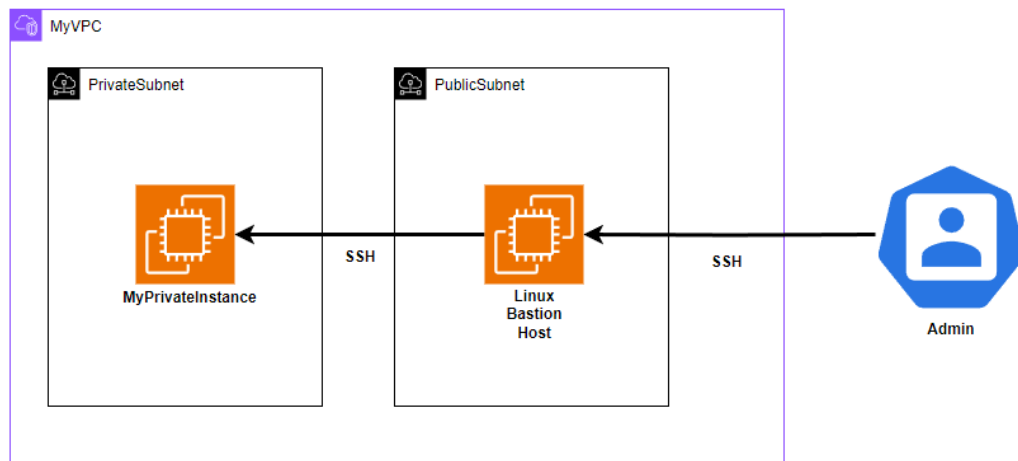


Guided Lab: Setting Up a Linux Bastion Host on AWS

Description

A Bastion host plays a vital role in securely accessing instances within private subnets, serving as a gateway for controlling resource management.

This guided lab will walk you through setting up a Bastion host using a Linux instance on AWS. By leveraging a Bastion host, administrators can securely manage private instances without exposing them to the internet, significantly enhancing the security posture of your AWS environment.



Prerequisites

This lab assumes you have basic knowledge of Amazon EC2 Instance, VPCs & basic network configuration in AWS Cloud.

If you find any gaps in your knowledge, consider taking the following lab:

- Creating an Amazon EC2 instance (Linux)
- Creating a Custom Virtual Private Cloud (VPC) from scratch

Objectives

By the end of this lab, participants will be able to:

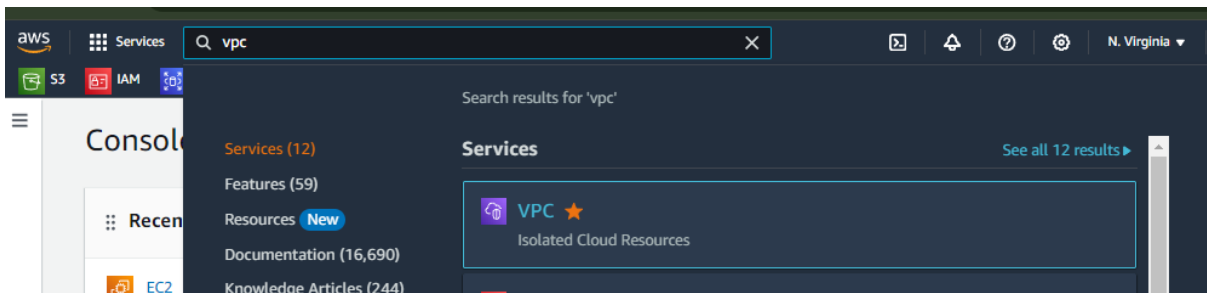
- Understand the concept and purpose of a Bastion host.
- Launch and configure a Linux Bastion host in the public subnet
- Access instances in the private subnet securely through the Bastion host.

Lab Steps

Setup Network Configurations

1. Create a VPC:

- Go to the VPC dashboard in the AWS Management Console.



- Click on Create VPC.
- Resource to create: VPC only
- Name: MyVPC
- CIDR block: 10.0.0.0/16
- Tenancy: Default
- Click Create.

VPC > Your VPCs > Create VPC

Create VPC [Info](#)

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only
 ☐ VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

MyVPC

IPv4 CIDR block [Info](#)

☒ IPv4 CIDR manual input
 ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.0.0.0/16

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

☒ No IPv6 CIDR block
 ☐ IPAM-allocated IPv6 CIDR block
 ☐ Amazon-provided IPv6 CIDR block
 ☐ IPv6 CIDR owned by me

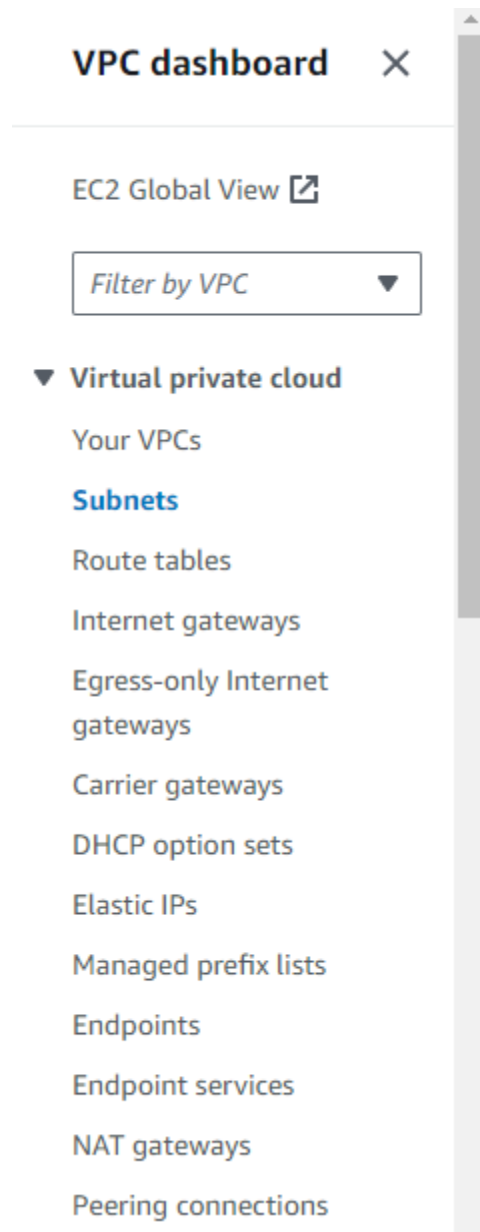
Tenancy [Info](#)

Default

Creating a VPC establishes an isolated network within AWS where you can launch AWS resources.

2. Create Subnets:

- Navigate to Subnets in th VPC dashboard.



- Create two subnets with the following configurations:
 - **Public Subnet:**
 - VPC ID: MyVPC
 - Subnet name: PublicSubnet
 - Availability Zone: Select one from the list (e.g. us-east-1a)
 - IPv4 subnet CIDR block: 10.0.1.0/24

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

PublicSubnet

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block [Info](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.1.0/24

256 IPs

< > ^ v

▼ Tags - optional

Key

Value - optional

Q Name

X

Q

PublicSubnet

X

Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel

Create subnet

- - - Add new subnet
 - Private Subnet: (Click on Add new subnet)
 - Name: PrivateSubnet
 - IPv4 subnet CIDR block: 10.0.2.0/24
 - Availability Zone: Same as the public subnet
 - Click Create.

Subnet 2 of 2

Subnet name

Create a tag with a key of 'Name' and a value that you specify.

PrivateSubnet

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block [Info](#)

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.2.0/24256 IPs

< > ^ v

▼ Tags - optional

Key

Value - optional

Q NameX

Q PrivateSubnetX

Remove

Add new tag

You can add 49 more tags.

Remove

Add new subnet

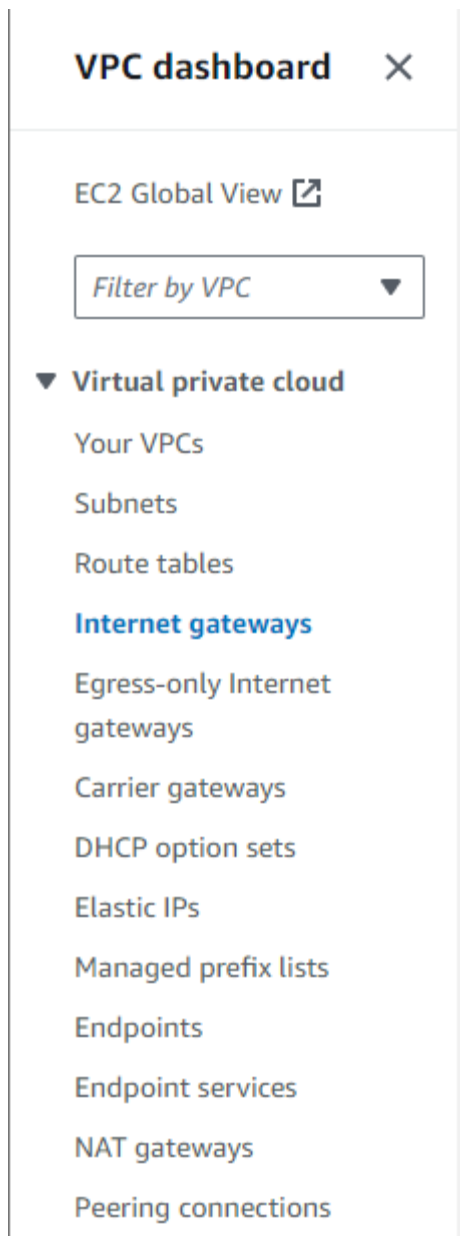
Cancel

Create subnet

Subnets segment the VPC network. Public subnets allow direct access to the internet, while private subnets do not.

3. Create an Internet Gateway:

- Go to Internet Gateways in the VPC dashboard.



- Click **Create internet gateway**.
- Name: MyIGW
- Click Create.
- Attach the Internet Gateway to MyVPC.

Internet gateway igw-0acc3bad30eae2d4c successfully attached to vpc-0990402021126b1ad

Notifications 0 0 2 0 0

VPC > Internet gateways > igw-0acc3bad30eae2d4c

igw-0acc3bad30eae2d4c / MyIGW

Actions

Details Info

Internet gateway ID
igw-0acc3bad30eae2d4c

State
Attached

VPC ID
vpc-0990402021126b1ad
MyVPC

Owner
134564086665

Tags

Manage tags

Search tags

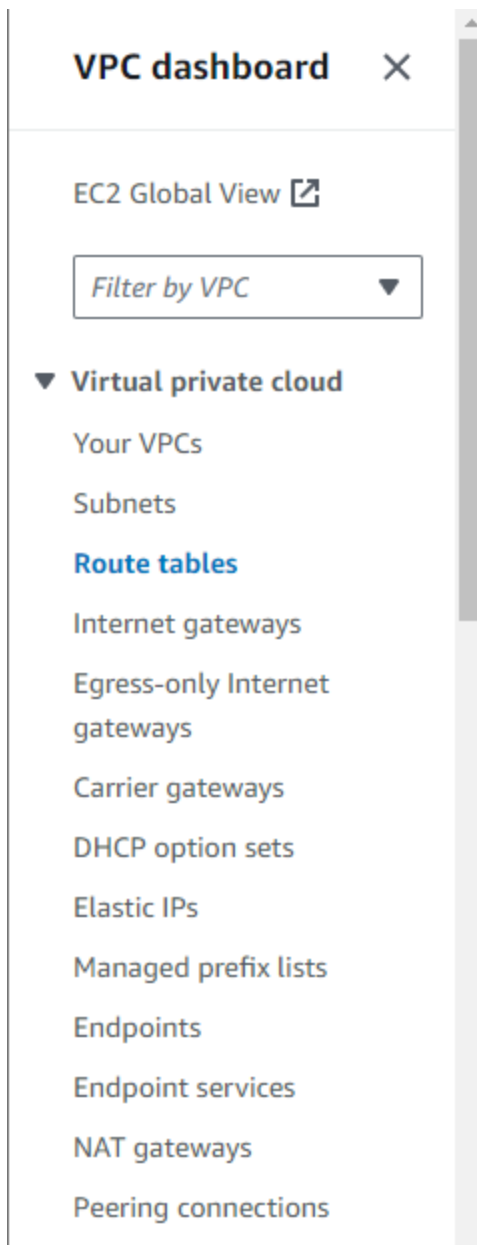
< 1 > ⚙

Key	Value
Name	MyIGW

4. Create a new Route Table for the Public Subnet

Take note that there should be a new Route Table created after creating a new VPC.

- Go to the Route Tables section in the VPC dashboard



- Click **Create route table**
 - Name: PublicRT
 - VPC: MyVPC
- Click **Create route table**

[VPC](#) > [Route tables](#) > rtb-0a5abc07f760051b0

rtb-0a5abc07f760051b0 / PublicRT

Actions ▼

Details [Info](#)

Route table ID
rtb-0a5abc07f760051b0

VPC
vpc-0990402021126b1ad
[MyVPC](#)

Main
No

Owner ID
134564086665

Explicit subnet
associations
–

Edge associations
–

Routes

Subnet associations

Edge associations

Route propagation

Tags

Routes (1)

Both ▼

Edit routes

< 1 > ⚙️

Destination ▼	Target ▼	Status ▼	Propagated ▼
10.0.0.0/16	local	Active ✔️	No

- Edit routes:
 - Add route 0.0.0.0/0 to target MyIGW.

Edit routes

Route 1

Destination

10.0.0.0/16

Target

local

Status

✓ Active

Q local X

Propagated

No

Route 2

Destination

Q 0.0.0.0/0 X

Target

Internet Gateway

Status

-

Q igw- X

Use: "igw-"

igw-0acc3bad30eae2d4c (MyIGW)

Propagated

No

Remove

Add route

Cancel

Preview

Save changes

- Save changes.
- Edit **Explicit subnet associations** in the **Subnet associations tab**: Associate PublicSubnet.

You have successfully updated subnet associations for rtb-0a5abc07f760051b0 / PublicRT.

VPC > Route tables > rtb-0a5abc07f760051b0

rtb-0a5abc07f760051b0 / PublicRT

Actions

Details

Info

Route table ID

rtb-0a5abc07f760051b0

VPC

vpc-0990402021126b1ad

MyVPC

Main

No

Owner ID

134564086665

Explicit subnet associations

subnet-09fff4db6d21f83e1 / PublicSubnet

Edge associations

-

Routes

Subnet associations

Edge associations

Route propagation

Tags

Explicit subnet associations (1)

Edit subnet associations

Find subnet association

< 1 >

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
PublicSubnet	subnet-09fff4db6d21f83e1	10.0.1.0/24	-

Subnets without explicit associations (1)

Edit subnet associations

Find subnet association

< 1 >

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
PrivateSubnet	subnet-0b7b659ece0b7e...	10.0.2.0/24	-

5. Create another Route Table for Private Subnet

- Create another route table: Name: PrivateRouteTable, VPC: MyVPC.
- Edit Explicit subnet associations in the **Subnet associations tab**: Associate PrivateSubnet.

[VPC](#) > [Route tables](#) > rtb-099170f58e90c5066

rtb-099170f58e90c5066 / PrivateRouteTable

Actions ▾

Details [Info](#)

Route table ID
rtb-099170f58e90c5066

VPC
vpc-0990402021126b1ad
| MyVPC

Main
No

Owner ID
134564086665

Explicit subnet associations
[subnet-0b7b659ece0b7ecda](#) / [PrivateSubnet](#)

Edge associations
-

[Routes](#) | [Subnet associations](#) | [Edge associations](#) | [Route propagation](#) | [Tags](#)

Explicit subnet associations (1)

Edit subnet associations

<input type="text" value="Find subnet association"/>				< 1 >		⚙
Name ▾	Subnet ID ▾	IPv4 CIDR ▾	IPv6 CIDR ▾			
PrivateSubnet	subnet-0b7b659ece0b7...	10.0.2.0/24	-			

Subnets without explicit associations (0)

Edit subnet associations

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

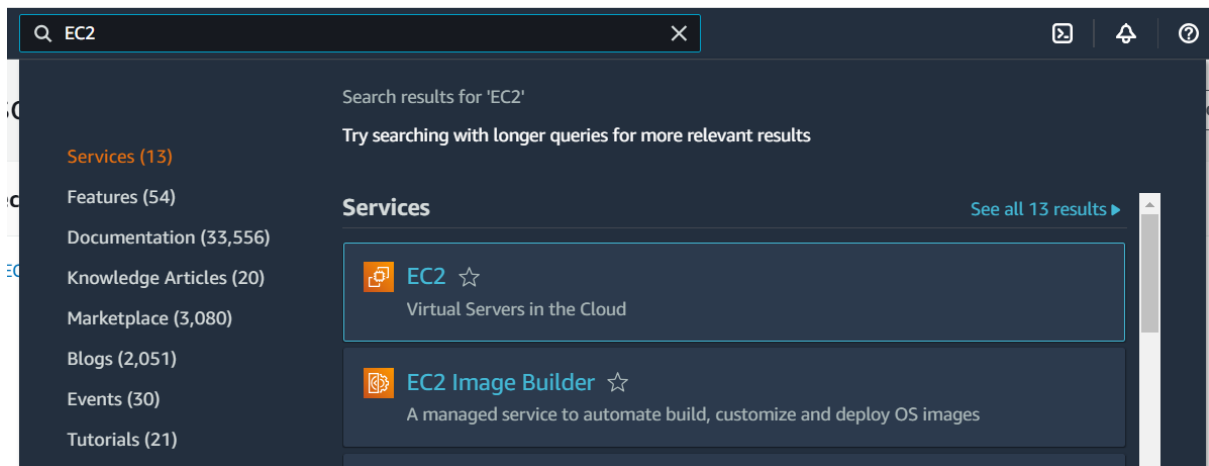
<input type="text" value="Find subnet association"/>				< 1 >		⚙
Name ▾	Subnet ID ▾	IPv4 CIDR ▾	IPv6 CIDR ▾			

No subnets without explicit associations
All your subnets are associated with a route table.

Launch Instances

1. Launch an EC2 Instance in the Public Subnet:

- Go to the EC2 dashboard.




- Click Launch Instance.
- Name: LinuxBastionHost
- AMI: Amazon Linux 2023 AMI
- Instance Type: t2.micro
- Create key pair: MyKeyPair
- Network Settings:
 - VPC: MyVPC,
 - Subnet: PublicSubnet
- Enable Auto-assign Public IP.

▼ 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*

MyKeyPair

 [Create new key pair](#)

▼ Network settings [Info](#)

VPC - *required* [Info](#)

vpc-0990402021126b1ad (MyVPC)
10.0.0.0/16



Subnet [Info](#)

subnet-09fff4db6d21f83e1

PublicSubnet

VPC: vpc-0990402021126b1ad Owner: 134564086665
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 10.0.1.0/24

 [Create new subnet](#) 

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

- Firewall (security groups):
 - Select **Create security group** Security group name: BastionSG
 - Description – *required*: Security Group for LinuxBastionHost

Security group name - *required*

BastionSG

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 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!\$*

Description - *required* [Info](#)

Security Group for LinuxBastionHost

- Configure Inbound Security Group rules: Allow SSH (port **22**) from your IP

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 119.111.228.217/32)

[Remove](#)

Type [Info](#)

ssh

Protocol [Info](#)

TCP


Port range [Info](#)

22

Source type [Info](#)

My IP

Name [Info](#)

 Add CIDR, prefix list or security

119.111.228.217/32 

Description - *optional* [Info](#)

e.g. SSH for admin desktop

- Review and Launch.

2. Launch another EC2 Instance in the Private Subnet:

- Name: MyPrivateInstance
- Follow the same steps, but
 - select PrivateSubnet
 - Key pair: use the same key pair MyKeyPair
 - Network Settings:
 - VPC: MyVPC
 - Subnet: PrivateSubnet
 - Disable Auto-assign Public IP.

▼
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*

MyKeyPair
▼

[Create new key pair](#)

▼
Network settings
Info

VPC - *required* | Info

vpc-0990402021126b1ad (MyVPC)
10.0.0.0/16
▼

Subnet | Info

subnet-0b7b659ece0b7ecda
PrivateSubnet
▼

VPC: vpc-0990402021126b1ad Owner: 134564086665
Availability Zone: us-east-1a IP addresses available: 251 CIDR: 10.0.2.0/24

[Create new subnet](#)

Auto-assign public IP | Info

Disable
▼

- Firewall (security groups): Select **Create security group**
 - Security group name: MyPrivateInstanceSG
 - Description – *required*: **Security group for Private Instance**

Security group name - *required*

MyPrivateInstanceSG

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 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-./()#,@[]+=&;()!\$*

Description - *required* [Info](#)

Security group for the Private Instance

- Inbound Security Group Rules: Allow SSH (port 22) from BastionSG.

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, sg-00a30c453e4e78a14)

Type [Info](#)

ssh

Source type [Info](#)

Custom

Add security group rule

Security groups

com.amazonaws.us-east-1.s3express
pl-d928c6b0

default
sg-07c5d43c9b36707

BastionSG
sg-00a30c453e4e78a14

Q Add CIDR, prefix list or security

sg-00a30c453e4e78a14 X

e.g. SSH for admin desktop

- Review and Launch.

Security groups control the inbound and outbound traffic to instances. The Bastion host must have SSH access to the private instance.

Access the PrivateInstance

1. Open up your terminal (e.g., Git Bash, PuTTY).
2. Firsts, we need to copy the Private Key (MyKeyPair.pem) to the NAT Instance in our local machine by:

- Going to directory of your .pem key

```
neil@Sol MINGW64 ~
$ cd Downloads/

neil@Sol MINGW64 ~/Downloads
$ |
```

- Copy, edit, and paste the following command accordingly:

Do not forget to change the <your_NAT_Instance_Key_Pair>, <Your_Private_Instance_Key_Pair>, and <public_IP_address_of_NAT_Instance> placeholders with the right value
If confirmation pops up in your terminal, type yes and ENTER


```
neil@Sol MINGW64 ~/Downloads
$ scp -i MyKeyPair.pem MyKeyPair.pem ec2-user@44.200.61.173:/home/ec2-user/
The authenticity of host '44.200.61.173 (44.200.61.173)' can't be established.
ED25519 key fingerprint is SHA256:uC8naihPonhF5NiA9Te4VMfix69Z5xP9bxipruchF8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '44.200.61.173' (ED25519) to the list of known hosts.
MyKeyPair.pem                                100% 1674      7.6KB/s   00:00

neil@Sol MINGW64 ~/Downloads
$ .....
```

```
ssh -i your-key-pair.pem ec2-user@<NATInstance-Public-IP>
```

```
neil@Sol MINGW64 ~/Downloads
$ ssh -i "MyKeyPair.pem" ec2-user@44.200.61.173
```

The terminal displays the following ASCII art:

```
      ,_
     ~\   _###_          Amazon Linux 2023
    ~~~ \_#####\
         \|####|
         \|##/\_____ https://aws.amazon.com/linux/amazon-linux-2023
           V~' '->
             ^
            / \
           ._.
          /_/
         /m/'
```

```
[ec2-user@ip-10-0-1-235 ~]$
```

```
[ec2-user@ip-10-0-1-235 ~]$ ls -l
total 4
-rw-r--r--. 1 ec2-user ec2-user 1674 Aug  6 06:22 MyKeyPair.pem
[ec2-user@ip-10-0-1-235 ~]$
```

```
sudo chmod 400 MyKeyPair.pem
```

```
[ec2-user@ip-10-0-1-235 ~]$ sudo chmod 400 MyKeyPair.pem
[ec2-user@ip-10-0-1-235 ~]$
```

```
ssh -i MyKeyPair.pem ec2-user@<PrivateInstance-Private-IP>
```

A prompt confirmation will pop to the terminal, answer yes

