

Source code

```
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;

import com.amazonaws.regions.Regions;

import com.amazonaws.services.ec2.AmazonEC2;

import com.amazonaws.services.ec2.AmazonEC2ClientBuilder;

import com.amazonaws.services.ec2.model.*;

import java.util.Collections;

public class InitializeVolumeFromSnapshot {

    public static void main(String[] args) {

        // Set the AWS region you want to use

        Regions region = Regions.US_EAST_1;

        // Create an Amazon EC2 client

        AmazonEC2 ec2Client = AmazonEC2ClientBuilder.standard()

            .withRegion(region)

            .withCredentials(DefaultAWSCredentialsProviderChain.getInstance())

            .build();

        // Set the instance ID of the EC2 instance where you want to attach the restored volume

        String instanceId = "your-instance-id";

        // Set the ID of the snapshot you want to restore

        String snapshotId = "your-snapshot-id";

        // Set the device name to which the restored volume will be attached

        String deviceName = "/dev/sdf";
```

```

try {
    // Create a new Amazon EBS volume from the snapshot
    CreateVolumeRequest createVolumeRequest = new CreateVolumeRequest()
        .withSnapshotId(snapshotId)
        .withAvailabilityZone(region.getName());
    Volume volume = ec2Client.createVolume(createVolumeRequest);

    // Wait until the volume becomes available
    WaitForVolumeAvailable(ec2Client, volume.getVolumeId());

    // Attach the volume to the EC2 instance
    AttachVolumeRequest attachVolumeRequest = new AttachVolumeRequest()
        .withInstanceId(instanceId)
        .withVolumeId(volume.getVolumeId())
        .withDevice(deviceName);
    ec2Client.attachVolume(attachVolumeRequest);

    System.out.println("Volume " + volume.getVolumeId() + " restored and attached successfully to
instance " + instanceId + ".");
} catch (Exception e) {
    System.err.println("Error: " + e.getMessage());
}
}

```

```

private static void WaitForVolumeAvailable(AmazonEC2 ec2Client, String volumeId) throws
InterruptedException {

```

```

    DescribeVolumesRequest describeVolumesRequest = new
DescribeVolumesRequest().withVolumeIds(volumeId);

```

```

    while (true) {

```

```
DescribeVolumesResult describeVolumesResult =  
ec2Client.describeVolumes(describeVolumesRequest);  
  
Volume volume = describeVolumesResult.getVolumes().get(0);  
  
if ("available".equalsIgnoreCase(volume.getState())) {  
    return;  
}  
  
Thread.sleep(5000); // Sleep for 5 seconds before checking again  
}  
  
}  
  
}
```