AWS Command Line
Part 2

This document assumes that you have already performed part 1 of AWS command line. Also you have connected to the AWS cloud using **aws configure** command with a user having full access permissions on ec2.

1. Create a VPC

```
aws  ec2  create-vpc  --cidr-block 172.16.0.0/16  --query  Vpc.VpcId  --output  text
```

This will create a new VPC and display the VPC id. Please copy the VPC id as it will be required in later commands.

2. Create subnets in the VPC. Here we will create 2 subnets. One subnet will be public (connected to Internet) and second subnet will be private.

```
aws  ec2   create-subnet  --vpc-id <paste-above-vpc-id>    --cidr-block 172.16.1.0/24
```

similarly create the second subnet

```
aws  ec2  create-subnet  --vpc-id <paste-above-vpc-id>  --cidr-block 172.16.10.0/24
```

c**opy subnet id displayed in the output. This subnet will be the public subnet.**

3. Create an Internet Gateway.

```
aws  ec2  create-internet-gateway   --query InternetGateway.InternetGatewayId  --output text
```

copy the Internet gateway id displayed.

4. Attach Internet Gateway to your VPC

```
aws  ec2  attach-internet-gateway   --vpc-id <paste-above-vpc-id>   --internet-gateway-id
<paste-internet-gateway-id>
```

5. Create a route table

```
aws  ec2  create-route-table   --vpc-id  <paste-above-vpc-id>   --query
RouteTable.RouteTableId           --output  text
```

copy the route table id displayed.

6. Create a route to allow traffic to Internet through the Internet Gateway.

```
aws   ec2  create-route   --route-table-id <paste-route-table-id>  --destination-cidr-block
0.0.0.0/0       --gateway-id  <paste-your-internet-gateway-id>
```

7. Check the route table and the route entry using

```
aws  ec2   describe-route-tables    --route-table-id  <paste-your-route-table-id>
```

8. Associate route table to the subnet that will be the public subnet.

aws ec2 associate-route-table --subnet-id <subnet-id> --route-table-id <paste-route-table-id>

9. Enable auto public address assignment for the EC2 instances attached to this public subnet.

aws ec2 modify-subnet-attribute --subnet-id <paste-your-subnet-id> --map-public-ip-on-launch

10. Create a key-pair.

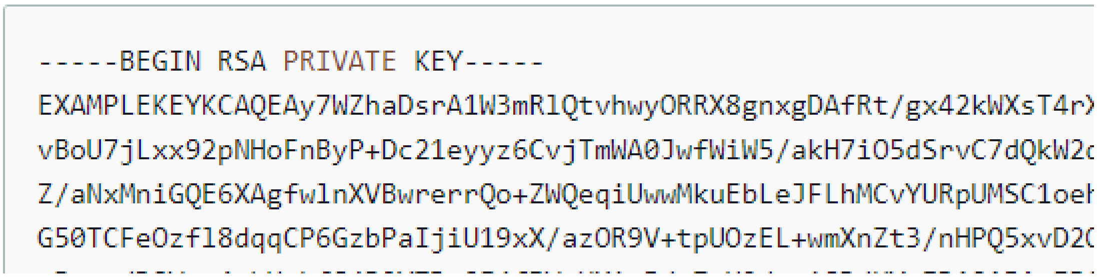
1. **aws ec2 create-key-pair --key-name <key-name> --query 'KeyMaterial' --output text > <key-name>.pem**

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-----
EXAMPLEKEYKCAQEAy7WZhaDsrA1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4r
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oe
G50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2
```

11. **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.

12. **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.

13.

```
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.

14.

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

15. Find public IP address of the EC2 instance.

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

16. 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.

<u>Clean Up</u>

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. Delete Subnet, VPC and Internet-Gateway

```
aws  ec2  delete-subnet   --subnet-id <paste-subnet-id>
```

```
aws  ec2  delete-route-table  --route-table-id <paste-route-table-id>
aws ec2  detach-internet-gateway   --internet-gateway-id  <paste-internet-gateway-id>

aws  ec2 delete-internet-gateway  --internet-gateway-id  <paste-your-internet-gateway-id>

aws  ec2 delete-vpc   --vpc-id <paste-your-vpc-id>
```