

EC2 SSH Key Recovery Using Recovery Instance Method

ASWIN VTK

Abstract

Loss of SSH access to an Amazon EC2 instance is a common operational issue caused by missing private keys, incorrect permissions, or accidental modification of SSH configuration files. This article explains a reliable and industry-accepted recovery technique known as the **Recovery Instance Method**. The method restores SSH access by temporarily attaching the root volume of the affected EC2 instance to a healthy instance, correcting the SSH authorized keys, and reattaching the volume. This document provides a detailed, step-by-step procedure suitable for academic, technical, or internal project submission.

1. Problem Statement

An EC2 instance (referred to as **OriginalEC2**) becomes inaccessible due to one or more of the following reasons:

- Loss of SSH private key file
- Accidental deletion or corruption of `~/.ssh/authorized_keys`
- Incorrect file or directory permissions
- SSH service misconfiguration

The objective is to restore SSH access without terminating the instance or losing data.

2. Prerequisites

Before starting the recovery process, ensure the following:

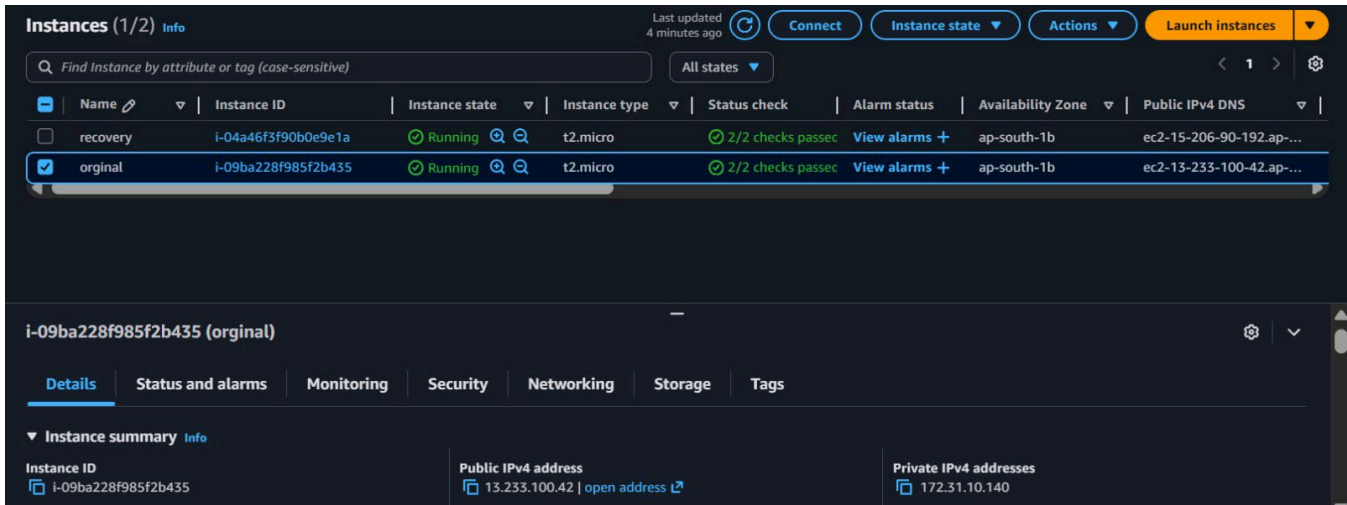
- AWS Console access with permissions to manage EC2 and EBS
 - One healthy EC2 instance (**RecoveryEC2**) that is accessible via SSH
 - Same operating system on both instances
 - Both instances are in the same AWS Availability Zone
-

3. Architecture Overview

The recovery process involves the following components:

- **OriginalEC2**: The instance with lost SSH access
- **Root EBS Volume**: The primary storage volume of OriginalEC2
- **RecoveryEC2**: A temporary EC2 instance with working SSH access

The root volume of OriginalEC2 is detached and attached to RecoveryEC2 for repair.



4. Step-by-Step Recovery Procedure

Step 1: Stop the Broken EC2 Instance

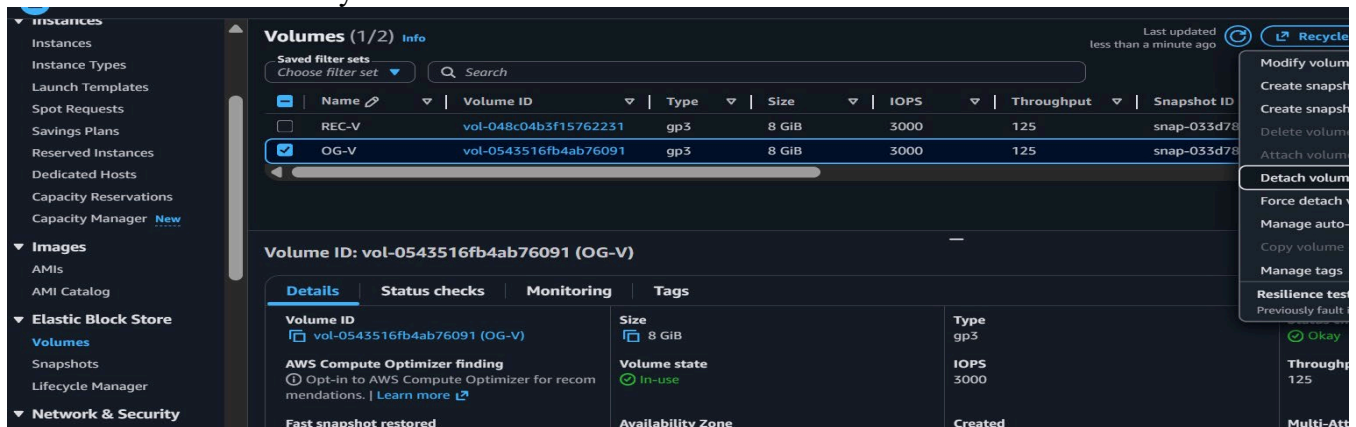
Stopping the instance ensures data integrity before detaching the root volume.

1. Open the AWS EC2 Dashboard
2. Select **OriginalEC2**
3. Choose **Instance state** → **Stop instance**
4. Wait until the instance state changes to **Stopped**

Step 2: Detach the Root EBS Volume

1. Select the stopped instance
2. Open the **Storage** tab
3. Click the **Volume ID** of the root volume
4. Choose **Actions** → **Detach volume**

The root disk is now safely detached.



Step 3: Attach the Volume to Recovery Instance

1. Select the detached volume
 2. Click **Actions** → **Attach volume**
 3. Choose **RecoveryEC2** as the target instance
 4. Specify a device name such as `/dev/sdf` or `/dev/xvdf`
-

Step 4: Mount the Attached Volume

SSH into **RecoveryEC2** and execute the following commands:

```
sudo su  
lsblk
```

Identify the newly attached disk (e.g., `xvdf`, `nvme1n1`, or `nvme1n1p1`).

Create a mount directory and mount the volume:

```
mkdir /Original  
mount -t xfs -o nouuid /dev/ xvdf /Original
```

Note: Adjust the device name based on the output of `lsblk`.

Step 5: Restore SSH Authorized Keys (Critical Step)

Copy the working SSH key from the recovery instance to the broken instance's(named original) root volume:

Path of key in recovery EC2 : `/home/ec2-user/.ssh/authorized_keys`

Path of key in original EC2 : `/Original/home/ec2-user/.ssh/authorized_keys`

```
cat /home/ec2-user/.ssh/authorized_keys >> /Original/home/ec2-user/.ssh/authorized_keys
```

Verify the file content:

```
cat /Original/home/ec2-user/.ssh/authorized_keys
```

This step ensures that the RecoveryEC2 SSH key is now authorized for OriginalEC2.

```
original bin boot dev etc home lib lib64 local media mnt opt original proc root run sbin srv sys user var
[root@ip-172-31-0-214 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda 202:0 0 8G 0 disk
├─xvda1 202:1 0 8G 0 part /
├─xvda127 259:0 0 1M 0 part
├─xvda128 259:1 0 10M 0 part /boot/efi
├─xvdf 202:80 0 8G 0 disk
├─xvdf1 202:81 0 8G 0 part /original
│
├─xvdf127 259:2 0 1M 0 part
├─xvdf128 259:3 0 10M 0 part
└─
[root@ip-172-31-0-214 ~]# cat /home/ec2-user/.ssh/authorized_keys >> /original/home/ec2-user/.ssh/authorized_keys
[root@ip-172-31-0-214 ~]# cat /Original/home/ec2-user/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDLBTmvpz1n1gtb5mlNE+4CP64qUG8m8qqus/P51soqs9IH4kykeMeXlWlX4d4iI/M3X0vGfPcS80d98cr62azf2z9Ld5rdsNP4Q0r4dZM43uHhWnOvTPGChjg2ePHHq45p
GIM8f6v9kmlr134CvaygCPQa2fec10ceqgWxjJGvUlgR0hubxJWtEubXkmaUwImI4dXpIlgPeA53P+/ehbM149bQcdhdc5TWw6V/J1W6IOAyEgK65Rf1lccomEXCtR3t6k0HcUfhag6amU1X3+BsB+1rq5Pp1q2d5j30rqd8
1FHlg2tPX0tH0209 lostkey
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCB40YDldoWJfAZ7jsiy0CEGPeEm0FSC5lsc4E3HUm4A4HXAFndI/THQ67Goigj0LWfIehf9K5A1vyB2aoIn9xXlL8pnFr4H253bI0fIq6aln9Y1b191buanYeENFAYPm10+hy40kVowZ
pD02Rqpc4fR9SXW1fEqXhBiJmFq4h0AKh0RG3Rpu294ulpbyuZ2nGTGfNZFEumbW3ScRbmnyKiuL4JmF9nJYZqYg/jwXc2YvtI8ZUVZ421Sqfr3MU/t3/UMtJB67UF3Edu5Pjpw6Zs6o6J2ueW3Nu9J3tjCqwtfpqSRK621McL1I1W
4MoPFRUmt1at+7 recovery
[root@ip-172-31-0-214 ~]#
```

Step 6: Unmount and Reattach the Volume

1. Unmount the volume:

Amount /Original

2. Detach the volume from RecoveryEC2 using the AWS Console
3. Reattach it to OriginalEC2
4. Use the original root device name (e.g., `/dev/xvda` or `/dev/sda1`)

Step 7: Start and Access the Recovered Instance

1. Start **OriginalEC2**
2. SSH using the **RecoveryEC2** key:

```
ssh -i recovery-key.pem ec2-user@<OriginalEC2-Public-IP>
```

The screenshot shows a terminal window titled "ec2-user@ip-172-31-1". The user has executed the command `C:\Users\MIS LAP MILMA\Downloads>ssh -i "recovery.pem" ec2-user@ec2-13-233-100-42.ap-south-1.compute.amazonaws.com`. A red arrow points to the filename `"recovery.pem"`. The terminal output displays the standard SSH warning about the host's fingerprint, which the user accepts by typing "yes". It also shows the URL `https://aws.amazon.com/linux/amazon-linux-2023` and the system logo for Amazon Linux 2023.

```
Microsoft Windows [Version 10.0.26200.7623]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MIS LAP MILMA\Downloads>ssh -i "recovery.pem" ec2-user@ec2-13-233-100-42.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-233-100-42.ap-south-1.compute.amazonaws.com (13.233.100.42)' can't be established.
ED25519 key fingerprint is SHA256:oT3LmClp8m6oAprZiiYVE/WTw8HgZeAg1YYHnePJt/Hw.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-233-100-42.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

#_
~\_ ##### Amazon Linux 2023
~~ \#####
~~ \#####
~~ \###|
~~ \#/ https://aws.amazon.com/linux/amazon-linux-2023
~~ V~! ~->
~~~
~~~
~~~
~~~
Last login: Sun Feb 1 18:25:26 2026 from 13.233.177.3
[ec2-user@ip-172-31-10-140 ~]$
```

Successful login confirms recovery.

6. Result and Validation

SSH access to the original EC2 instance is successfully restored without data loss. The instance functions normally with its original configuration intact.

8. Conclusion

The Recovery Instance Method is a safe, effective, and AWS-recommended approach for restoring SSH access to EC2 instances. This technique avoids rebuilding infrastructure and ensures business continuity. Understanding this procedure is essential for cloud administrators and DevOps engineers managing production environments.
