# 

# **Record SSH Sessions Established Through a Bastion Host**

**Let’s the Bastion Journey begins**



# **Contents**

1. High-level Description
2. Benefits/Necessity of Bastion Host
3. Demo of Bastion Solution
4. Low-level (deeper level) Architecture
5. Things I learnt

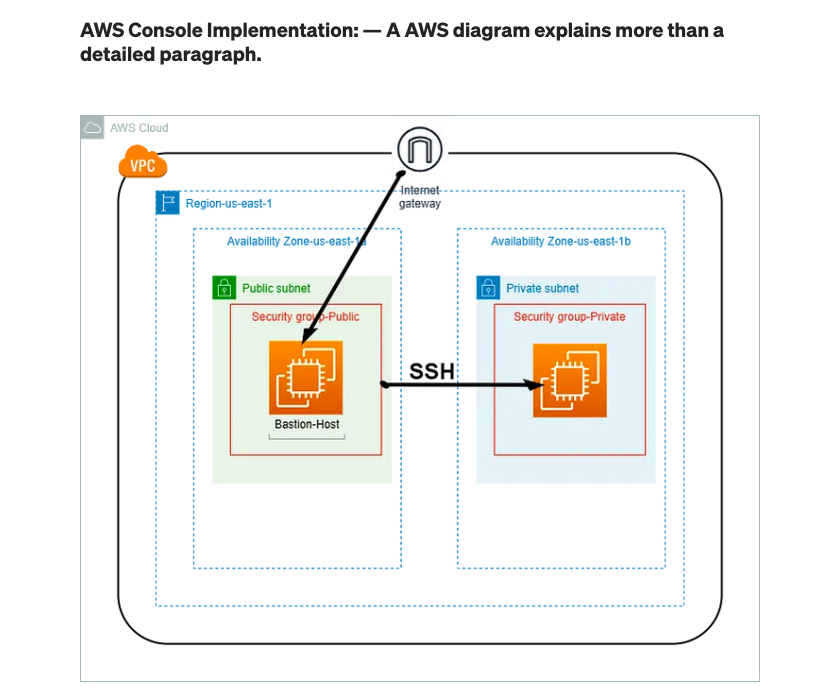
**High-level Description**

**What is a Bastion host?**

Bastion host or the jump server is an entry point to the actual server in the private network (or private subnet). In this way, bastion host provides an additional layer of protection to the actual server from any external harmful actors on the internet.

The Bastion host is in a public subnet, so using Bastion host, we can access private EC2 instance within same VPC by SSH connection.

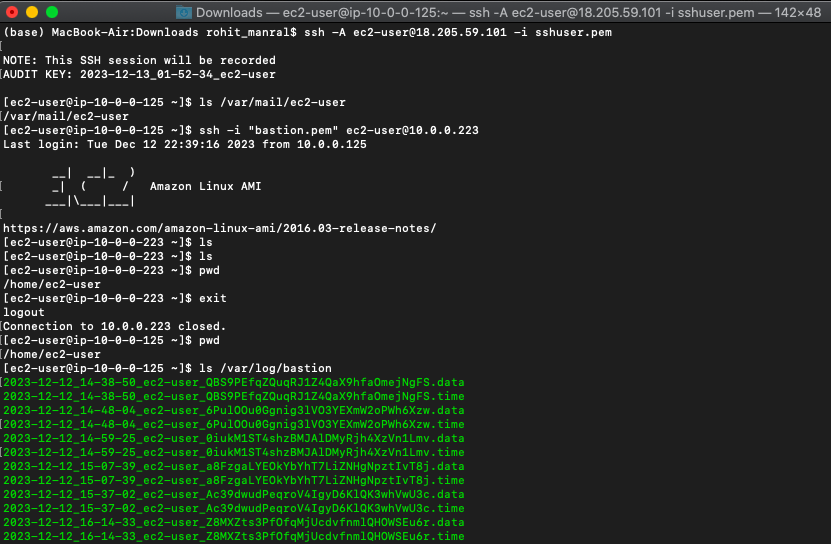
Let's see how to implement this with following diagram:

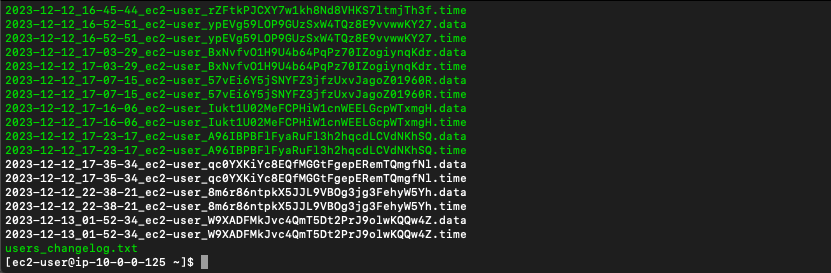


**Figure: SSH into a private EC2 instance through Bastion host**

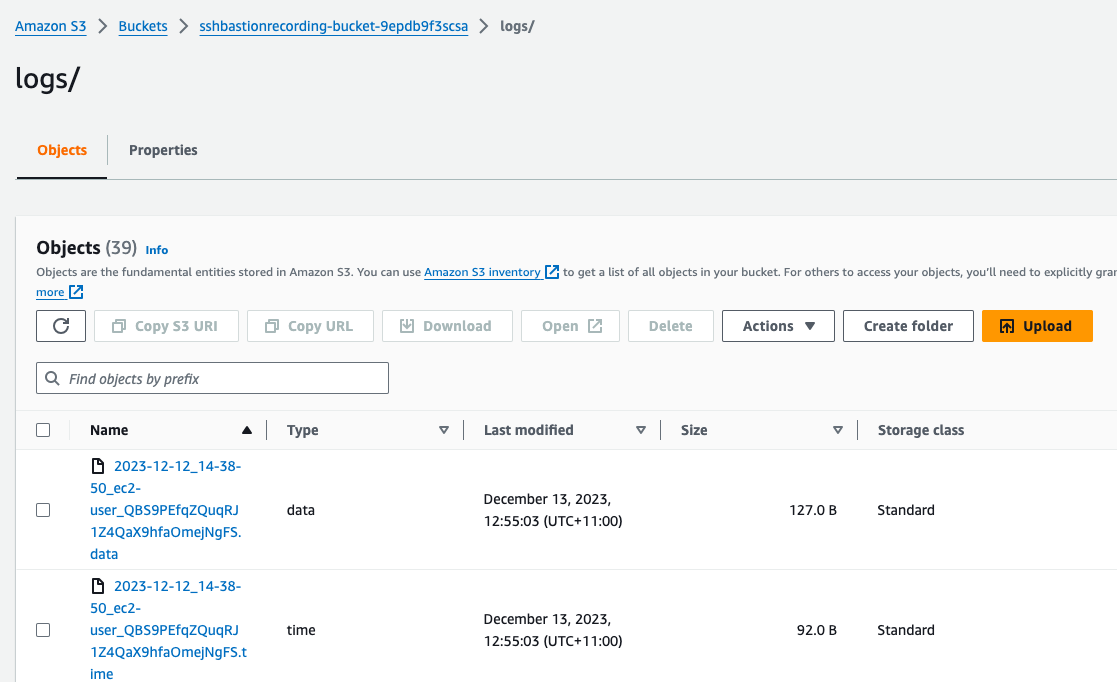
**Results of SSH sessions recorded by Bastion Host?**

1. **Terminal**

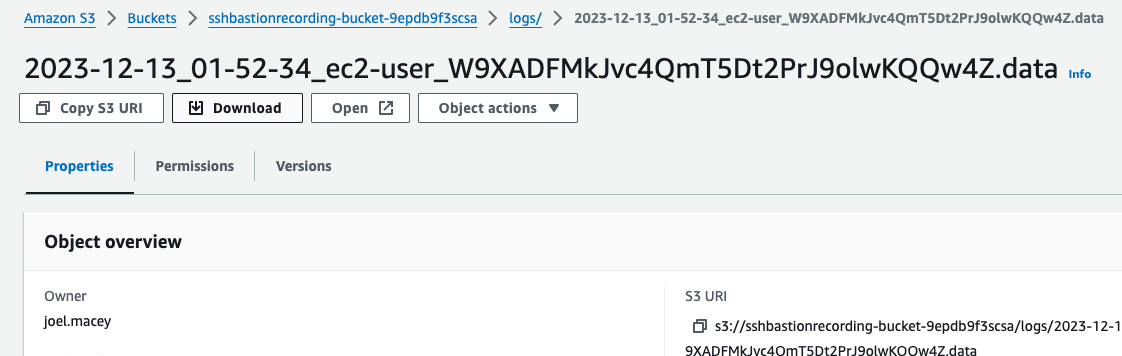




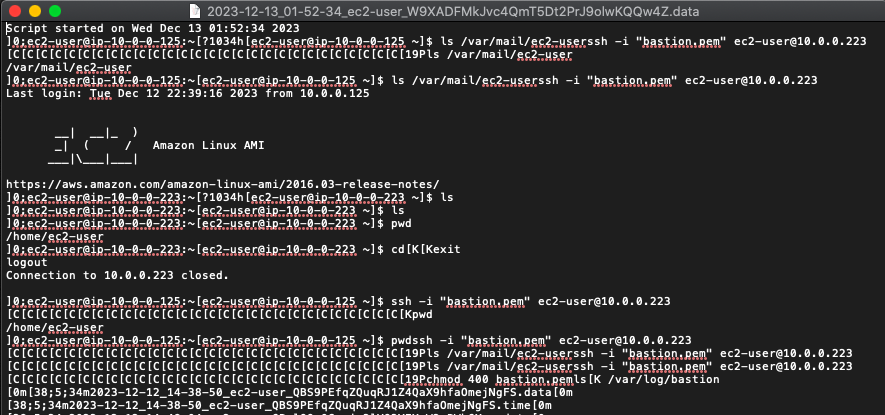
1. **Log activity** files getting saved in **S3 Bucket Logs**



We can download any data object from here



After downloading this object we will open it.



Here, we can see all the terminal details of this particular SSH session through our Bastion Host

**Benefits/Necessity of Bastion Host**

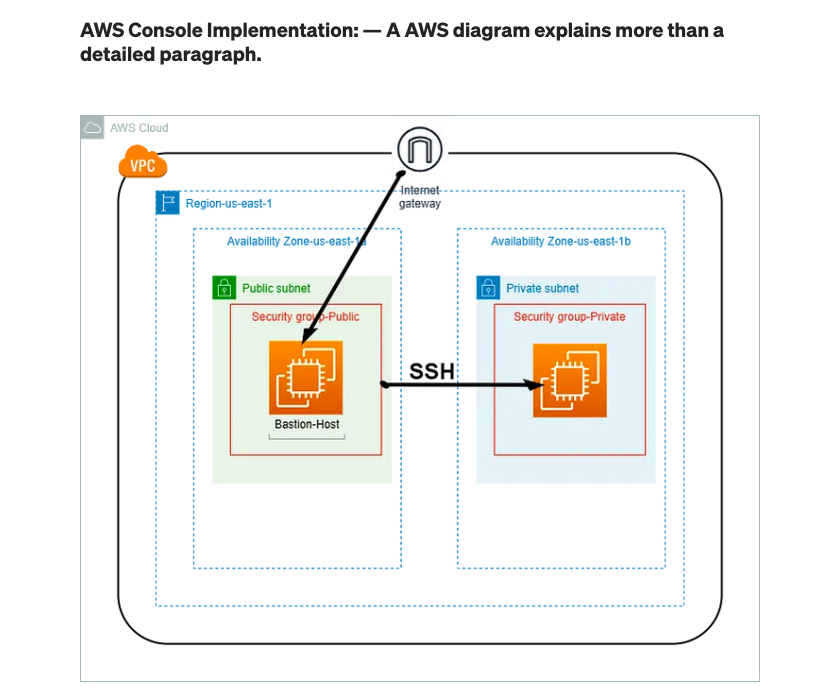
* Bastion hosts can be a valuable resource for companies, improving security and limiting access to shared resources.
* Bastion host provides an additional layer of protection to the actual server (residing in a private subnet) from any external harmful actors on the internet.
* The bastion host processes and filters all incoming traffic and prevents malicious traffic from entering into the network by acting much like a gateway.
* The most common examples of bastion hosts are mail, domain name system, Web and File Transfer Protocol (FTP) servers. Firewalls and routers can also become bastion hosts.
* Bastion hosts (also commonly called bastion servers) are typically configured with a bare minimum operating system with protocol-specific servers such as OpenSSH server or RDP gateway.

**Bastions are not only for SSH!**

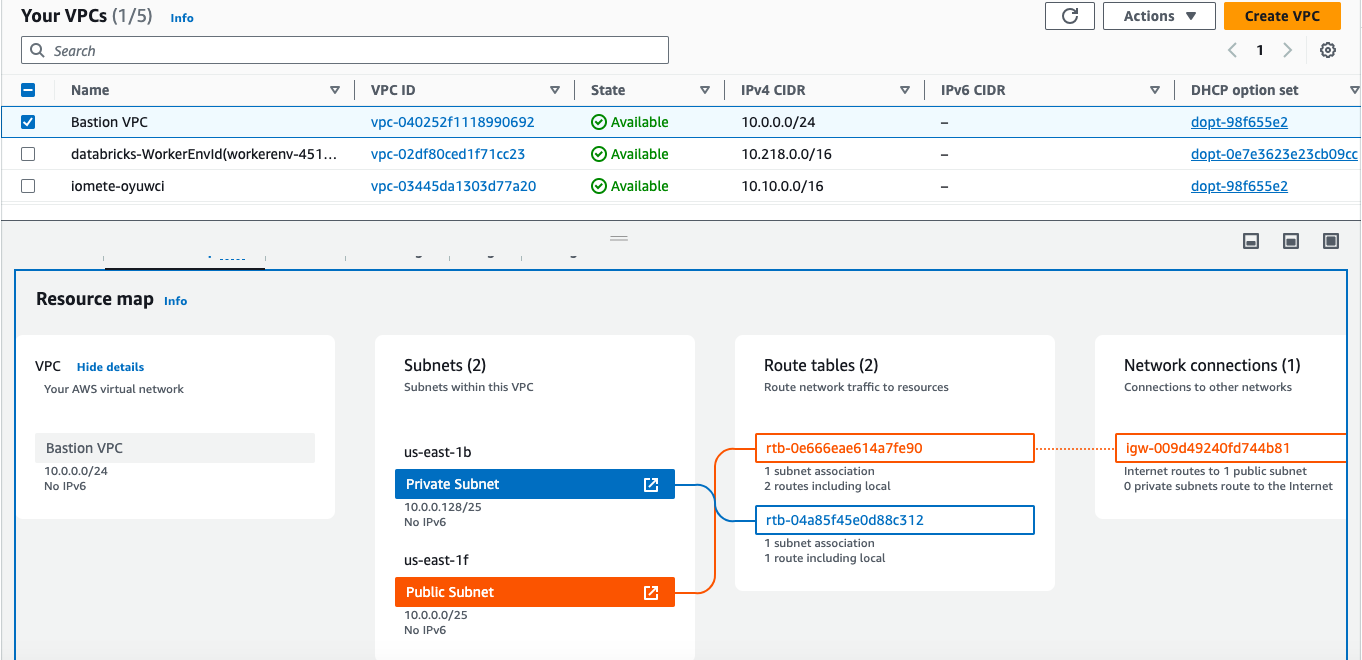
It is important to mention that Bastions are not only for SSH. Most of the time, people relate to bastion host as an SSH jump server, which is correct but does not cover all use cases. While a Linux server configured with OpenSSH (setup as SSH Jump server) is a typical example of a bastion host, a bastion can sit in front of any protocol. For example, you can use a bastion for database access, RDP access, and internal web application access. In fact, any internal endpoints which should not accept direct network access should be placed behind a bastion for extra security.

**Demo of Bastion Host solution using SSH and AWS console**

**Bastion Low Level Architecture**



## Setup VPC and subnets



## ****Bastion host Setup****

Network details — Public subnets with route table associated with its project Internet Gateway. For public security group will allow SSH from selected IP address but for this demo it allows from anywhere. Then for the private security group we must allow only the public EC2 instance IP address of the same VPC.

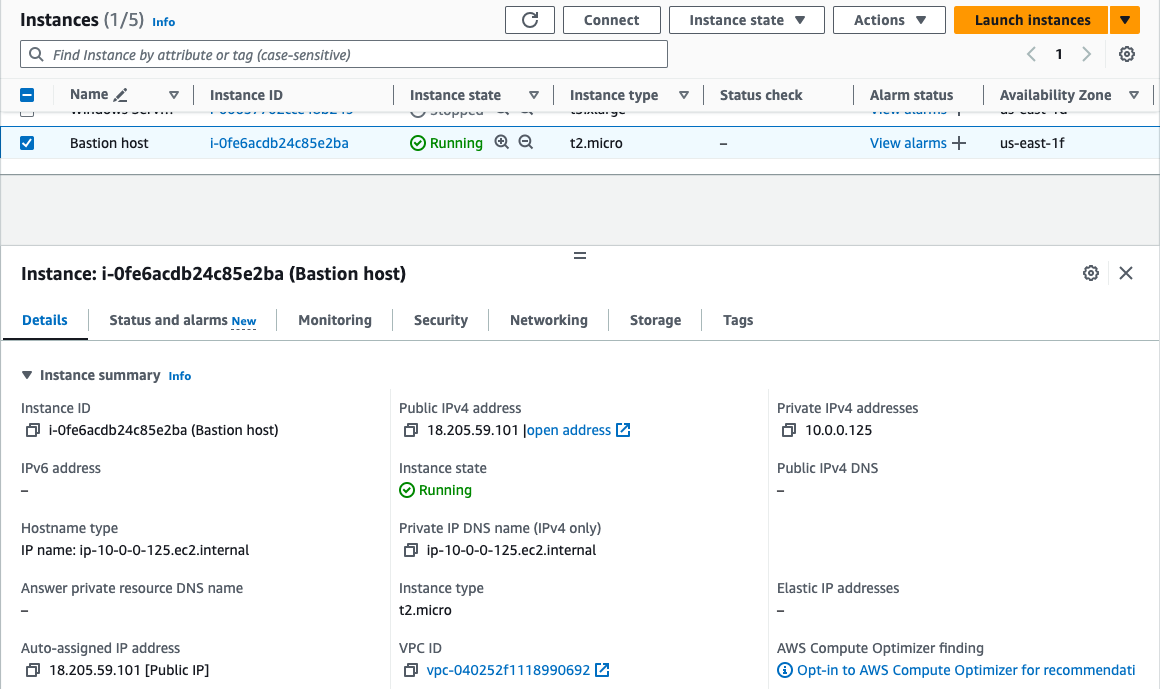
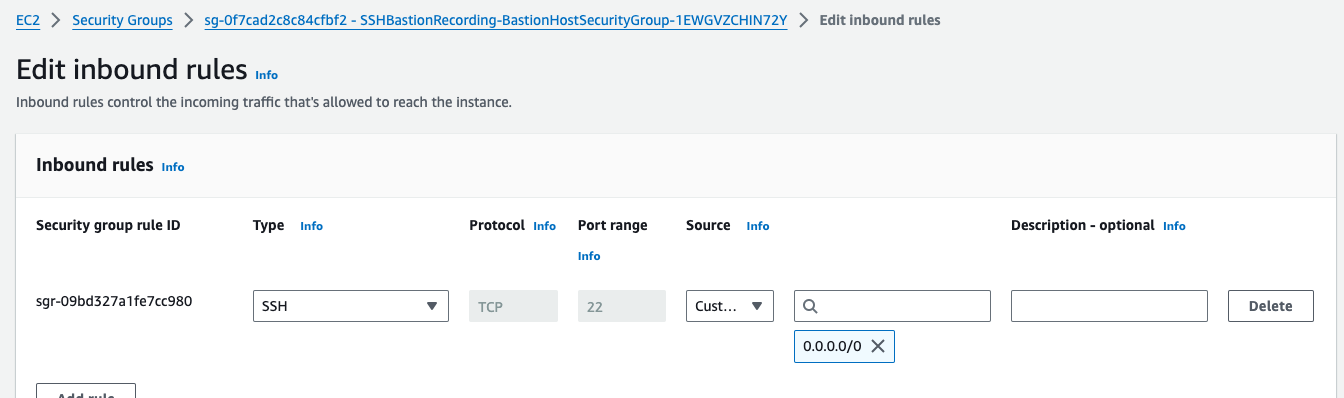


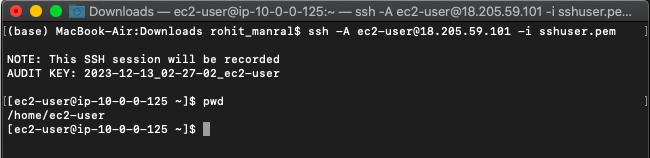
Figure: Network details of Bastion host

**Bastion- Security Group**

****

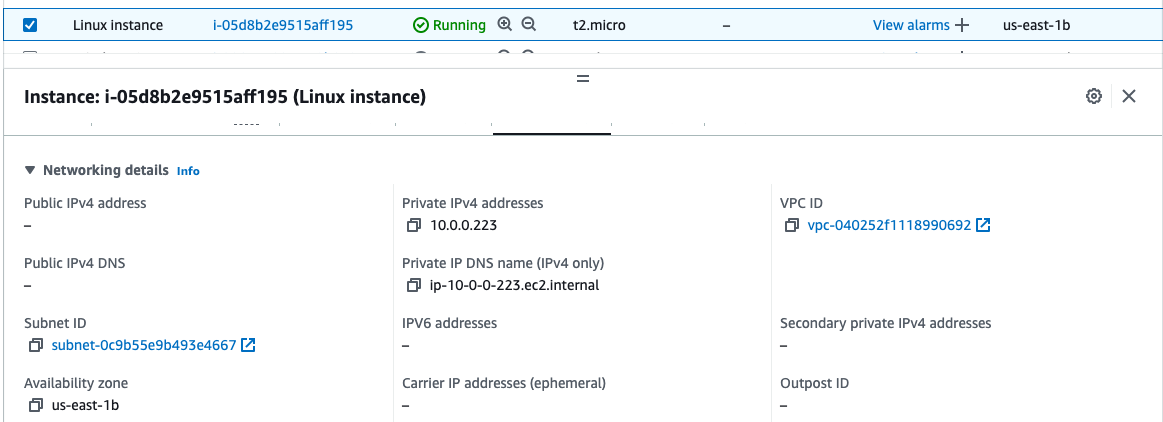
## Connect to Bastion host

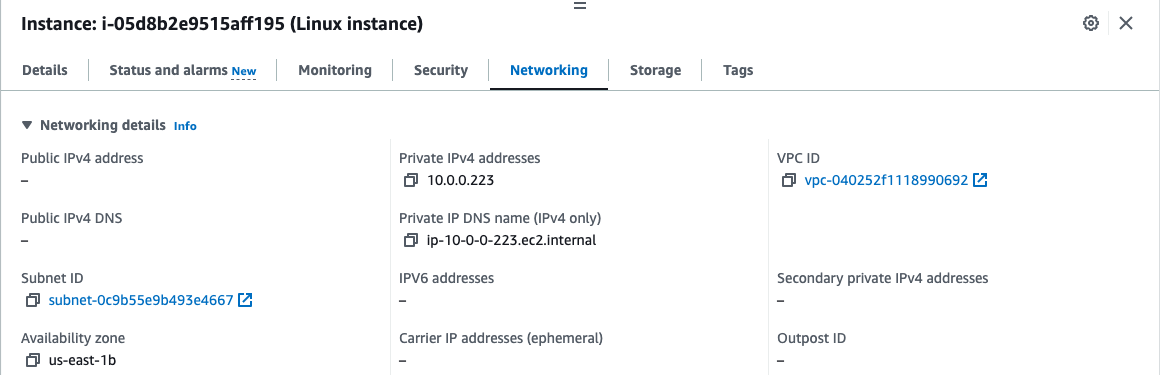
I used Mac Terminal to connect.



## Private EC2 Setup

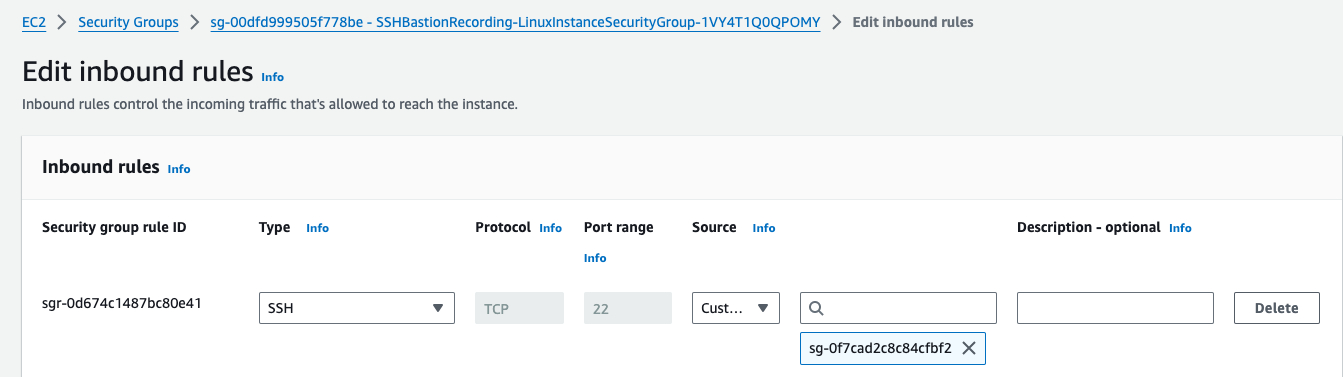
Creating the Private EC2 instance with Project VPC and the private subnet.





Yes, we are all done with the setup of private EC2 instance, now let's connect to the instance from Bastion Host instance.

**Private Linux- Security Group**



## Steps to connect through the Bastion Host into Private EC2 instance.

## 1. Connect to Bastion host first by ssh.

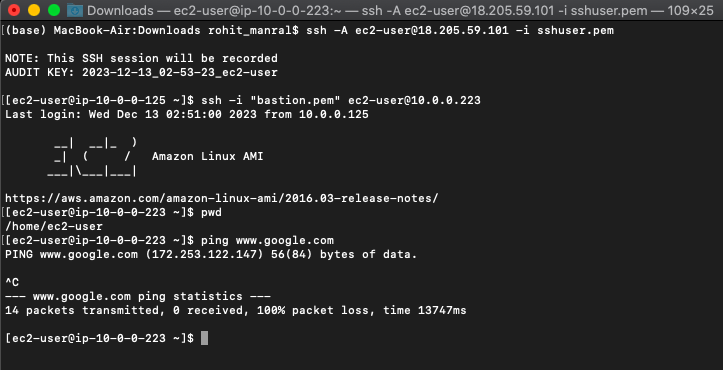
## 2. Once you're in-Bastion host. Copy your already created key pair for Bastion host.

## 3. now use nano or vim to create filename-keypair.pem, then paste the copied key into file, save and exit.

## 4. change permission to the created keypair by using $ chmod 400 <keypair>

## 5. use the cmd -> ssh -i “Filename-KeyPair.pem” username@Private\_IP\_address

## It's done, connected to the bastion host!



Finally, it worked and verified if the private instance has internet access by using ping. I hope this short article is useful, this is going to be useful for myself for future references.

# **Any Questions?**



**References**

* <https://aws.amazon.com/blogs/security/how-to-record-ssh-sessions-established-through-a-bastion-host/>
* <https://www.strongdm.com/blog/bastion-hosts-ssh-logging>
* <https://www.ezeelogin.com/blog/how-to-record-terminal-ssh-sessions/>
* <https://www.youtube.com/watch?v=dyVfnzUy2ys>