**Task: Recover EC2 Access if Key Pair is Lost**

**Name:** Abhishek Hengade

**Batch:** B-56

This guide provides a detailed method to regain access to an EC2 instance if you have lost the .pem private key file. Since the original key cannot be recovered, this process involves replacing the old, lost public key on the instance with a new one.

**1. Understanding the Recovery Strategy**

The core strategy is to detach the root EBS volume from the inaccessible instance and attach it as a secondary volume to a temporary "helper" instance. From the helper instance, we will mount the volume and overwrite the lost public key in the authorized\_keys file with a new public key. Finally, we will re-attach the volume to the original instance, allowing us to connect using a new key pair.

**Prerequisites:** You must be able to launch a new EC2 instance in the **same Availability Zone** as the inaccessible instance.

**2. Step-by-Step Recovery Process**

**Part 1: Preparation**

**Step 1: Create a New Key Pair and Stop the Instance**

* In the EC2 console, navigate to **Key Pairs** and create a new key pair. Download and securely store the new .pem file.
* Navigate to **Instances**, select your inaccessible instance, and **stop** it. This is crucial for detaching the root volume safely.

**Step 2: Launch a Temporary "Helper" Instance** Launch a new EC2 instance in the **same Availability Zone** as your stopped instance. When launching, select the **new key pair** you just created. This helper instance will be terminated after the recovery process.

**Step 3: Detach the Root Volume**

* Select your stopped, inaccessible instance. In the **Storage** tab, find the **Volume ID** of the root device (e.g., /dev/sda1).
* Navigate to the **Volumes** section, select that volume by its ID, click **Actions**, and choose **Detach volume**. Wait for its status to become **"available."**

**Part 2: Modifying the Volume**

**Step 4: Attach the Volume to the Helper Instance**

* With the volume still selected, click **Actions** and **Attach volume**.
* Select your running **helper** instance from the dropdown.
* For the device name, enter something like /dev/sdf to attach it as a secondary, non-root disk.

**Step 5: Connect to the Helper Instance and Mount the Volume** SSH into your helper instance using the **new** .pem file. Then, run the following commands to mount the attached volume and access its file system.

Bash

# List block devices to identify the volume (e.g., xvdf1 is the partition)

lsblk

# Create a temporary mount point

sudo mkdir /mnt/recovery

# Mount the volume's partition to the new directory

sudo mount /dev/xvdf1 /mnt/recovery

**Step 6: Replace the Authorized Key** Now, copy the public key from the helper instance's authorized\_keys file to the one on the mounted volume. This replaces the old, lost key with your new one.

Bash

# The path may be /home/ubuntu/.ssh/ depending on your AMI

sudo cp /home/ec2-user/.ssh/authorized\_keys /mnt/recovery/home/ec2-user/.ssh/authorized\_keys

**Step 7: Unmount the Volume** After copying the key, unmount the volume from the helper instance.

Bash

sudo umount /mnt/recovery

**Part 3: Finalizing the Recovery**

**Step 8: Re-attach the Volume to the Original Instance**

* In the AWS console, detach the volume from the helper instance.
* Once its status is **"available,"** attach it back to your original (still stopped) instance.
* **Crucially**, you must set the device name back to its original value (e.g., /dev/sda1) to designate it as the root volume.

**Step 9: Start and Connect to the Original Instance** Start your original instance. Once it's running, you can now SSH into it using the **new key pair**. The connection will be successful.

**Step 10: Clean Up** Don't forget to **terminate** the temporary helper instance to avoid incurring unnecessary charges.

