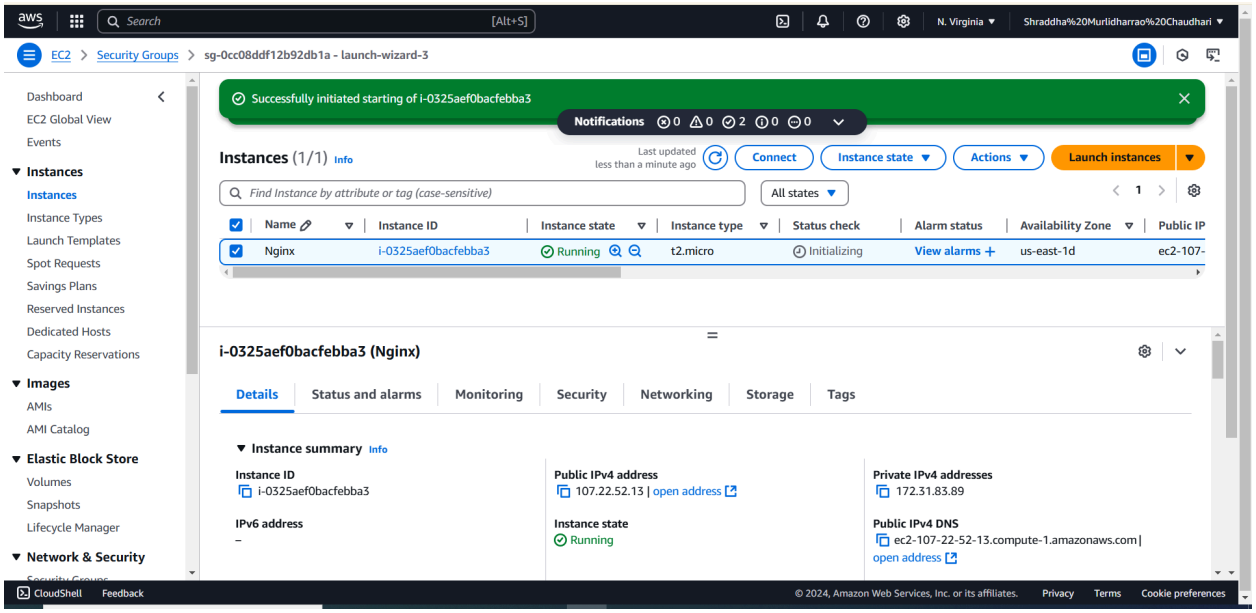
Documentations:

1. Setting Up Nginx with a Custom 404 Error Page

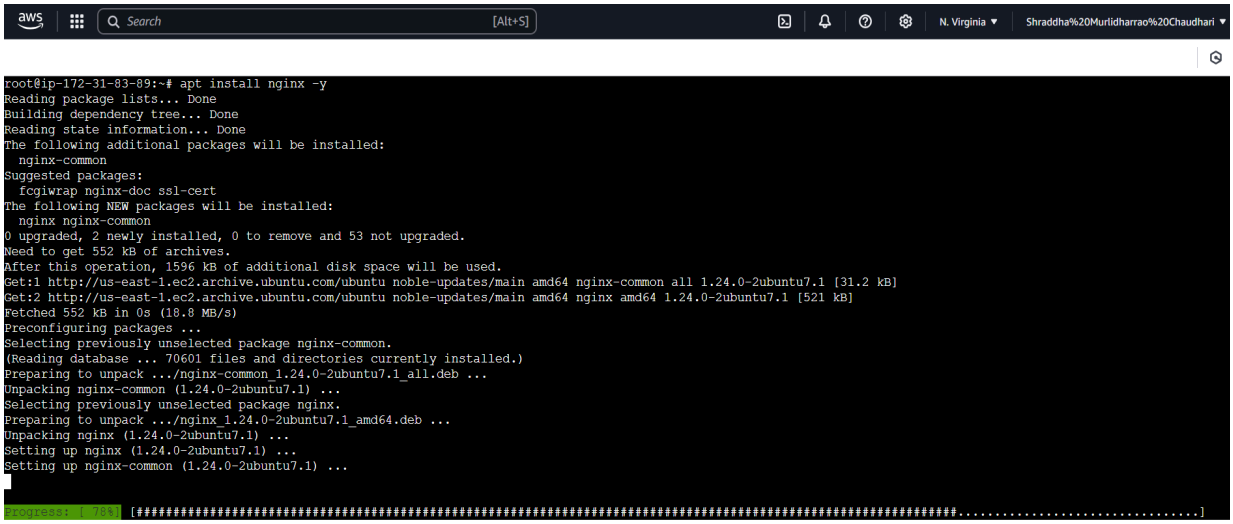
This document outlines the step-by-step process of configuring an Nginx server to display a custom 404 error page when a user accesses a non-existent route.

Step 1: Launch the Instance



Step 2: Install Nginx

-> apt update
-> apt install nginx -y

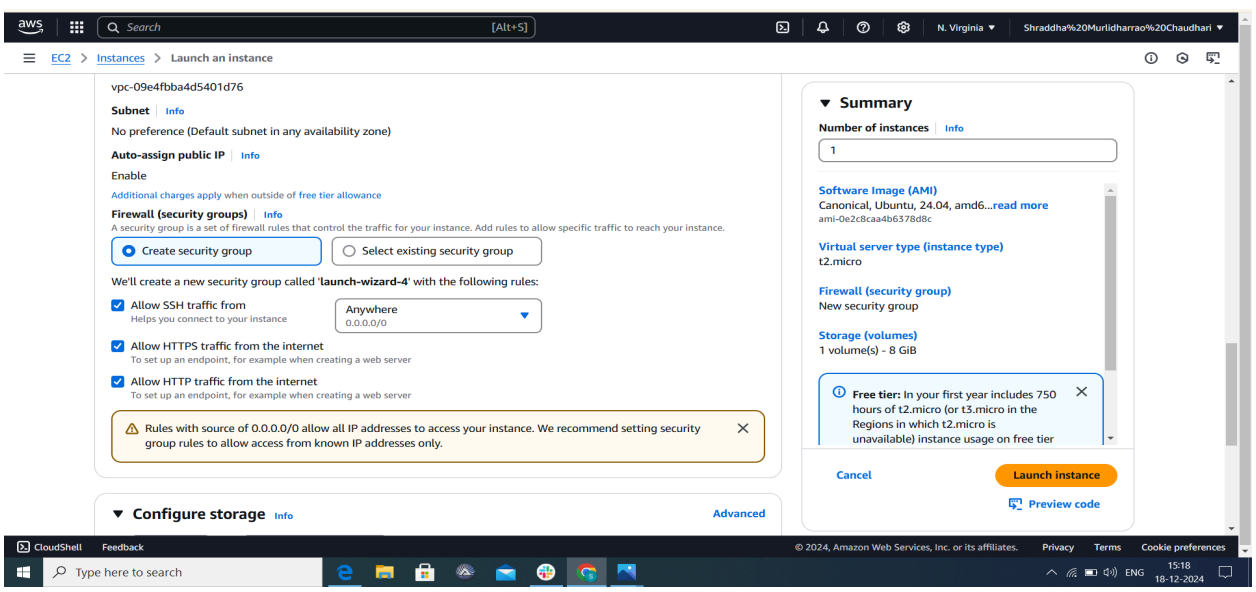


Verify that Nginx is installed and running:

-> 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

```
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

```
<!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

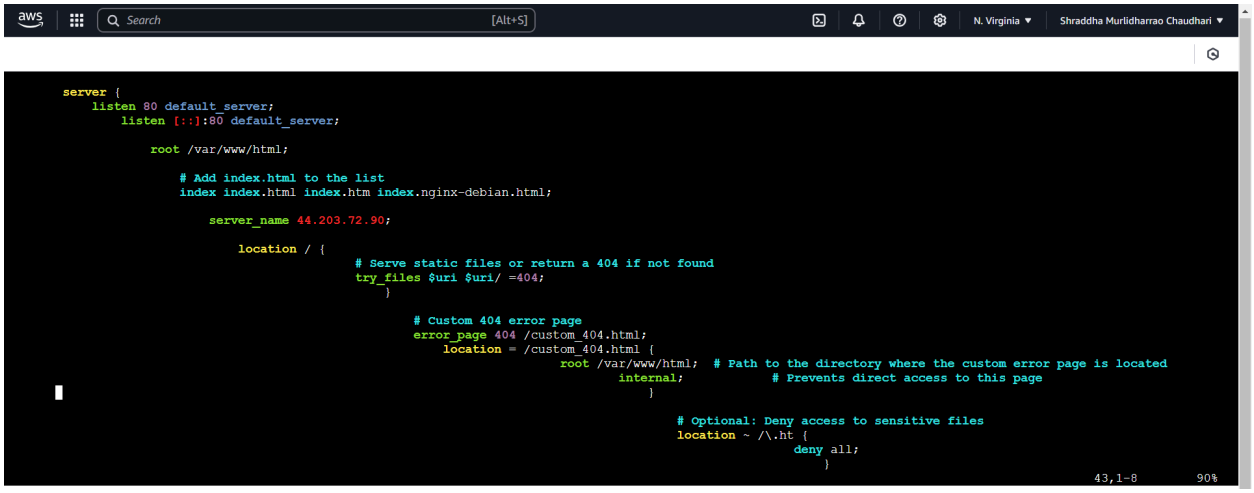
Verify the file exists:

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

<u>Step 5: Update the Nginx Configuration:</u>

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

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

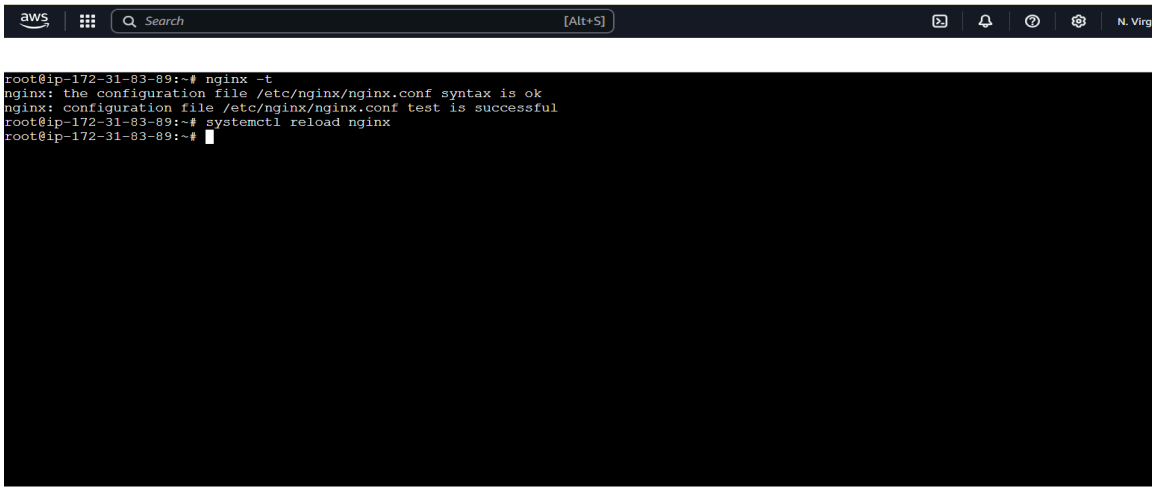


Save and close the file

Step 6: Test the Configuration:

Validate the Nginx Configuration:

-> sudo nginx -t



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.

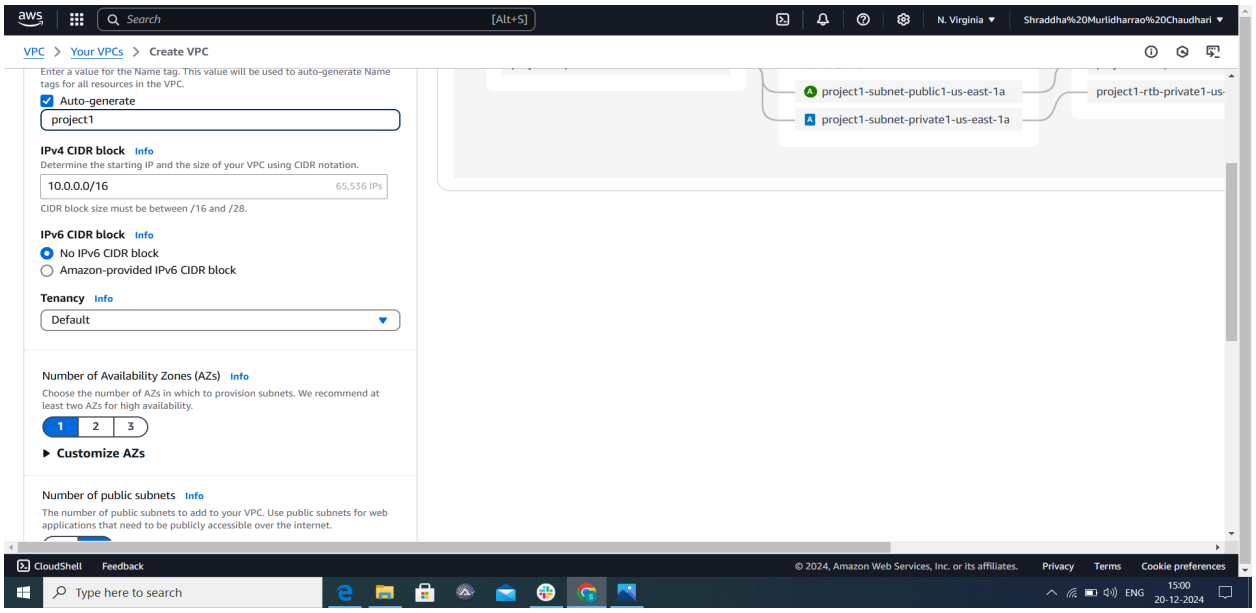
Create Subnets:

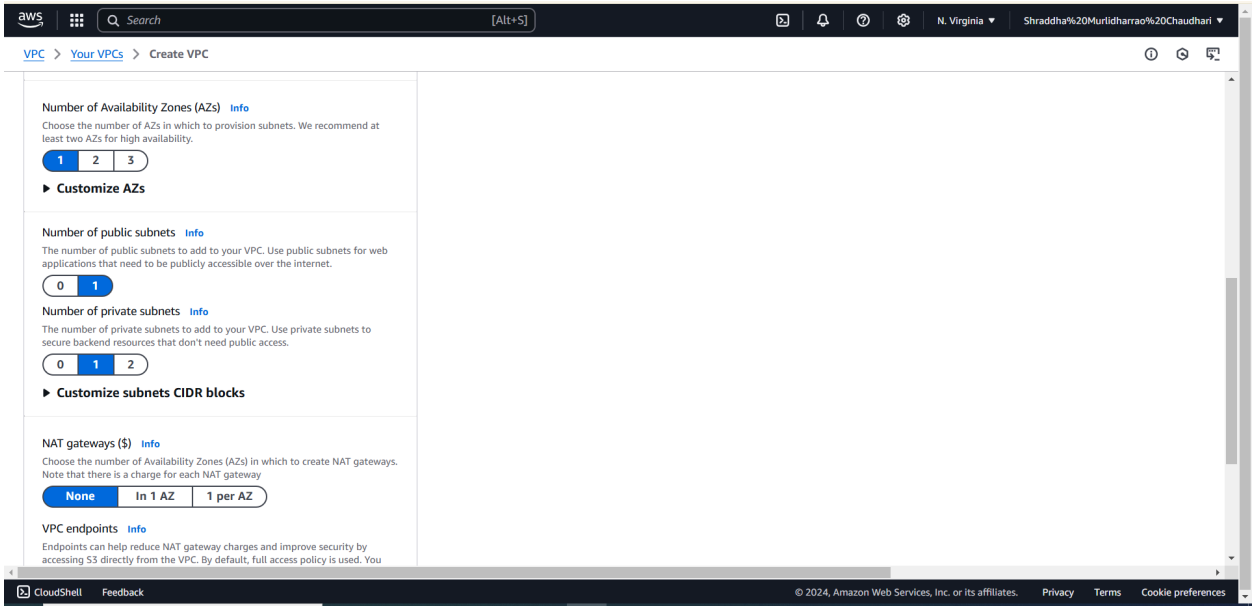
- Create a public subnet:
    - Name tag: PublicSubnet
    - CIDR block: 10.0.1.0/24
    - Associate it with MyVPC.
- Create a private subnet:
    - Name tag: PrivateSubnet
    - CIDR block: 10.0.2.0/24

- Associate it with MyVPC.

Configure Internet Gateway:

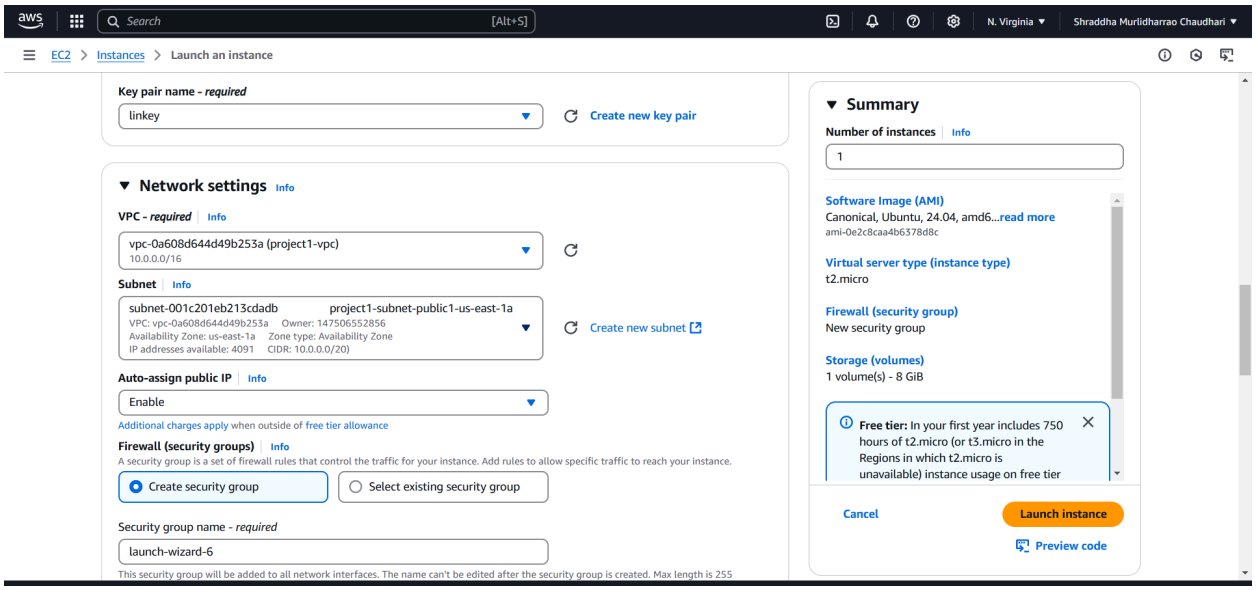
- Attach an Internet gateway to MyVPC.
- Update the route table of PublicSubnet to route traffic to the Internet Gateway.





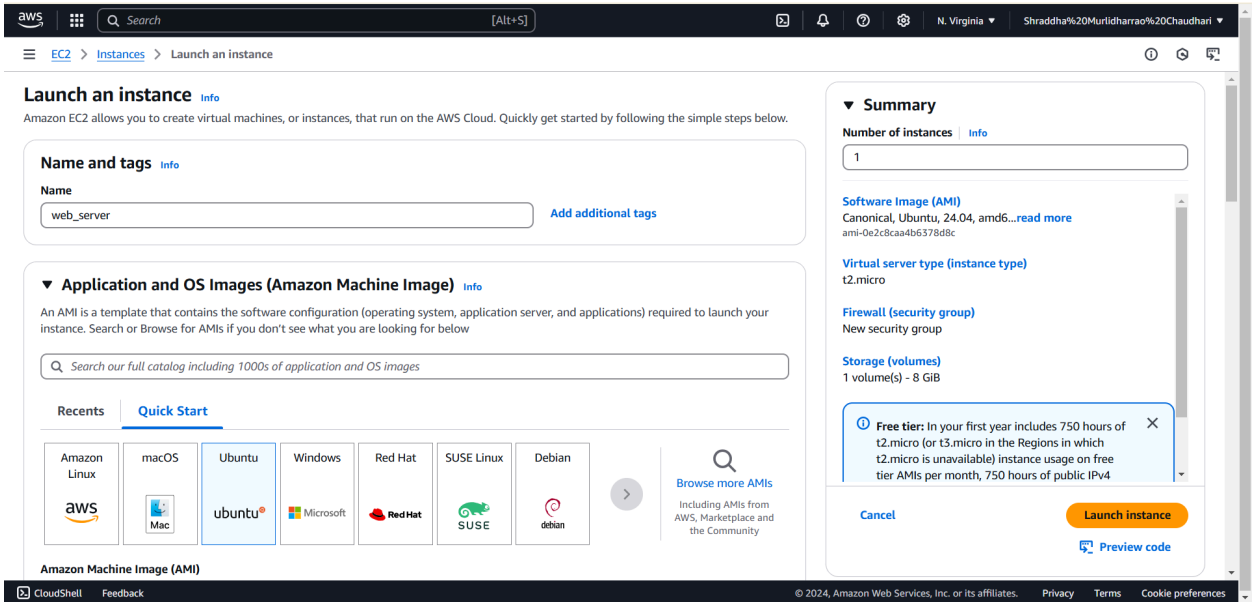
Step 2: Launch Instances

Launch Bastion Host in Public Subnet:

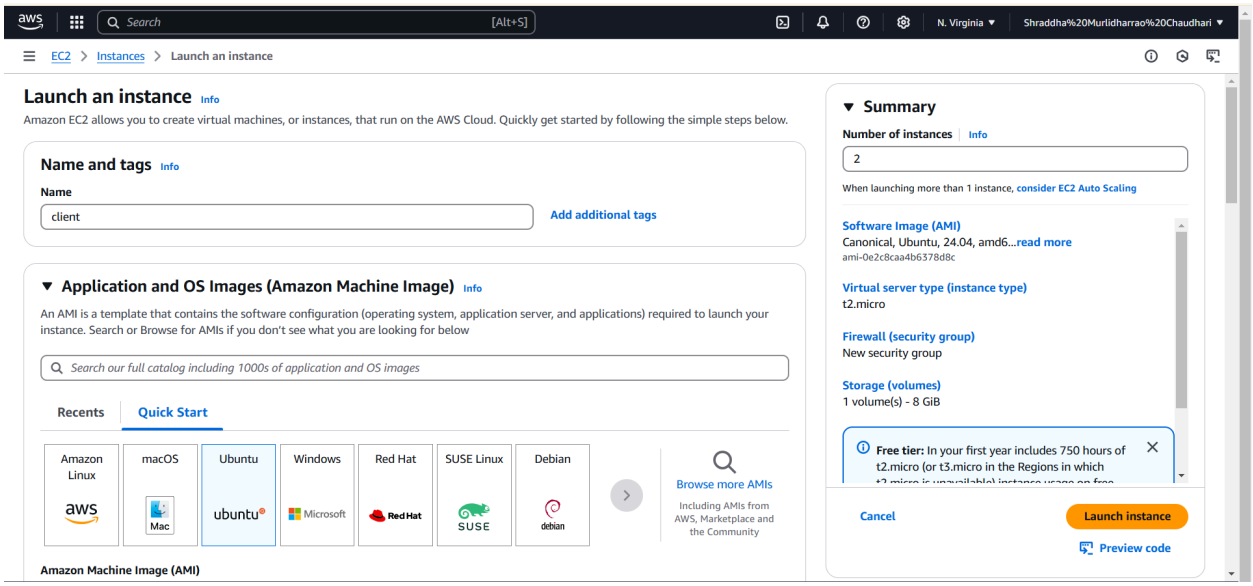


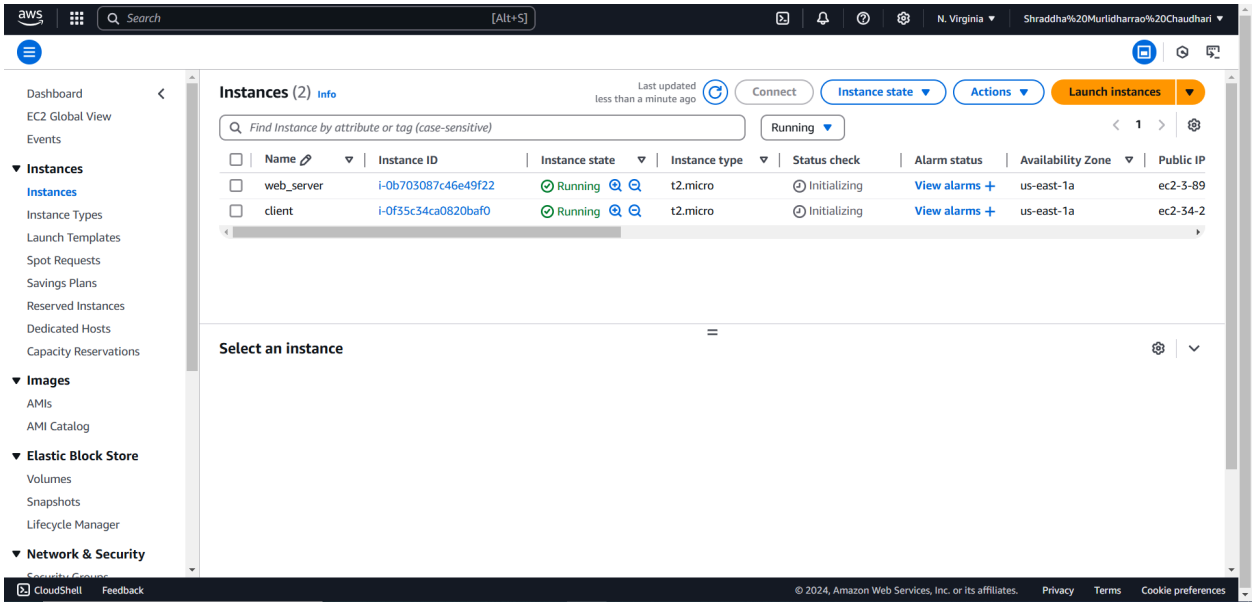
Launch Web Server and Client in Private Subnet:

Web server:


aws
Search
[Alt+S]
N. Virginia
Shraddha%20Murlidharrao%20Chaudhari
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
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
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

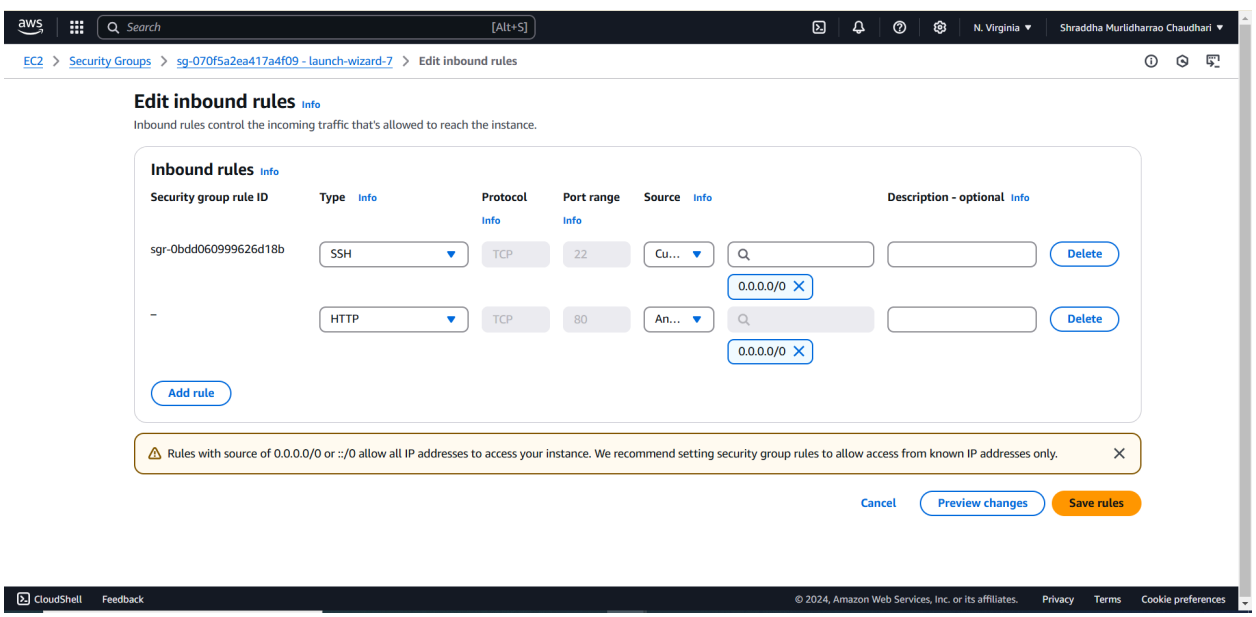
Client:


aws
Search
[Alt+S]
N. Virginia
Shraddha%20Murlidharrao%20Chaudhari
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
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
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
Cancel
Launch instance
Preview code



Security Group:

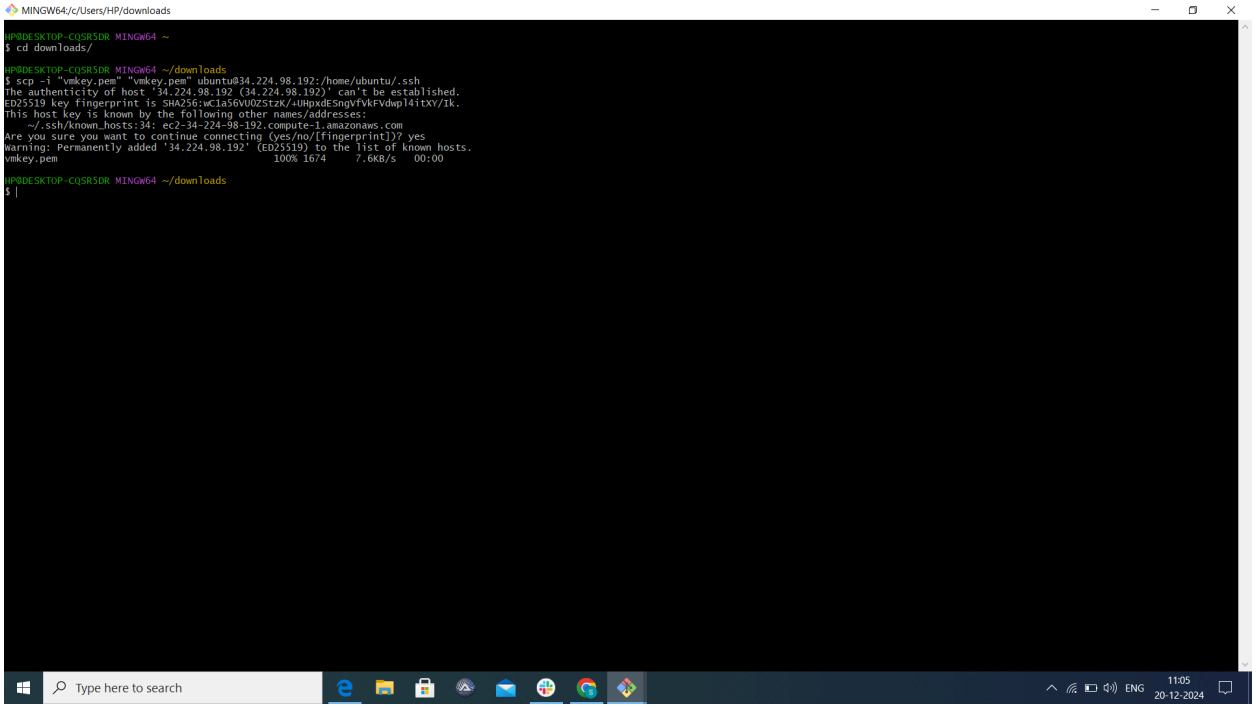
- Allow SSH (port 22) from the bastion host.
- Allow HTTP (port 80) within the private network.



Step 3: Connect Bastion Host

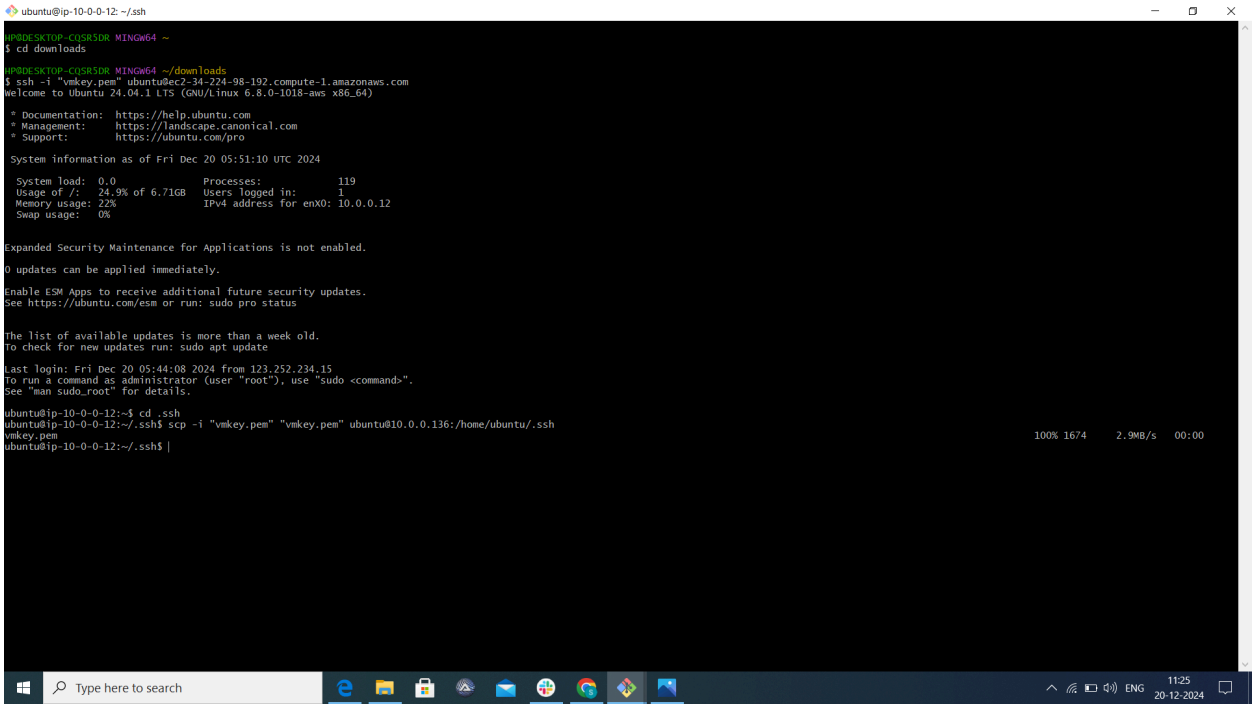
1. **Go to downloads and copy key pair from local to bastion** host:

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



2. **SSH into Bastion Host:**

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



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
```

- 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 SHA256:2bRMpl1OA76lPz5su68SJKRkr5t36oF186Hz1uoNCP0.
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.

```
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.
