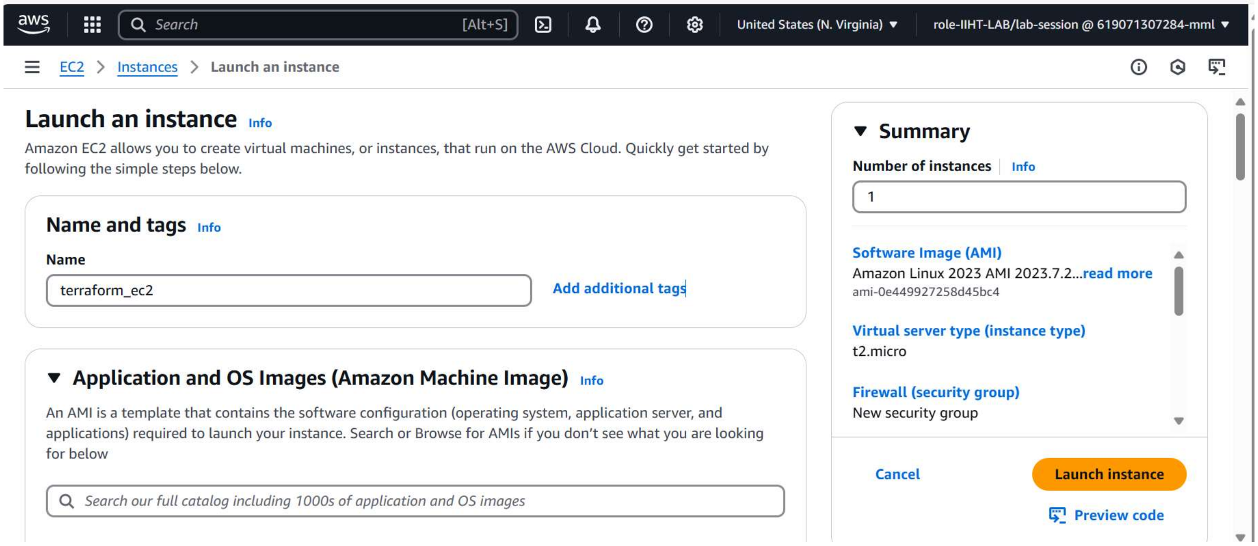
Launch an EC2 Instance

1. Enter the name of the instance as below


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > Launch an instance

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: terraform_ec2

Add additional tags

▼ **Application and OS Images (Amazon Machine Image)** Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

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

▼ **Summary**

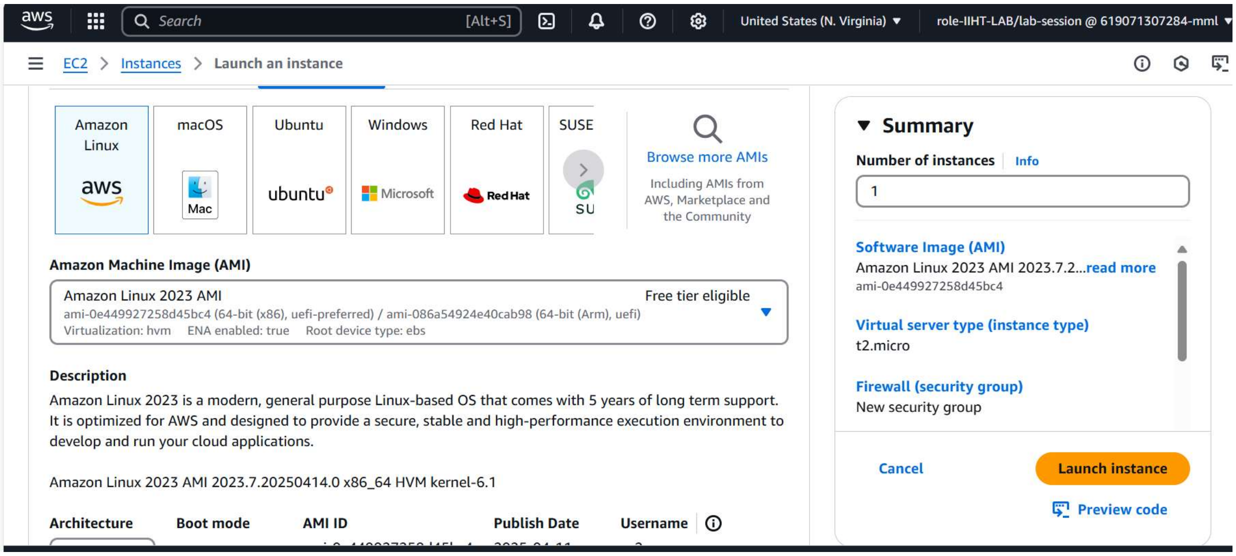
Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0e449927258d45bc4

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

2. Select Amazon Linux OS


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > Launch an instance
Amazon Linux
aws
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE
SU
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI — Free tier eligible
ami-0e449927258d45bc4 (64-bit (x86), uefi-preferred) / ami-086a54924e40cab98 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.7.20250414.0 x86_64 HVM kernel-6.1

▼ **Summary**

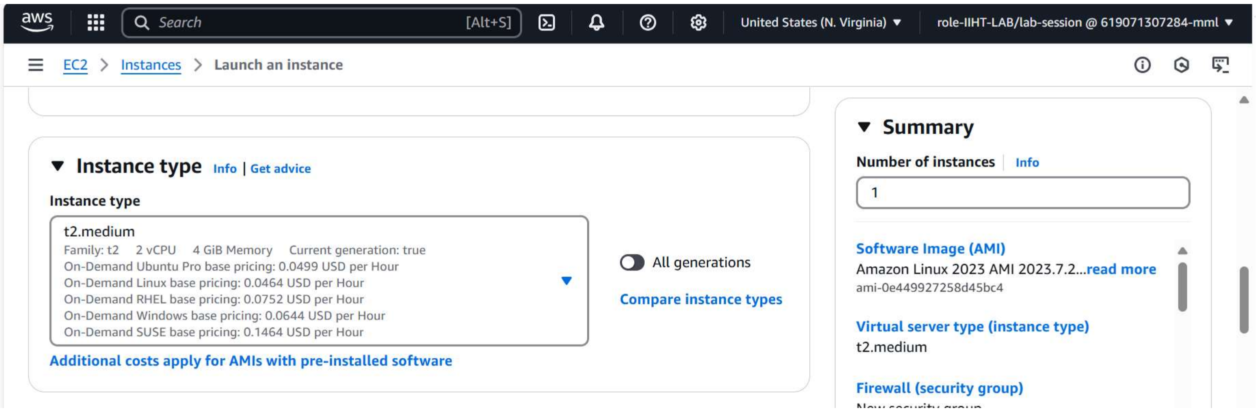
Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0e449927258d45bc4

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

3. Select t2.medium machine type


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > Launch an instance

▼ **Instance type** Info | Get advice

Instance type

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Ubuntu Pro base pricing: 0.0499 USD per Hour
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.0752 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

Additional costs apply for AMIs with pre-installed software

▼ **Summary**

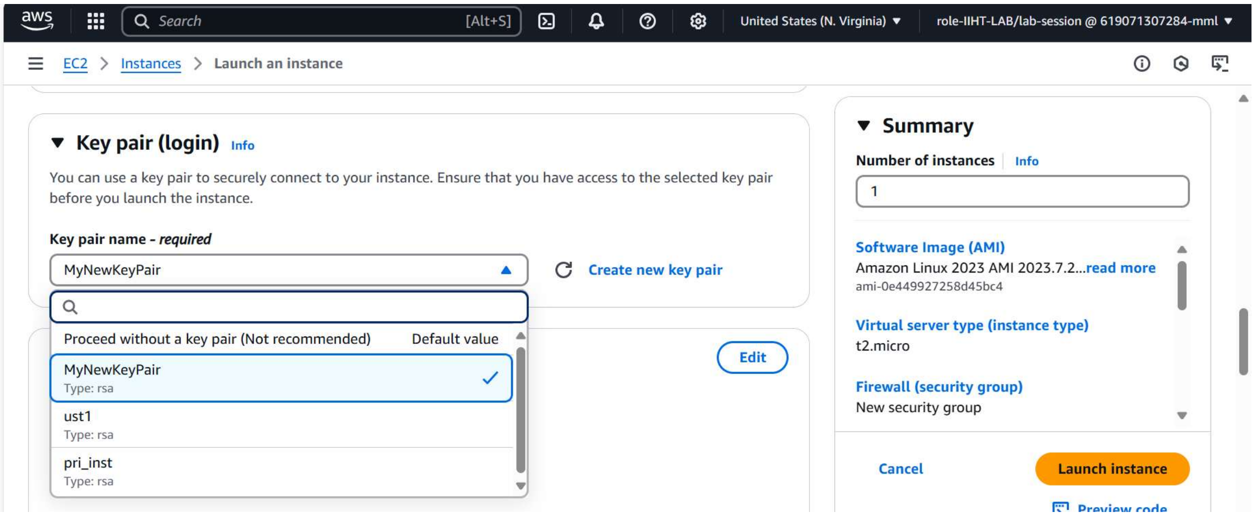
Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0e449927258d45bc4

Virtual server type (instance type)
t2.medium

Firewall (security group)

4. Select the key pair
   Make you have in your local machine downloads,
   If not create new key pair and download the **.pem** key file


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > Launch an instance
Info
Create new key pair
Edit
Info
Cancel
Launch instance
Preview code

▼ 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*

MyNewKeyPair ▲

Proceed without a key pair (Not recommended) Default value

MyNewKeyPair ✓
Type: rsa

ust1
Type: rsa

pri_inst
Type: rsa

▼ Summary

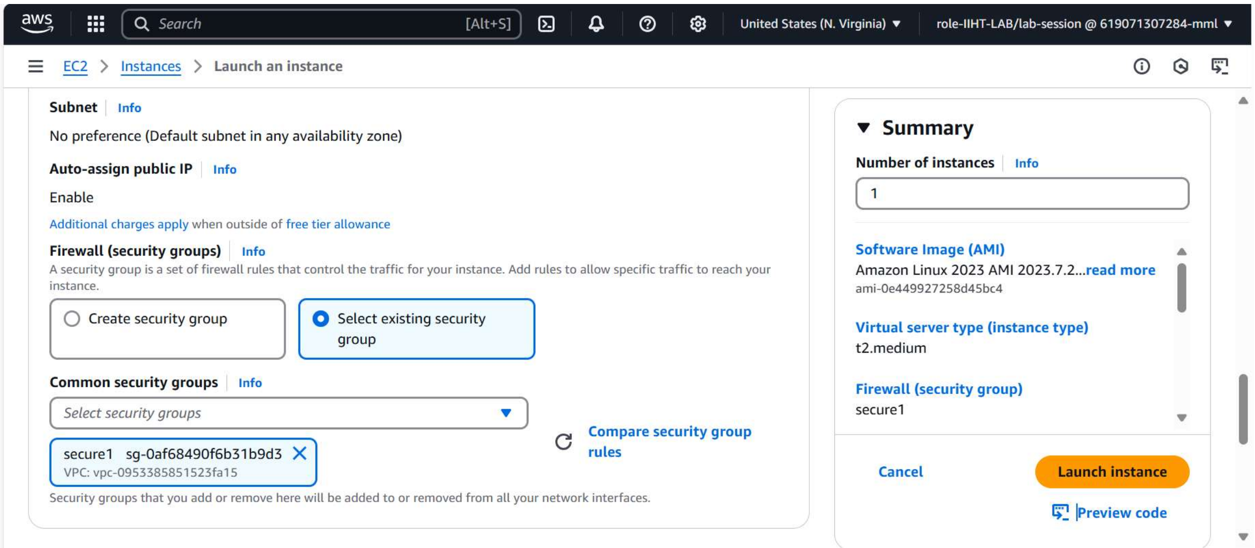
Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0e449927258d45bc4

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

5. Select the security group which has the following inbound ports open
   22 port & 80 port


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > Launch an instance
Info
Compare security group rules
Info
Cancel
Launch instance
Preview code

Subnet Info
No preference (Default subnet in any availability zone)

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 ▼

secure1 sg-0af68490f6b31b9d3 ✕
VPC: vpc-0953385851523fa15

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

▼ Summary

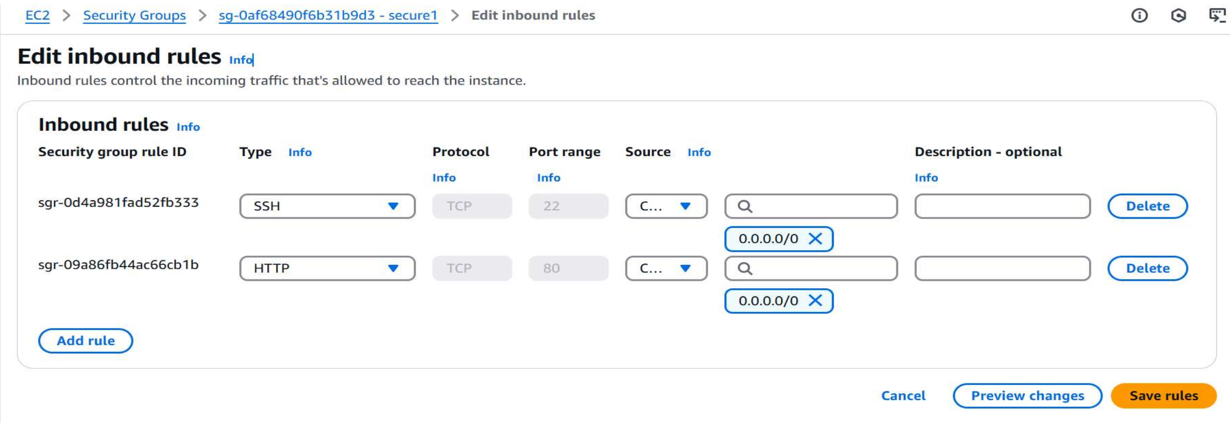
Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.7.2...read more
ami-0e449927258d45bc4

Virtual server type (instance type)
t2.medium

Firewall (security group)
secure1

Note : If you don't have an existing security group with the above ports or if you are unsure about ports create new security group and add the ports


EC2 > Security Groups > sg-0af68490f6b31b9d3 - secure1 > Edit inbound rules
Info
Delete
Delete
Add rule
Cancel
Preview changes
Save rules

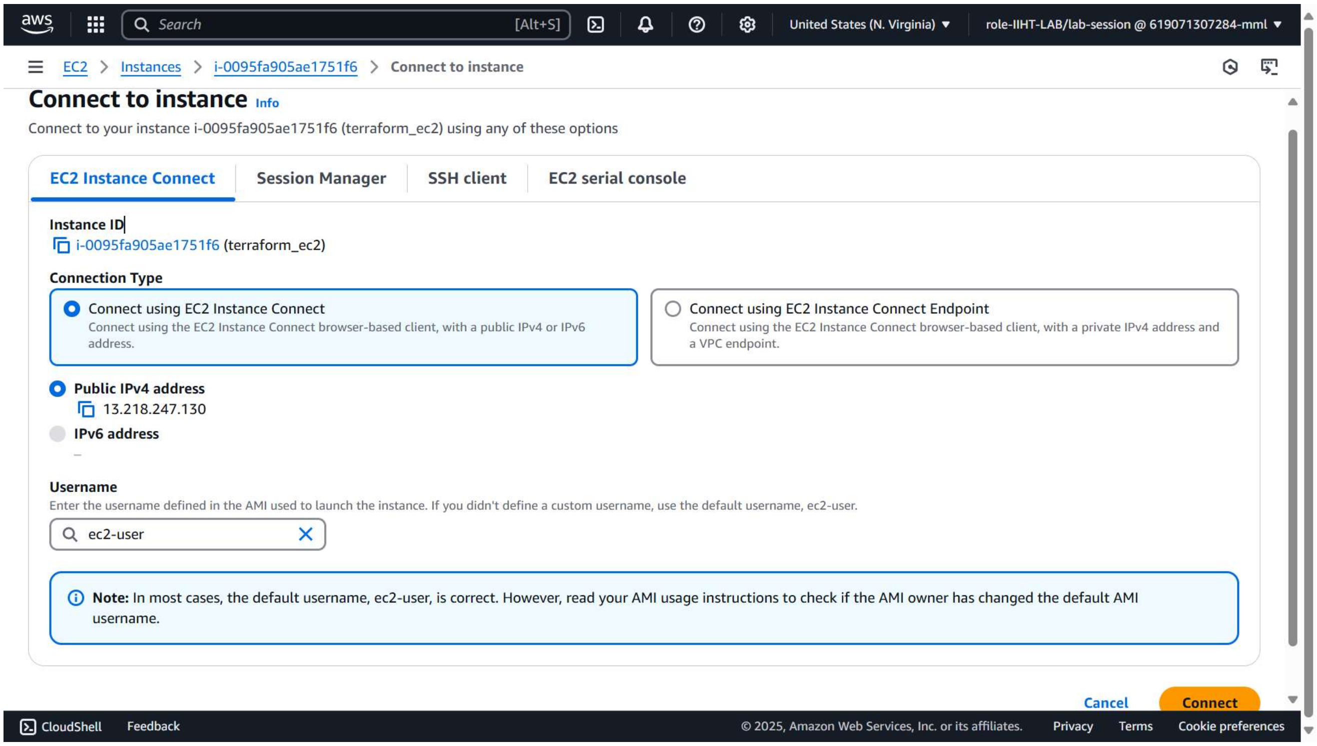
Edit inbound rules Info
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info
sgr-0d4a981fad52fb333	SSH ▼	TCP	22	C... ▼	0.0.0.0/0 ✕	
sgr-09a86fb44ac66cb1b	HTTP ▼	TCP	80	C... ▼	0.0.0.0/0 ✕	

6. Click on **Launch Instance**

7. Now connect the Instance using console connect option


aws
Search [Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-mml ▼
EC2 > Instances > i-0095fa905ae1751f6 > Connect to instance

Connect to instance Info

Connect to your instance i-0095fa905ae1751f6 (terraform_ec2) using any of these options

EC2 Instance Connect
Session Manager
SSH client
EC2 serial console

Instance ID

i-0095fa905ae1751f6 (terraform_ec2)

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**
13.218.247.130

○ **IPv6 address**
–

Username

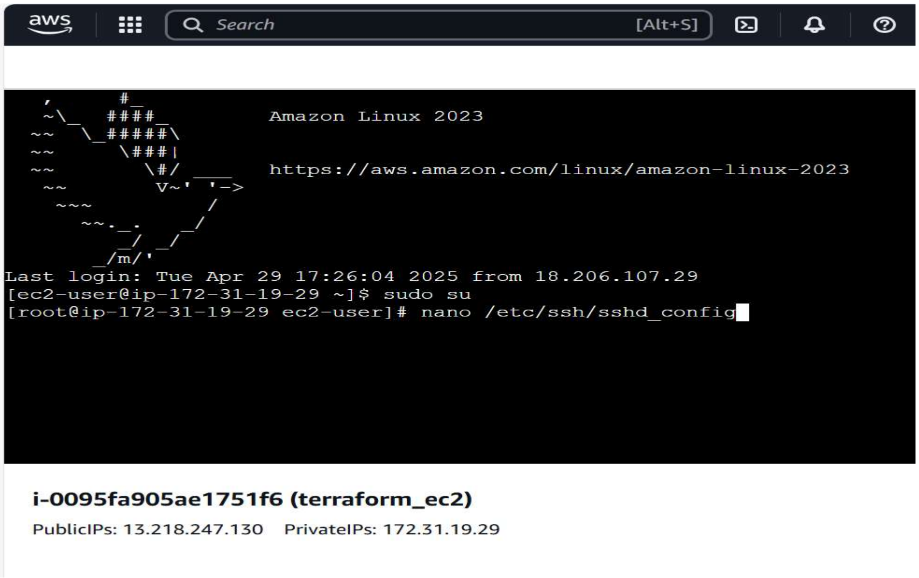
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Note: If you are unable to connect to the instance using connect option, check the security group inbound rules weather you added the **22, SSH port**

8. Enter *sudo su* for superuser privillages
9. Enter the following command to edit the sshd config file
   nano /etc/ssh/sshd_config

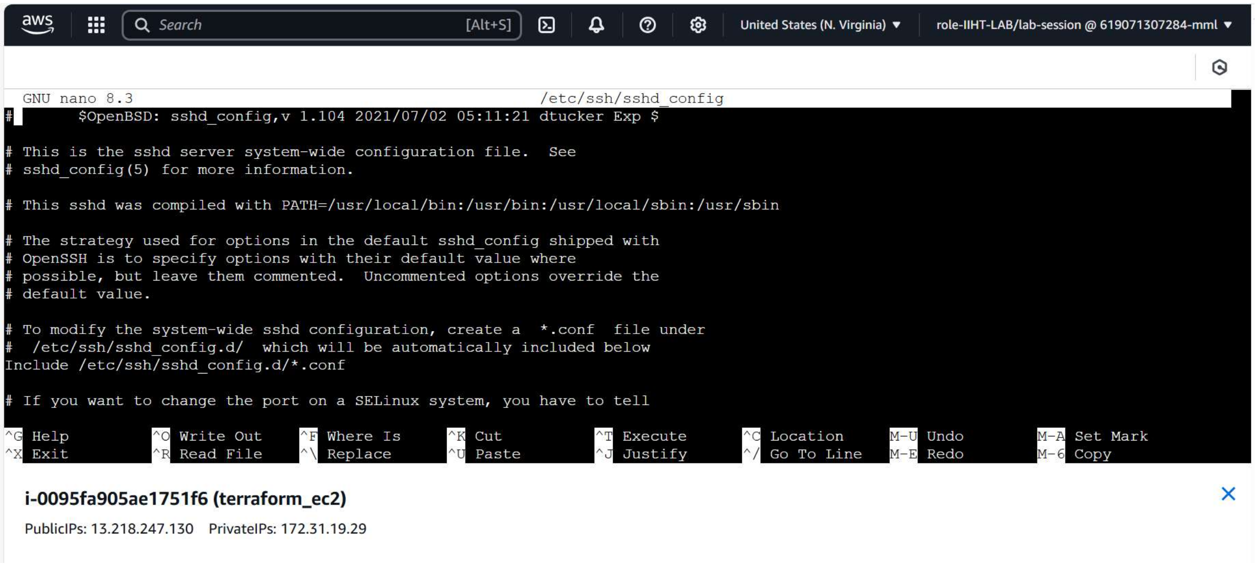


```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Apr 29 17:26:04 2025 from 18.206.107.29
[ec2-user@ip-172-31-19-29 ~]$ sudo su
[root@ip-172-31-19-29 ec2-user]# nano /etc/ssh/sshd_config
```

i-0095fa905ae1751f6 (terraform_ec2)

PublicIPs: 13.218.247.130   PrivateIPs: 172.31.19.29

10. You can see the **sshd_config** file opened in the nano editor

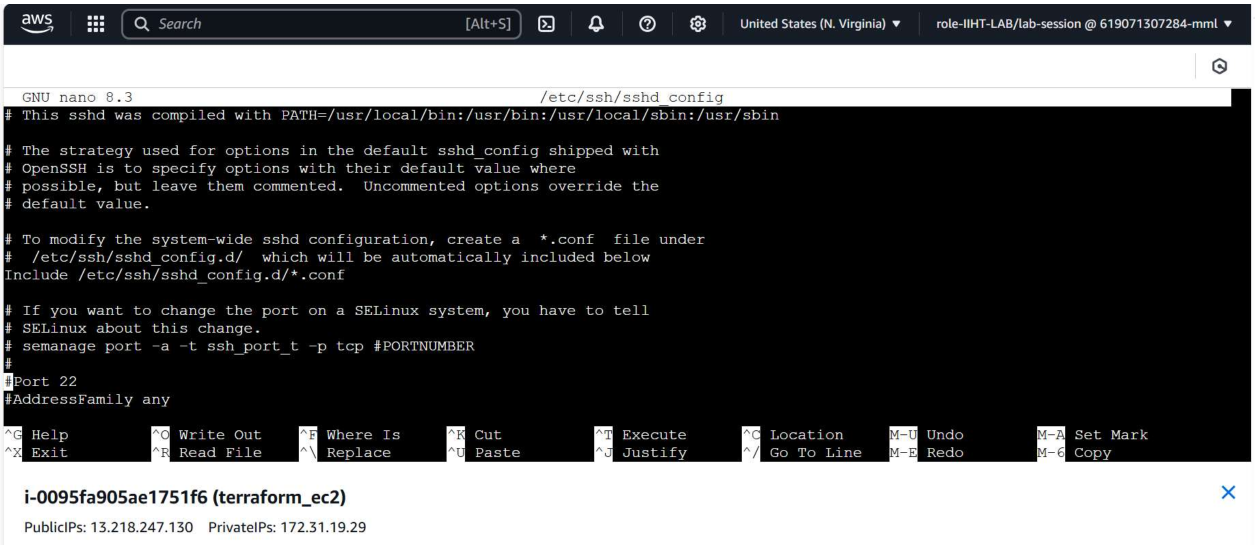


```
  GNU nano 8.3                                  /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.104 2021/07/02 05:11:21 dtucker Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
```

i-0095fa905ae1751f6 (terraform_ec2)

PublicIPs: 13.218.247.130   PrivateIPs: 172.31.19.29

11. Scroll down a little and you can find the Port 22 commented down

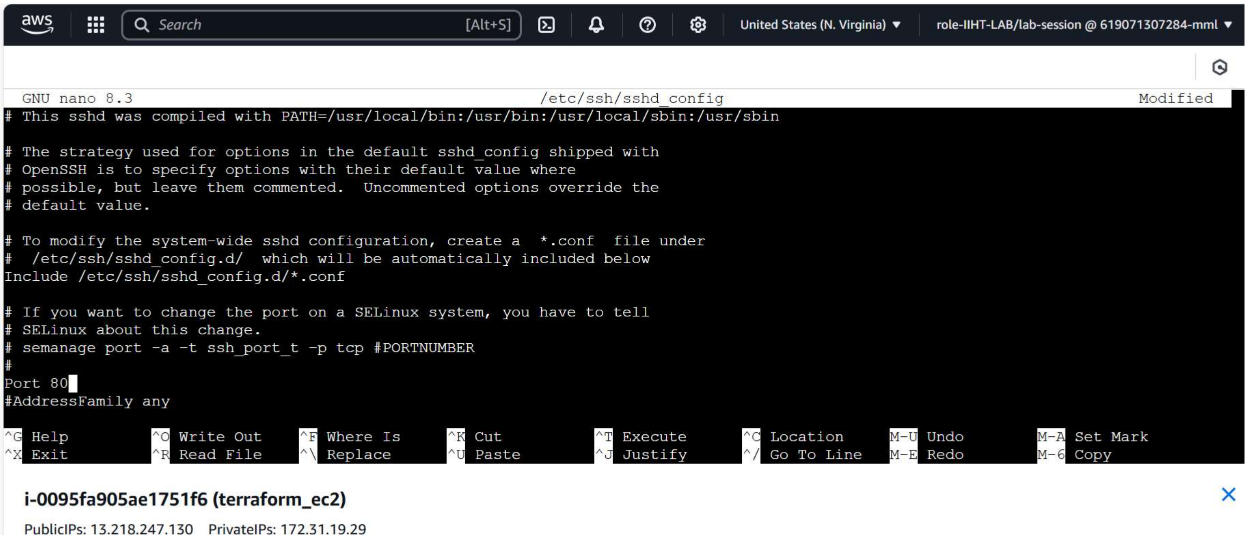


```
  GNU nano 8.3                                  /etc/ssh/sshd_config
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
#Port 22
#AddressFamily any
```

i-0095fa905ae1751f6 (terraform_ec2)

PublicIPs: 13.218.247.130   PrivateIPs: 172.31.19.29

12. Uncomment (remove the "#") before the Port 22 and change the **22 to 80**



```
  GNU nano 8.3                                  /etc/ssh/sshd_config                                  Modified
# This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 80
#AddressFamily any
```

i-0095fa905ae1751f6 (terraform_ec2)

PublicIPs: 13.218.247.130   PrivateIPs: 172.31.19.29

Note: Be cautious while editing this file, since this is a config file, any mistake leads to errors.

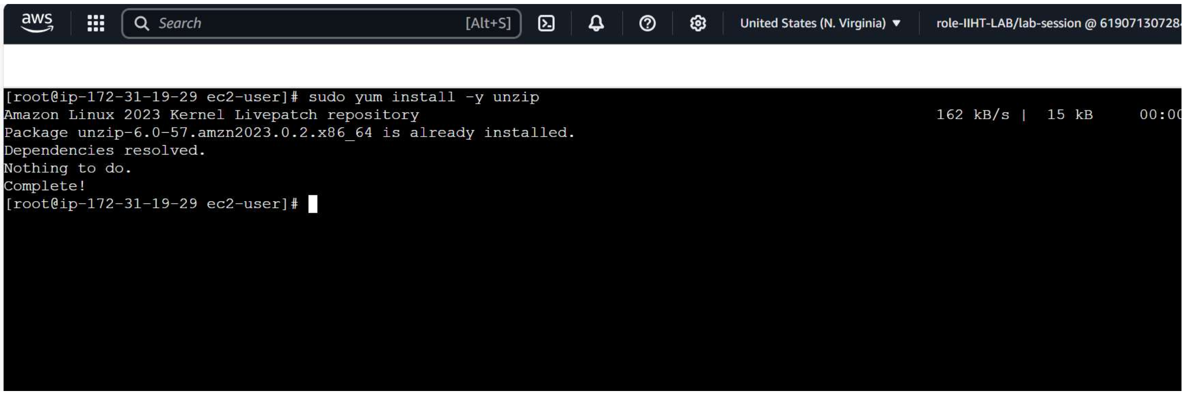
13. Save the sshd_config file
    Click **Cntrl + O** to save the file
14. Exit the editor
    Click **Cntrl + X** to exit the editor

Installing Terraform on the ec2 Instance

Before doing this come to the home directory of the machine by entering following command

cd /home/ec2-user

1. Run the following command to install **unzip** package
   sudo yum install -y unzip

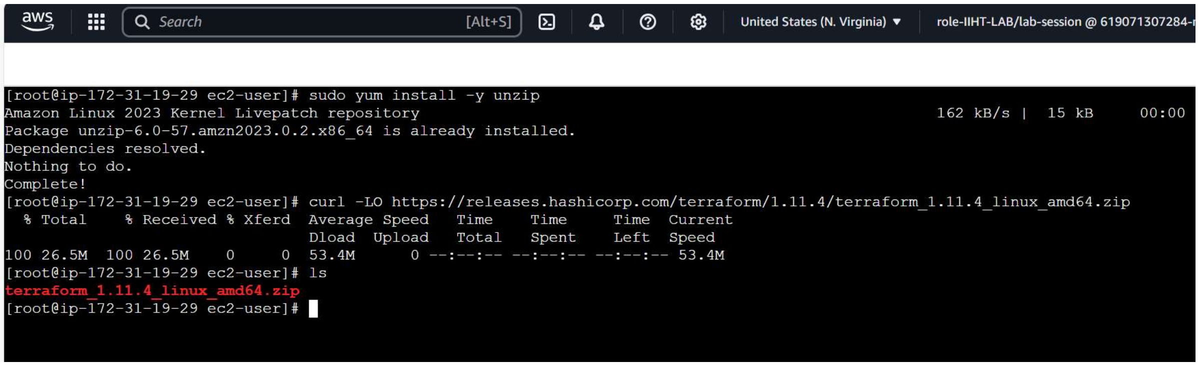

aws
Search
[Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 61907130728

```
[root@ip-172-31-19-29 ec2-user]# sudo yum install -y unzip
Amazon Linux 2023 Kernel Livepatch repository                                      162 kB/s |  15 kB     00:0
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-19-29 ec2-user]# █
```

2. Run the following command to download the terraform zip file

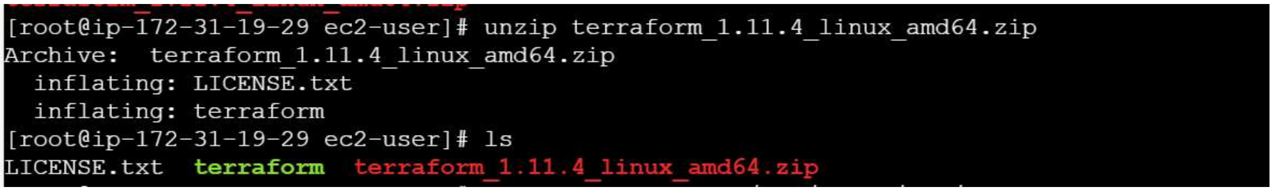
curl -LO https://releases.hashicorp.com/terraform/1.11.4/terraform_1.11.4_linux_amd64.zip

You should find a zip as shown below when you run **ls** command.


aws
Search
[Alt+S]
United States (N. Virginia) ▼
role-IIHT-LAB/lab-session @ 619071307284-

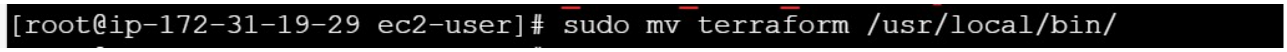
```
[root@ip-172-31-19-29 ec2-user]# sudo yum install -y unzip
Amazon Linux 2023 Kernel Livepatch repository                                      162 kB/s |  15 kB     00:00
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-19-29 ec2-user]# curl -LO https://releases.hashicorp.com/terraform/1.11.4/terraform_1.11.4_linux_amd64.zip
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 26.5M  100 26.5M    0     0  53.4M      0 --:--:-- --:--:-- --:--:-- 53.4M
[root@ip-172-31-19-29 ec2-user]# ls
terraform_1.11.4_linux_amd64.zip
[root@ip-172-31-19-29 ec2-user]# █
```

3. Unzip the download zip file using the following command
   unzip terraform_1.11.4_linux_amd64.zip



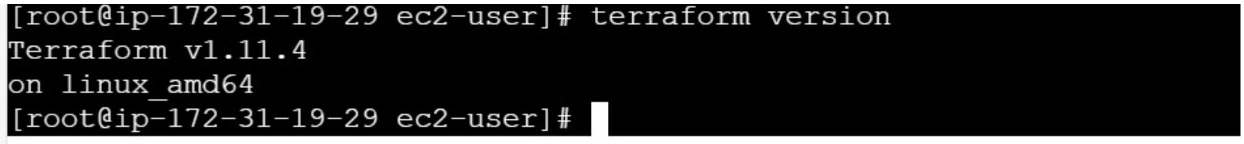
```
[root@ip-172-31-19-29 ec2-user]# unzip terraform_1.11.4_linux_amd64.zip
Archive:  terraform_1.11.4_linux_amd64.zip
  inflating: LICENSE.txt
  inflating: terraform
[root@ip-172-31-19-29 ec2-user]# ls
LICENSE.txt  terraform  terraform_1.11.4_linux_amd64.zip
```

4. Move the folder to usr/local/bin
   sudo mv terraform /usr/local/bin/



```
[root@ip-172-31-19-29 ec2-user]# sudo mv terraform /usr/local/bin/
```

5. Check terraform installation by entering the following command
   terraform version

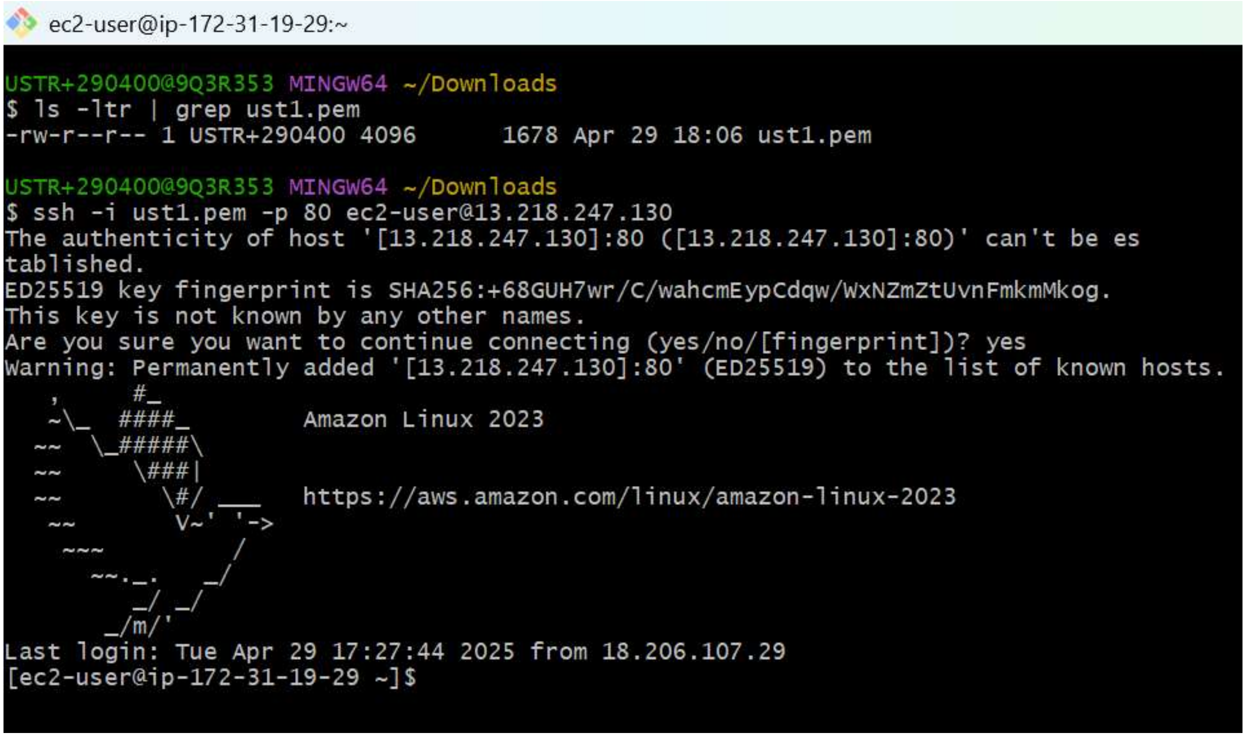


```
[root@ip-172-31-19-29 ec2-user]# terraform version
Terraform v1.11.4
on linux_amd64
[root@ip-172-31-19-29 ec2-user]#
```

Now try connecting to your ec2 instance from your local machine

1. Open git bash or Powershell and try connecting to ec2 instance using ssh
2. Navigate to the folder where you have your public key of the instance (key pair associated to the instance)
3. Enter the following command
   ssh -i <key_pair_name> -p 80 ec2-user@<public-ip-of-ec2-instance>
   example: ssh -i my_ec2_key.pem -p 80 ec2-user@13.218.247.130

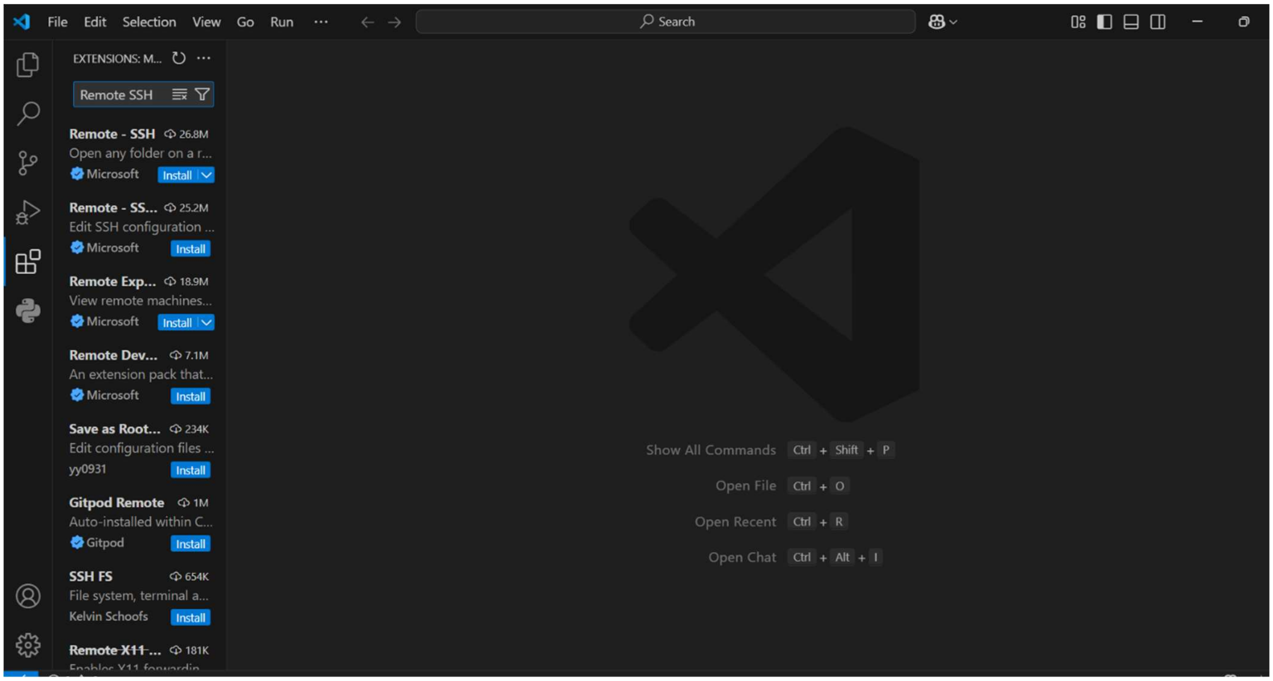
   if you are prompted to continue connecting,
   Type **yes** and click **enter**
4. If you see the following then you are all set to go to next step



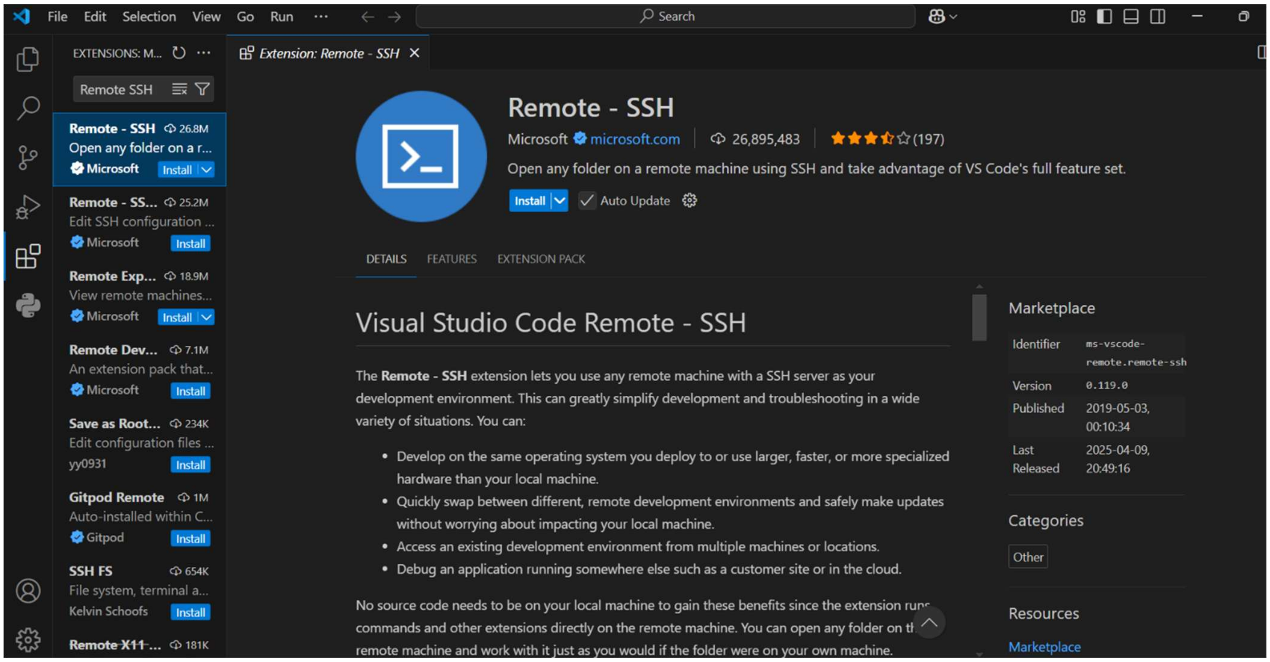
```
USTR+290400@9Q3R353 MINGW64 ~/Downloads
$ ls -ltr | grep ust1.pem
-rw-r--r-- 1 USTR+290400 4096      1678 Apr 29 18:06 ust1.pem

USTR+290400@9Q3R353 MINGW64 ~/Downloads
$ ssh -i ust1.pem -p 80 ec2-user@13.218.247.130
The authenticity of host '[13.218.247.130]:80 ([13.218.247.130]:80)' can't be es
tablished.
ED25519 key fingerprint is SHA256:+68GUH7wr/C/wahcmEypCdqw/WxNZmZtUvnFmkmMkog.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[13.218.247.130]:80' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Apr 29 17:27:44 2025 from 18.206.107.29
[ec2-user@ip-172-31-19-29 ~]$
```

Installing Extensions to vs code

1. Open your vs code and install the following extensions
   Click on Extensions in the sidebar
   Enter "**Remote SSH**" in the search top
   Select the first one.



2. Click on Install



Remote - SSH

Microsoft microsoft.com | 26,895,483 | (197)

Open any folder on a remote machine using SSH and take advantage of VS Code's full feature set.

Visual Studio Code Remote - SSH

The **Remote - SSH** extension lets you use any remote machine with a SSH server as your development environment. This can greatly simplify development and troubleshooting in a wide variety of situations. You can:

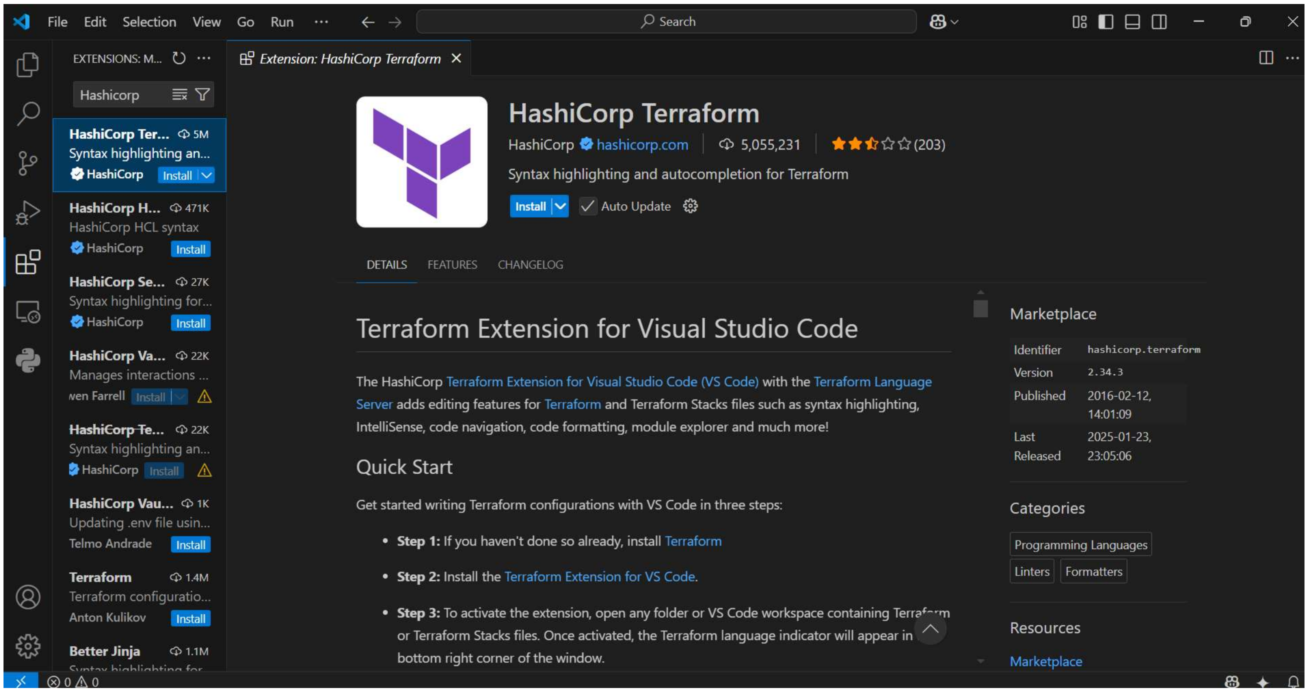
- Develop on the same operating system you deploy to or use larger, faster, or more specialized hardware than your local machine.
- Quickly swap between different, remote development environments and safely make updates without worrying about impacting your local machine.
- Access an existing development environment from multiple machines or locations.
- Debug an application running somewhere else such as a customer site or in the cloud.

No source code needs to be on your local machine to gain these benefits since the extension runs commands and other extensions directly on the remote machine. You can open any folder on th remote machine and work with it just as you would if the folder were on your own machine.

Marketplace

Identifier	ms-vscode-remote.remote-ssh
Version	0.119.0
Published	2019-05-03, 00:10:34
Last Released	2025-04-09, 20:49:16

3. Install HashiCorp Terraform Extension
   Search HashiCorp in the search bar
   Click on first one
   Click on Install



HashiCorp Terraform

HashiCorp hashicorp.com | 5,055,231 | ★★★☆☆ (203)

Syntax highlighting and autocompletion for Terraform

Terraform Extension for Visual Studio Code

The HashiCorp Terraform Extension for Visual Studio Code (VS Code) with the Terraform Language Server adds editing features for Terraform and Terraform Stacks files such as syntax highlighting, IntelliSense, code navigation, code formatting, module explorer and much more!

Quick Start

Get started writing Terraform configurations with VS Code in three steps:

- **Step 1:** If you haven't done so already, install Terraform
- **Step 2:** Install the Terraform Extension for VS Code.
- **Step 3:** To activate the extension, open any folder or VS Code workspace containing Terraform or Terraform Stacks files. Once activated, the Terraform language indicator will appear in the bottom right corner of the window.

Marketplace

Identifier	hashicorp.terraform
Version	2.34.3
Published	2016-02-12, 14:01:09
Last Released	2025-01-23, 23:05:06

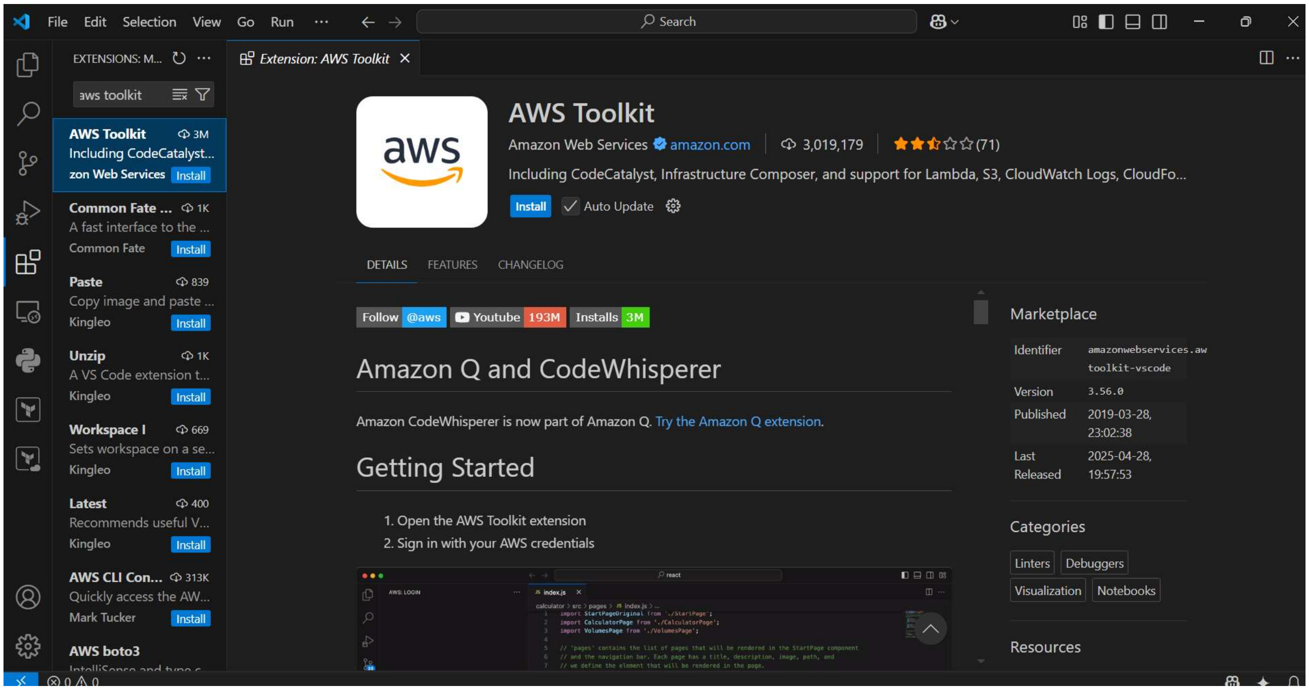
Categories

Programming Languages | Linters | Formatters

Resources

Marketplace

4. Install AWS Toolkit Extension
   search **aws toolkit** in the extension search bar
   Click on the first one and click on Install



AWS Toolkit

Amazon Web Services amazon.com | 3,019,179 | ★★★☆☆ (71)

Including CodeCatalyst, Infrastructure Composer, and support for Lambda, S3, CloudWatch Logs, CloudFo...

Amazon Q and CodeWhisperer

Amazon CodeWhisperer is now part of Amazon Q. Try the Amazon Q extension.

Getting Started

1. Open the AWS Toolkit extension
2. Sign in with your AWS credentials

Marketplace

Identifier	amazonwebservices.aws-toolkit-vscode
Version	3.56.0
Published	2019-03-28, 23:02:38
Last Released	2025-04-28, 19:57:53

Categories

Linters | Debuggers | Visualization | Notebooks

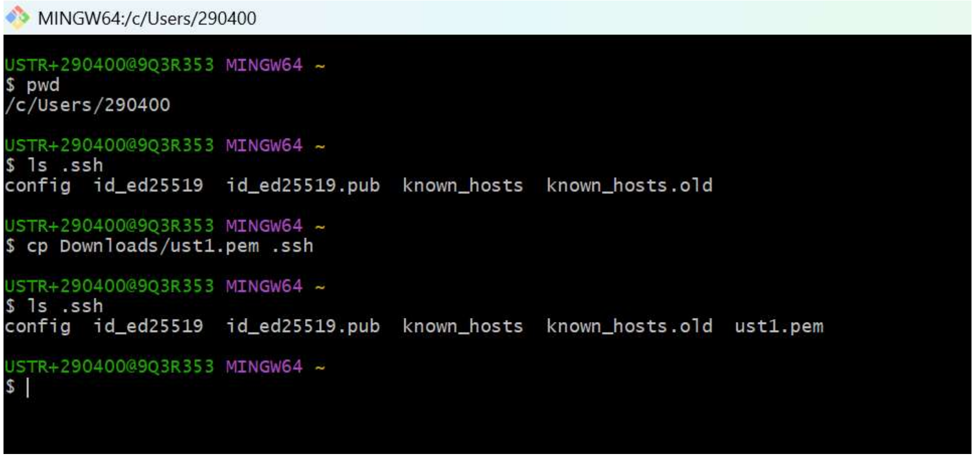
Resources

Connecting your ec2 from vscode

1. Copy the pem file to the .ssh folder of your local machine
   Execute the command in git bash and check the paths of the pem file and .ssh folder before copying
   cp <pem file path> <.ssh folder path>
   Example:-
   Here In my case I have the pem file in my Downloads folder
   Make sure you are in the home directory of the user (your employee id user)

Run the following command if you are in the same path and you have your pem file in the **Downloads** folder

1. pwd
   check you are at the home directory of user
   here its: /c/Users/290400
2. ls .ssh
   to check you have the folder and it has config file in it
3. cp Downloads/ust1.pem .ssh
   copies the pem file to the .ssh folder
4. ls .ssh
   checking weather copy is successful



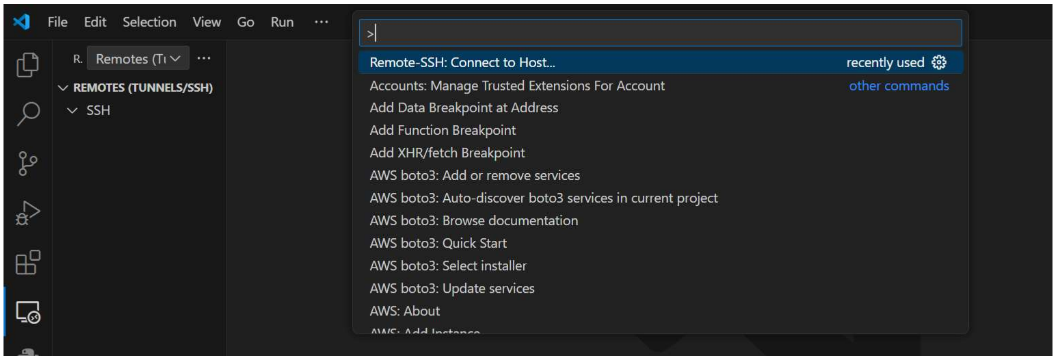
MINGW64:/c/Users/290400

```
USTR+290400@9Q3R353 MINGW64 ~
$ pwd
/c/Users/290400

USTR+290400@9Q3R353 MINGW64 ~
$ ls .ssh
config  id_ed25519  id_ed25519.pub  known_hosts  known_hosts.old

USTR+290400@9Q3R353 MINGW64 ~
$ cp Downloads/ust1.pem .ssh

USTR+290400@9Q3R353 MINGW64 ~
$ ls .ssh
config  id_ed25519  id_ed25519.pub  known_hosts  known_hosts.old  ust1.pem

USTR+290400@9Q3R353 MINGW64 ~
$ |
```

Now the relative path of the pem file should be something like below:
~/.ssh/ust1.pem (replace with your key name)
make a note, we need it in the next step

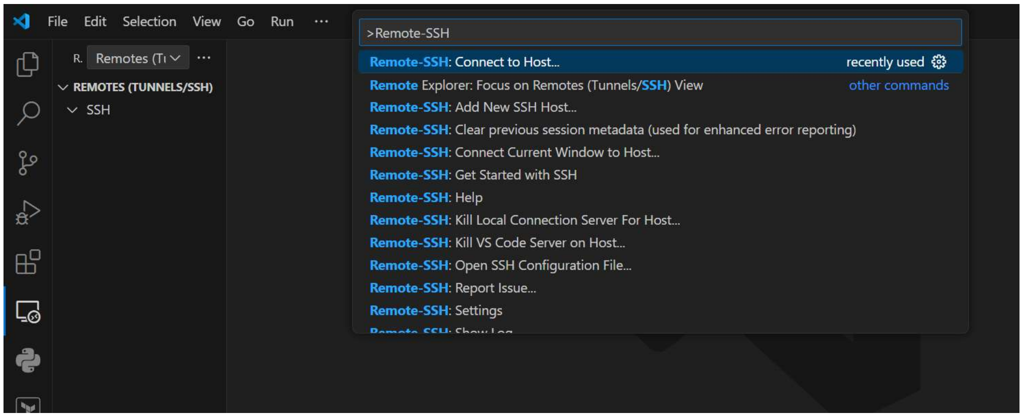
In vscode click F1 (check whether you have turned fn lock)

Enter Remote SSH in the search bar and select Remote SSH as shown below



>|

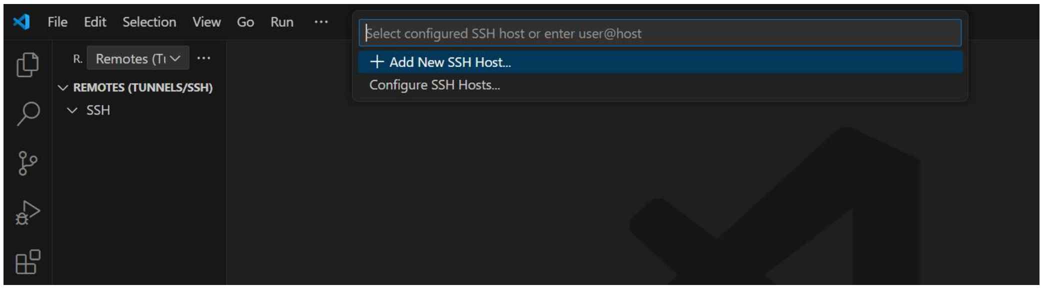
File
Edit
Selection
View
Go
Run
Remotes (Tı
REMOTES (TUNNELS/SSH)
SSH
Remote-SSH: Connect to Host...
recently used
Accounts: Manage Trusted Extensions For Account
other commands
Add Data Breakpoint at Address
Add Function Breakpoint
Add XHR/fetch Breakpoint
AWS boto3: Add or remove services
AWS boto3: Auto-discover boto3 services in current project
AWS boto3: Browse documentation
AWS boto3: Quick Start
AWS boto3: Select installer
AWS boto3: Update services
AWS: About



>Remote-SSH

File
Edit
Selection
View
Go
Run
Remotes (Tı
REMOTES (TUNNELS/SSH)
SSH
Remote-SSH: Connect to Host...
recently used
Remote Explorer: Focus on Remotes (Tunnels/SSH) View
other commands
Remote-SSH: Add New SSH Host...
Remote-SSH: Clear previous session metadata (used for enhanced error reporting)
Remote-SSH: Connect Current Window to Host...
Remote-SSH: Get Started with SSH
Remote-SSH: Help
Remote-SSH: Kill Local Connection Server For Host...
Remote-SSH: Kill VS Code Server on Host...
Remote-SSH: Open SSH Configuration File...
Remote-SSH: Report Issue...
Remote-SSH: Settings

Now you will be shown 2 options like this, select first option i.e., **+Add New SSH Host..**

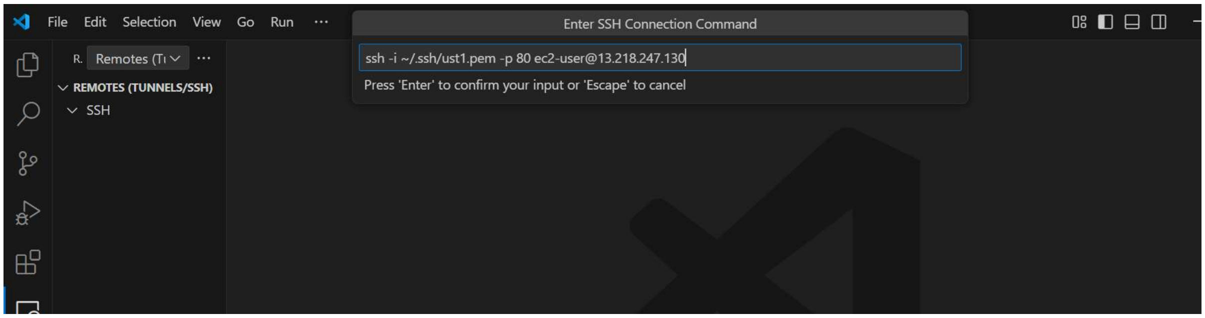


Select configured SSH host or enter user@host

+ Add New SSH Host...

Configure SSH Hosts...

Then Enter the following command by replacing your key-pair (pem file) name and your ec2 instance public IP

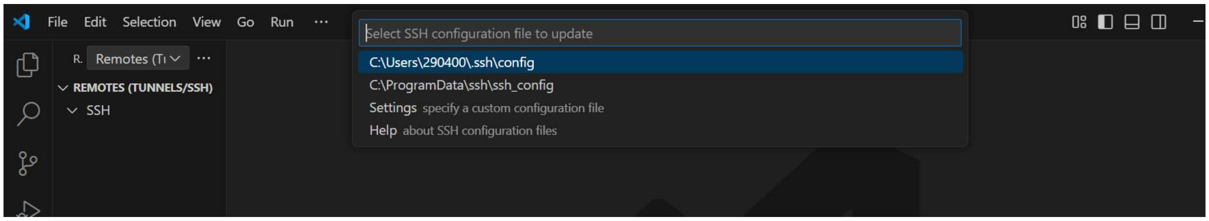


Enter SSH Connection Command

```
ssh -i ~/.ssh/ust1.pem -p 80 ec2-user@13.218.247.130
```

Press 'Enter' to confirm your input or 'Escape' to cancel

Now select the ssh config where the Host details should be added
Select the first one (.ssh folder should be present in your home directory of user)



Select SSH configuration file to update

C:\Users\290400\.ssh\config

C:\ProgramData\ssh\ssh_config

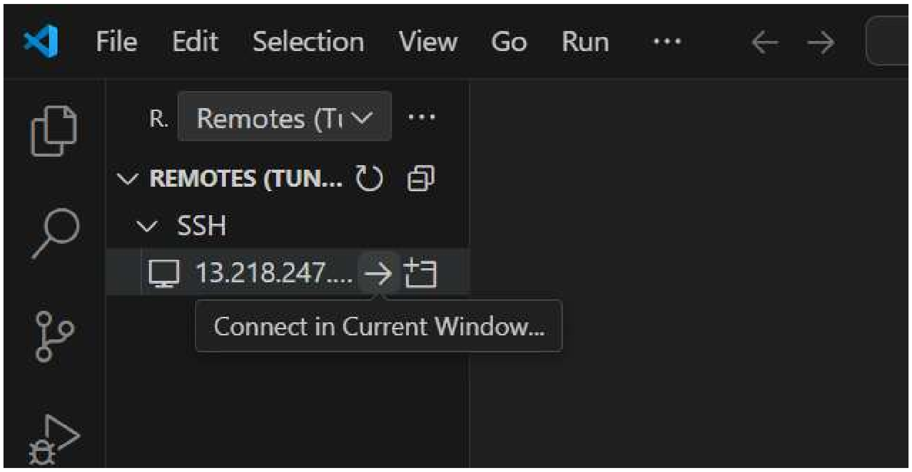
Settings specify a custom configuration file

Help about SSH configuration files

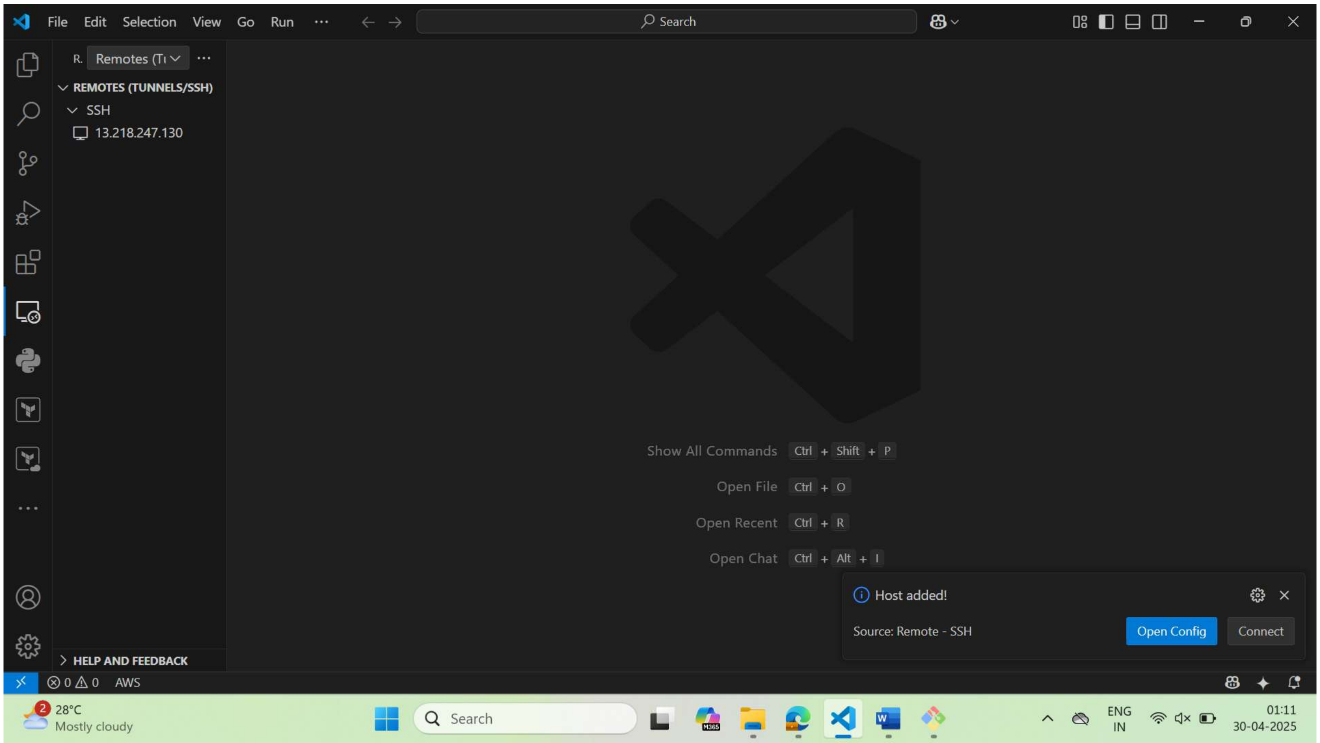
After that you should be able to see the Public IP in the left side as below (if adding host is successful)

You will be prompted to connect to the instance on the right bottom, click on connect

Note: If you miss the above option you can choose the connect option present in the left side as below (click on right pointing arrow beside the Public IP)



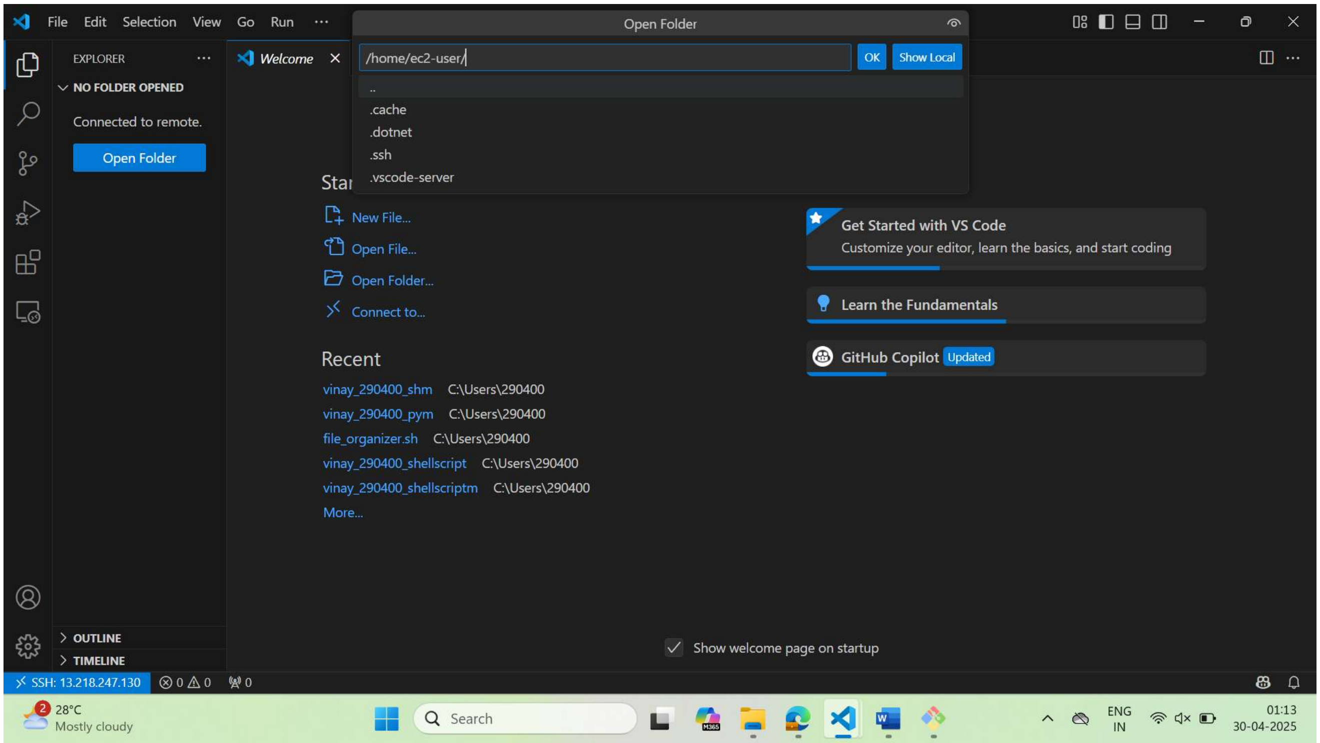
File Edit Selection View Go Run ···
R. Remotes (Tı
REMOTES (TUN...
SSH
13.218.247....
Connect in Current Window...



Host added!

Source: Remote - SSH

A new window will be opened and select the folder to be open from the Remote ec2 instance

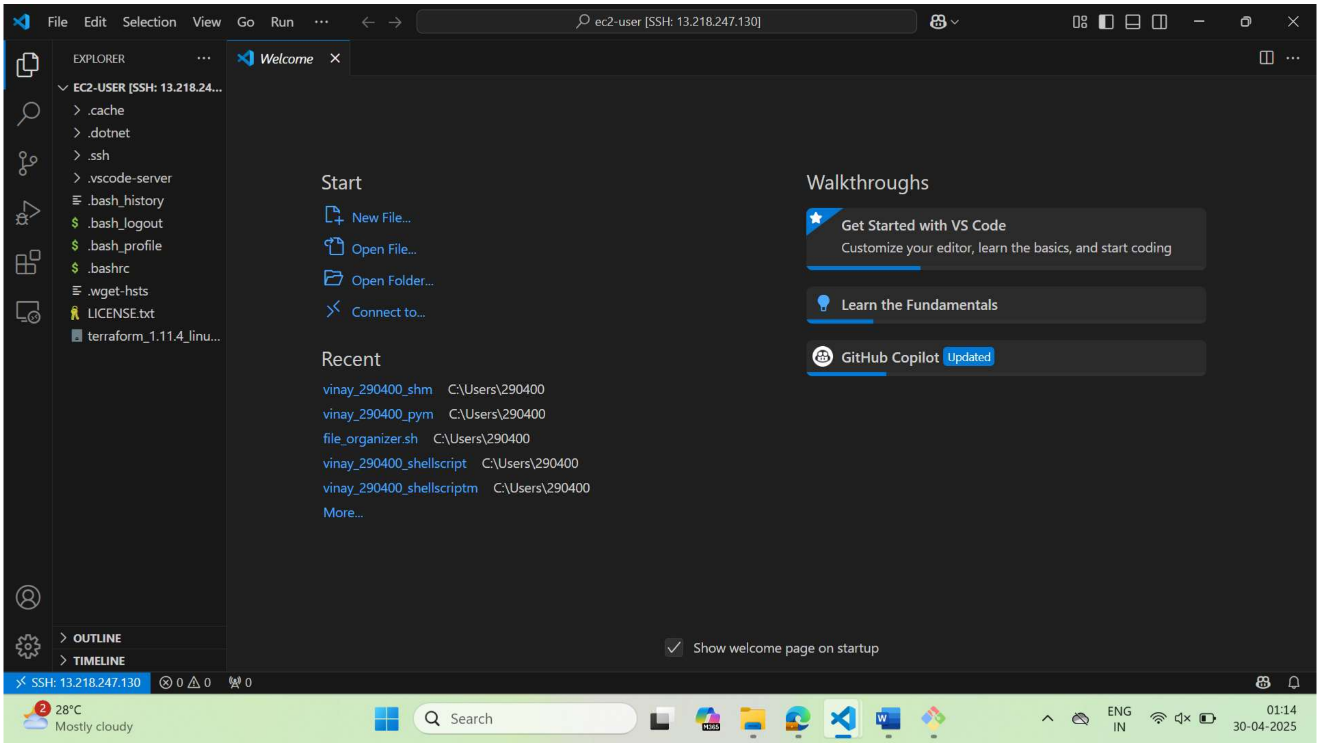


Connected to remote.

```
/home/ec2-user/
..
.cache
.dotnet
.ssh
.vscode-server
```

vinay_290400_shm	C:\Users\290400
vinay_290400_pym	C:\Users\290400
file_organizer.sh	C:\Users\290400
vinay_290400_shellscript	C:\Users\290400
vinay_290400_shellscriptm	C:\Users\290400

If everything works well you will get the folders as below


File
Edit
Selection
View
Go
Run
ec2-user [SSH: 13.218.247.130]
EXPLORER
Welcome
EC2-USER [SSH: 13.218.24...
.cache
.dotnet
.ssh
.vscode-server
.bash_history
.bash_logout
.bash_profile
.bashrc
.wget-hsts
LICENSE.txt
terraform_1.11.4_linu...
Start
New File...
Open File...
Open Folder...
Connect to...
Recent
vinay_290400_shm C:\Users\290400
vinay_290400_pym C:\Users\290400
file_organizer.sh C:\Users\290400
vinay_290400_shellscript C:\Users\290400
vinay_290400_shellscriptm C:\Users\290400
More...
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
Learn the Fundamentals
GitHub Copilot Updated
Show welcome page on startup
OUTLINE
TIMELINE
SSH: 13.218.247.130
0 0
0
28°C
Mostly cloudy
Search
ENG
IN
01:14
30-04-2025