


# Draft a document detailing instruction for how to fulling install our Omeka setup

Cover the following:

- Creating EC2 instance – [sharepoint](#) (but it is for downloading omeka-s) so do until EC2 instance
  - Installing LAMP – [github](#)
  - Installing Omeka
  - Installing the Module
  - Importing using the module
- Use screenshots for clarity


## Launch EC2 Instance

1. Create an amazon web server account at [aws.amazon.com](https://aws.amazon.com) or [signup page](#).



**Explore Free Tier products with a new AWS account.**

To learn more, visit [aws.amazon.com/free](https://aws.amazon.com/free).



### Sign up for AWS

**Root user email address**  
Used for account recovery and some administrative functions

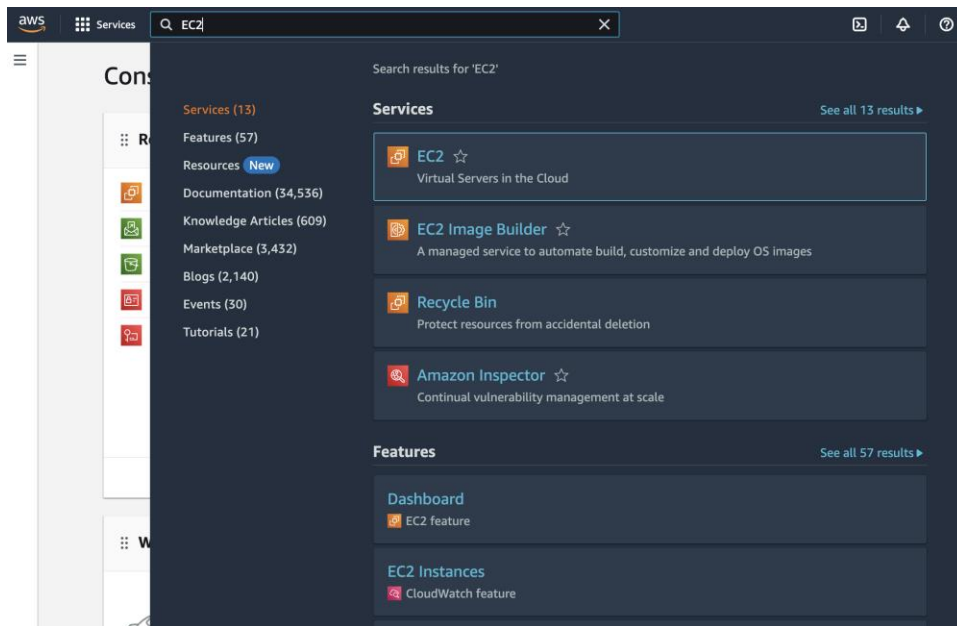
**AWS account name**  
Choose a name for your account. You can change this name in your account settings after you sign up.

**Verify email address**

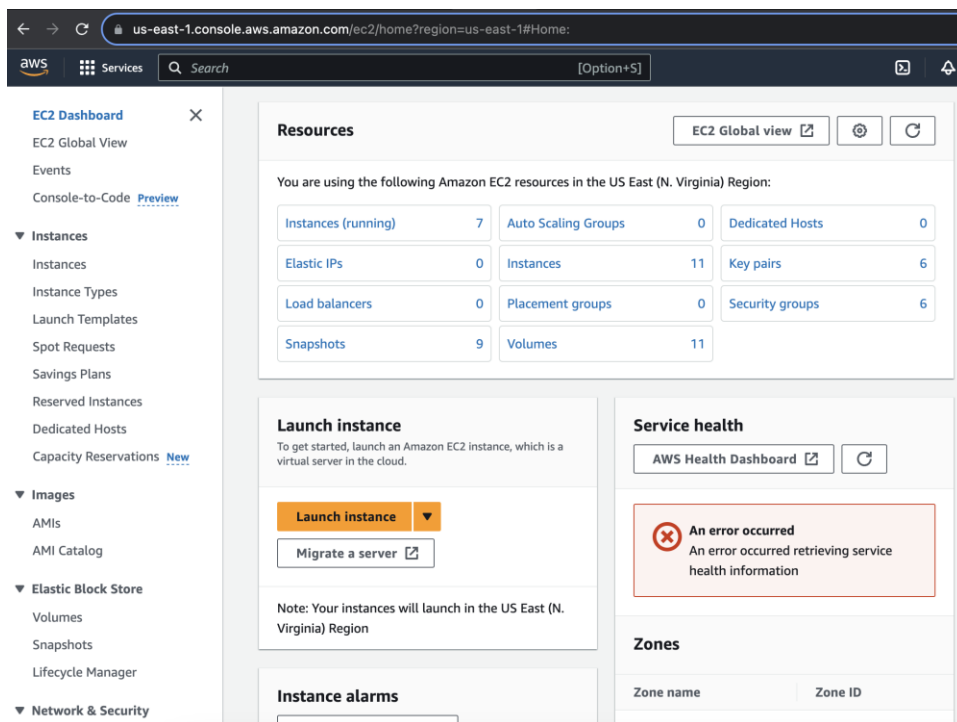
OR

**Sign in to an existing AWS account**

2. After setup, use the search box to find the link to EC2 (Virtual Servers in the Cloud) and click on it.



3. Scroll down and click on **the Launch Instance** button.



4. Name your instance anything you'd like, then under "Application and OS Images (Amazon Machine Image)" select "Ubuntu Server 22.04 LTS (HVM), SSD Volume Type" as your Amazon Machine Image (AMI). Under Instance type choose "t2.micro".

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

Recents

My AMIs

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)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Free tier eligible

ami-07d9b9ddc6cd8dd30 (64-bit (x86)) / ami-0568072f574d822a4 (64-bit (Arm))

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

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-02-07

Architecture

AMI ID

64-bit (x86)

ami-07d9b9ddc6cd8dd30

Verified provider

▼ Instance type

Info | Get advice

Instance type

t2.micro

Free tier eligible

Family: t2 1 vCPU 1 GiB Memory Current generation: true

On-Demand Windows base pricing: 0.0162 USD per Hour

On-Demand SUSE base pricing: 0.0116 USD per Hour

On-Demand RHEL base pricing: 0.0716 USD per Hour

On-Demand Linux base pricing: 0.0116 USD per Hour

All generations

Compare instance types

Additional costs apply for AMIs with pre-installed software

- Under Key pair click “Create new key pair”. Name it anything you’d like, ideally without spaces, and keep the default RSA and .pem settings. After clicking “Create key pair” a .pem file will download. We’ll be moving this file elsewhere later.

### Create key pair

×

#### Key pair name

Key pairs allow you to connect to your instance securely.

testing\_key\_pairs

The name can include up to 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

- Under Network Settings, choose “Create security group”, then select “Allow SSH traffic from” and “Allow HTTP traffic from the internet”.

▼ Network settings

Info

Edit

Network

Info

vpc-0f0911228c5151595

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

We'll create a new security group called 'launch-wizard-4' with the following rules:

☒ Allow SSH traffic from
 

Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet
 

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
 

To set up an endpoint, for example when creating a web server

⚠ 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.

×

## 7. Review your selections then click “Launch Instance”

AWS

Services

Search

[Option+5]

N. Virginia

Aayush\_Bhatta @ 6235-3987-751

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

We'll create a new security group called 'launch-wizard-4' with the following rules:

☒ Allow SSH traffic from
 

Helps you connect to your instance

Anywhere

0.0.0.0/0

☐ Allow HTTPS traffic from the internet
 

To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
 

To set up an endpoint, for example when creating a web server

⚠ 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.

×

▼ Configure storage

Info

Advanced

1x 8 GIB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage

×

▼ Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more

ami-07d9b9ddc6c8bd830

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

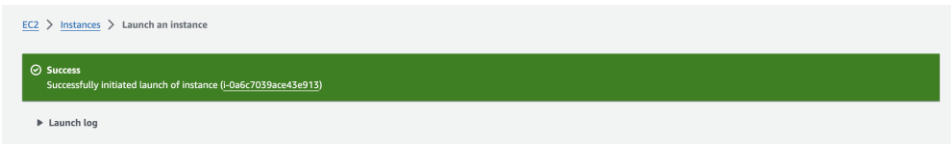
Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of

Cancel

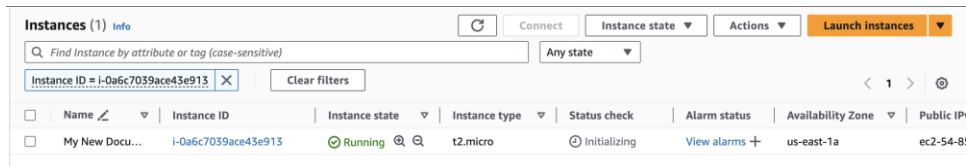
Launch instance

Review commands

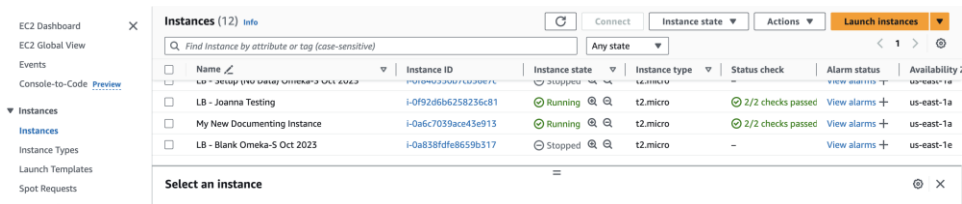
## 8. After launching the instance, you will see something like this in the screen.



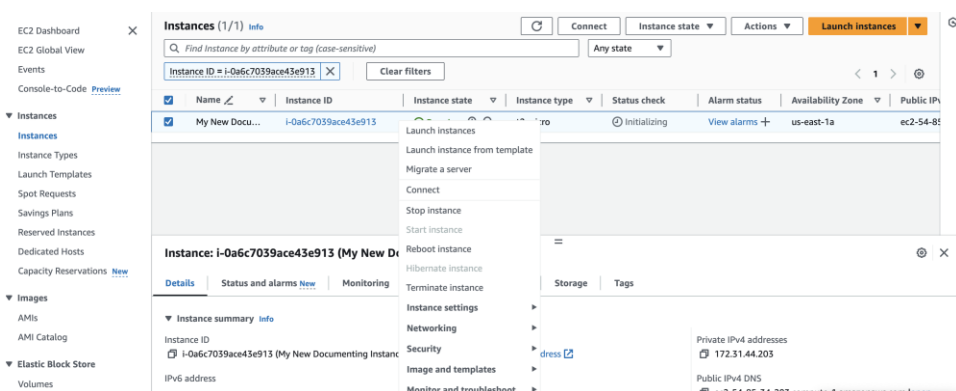
- Click on the link attached with the **instance id** (i-0a6c7039ace43e913 in this case) that you can see after “Successfully initiated launch of instance” and it will take you to the screen like below:



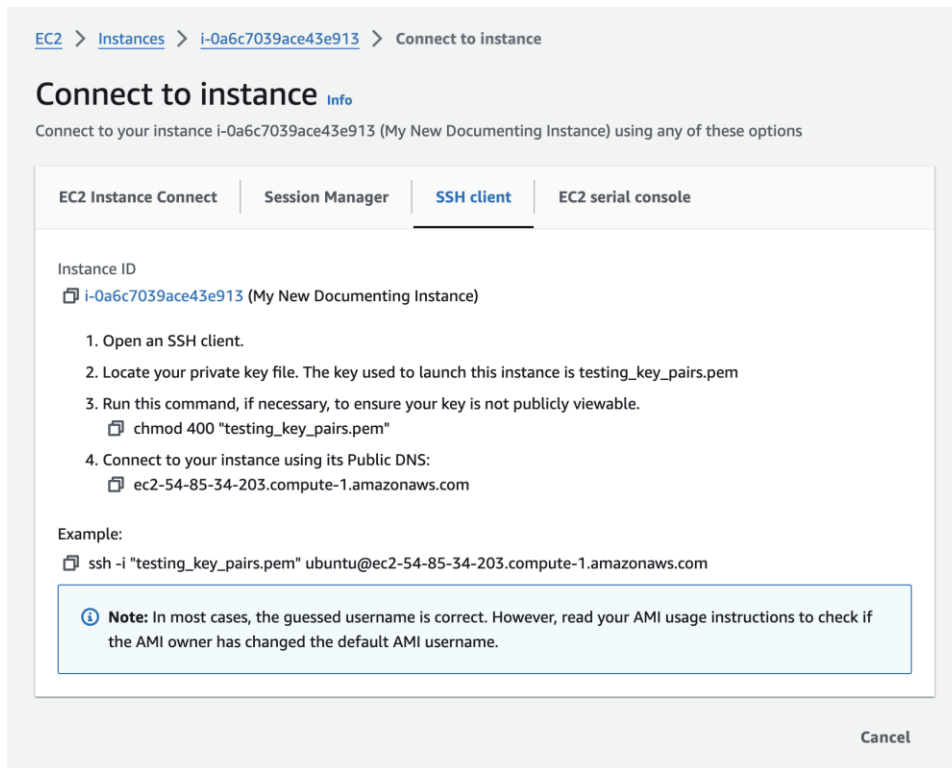
OR you can click on the link attached with “Instances” keyword in the top left corner after “EC2 >” that can be seen as “EC2 > Instances > Launch an instance” and after that you can see many instances if there are others along with the instance you recently created like below:



- After you see your instance either way, on the subsequent panel, RIGHT click on instance we created and select ‘Connect’.



- Select “SSH Client” and keep this window open for later.



12. Open the terminal on your local device (Applications/Utilities/Terminal for OS X users) *\*Note that your color scheme may vary from these screenshots.*

Change directories to the current user's home directory:

```
cd ~ls
```

Check if there is an existing .ssh directory:

```
ls -a
```

```
aayushbhatta@administrators-MacBook-Pro ~ % ls -a
.                  .lessht           Documents
..                .mysql_history    Downloads
.CFUserTextEncoding .npm              Library
.DS_Store         .vscode           Movies
.Trash            .zprofile         Music
.bash_history     .zsh_history      Pictures
.composer         .zsh_sessions     Public
.gitconfig        Desktop
aayushbhatta@administrators-MacBook-Pro ~ %
```

If there is no such directory, create a new directory called .ssh:

```
mkdir -p ~/.ssh
```

```
aayushbhatta@administrators-MacBook-Pro ~ % mkdir -p ~/.ssh
aayushbhatta@administrators-MacBook-Pro ~ % ls -a
.                .lessht         Desktop
..               .mysql_history  Documents
.CFUserTextEncoding .npm            Downloads
.DS_Store        .ssh            Library
.Trash           .vscode         Movies
.bash_history    .zprofile       Music
.composer        .zsh_history    Pictures
.gitconfig       .zsh_sessions  Public
aayushbhatta@administrators-MacBook-Pro ~ %
```

Here you can see .ssh folder.

13. Move the .pem file earlier downloaded from AWS, out of the Downloads folder and into the .ssh directory:

```
mv ~/Downloads/[YourKeyPairName].pem ~/.ssh
```

*\*Note: YourKeyPairName.pem is what you generated in Amazon EC2 in step 4.*

```
aayushbhatta@administrators-MacBook-Pro ~ % mv ~/Downloads/testing_key_pairs.pem ~/.ssh
```

14. Enter and run the following command. This gives read permissions to owner and no permissions to others and group for all .pem files in the .ssh directory:

```
chmod 400 ~/.ssh/*.pem
```

15. Back in the “Connect to instance” browser window we left open in step 11, you will find an example ssh command. After confirming you are in your .ssh directory in the terminal, run your example ssh command provided in AWS in your terminal. If prompted, type ‘yes’. If you receive “Welcome to Ubuntu 22.04.3”, congratulations, you are now remotely accessing your AWS server instance! Please proceed to step 17 by skipping 16.






## Connect to instance [Info](#)

Connect to your instance i-0a6c7039ace43e913 (Aayush - My New Documenting Instance) using any of these options

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


Instance ID

 i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is testing\_key\_pairs.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "testing_key_pairs.pem"`
4. Connect to your instance using its Public DNS:  
 `ec2-54-85-34-203.compute-1.amazonaws.com`

Example:

 `ssh -i "testing_key_pairs.pem" ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com`

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

Cancel

```
aayushbhatta@administrators-MacBook-Pro .ssh % ssh -i "testing_key_pairs.pem" ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-aws x86_64)
```


16. If for some reason the command does not work or there is an error, return to the "Connect to instance" browser window we left open in step 11, you will find your Public DNS. You'll need this to connect to your EC2 instance from the terminal. Follow the formula below for the command, ensuring your pwd is Users/[yourUsername]/.ssh meaning you are in .ssh directory.


## Connect to instance [Info](#)


Connect to your instance i-0a6c7039ace43e913 (Aayush - My New Documenting Instance) using any of these options

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

Instance ID


 i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is testing\_key\_pairs.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "testing_key_pairs.pem"`

4. Connect to your instance using its Public DNS:  
 `ec2-54-85-34-203.compute-1.amazonaws.com`

Example:

 `ssh -i "testing_key_pairs.pem" ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com`

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

```
ssh -i "YourKeyName.pem" ubuntu@YourPublicDNS
```

*\*Note: You must be in the .ssh directory for this command to work as this command assumes the .pem file is local.*

## Configure Ubuntu for Omeka-S

17. Update Ubuntu libraries. This command will generate a wall of text.

```
sudo apt-get update
```

```
ubuntu@ip-172-31-44-203:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Fetched 229 kB in 1s (329 kB/s)
Reading package lists... Done
```

18. Upgrade Ubuntu, type “Yes” or “Y” if prompted, and use the spacebar to select all the options if prompted “Which services should be restarted?”, then hit return/enter. If you don’t get any issues, then go to step 19 after this.

```
sudo apt-get upgrade
```

However, just in case you get something like below in your terminal, then you should reboot your instance. But before that select enter / return.

```
.ssh — ubuntu@ip-172-31-44-203: ~ — ssh -i testing_key_pairs.pem ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com — 88x29
Scanning linux images...

Package configuration

Pending kernel upgrade

Newer kernel available

The currently running kernel version is 6.2.0-1018-aws which is not the expected
kernel version 6.5.0-1015-aws.

Restarting the system to load the new kernel will not be handled automatically,
so you should consider rebooting.

&ltOk>
```

After you return / enter, if you get the page like below in the terminal select all the options that you can see using “space” and use dropdown to go to other options and select enter/return after selecting everything like below.

```
.ssh — ubuntu@ip-172-31-44-203: ~ — ssh -i testing_key_pairs.pem ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com — 88x29
Package configuration

  Daemons using outdated libraries

Which services should be restarted?

[*] dbus.service
[*] networkd-dispatcher.service
[*] packagekit.service
[*] polkit.service
[*] ssh.service
[*] systemd-journald.service
[*] systemd-logind.service
[*] systemd-manager
[*] systemd-networkd.service
[*] systemd-resolved.service
[*] systemd-udev.service
[*] unattended-upgrades.service
[*] user@1000.service

<Ok>          <Cancel>
```

After doing this go back to your instance in AWS and select the instance that you are working with.

Instances (1/13) info

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/> Aayush - My New Documenting Instance	i-0a6c7039ace43e913	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1
<input type="checkbox"/> Aayush - My OmekaS Rosenwald School	i-0cbba6876c39a242b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1
<input type="checkbox"/> Aayush (Terminate?) - Rosenwald School Col...	i-00a24a3716c5b974a	Stopped	t2.micro	-	View alarms +	us-east-1
<input type="checkbox"/> I.D. - Rosenwald Family Conn...	i-0178247b7c5b974a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1

Instance: i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)

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

▼ Instance summary info

Instance ID i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)	Public IPv4 address 54.85.34.203 <a href="#">open address</a>	Private IPv4 addresses 172.31.44.203
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-85-34-203.compute-1.amazonaws.com <a href="#">open address</a>
Hostname type IP name: ip-172-31-44-203.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-203.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

After doing that select on “Instance state” button and select “Reboot instance”.

Instances (1/13) info

Find Instance by attribute or tag (case-sensitive)

All states

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/> Aayush - My New Documenting Instance	i-0a6c7039ace43e913	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1
<input type="checkbox"/> Aayush - My OmekaS Rosenwald School	i-0cbba6876c39a242b	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1
<input type="checkbox"/> Aayush (Terminate?) - Rosenwald School Col...	i-00a24a3716c5b974a	Stopped	t2.micro	-	View alarms +	us-east-1
<input type="checkbox"/> I.D. - Rosenwald Family Conn...	i-0178247b7c5b974a	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1

Instance: i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)

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

▼ Instance summary info

Instance ID i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)	Public IPv4 address 54.85.34.203 <a href="#">open address</a>	Private IPv4 addresses 172.31.44.203
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-54-85-34-203.compute-1.amazonaws.com <a href="#">open address</a>
Hostname type IP name: ip-172-31-44-203.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-44-203.ec2.internal	Elastic IP addresses -
Answer private resource DNS name IPv4 (A)	Instance type t2.micro	

Stop instance

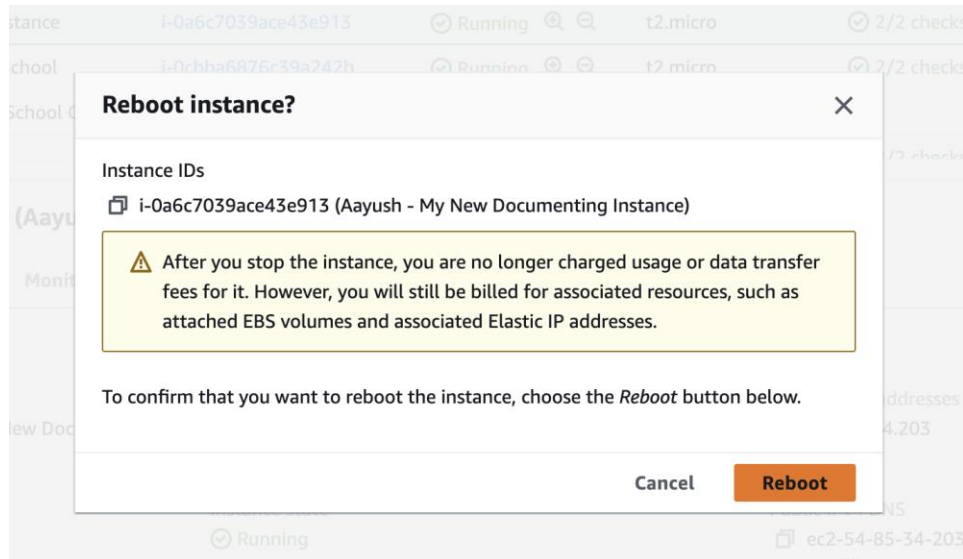
Start instance

Reboot instance

Hibernate instance

Terminate instance

Then select “Reboot”.



Then come back to the terminal and select up arrow or copy and paste the code from step 15 from AWS again and hit enter / return.

```
aayushbhatta@administrators-MacBook-Pro .ssh % ssh -i "testing_key_pairs.pem" ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1015-aws x86_64)
```

After that run both the commands on ubuntu again that you ran at the beginning of step 17 and step 18.

```
ubuntu@ip-172-31-44-203:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
ubuntu@ip-172-31-44-203:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  ubuntu-advantage-tools ubuntu-pro-client-l10n
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

## 19. Installing Server Packages

After the server is up and running, we need to get the components that are needed to run a web server installed. I'll use short-hand here to install a bunch of packages (and their dependencies). Then tell the Apache daemon to enable the `mod_rewrite` module that Omeka-S uses to make "pretty" URLs.

*\*Note: When logging onto the AWS server, you may be put into the `ubuntu` directory. You need to get to the main directory. Change the directory and verify by listing the files in that directory.*

```
cd /
```

```
ls
```

```
ubuntu@ip-172-31-44-203:~$ cd /
ubuntu@ip-172-31-44-203:/$ ls
bin  dev  home  lib32  libx32  media  opt  root  sbin  srv  tmp  var
boot  etc  lib  lib64  lost+found  mnt  proc  run  snap  sys  usr
```

```
sudo apt-get -y install apache2 php php-xsl php-mysql php-curl php-mbstring wget mysql-server zip imagemagick
sendmail
```

```
sudo a2enmod rewrite
```

As you can see in the terminal, to activate the new configuration, you need to run:

```
sudo systemctl restart apache2
```


Confirm the web server is running by visiting your Public DNS in your browser. Your Public DNS can be found if you return to the "Connect to instance" browser window we left open in step 11. There you will find your Public DNS, paste it into your browser. You should see a page in the browser that says, "It Works!".


## Connect to instance [Info](#)

Connect to your instance i-0a6c7039ace43e913 (Aayush - My New Documenting Instance) using any of these options

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

Instance ID


 i-0a6c7039ace43e913 (Aayush - My New Documenting Instance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is testing\_key\_pairs.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.  
 `chmod 400 "testing_key_pairs.pem"`
4. Connect to your instance using its Public DNS:

 ec2-54-85-34-203.compute-1.amazonaws.com

Example:

 `ssh -i "testing_key_pairs.pem" ubuntu@ec2-54-85-34-203.compute-1.amazonaws.com`

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

Cancel



## Configure MySQL

20. Start by logging in to MySQL. By default, the 'sudo' command authenticates your Ubuntu MySQL installation, so a password is not needed. When prompted to enter a password, simply hit return/enter. If not, you don't need to enter any password.

```
sudo mysql -u root -p:
```

```
ubuntu@ip-172-31-44-203:/$ sudo mysql -u root -p:
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

```
ubuntu@ip-172-31-44-203:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.36-0ubuntu0.22.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Now let's create a database for Omeka-S, set a password for a newly created user, and give that user permission to the new database.

Replace “YourDatabaseName” with the database name of your choosing, “new\_user” with the username of your choosing and “YourNewPassword” with the password of your choosing. Be sure to store the new username and password in a secure location for use in a couple steps.

```
CREATE DATABASE YourDatabaseName;
CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'YourNewPassword';

GRANT ALL PRIVILEGES on YourDatabaseName.* to 'new_user'@localhost;

FLUSH PRIVILEGES;

Exit;
```



```
mysql> CREATE DATABASE for_documentation;
Query OK, 1 row affected (0.01 sec)

mysql> CREATE USER 'rosenwald_fund'@'localhost' IDENTIFIED BY ' ';
Query OK, 0 rows affected (0.03 sec)

mysql> GRANT ALL PRIVILEGES on for_documentation.* to 'rosenwald_fund'@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

Exit MYSQL by writing “exit” and hit enter / return.

```
mysql> exit
Bye
```

## Download and Configure Omeka-S

21. The default location for the web applications for Apache2 is `/var/www/`. For the purposes of this tutorial, we'll download the Omeka-S application then mv the files to `/var/www/`.

Assuming you're still logged on to your server, you will need to issue the following commands to download Omeka-S:

```
cd /tmp
git clone https://github.com/omeka/omeka-s.git
sudo mv omeka-s /var/www/html
sudo chmod -R 777 /var/www/html/omeka-s/files
sudo service apache2 restart
```

Now that the database is set up, we need to let Omeka-S know where to go to connect to the database.

22. You will now need the database name, username and password you set in MySQL a couple steps ago. Let's open the Omeka-S `database.ini` file in a terminal text editor:

```
cd /var/www/html/omeka-s/config
```

```
sudo nano database.ini
```

If database.ini is empty for you then feel free to use this:

```
user = ""
password = ""
dbname = ""
host = ""
:port =
:unix_socket =
:log_path =
```

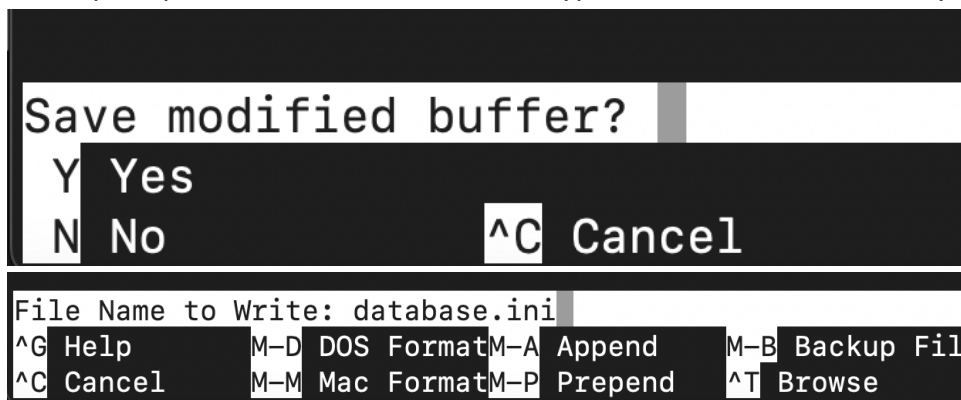
Now, using the arrow keys to navigate, populate the MySQL user, password and dbname you chose. For host, type "localhost".



```
.ssh — ubuntu@ip-172-31-44-203: /var/www/html/omeka-s/config
GNU nano 6.2 database.ini *
user = "rosenwald_fund"
password = ""
dbname = "for_documentation"
host = "localhost"
:port =
:unix_socket =
:log_path =
```

Exit nano by holding the 'control' and 'X' keys simutaniously.

When prompted "Save modified buffer?" type "Y" then hit the enter Key to save the filename.



```
Save modified buffer?
Y Yes
N No      ^C Cancel

File Name to Write: database.ini
^G Help      M-D DOS FormatM-A Append      M-B Backup File
^C Cancel    M-M Mac FormatM-P Prepend    ^T Browse
```

Now restart Apache

```
sudo service apache2 restart
```

23. Next, we'll set AllowOverride to 'All':

```
sudo nano /etc/apache2/apache2.conf
```

Scroll down until you see "<Directory /var/www/>", and where it says "AllowOverride None", change it to "AllowOverride All". If you have a mac, you can type "ctrl + w" and type "/var/www/" so you will find the specific place fast otherwise you can use keys to move around and find the specific place.

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Then, hit "ctrl + x" and type "Y" and hit enter / return.

Now restart Apache

```
sudo service apache2 restart
```

24. In your browser navigate to <http://yourPublicDNS/omeka-s> and congratulations, you should be ready to install.