

I have emphasized tagging everything you create because it will save a lot of frustration later when trying to figure out which resource is which.

Here are the steps.

If you don't have the CLI configured download and configure it :

<http://docs.aws.amazon.com/cli/latest/userguide/installing.html>

Create VPC:

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
```

Tag That VPC:

```
aws ec2 create-tags --resources vpc-d363afab --tags Key=Name,Value=CLI-VPC
```

Create a public Subnet:

```
aws ec2 create-subnet --vpc-id vpc-d363afab --cidr-block 10.0.1.0/24
```

Tag:

```
aws ec2 create-tags --resources subnet-7314ad17 --tags Key=Name,Value=CLI-Public-Subnet
```

Create a private Subnet:

```
aws ec2 create-subnet --vpc-id vpc-d363afab --cidr-block 10.0.2.0/24
```

Tag:

```
aws ec2 create-tags --resources subnet-4109b025 --tags Key=Name,Value=CLI-Private-Subnet
```

Create an Internet Gateway:

```
aws ec2 create-internet-gateway
```

Tag:

```
aws ec2 create-tags --resources igw-afdd01d6 --tags Key=Name,Value=CLI-Internet-Gateway
```

Attach Internet Gateway:

```
aws ec2 attach-internet-gateway --internet-gateway-id igw-5d685a38 --vpc-id vpc-d363afab
```

Allocate Elastic IP:

```
aws ec2 allocate-address --domain vpc
```

Create a Nat-gateway and place it in the public Subnet:

```
aws ec2 create-nat-gateway --subnet-id subnet-1a2b3c4d --allocation-id eipalloc-37fc1a52
```

Tag:

```
aws ec2 create-tags --resources nat-0e4d97e539eadf232 --tags Key=Name,Value=CLI-Nat-Gateway
```

Create Route Table 1 for public Subnet:

```
aws ec2 create-route-table --vpc-id vpc-d363afab
```

Tag:

```
aws ec2 create-tags --resources rtb-14c3736e --tags Key=Name,Value=CLI-PUBLIC_RT
```

Create Route Table 2 for private Subnet:

```
aws ec2 create-route-table --vpc-id vpc-d363afab
```

Tag:

```
aws ec2 create-tags --resources rtb-cbc070b1 --tags Key=Name,Value=CLI-PRIVATE_RT
```

Create a route to the internet in Route Table 1:

```
aws ec2 create-route --route-table-id rtb-14c3736e --destination-cidr-block 0.0.0.0/0 --gateway-id igw-afdd01d6
```

Create a route to the internet in Route Table 2 via Nat:

```
aws ec2 create-route --route-table-id rtb-cbc070b1 --destination-cidr-block 0.0.0.0/0 --gateway-id nat-0e4d97e539eadf232
```

Associate Route Table 1 to PublicSubnet :

```
aws ec2 associate-route-table --route-table-id rtb-14c3736e --subnet-id subnet-7314ad17
```

Associate Route Table 2 to PrivateSubnet:

```
aws ec2 associate-route-table --route-table-id rtb-1245623e --subnet-id subnet-234567as
```

Create a Security Group for Web Access and SSH:

```
aws ec2 create-security-group --group-name CLI-WEB-SecurityGroup --description "My security group" --vpc-id vpc-d363afab
```

Tag:

```
aws ec2 create-tags --resources sg-03ca1371 --tags Key=Name,Value=CLI_SECURITY_GROUP
```

Add Ingress Port 22 and 80:

```
aws ec2 authorize-security-group-ingress --group-id sg-c3ed34b1 --protocol tcp --port 22 --cidr 0.0.0.0/0
```

```
aws ec2 authorize-security-group-ingress --group-id sg-c3ed34b1 --protocol tcp --port 80 --cidr 0.0.0.0/0
```

Create Key Pair and copy the key part and write it to a file MyKeyPairCLI.pem :

```
aws ec2 create-key-pair --key-name MyKeyPairCLI
```

Change the permissions on that file to restrict access:

```
chmod 400 MyKeyPairCLI.pem
```

Launch EC2 Instance In public subnet with Amazon AMI ami-8c1be5f6 :

```
aws ec2 run-instances --image-id ami-8c1be5f6 --count 1 --instance-type t2.micro --key-name MyKeyPairCLI --security-group-ids sg-c3ed34b1 --subnet-id subnet-7314ad17 --associate-public-ip-address
```

TAG EC2:

```
aws ec2 create-tags --resources i-05c8b15394d0905b8 --tags Key=Name,Value=CLI_EC2
```

Describe Instance to get the IP or check the console:

```
aws ec2 describe-instances
```

SSH into your web browser:

```
ssh ec2-user@34.34.234.4 -i MyKeyPairCLI.pem
```

And you're now logged in! Have fun with your setup.

