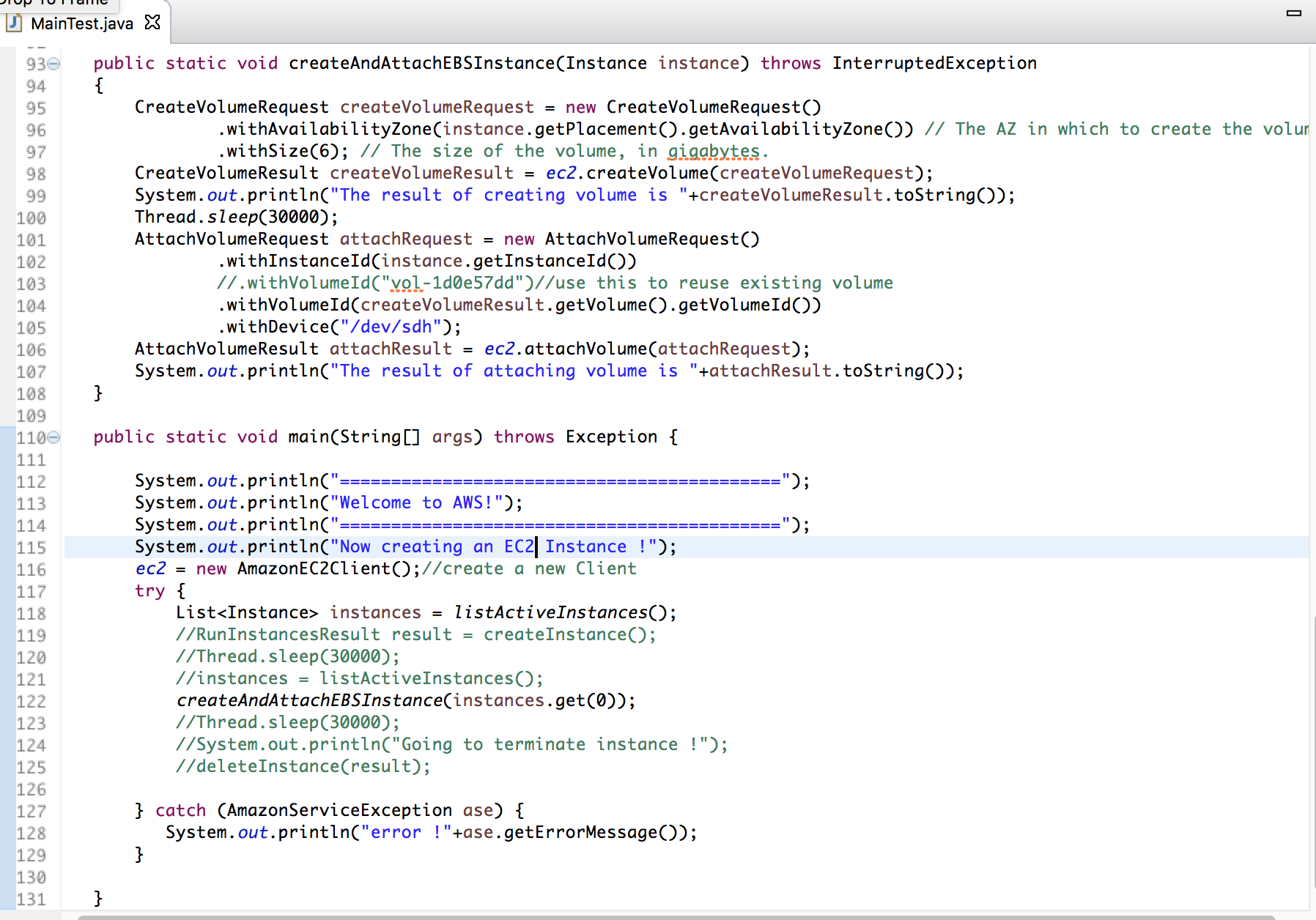
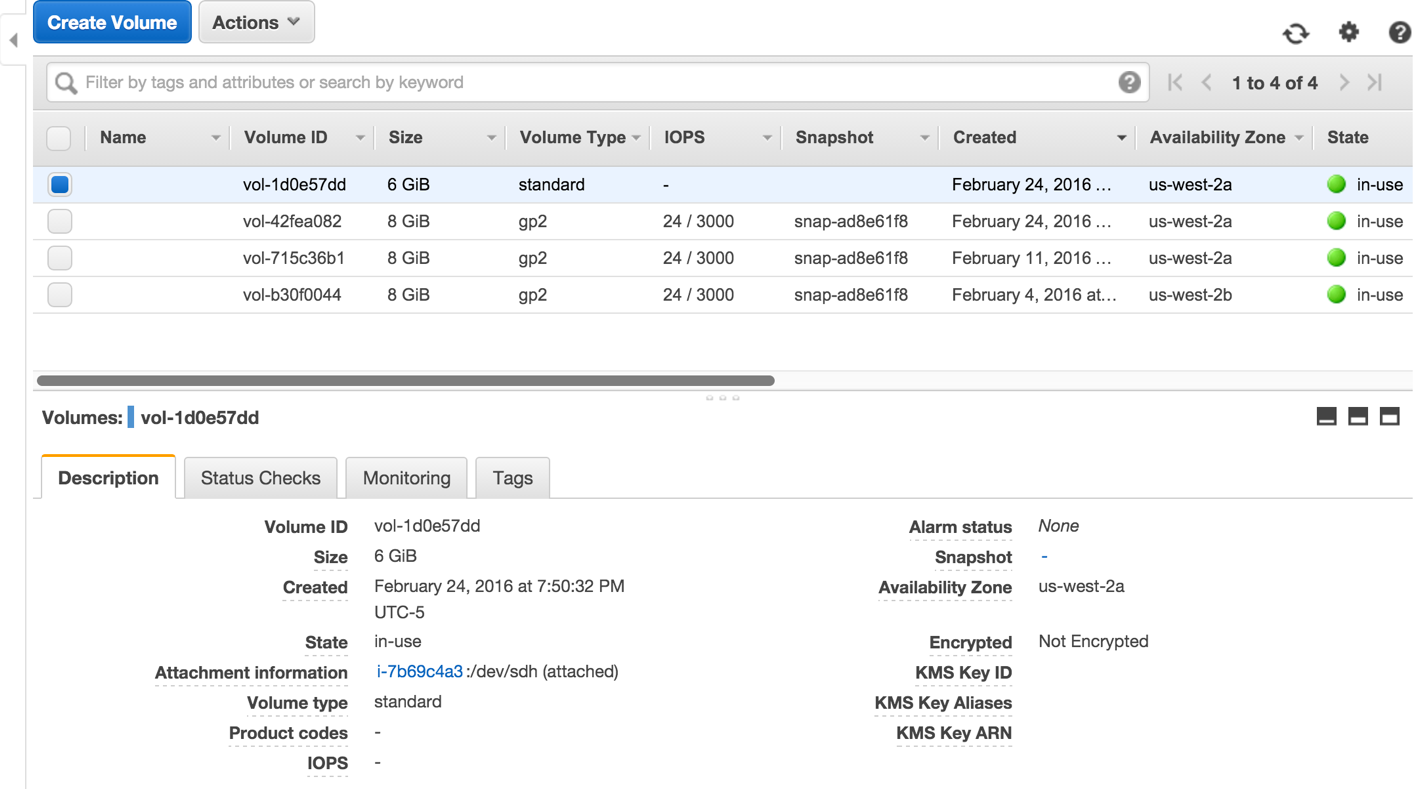
**Mini Home Work -3**

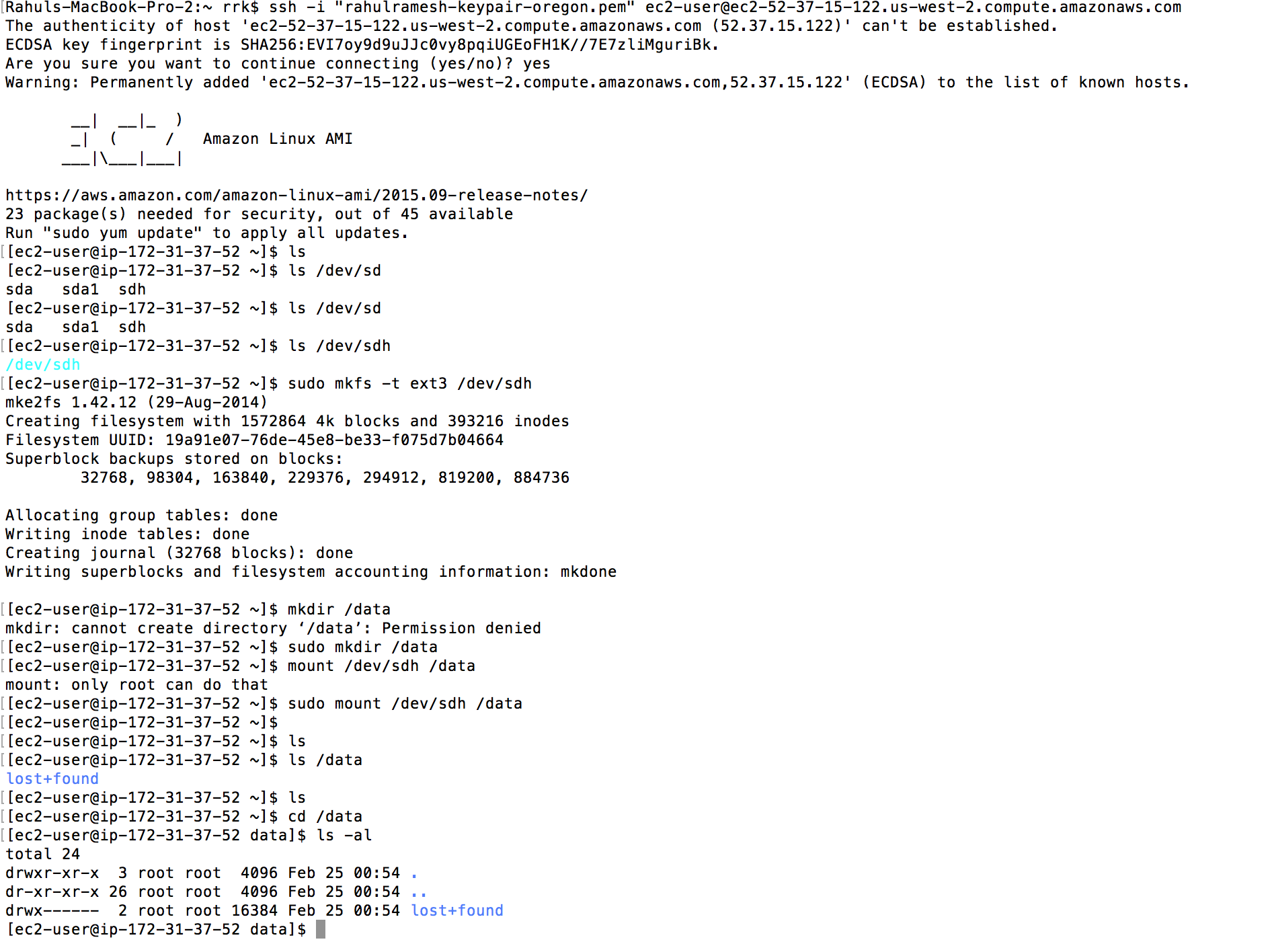
* **EBS Volume created and Attached programmatically using Java SDK for Amazon Web Service and Eclipse Mars IDE**



**Fig 1** - Eclipse Console Screen Shot

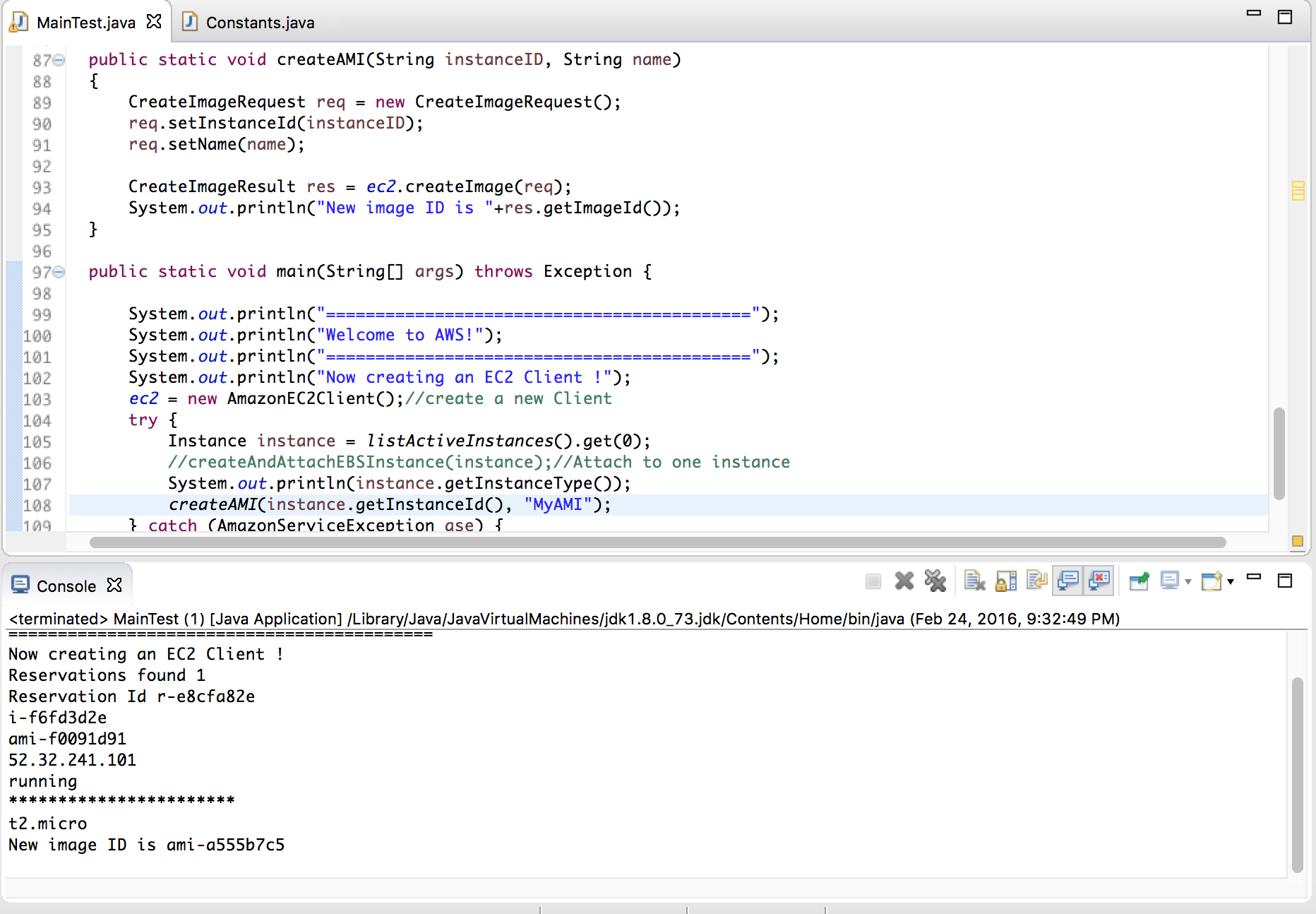


**Fig 2** – EBS Volume Running in AWS Console

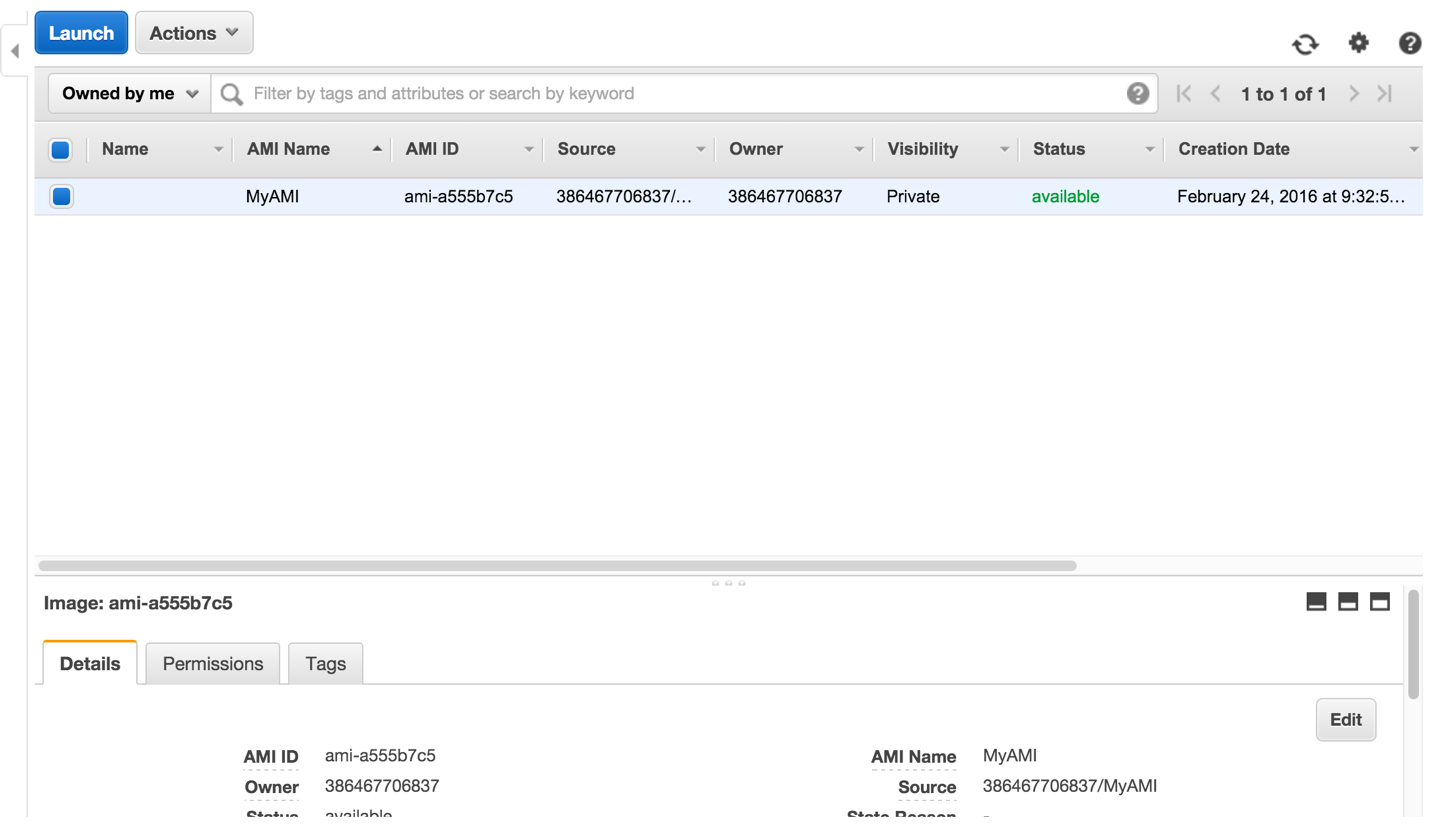


**Fig 3** – SSH to Program created EBS Volume /dev/sdh

* **Create AMI**

****

**Fig 4- Programmatically created AMI named “MyAMI”**

****

**Fig 5- Confirmation of AMI Image created in Console**

**Source Code for both EBS and AMI Creation**

/\***MainTest.java**\*/

package com.aws.assignment2;

import java.util.ArrayList;

/\*

\* Copyright 2010-2016 Amazon.com, Inc. or its affiliates. All Rights Reserved.

\*

\* Licensed under the Apache License, Version 2.0 (the "License").

\* You may not use this file except in compliance with the License.

\* A copy of the License is located at

\*

\* http://aws.amazon.com/apache2.0

\*

\* or in the "license" file accompanying this file. This file is distributed

\* on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either

\* express or implied. See the License for the specific language governing

\* permissions and limitations under the License.

\*/

import java.util.List;

import com.amazonaws.AmazonClientException;

import com.amazonaws.AmazonServiceException;

import com.amazonaws.regions.Region;

import com.amazonaws.regions.Regions;

import com.amazonaws.services.ec2.AmazonEC2;

import com.amazonaws.services.ec2.AmazonEC2Client;

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

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

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

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

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

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

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

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

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

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

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

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

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

import com.aws.assignment2.util.Constants;

public class MainTest {

public static AmazonEC2 ec2 = null;

public static List<Instance> listActiveInstances() throws AmazonClientException {

ec2.setRegion(Region.getRegion(Regions.US\_WEST\_2));

List<Instance> instances = new ArrayList<Instance>();

List<String> serviceInstanceIds = new ArrayList<String>();

serviceInstanceIds.add(Constants.INSTANCE1);

serviceInstanceIds.add(Constants.INSTANCE2);

serviceInstanceIds.add(Constants.INSTANCE3);

DescribeInstancesRequest request = new DescribeInstancesRequest();

request.setInstanceIds(serviceInstanceIds);

DescribeInstancesResult response = ec2.describeInstances(request);

List<Reservation> reservations = response.getReservations();

System.out.println("Reservations found "+reservations.size());

for (Reservation reservation : reservations) {

System.out.println("Reservation Id "+reservation.getReservationId());

for (Instance instance : reservation.getInstances()) {

instances.add(instance);

System.out.println(instance.getInstanceId());

System.out.println(instance.getImageId());

System.out.println(instance.getPublicIpAddress());

System.out.println(instance.getState().getName());

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

return instances;

}

public static void createAndAttachEBSInstance(Instance instance) throws InterruptedException

{

CreateVolumeRequest createVolumeRequest = new CreateVolumeRequest()

.withAvailabilityZone(instance.getPlacement().getAvailabilityZone()) // The AZ in which to create the volume.

.withSize(6); // The size of the volume, in gigabytes.

CreateVolumeResult createVolumeResult = ec2.createVolume(createVolumeRequest);

System.out.println("The result of creating volume is "+createVolumeResult.toString());

Thread.sleep(30000);

AttachVolumeRequest attachRequest = new AttachVolumeRequest()

.withInstanceId(instance.getInstanceId())

//.withVolumeId("vol-1d0e57dd")//use this to reuse existing volume

.withVolumeId(createVolumeResult.getVolume().getVolumeId())

.withDevice("/dev/sdh");

AttachVolumeResult attachResult = ec2.attachVolume(attachRequest);

System.out.println("The result of attaching volume is "+attachResult.toString());

}

public static void createAMI(String instanceID, String name)

{

CreateImageRequest req = new CreateImageRequest();

req.setInstanceId(instanceID);

req.setName(name);

CreateImageResult res = ec2.createImage(req);

System.out.println("New image ID is "+res.getImageId());

}

public static void main(String[] args) throws Exception {

System.out.println("===========================================");

System.out.println("Welcome to AWS!");

System.out.println("===========================================");

System.out.println("Now creating an EC2 Client !");

ec2 = new AmazonEC2Client();//create a new Client

try {

Instance instance = listActiveInstances().get(0);

createAndAttachEBSInstance(instance);//Attach to one instance

System.out.println(instance.getInstanceType());

createAMI(instance.getInstanceId(), "MyAMI");

} catch (AmazonServiceException ase) {

System.out.println("error !"+ase.getErrorMessage());

}

}

}

/\***Constants.java**\*/

package com.aws.assignment2.util;

public final class Constants {

public static final String INSTANCE1 = "i-f6fd3d2e";

public static final String INSTANCE2 = "i-fdcf673a";

public static final String INSTANCE3 = "i-7b69c4a3";

}