



Practical - 1: AWS EC2 Key Pair Lost Recovery

Objective: To regain SSH access to an existing Amazon EC2 instance ("Old web server") after the associated key pair file (`key-pair-1.pem`) was lost/deleted, by injecting a new public key.

Note: The steps shown in the original document involve deleting the key locally, launching a new instance, using the new instance's key to access the old instance's volume, and finally starting the old instance with the new key.

Phase 1: Replicate the 'Lost' Key Scenario (Initial Setup & Deletion)

This phase establishes the scenario by showing the initial successful connection and the subsequent deletion of the key file.

1. Launch Initial Instance (Old web server):

- Launch an EC2 instance named "**Old web server**".
- Use **Amazon Linux 2023 AMI** and **t3.micro** instance type.
- Ensure the instance is running and has a key pair associated (e.g., `key-pair-1.pem`).

2. Initial Connection and Validation:

Click on the Instance id and paste it on GIT Bash

```
$ ssh -i "key-pair-1.pem" ec2-user@<OLD_INSTANCE_PUBLIC_IP>
```

- Verify access by creating a file (e.g., `pranav`) and use `ls`.

3. Simulate Key Loss:

- Log out of the instance - logout.

Delete the key pair file locally from your machine (simulating a permanent loss).

Bash

```
$ rm KEYPAIRNAME.pem
```

- The instance's root volume (`vol-090...`) remains intact.
-

Phase 2: Launch Temporary Instance and Stop Old Instance

To gain access to the old instance's root volume, we must attach it to a new, accessible temporary instance.

1. **Launch a Temporary Instance (New web server):**
 - Launch a second EC2 instance named "**New web server**".
 - Use the same AMI and instance type (e.g., Amazon Linux 2023, t3.micro).
 - **CRITICAL:** Associate a **new key pair** (e.g., `new-key-pair.pem`) with this instance.
 - (Optional but done in the practical) Modify the storage size during launch to **9 GiB**.
 - (Optional but done in the practical) Configure Security Group to **Allow HTTP traffic from the internet**.
 -
 2. **Stop the Old Instance:**
 - Navigate to the EC2 Instances dashboard.
 - Select the "**Old web server**" instance.
 - Change its Instance state to **Stopped**. This is necessary to detach its root volume.
-

Phase 3: Detach and Attach Volume

We will detach the old instance's root volume and attach it as a secondary data volume to the new, running instance.

1. **Detach Volume from Old Instance:**
 - In the EC2 menu, go to **Elastic Block Store -> Volumes**.
 - Identify the root volume associated with the "**Old web server**" (e.g., `vol-090...`, Size: 8 GiB/9 GiB).
 - Select the volume, go to **Actions -> Detach volume**.
 - Wait for the volume status to become **available**.
2. **Attach Volume to New Instance:**
 - Select the detached volume again.
 - Go to **Actions -> Attach volume**.
 - For the **Instance**, select the "**New web server**" instance (e.g., `i-0a1...`).

- Set the **Device name** to a standard data device name like `/dev/xvdf` (or `/dev/sdf`).
 - Click **Attach volume**.
-

Phase 4: Inject New Key into Old Volume

Now, we connect to the new instance and use it to update the key on the old instance's file system (which is now mounted as a data volume).

1. Connect to New Instance and Identify Volume:

Connect to the "**New web server**" using the **new key pair** (`new-key-pair.pem`).

Bash

```
$ ssh -i "new-key-pair.pem" ec2-user@<NEW_INSTANCE_PUBLIC_IP>
```

- Run `lsblk` to list block devices. You should see the new instance's root volume (`nvme0n1`, 9G) and the attached old volume (`nvme1n1`, 8G).

2. Mount the Old Volume:

Create a mount point and mount the partition containing the old instance's file system (typically the first partition, e.g., `/dev/nvme1n1p1`).

Bash

```
sudo mkdir /mnt/oldroot  
sudo mount -o rw,nouuid /dev/nvme1n1p1 /mnt/oldroot
```

- Verify the content of the mounted volume (the root of the old system).

```
ls /mnt/oldroot
```

3. Inject the New Public Key:

Copy the public key from the **new instance** into the **old instance's authorized_keys** file on the mounted volume.

Bash

```
$ sudo cat /home/ec2-user/.ssh/authorized_keys >>  
/mnt/oldroot/home/ec2-user/.ssh/authorized_keys
```

Note: The command shown in the original document is truncated.

4. Unmount the Volume:

Unmount the volume safely.

Bash

```
$ sudo umount /mnt/oldroot
```

Phase 5: Re-attach and Start Old Instance

The volume now has the new public key and can be reattached to its original instance.

1. Detach Volume from New Instance:

- Go back to the AWS console **Volumes** page.
- Select the old volume (e.g., **vol-090...**).
- Go to **Actions -> Detach volume**.

2. Attach Volume to Old Instance:

- Select the volume again.
- Go to **Actions -> Attach volume**.
- For **Instance**, select the "Old web server" instance (e.g., **i-043...**).
- For **Device name**, enter the original root volume device name, typically **/dev/xvda** (or **/dev/sda1** depending on the AMI).
- Click **Attach volume**.

3. Start the Old Instance:

- Go to the **Instances** dashboard.
 - Select the "Old web server" instance.
 - Go to **Instance state -> Start instance**.
-

Phase 6: Validation (Connecting to Old Instance)

1. Connect with New Key Pair:

Once the "Old web server" instance is **Running**, attempt to connect using the **new key pair** (**new-key-pair.pem**) and the old instance's Public IP.

Bash

```
$ ssh -i "new-key-pair.pem" ec2-user@<OLD_INSTANCE_PUBLIC_IP>
```

- **Validation:** You should successfully connect and see the file (**pranav**) created in Phase 1.