Day 16

"CLOUD SECURITY"

Disable SSH Public Key Authentication and Enable Password on AWS Linux:

Why Should You Do This?

1. Restrict Unauthorized Access from Lost Keys:
    a. If a public/private key pair is compromised, attackers can log in silently without knowing any passwords.
    b. Disabling public key auth forces the use of passwords, which can be rotated or enforced with strong policies.
2. Centralized Authentication Control:
    a. Passwords can be tied to centralized systems (e.g., LDAP or PAM).
    b. Easier to manage password policies and auditing compared to tracking scattered SSH keys.
3. Mitigate Insider Threats:
    a. Disabling public key auth prevents ex-employees from accessing systems using leftover keys they once uploaded.
4. Tighter Control over Access Patterns:
    a. With password authentication, admins can enforce MFA, session timeout, and other PAM-based controls.

What If Not Done?

1. Silent Unauthorized Logins:
    a. If an attacker gets hold of a private key (e.g., via laptop theft or GitHub leaks), they can access systems without detection.
2. No Password Rotation:
    a. SSH keys don't expire or require rotation unless manually managed, making it harder to enforce periodic access changes.
3. Lack of Visibility:
    a. It's more difficult to audit or log who used which key, especially if keys are reused or poorly named.
4. Hard to Revoke Access Quickly:
    a. Removing access means identifying and deleting keys from multiple instances — slower than disabling a user password.

Why AWS Allows SSH Public Key Authentication by Default:

1. Public Key Auth is Actually More Secure (When Managed Properly): Public key authentication is more secure than password-based login in many scenarios. It resists brute-force attacks and credential stuffing. You can't just "guess" a private key like you can a password.
2. User-Specific Access Control: Each user (admin/dev/ops) can have their own key pair. No need to share a common password — a best practice in secure environments.
3. No Passwords Stored on the System: Since passwords aren't used or stored, there's no risk of theft via local file compromise, shoulder surfing, or password reuse.

4. Automation-Friendly: Key-based SSH access allows tools (scripts, CI/CD pipelines, Ansible, etc.) to automate logins securely — which wouldn't be possible with password prompts.
5. Easier Initial Setup for Admins: When spinning up an EC2 instance, uploading a public key allows zero-touch, passwordless login — saving setup time and reducing friction.

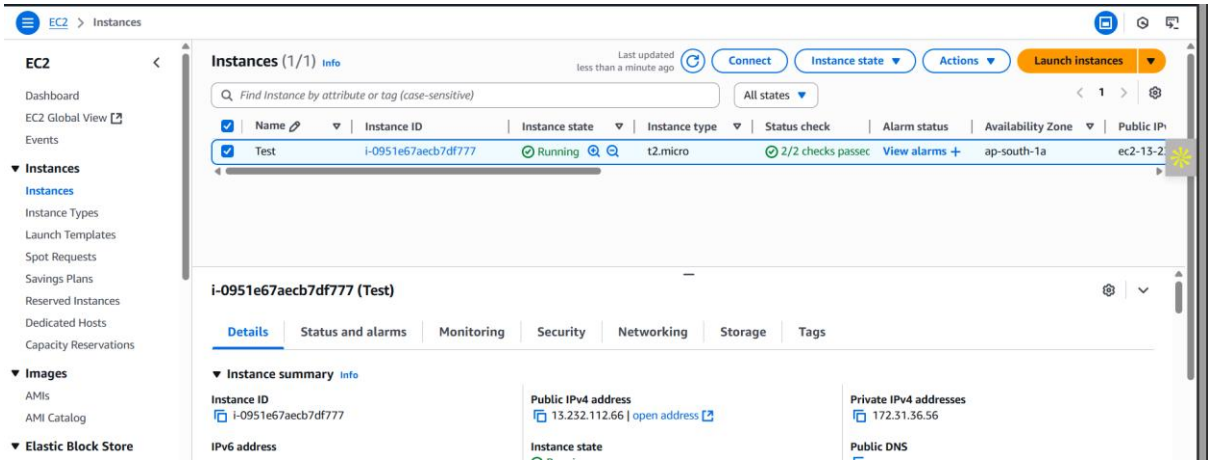
So, Why Might You Still Disable It?

- You're in a high-security, compliance-focused environment (e.g., banking, government, or military) that:
  - Requires central identity authentication.
  - Needs full auditing of all access events.
  - Does not permit unmanaged or user-uploaded keys.

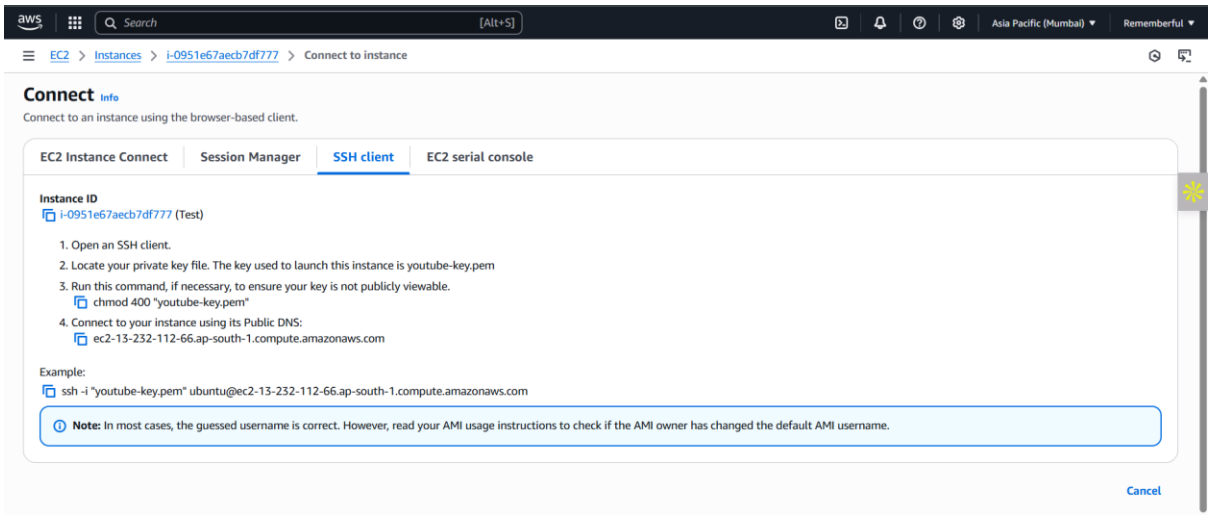
<u>Disable SSH Public Key Authentication and Enable Password on AWS Linux:</u>

Steps:

Create one instance: then click on the 'connect' button.



Following screen will appear: go to the "SSH client" section.



Open the terminal in the host OS (here it is windows): and copy paste the "example" given there in the above screenshot. Following screen will appear:

```
C:\Users\Aditya\Downloads>ssh -i "D:\Cybersecurity\youtube-key.pem" ubuntu@ec2-13-232-112-66.ap-south-1.compute.amazonaws.com
Welcome 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 Tue Aug  5 09:50:57 UTC 2025

  System load:  0.0               Processes:             104
  Usage of /:   25.4% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 172.31.36.56
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-36-56:~$
```

Clearly, we are in the ubuntu: now turn yourself to the admin as shown below and open the file using any text editor like vim or nano, the file path is /etc/ssh/sshd_config

```
ubuntu@ip-172-31-36-56:~$ sudo su
root@ip-172-31-36-56:/home/ubuntu# vi /etc/ssh/sshd_config
```

```
root@ip-172-31-36-56: /home

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

"/etc/ssh/sshd_config" 131L, 3517B                                    1,0-1         Top
```

Look for the section as shown below, turn them “no”:

```
#PermitRootLogin prohibit-password
#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
```

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

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no
```

Now, save and exit, and then restart the service as shown below:

```
root@ip-172-31-36-56:/home/ubuntu# sudo service ssh restart
root@ip-172-31-36-56:/home/ubuntu#
```

Then put the password for the user name root: and then you are done.

```
root@ip-172-31-36-56:/home/ubuntu# sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
root@ip-172-31-36-56:/home/ubuntu#
```

--The End--