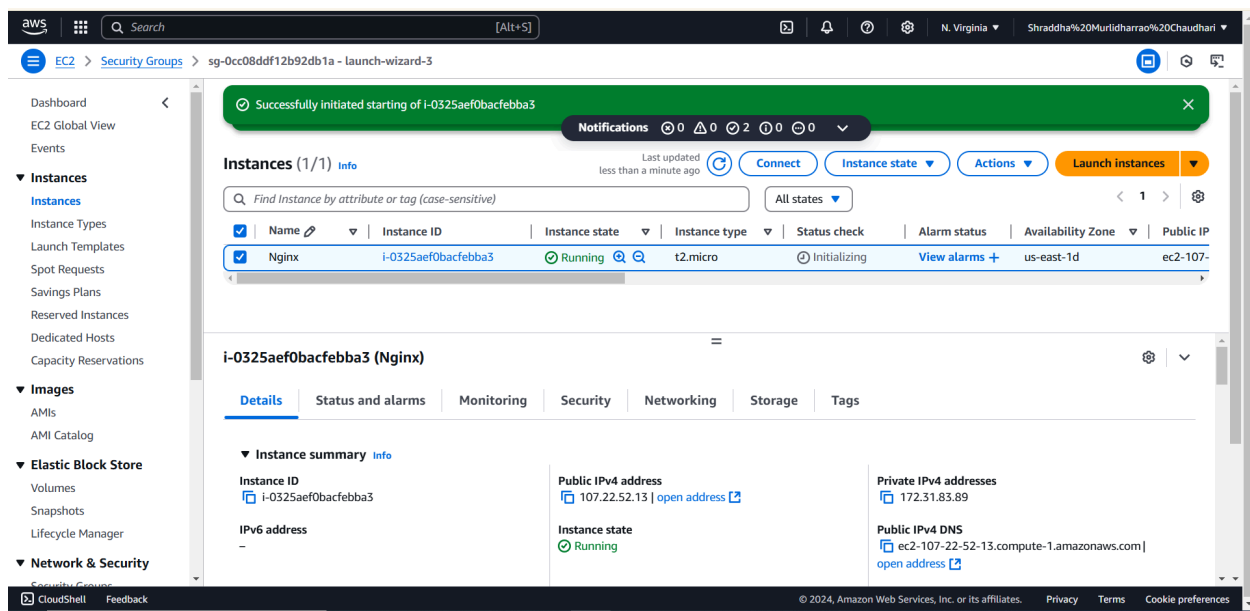


# Documentations:

## 1. Setting Up Nginx with a Custom 404 Error Page

This document outlines the step-by-step process of configuring an Nginx server to display a custom 404 error page when a user accesses a non-existent route.

### Step 1: Launch the Instance



### Step 2: Install Nginx

-> apt update

-> apt install nginx -y

```
aws
[Alt+S]
N. Virginia
Shraddha%20Murtidharrao%20Chaudhari

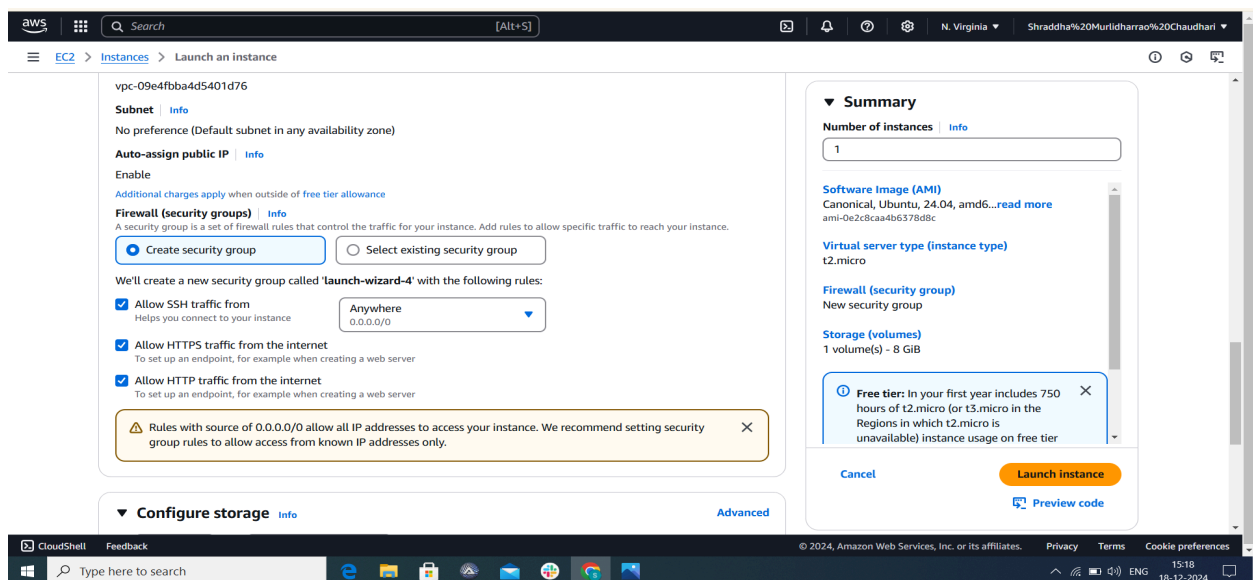
root@ip-172-31-83-89:~# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  nginx-common
Suggested packages:
  fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  nginx nginx-common
0 upgraded, 2 newly installed, 0 to remove and 53 not upgraded.
Need to get 552 kB of archives.
After this operation, 1596 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx-common all 1.24.0-2ubuntu7.1 [31.2 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 nginx amd64 1.24.0-2ubuntu7.1 [521 kB]
Fetched 552 kB in 0s (18.8 MB/s)
Preconfiguring packages ...
Selecting previously unselected package nginx-common.
(Reading database ... 70601 files and directories currently installed.)
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7.1_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7.1) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7.1_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx (1.24.0-2ubuntu7.1) ...
Setting up nginx-common (1.24.0-2ubuntu7.1) ...
```

Verify that Nginx is installed and running:

-> **systemctl status nginx**

## Step 3: Configure Firewall Rules:

Allow HTTP/HTTPS traffic through the firewall:



-> ufw allow 'Nginx Full'

Check the firewall status:

-> ufw status

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-0325aef0bacfebba38&osUser=ubuntu&sshPort=22&addressFamily=i...
aws [Alt+S] N. Virginia Shradha Murdharao Chaudhari

root@ip-172-31-83-89:~# ufw status
Status: inactive
root@ip-172-31-83-89:~# ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
root@ip-172-31-83-89:~# ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
root@ip-172-31-83-89:~# ufw status
Status: active

To Action From
--
Nginx Full ALLOW Anywhere
Nginx Full (v6) ALLOW Anywhere (v6)

root@ip-172-31-83-89:~#
```

#### Step 4: Create a Custom 404 Error Page:

Create a custom HTML file to serve as the 404 error page:

-> vim/var/www/html/custom\_404.html

```
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&instanceId=i-0325aef0bacfebba38&osUser=ubuntu&sshPort=22&addressFamily=i...
aws [Alt+S] N. Virginia Shradha Murdharao Chaudhari

<!DOCTYPE html>
<html>
<head>
<title>404 Not Found</title>
</head>
<body>
<h1>Oops! Page Not Found</h1>
<p>The page you're looking for doesn't exist.</p>
</body>
</html>

"/var/www/html/custom_404.html" 11L, 183B
```

Save and close the file

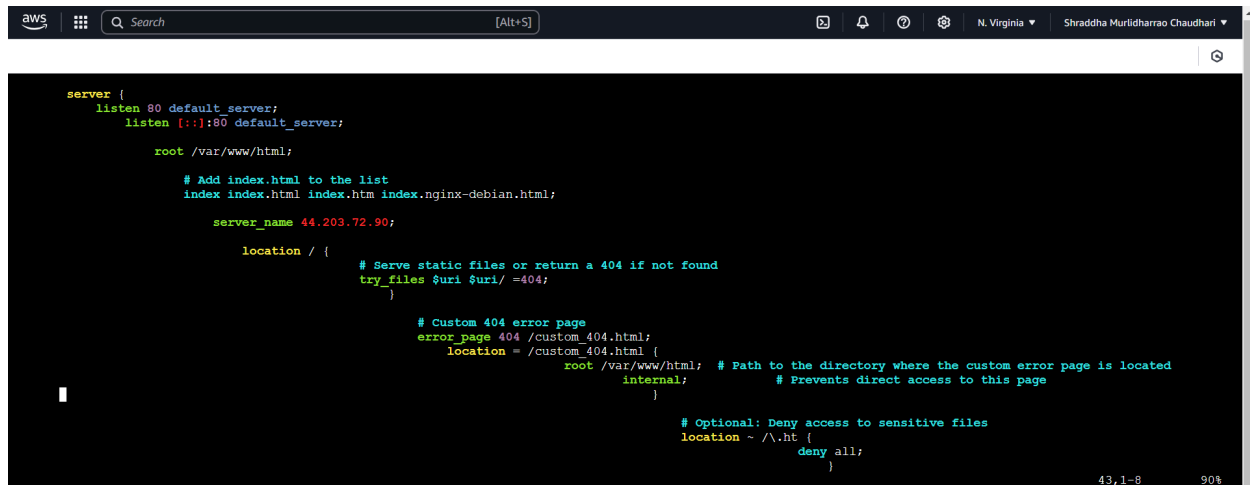
Verify the file exists:

-> `ls -l /var/www/html/custom_404.html`

### Step 5: Update the Nginx Configuration:

Modify the Nginx default server block to include the custom 404 page configuration.

-> `vim/etc/nginx/sites-available/default`



```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;

    # Add index.html to the list
    index index.html index.htm index.nginx-debian.html;

    server_name 44.203.72.90;

    location / {
        # Serve static files or return a 404 if not found
        try_files $uri $uri/ =404;
    }

    # Custom 404 error page
    error_page 404 /custom_404.html;
    location = /custom_404.html {
        root /var/www/html; # Path to the directory where the custom error page is located
        internal;           # Prevents direct access to this page
    }

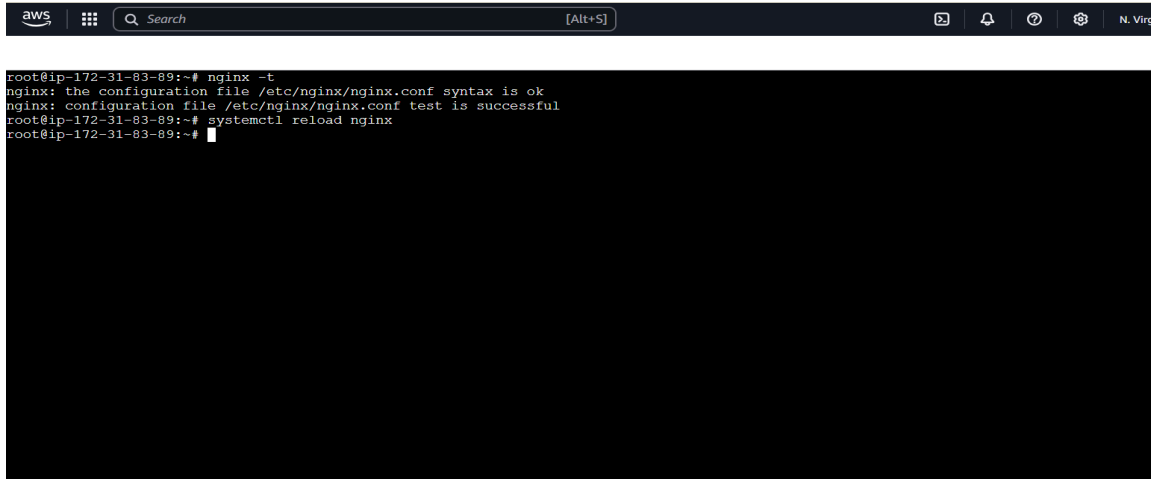
    # Optional: Deny access to sensitive files
    location ~ /\.ht {
        deny all;
    }
}
```

Save and close the file

### Step 6: Test the Configuration:

Validate the Nginx Configuration:

-> `sudo nginx -t`

A screenshot of an AWS terminal window. The terminal shows a series of commands and their outputs. The first command is 'nginx -t', which outputs 'nginx: the configuration file /etc/nginx/nginx.conf syntax is ok' and 'nginx: configuration file /etc/nginx/nginx.conf test is successful'. The second command is 'systemctl reload nginx', which is followed by a prompt 'root@ip-172-31-83-89:~#'. The terminal window has a dark background and a light-colored text. The top of the window shows the AWS logo, a search bar, and some system icons.

```
root@ip-172-31-83-89:~# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ip-172-31-83-89:~# systemctl reload nginx
root@ip-172-31-83-89:~#
```

Reload Nginx to apply changes:

-> **systemctl reload nginx**

Step 7: Test the Custom 404 Error Page

Use **curl** to test:

-> **curl -i http://<your\_server\_ip>/nonexistent**

Browser Test:

1. Open a browser and navigate to **http://<your\_server\_ip>/nonexistent**.
2. You should see your custom 404 error page with the message:
  - "Oops! Page Not Found"
  - "The page you're looking for doesn't exist."



## Oops! Page Not Found

The page you're looking for doesn't exist.

---

## 2.Setting Up AWS EC2 Web Server and Client in a VPC Using Bastion Host (Ubuntu-Based):

### Step 1: Create VPC and Subnets

1. Navigate to the VPC Service:
  - Log in to the AWS Management Console.
  - Go to VPC under the Networking & Content Delivery section.
2. Create a VPC:
  - Click Create VPC.
  - Set the following:
    - Name tag: MyVPC
    - IPv4 CIDR block: 10.0.0.0/16
  - Click Create VPC.

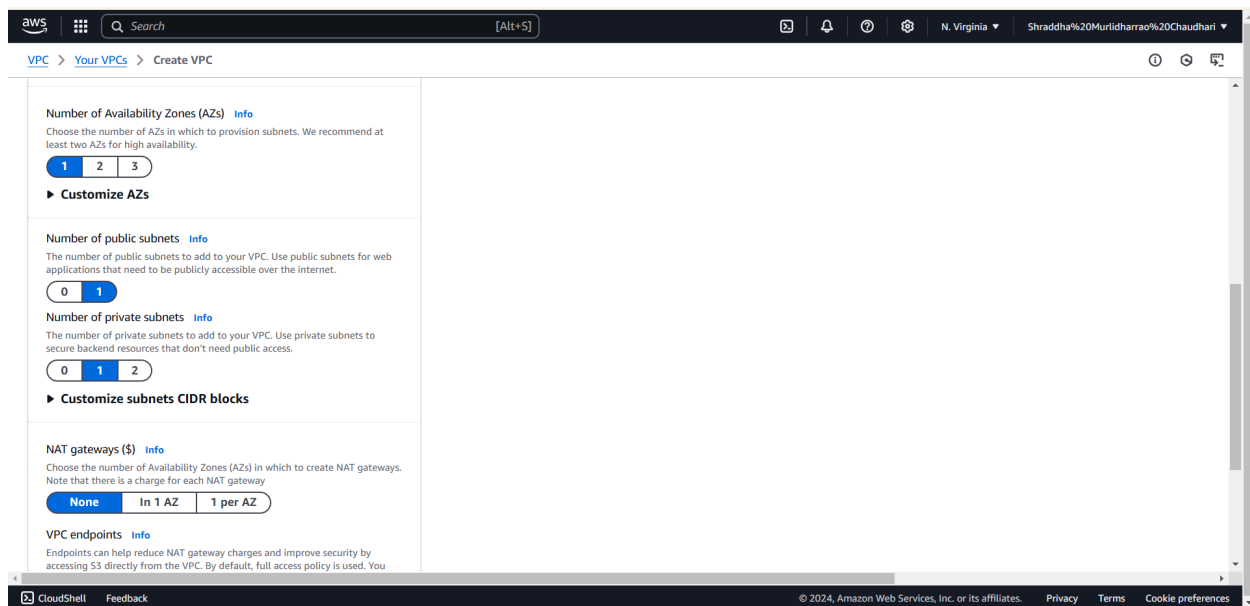
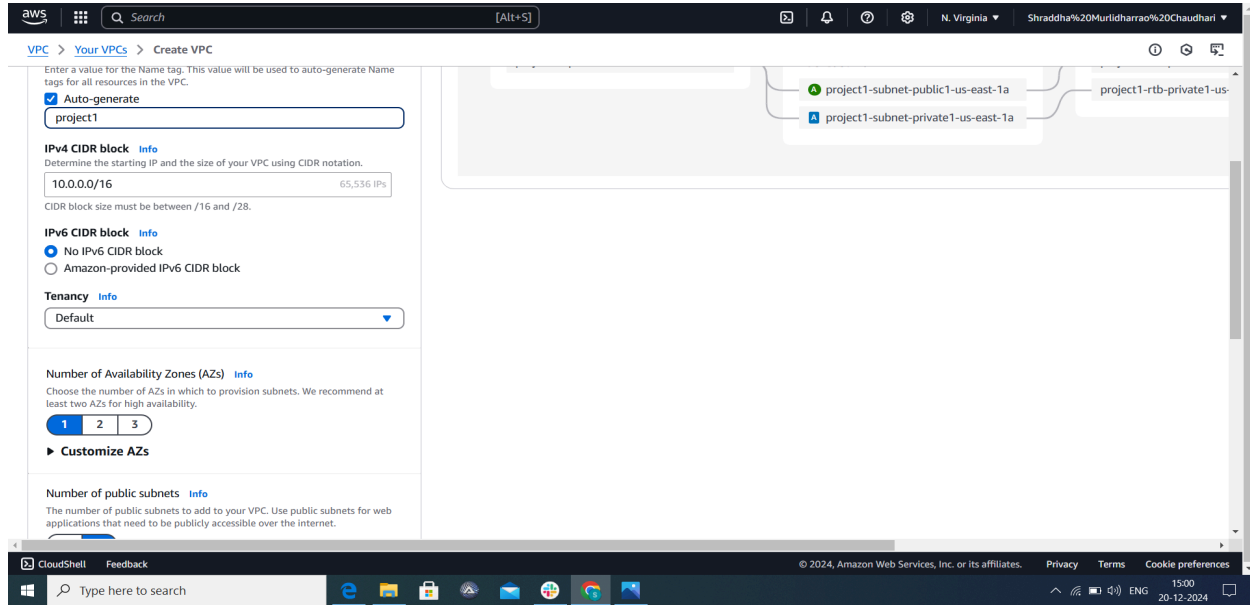
Create Subnets:

- Create a public subnet:
  - Name tag: PublicSubnet
  - CIDR block: 10.0.1.0/24
  - Associate it with MyVPC.
- Create a private subnet:
  - Name tag: PrivateSubnet
  - CIDR block: 10.0.2.0/24

- Associate it with MyVPC.

## Configure Internet Gateway:

- Attach an Internet gateway to MyVPC.
- Update the route table of PublicSubnet to route traffic to the Internet Gateway.



## Step 2: Launch Instances

### Launch Bastion Host in Public Subnet:

aws Search [Alt+S] N. Virginia Shraddha Murtidharao Chaudhari

EC2 > Instances > Launch an instance

Key pair name - *required*  
linkey [Create new key pair](#)

▼ Network settings [Info](#)

VPC - *required* [Info](#)  
vpc-0a608d644d49b253a (project1-vpc) [10.0.0.0/16](#) [Create new VPC](#)

Subnet [Info](#)  
subnet-001c201eb213cdadb project1-subnet-public1-us-east-1a  
VPC: vpc-0a608d644d49b253a Owner: 147506552856  
Availability Zone: us-east-1a Zone type: Availability Zone  
IP addresses available: 4091 CIDR: 10.0.0.0/20 [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

Security group name - *required*  
launch-wizard-6

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255

▼ Summary

Number of instances [Info](#)  
1

Software Image (AMI)  
Canonical, Ubuntu, 24.04, amd64...[read more](#)  
ami-0e2c8caa4b6378d8c

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

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

### Launch Web Server and Client in Private Subnet:

#### Web server:



aws

Search

[Alt+S]

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

web\_server

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

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Summary

Number of instances

Info

1

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...

read more

ami-0e2c8caa4b6378d8c

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, 750 hours of public IPv4

Cancel

Launch instance

Preview code

CloudShell

Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

Terms

Cookie preferences

Client:

aws

Search

[Alt+S]

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

client

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

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Summary

Number of instances

Info

2

When launching more than 1 instance, consider EC2 Auto Scaling

Software Image (AMI)

Canonical, Ubuntu, 24.04, amd6...

read more

ami-0e2c8caa4b6378d8c

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

Cancel

Launch instance

Preview code

CloudShell

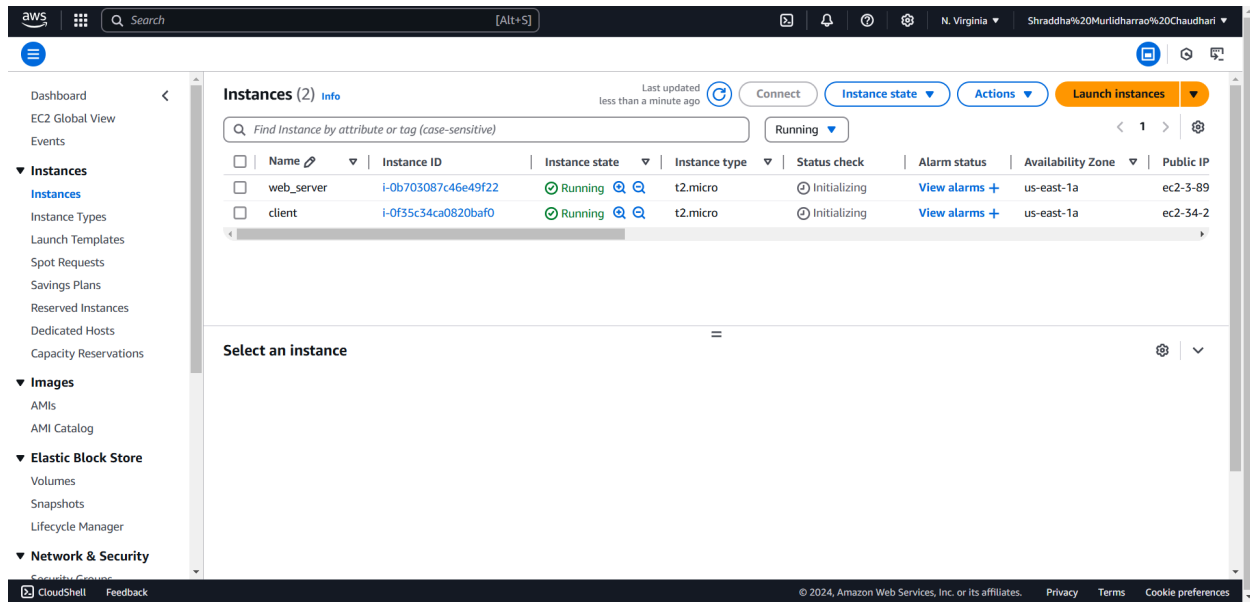
Feedback

© 2024, Amazon Web Services, Inc. or its affiliates.

Privacy

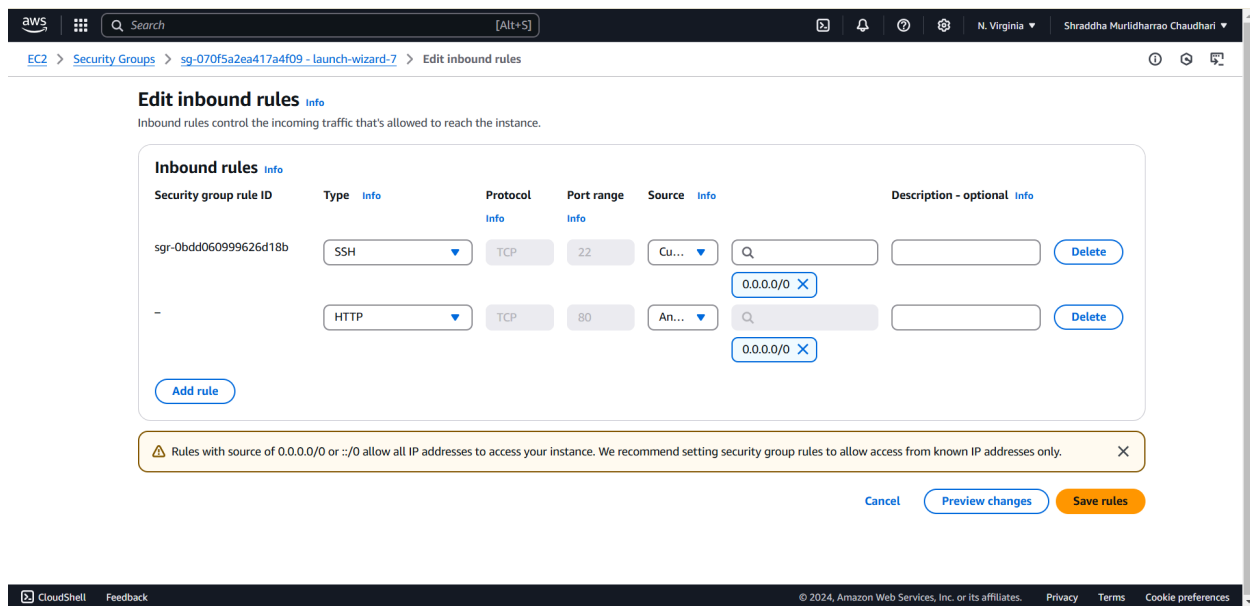
Terms

Cookie preferences



## Security Group:

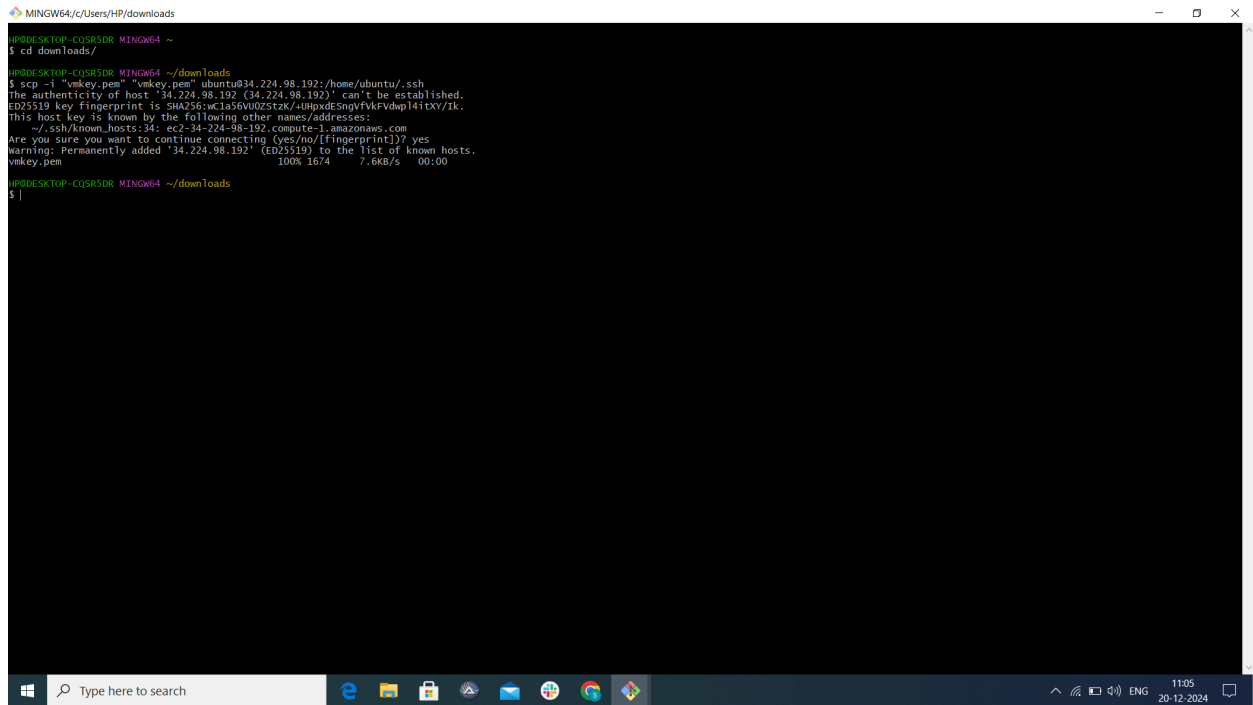
- Allow SSH (port 22) from the bastion host.
- Allow HTTP (port 80) within the private network.



## Step 3: Connect Bastion Host

## 1. Go to downloads and copy key pair from local to bastion host:

-> `scp -i <keypair.pem> <file name>  
ubuntu@<web-server-public-ip>:/home/ubuntu/.ssh`



```
MINGW64 ~/Downloads
$ cd downloads/

MINGW64 ~/Downloads
$ scp -i "vkkey.pem" vkkey.pem ubuntu@34.224.98.192:/home/ubuntu/.ssh
The authenticity of host '34.224.98.192 (34.224.98.192)' can't be established.
ED25519 key fingerprint is SHA256:wc1a56VU0Zstzk/4HpxdESngvFvkFvdwpl4itXY/1k.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:34: ec2-34-224-98-192.compute-1.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.224.98.192' (ED25519) to the list of known hosts.
vkkey.pem
100% 1674 7.6KB/s 00:00

MINGW64 ~/Downloads
$
```

## 2. SSH into Bastion Host:

→ `ssh -i <keypair.pem> ubuntu@<bastion-public-ip>`

```
ubuntu@ip-10-0-0-12: ~/ssh
HP@DESKTOP-CQSR3DR MINGW64 ~
$ cd downloads
HP@DESKTOP-CQSR3DR MINGW64 ~/downloads
$ ssh -i "vmkey.pem" ubuntu@ec2-34-224-98-192.compute-1.amazonaws.com
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Dec 20 05:51:10 UTC 2024

System load:  0.0          Processes:    119
Usage of /:   24.9% of 6.71GB   Users logged in: 1
Memory usage: 22%            IPv4 address for enx0: 10.0.0.12
Swap usage:   0%

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

Last login: Fri Dec 20 05:44:08 2024 from 123.252.234.15
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-0-0-12:~$ cd ssh
ubuntu@ip-10-0-0-12:~/ssh$ scp -i "vmkey.pem" "vmkey.pem" ubuntu@10.0.0.136:/home/ubuntu/.ssh
vmkey.pem
ubuntu@ip-10-0-0-12:~/ssh$ |
```

### 3. SSH into private instances from Bastion Host:

For Web Server:

-> `ssh -i <keypair.pem> ubuntu@<web-server-private-ip>`

```

ubuntu@ip-10-0-0-12:~/ssh$ ssh -i "vmkey.pem" ubuntu@10.0.0.136
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Dec 20 05:48:34 UTC 2024

System load:  0.0               Processes:    105
Usage of /:   24.9% of 6.71GB   Users logged in: 1
Memory usage: 21%              IPv4 address for enx0: 10.0.0.136
Swap usage:   0%

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
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Fri Dec 20 05:19:39 2024 from 10.0.0.12
ubuntu@ip-10-0-0-136:~$ pwd
/home/ubuntu
ubuntu@ip-10-0-0-136:~$ cd ~

```

- For client:

-> `ssh -i <keypair.pem> ubuntu@<client-private-ip>`

```

ubuntu@ip-10-0-0-135: ~
ubuntu@ip-10-0-0-12:~/ssh$ ssh -i "vmkey.pem" ubuntu@10.0.0.135
The authenticity of host '10.0.0.135 (10.0.0.135)' can't be established.
ED25519 key fingerprint is SHA256:2b8NpL10A7G1p2ssu685JX8krst36of186HzLu0NCP0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.135' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Fri Dec 20 05:49:29 UTC 2024

System load:  0.0               Processes:    104
Usage of /:   24.6% of 6.71GB   Users logged in: 0
Memory usage: 21%              IPv4 address for enx0: 10.0.0.135
Swap usage:   0%

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.

```

## Step 4: Transfer Files Using SCP

1. Create a test file on the web server:

-> `echo "Test File from Client" >grg.txt`

**2. Transfer file to client:** From the web server instance:

```
-> scp -i <keypair.pem> testfile.txt  
ubuntu@<web-server-private-ip>:/home/ubuntu
```

**3. Verify File on Web Server:** On the web server instance:

```
-> ls
```

Confirm **grg.txt** is present.

```
scp: connection closed  
ubuntu@ip-10-0-0-136:~$ cd .ssh/  
ubuntu@ip-10-0-0-136:~/ssh$ touch grg.txt  
ubuntu@ip-10-0-0-136:~/ssh$ scp -i "vmkey.pem" "grg.txt" ubuntu@10.0.0.135:/home/ubuntu  
grg.txt  
ubuntu@ip-10-0-0-136:~/ssh$ |
```

The VPC was successfully configured with public and private subnets. The bastion host allows secure SSH access to private instances, and a web server-client configuration was implemented with secure file transfer via SCP.

---