PROJECT DOCUMENTATION: LAUNCH YOUR LINUX MACHINE ON AWS WITH PUTTY 🎉 💻

BY KHUSHI NANDWANI

PROJECT OVERVIEW 🚀

In this project, you'll learn how to launch a Linux instance on Amazon Web Services (AWS) and connect to it using PuTTY, a popular SSH client for Windows. This guide will walk you through the entire process, from setting up an AWS EC2 instance to accessing it securely through PuTTY. By the end of this project, you'll have your own Linux server running in the cloud, ready for you to explore and customize!

PREREQUISITES 🔑

Before starting this project, make sure you have the following:

🌐 **AWS Account**: An active AWS account to launch EC2 instances.
💻 **PuTTY Software**: A free SSH client to securely connect to your Linux instance.
🔑 **PuTTYgen**: A tool to convert your AWS key pair (.pem) into a PuTTY-compatible format (.ppk).
🖥️ **Basic Knowledge of SSH**: Understanding SSH for connecting to remote servers.

⌨ **Command Line Familiarity**: Comfortable using basic Linux commands (e.g., ls, cd, pwd).

Once you've got these, you're all set to launch your AWS Linux instance and connect using PuTTY! 🚀

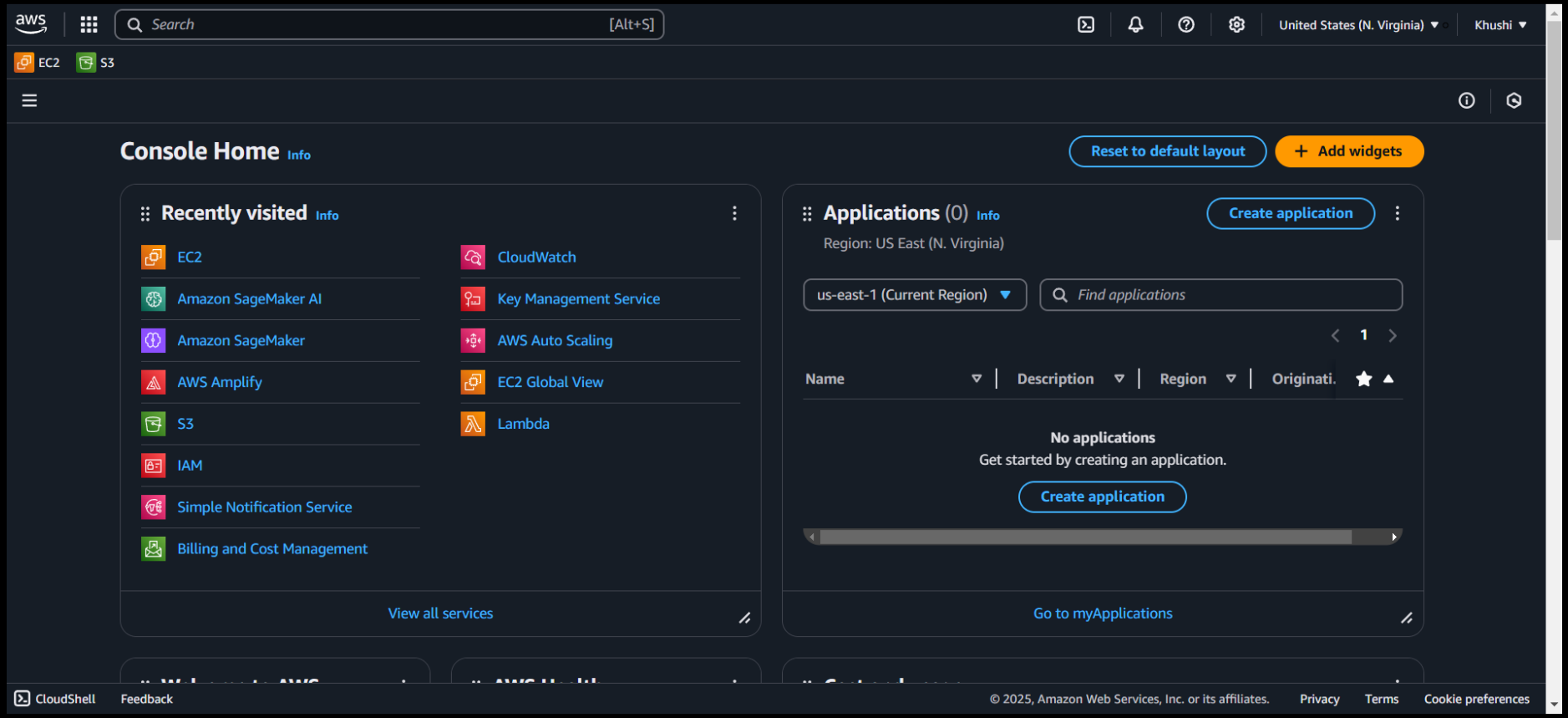
WHAT IS PUTTY? 💻

PuTTY is a free and open-source SSH client that allows you to securely connect to remote servers, typically Linux-based, from a Windows machine. It enables you to access and manage your server via the command line. 🖥️🔒

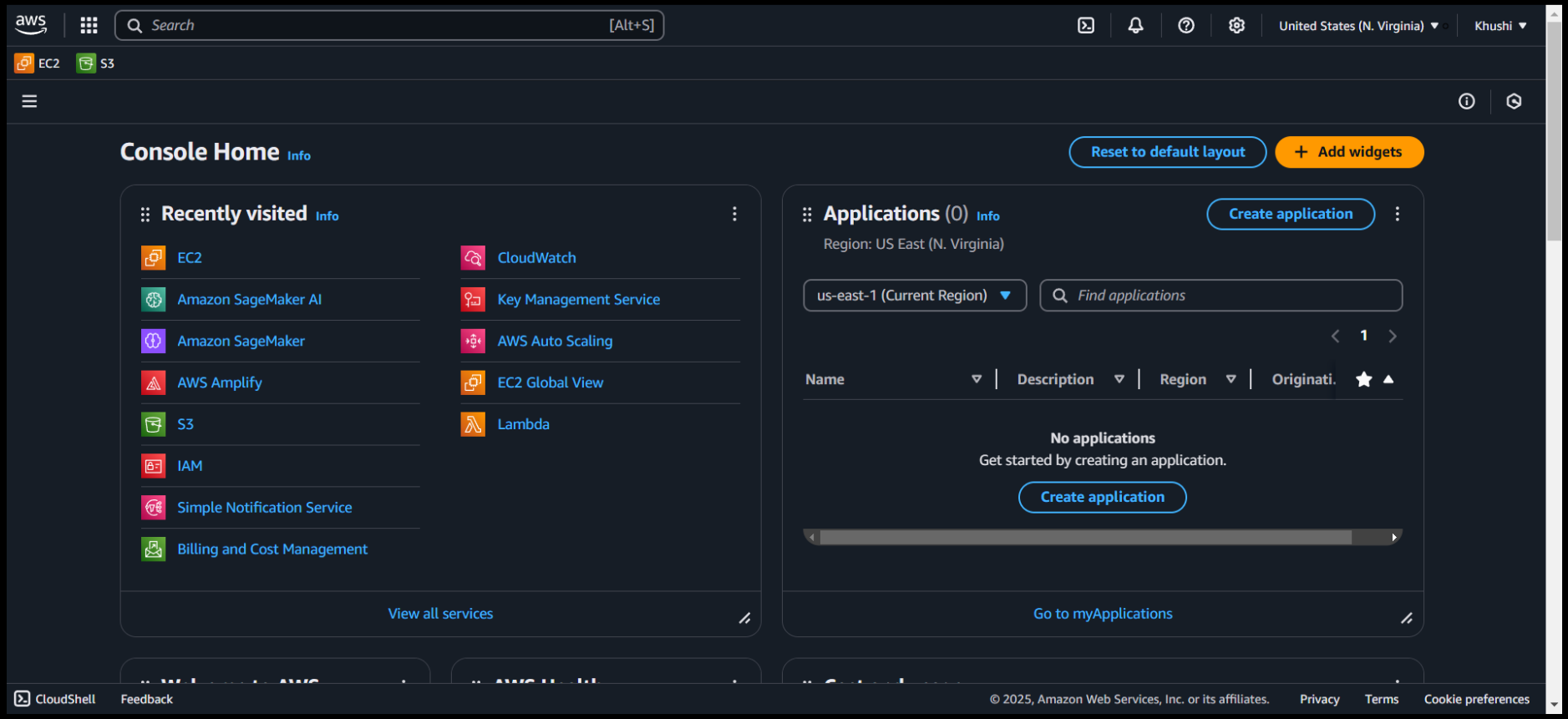
Ready to get your Linux server running on AWS and connect to it using PuTTY? Let's go through the steps together in a fun and easy way! 🎉

I. LAUNCH YOUR EC2 INSTANCE ON AWS 🚀

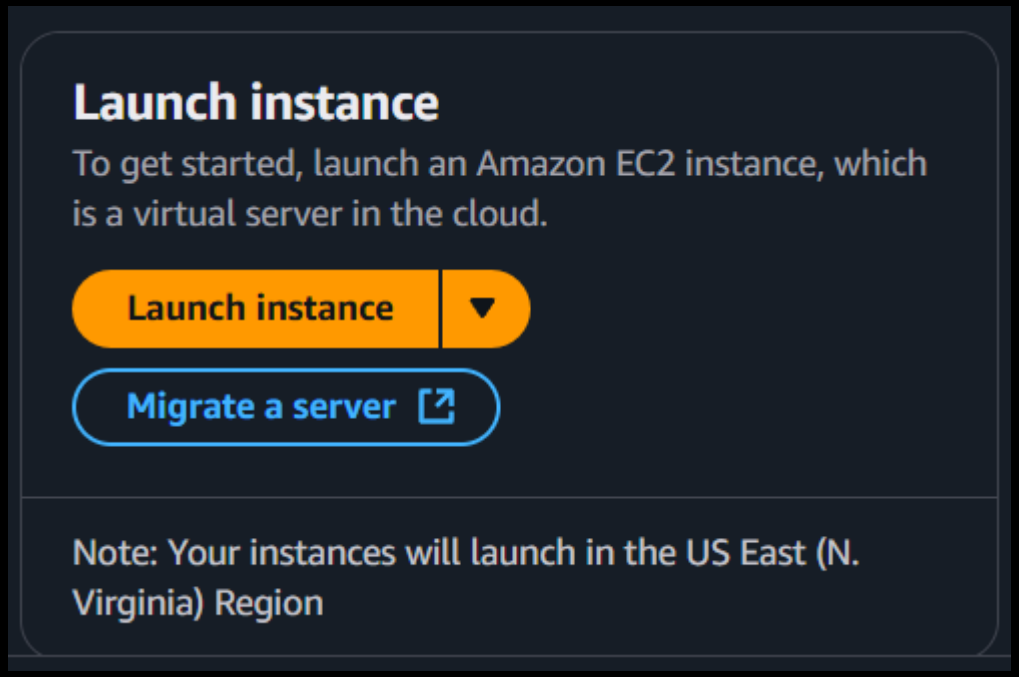
1. **Log in** to your **AWS Management Console.** It's time to get started! 😎



2. Head over to **EC2 Dashboard** under **Services**, the heart of your cloud adventure.

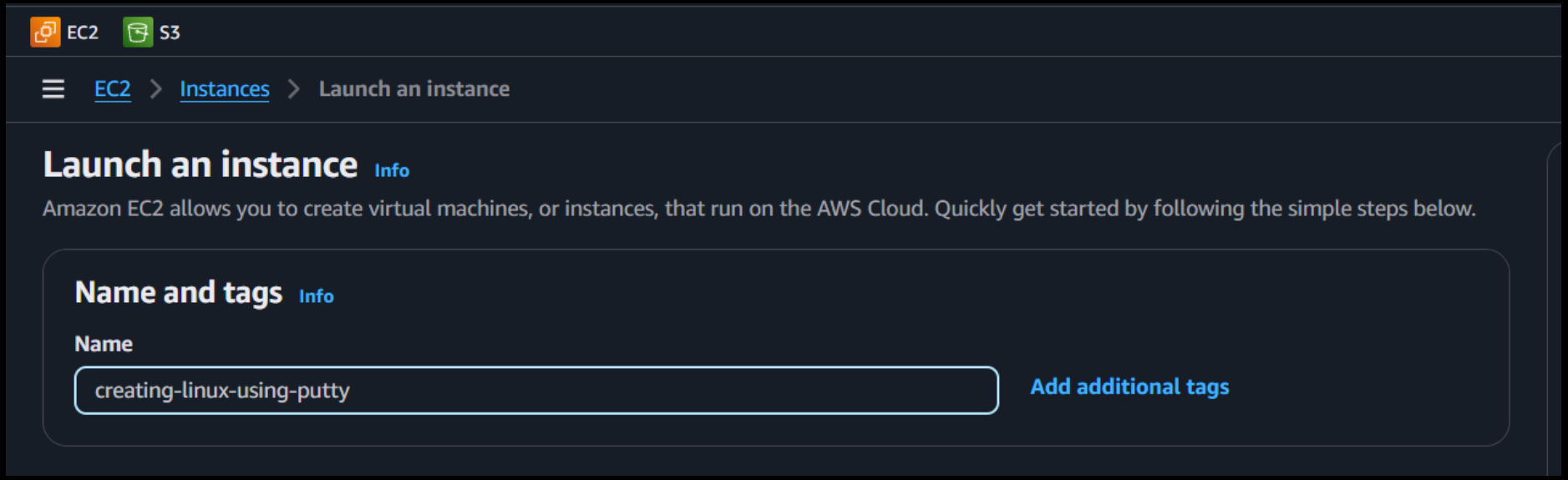


3. Click the **Launch Instance** button and get ready for takeoff. 🛫

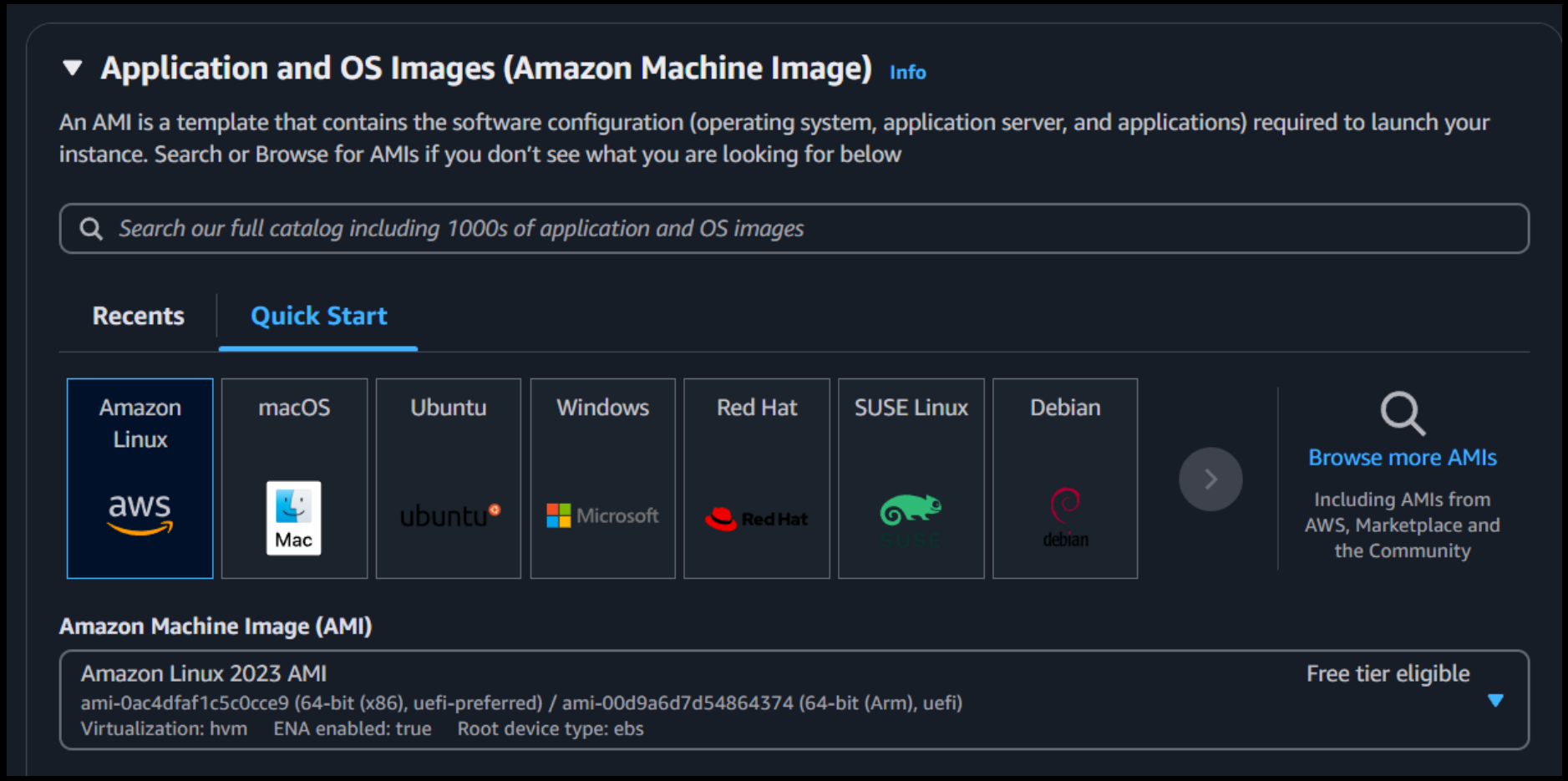


4. **Give a name** to your instance, this helps you easily identify it later. 🏷️

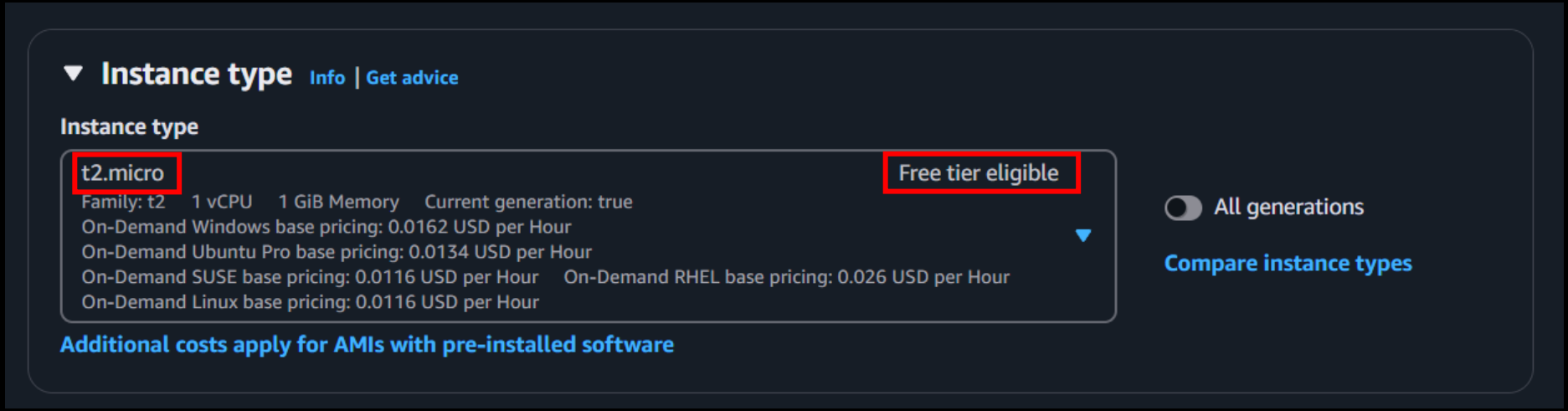
   (Eg – creating-linux-using-putty)



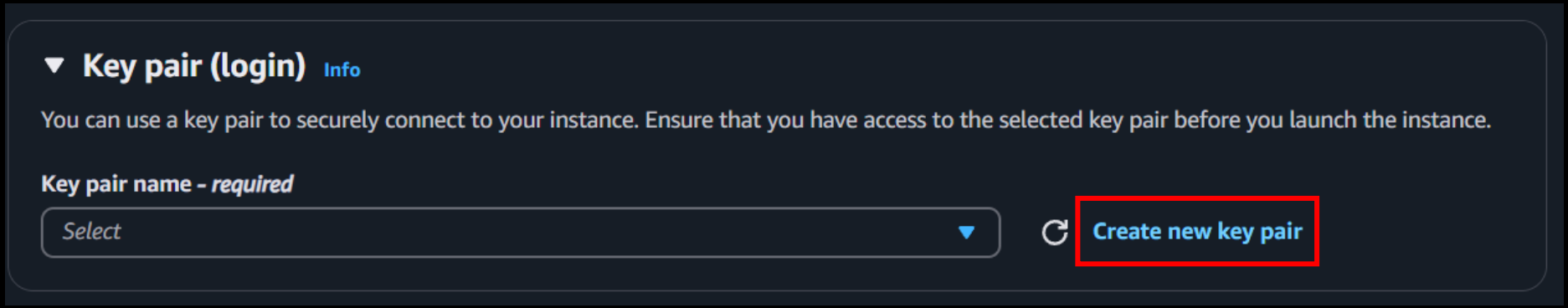
5. Pick your **Amazon Machine Image (AMI)** (e.g., **Ubuntu** or **Amazon Linux**).



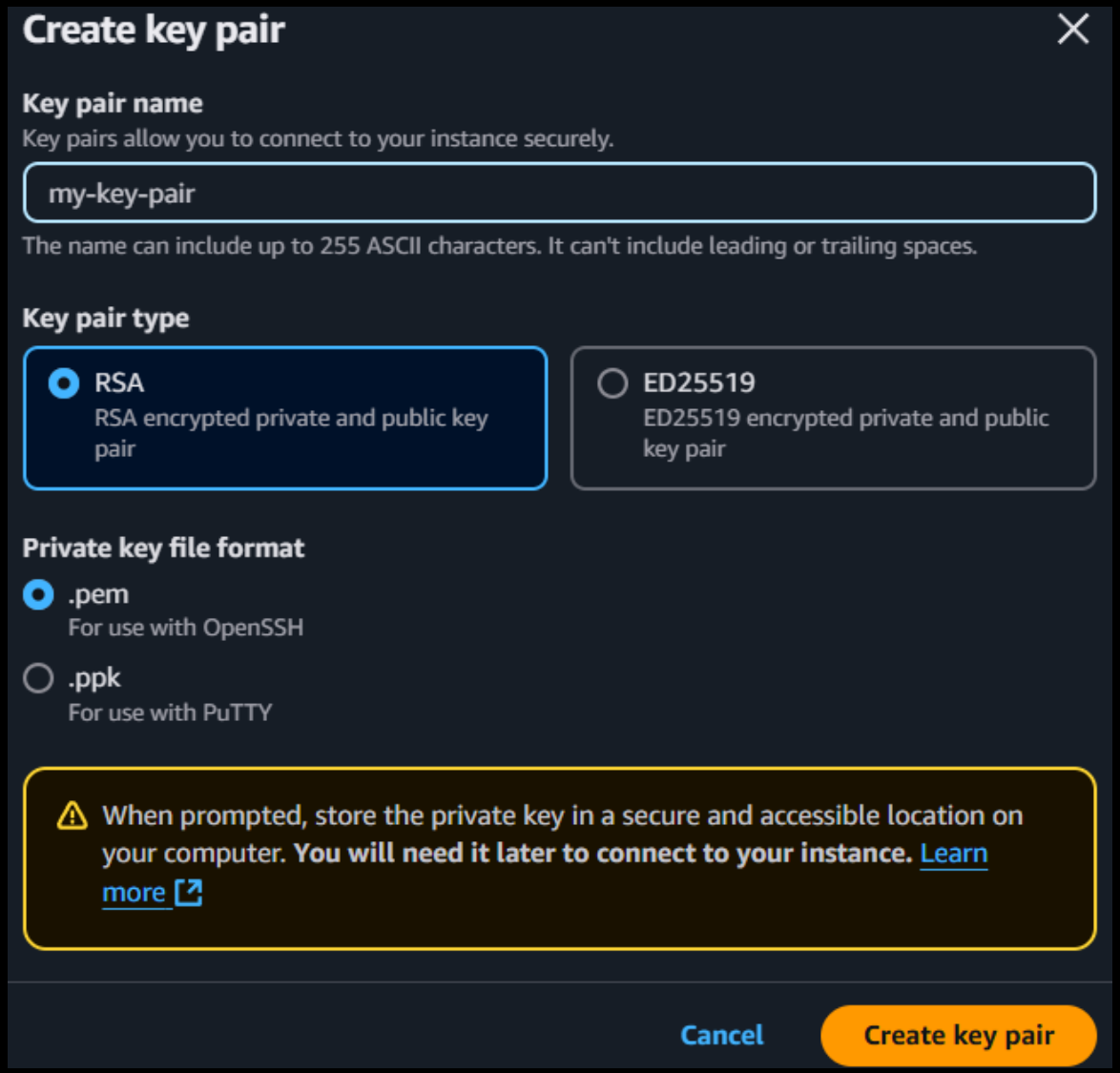
6. **Choose an Instance Type**: Select **t2.micro**, which is perfect for beginners and falls within the free tier offering 🎁.



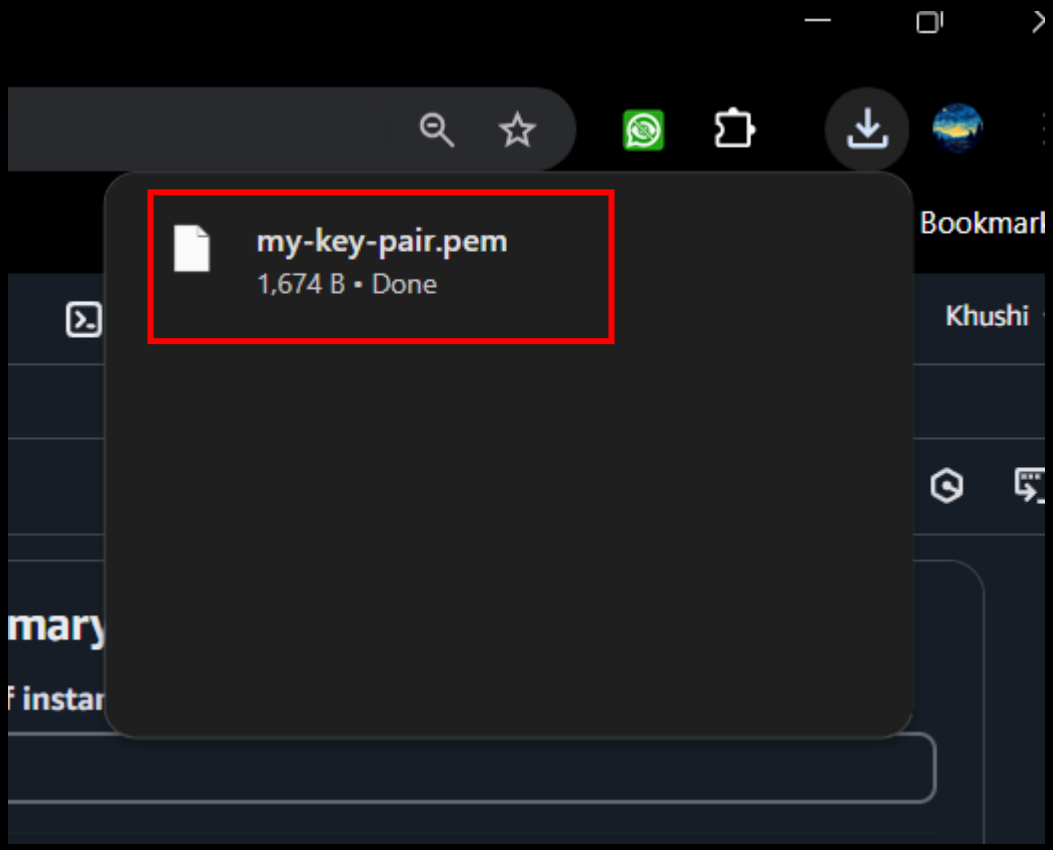
7. In the **Key Pair** section, click on **Create a new pair** to generate a new key pair for secure access. 🔑



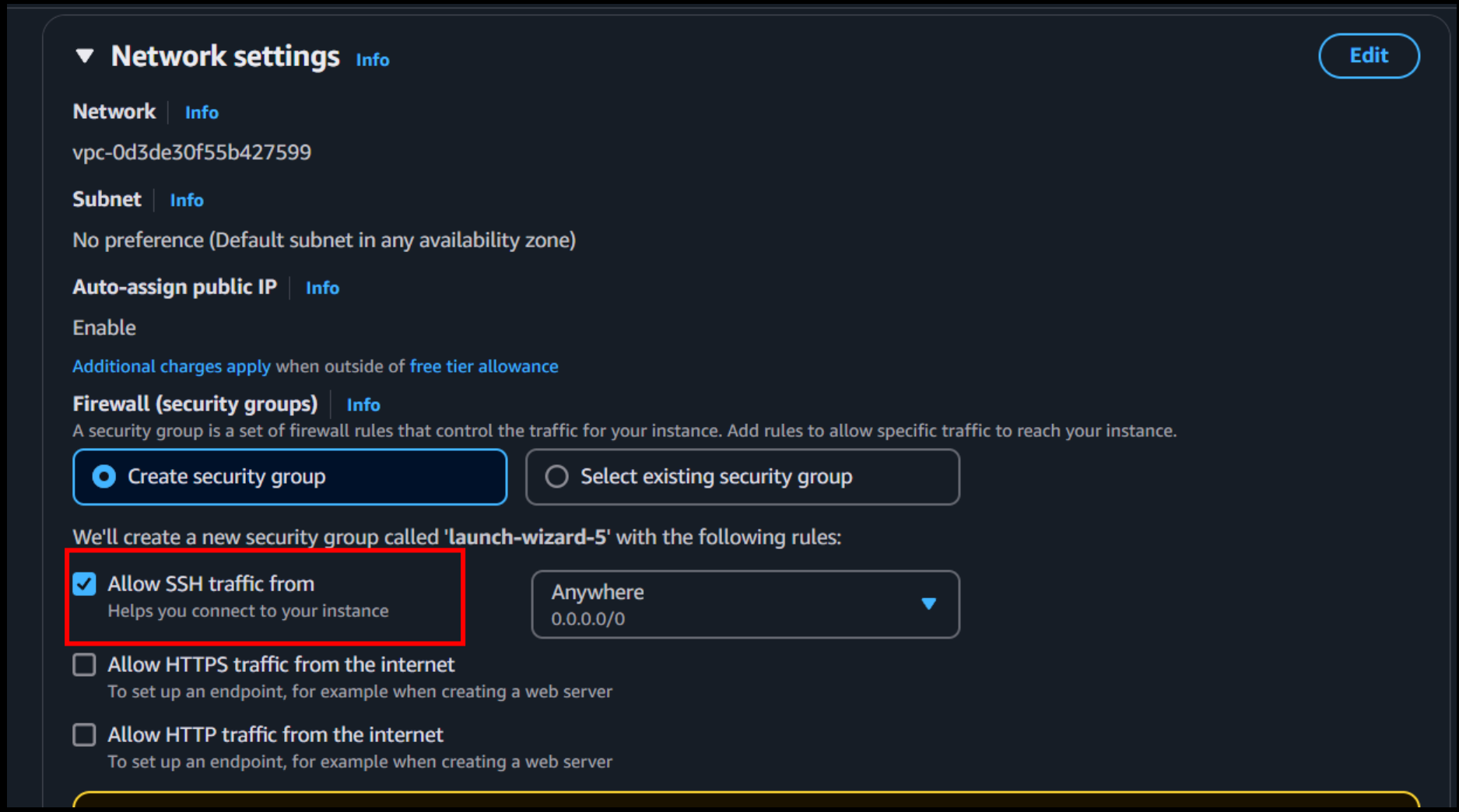
8. Give your key pair a name (e.g., **my-key-pair**) 📜



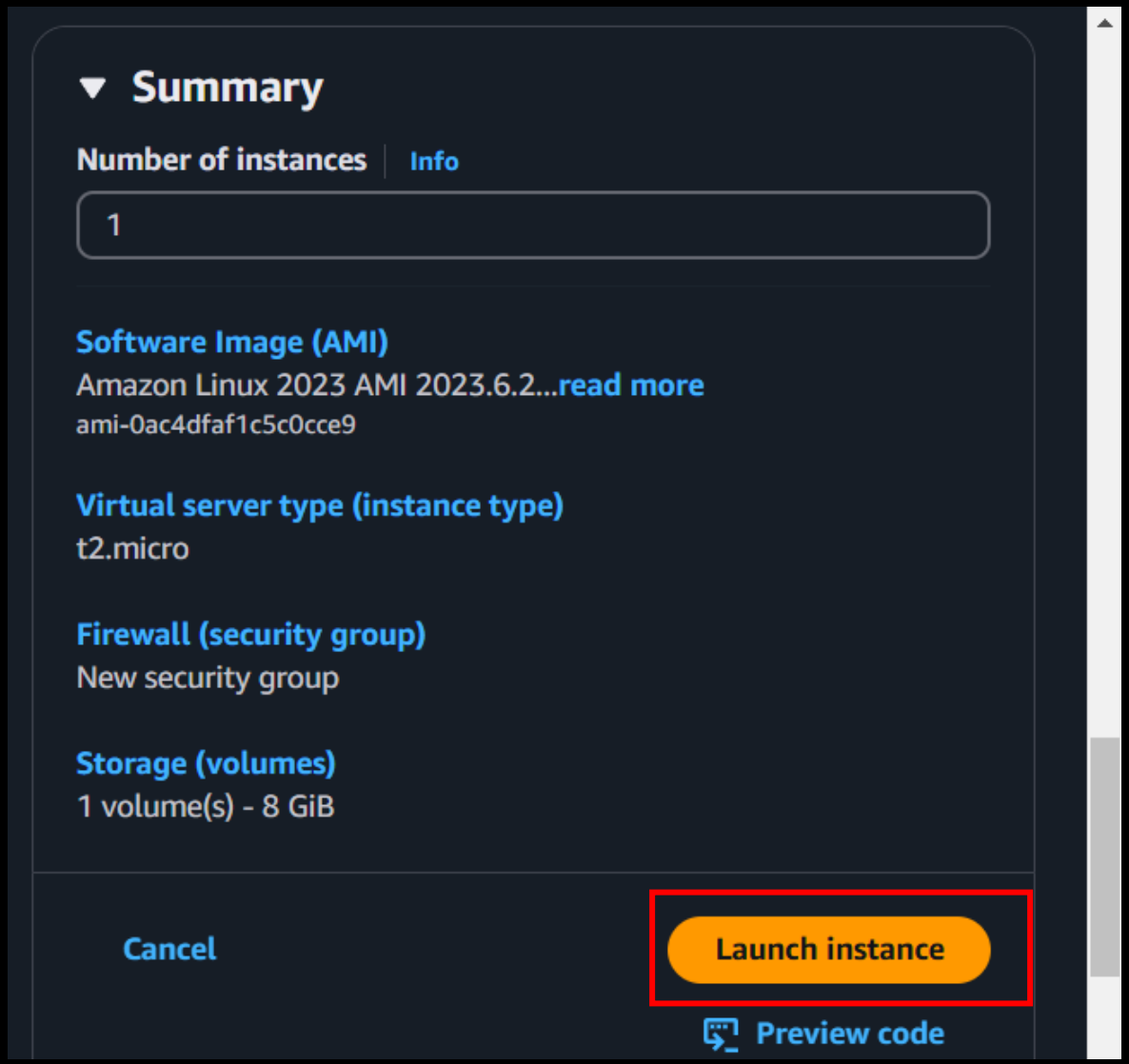
9. Hit **Create key pair** and don't forget to **download the private key (.pem file)** when prompted. You'll need this to access your server securely! 🔑



10. Configure your **Security Group**, just make sure **SSH** (port 22) is allowed. 🔐



11. See the instance **Summary** and click on **launch Instance** 📑



What is PuTTYgen? 🔑

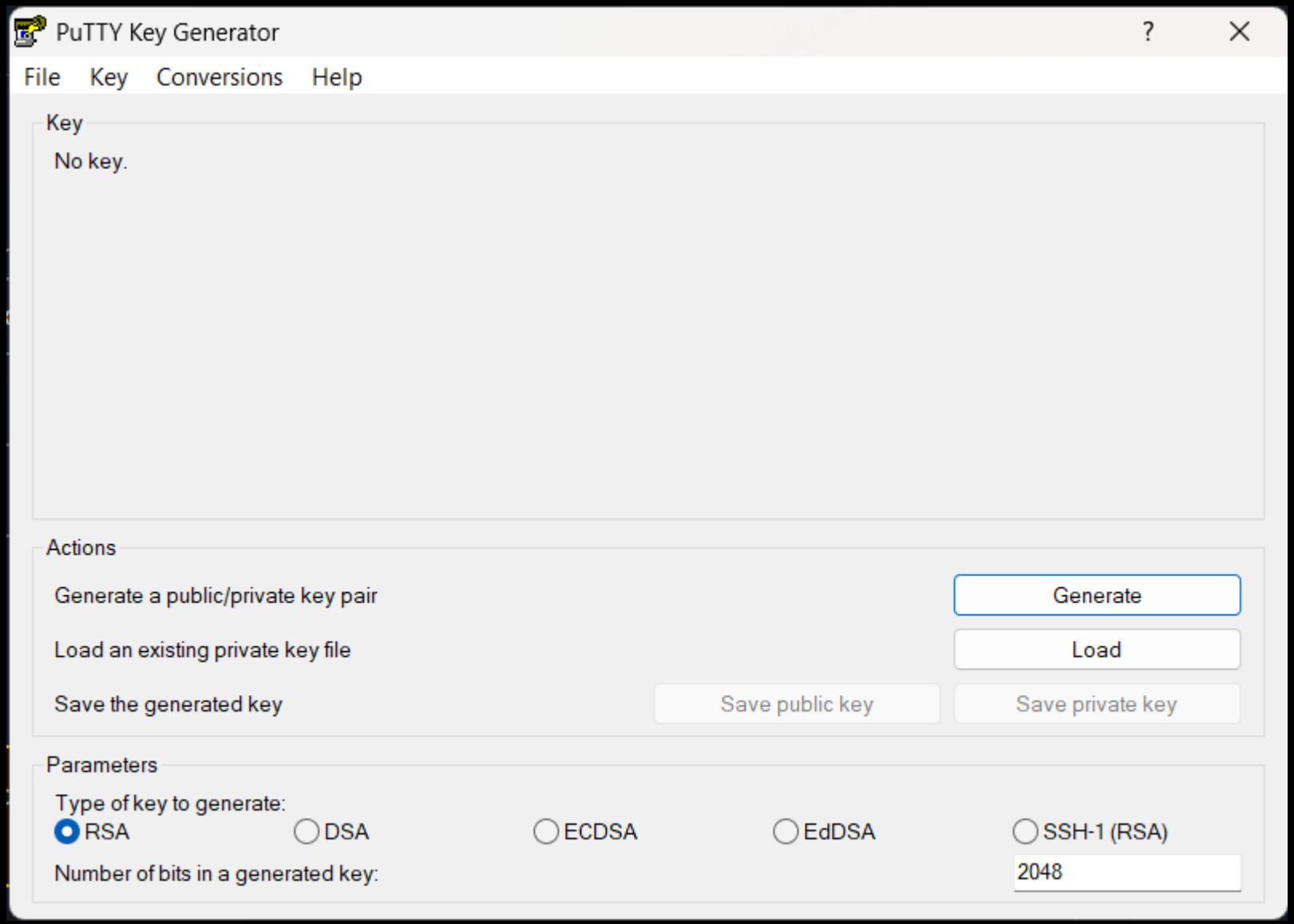
PuTTYgen is a tool used to generate and convert SSH key pairs. It allows you to convert AWS EC2 key pairs **(.pem files)** into PuTTY-compatible format **(.ppk)**, which is required for securely connecting to your server via PuTTY. 🔒💻

Why We Use PuTTYgen? 🔑

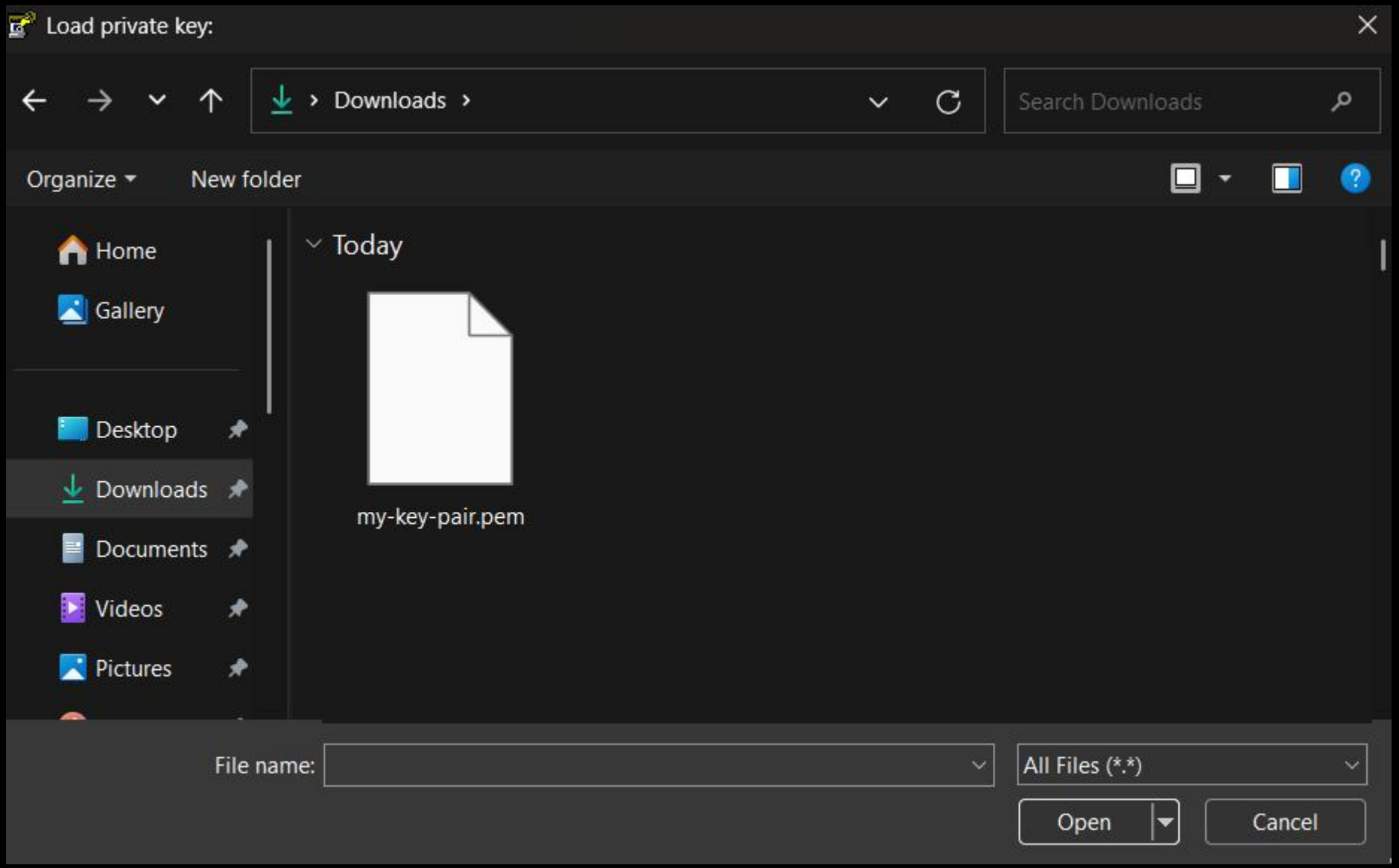
We use **PuTTYgen** to convert AWS EC2 key pairs **(.pem files)** into a PuTTY-compatible format **(.ppk)**, allowing us to securely connect to a remote Linux server using **PuTTY.** Without this conversion, PuTTY cannot authenticate the connection. 🔐💻

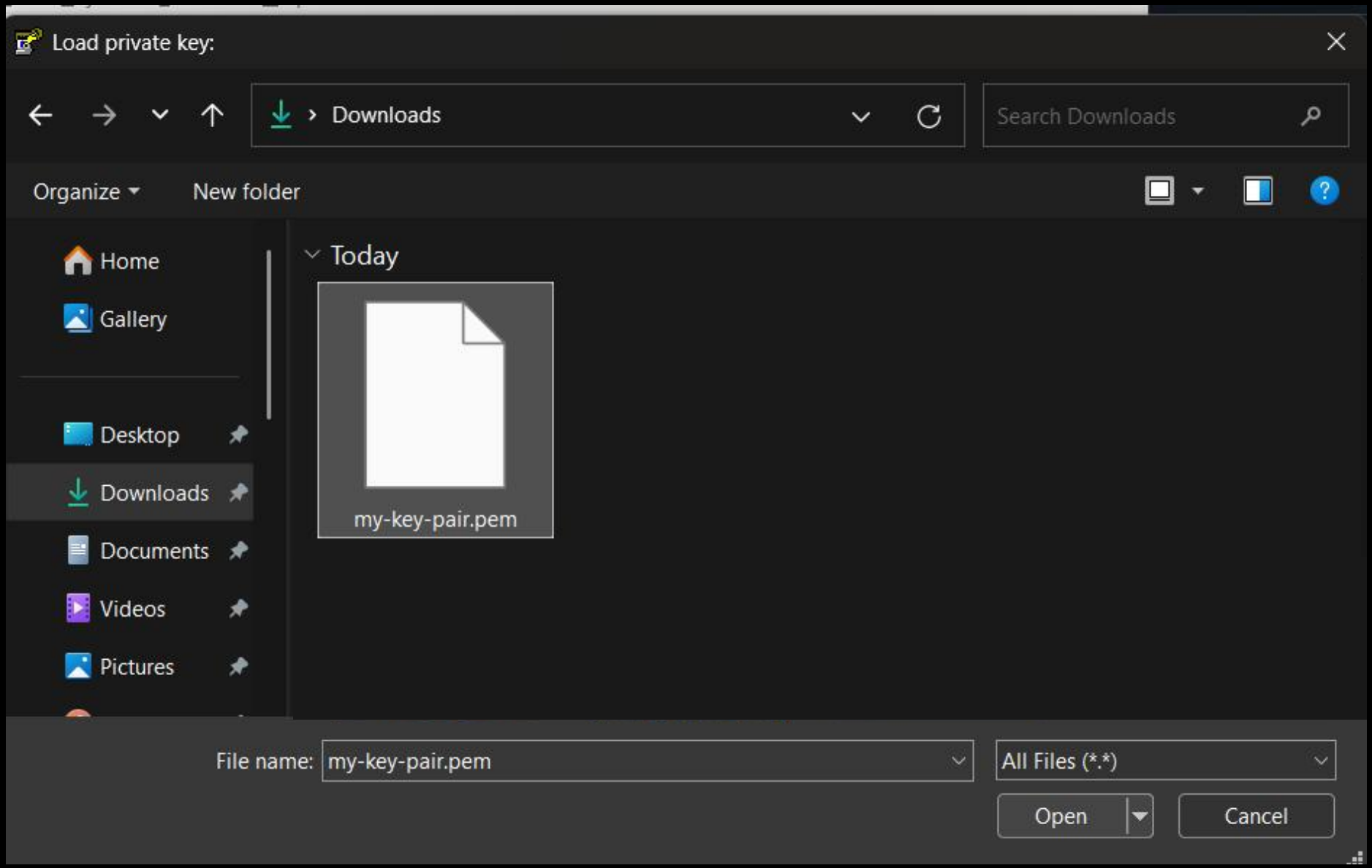
II. CONVERT YOUR .PEM FILE TO .PPK USING PUTTYGEN 🔑

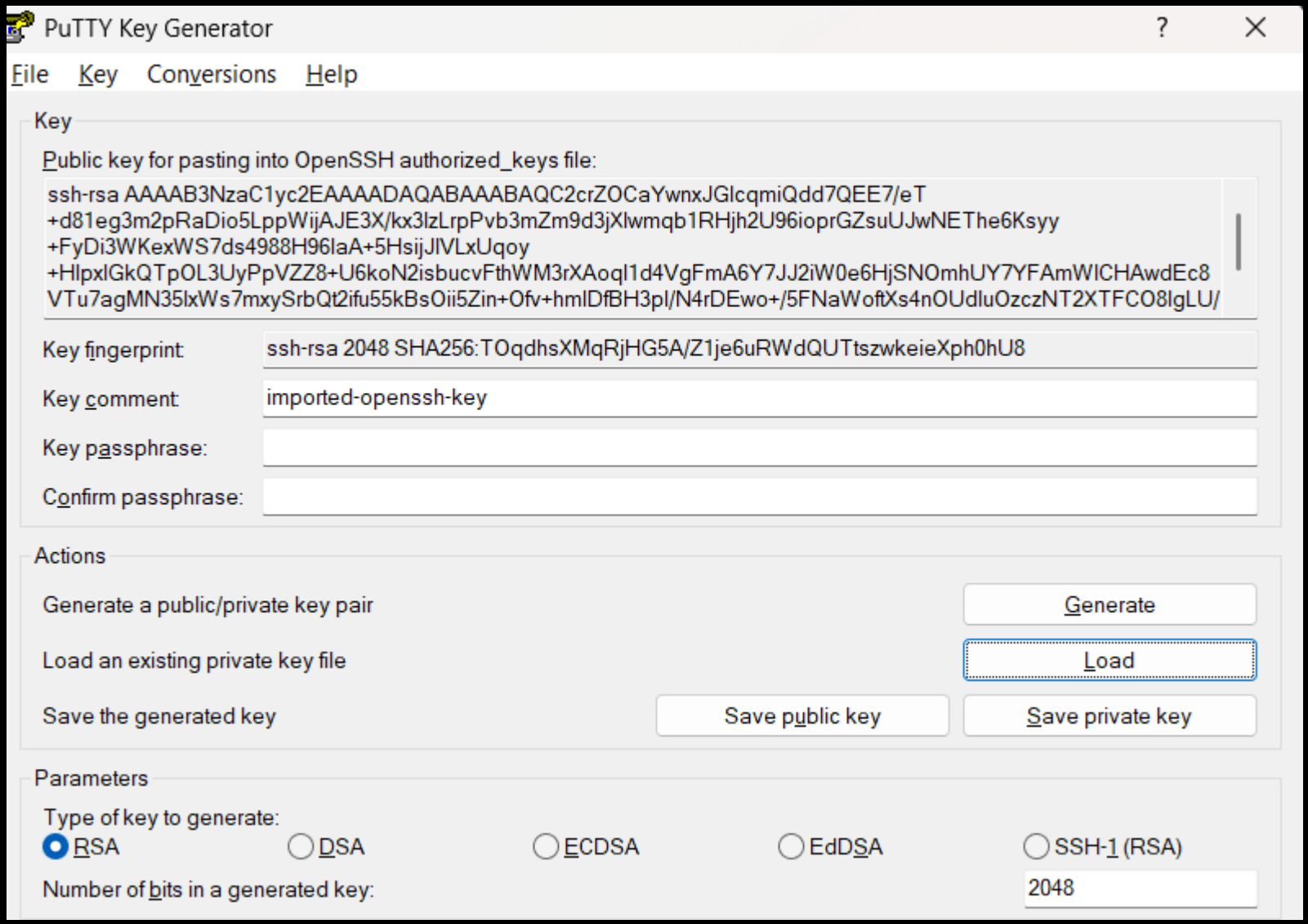
1. Open **PuTTYgen**, your magical key converter! ✨

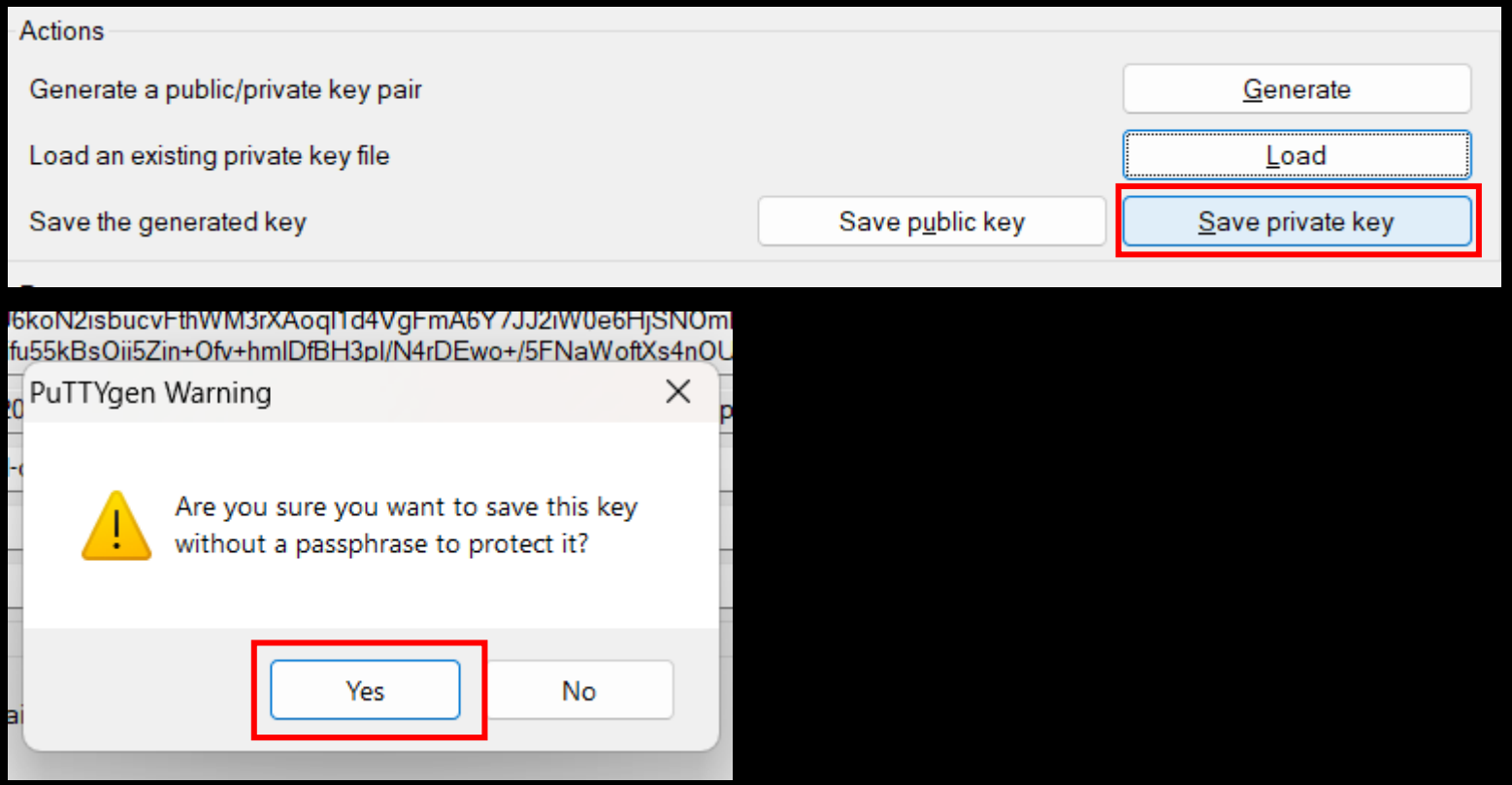

PuTTY Key Generator
File
Key
Conversions
Help
Key
No key.
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048

2. Click **Load** and select your freshly downloaded **.pem file.** 🔓

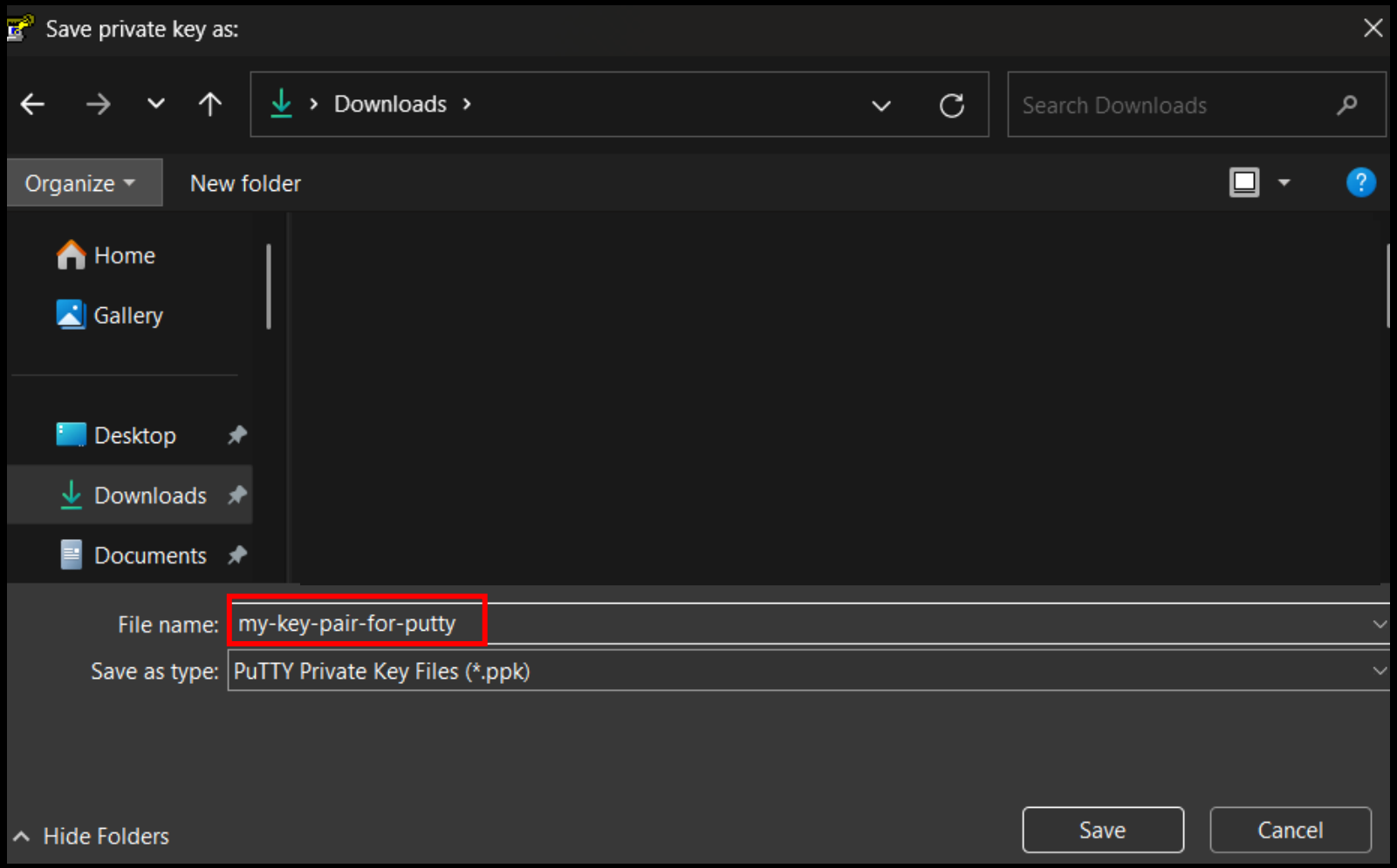

Load private key:
Downloads
Search Downloads
Organize
New folder
Home
Gallery
Desktop
Downloads
Documents
Videos
Pictures
Today
my-key-pair.pem
File name:
All Files (*.*)
Open
Cancel

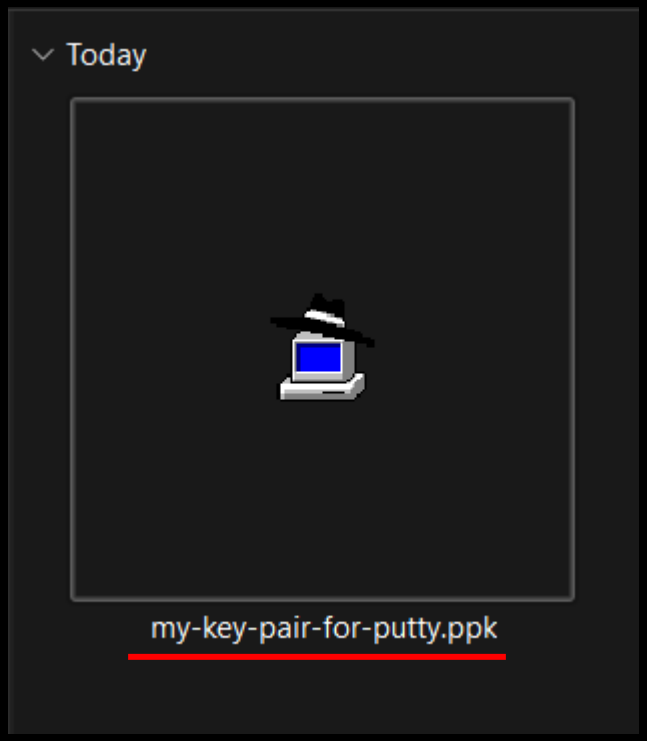

Load private key:
Downloads
Search Downloads
Organize
New folder
Home
Gallery
Desktop
Downloads
Documents
Videos
Pictures
Today
my-key-pair.pem
File name:
my-key-pair.pem
All Files (*.*)
Open
Cancel


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC2crZOCaYwnxJGlcqmiQdd7QEE7/eT
+d81eg3m2pRaDio5LppWijAJE3X/kx3lzLrpPvb3mZm9d3jXlwmqb1RHjh2U96ioprGZsuUJwNEThe6Ksyy
+FyDi3WKexWS7ds4988H96laA+5HsijJlVLxUqoy
+HlpxlGkQTpOL3UyPpVZZ8+U6koN2isbucvFthWM3rXAoql1d4VgFmA6Y7JJ2iW0e6HjSNOmhUY7YFAmWlCHAwdEc8
VTu7agMN35lxWs7mxySrbQt2ifu55kBsOii5Zin+Ofv+hmlDfBH3pl/N4rDEwo+/5FNaWoftXs4nOUdluOzczNT2XTFCO8lgLU/
Key fingerprint:
ssh-rsa 2048 SHA256:TOqdhsXMqRjHG5A/Z1je6uRWdQUTtszwkeieXph0hU8
Key comment:
imported-openssh-key
Key passphrase:
Confirm passphrase:
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048


Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
PuTTYgen Warning
Are you sure you want to save this key without a passphrase to protect it?
Yes
No

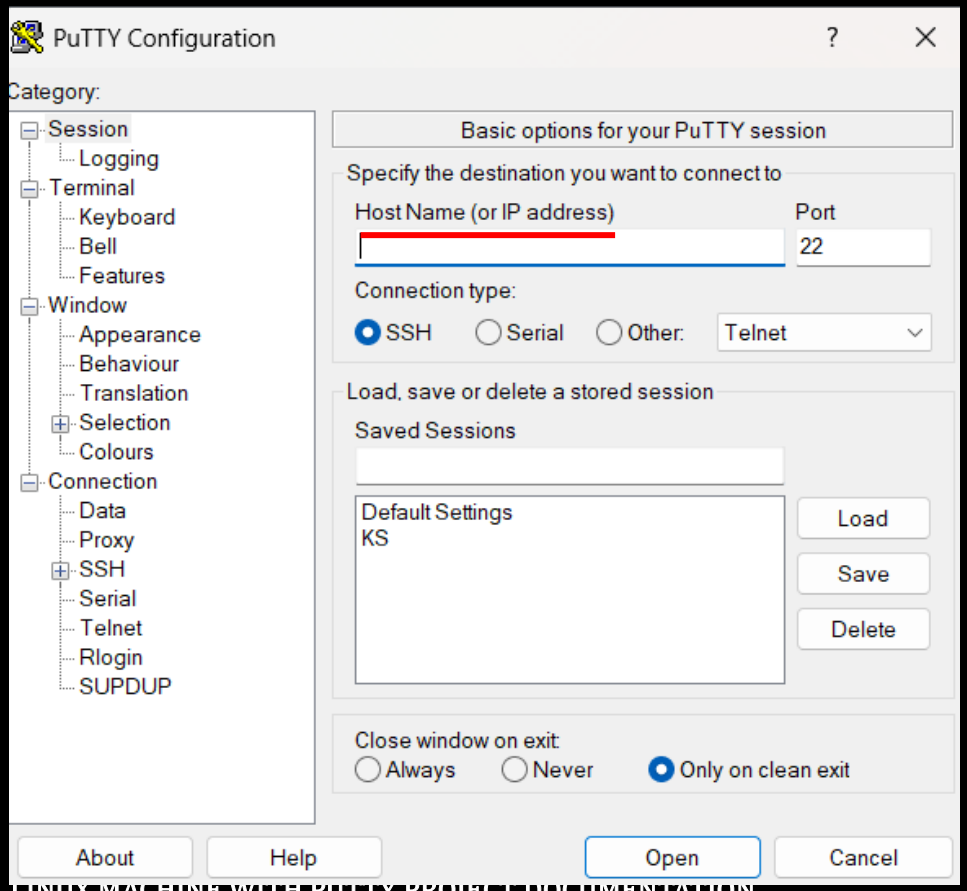
3. Save the private key as a **.ppk** file (this is PuTTY's preferred format). 🔑

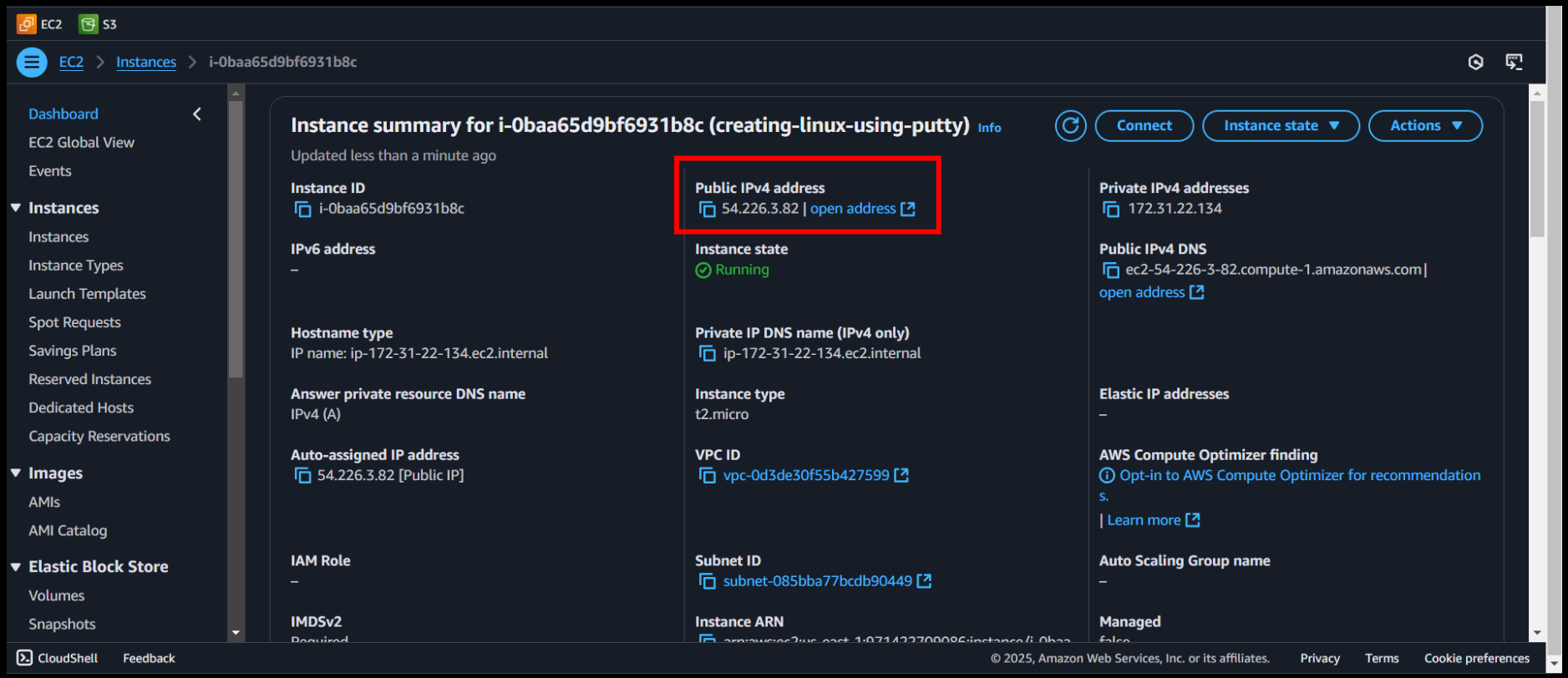


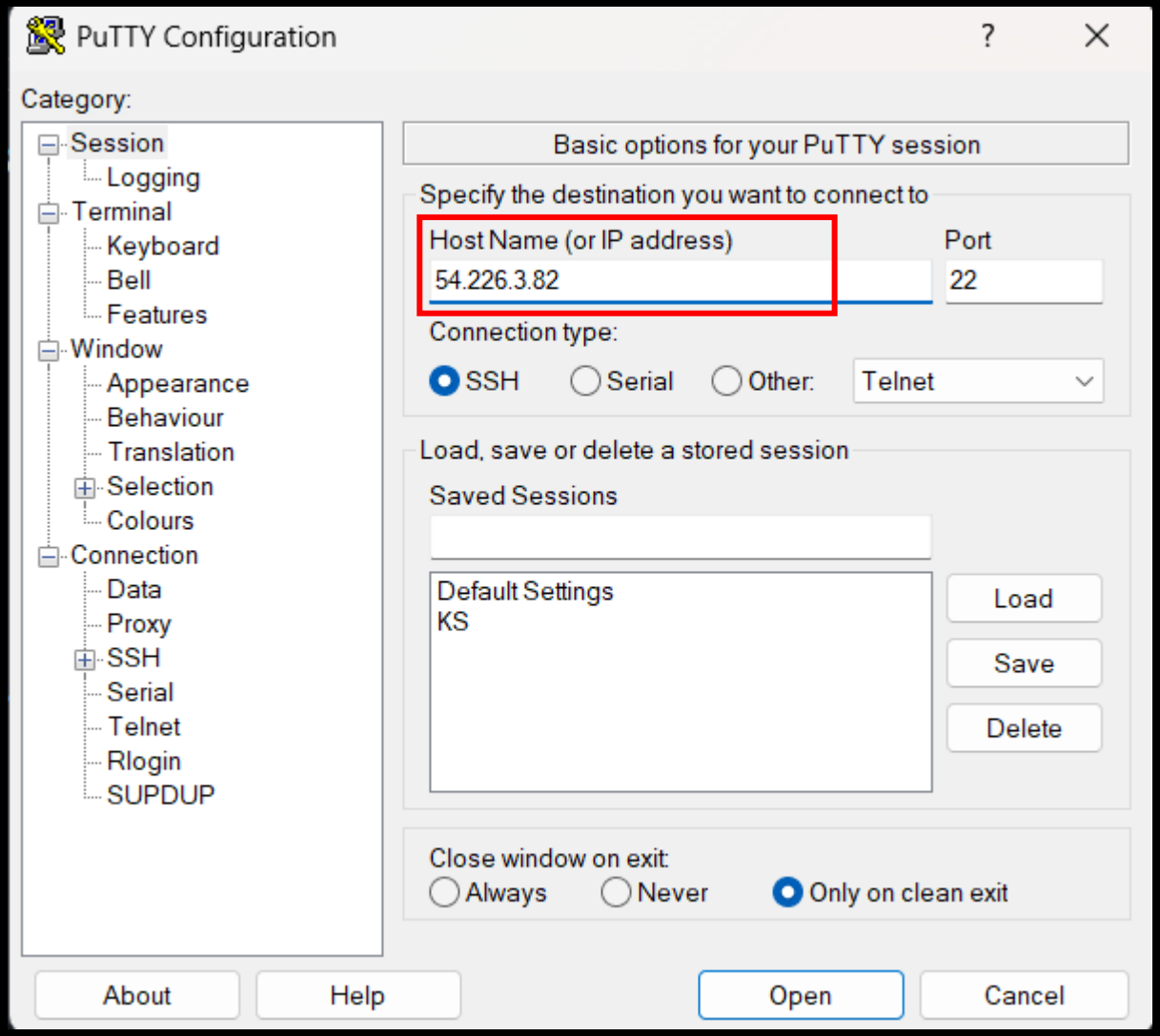

Today
my-key-pair-for-putty.ppk

III. CONNECT TO YOUR EC2 INSTANCE WITH PUTTY 💻

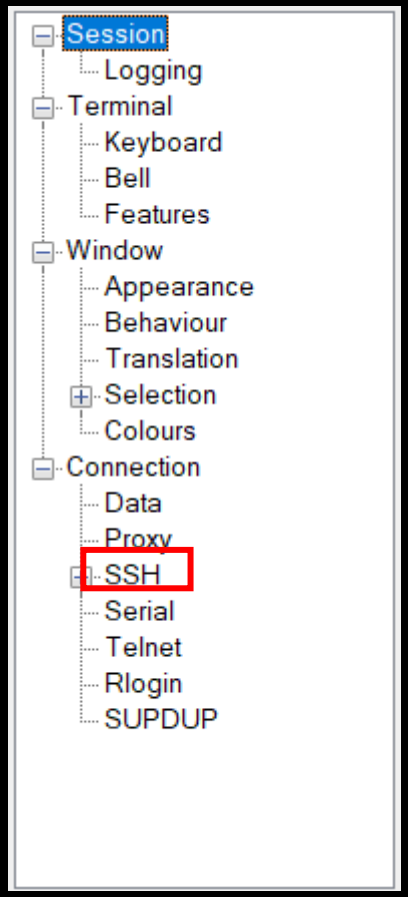
1. Open **PuTTY** and get ready to teleport into your EC2 instance. 🌐
2. In the **Host Name** field, enter the **Public IP Address** or **Public DNS** of your EC2 instance. 🚀

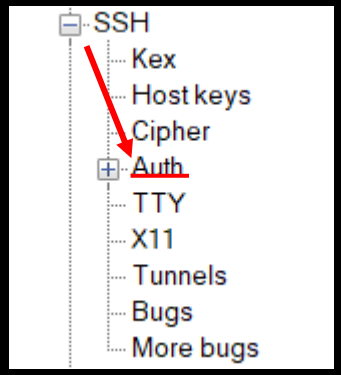


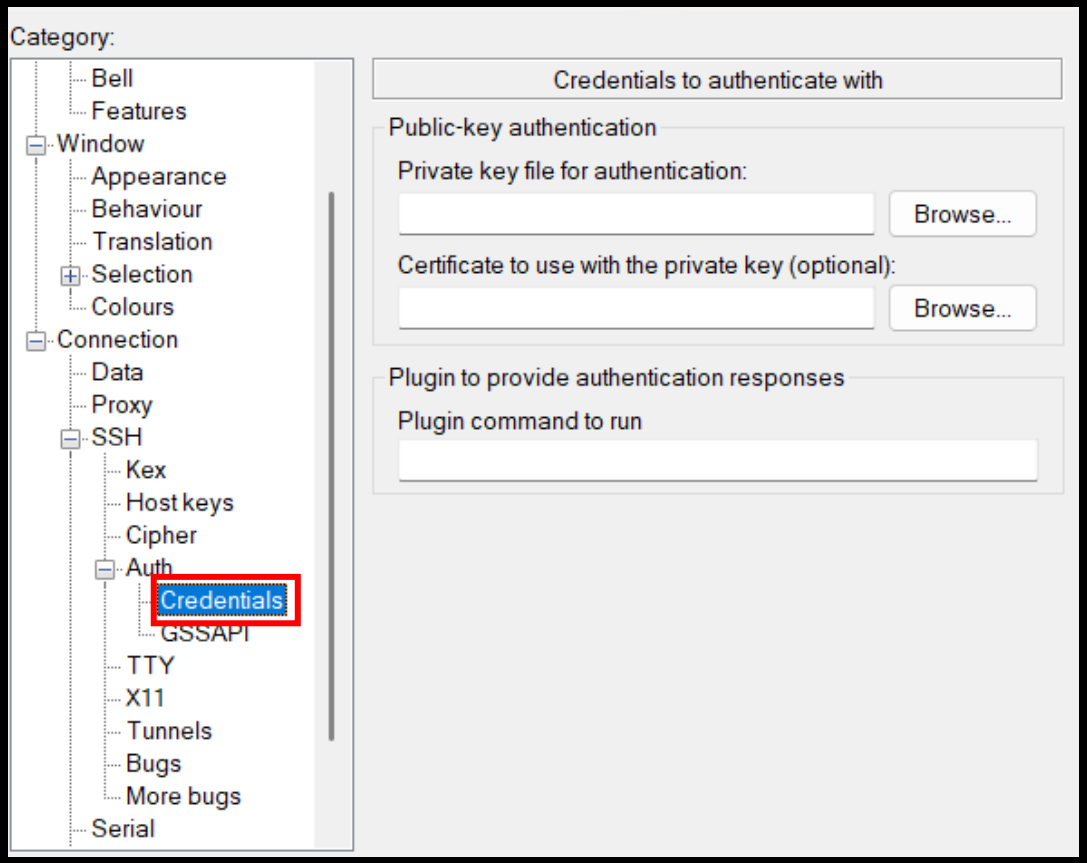

EC2
S3
EC2 > Instances > i-0baa65d9bf6931b8c
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Instance summary for i-0baa65d9bf6931b8c (creating-linux-using-putty) Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-0baa65d9bf6931b8c
Public IPv4 address
54.226.3.82 | open address
Private IPv4 addresses
172.31.22.134
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-54-226-3-82.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-172-31-22-134.ec2.internal
Private IP DNS name (IPv4 only)
ip-172-31-22-134.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–
Auto-assigned IP address
54.226.3.82 [Public IP]
VPC ID
vpc-0d3de30f55b427599
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
–
Subnet ID
subnet-085bba77bcdb90449
Auto Scaling Group name
–
IMDSv2
Instance ARN
Managed
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


PuTTY Configuration
Category:
Session
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Serial
Telnet
Rlogin
SUPDUP
Basic options for your PuTTY session
Specify the destination you want to connect to
Host Name (or IP address)
Port
54.226.3.82
22
Connection type:
SSH
Serial
Other:
Telnet
Load, save or delete a stored session
Saved Sessions
Default Settings
KS
Load
Save
Delete
Close window on exit:
Always
Never
Only on clean exit
About
Help
Open
Cancel

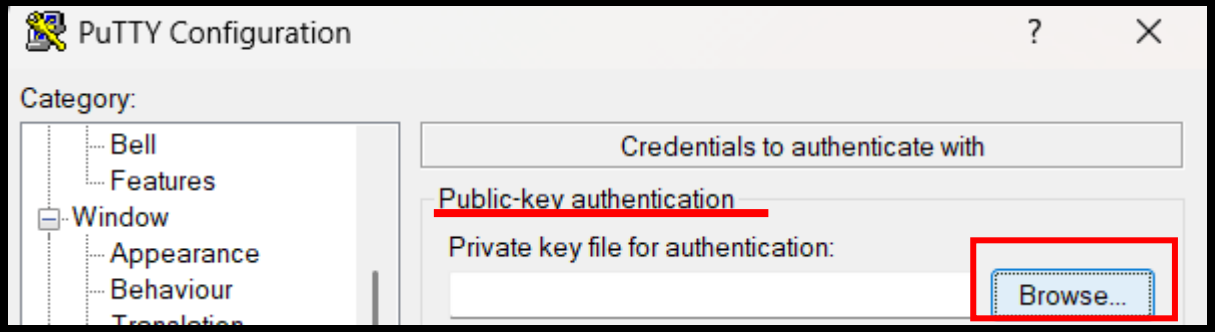
3. Go to **SSH > Auth** on the left menu. 📂

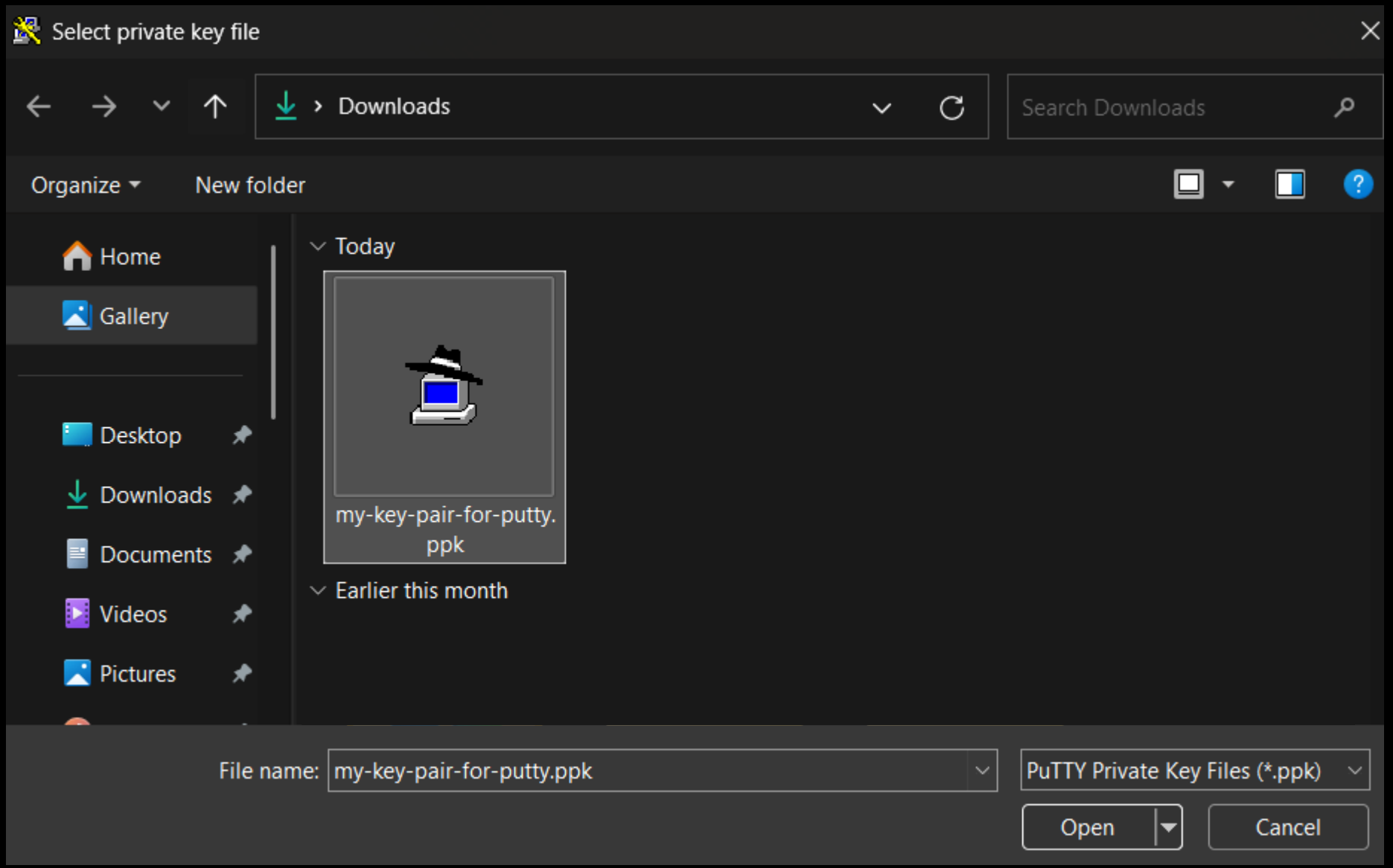


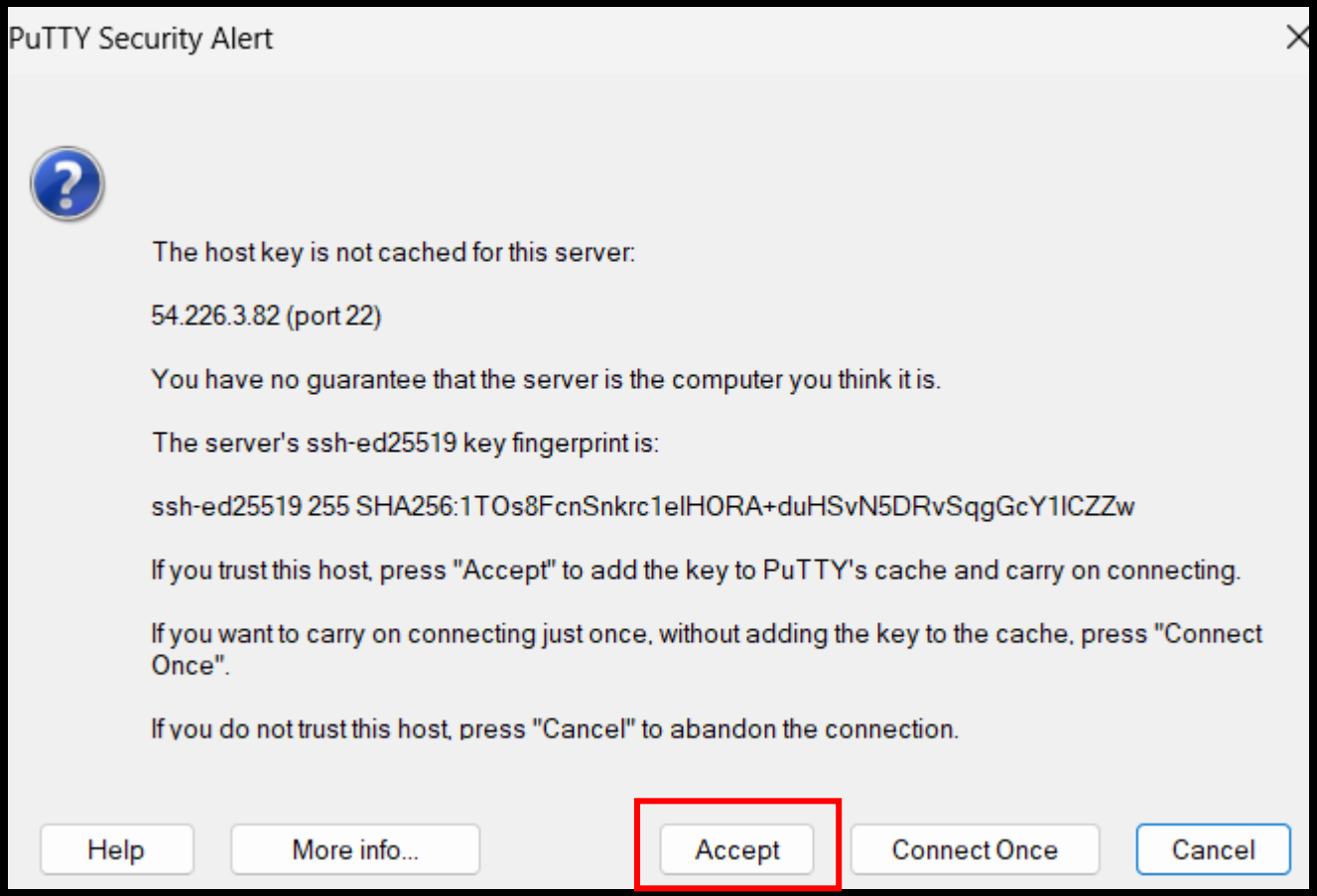



Category:
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
Credentials
GSSAPI
TTY
X11
Tunnels
Bugs
More bugs
Serial
Credentials to authenticate with
Public-key authentication
Private key file for authentication:
Browse...
Certificate to use with the private key (optional):
Browse...
Plugin to provide authentication responses
Plugin command to run

4. Under **Private key file for authentication**, browse and select your **.ppk file.** 🔑




Select private key file
Downloads
Search Downloads
Organize
New folder
Home
Gallery
Desktop
Downloads
Documents
Videos
Pictures
Today
my-key-pair-for-putty.ppk
Earlier this month
File name:
my-key-pair-for-putty.ppk
PuTTY Private Key Files (*.ppk)
Open
Cancel

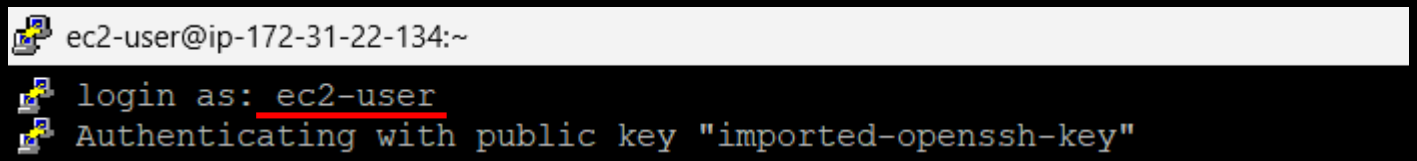

PuTTY Security Alert
The host key is not cached for this server:
54.226.3.82 (port 22)
You have no guarantee that the server is the computer you think it is.
The server's ssh-ed25519 key fingerprint is:
ssh-ed25519 255 SHA256:1TOs8FcnSnkrc1elHORA+duHSvN5DRvSqgGcY1lCZZw
If you trust this host, press "Accept" to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, press "Connect Once".
If you do not trust this host, press "Cancel" to abandon the connection.
Help
More info...
Accept
Connect Once
Cancel

5. Hit **Open** and voilà, you're almost there! 🎉

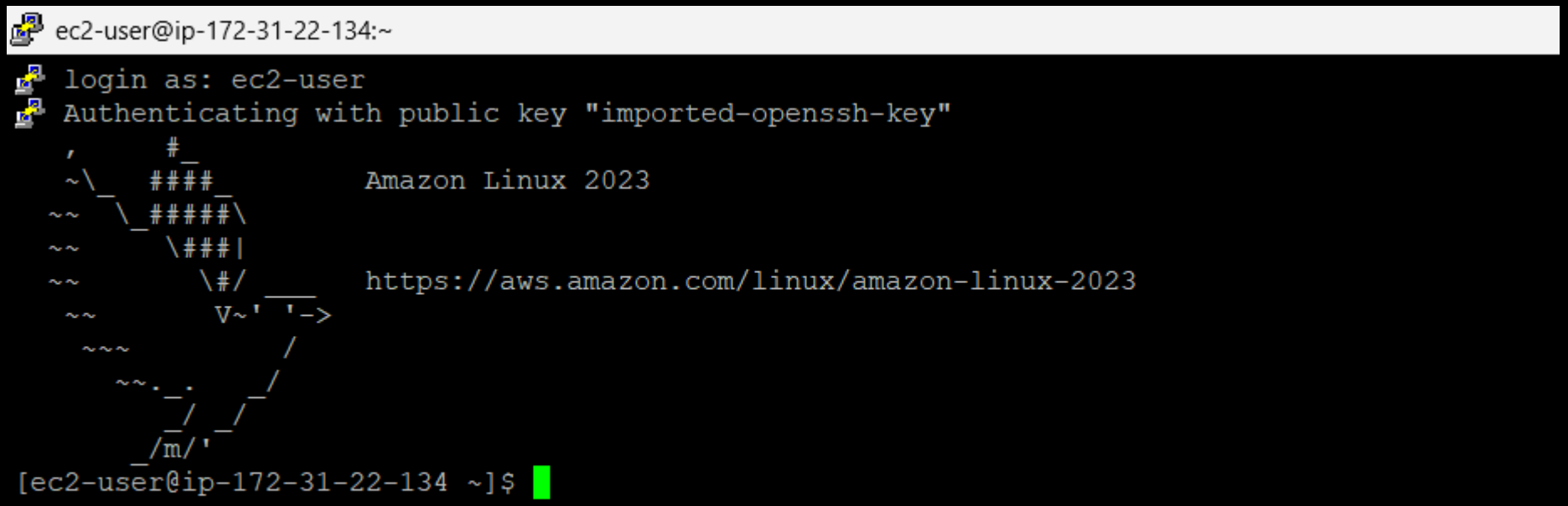
IV. LOGIN LIKE A PRO 👤

1. Depending on your AMI, use the **default username** to log in: -

- **Amazon Linux**: ec2-user 🐧
- **Ubuntu**: ubuntu 🌱
- **CentOS**: centos 💻



2. You’re officially inside your shiny new Linux machine! 🖥️ ✨



V. VERIFY YOUR CONNECTION 🖥️

1. Let's check if everything is working smoothly! 🌟
   Here’s what you can do after logging in:

   - **Switch to superuser** by running:

```
       _/m/'
[ec2-user@ip-172-31-22-134 ~]$ sudo su
[root@ip-172-31-22-134 ec2-user]#
```

   This grants you administrative privileges. 🔧

   - **Create a new directory** named **AWS**:

```
       _/m/'
[ec2-user@ip-172-31-22-134 ~]$ sudo su
[root@ip-172-31-22-134 ec2-user]# mkdir AWS
```

   This will create a folder to keep things organized. 📂

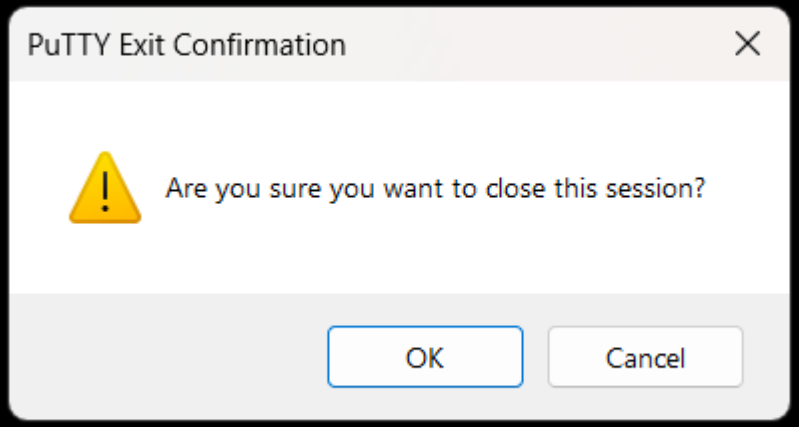
- **Create a file called f1 and append content using cat:**

```
[root@ip-172-31-22-134 ec2-user]# cat >> f1
Hello, we just saw how to create a linux instance using putty
[root@ip-172-31-22-134 ec2-user]# cat f1
Hello, we just saw how to create a linux instance using putty
```

This ensures that everything is working smoothly and you're set up on your Linux instance! 🎉

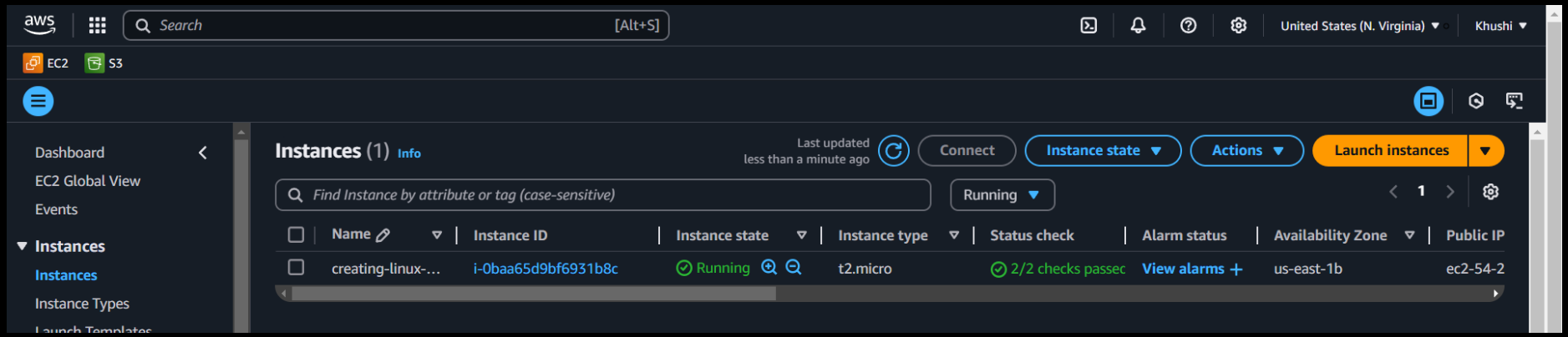
VI. CLEANUP AND EXIT 🧹

1. **Exit the session** cleanly:
    - If you're done, you can **close the PuTTY window** or **select the "X" mark** in the top-right corner to terminate the session. Confirm the action if prompted.

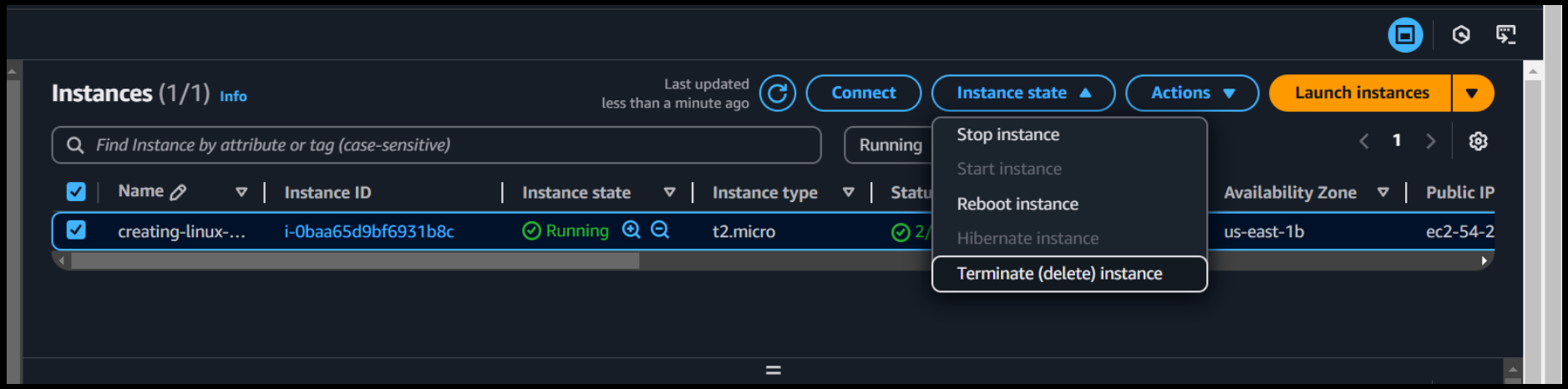


2. Delete the EC2 Instance:

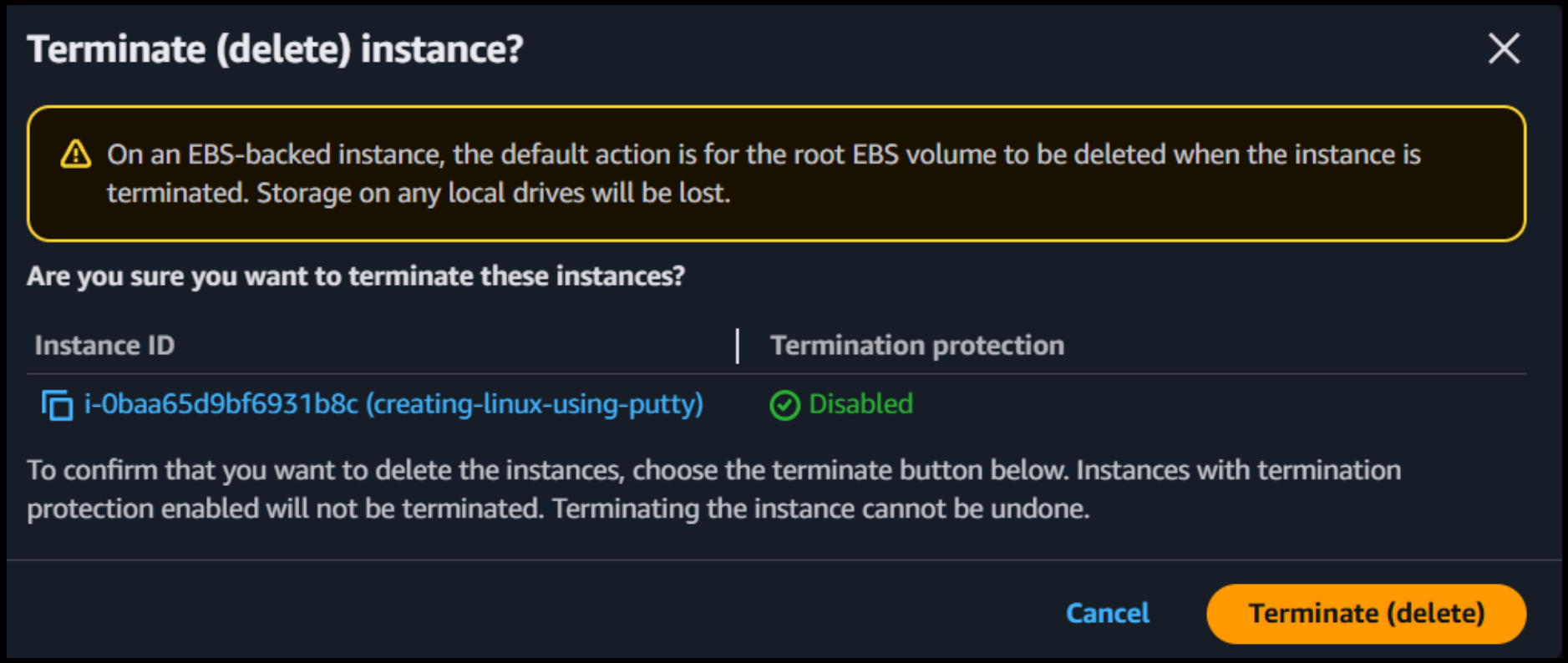
- Go back to your **AWS Management Console.**
- Navigate to **EC2** and find the instance you created earlier.



- Select your **EC2 instance** and click **Terminate** from the **Actions** dropdown.



- Confirm that you want to terminate the instance. It will stop running, and you won't be charged for the instance anymore. 🚫 💸



CONGRATULATIONS! YOU DID IT 🎉

Congrats! 🎉 You've successfully launched your Linux instance on AWS and connected like a pro using PuTTY! 🚀 Now you're ready to explore the power of cloud computing and customize your environment to your needs. 🌥️

The cloud is not just a place, it's the future of limitless possibilities. ☁️