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:

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

--------------------------------------------------------------------------------------------------------------------------------

CloudWatch

----------------------------------------------------------------------------------------------------

aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2018-06-01T23:18:00Z --end-time 2018-06-19T23:18:00Z --period 3600 --namespace AWS/EC2 --statistics Average --dimensions Name=InstanceId,Value=i-07442b7dca24a5740

aws cloudwatch get-metric-statistics --metric-name CPUUtilization --start-time 2018-06-01T23:18:00Z --end-time 2018-06-19T23:18:00Z --period 3600 --namespace AWS/EC2 --statistics Maximum --dimensions Name=InstanceId,Value=i-07442b7dca24a5740

----------------------------------------------------------------------------------------------------

EC2

----------------------------------------------------------------------------------------------------

List Instances:

aws ec2 describe-instances  
aws ec2 describe-instances --output text  
aws ec2 describe-instances --output json  
aws ec2 describe-instances --output table

List only few things like a dashboard:

aws ec2 describe-instances Ðquery 'Reservations[].Instances[].[Placement.AvailabilityZone, State.Name, InstanceId, InstanceType,VpcId,SubnetId,ImageId,Tags[?Key==`Name`].Value|[0]]'

Create a key pair and output that to a file:

aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text > MyKeyPair.pem

\*After that make sure to chmod 400 MyKeyPair file.

Create an instance:

aws ec2 run-instances --image-id ami-8c1be5f6 --instance-type t2.micro --key-name MyKeyPair

This will create an instance in the default VPC. Specify the subnet name and the right security groups within that VPC if you want to be specific.

Create an instance in an specific subnet:

aws ec2 run-instances --image-id ami-8c1be5f6 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-beb3eacc --subnet-id subnet-ed36c3c2

If needed change the ingress of a security group:

aws ec2 authorize-security-group-ingress --group-id sg-814134f2 --protocol tcp --port 22 --cidr 0.0.0.0/0

Terminate 1 or multiple ec2s at once :

aws ec2 terminate-instances --instance-ids i-0b20d7680fa0e6ba0 i-00251da28fa34ffd1

aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my root volume snapshot."

aws ec2 create-volume --size 10 --region us-east-1 --availability-zone us-east-1a --volume-type gp2

aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-id i-01474ef662b89480 --device /dev/sdf

To make an EBS volume available for use on Linux

1. Connect to your instance using SSH.

2. Use the lsblk command to view your available disk devices and their mount points (if applicable) to help you determine the correct device name to use.

3. New volumes are raw block devices, and you must create a file system on them before you can mount and use them. Use the sudo file -s dev/xvdf command to list special information, such as file system type.

4. Use the following command to create an ext4 file system on the volume. Substitute the device name: sudo mkfs -t ext4 /dev/xvdf

5. Make a mount point : [ec2-user ~]$ sudo mkdir mount\_point

6. Use the following command to mount the volume at the location you just created: [ec2-user ~]$ sudo mount /dev/xvdf mount\_point

7. Write files to your new mount point.

-------------------------------------------------------------------------------------------------------

S3

---------------------------------------------------------