

# 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.

**Instances (1/2) Info**

Last updated 4 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
recovery	i-04a46f3f90b0e9e1a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1b	ec2-15-206-90-192.ap...
<b>orginal</b>	i-09ba228f985f2b435	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	ap-south-1b	ec2-13-233-100-42.ap...

**i-09ba228f985f2b435 (orginal)**

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

**Instance summary** [Info](#)

Instance ID <a href="#">i-09ba228f985f2b435</a>	Public IPv4 address <a href="#">13.233.100.42   open address</a>	Private IPv4 addresses <a href="#">172.31.10.140</a>
--	---	---

## 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.

**Volumes (1/2) Info**

Last updated less than a minute ago [Recycle](#)

Choose filter set [Search](#)

Name	Volume ID	Type	Size	IOPS	Throughput	Snapshot ID
REC-V	vol-048c04b3f15762231	gp3	8 GiB	3000	125	snap-033d78
<b>OG-V</b>	<b>vol-0543516fb4ab76091</b>	<b>gp3</b>	<b>8 GiB</b>	<b>3000</b>	<b>125</b>	<b>snap-033d78</b>

**Volume ID: vol-0543516fb4ab76091 (OG-V)**

[Details](#) [Status checks](#) [Monitoring](#) [Tags](#)

Volume ID <a href="#">vol-0543516fb4ab76091 (OG-V)</a>	Size <a href="#">8 GiB</a>	Type gp3
AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.   <a href="#">Learn more</a>	Volume state In-use	IOPS 3000
Fast snapshot restored	Availability Zone	Created

**Actions**

- Modify volume
- Create snapshot
- Force detach
- Manage auto-
- Copy volume
- Manage tags
- Resilience test

Previously faulted: **OK**

Throughput: 125

Multi-Att:

---

### 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 [ ] usr 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 /
└─xvda12 202:17   0   1M  0 part
└─xvda128 202:80   0   8G  0 disk
└─xvdf   202:81   0   8G  0 part /Original
└─xvdf1  202:81   0   8G  0 part /original .
└─xvdf127 202:81   0   1M  0 part
└─xvdf128 202:81   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 AAAAB3NzaC1yc2EAAAQAAQDLeIMpvzIgqIgtb5mLNE+4CP64qouBG8mgus/P+1eqe9IN4kykMeXLWIXd4oi/M93XOVECPSR80dU9tcr62awzfzg91d5rDsNPL40QtDzBm49zuHVhnOVBTPGohjgzbePBhgol8q8nGLMtPf67YgmkTl+3Y+CsycytQ9mLa2fec10ceqgwxEjJgbJzq0hubxjWtEuBXKgmIwblm/D4kpHlgPeA53P+ehDH149BQfdDhc51TWV6V/JIW61OayBcG16SF1ICcmZXCIT3El6k0HcuFHagD6QmUlX3+BJB+1Rq5PpIg2dsJS0FqD8r1fHq1g2tpX0tH0209 lostkey
ssh-rsa AAA2B3NzaC1yc2EAAAQAAQBAAAAQC4b0YD1dc0VfAZjsiy0cEGpeEm0FScs1sC4EHuM4HXAFnd/1IHQG7Goj1OTWFieh9KSATvy82a0ln9xXLl5pnP+4H25jb0fiqF6alnJ9lb19lwauYorNFAYFBmEl0+h40kWovZegpD027qpcf3AR9WX1ExQhKbiJMfQ4hDkhr0RgJ8RpupU94ulpbvJ2nTGNFZBumhW3ScrBonnYKiu1lGTaqln/JyFqGy/jwXc2YvtI8zUVZ4isQfr3N3/t3/uMtJB67UFJBdu5PjpWt6z5so6J2ueW3NU9glzjcQwtfpqSRK621McL11jw#4+4emMoPrRtUm+lat+7 recovery
[root@ip-172-31-0-214 /]#

```

## Step 6: Unmount and Reattach the Volume

- Unmount the volume:

**umount /Original**

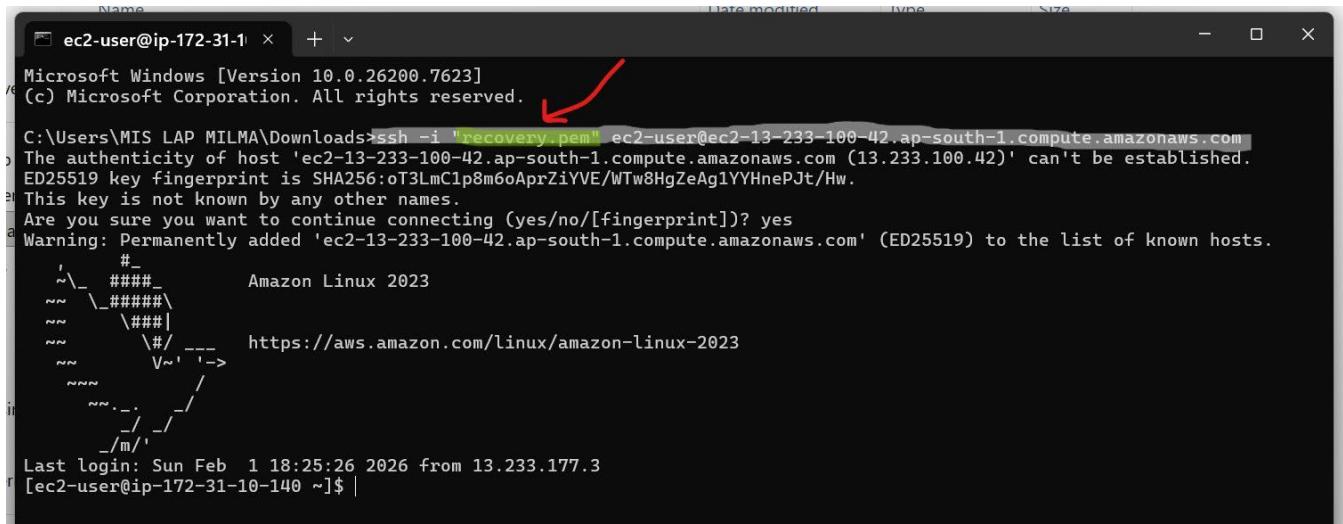
- Detach the volume from RecoveryEC2 using the AWS Console
- Reattach it to OriginalEC2
- Use the original root device name (e.g., /dev/xvda or /dev/sda1)

## Step 7: Start and Access the Recovered Instance

- Start **OriginalEC2**

- SSH using the RecoveryEC2 key:

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



```

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

C:\Users\MISS 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:oT3LmC1p8m6oAprZiYVE/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
     \~`-'>
      ~~. /_
       /_/
      /_/
     /_m/` 

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.

---