README.md

To create an EC2 instance programatically

The AWS SDK(Software Development Kit) provides an API for Amazon Web Services. Using the SDK, you can easily build applications that work with AWS services.

AWS SDK for Java

1. Create an AWS account

- Create a user and get an access key
- Store it in `~/.aws/credentials`

```
[default]
aws_access_key_id = [ID]
aws_secret_access_key = [KEY]
```

- Set up `~/.aws/config`

```
[default]
region = us-east-1
```

2. Set up a suitable Java Development Environment and set the environment path

3. Install Maven

- Apache Maven is the most popular build and dependency resolution tool for Java like Npm and Pip
- To test the Maven installation `mvn -v`

4. Build a Java project with Maven

```
$ mvn archetype:generate
        -DgroupId=[package-name]
        -DartifactId=[project-name]
        -DarchetypeArtifactId=maven-archetype-quickstart
        -DinteractiveMode=false
```

5. Define a Maven build in *pom.xml*


```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
...
...
...
<properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
</properties>
<dependencyManagement>
  <dependencies>
    <dependency>
      <groupId>com.amazonaws</groupId>
      <artifactId>aws-java-sdk-bom</artifactId>
      <version>1.11.327</version>
      <type>pom</type>
      <scope>import</scope>
    </dependency>
  </dependencies>
</dependencyManagement>
```

```
...
...
  <build>
          <plugins>
              <plugin>
                  <groupId>org.apache.maven.plugins</groupId>
                  <artifactId>maven-shade-plugin</artifactId>
                  <version>3.2.4</version>
                  <executions>
                      <execution>
                          <phase>package</phase>
                          <goals>
                              <goal>shade</goal>
                          </goals>
                          <configuration>
                              <transformers>
                                  <transformer
                                      implementation="org.apache.maven.plugins.shade.resource.ManifestResourceTransformer">
                                      <mainClass>[package_name].[class_name]</mainClass>
                                  </transformer>
                              </transformers>
                          </configuration>
                      </execution>
                  </executions>
              </plugin>
          </plugins>
  </build>
</project>
```


6. Declare dependencies in *pom.xml* - ex) EC2 module

- Example Code

7. Write code

- Create a security group

    - Optinially set up ingress rules
- Create a key pair
- Create an instance with the security group and the key pair attached to it

8. Example code

```
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;
import com.amazonaws.services.ec2.model.CreateKeyPairRequest;
import com.amazonaws.services.ec2.model.CreateKeyPairResult;
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;
import java.util.List;

public static void main( String[] args ) {
        String sgName = "securityGroupForDemo";
        String sgDesc = "This is a security group for demo";
        String keyName = "COMS-6998-demo-key";
        String instanceName = "COMS-6998-demo-instance";
        String amiId = "ami-06b263d6ceff0b3dd"; // Ubuntu 18.04 LTS
        int minInstance = 1;
        int maxInstance = 1;

        createSecurityGroup(sgName, sgDesc);
```

```
        createKeyPair(keyName);
        createInstance(instanceName, amiId, sgName, keyName, minInstance, maxInstance);
    }

    public static void createSecurityGroup(String groupName, String desc) {
        final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();
        CreateSecurityGroupRequest createRequest = new CreateSecurityGroupRequest()
                                                        .withGroupName(groupName)
                                                        .withDescription(desc);

        CreateSecurityGroupResult createResponse = ec2.createSecurityGroup(createRequest);

     }

     public static void createKeyPair(String keyName) {
        final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();
        CreateKeyPairRequest request = new CreateKeyPairRequest().withKeyName(keyName);
        CreateKeyPairResult response = ec2.createKeyPair(request);
     }

     public static void createInstance(String name, String amiId, String sgName, String keyName, int min, int max) {
        final AmazonEC2 ec2 = AmazonEC2ClientBuilder.defaultClient();
        RunInstancesRequest runRequest = new RunInstancesRequest()
                                        .withImageId(amiId)
                                        .withInstanceType(InstanceType.T1Micro)
                                        .withMaxCount(min)
                                        .withMinCount(max)
                                        .withKeyName(keyName)
                                        .withSecurityGroups(sgName);

        RunInstancesResult runResponse = ec2.runInstances(runRequest);

        String reservationId = runResponse.getReservation().getInstances().get(0).getInstanceId();

        Tag tag = new Tag()
            .withKey("Name")
            .withValue(name);
```

```
        CreateTagsRequest tagRequest = new CreateTagsRequest()
            .withResources(reservationId)
            .withTags(tag);

        CreateTagsResult tagResponse = ec2.createTags(tagRequest);

        System.out.printf("EC2 instance %s started based on AMI %s\n", reservationId, amiId);
}
```

AWS SDK for Python

1. Create an AWS account

- Create a user and get an access key
- Store it in `~/.aws/credentials`

```
[default]
aws_access_key_id = [ID]
aws_secret_access_key = [KEY]
```

- Set up `~/.aws/config`

```
[default]
region = us-east-1
```

2. Install boto3

```
$ pip install boto3
```

3. Write code

- Create a security group
  - Optinially set up ingress rules
- Create a key pair
- Create an instance with the security group and the key pair attached to it

4. Example code

```
import boto3

def createSG(sgName,sgDesc):
    ec2 = boto3.client('ec2')
    res = ec2.create_security_group (
        GroupName = sgName,
        Description = sgDesc
    )

def createKeyPair(name):
    ec2 = boto3.resource('ec2')
    res = ec2.create_key_pair(KeyName = name)

def createEC2(amiId, keyName, sgName, instType = 't1.micro', minInst = 1, maxInst = 1):
    ec2 = boto3.resource('ec2')
    instances = ec2.create_instances (
        ImageId = amiId,
        MinCount = minInst,
        MaxCount = maxInst,
        InstanceType = instType,
        KeyName = keyName,
        SecurityGroups=[sgName]
    )

if __name__ == '__main__':
    sgName = 'securityGroupForDemo'
```

```
sgDesc = 'This is a security group for demo'
keyName = 'COMS-6998-demo-key'
amiId = 'ami-06b263d6ceff0b3dd'

createSG(sgName, sgDesc)
createKeyPair(keyName)
createEC2(amiId, keyName, sgName)
print('Done')
```