**BAPATLA ENGINEERING COLLEGE**

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**HOME ASSIGNMENT**

**SUBJECT: CLOUD COMPUTING**

**FACULTY: K.SAI PRASANTH SIR**

**FROM:**

**NAME: K.SIRISHA**

**CLASS:3/4 B.TECH**

**SECTION:IT-A**

**ROLL NO:16**

**Y18AIT438**

Java application to launchEC2 instance

**Creating Security Group:**

Source Code : securitygroup.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| package com.amazonaws.samples.sirisha; import com.amazonaws.services.ec2.AmazonEC2; import com.amazonaws.services.ec2.AmazonEC2ClientBuilder; import com.amazonaws.services.ec2.model.CreateSecurityGroupRequest; import com.amazonaws.services.ec2.model.CreateSecurityGroupResult; import com.amazonaws.services.ec2.model.AuthorizeSecurityGroupIngressRequest; import com.amazonaws.services.ec2.model.AuthorizeSecurityGroupIngressResult; import com.amazonaws.services.ec2.model.IpPermission; import com.amazonaws.services.ec2.model.IpRange;  /\*\*  \* Creates an EC2 security group.  \*/ public class SECURITYGROUP {     public static void main(String[] args)     {                  String group\_name = "securitygroup";         String group\_desc = "description";        // String vpc\_id = args[2];          final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();          CreateSecurityGroupRequest create\_request = new             CreateSecurityGroupRequest()                 .withGroupName(group\_name)                 .withDescription(group\_desc);                // .withVpcId(vpc\_id);          CreateSecurityGroupResult create\_response =             ec2.createSecurityGroup(create\_request);          System.out.printf(             "Successfully created security group named %s",             group\_name);          IpRange ip\_range = new IpRange()             .withCidrIp("[0.0.0.0/0](http://0.0.0.0/0)");          IpPermission ip\_perm = new IpPermission()             .withIpProtocol("tcp")             .withToPort(80)             .withFromPort(80)             .withIpv4Ranges(ip\_range);          IpPermission ip\_perm2 = new IpPermission()             .withIpProtocol("tcp")             .withToPort(22)             .withFromPort(22)             .withIpv4Ranges(ip\_range);          AuthorizeSecurityGroupIngressRequest auth\_request = new             AuthorizeSecurityGroupIngressRequest()                 .withGroupName(group\_name)                 .withIpPermissions(ip\_perm, ip\_perm2);          AuthorizeSecurityGroupIngressResult auth\_response =             ec2.authorizeSecurityGroupIngress(auth\_request);          System.out.printf(             "Successfully added ingress policy to security group %s",             group\_name);     } }     |  |  | | --- | --- | |  |  | |  |  |

Graphical user interface, text

Description automatically generated

**Creating Key Pair:**

Source Code : keypair.java

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| package com.amazonaws.samples.sirisha; import com.amazonaws.services.ec2.AmazonEC2; import com.amazonaws.services.ec2.AmazonEC2ClientBuilder; import com.amazonaws.services.ec2.model.CreateKeyPairRequest; import com.amazonaws.services.ec2.model.CreateKeyPairResult;  /\*\*  \* Creates an EC2 key pair  \*/ public class KEYPAIR {     public static void main(String[] args)     {                  String key\_name ="key";          final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();          CreateKeyPairRequest request = new CreateKeyPairRequest()             .withKeyName(key\_name);          CreateKeyPairResult response = ec2.createKeyPair(request);          System.out.printf(             "Successfully created key pair named %s",             key\_name);     } }       |  |  |  | | --- | --- | --- | |  |  | | |  | | |  |  | | |  |  |

**Running Ec2 Instance:**

Source code : ec2instance.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| package com.amazonaws.samples.sirisha; import com.amazonaws.services.ec2.AmazonEC2; import com.amazonaws.services.ec2.AmazonEC2ClientBuilder; import com.amazonaws.services.ec2.model.InstanceType; import com.amazonaws.services.ec2.model.RunInstancesRequest; import com.amazonaws.services.ec2.model.RunInstancesResult; import com.amazonaws.services.ec2.model.Tag; import com.amazonaws.services.ec2.model.CreateTagsRequest; import com.amazonaws.services.ec2.model.CreateTagsResult;  /\*\*  \* Creates an EC2 instance  \*/ public class INSTANCE {     public static void main(String[] args)     {                  String name = "abc\_instance";         String ami\_id = "ami-03d64741867e7bb94";          final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();          RunInstancesRequest run\_request = new RunInstancesRequest()             .withImageId(ami\_id)             .withInstanceType(InstanceType.T2Micro)             .withMaxCount(1)             .withMinCount(1);          RunInstancesResult run\_response = ec2.runInstances(run\_request);          String reservation\_id = run\_response.getReservation().getInstances().get(0).getInstanceId();          Tag tag = new Tag()             .withKey("Name")             .withValue(name);          CreateTagsRequest tag\_request = new CreateTagsRequest()             .withResources(reservation\_id)             .withTags(tag);          CreateTagsResult tag\_response = ec2.createTags(tag\_request);          System.out.printf(             "Successfully started EC2 instance %s based on AMI %s",             reservation\_id, ami\_id);     } }   |  |  | | --- | --- | |  |  | |  |  |