

AWS Deployment Serverful Architecture - EC2 Instance

EC006 - AWS Cloud Services and Infrastructure

Activity

AWS Deployment Serverful Architecture - EC2 Instance

Activity Output

```
ubuntu@ip-172-31-16-215: ~
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-16-215 :~$
```

Activity: Instructions

- Review the topics discussed in the previous session.
- Answer the quiz form about the AWS Deployment Serverful Architecture - EC2 Instance

Deploying a MERN Stack App to Amazon EC2

AWS Deployment Serverful Architecture - EC2 Instance

Deploying MERN Stack App to Amazon EC2

What is a MERN Stack App?

A **MERN Stack application** is made up of a front-end app built with **React** that connects to a **back-end api built with Node.js + Express + MongoDB**, hence the name **MERN Stack (Mongo, Express, React, Node)**.

Other variations of the stack include the **MEAN Stack that has an Angular front-end**, and the **MEVN Stack that has a Vue.js front-end**.

Deploying MERN Stack App to Amazon EC2

For this session, we will focus on setting up the server on AWS EC2, then deploying and configuring the front-end and back-end pieces of the MERN stack app to work together.

Amazon Elastic Cloud Compute (EC2)

AWS Deployment Serverful Architecture - EC2 Instance

Amazon Elastic Cloud Compute (EC2)

Amazon EC2 (Elastic Compute Cloud)

EC2 is a virtual server in the cloud that allows users to run applications on the AWS infrastructure. With EC2, users can choose from a variety of instance types and sizes, depending on the requirements of the application.

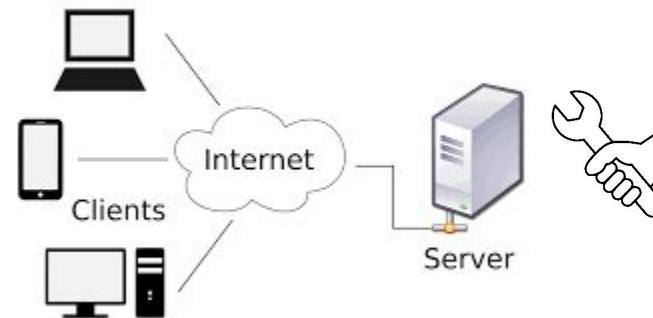


Review on Serverful Architecture

AWS Deployment Serverful Architecture - EC2 Instance

Review on Serverful Architecture

Traditionally, when you **build a software application**, you need to have a **server** to host the application. This is known as a "**serverful architecture**".



With serverful architecture, you would have to manage the server yourself, including its maintenance, scaling, and security.

Review on Serverful Architecture

For this session, we will be utilizing EC2 as the physical server and which will contain several applications to simulate a serverful architecture.

What is FileZilla?

EC2 Instance

What is FileZilla?

The **FileZilla** software program is a **free-to-use (open source) FTP utility**, **allowing a user to transfer files** from a **local computer to a remote computer**.



Source: <https://www.computerhope.com/jargon/f/filezilla.htm>

What is FileZilla?

We will be utilizing FileZilla to transfer files from our device to our EC2 instance.

Code Discussion

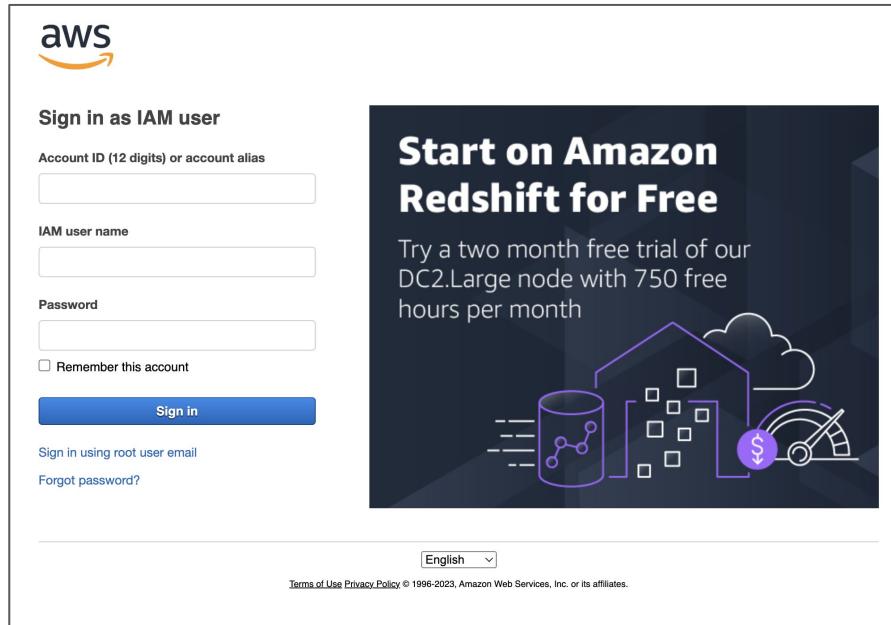
AWS Deployment Serverful Architecture - EC2 Instance

Create a new Ubuntu server on AWS EC2

AWS Deployment Serverful Architecture - EC2 Instance

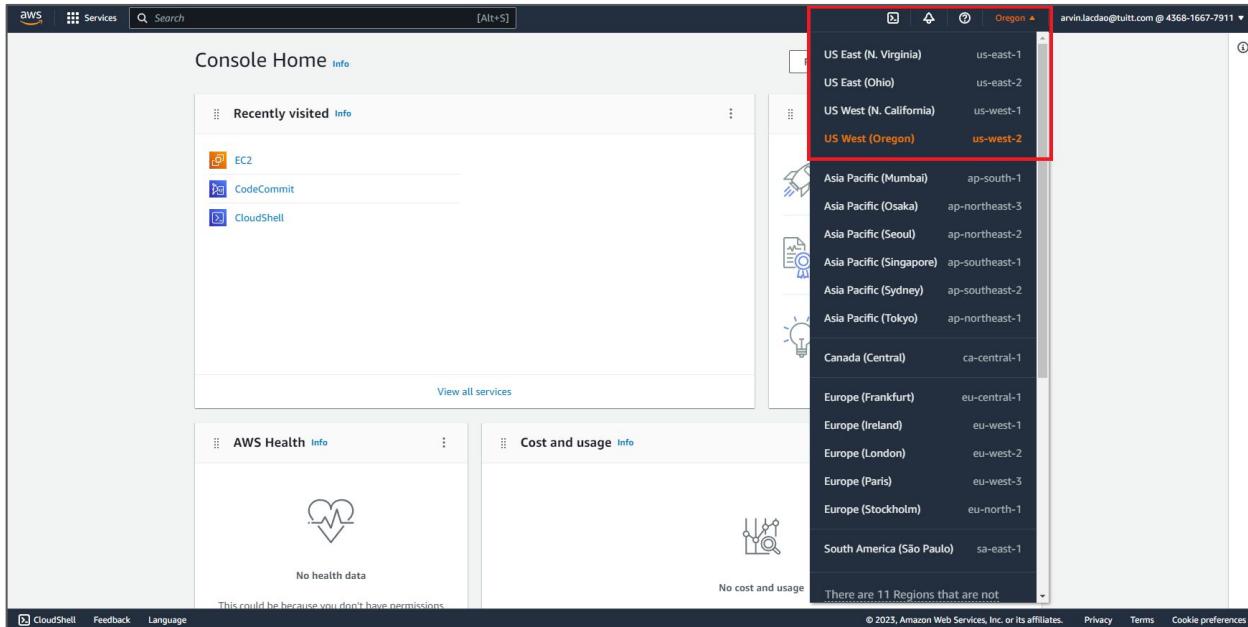
Create a new Ubuntu server on AWS EC2

1. **Sign in** to the AWS Management Console.



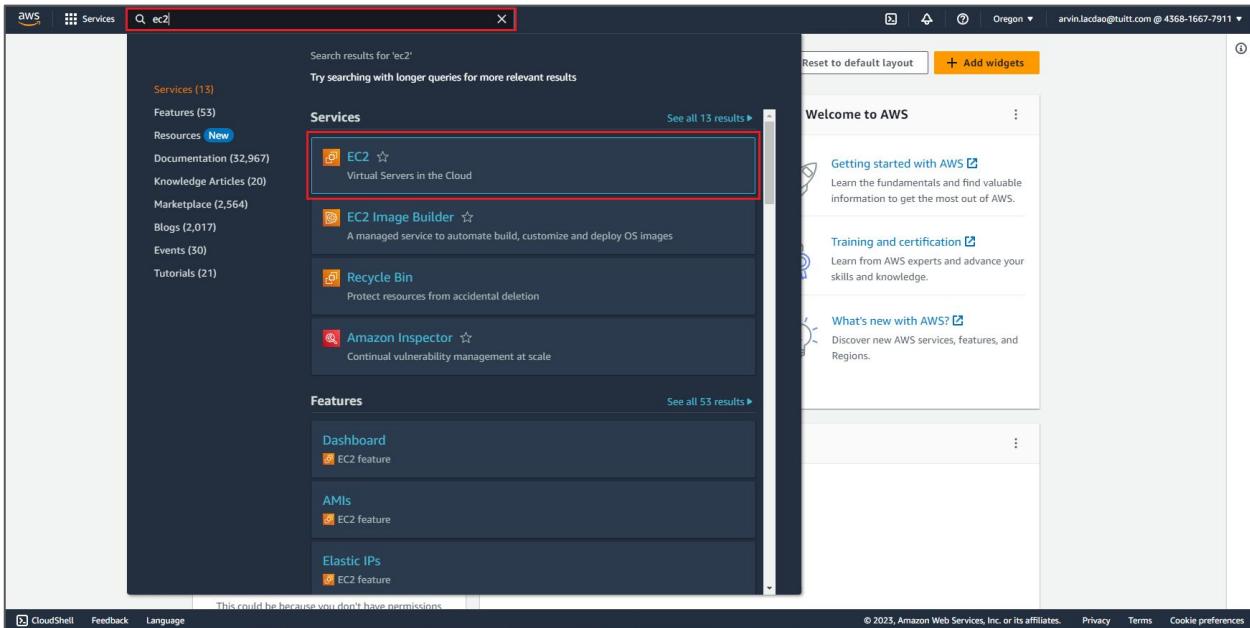
Create a new Ubuntu server on AWS EC2

2. **Check region settings** and must be set to **US West (Oregon)**.



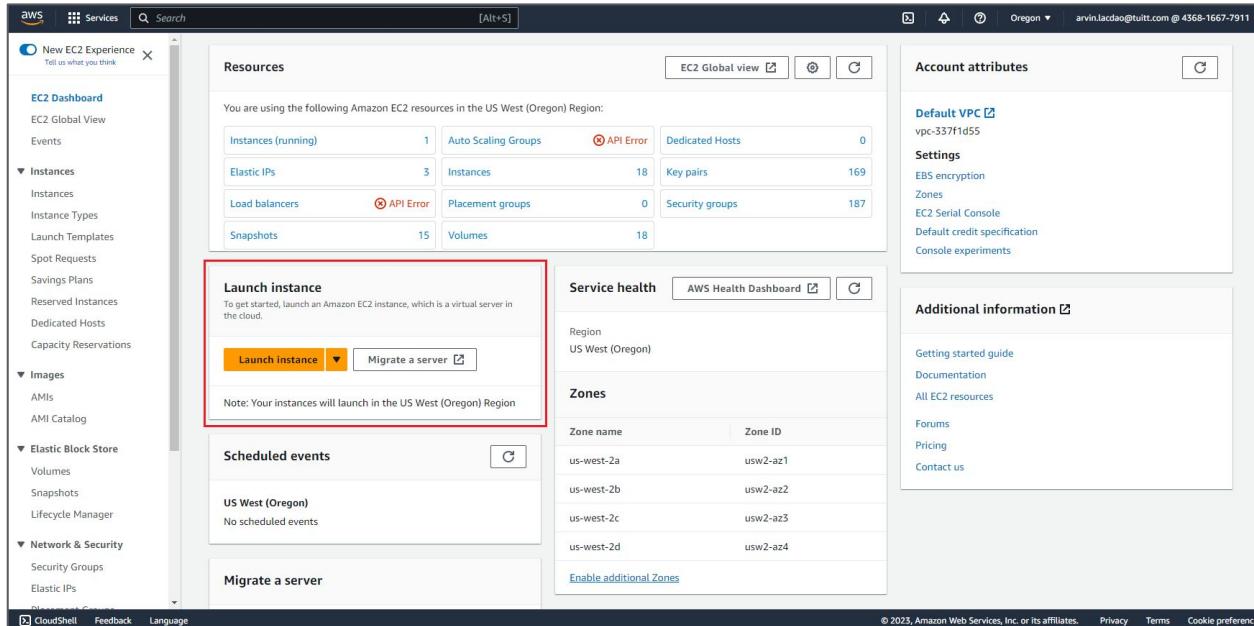
Create a new Ubuntu server on AWS EC2

3. Go to the **EC2 Service section**.



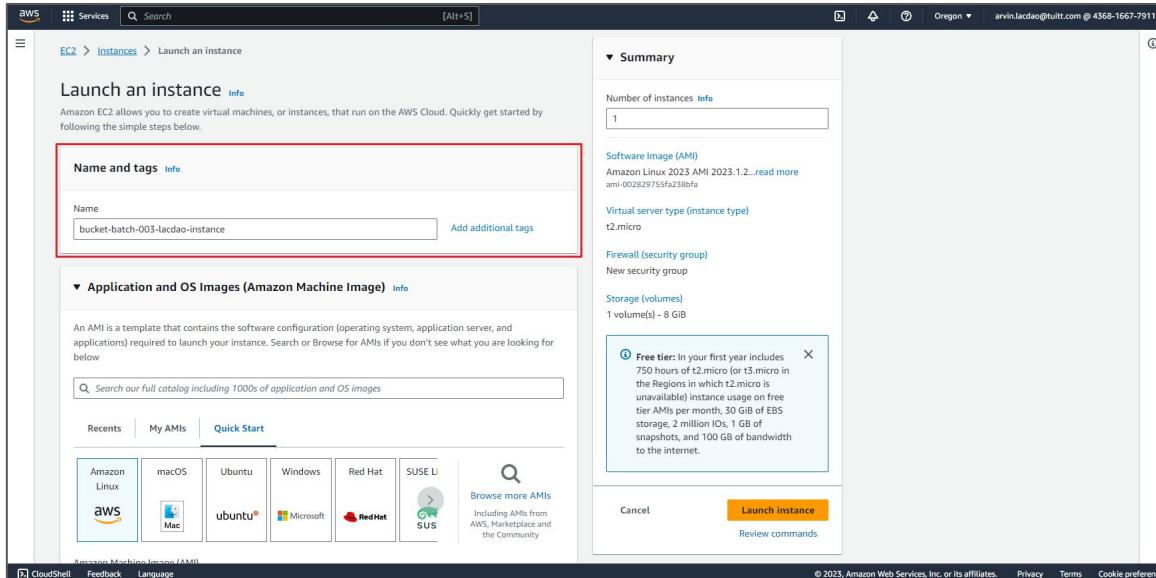
Create a new Ubuntu server on AWS EC2

4. Click the **Launch Instance** button.



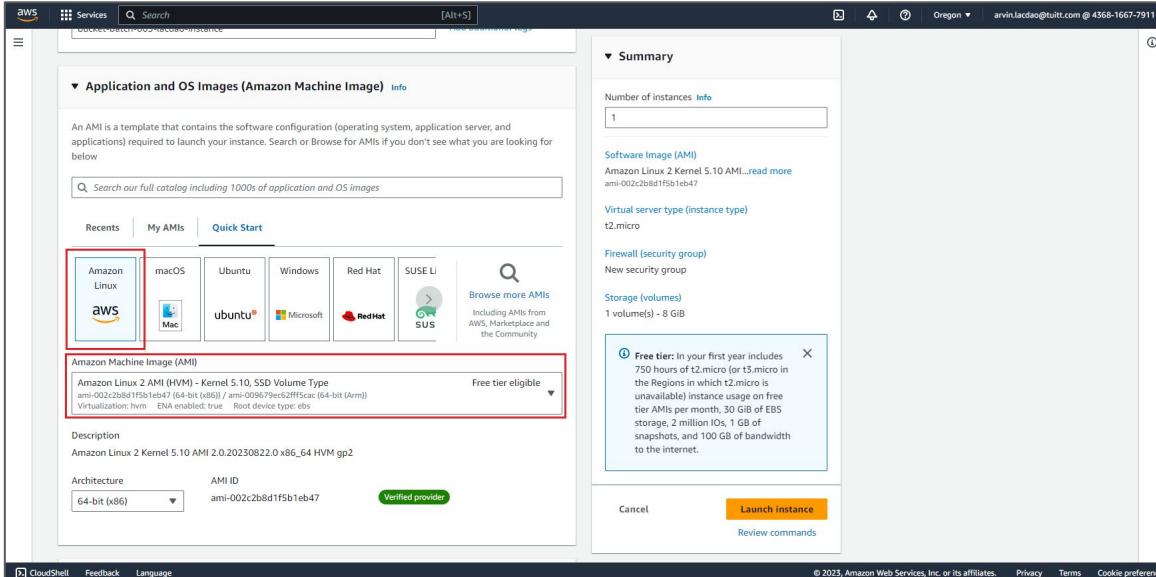
Create a new Ubuntu server on AWS EC2

5. **Add a name to the instance.** For the instance name, use the naming convention **bucket-batch-SID-lastname-instance**.



Create a new Ubuntu server on AWS EC2

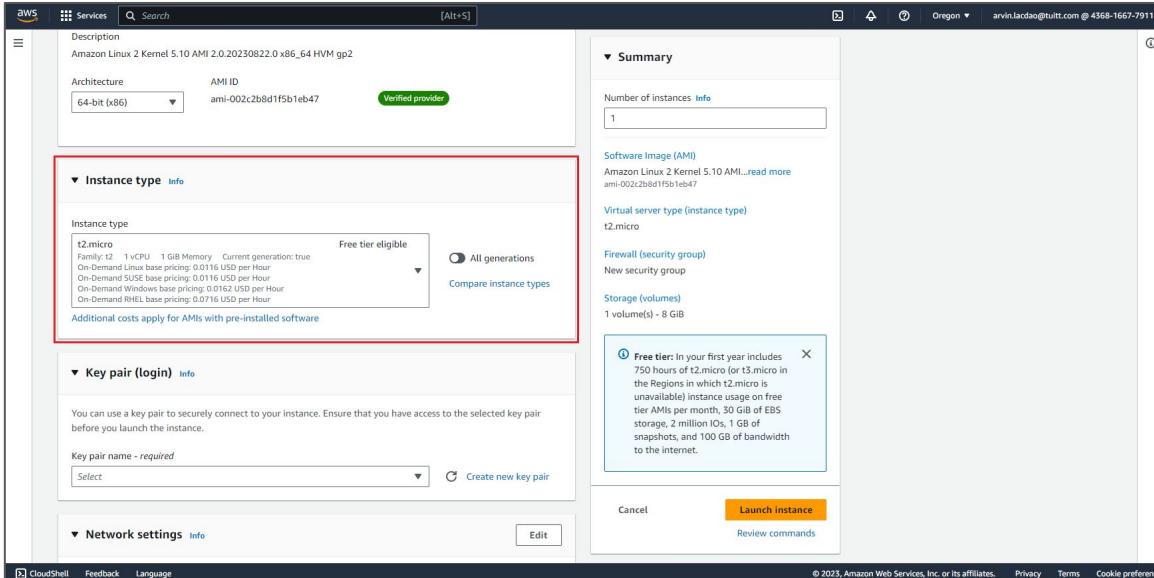
6. Choose AMI - **Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type** and **make sure Free tier eligible is selected**.



The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with 'Services' and a search bar. The main area has a heading 'Application and OS Images (Amazon Machine Image)'. Below it, a sub-section titled 'Amazon Machine Image (AMI)' is highlighted with a red box. It lists 'Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type' as the selected AMI. To the right of this section, a 'Free tier eligible' badge is also highlighted with a red box. The right side of the screen displays a 'Summary' panel with details like 'Number of instances: 1', 'Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...', 'Virtual server type (instance type): t2.micro', and a note about the 'Free tier' which includes 750 hours of usage per month. At the bottom right of the summary panel is a large orange 'Launch instance' button.

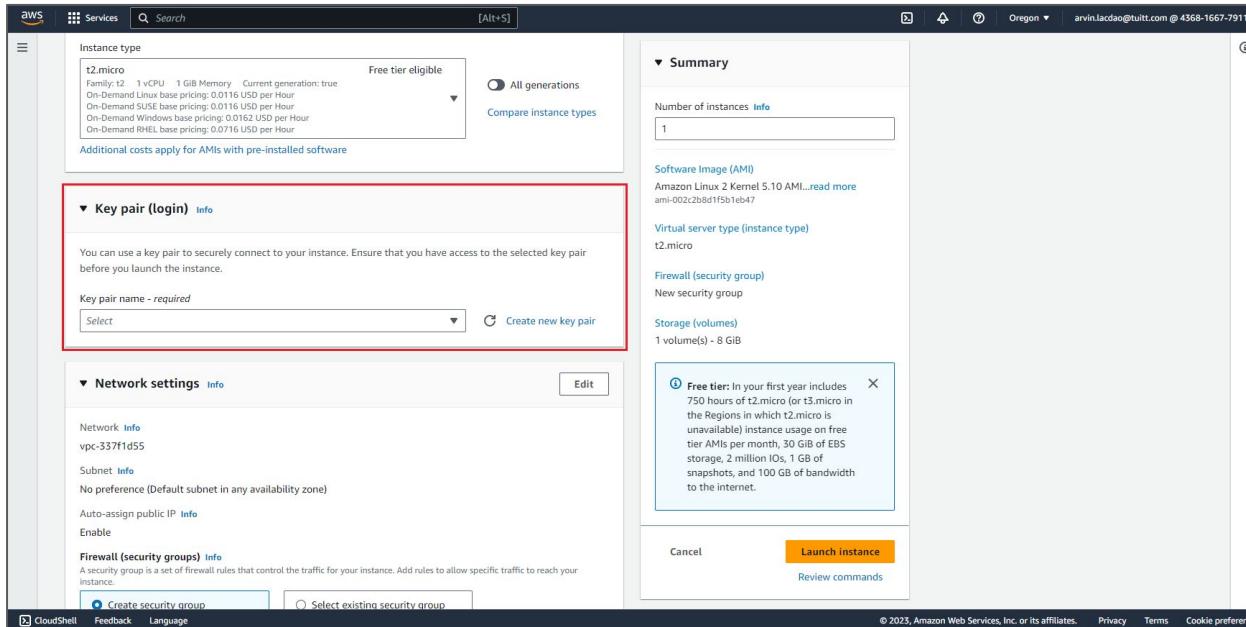
Create a new Ubuntu server on AWS EC2

7. Choose Instance Type - Select the "t2.micro" (Free tier eligible) instance type



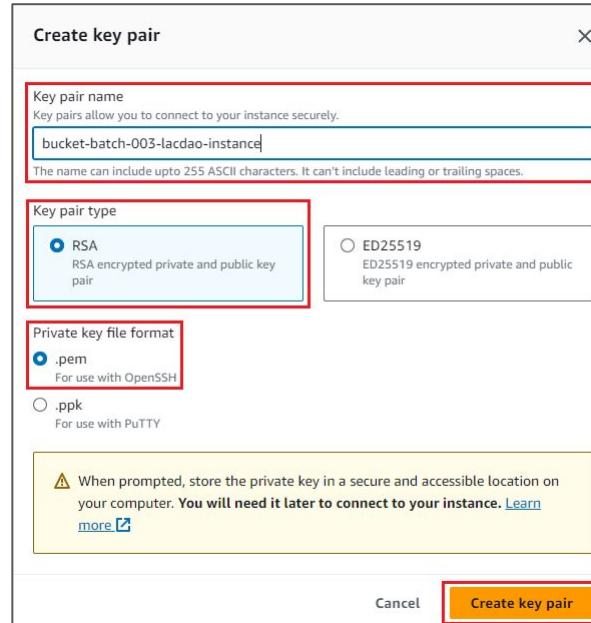
Create a new Ubuntu server on AWS EC2

8. Key pair (login) - **Create new key pair**



Create a new Ubuntu server on AWS EC2

9. Add the key pair name and click on the **Create key pair button**.



Create a new Ubuntu server on AWS EC2

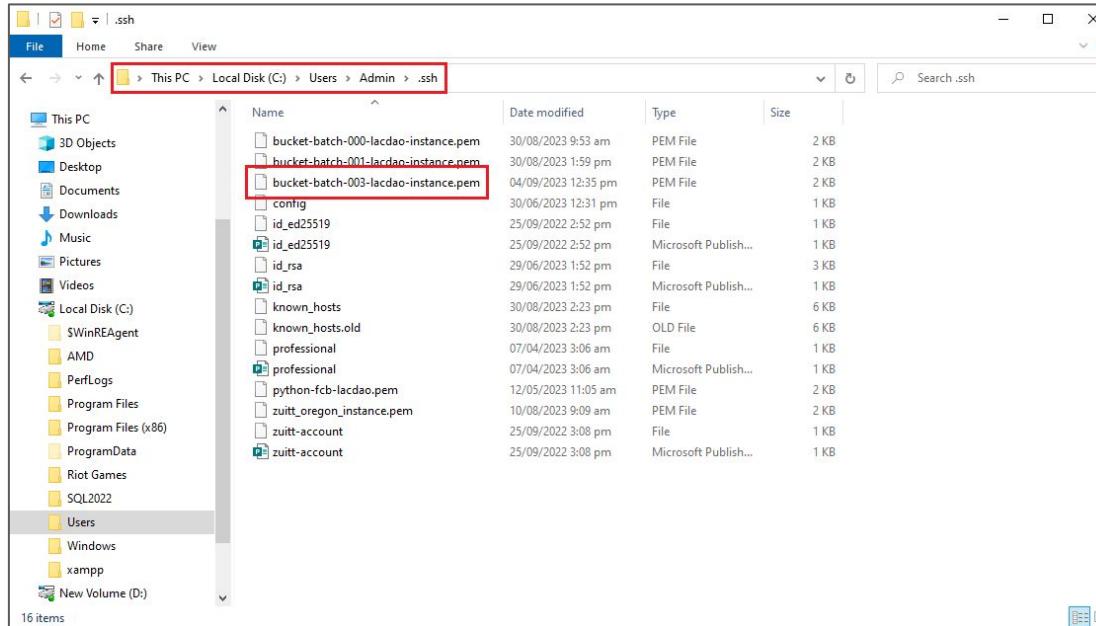
Important Note

A file will be automatically downloaded with the same name as the created key pair after clicking the Create key pair button.

You can check the browser's download history or check your device's default download folder to find the pem (Privacy Enhanced Mail) file.

Create a new Ubuntu server on AWS EC2

10. Transfer the .pem file to your device's .ssh folder.



Create a new Ubuntu server on AWS EC2

Important Note

The SSH folder is located in different folders depending on the OS (Operating System).

Windows:

C:\Users\<Username>\.ssh

MacOS:

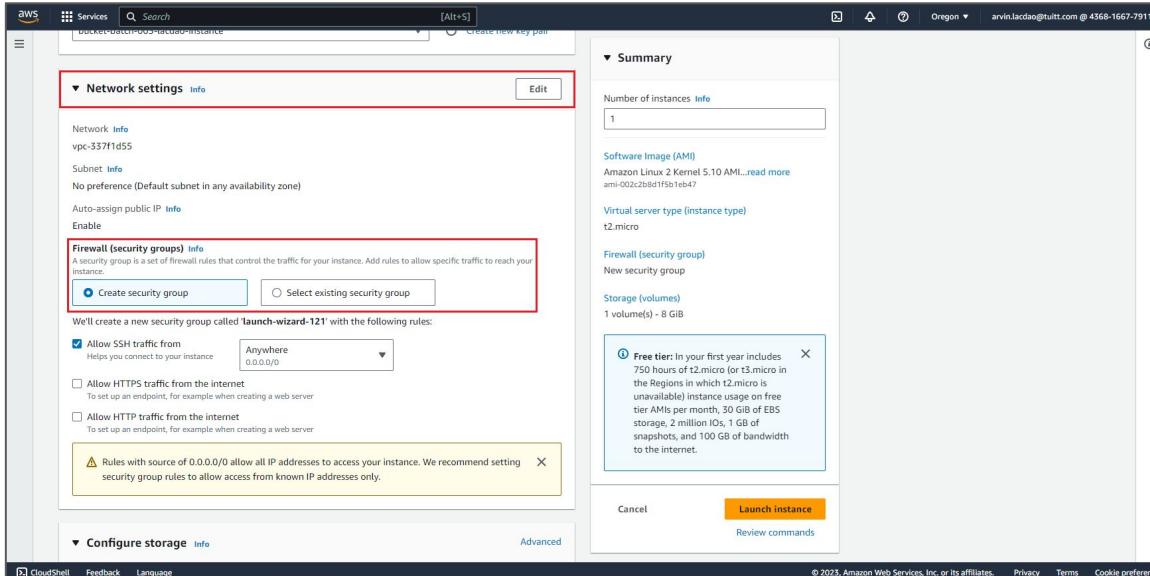
Open Finder, press command + shift + g on the keyboard then type "~/.ssh"

Linux:

/home/<Username>/.ssh

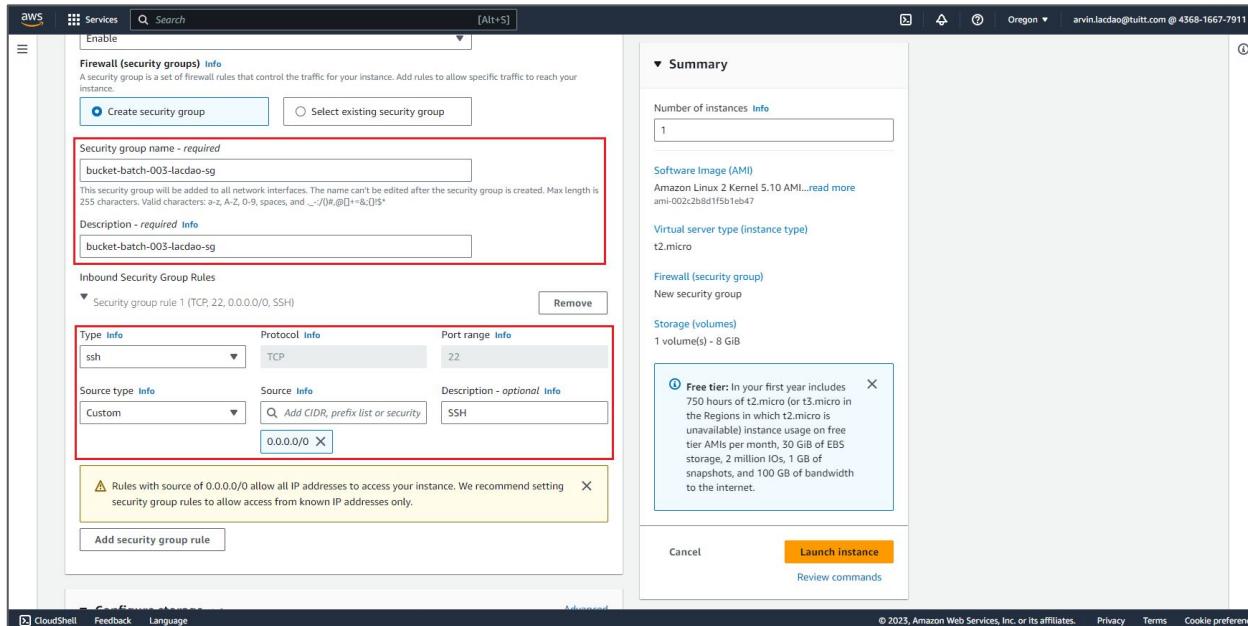
Create a new Ubuntu server on AWS EC2

11. Configure Security Group by clicking **Edit** button at **Network settings** section



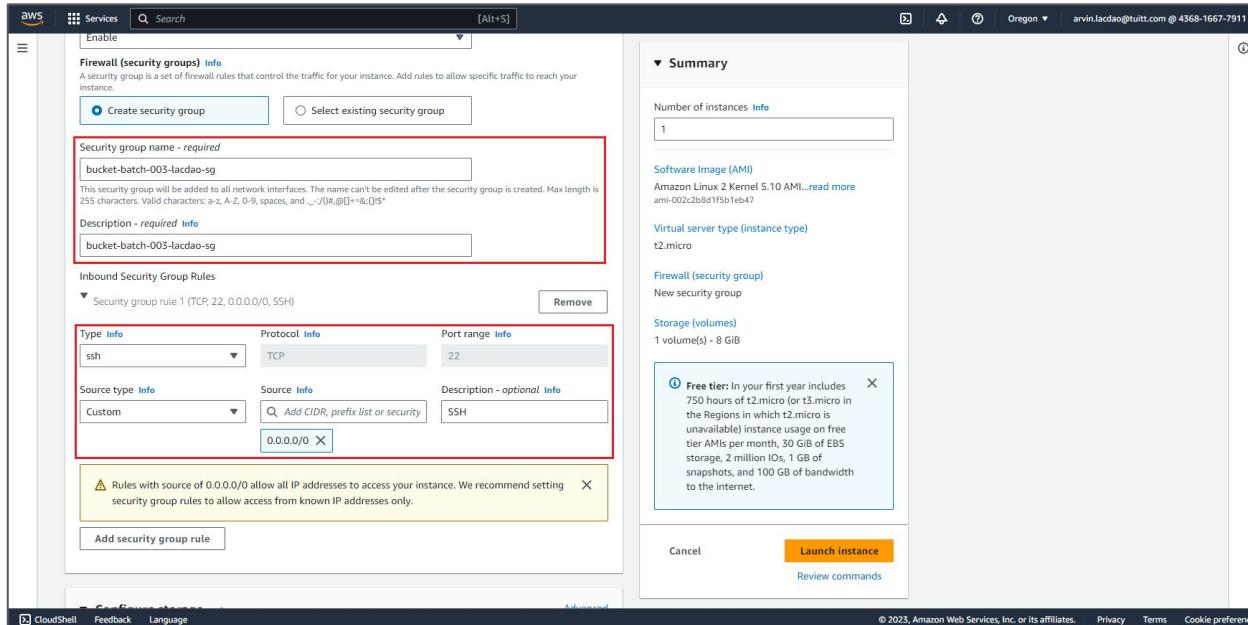
Create a new Ubuntu server on AWS EC2

12. Configure Security Group - Rule 1



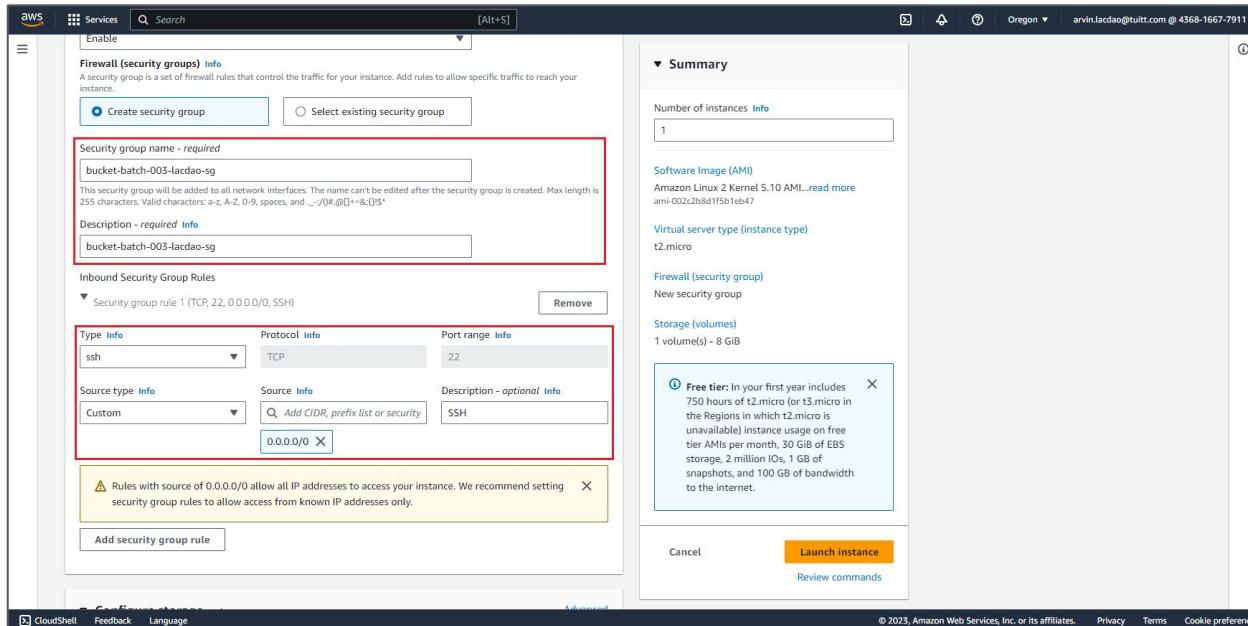
Create a new Ubuntu server on AWS EC2

13. Add security group rule by clicking the button.



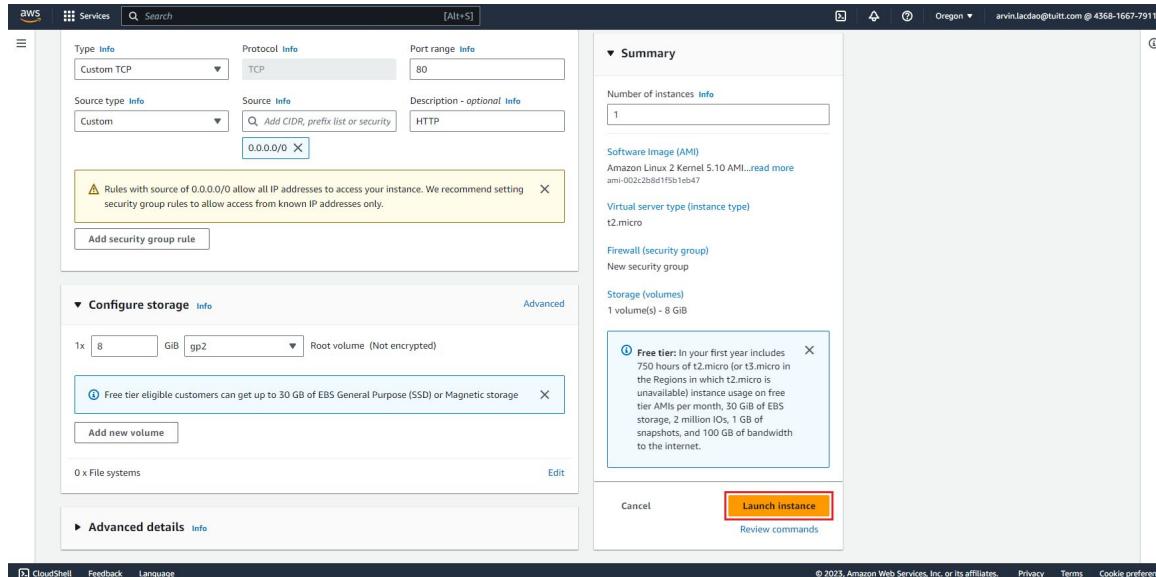
Create a new Ubuntu server on AWS EC2

14. Configure Security Group - Rule 2



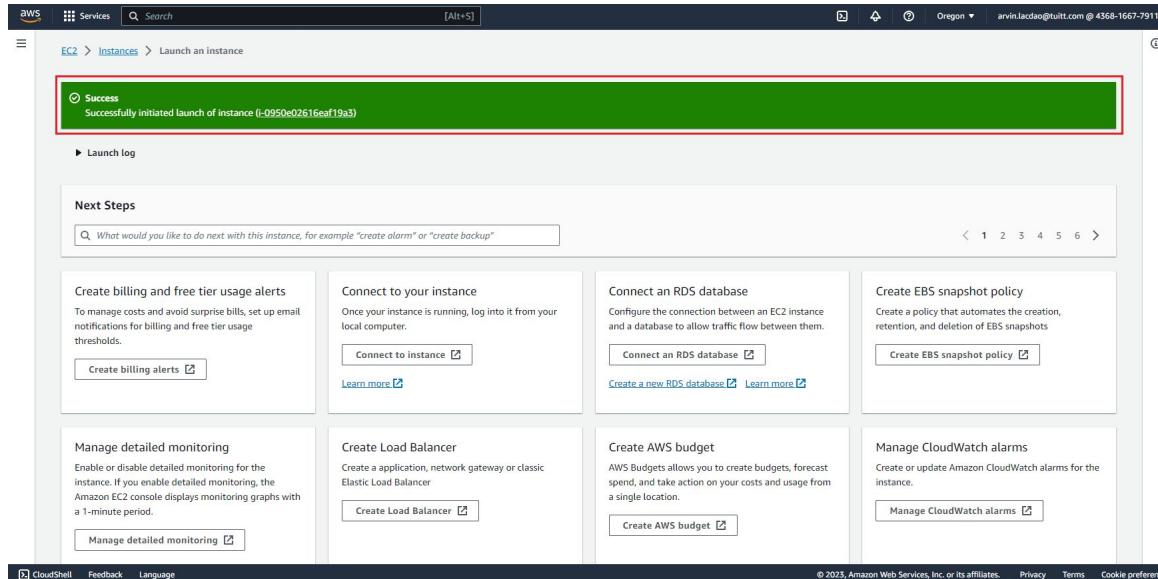
Create a new Ubuntu server on AWS EC2

15. Click **Launch Instance**. A **success message** will be shown after indicating the **creation of the instance**.



Create a new Ubuntu server on AWS EC2

15. Click **Launch Instance**. A **success message will be shown after** indicating the **creation of the instance**.

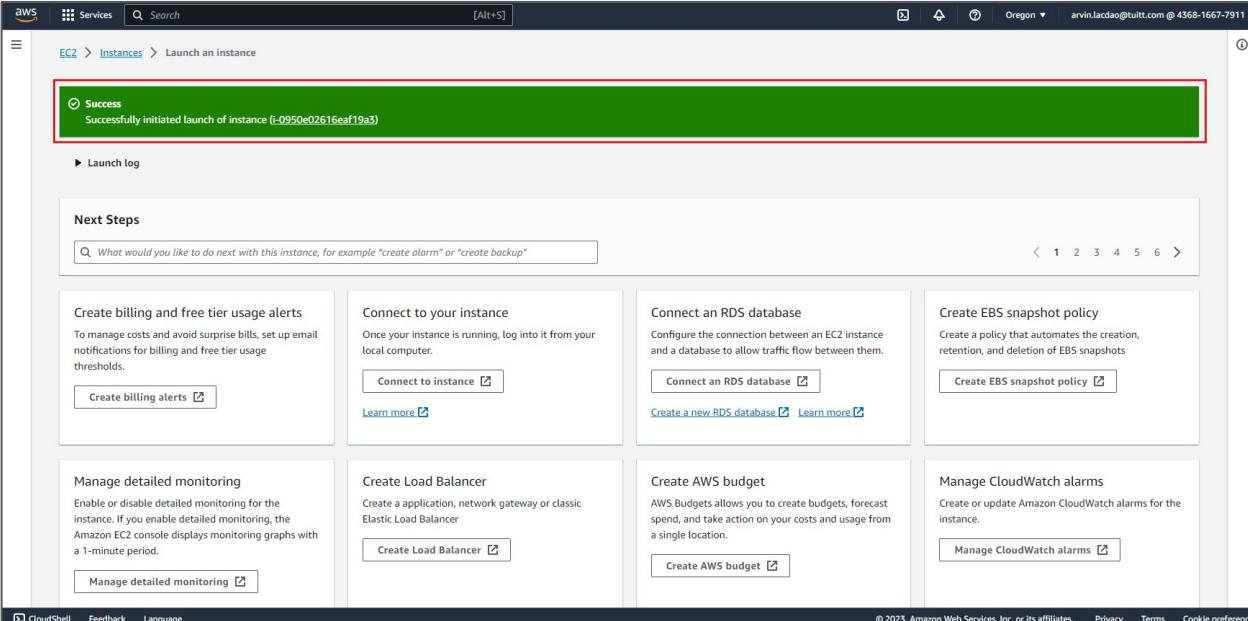


Connect to EC2 Instance via SSH

AWS Deployment Serverful Architecture - EC2 Instance

Connect to EC2 Instance via SSH

1. **Check the status of the instance** if it is currently **running**.



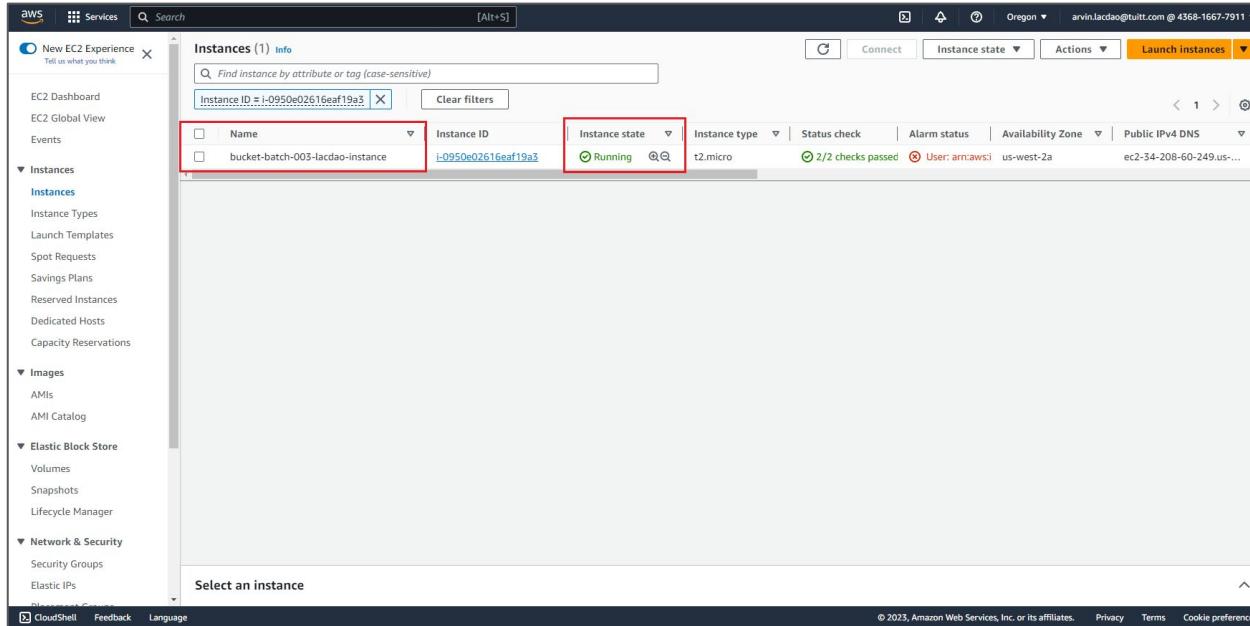
The screenshot shows the AWS EC2 Instances launch page. At the top, there is a green success message box with a red border containing the text: "Success" and "Successfully initiated launch of instance (i-0950e02616ef19a3)". Below this, there is a "Next Steps" section with several options:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy
- Manage detailed monitoring
- Create Load Balancer
- Create AWS budget
- Manage CloudWatch alarms

At the bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information.

Connect to EC2 Instance via SSH

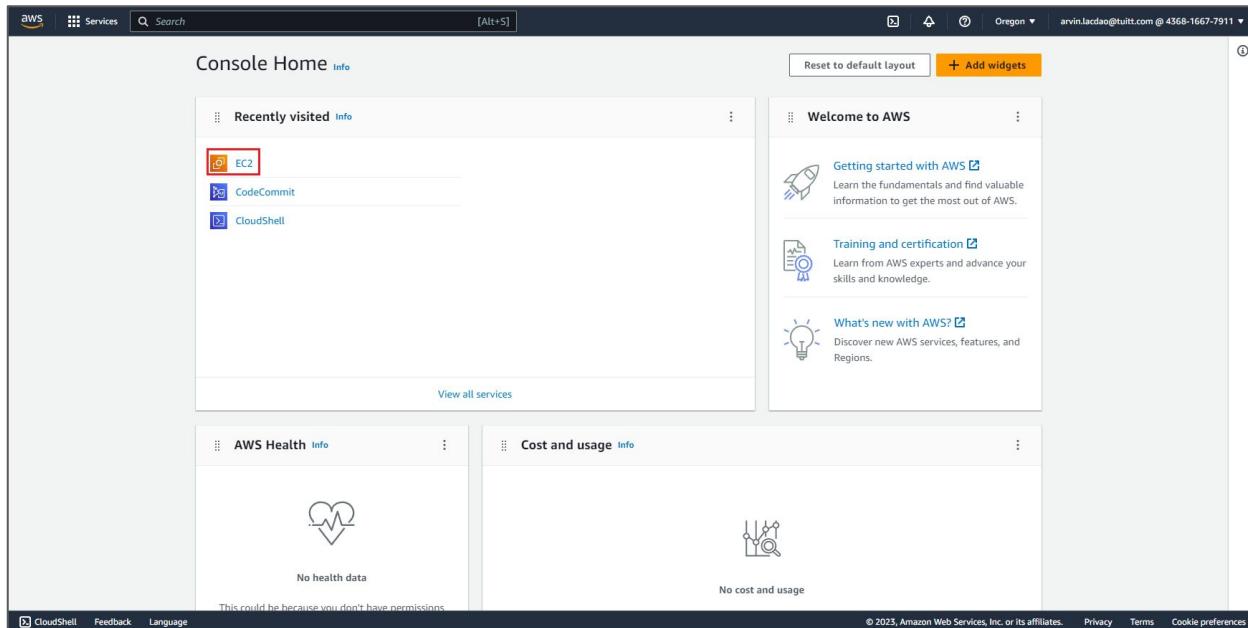
1. **Check the status of the instance** if it is currently **running**.



The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs), and CloudWatch Metrics (CloudWatch Metrics Home, Metrics Insights). The main content area has a header "Instances (1) Info" with a search bar and filter buttons. Below the header, there's a table with one row. The first column contains checkboxes for "Name" and "bucket-batch-003-lacdao-instance". The second column shows the "Instance ID" as i-0950e02616eaf19a3. The third column shows the "Instance state" as "Running" with a green checkmark icon. The fourth column shows the "Instance type" as t2.micro. The fifth column shows the "Status check" as "2/2 checks passed". The sixth column shows the "Alarm status" as "User:aws:ec2-34-208-60-249.us...". The seventh column shows the "Availability Zone" as us-west-2a. The eighth column shows the "Public IPv4 DNS" as ec2-34-208-60-249.us... At the bottom of the table, there's a button labeled "Select an instance". At the very bottom of the page, there are links for CloudShell, Feedback, Language, and a footer with copyright information: © 2023, Amazon Web Services, Inc. or its affiliates.

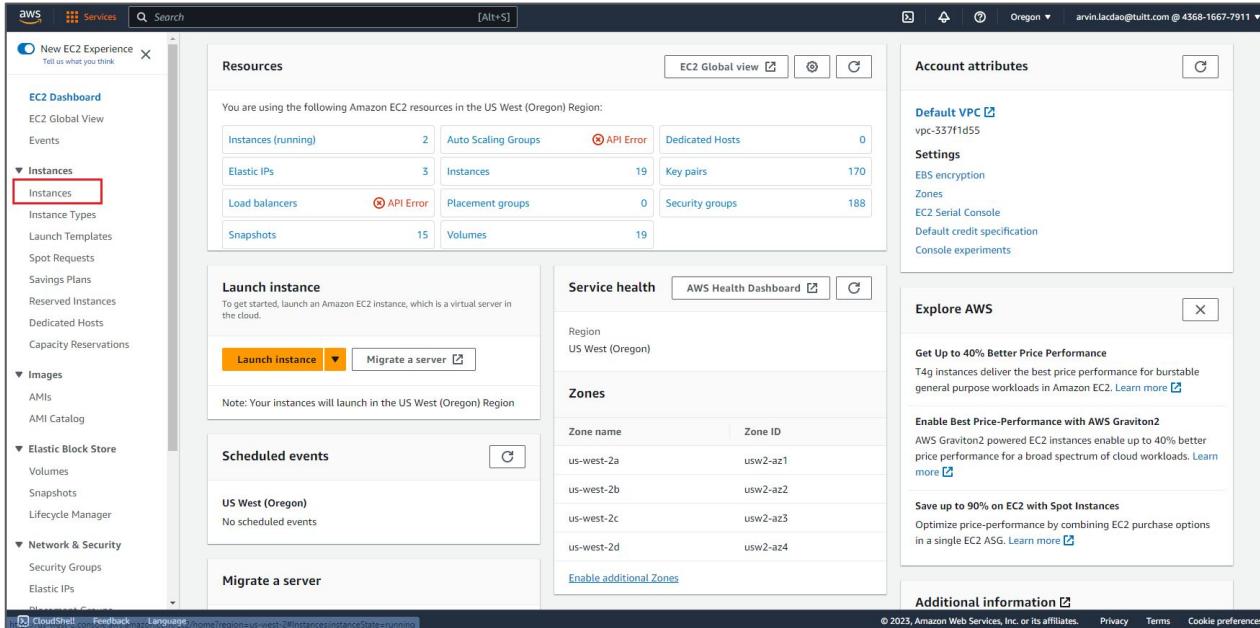
Connect to EC2 Instance via SSH

2. Alternatively, you can **check the instance status via the instance list**.



Connect to EC2 Instance via SSH

2. Alternatively, you can **check the instance status via the instance list**.



The screenshot shows the AWS EC2 Dashboard in the US West (Oregon) Region. The left sidebar navigation includes: EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, and Firewall Rules.

The main content area displays the following metrics:

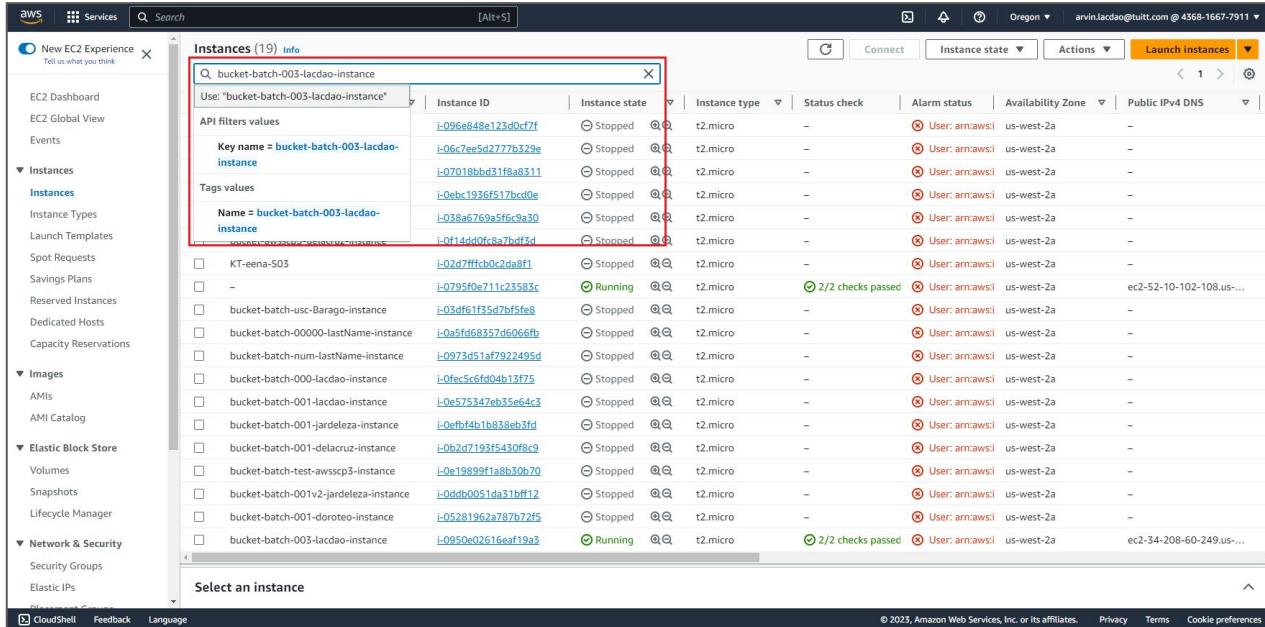
Category	Value
Instances (running)	2
Elastic IPs	3
Load balancers	0
Snapshots	15
Auto Scaling Groups	0
Instances	19
Placement groups	0
Volumes	19
Dedicated Hosts	0
Key pairs	170
Security groups	188

Below the metrics, there are sections for "Launch instance" (with "Launch instance" and "Migrate a server" buttons), "Service health" (Region: US West (Oregon)), "Scheduled events" (US West (Oregon)), and "Migrate a server".

The right sidebar shows "Account attributes" including Default VPC (vpc-337f1d55), Settings (EBS encryption, Zones, EC2 Serial Console, Default credit specification, Console experiments), and "Explore AWS" sections for T4g instances, AWS Graviton2, and Spot Instances. It also includes an "Additional information" section.

Connect to EC2 Instance via SSH

2. Alternatively, you can **check the instance status via the instance list**.

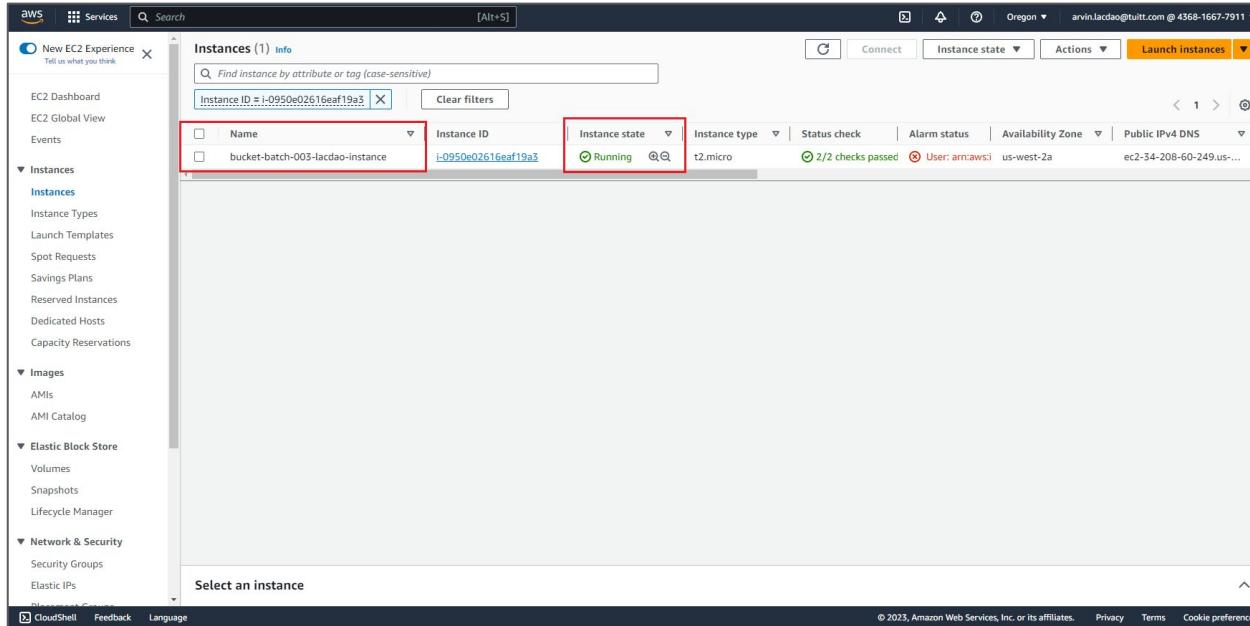


The screenshot shows the AWS EC2 Instances list with 19 instances. A search bar at the top contains the query "Use: 'bucket-batch-003-lacdao-instance'". Below the search bar, there are sections for "API filters values" and "Tags values", both of which show the key name "bucket-batch-003-lacdao-instance". The main table lists instances with columns: Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. Two instances are highlighted with a red border: one with status "Running" and another with status "Stopped". Both highlighted instances have a green "2/2 checks passed" badge.

Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
i-096e848e123d0cf7f	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
i-06c7ee5d2777b329e	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
i-d7018bb31f8a8311	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
i-0eb1c1936f517bcdc0e	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
i-038a6769a5f6c9a30	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
i-0f14d0fc8a7bfdf3d	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
KT-enea-503	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
-	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-usc-Barago-instance	Running	t2.micro	2/2 checks passed	User: armaws:1	us-west-2a	ec2-52-10-102-108.us...
bucket-batch-00000-lastName-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-num-lastName-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-000-lacdao-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-001-lacdao-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-001-jardeleza-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-001-delacruz-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-test-awsscp3-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-001v2-jardeleza-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-001-dotoreto-instance	Stopped	t2.micro	-	User: armaws:1	us-west-2a	-
bucket-batch-003-lacdao-instance	Running	t2.micro	2/2 checks passed	User: armaws:1	us-west-2a	ec2-34-208-60-249.us...

Connect to EC2 Instance via SSH

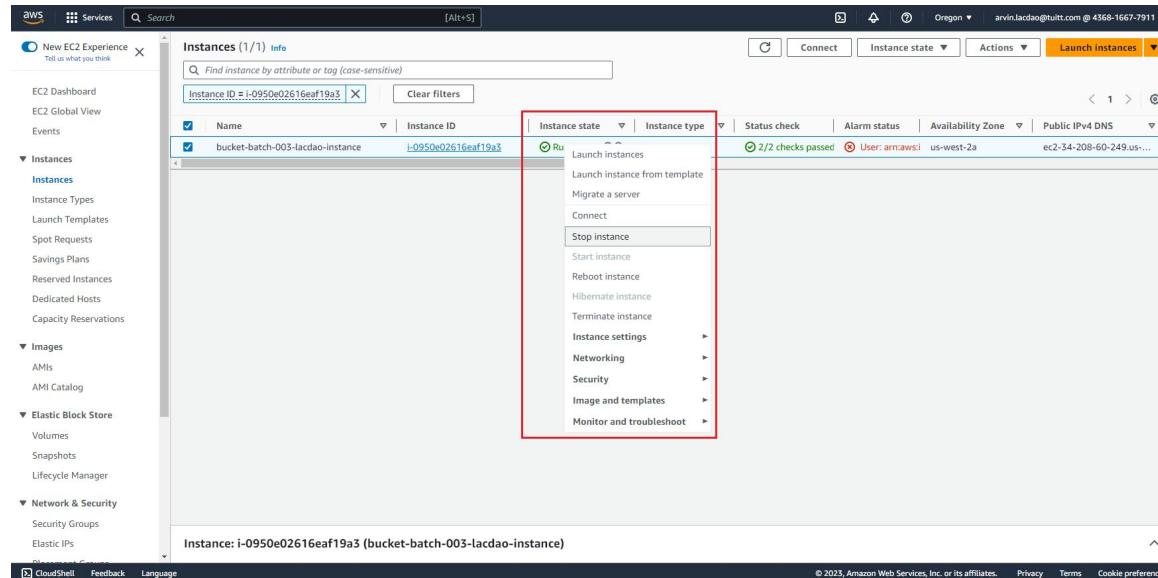
2. Alternatively, you can **check the instance status via the instance list**.



The screenshot shows the AWS EC2 Instances list. On the left, there's a navigation sidebar with links like New EC2 Experience, EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs), and CloudWatch Metrics (CloudWatch Metrics Home, Metrics Insights). The main area displays a table for 'Instances (1) Info'. The table has columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. A search bar at the top says 'Find instance by attribute or tag (case-sensitive)' with a placeholder 'Instance ID = i-0950e02616eaf19a3'. Below it are 'Clear filters' and 'Actions' buttons. The first row in the table is highlighted with a red border. It shows 'Name' as 'bucket-batch-003-lacdao-instance', 'Instance ID' as 'i-0950e02616eaf19a3', 'Instance state' as 'Running' (with a green circle icon), 'Instance type' as 't2.micro', 'Status check' as '2/2 checks passed', 'Alarm status' as 'User:aws:ec2-34-208-60-249.us...', and 'Availability Zone' as 'us-west-2a'. At the bottom of the table, there's a message 'Select an instance'.

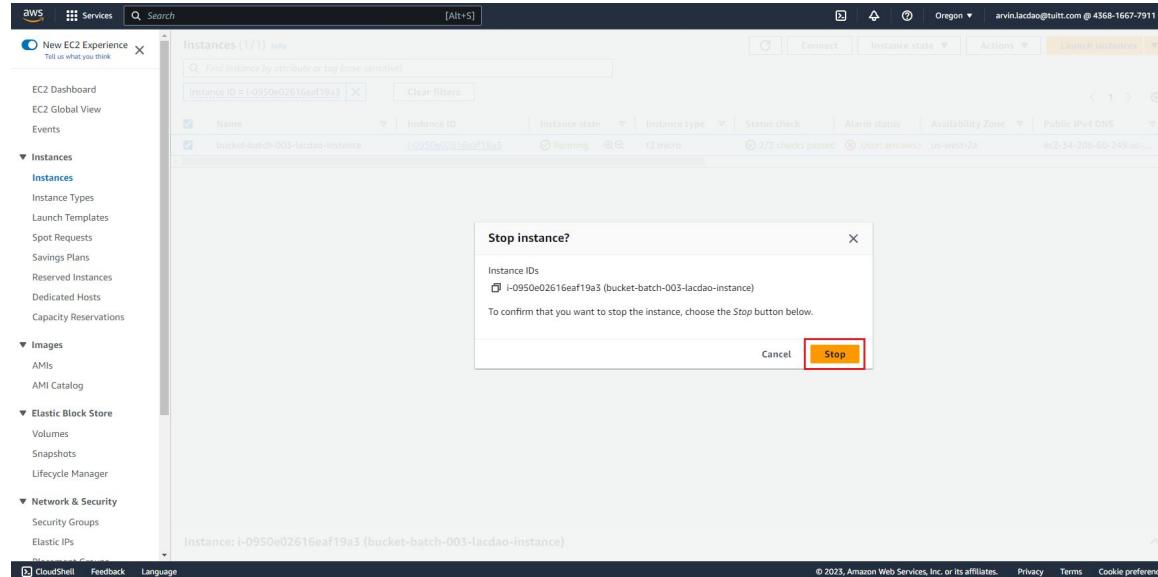
Connect to EC2 Instance via SSH

3. Stop an instance by right clicking on the status and selecting Stop Instance.



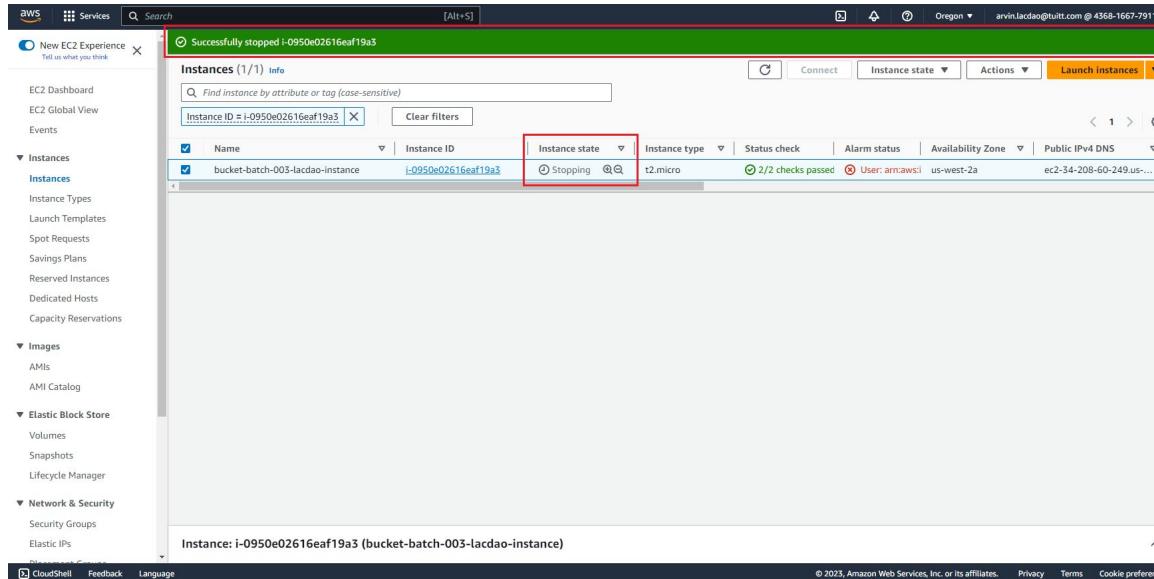
Connect to EC2 Instance via SSH

3. Stop an instance by right clicking on the status and selecting Stop Instance.



Connect to EC2 Instance via SSH

3. Stop an instance by right clicking on the status and selecting Stop Instance.



The screenshot shows the AWS EC2 Instances page. A green success message at the top says "Successfully stopped i-0950e02616eaf19a3". The main table lists one instance:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
bucket-batch-003-lacdao-instance	i-0950e02616eaf19a3	Stopping	t2.micro	2/2 checks passed	User: arvin.lacdao@tuitt.com	us-west-2a	ec2-34-208-60-249.us...

A red box highlights the "Stopping" status in the "Instance state" column. At the bottom, a modal window titled "Instance: i-0950e02616eaf19a3 (bucket-batch-003-lacdao-instance)" shows the instance's details and has a "Stop" button.

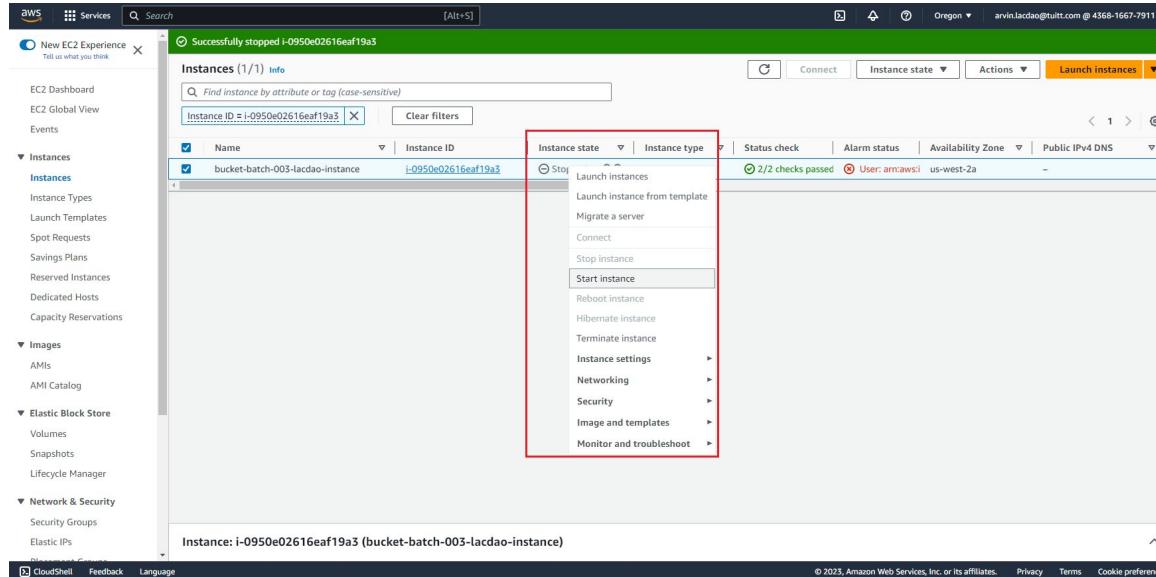
Create a new Ubuntu server on AWS EC2

Important Note

It is **good practice to stop an instance when not in use** to ensure that no charges are applied.

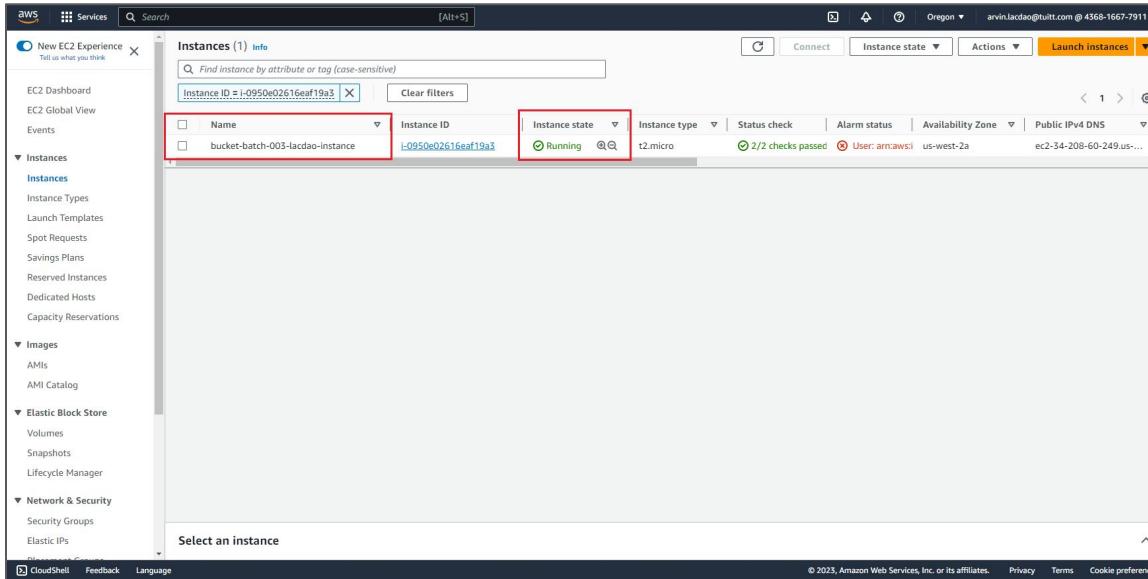
Connect to EC2 Instance via SSH

4. Start an instance by right clicking on the status and selecting Start Instance.



Connect to EC2 Instance via SSH

4. Start an instance by right clicking on the status and selecting Start Instance.



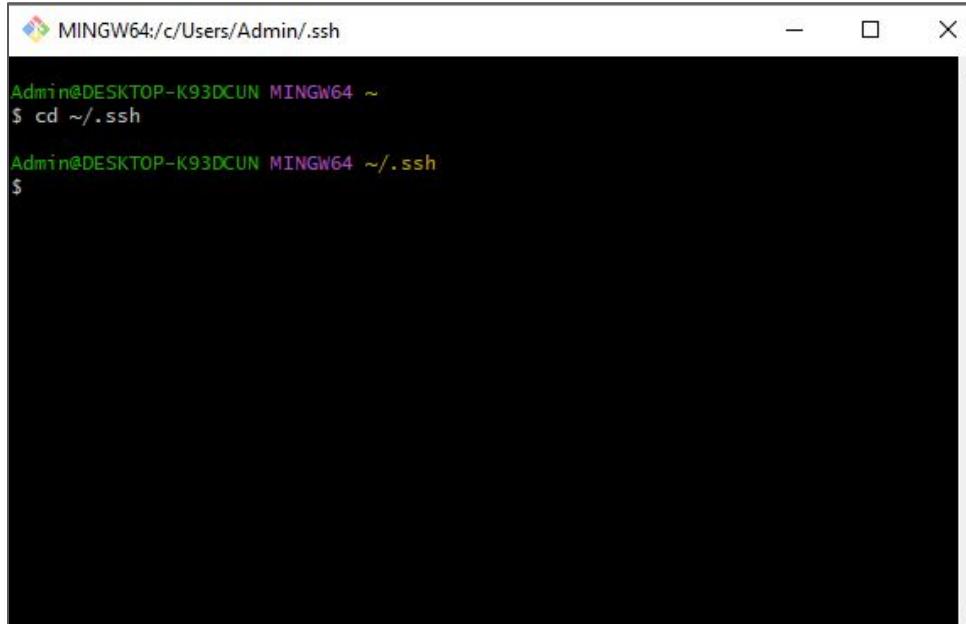
Connect to EC2 Instance via SSH

Important Note

Starting an instance might take a few moments, refreshing the page from time to time to check if the status is Running to ensure that it would be accessible for use.

Connect to EC2 Instance via SSH

-
5. Open a terminal and navigate to the **.ssh folder**.



```
MINGW64:/c/Users/Admin/.ssh
-
Admin@DESKTOP-K93DCUN MINGW64 ~
$ cd ~/.ssh

Admin@DESKTOP-K93DCUN MINGW64 ~/.ssh
$
```

Connect to EC2 Instance via SSH

6. Connect to the EC2 Instance.

Trigger the command below:

```
ssh -i "<keypair-name.pem>" ec2-user@<port> -o "ServerAliveInterval 120"
```

```
ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o  
"ServerAliveInterval 120"
```

The command above does the following

1. **Uses the pem file** downloaded when creating a keypair to **authenticate the device**.
2. **Connects to the instance as ec2user** and the **corresponding IPv4 address of the instance**.
3. **The flag -o ensures that the connection to the instance is alive for 120 minutes** to prevent getting disconnected from the instance when it is idle.

Connect to EC2 Instance via SSH

Important Note

For MacOS users, an additional command is required to be able to connect to an instance:

chmod 400 <path-to-key-file>

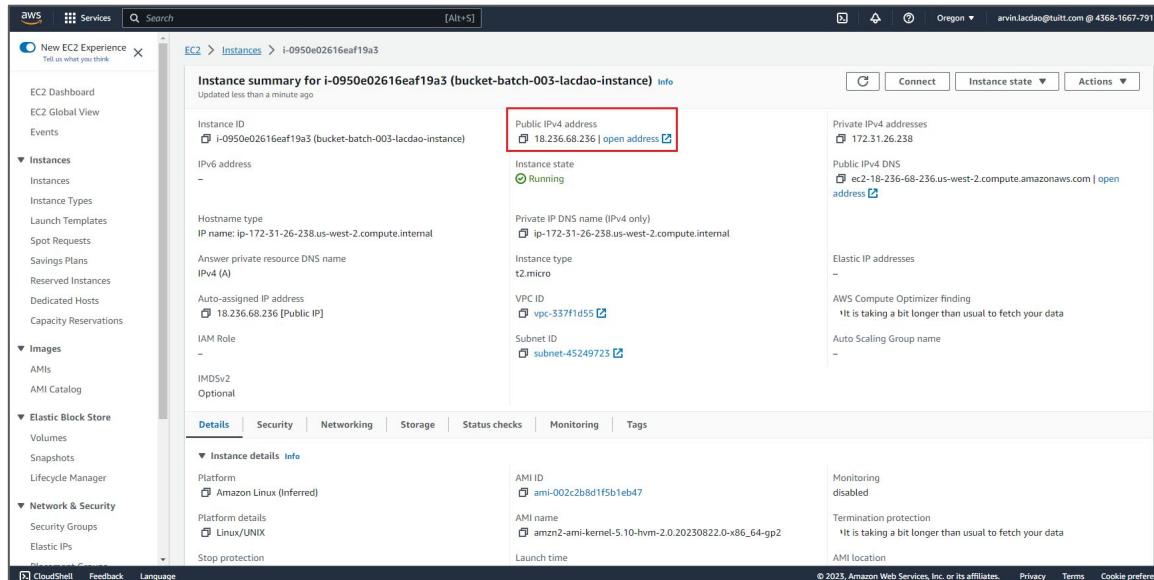
chmod 400 ~/.ssh/bucket-batch-003-lacdao-instance.pem

This allows a change modification of to allow read access to the pem file.

Connect to EC2 Instance via SSH

6. Connect to the EC2 Instance.

You may find the **IPv4 address** in the **instance details page**.



The screenshot shows the AWS EC2 Instances Details page for an instance named `i-0950e02616eaf19a3`. The Public IPv4 address is highlighted in red as `18.236.68.236`. Other visible details include:

- Public IPv4 address:** `18.236.68.236` (with a link to open address)
- Instance state:** Running
- Private IP-DNS name (IPv4 only):** `ip-172-31-26-238.us-west-2.compute.internal`
- Instance type:** t2.micro
- VPC ID:** `vpc-537f1d55`
- Subnet ID:** `subnet-45249723`
- Platform:** Amazon Linux (Inferred)
- AMI ID:** `ami-002c2b8d1f5b1eb47`
- AMI name:** `amzn2-ami-kernel-5.10-hvm-2.0.20230822.0-x86_64-gp2`
- Launch time:** (not explicitly shown)

Connect to EC2 Instance via SSH

6. Connect to the EC2 Instance.

If a prompt appears to continue connecting type yes and press enter.

```
The authenticity of host '18.236.68.236 (18.236.68.236)' can't be established.  
ED25519 key fingerprint is SHA256:TC118w2j0vbxVDCiZE7hS2TxD614+er66oy9pxGbeyc.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

Connect to EC2 Instance via SSH

6. Connect to the EC2 Instance.

Once connected, you should see the following on your terminal.

```
Admin@DESKTOP-K93DCUN MINGW64 ~/ssh
$ ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"

[ec2-user@ip-172-31-26-238 ~]$
```

Installing Node Version Manager and FileZilla

AWS Deployment Serverful Architecture - EC2 Instance

Installing Node Version Manager and FileZilla

1. **Update the packages** in the **EC2 Linux instance** by triggering the command **sudo yum update**.

```
Admin@DESKTOP-K93DCUN MINGW64 ~/.ssh
$ ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"

          _\|_ _\|_
         _\| ( _\| /   Amazon Linux 2 AMI
           \_\|_|_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-26-238 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                         | 3.7 kB     00:00
No packages marked for update
[ec2-user@ip-172-31-26-238 ~]$
```

Installing Node Version Manager and FileZilla

2. **Install node version manager (nvm) using the command `curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash`.**

```
[ec2-user@ip-172-31-26-238 ~]$ curl -o- https://raw.githubusercontent.com/nvm-sh/v0.38.0/install.sh | bash
      % Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload   Total   Spent    Left  Speed
100 14926  100 14926    0     0   337k    0 --:--:-- --:--:-- --:--:--  338k
=> Downloading nvm as script to '/home/ec2-user/.nvm'

=> Appending nvm source string to /home/ec2-user/.bashrc
=> Appending bash_completion source string to /home/ec2-user/.bashrc
=> Close and reopen your terminal to start using nvm or run the following to use
it now:

export NVM_DIR="$HOME/.nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads
nvm bash_completion
[ec2-user@ip-172-31-26-238 ~]$
```

Installing Node Version Manager and FileZilla

3. **Check the installation** of the **node version manager** using the command **nvm --version**.

This will return an error **nvm: command not found** even though it has been successfully installed.

This is common when installing packages that uses the terminal. In most devices, **closing all terminals and reopening them will allow this to work**. For EC2 instances, the instance connection must be reset which will be done in the next step.

```
[ec2-user@ip-172-31-26-238 ~]$ nvm --version
-bash: nvm: command not found
[ec2-user@ip-172-31-26-238 ~]$ |
```

Installing Node Version Manager and FileZilla

4. Restart the connection to the EC2 instance.

Type the command **exit** to **disconnect from the EC2 Instance**.

```
Admin@DESKTOP-K93DCUN MINGW64 ~/ssh
$ ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"
Last login: Mon Sep  4 08:18:42 2023 from 119.111.241.171
[ec2-user@ip-172-31-26-238 ~]$ exit
logout
Connection to 18.236.68.236 closed.

Admin@DESKTOP-K93DCUN MINGW64 ~/ssh
$ |
```

Installing Node Version Manager and FileZilla

4. Restart the connection to the EC2 instance.

Trigger the command below:

```
ssh -i "<keypair-name.pem>" ec2-user@<port> -o "ServerAliveInterval 120"
```

```
ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"
```

```
nvm -version
```

Installing Node Version Manager and FileZilla

4. Restart the connection to the EC2 instance.

```
Admin@DESKTOP-K93DCUN MINGW64 ~/ssh
$ ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"
Last login: Mon Sep 4 07:19:28 2023 from 119.111.241.171

[ec2-user@ip-172-31-26-238 ~]$ nvm --version
0.38.0
[ec2-user@ip-172-31-26-238 ~]$ |
```

Installing Node Version Manager and FileZilla

5. Install **NodeJS 16** with the command **nvm install 16.0.0**.

```
Admin@DESKTOP-K93DCUN MINGW64 ~/.ssh
$ ssh -i "bucket-batch-003-lacdao-instance.pem" ec2-user@18.236.68.236 -o "ServerAliveInterval 120"
Last login: Mon Sep  4 08:21:11 2023 from 119.111.241.171

          _\|_ _\|_
         _\|_(_\|_ /  Amazon Linux 2 AMI
           \_\|_\_|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-26-238 ~]$ nvm install 16.0.0
Downloading and installing node v16.0.0...
Downloading https://nodejs.org/dist/v16.0.0/node-v16.0.0-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v16.0.0 (npm v7.10.0)
Creating default alias: default -> 16.0.0 (-> v16.0.0)
[ec2-user@ip-172-31-26-238 ~]$
```

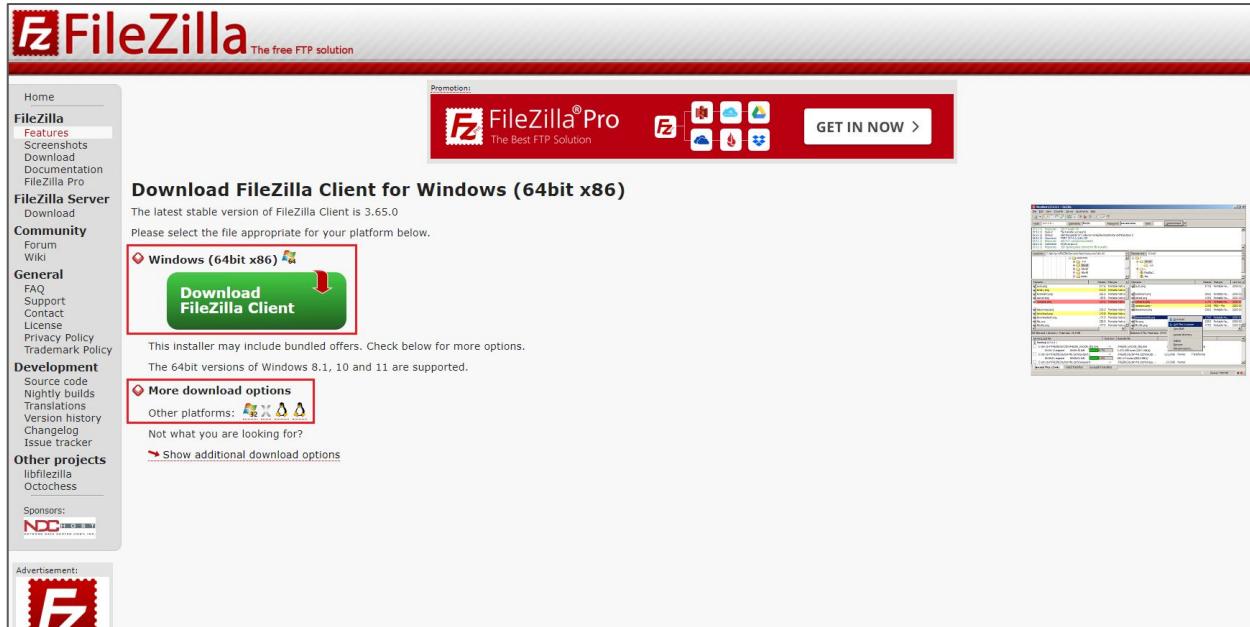
Installing Node Version Manager and FileZilla

6. Trigger the command **`sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8000`** to process requests on port 80 in the EC2 instance to 8000.

```
[ec2-user@ip-172-31-26-238 ~]$ sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8000
[ec2-user@ip-172-31-26-238 ~]$ |
```

Installing Node Version Manager and FileZilla

7. Install FileZilla.



Installing Node Version Manager and FileZilla

7. Install FileZilla.

Please select your edition of FileZilla Client

	FileZilla	FileZilla with manual	FileZilla Pro	FileZilla Pro + CLI
Standard FTP	Yes	Yes	Yes	Yes
FTP over TLS	Yes	Yes	Yes	Yes
SFTP	Yes	Yes	Yes	Yes
Comprehensive PDF manual	-	Yes	Yes	Yes
Amazon S3	-	-	Yes	Yes
Backblaze B2	-	-	Yes	Yes
Dropbox	-	-	Yes	Yes
Microsoft OneDrive	-	-	Yes	Yes
Google Drive	-	-	Yes	Yes
Google Cloud Storage	-	-	Yes	Yes
Microsoft Azure Blob + File Storage	-	-	Yes	Yes
WebDAV	-	-	Yes	Yes
OpenStack Swift	-	-	Yes	Yes
Box	-	-	Yes	Yes
Site Manager synchronization	-	-	Yes	Yes
Command-line interface	-	-	-	Yes
Batch transfers	-	-	-	Yes

Download **Select** **Select** **Select**

Installing Node Version Manager and FileZilla

Important Note

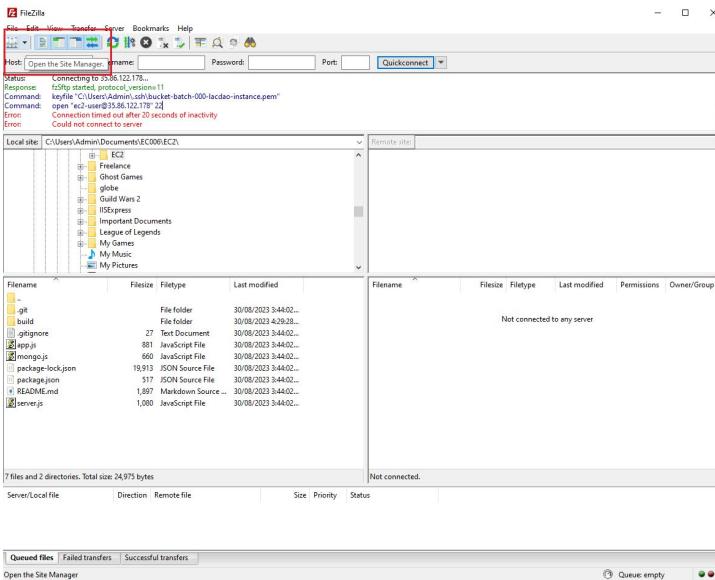
The installation **may have optional offers**, choose **decline** and click on **Next**.

Click on **Next for the succeeding windows** until the option to **Install** is available and wait for it to finish.

Installing Node Version Manager and FileZilla

8. Connect to the EC2 Instance using FileZilla.

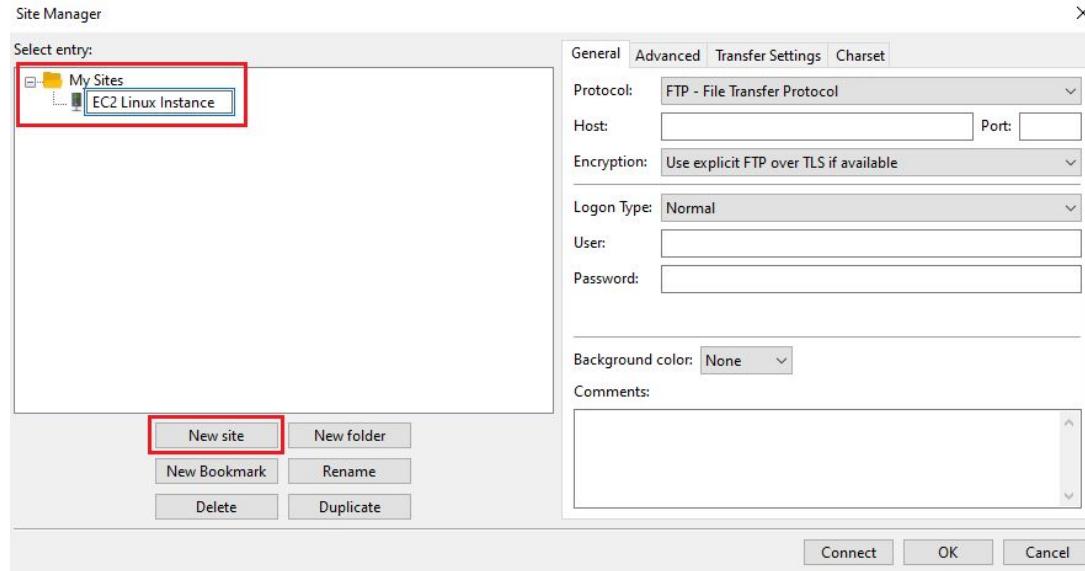
Open **FileZilla** and click **Open the site Manager**



Installing Node Version Manager and FileZilla

8. Connect to the EC2 Instance using FileZilla.

Click on the **New Site Button** and give it a name of **EC2 Linux Instance**



Installing Node Version Manager and FileZilla

8. Connect to the EC2 Instance using FileZilla.

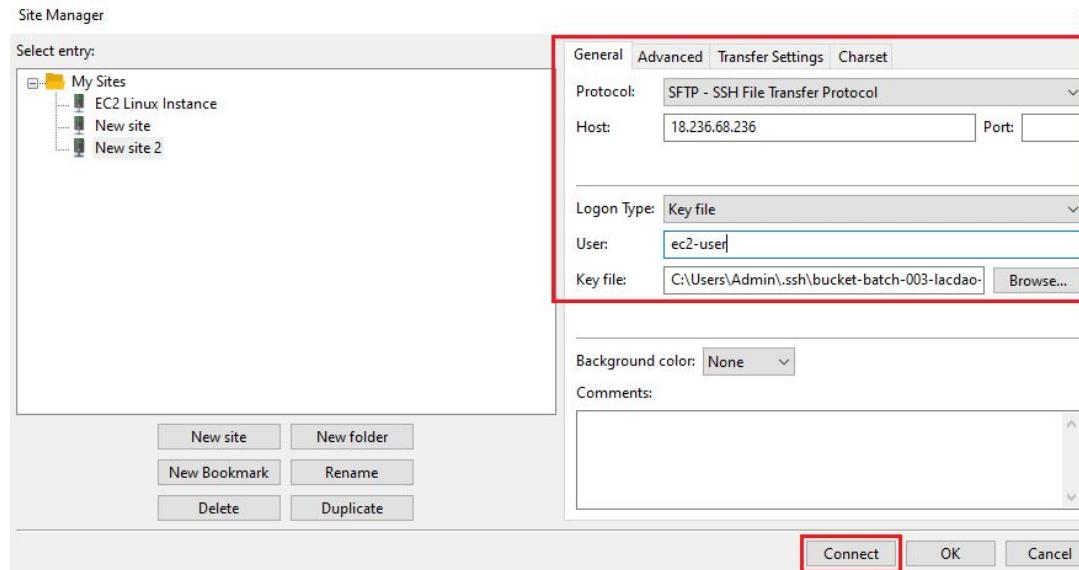
Create your site with the following details and click **Connect**:

- Protocol: SFTP - SSH File Transfer Protocol
- Host: EC2 instance IPv4 address
- Logon Type: Key file
- User: ec2-user
- Key file: Browse your keypair.pem inside the .ssh folder

Installing Node Version Manager and FileZilla

8. Connect to the EC2 Instance using FileZilla.

Create your site with the following details and click **Connect**:



Quiz Form

Tools to Interact with AWS Services

Quiz Form

Click on this [link](#) to navigate to the quiz form for the session on Introduction to AWS and EC2.

Make sure to **tick the checkbox and record your email with your response.**