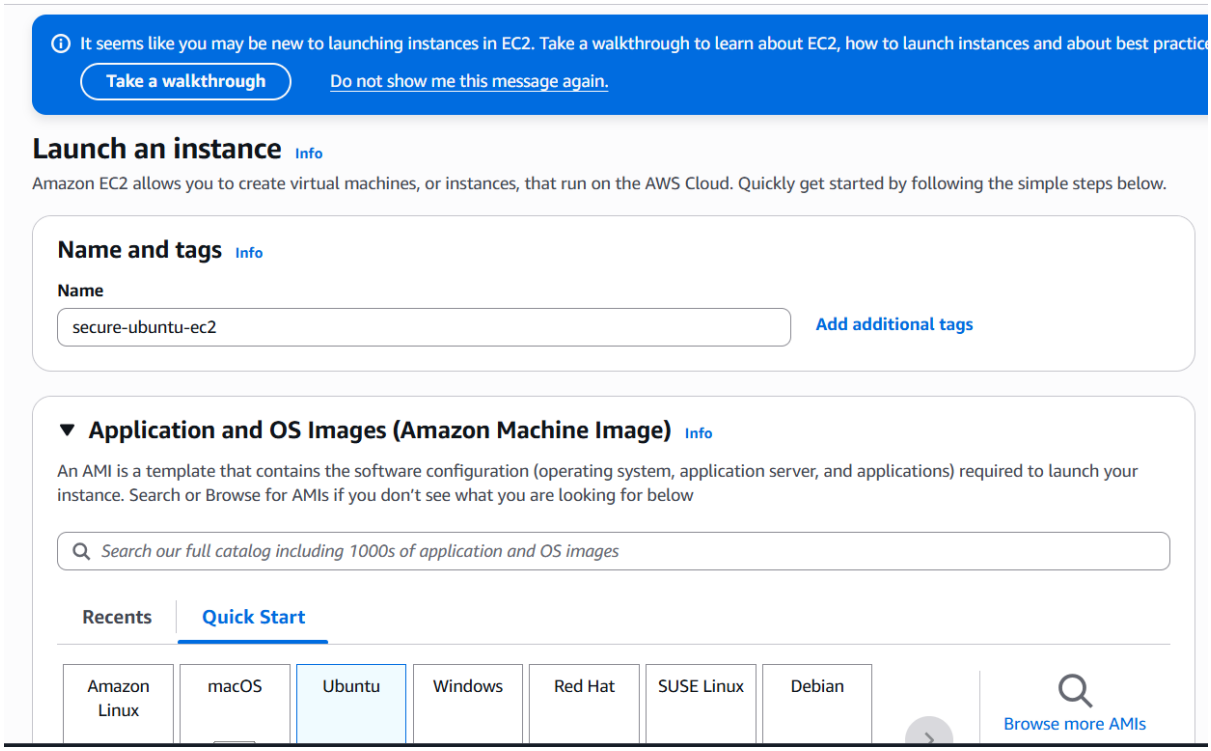
**EC2 Instance Security Hardening Project**

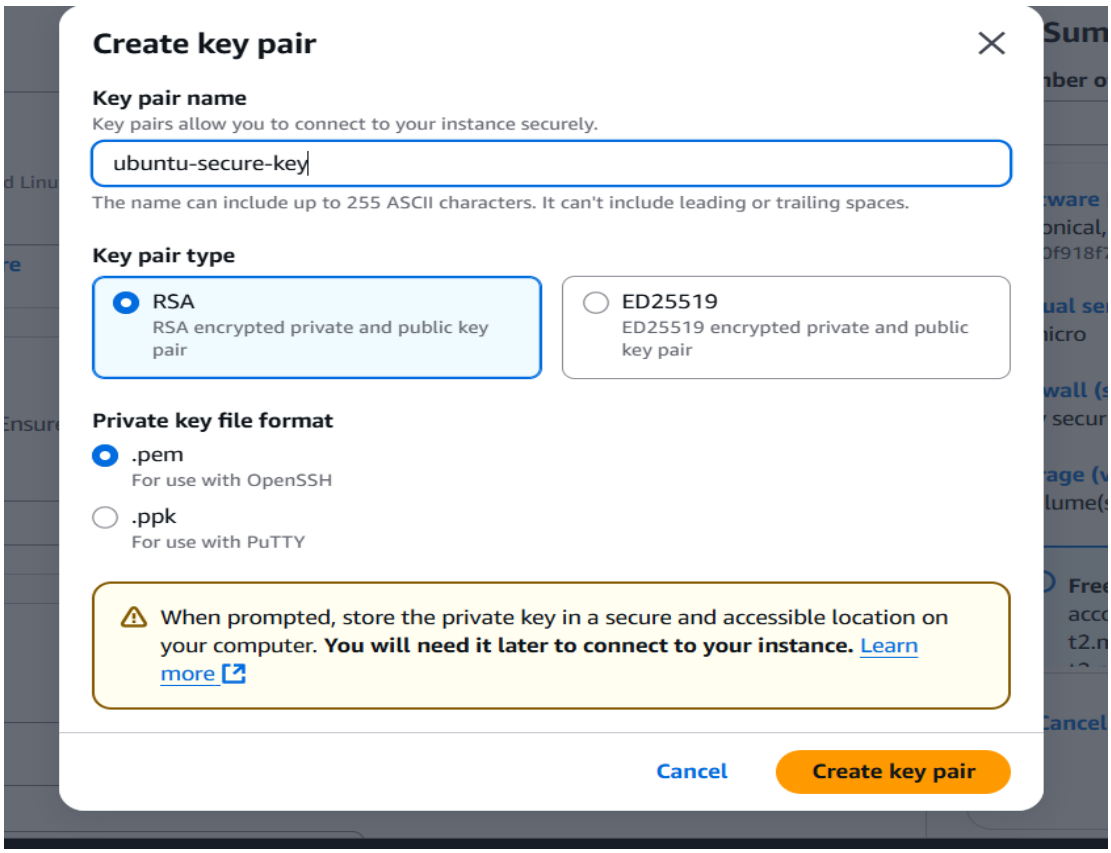
**Project Type:**
Hands-on AWS Cloud Security Practice

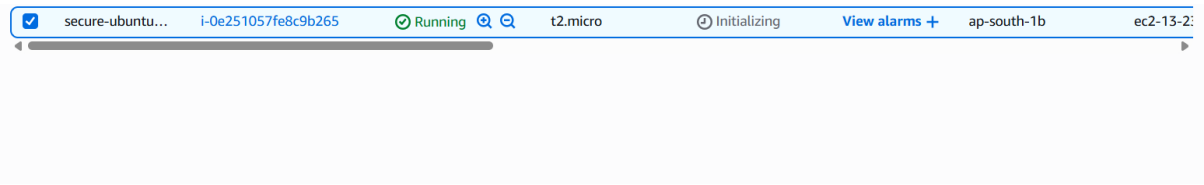
**Project Duration:**
1 Day (Completed on: July 17, 2025)

**Tools & Services Used:**

- AWS EC2
- Ubuntu 24.04
- SSH
- UFW Firewall
- Fail2Ban
- Git Bash (for secure key usage)


It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practic
Take a walkthrough
Do not show me this message again.
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
secure-ubuntu-ec2
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


Create key pair
Key pair name
Key pairs allow you to connect to your instance securely.
ubuntu-secure-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

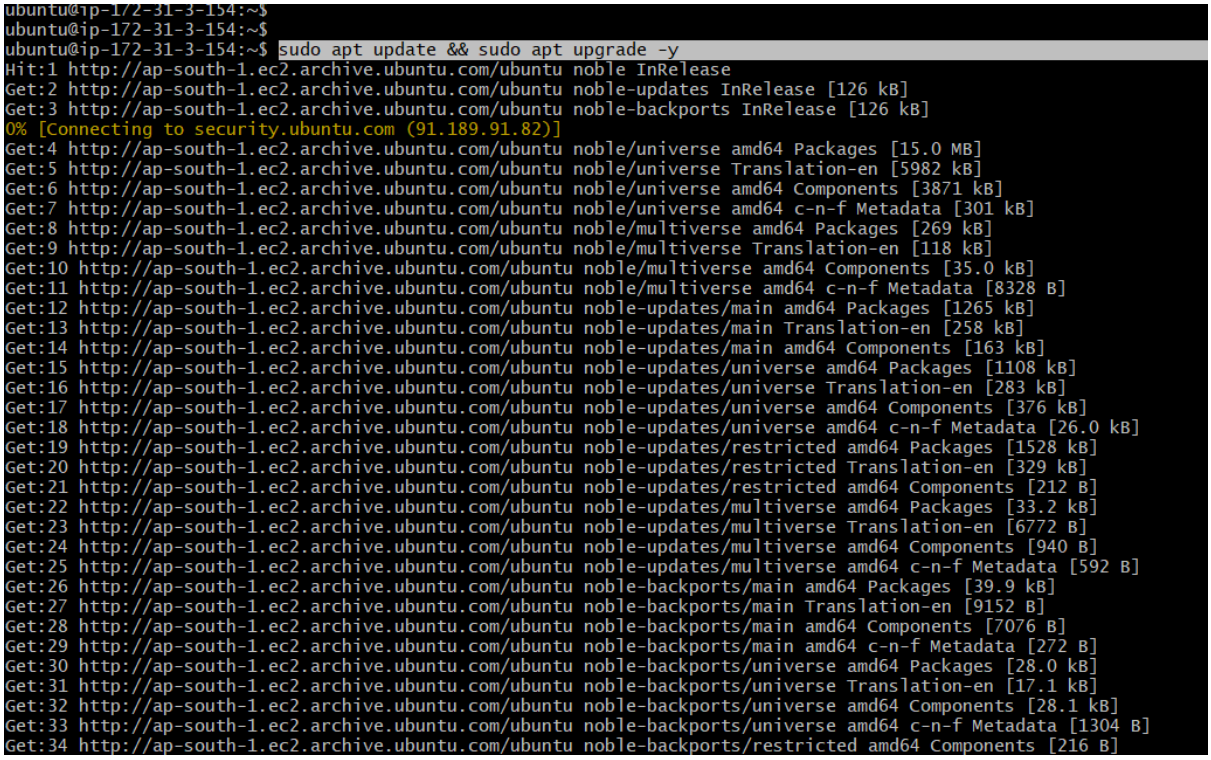



Launched an EC2 Ubuntu instance with a secure key pair and restricted SSH access using a custom security group.

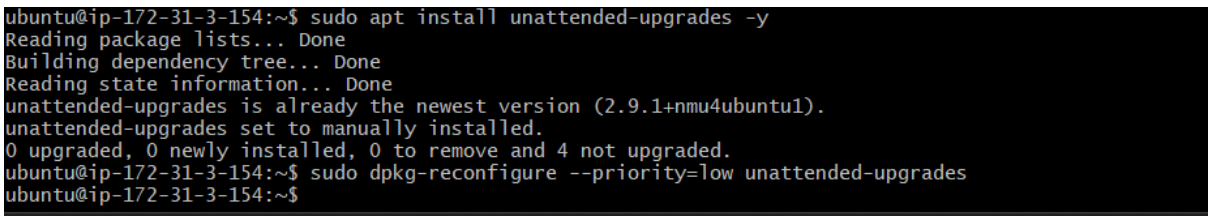
```
p@LAPTOP-GKL13PH0 MINGW64 ~/Downloads
 ^[[200~chmod 400 ubuntu-secure-key.pem
ash: $'\E[200~chmod': command not found

p@LAPTOP-GKL13PH0 MINGW64 ~/Downloads
 chmod 400 ubuntu-secure-key.pem

p@LAPTOP-GKL13PH0 MINGW64 ~/Downloads
 ^[[200~ssh -i ubuntu-secure-key.pem ubuntu@<your-ec2-public-ip>
ash: syntax error near unexpected token `newline'

p@LAPTOP-GKL13PH0 MINGW64 ~/Downloads
 ssh -i ubuntu-secure-key.pem ubuntu@13.232.235.170
he authenticity of host '13.232.235.170 (13.232.235.170)' can't be established.
D25519 key fingerprint is SHA256:4XfF+rEjrFdq08v7sej0NaVMyXDZuduobk4iBHp6Y0I.
his key is not known by any other names.
re you sure you want to continue connecting (yes/no/[fingerprint])? Y
lease type 'yes', 'no' or the fingerprint: yes
arning: Permanently added '13.232.235.170' (ED25519) to the list of known hosts.
elcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1029-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:     https://landscape.canonical.com
* Support:        https://ubuntu.com/pro

System information as of Thu Jul 17 15:42:53 UTC 2025

 System load:  0.0               Processes:             104
 Usage of /:   25.5% of 6.71GB   Users logged in:       0
 Memory usage: 20%               IPv4 address for enX0: 172.31.3.154
 Swap usage:   0%


xpanded Security Maintenance for Applications is not enabled.
```

```
See "man sudo_root" for details.

ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
ubuntu@ip-172-31-3-154:~$
```

Successfully connected to the EC2 Ubuntu instance using the secure key pair through SSH. Verified access is limited to my system only for security.




Updated and patched the EC2 Ubuntu system using apt update and apt upgrade to ensure the instance has the latest security fixes.




Installed and enabled unattended-upgrades on the EC2 instance to allow automatic installation of security patches in the background.

Secure the Instance with UFW Firewall

```
ubuntu@ip-172-31-3-154:~$ sudo apt install ufw -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.2-6).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
ubuntu@ip-172-31-3-154:~$ sudo ufw allow OpenSSH
Rules updated
Rules updated (v6)
ubuntu@ip-172-31-3-154:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
ubuntu@ip-172-31-3-154:~$
```

Installed and enabled UFW firewall. Allowed only SSH connections to maintain remote access while blocking all other ports by default.

```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no
```

```
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
Try: sudo apt install <deb name>
ubuntu@ip-172-31-3-154:~$ sudo nano /etc/ssh/sshd_config
ubuntu@ip-172-31-3-154:~$ sudo systemctl restart ssh
ubuntu@ip-172-31-3-154:~$ sudo systemctl restart ssh
ubuntu@ip-172-31-3-154:~$
```

Modified the SSH configuration file to disable root login and password-based authentication, ensuring only key-based access is allowed.

```
ubuntu@ip-172-31-3-154:~$ sudo systemctl restart ssh
ubuntu@ip-172-31-3-154:~$ sudo apt install fail2ban -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyasyncore python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyasyncore python3-pyinotify whois
0 upgraded, 4 newly installed, 0 to remove and 4 not upgraded.
Need to get 496 kB of archives.
After this operation, 2572 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-pyasyncore all 1.0.2-2 [10.1 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 fail2ban all 1.0.2-3ubuntu0.1 [409 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 python3-pyinotify all 0.9.6-2ubuntu1 [25.0 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 whois amd64 5.5.22 [51.7 kB]
Fetched 496 kB in 0s (17.2 MB/s)
Selecting previously unselected package python3-pyasyncore.
(Reading database ... 101281 files and directories currently installed.)
Preparing to unpack .../python3-pyasyncore_1.0.2-2_all.deb ...
Unpacking python3-pyasyncore (1.0.2-2) ...
Selecting previously unselected package fail2ban.
Preparing to unpack .../fail2ban_1.0.2-3ubuntu0.1_all.deb ...
Unpacking fail2ban (1.0.2-3ubuntu0.1) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../python3-pyinotify_0.9.6-2ubuntu1_all.deb ...
Unpacking python3-pyinotify (0.9.6-2ubuntu1) ...
Selecting previously unselected package whois.
```

Installed Fail2Ban on the EC2 Ubuntu instance to monitor login attempts and automatically block IPs with repeated failed logins.

```
ser sessions running outdated binaries:
 ubuntu @ session #7: sshd[1179,1293]
 ubuntu @ user manager service: systemd[1184]

o VM guests are running outdated hypervisor (qemu) binaries on this host.
buntu@ip-172-31-3-154:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
buntu@ip-172-31-3-154:~$ sudo systemctl enable fail2ban
udo systemctl start fail2ban
ynchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
xecuting: /usr/lib/systemd/systemd-sysv-install enable fail2ban
buntu@ip-172-31-3-154:~$
```

Enabled and started Fail2Ban service to protect against SSH brute-force attacks in real time.

```
ubuntu@ip-172-31-3-154:~$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
ubuntu@ip-172-31-3-154:~$ sudo systemctl enable fail2ban
sudo systemctl start fail2ban
Synchronizing state of fail2ban.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable fail2ban
ubuntu@ip-172-31-3-154:~$
```

Fail2Ban was configured to start on boot and run actively, protecting the EC2 instance against SSH-based brute-force attacks.

This project involved securing an Ubuntu-based EC2 instance through a series of best practices. The instance was configured with key-based SSH access, system updates were applied, and unattended upgrades enabled. A firewall was activated via UFW to allow only SSH, while root login and password authentication were disabled. Finally, Fail2Ban was installed to monitor and block brute-force attacks automatically.