subnet Info

subnet-079

a9

VPC: vpc-02

id8

Owner: 9

7

Availability Zone: us-east-1b

Zone type: Availability Zone

IP addresses available: 251

CIDR: 1

.0/24)

Auto-assign public IP Info

Enable

Additional charges apply when outside of free tier allowance

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

cfst-3356-d953f51f6

4b834ba-

EC2SecurityGroup

sg-03

VPC: vpc-0

id8

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

- Under **Advanced details**, click the dropdown arrow to expand.

► **Advanced details** Info

- Scroll down to **Metadata version** and select **V1 and V2 (token optional)**.

Metadata version Info

V1 and V2 (token optional)

⚠ EC2 recommends using metadata version 2 unless you explicitly require metadata version 1.

- Scroll down to **User data** and paste in the following bootstrap script (This time, the **-y** flag for mysql has been added.):

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install apache2 unzip -y
sudo systemctl enable apache2
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
echo '<html><h1>Bootstrap Demo</h1><h3>Availability Zone: ' >
/var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/placement/availability-zone >>
/var/www/html/index.html
echo '</h3> <h3>Instance Id: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html
echo '</h3> <h3>Public IP: ' >> /var/www/html/index.html
```

```
curl http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html
echo '</h3><h3>Local IP: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html
echo '</h3></html>' >> /var/www/html/index.html
sudo apt-get install mysql-server -y
sudo systemctl enable mysql
```

User data - optional [Info](#)

Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install apache2 unzip -y
sudo systemctl enable apache2
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscli2.zip"
unzip awscli2.zip
sudo ./aws/install
echo '<html><h1>Bootstrap Demo</h1><h3>Availability Zone: ' > /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/placement/availability-zone >> /var/www/html/index.html
echo '</h3><h3>Instance ID: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/instance-id >> /var/www/html/index.html
echo '</h3><h3>Public IP: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html
echo '</h3><h3>Local IP: ' >> /var/www/html/index.html
curl http://169.254.169.254/latest/meta-data/local-ipv4 >> /var/www/html/index.html
echo '</h3></html>' >> /var/www/html/index.html
sudo apt-get install mysql-server -y
sudo systemctl enable mysql
```

Firewall (security group)

cfst
d95
ba-EC2SecurityGroup-xRdtx8FTluF7 i464b834

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Preview code](#)

11. Click **Launch instance**.

3.2. Connect to and View **webserver-03**

- Once the instance launch has been successfully initiated, click **View all instances**.
- Click the refresh button, if **webserver-02** is not displayed. It may take a few minutes for **webserver-03** to complete its configuration.
- While you are waiting for **webserver-03** to complete its setup, check if the *index.html* file for **webserver-02** has been configured correctly.
- Select **webserver-02** from the **Instances** list and copy the **Public IPv4 address**.

Instances (1/3) [Info](#)

Find Instance by attribute or tag (case-sensitive) [All states](#)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	webserver-01	i-0f	f	Running	t3.micro	3/3 checks passed	us-east-1c
<input checked="" type="checkbox"/>	webserver-02	i-0f	2f	Running	t3.micro	3/3 checks passed	us-east-1b
<input type="checkbox"/>	webserver-03	i-0f	f	Running	t3.micro	3/3 checks passed	us-east-1b

i-0240815e805a1e62f (webserver-02)

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

▼ Instance summary [Info](#)

Instance ID

i-0f f

IPv6 address

-

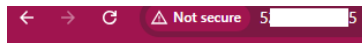
Public IPv4 address

5. 15 | [open address](#)

Instance state

Running

- Paste the IP address in a new browser tab to access the Apache web page. Observe the information that's returned.



Bootstrap Demo

Availability Zone: us-east-1b

Instance Id: i-02f

Public IP: 54.184.253.5

Local IP: 10.0.3.4

- Navigate back to the EC2 dashboard and click the refresh button to check the status of **webserver-03**.
- Once the **webserver-03** instance has passed status checks, select this instance, and click **Connect**.

Instances (1/3)

Info

Last updated 3 minutes ago

Refresh

Connect

Find Instance by attribute or tag (case-sensitive)

All states

<div><div></div></div>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Pub
<div><input type="checkbox"/></div>	webserver-01	i-01ff	<div>Running</div>	t3.micro	<div>3/3 checks passed</div>	<div>View alarms +</div>	us-east-1c	ec2-154.184.253.5	5
<div><input type="checkbox"/></div>	webserver-02	i-022f	<div>Running</div>	t3.micro	<div>3/3 checks passed</div>	<div>View alarms +</div>	us-east-1b	ec2-154.184.253.5	5
<div><input checked="" type="checkbox"/></div>	webserver-03	i-02f	<div>Running</div>	t3.micro	<div>3/3 checks passed</div>	<div>View alarms +</div>	us-east-1b	ec2-184.104.253.5	2...

- Select **EC2 Instance Connect** and click **Connect** to connect to the **webserver-03** instance in a new terminal window.

Connect to instance [Info](#)

Connect to your instance i-02f (webserver-03) using any of these options

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

Instance ID: i-02f (webserver-03)

Connection Type

☒ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 or IPv6 address.

☐ Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IPv4 address
☒ 54.184.253.5
☐ IPv6 address

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ubuntu.

Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

[Cancel](#) [Connect](#)

- In the terminal, check if Apache was installed correctly. The output should display Active: active (running): **sudo systemctl status apache2**

```

ubuntu@ip-10.0.0.10:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-14 17:32:32 UTC; 9min ago
     Docs: https://httpd.apache.org/docs/2.4/
  Main PID: 1942 (apache2)
    Tasks: 55 (limit: 1077)
  Memory: 5.4M (peak: 5.8M)
     CPU: 64ms
  CGroup: /system.slice/apache2.service
          └─1942 /usr/sbin/apache2 -k start
            └─1944 /usr/sbin/apache2 -k start
              └─1945 /usr/sbin/apache2 -k start

Apr 14 17:32:32 ip-10.0.0.10 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Apr 14 17:32:32 ip-10.0.0.10 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-10.0.0.10:~$

```

10. Verify apache2 is running. The output should display a few apache2 processes: **ps aux | grep apache**

```

ubuntu@ip-10.0.0.10:~$ ps aux | grep apache
root        1942  0.0  0.4  6876  3900 ?        Ss   17:32   0:00 /usr/sbin/apache2 -k start
www-data    1944  0.0  0.4 1212960 4432 ?        Sl   17:32   0:00 /usr/sbin/apache2 -k start
www-data    1945  0.0  0.4 1212960 4560 ?        Sl   17:32   0:00 /usr/sbin/apache2 -k start
www-data    2056  0.0  0.1  3616  1416 ?        Ss   17:32   0:00 /usr/bin/htcacheclean -d 120 -p /var/cache/apache2/mod_cache_disk -l 300M -n
ubuntu@ip-10.0.0.10:~$

```

11. Verify mysql was installed: **systemctl status mysql**

12. This time, it is running.

```

ubuntu@ip-10.0.0.10:~$ systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-04-14 17:33:29 UTC; 9min ago
     Main PID: 3066 (mysqld)
    Status: "Server is operational"
      Tasks: 37 (limit: 1077)
  Memory: 353.5M (peak: 379.0M)
     CPU: 4.404s
  CGroup: /system.slice/mysql.service
          └─3066 /usr/sbin/mysqld

Apr 14 17:33:28 ip-10.0.0.10 systemd[1]: Starting mysql.service - MySQL Community Server...
Apr 14 17:33:29 ip-10.0.0.10 systemd[1]: Started mysql.service - MySQL Community Server.
ubuntu@ip-10.0.0.10:~$

```

13. Confirm mysql processes: **ps aux | grep mysql**

14. Verify AWS CLI tool was installed: **aws --version**

```

ubuntu@ip-10.0.0.10:~$ ps aux | grep mysql
mysql      3066  0.7 39.2 1786072 367448 ?        Ssl  17:33   0:04 /usr/sbin/mysqld
ubuntu@ip-10.0.0.10:~$
ubuntu@ip-10.0.0.10:~$ aws --version
aws-cli/2.26.1 Python/3.13.2 Linux/x86_64-aws exe/x86_64.ubuntu.24
ubuntu@ip-10.0.0.10:~$

```

15. Navigate back to the EC2 dashboard and copy the **Public IP** address for **webserver-03**.

☑ webserv-03 i- 9f Running 🔍 t3.micro 3/3 checks passed

i-0. 9f (webserv-03)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary [Info](#)

Instance ID i-0f9f9f9f

Public IPv4 address 3.141.59.26 | [open address](#)

Public IPv4 address copied

16. Paste the IP address in a new tab to confirm the Apache web page for **webserv-03**.

← → ↻ ⚠ Not secure 3.141.59.26

Bootstrap Demo

Availability Zone: us-east-1b

Instance Id: i-0f9f9f9f

Public IP: 3.141.59.26

Local IP: 10.0.1.10