AWS Command Line Interface
Part-1

To install AWS Command Line Interface on your machine go to the following link.
https://aws.amazon.com/cli/

On the right side of the page, you will find the download options for Windows, Mac and Linux operating system.

Download and install the CLI as per your OS.

1. Before you start using command line interface, go to the AWS console. Login as root user. Go to IAM service console. Create a user and give that user EC2FullAccess and IAMFullAccess permissions. Copy Access ID and Secret Access Key for the user in a file.

AWS CLI Commands

2. **aws configure**

Use above command to connect (Authenticate) to the AWS cloud. The command will ask for Access ID and Access Key. Paste these from the file where you copied for the user in the above step. Also specify a region like ap-south-1 when prompted. In other words you logged in to AWS cloud using above user identity.
After this all the operations that you perform using command line, will be by the user that you created in step 1.

3. **aws describe-instances**

This command describes information about all EC2 instances for your organization in the specified region. You can find instance ID of an EC2 instance and use following command to display information about that EC2 instance only.

Aws describe-instance --instance-id <Paste-your-instance-id>

Create a Group, User and IAM Policy using AWS Command line.

4. **aws iam list-groups**

This command displays a list of groups created in your IAM service.

5. **aws iam create-group --group-name <new-group-name>**

The above command will create a new group by the name specified in <new-group-name>.
You can verify that the group is created using the command in step 4.

6. **aws iam list-users**

This command displays a list of users created in your IAM service.

7. **aws iam create-user --user-name <new-user-name>**

The above command will create a new user by the name specified in <new-user-name>.

You can verify that the user is created using the command in step 6.

8. **aws iam add-user-to-group --user-name <username> --group-name <group-name>**

The above command will add a user specified in <username> to group name specified in <group-name>.

9. **aws iam get-group --group-name <group-name>**

You can verify that the user is added to the group using this command.

10. Create a json file that contains the policy that provides a user full EC2 permissions for ap-south-1 region (if you want a different region then replace the region name in the following json code.)

Use notepad or vi or any editor and paste following json code in the file (keep indenting as it is) and save the file with any name and json as extension.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestedRegion": "ap-south-1"
                }
            }
        }
    ]
}
```

11. **aws iam put-group-policy --group-name <group-name> --policy-name <policy-name> --policy-document file://<filename.json>**

This command will create a policy that will provide ec2 full access but only for ap-south-1 region and attach it to the group specified by <group-name>. Thus users who are members of this group will get this permission.

12. **aws iam list-group-policies --group-name <group-name>**

Create an EC2 Instance using AWS command line.

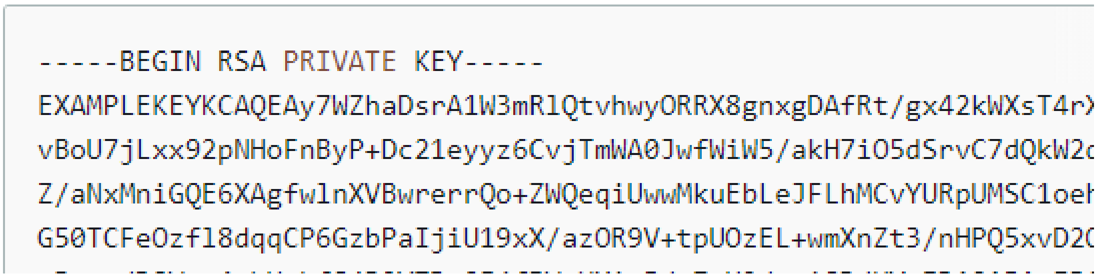
1. **aws ec2 create-key-pair --key-name <key-name> --query 'KeyMaterial' --output text > <key-name>.pem**

If the above command doesn’t work on Windows then open Power Shell and type following command.

1. **aws ec2 create-key-pair --key-name <key-name> --query 'KeyMaterial' --output text | out-file -encoding ascii -filepath <key-name>.pem**

Remember the directory where you created thie key.

If you edit the resulting key, it should look like as shown below.



```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEKEYKCAQEAy7WZhaDsrA1W3mR1QtvhwyORRX8gnxgDAfRt/gx42kWXsT4r
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oe
G50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2
```

2. **aws ec2 describe-key-pairs**

This command will display all the key pairs created in your EC2 service.

3. **aws ec2 create-security-group --group-name <security-group-name> --description “Security group for Cli practical”**

This command will create a new security group. This will be attached to the EC2 instance that we will create later using AWS CLI.

4. **aws ec2 describe-security-groups**

This command will display all the security groups present in your EC2 service. Check if the new security group created above is shown in the list.

aws ec2 describe-security-groups --group-name i<security-group-name> --query SecurityGroups[*].GroupId

This command will return the security group id created above. Copy it as it is required in the later step.

5. **aws ec2 authorize-security-group-ingress --group-name <above-security-group-name> --protocol tcp --port 22 --cidr 0.0.0.0/0**

This command will add a rule to allow ssh connection from any IP address. This will allow us to connect to out EC2 instance that we will create next.

6.

```
aws ec2 run-instances  --image-id <image-id> --count 1   --instance-type t2.micro  --key-name <key-name>  --security-group-id <security-group-id>
```

This command will create an EC2 instance. The image id can be copied from the AWS console from the launch instance options AMI page.
In the information displayed copy the Instance Id of this new EC2 instance.

7. Find public IP address of the EC2 instance.

```
aws ec2 describe-instances  --instance-id <Your-EC2-Instance-id> --query "Reservations[*].Instances[*].PublicIpAddress" --output text
```

8. Following command will display more information about all the EC2 instance.

```
aws ec2 describe-instances --query "Reservations[*].Instances[*].[PublicIpAddress, KeyName, PrivateIpAddress, InstanceId, State.Name]" --output text
```

This displays
Public IP Address , Key , Private IP Address, Instance ID and Current State of the Instance.

Now use putty and the public key and the public IP address of this EC2 instance and try to connect.

Clean Up

1. Delete EC2 instance

```
aws ec2 terminate-instances   --instance-id  <Your-EC2-Instance-id>
```

2. Delete key pair

```
aws ec2 delete-key-pair  --key-name <your-key-name>
```

3. Delete Security Group

```
aws ec2 delete-security-group --group-name <security-group-name>
```

4. Remove user from a group

```
aws iam remove-user-from-group  --user-name <user-name> --group-name <group-name>
```

5. Delete policy attached to the group in step 11.

```
aws iam delete-group-policy --policy-name <policy-name>  --group-name <group-name>
```

6. Delete group

```
aws iam delete-group --group-name <group-name>
```

7. Delete User

```
aws iam delete-user --user-name <user-name>
```