

AWS EC2 Instance | Steps to Connect Windows to EC2 Instance & Transfer Files From Windows to EC2 Instance through WinSCP & PuTTY

Prepared by: Aravind Panchanathan

Divyansh Nigam

Mayuri Jadhav

Date: March 22, 2024

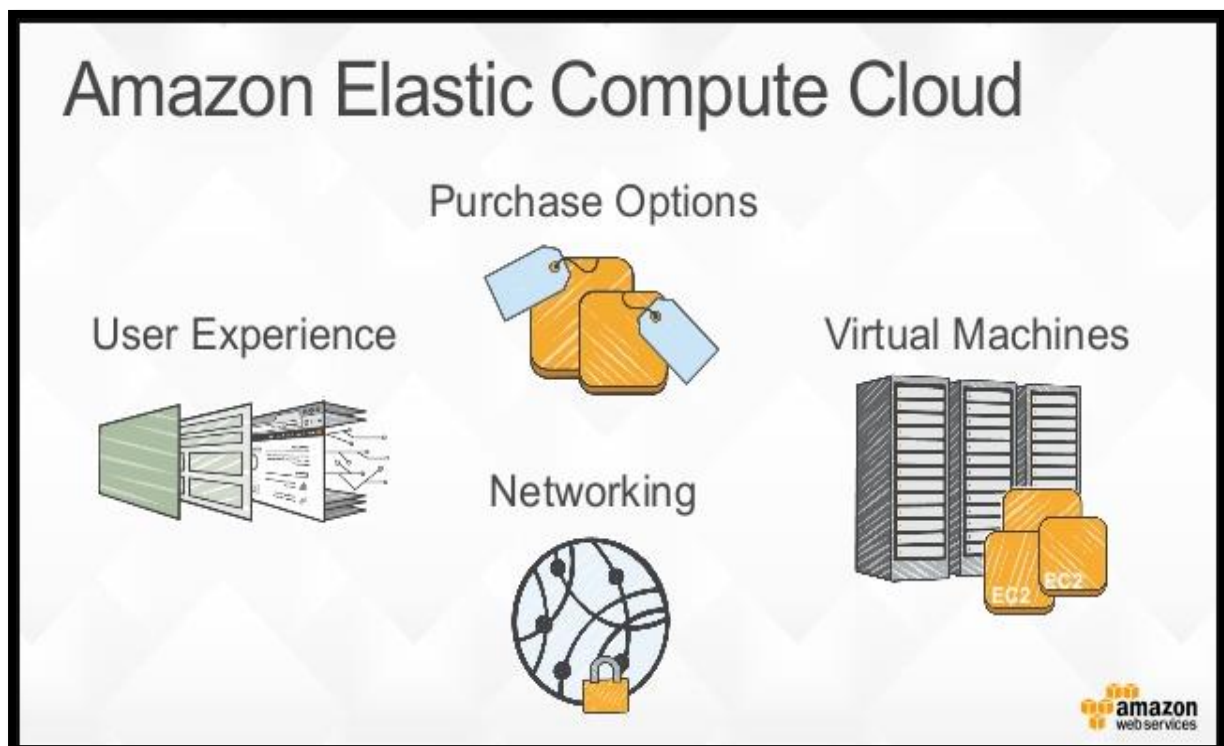
In this manual, we will cover AWS EC2 Instance (Amazon Elastic Compute Cloud), a computing service from AWS and in this guide, we will provide detailed instructions on establishing a connection to a Windows EC2 instance.

1. Introduction to Amazon EC2:

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It allows users to easily scale computing resources to meet changing demands and provides complete control over their virtual environments. EC2 offers a range of features including reliability, security, flexibility, and cost-effectiveness.

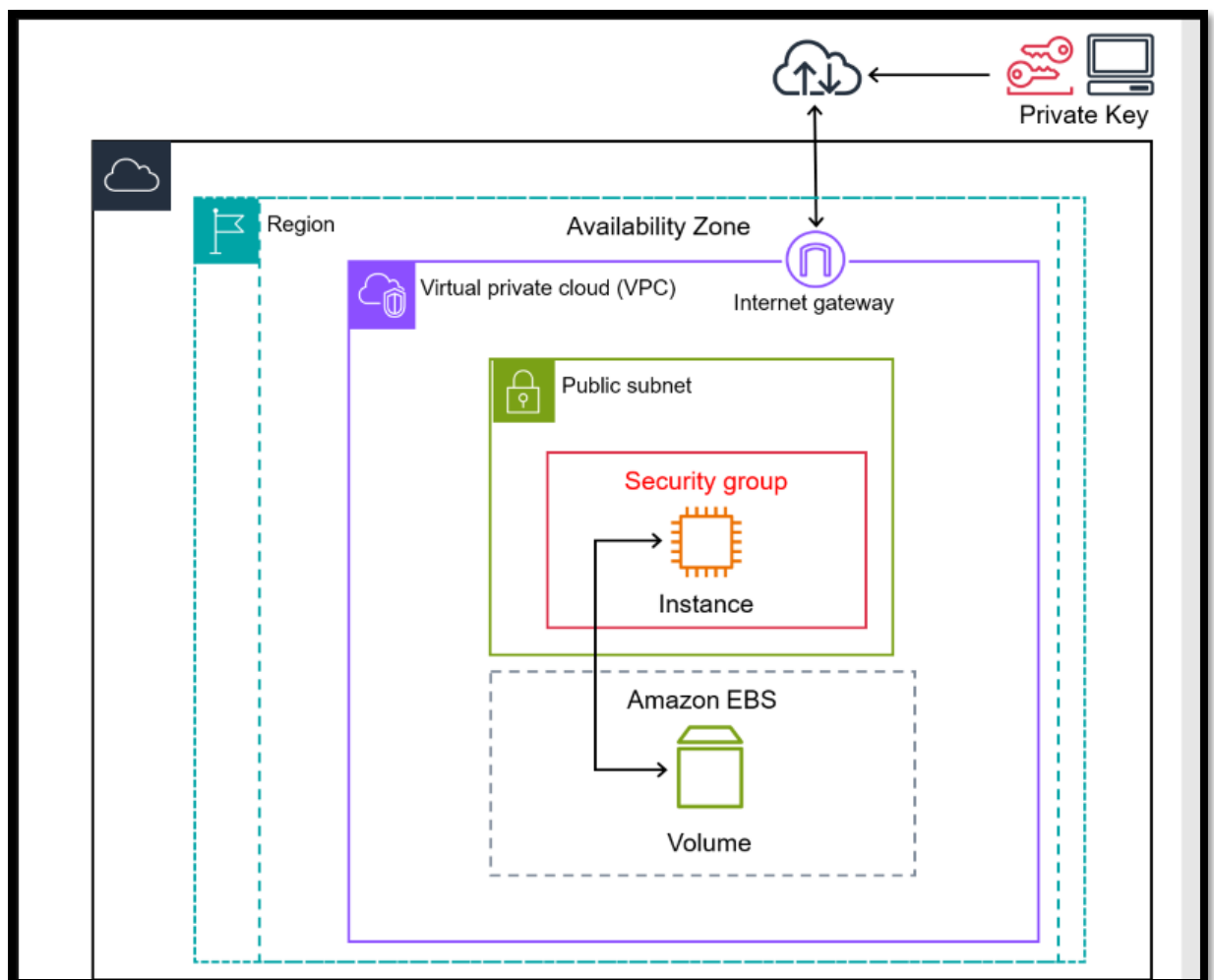
2. Why Use Amazon Elastic Compute Cloud (EC2):

- **Eliminates Hardware Investment:** EC2 eliminates the need for upfront investment in hardware, enabling faster application development and deployment.
- **Scalability:** EC2 allows users to launch and manage virtual servers as per their requirements, scaling up or down as needed.
- **Customization:** Users can choose from pre-configured Amazon Machine Images (AMIs) or create their own, customizing settings for security, networking, and more.
- **Cost-Effectiveness:** EC2 follows a "pay-for-what-you-use" billing model, allowing users to minimize costs by only paying for resources consumed.



3. How does AWS EC2 work?

It's simple to get up and running with EC2. You have a choice of pre-configured, templated Amazon Machine Images (AMI) to use for a quick launch. Or, if you prefer, you can create your own AMI that contains all of your libraries, data, applications, and relevant configuration settings. EC2 allows you to customize settings by configuring security and network access. Because you can instantly scale your VM environment to address usage spikes or drops, you have the power to control the number of resources being used at any point in time. The service's elasticity supports the minimized costs of a “pay-for-what-you-use” billing model.



4. Features of Amazon Elastic Compute Cloud:

- **Reliability:** EC2 offers a highly reliable environment with rapid instance replacement.
- **Security:** EC2 operates within Amazon Virtual Private Cloud (VPC), providing a secure and robust network environment.
- **Flexibility:** EC2 provides tools for developers and system administrators to build fault-tolerant applications.
- **Cost-Effectiveness:** Users pay only for the resources used on an hourly basis, with multiple purchase plans available.

5. Use Cases of AWS EC2 Instances:

- **Hosting Environments:** EC2 is commonly used for hosting websites and applications, providing a dynamic and scalable environment.
- **Backup and Disaster Recovery:** EC2 serves as a medium for performing disaster recovery in active and passive environments, ensuring minimal downtime.
- **High-Performance Computing:** EC2 offers virtualized servers with high compute power, catering to organizations with high-performance computing needs.
- **Development and Test Environments:** EC2's scalability facilitates the creation and deployment of large-scale testing and development environments.
- **Banking and Financial Sector:** EC2 meets the scalability and security demands of sectors like banking and finance, providing highly secure services.

6. Amazon Machine Images (AMIs):

Amazon Machine Images (AMIs) are pre-configured templates used to quickly launch EC2 instances. These templates include operating systems, applications, and security settings, streamlining deployment and saving time. Users can customize or create their own AMIs to suit specific requirements. AMIs cover a range of needs, from general workloads to specialized environments like databases, applications, machine learning, and security tools. Leveraging AMIs ensures efficient and reliable deployments tailored to diverse use cases in AWS EC2.

7. Instance Types:

AWS EC2 offers various instance types designed to meet specific computing needs, differing in CPU, memory, storage, and networking capacity. By understanding the different instance types and their capabilities, you can choose the right one to optimize your workload performance and cost-effectiveness.

1. General Purpose Instances:

- Suitable for a wide range of workloads such as small to medium-sized databases, web servers, and development environments.
- Examples: t2, m5, m6g.
- Example usage: t2.micro for small-scale WordPress websites, m5.large for small-scale e-commerce websites.

2. Compute Optimized Instances:

- Ideal for compute-intensive workloads like high-performance computing (HPC), scientific modelling, and data analytics.
- Examples: c5, c6g, c6gd.

3. Memory Optimized Instances:

- Designed for workloads requiring large memory, such as in-memory databases, big data processing, and analytics.
- Examples: r5, r6g, x1e.

4. Storage Optimized Instances:

- Suitable for storage-intensive workloads demanding high disk throughput and low latency, like data warehousing and log processing.
- Examples: i3, i3en, d3.

5. GPU Instances:

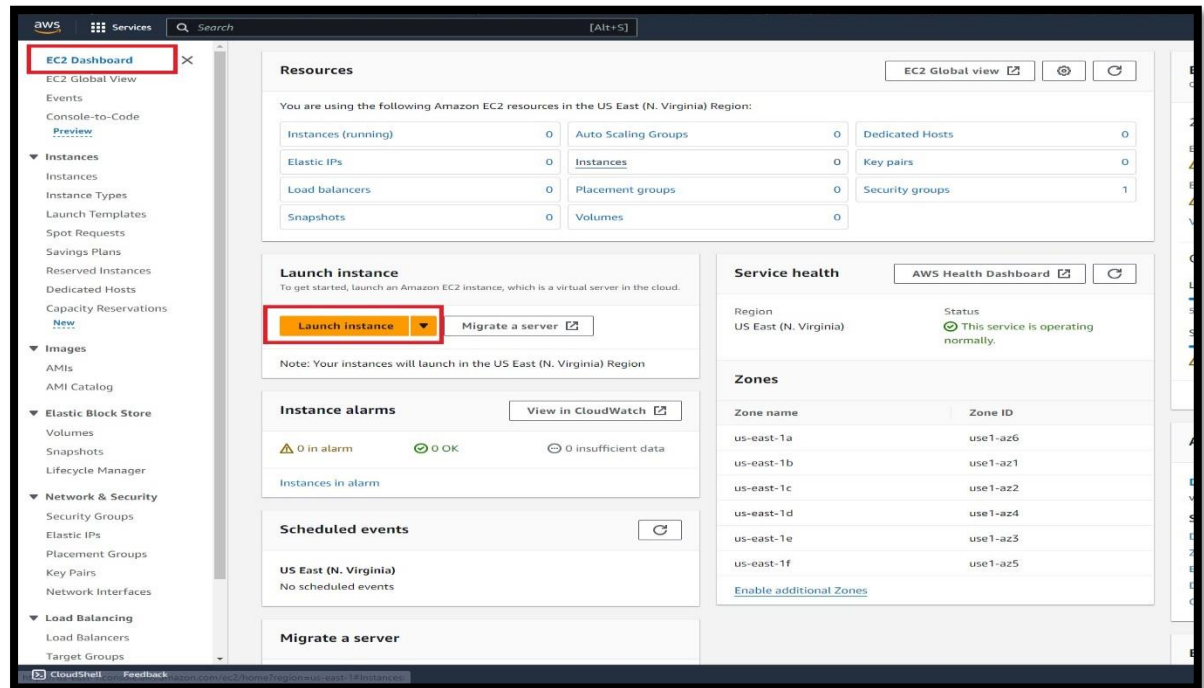
- Tailored for workloads demanding high-performance GPU resources, such as machine learning, deep learning, and graphics rendering.
- Examples: p3, g4, infl.

6. High-Performance Computing Instances:

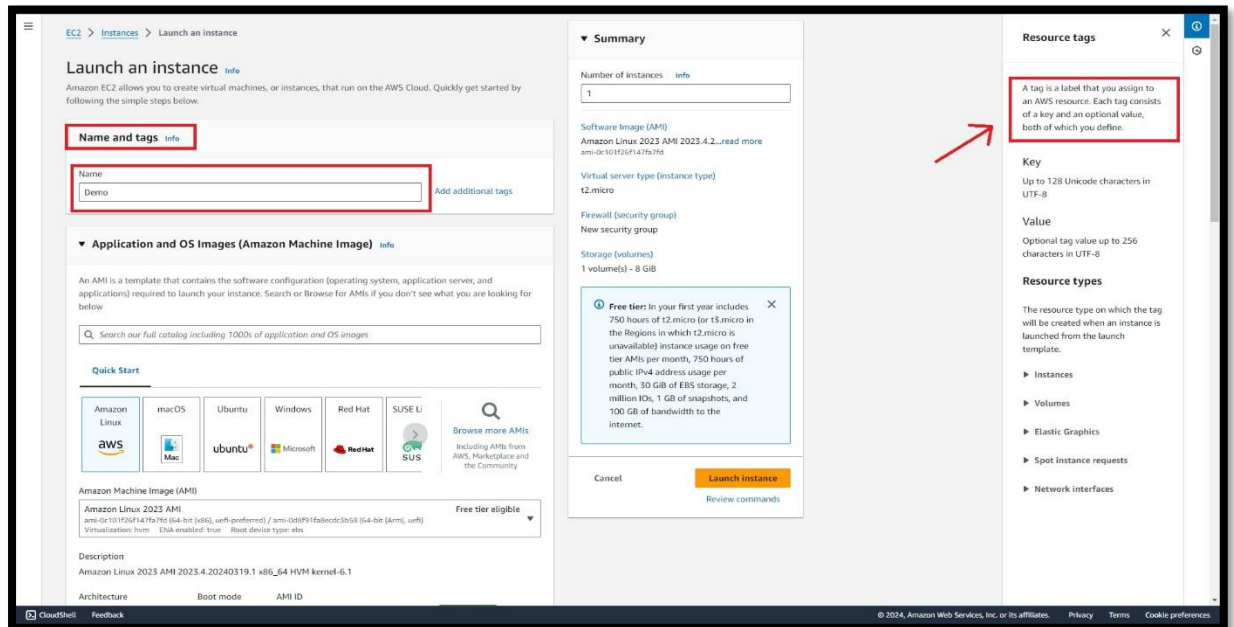
- Designed for high-performance computing (HPC) tasks with high network performance requirements.
- Examples: hpc6id, hpc6a.

8 Steps to Create AWS Windows EC2 Instance

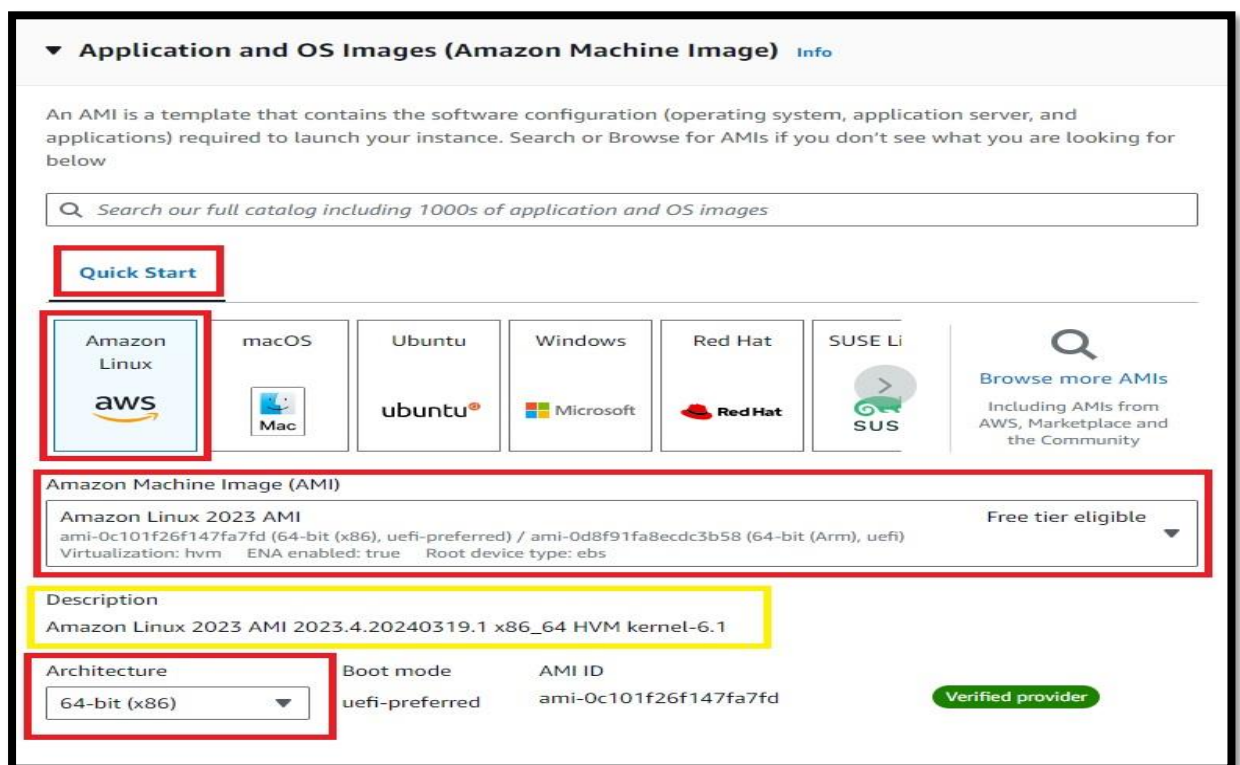
Step 1: Log in to your AWS account and access the EC2 dashboard. Open your web browser, navigate to the AWS Management Console, and sign in with your credentials. Click on "Services" in the navigation bar, select "EC2" from the dropdown menu, and you'll be directed to the EC2 dashboard. From there, proceed to launch a new instance by clicking on "Launch Instance."



Step 2: In the "Name and Tags" configuration phase, you have the option to attach tags to an instance. These tags facilitate the organization of AWS resources based on various criteria, such as ownership, environment, or function. For instance, you can define tags for your EC2 instances to track the owner and stack level of each instance effectively.



Step 3: Choose "Amazon Linux" from the QuickStart options and opt for the "Amazon Linux 2023 AMI ". While alternative AMIs are available to suit specific requirements, for this instance, select the " Amazon Linux 2023 AMI “ and Architecture as “ 64-bit (x86) “as we are deploying a Linux Server.



Step 4: Choose the t2. micro instance type. While alternative instance types are available for selection, note that they incur charges. Opting for the t2. micro instance ensures eligibility for the AWS Free Tier and its associated limited resources.

▼ **Instance type** Info | Get advice

Instance type

t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

Q |

Get advice on instance type selection...

t2.nano
Family: t2 1 vCPU 0.5 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0058 USD per Hour
On-Demand SUSE base pricing: 0.0058 USD per Hour
On-Demand Windows base pricing: 0.0081 USD per Hour

t2.micro Free tier eligible ✓
Family: t2 1 vCPU 1 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.0716 USD per Hour
On-Demand Linux base pricing: 0.0116 USD per Hour

t2.small
Family: t2 2 vCPU 2 GiB Memory Current generation: true
On-Demand Windows base pricing: 0.032 USD per Hour
On-Demand Linux base pricing: 0.023 USD per Hour
On-Demand RHEL base pricing: 0.083 USD per Hour
On-Demand SUSE base pricing: 0.053 USD per Hour

t2.medium
Family: t2 2 vCPU 4 GiB Memory Current generation: true
On-Demand Linux base pricing: 0.0464 USD per Hour
On-Demand RHEL base pricing: 0.1064 USD per Hour
On-Demand Windows base pricing: 0.0644 USD per Hour
On-Demand SUSE base pricing: 0.1464 USD per Hour

☐ All generations
[Compare instance types](#)

ss to the selected key pair

[Create new key pair](#)
decrypted password to connect to

[Edit](#)

Step 5: Choose either an existing key pair or generate a new one. For a new key pair, input "Demo key" or any other name as the name and select RSA as encryption for the key and select ".ppk " as the private key format as we use Putty in the future and proceed to create the key pair.

▼ **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*

Select ▼

[Create new key pair](#)

For Windows instances, you use a key pair to decrypt the administrator password. You then use the decrypted password to connect to your instance.

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

Demo Key

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

☒ **RSA**
RSA encrypted private and public key pair

☐ **ED25519**
ED25519 encrypted private and public key pair

Private key file format

☐ .pem
For use with OpenSSH

☒ .ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#) [↗](#)

Cancel **Create key pair**

Step 6: Now, maintain the default settings and proceed to launch the instance by selecting the "Launch Instance" button.

▼ Summary

Number of instances

Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.4.2...[read more](#)

ami-0c101f26f147fa7fd

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 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

×

Cancel

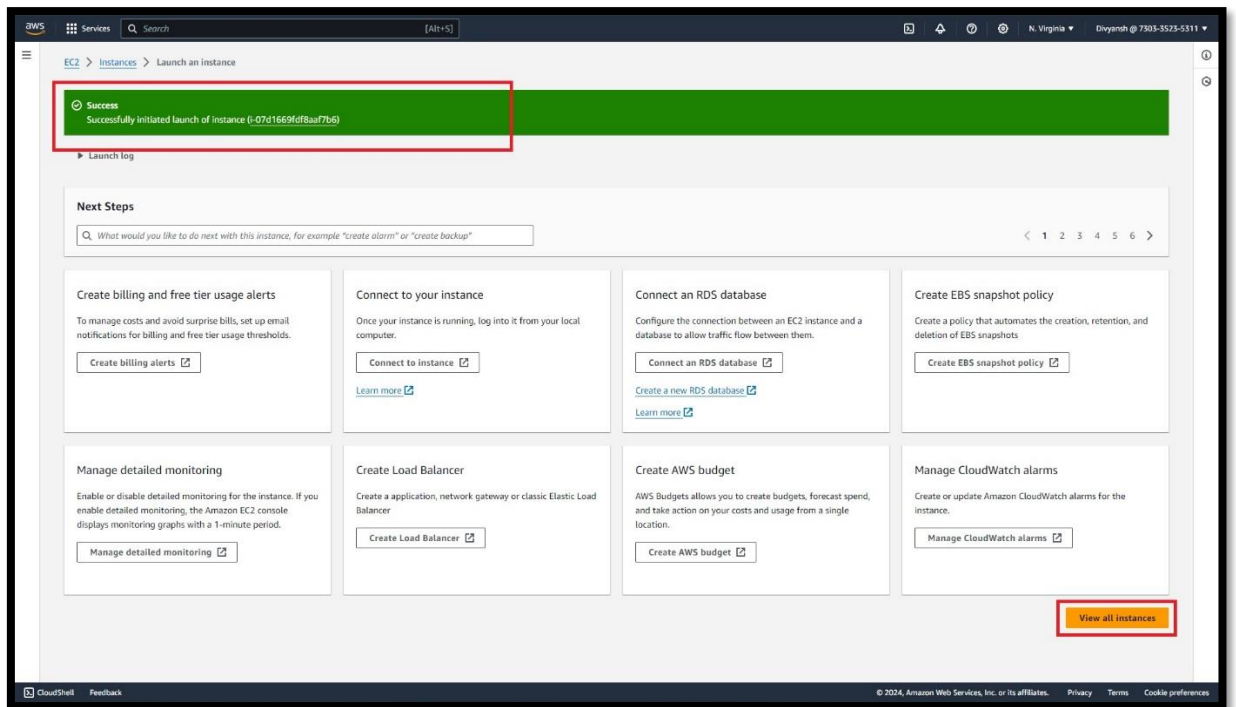
→

Launch instance

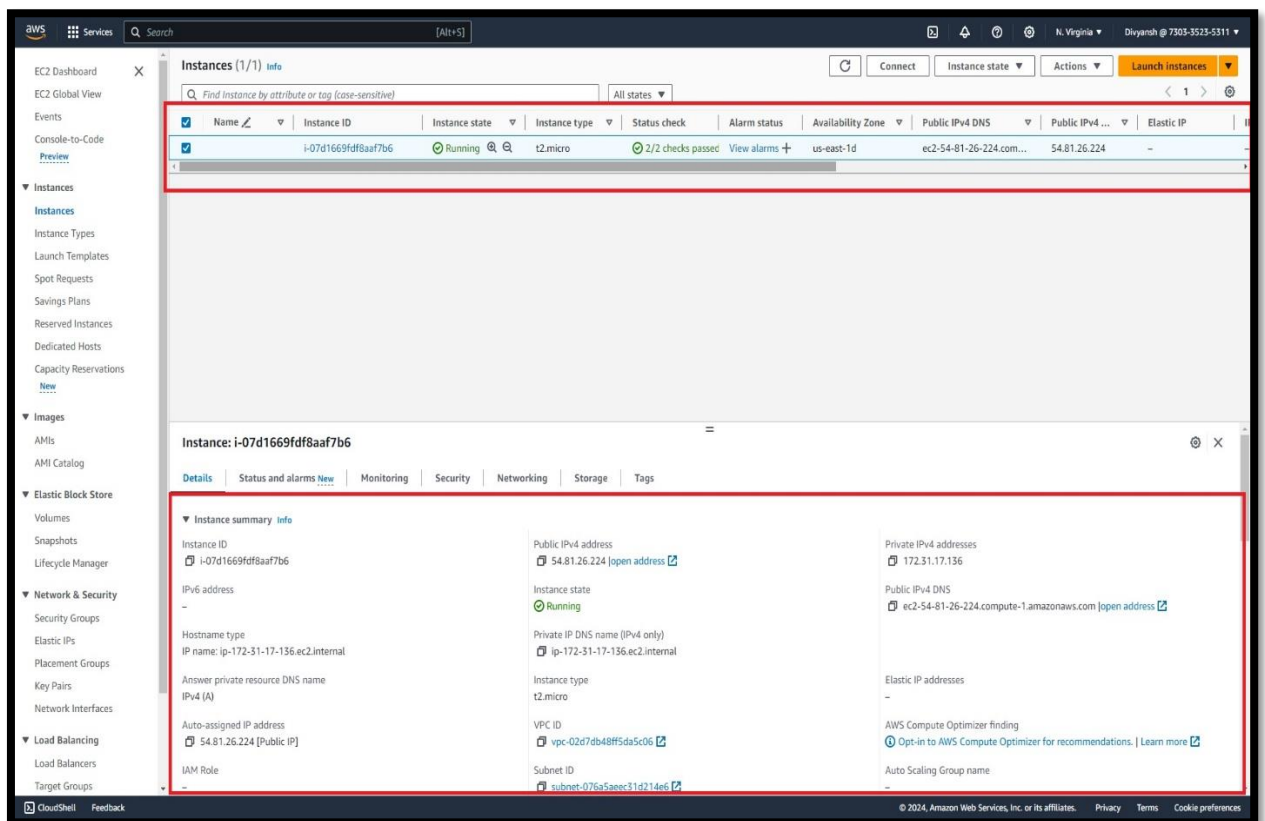
[Review commands](#)

Step 7: Upon completion of the instance configuration, a success message confirming the successful initiation of the instance launch will appear. Next, click on "View all Instances" to review your instance launch status and

configurations, ensuring that your EC2 instance creation process has been successfully initiated and is underway.



Step 8: Upon launching, the instance will be in an operational state, indicated by the "Running" status and all checks passing successfully. Additionally, essential details such as Instance ID, Public IPv4 address, and Private IPv4 addresses will become visible.

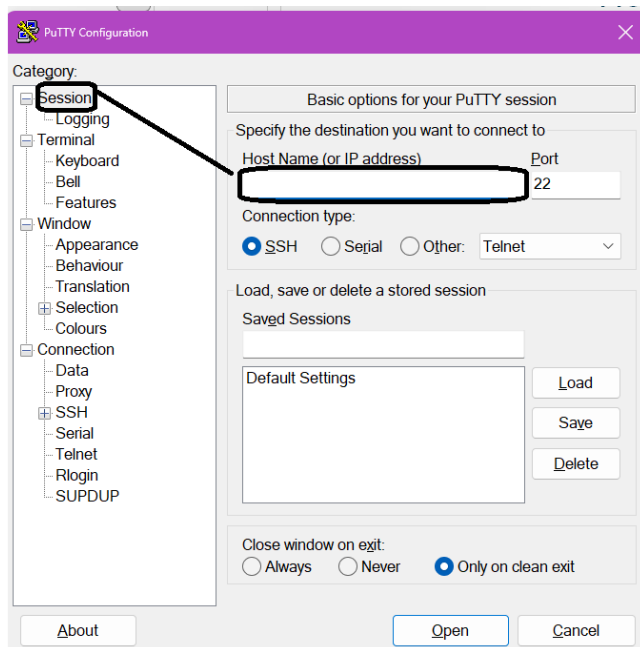


Step 9: We need to establish a connection to the EC2 Instance from a local machine. We'll utilize Putty as the interface between the local host machine

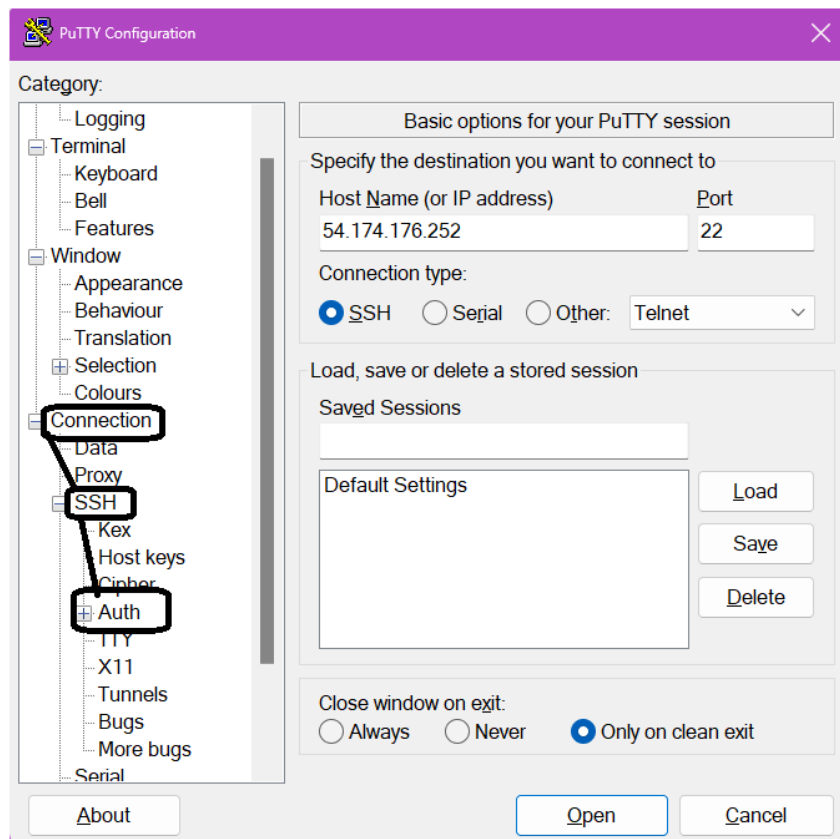
and the EC2 Instance. Our login credentials will be as the 'ec2-user', and authentication will be done using the Putty file.

Step 9.1: Download putty (<https://www.putty.org/>)

Step 9.2: Open putty and Navigate to Session Module, give the host IP Address, this can be found in the instance details.

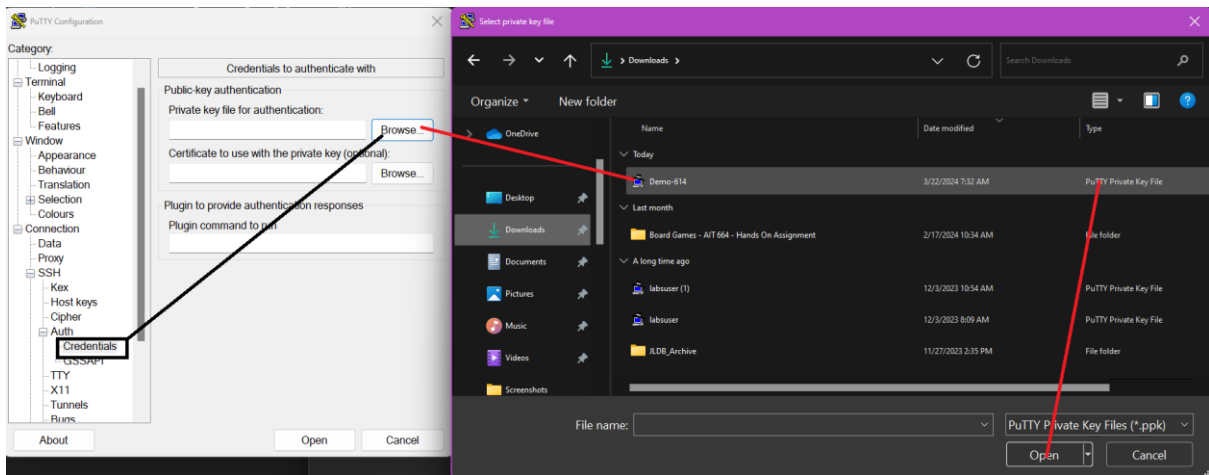


Step 9.3: Now Navigate to Connections > SSH > Authentication.

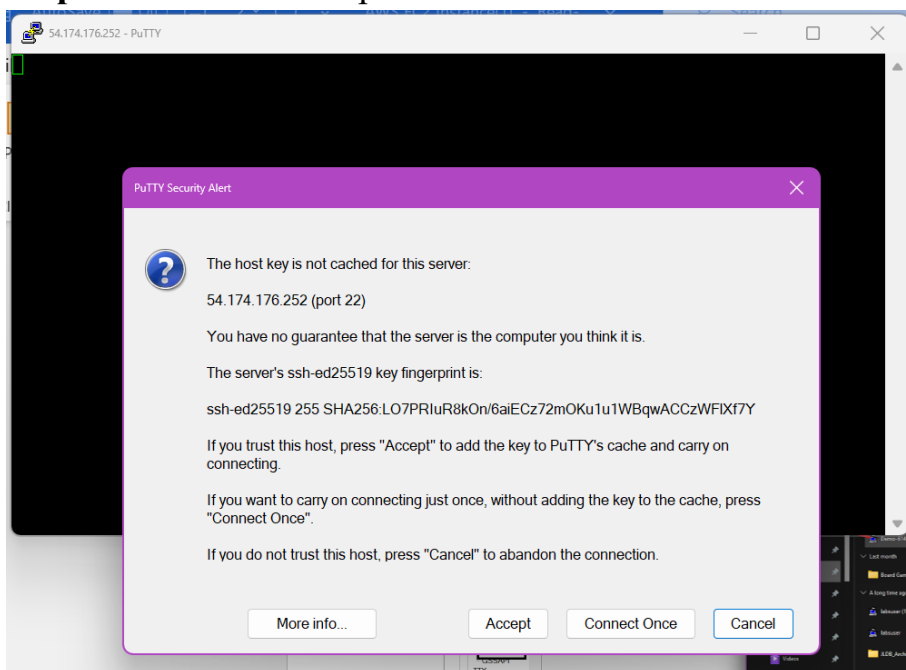


Step 9.4: Navigate to Authentication > Credentials > Select the .ppk file

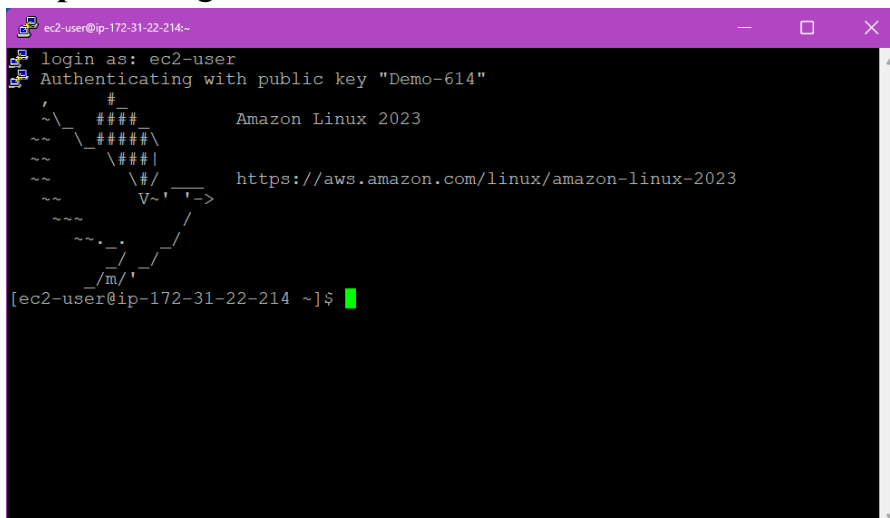
Note: .ppk file, is the file which will be downloaded once we create key pair.



Step 9.5: Click on accept.



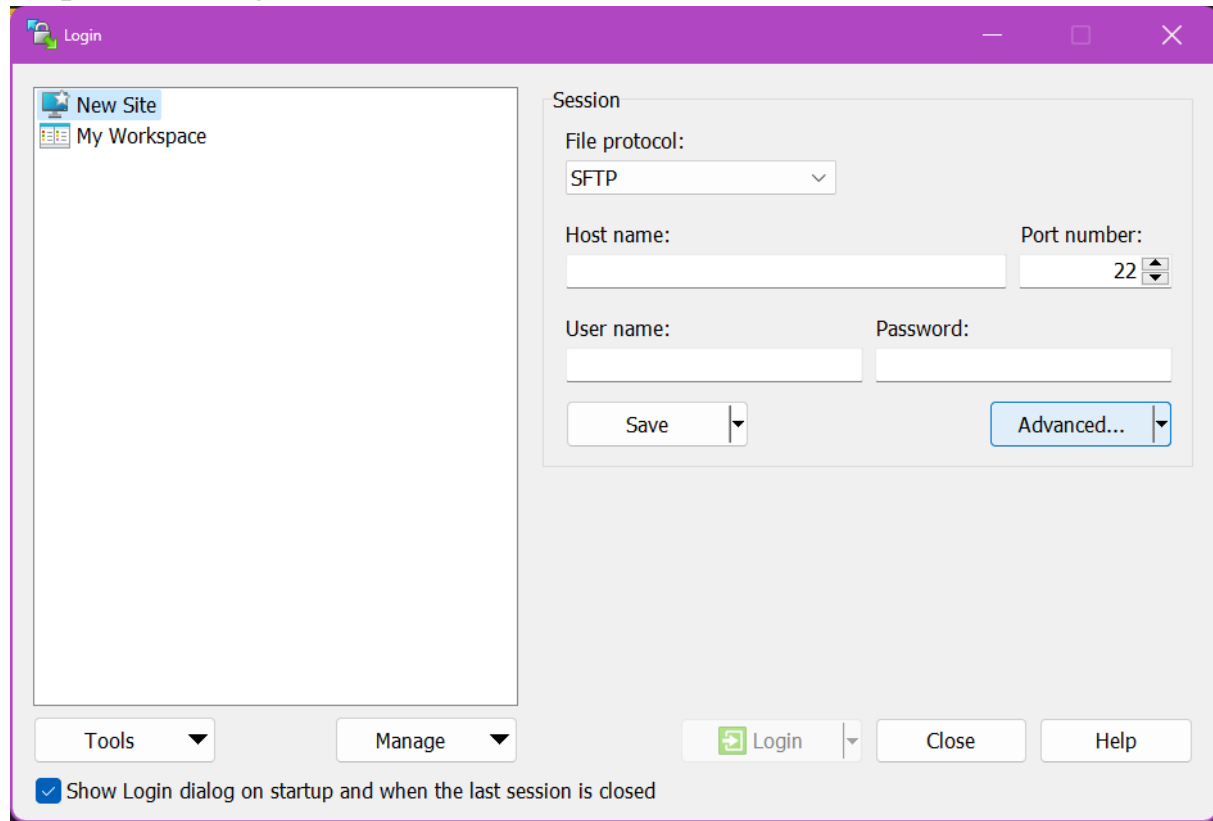
Step 9.6: Login in as ec2-user



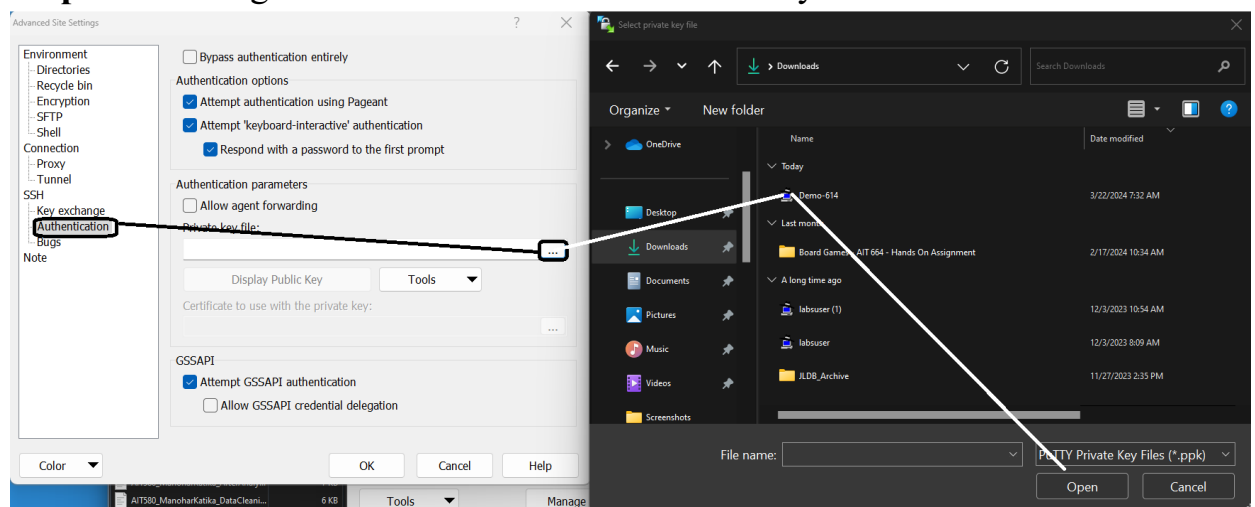
Step 10: Connect WinSCP and EC2 Instance, through same ppk file as authentication method, WinSCP support File Transfer protocol, where in we can transfer files from remote desktop to EC2 Instance.

Step 10.1: Download WinSCP (<https://winscp.net/eng/index.php>)

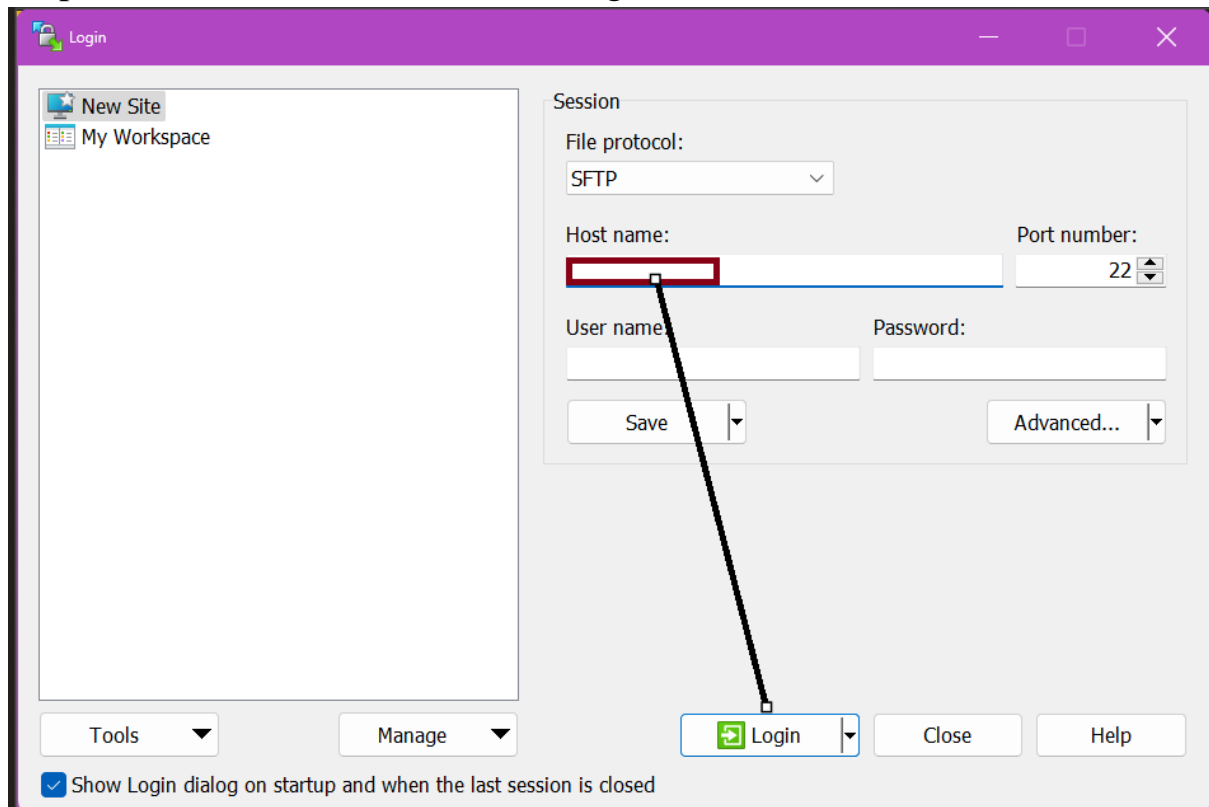
Step 10.2: Navigate to NewSite > Click Advanced.



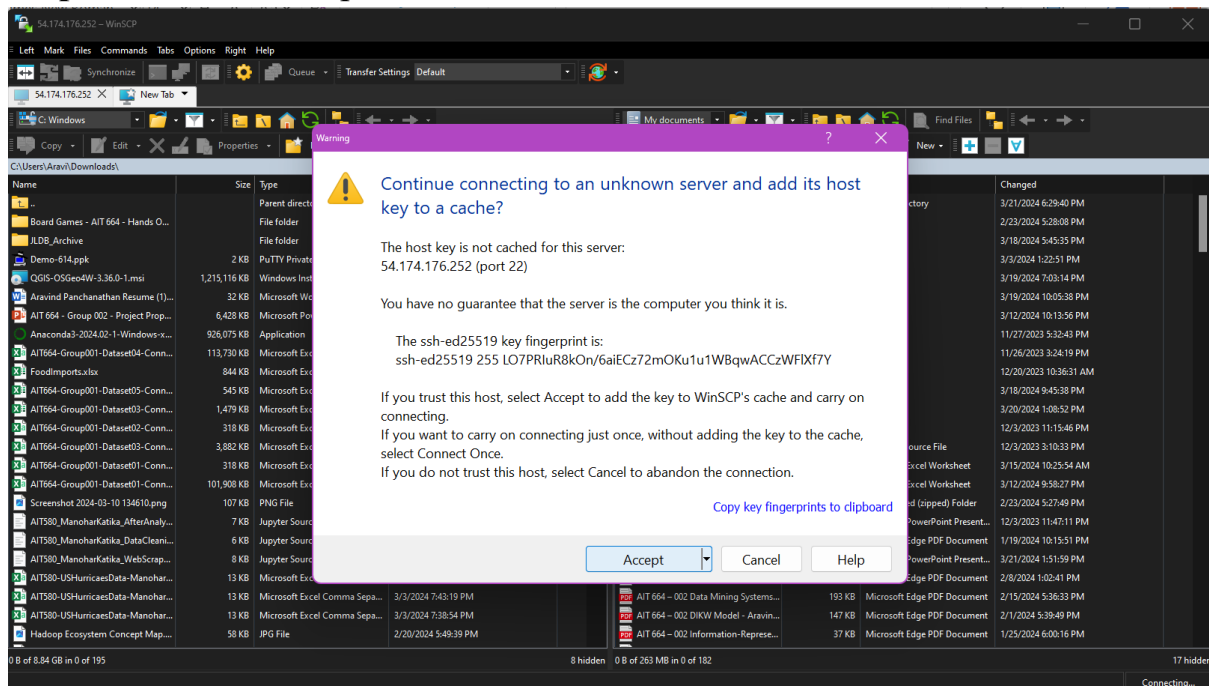
Step 10.3: Navigate to authenticate and select the key as shown below.



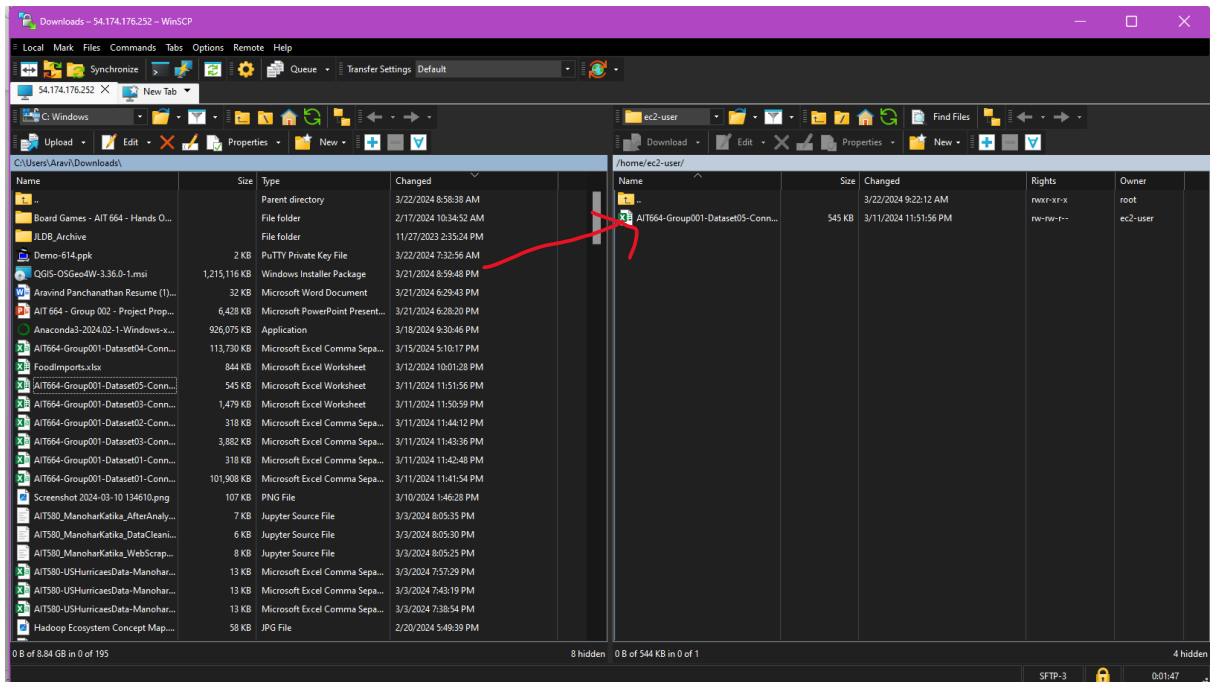
Step 10.4: Enter Hostname and then Login.



Step 10.5: Click Accept



Step 10.6: Select the file you want in your EC2 Instance, transfer the files. Drag & drop from left to right.



Step 10.7: We can navigate through putty and type ls, you should be able to see the transferred file from local machine to ec2 instance

