Install AWS CLI

```
sudo snap install aws-cli –classic

aws configure

AWS Access Key ID [None]: ********************
AWS Secret Access Key [None]: ****************************
Default region name [None]: ******
Default output format [None]:
```

VPC

Create a VPC

```
-> aws ec2 create-vpc --cidr-block 10.0.0.0/20
```

Giving name to created VPC

```
-> aws ec2 create-tags --resources vpc-0ac8ec3d7ac633b3f --tags Key=Name,Value=NewVPC
```

Subnets

Public Subnet 1

Create public subnet 1 in NewVPC

```
-> aws ec2 create-subnet --vpc-id vpc-0ac8ec3d7ac633b3f --cidr-block 10.0.1.0/24 --
availability-zone ap-south-1a
```

Assign the name to public subnet 1 as 'PublicSubnet1'

```
-> aws ec2 create-tags --resources subnet-0a5083f882cf209c0 --tags
Key=Name,Value=PublicSubnet1
```

Public Subnet 2

Create public subnet 2 in New VPC

```
-> aws ec2 create-subnet --vpc-id vpc-0ac8ec3d7ac633b3f --cidr-block 10.0.2.0/24 --
availability-zone ap-south-1b
```

Assign the name to public subnet 2 as 'PublicSubnet2'

```
-> aws ec2 create-tags --resources subnet-0e1603ed47a79c6a4 --tags
Key=Name,Value=PublicSubnet2
```

Private Subnet 1

Create private subnet 1 in New VPC

```
-> aws ec2 create-subnet --vpc-id vpc-0ac8ec3d7ac633b3f --cidr-block 10.0.3.0/24 --
availability-zone ap-south-1b
```

Assign the name to private subnet 1 as PrivateSubnet1

```
-> aws ec2 create-tags --resources subnet-0ee3aa017f3c459ad --tags
Key=Name,Value=PrivateSubnet1
```

Private Subnet 2

Create private subnet 2 in New VPC

```
-> aws ec2 create-subnet --vpc-id vpc-0ac8ec3d7ac633b3f --cidr-block 10.0.4.0/24 --availability-zone ap-south-1b
```

Assign the name to private subnet 2 as PrivateSubnet2

```
-> aws ec2 create-tags --resources subnet-0b66c001c91288ecc --tags Key=Name,Value=PrivateSubnet2
```

Route Tables

Public Route Table

Create Public Route Table in NewVPC

```
-> aws ec2 create-route-table --vpc-id vpc-0ac8ec3d7ac633b3f
```

Assign Name to Public Route Table

```
-> aws ec2 create-tags --resources rtb-0bf29cc8eaa99227d --tags Key=Name,Value=PublicRouteTable
```

Associate 2 public subnets to PublicRouteTable

```
-> aws ec2 associate-route-table --route-table-id rtb-0bf29cc8eaa99227d --subnet-id subnet-0a5083f882cf209c0
```

```
-> aws ec2 associate-route-table --route-table-id rtb-0bf29cc8eaa99227d --subnet-id subnet-0e1603ed47a79c6a4
```

Private Route Table

Create Private Route Table in NewVPC

```
-> aws ec2 create-route-table --vpc-id vpc-0ac8ec3d7ac633b3f
```

Assign Name to Private Route Table

```
-> aws ec2 create-tags --resources rtb-07a3ffabeafba520f --tags Key=Name,Value=PrivateRouteTable
```

Associate 2 Private subnets to PrivateRouteTable

```
-> aws ec2 associate-route-table --route-table-id rtb-07a3ffabeafba520f --subnet-id subnet-0ee3aa017f3c459ad
```

```
-> aws ec2 associate-route-table --route-table-id rtb-07a3ffabeafba520f --subnet-id subnet-0b66c001c91288ecc
```

Internet Gateway

Create Internet Gateway

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

Assign Name to Internet Gateway

```
-> aws ec2 create-tags --resources igw-0280f1e2953d28645 --tags
Key=Name,Value=InternetGateway
```

Attach Internet Gateway to VPC

```
-> aws ec2 attach-internet-gateway --vpc-id vpc-0ac8ec3d7ac633b3f --internet-gateway-
id igw-0280f1e2953d28645
```

Routing the PublicRouteTable to the Internet Gateway

```
-> aws ec2 create-route --route-table-id rtb-0bf29cc8eaa99227d --destination-cidr-
block 0.0.0.0/0 --gateway-id igw-0280f1e2953d28645
```

Allocate Elastic IP

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

NAT Gateway

Creating NAT gateway in PublicSubnet2

```
-> aws ec2 create-nat-gateway --subnet-id subnet-0e1603ed47a79c6a4 --allocation-id
eipalloc-0ec6d43337d620b0f
```

Assigning name to created NAT gateway as NATGateway

```
-> aws ec2 create-tags --resources nat-0701db2a2f390297e --tags
Key=Name,Value=NATGateway
```

Route the PrivateRouteTable to the NAT Gateway

```
-> aws ec2 create-route --route-table-id rtb-07a3ffabeafba520f --destination-cidr-
block 0.0.0.0/0 --nat-gateway-id nat-0701db2a2f390297e
```

AMI

Creating image of Frontend Instance (already configured)

```
aws ec2 create-image --instance-id i-0b65400ae6c85a2ae --name "FrntendEc2AMI" --no-
reboot
```

Creating image of Backend-server Instance (already configured)

```
aws ec2 create-image --instance-id i-06f47454e3394bcb8 --name "BackendEc2AMI" --no-
reboot
```

Creating image of Database Instance (already configured)

```
aws ec2 create-image --instance-id i-0185bf23d576cdc2d --name "DatabaseEc2AMI" --no-
reboot
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-image --instance-id i-0b65400ae6c85a2ae --name "FrntendEc2AMI" --no-reboot
{
    "ImageId": "ami-07ad6baf20038a6ba"
}

DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-image --instance-id i-06f47454e3394bcb8 --name "BackendEc2AMI" --no-reboot
{
    "ImageId": "ami-0f60148b9fb4a44a3"
}

DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-image --instance-id i-0185bf23d576cdc2d --name "DatabaseEc2AMI" --no-reboot
{
    "ImageId": "ami-011ffab606315f6ad"
}
```

List the Amazon Machine Images(AMI) with ami-id and name

```
aws ec2 describe-images --owners self --query 'Images[*].[ImageId,Name]' --output
table
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 describe-images --owners self --query 'Images[*].[ImageId,Name]' --output table
-------------------------------------------
|             DescribeImages              |
+-------------------------+-----------------+
|  ami-07750c1d9625b72f2  |  FrontendEc2AMI |
|  ami-0f60148b9fb4a44a3  |  BackendEc2AMI  |
|  ami-0621f682dd0af91e6  |  Backend-AMI    |
|  ami-0d4e64c6f55c335e4  |  Frontend-ami   |
|  ami-011ffab606315f6ad  |  DatabaseEc2AMI |
+-------------------------+-----------------+
```

Launch a Frontend instance with existed ami, in PublicSubnet1

Launch an Ec2 instance

```
Aws ec2 run-instances --image-id ami-07750c1d9625b72f2 --instance-type t2.micro --
subnet-id subnet-0a5083f882cf209c0 --key-name project-1 --associate-public-ip-address
```

Assign name to Frontend instance as NewFrontend

```
aws ec2 create-tags --resources i-0b914903718557bf4 --tags Key=Name,Value=NewFrontend
```

Display the Ec2 id, name, type & Public IP

```
aws ec2 describe-instances  --instance-ids i-0b914903718557bf4 --query
'Reservations[*].Instances[*].[InstanceId, Tags[?Key==`Name`].Value | [0],
InstanceType, PublicIpAddress]' --output table
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-tags --resources i-0b914903718557bf4 --tags Key=Name,Value=NewFrontend

DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 describe-instances  --instance-ids i-0b914903718557bf4 --query 'Reservations[*].Instances[*].[InstanceId, Tags[?Key==`Name`].Value | [0], InstanceType, Public
Ip]' --output table
-------------------------------------------------------------------
|                       DescribeInstances                         |
+---------------------+-------------+----------+------------------+
|  i-0b914903718557bf4|  NewFrontend|  t2.micro|  13.201.16.92    |
+---------------------+-------------+----------+------------------+
```

Launch a Backend instance with existed ami, in PrivateSubnet1

Launch an Ec2 instance

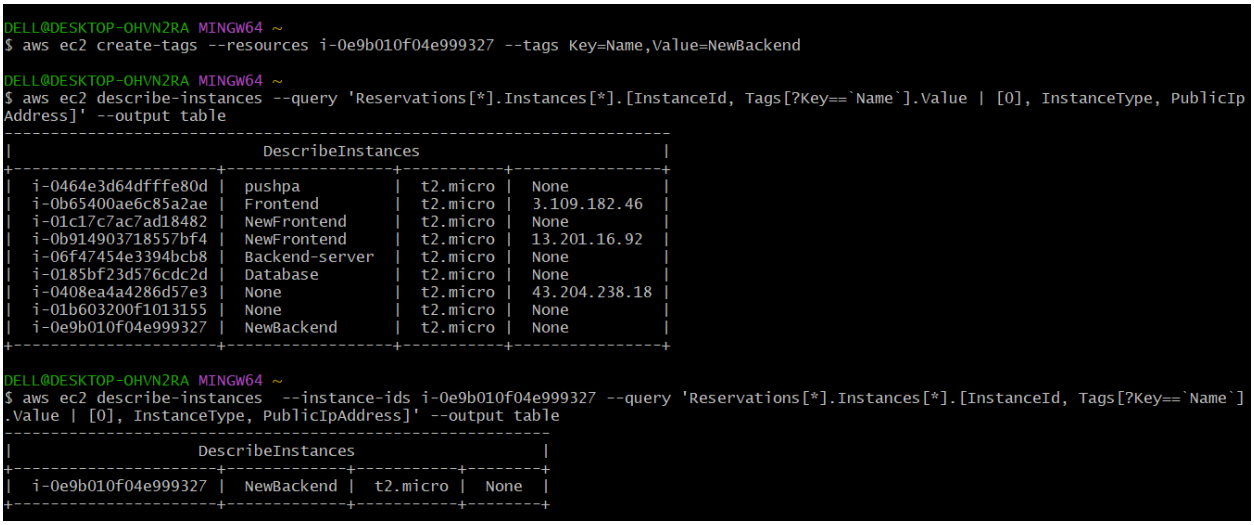
```
Aws ec2 run-instances  --image-id ami-0f60148b9fb4a44a3 --instance-type t2.micro --
subnet-id subnet-0ee3aa017f3c459ad --key-name project-1
```

Assign name to Frontend instance as NewBackend

```
aws ec2 create-tags --resources i-0e9b010f04e999327 --tags Key=Name,Value=NewBackend
```

Display the Ec2 id,name,type & Public IP

```
aws ec2 describe-instances  --instance-ids i-0e9b010f04e999327 --query
'Reservations[*].Instances[*].[InstanceId, Tags[?Key==`Name`].Value | [0],
InstanceType, PublicIpAddress]' --output table
```



Launch a Database instance with existed ami, in PrivateSubnet1

Launch an Ec2 instance

```
Aws ec2 run-instances --image-id ami-011ffab606315f6ad --instance-type t2.micro --
subnet-id subnet-0c8e31d0e6e06b0ee --key-name project-1
```

Assign name to Frontend instance as NewDatabase

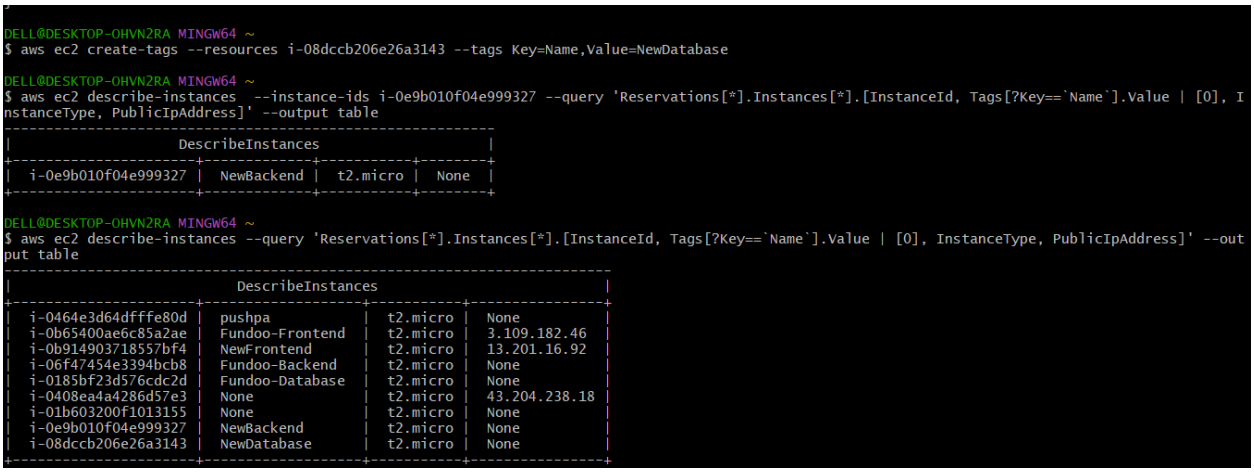
```
aws ec2 create-tags --resources i-08dccb206e26a3143 --tags Key=Name,Value=NewDatabase
```

To display the id,Name & Public Ip of specific Ec2 instance

```
aws ec2 describe-instances  --instance-ids i-08dccb206e26a3143 --query
'Reservations[*].Instances[*].[InstanceId, Tags[?Key==`Name`].Value | [0],
InstanceType, PublicIpAddress]' --output table
```

To Display all the Ec2 instance in table format

```
aws ec2 describe-instances --query 'Reservations[*].Instances[*].[InstanceId,
Tags[?Key==`Name`].Value | [0], InstanceType, PublicIpAddress]' --output table
```



Security Groups

Frontend Security Groups

```
aws ec2 create-security-group --group-name Frontend-sg-CLI --description "Security
group for frontend instance using CLI" --vpc-id vpc-0ac8ec3d7ac633b3f
```

"GroupId": "sg-0abe4db7856c0aaca"

Allow HTTP (Port 80)

```
aws ec2 authorize-security-group-ingress --group-id sg-0abe4db7856c0aaca --protocol
tcp --port 80 --cidr 0.0.0.0/0
```

Allow Custom TCP (Port 8000)

```
aws ec2 authorize-security-group-ingress --group-id sg-0abe4db7856c0aaca --protocol
tcp --port 8000 --cidr 0.0.0.0/0
```

Allow SSH (Port 22)

```
aws ec2 authorize-security-group-ingress --group-id sg-0abe4db7856c0aaca --protocol
tcp --port 22 --cidr 0.0.0.0/0
```

Attach the Security Group to Frontend Instance

```
aws ec2 modify-instance-attribute --instance-id i-0b914903718557bf4 --groups sg-
0abe4db7856c0aaca
```

To display the rules of Frontend security group

```
aws ec2 describe-security-groups --group-ids sg-0abe4db7856c0aaca --query
'SecurityGroups[*].IpPermissions' --output table
```

Backend Security Groups

```
aws ec2 create-security-group --group-name BackendSecurityGroup --description
"Security group for Backend instance using CLI" --vpc-id vpc-0ac8ec3d7ac633b3f
```

"GroupId": "sg-0e8e789d76e1d6c89"

Allow SSH (Port 22)

```
aws ec2 authorize-security-group-ingress --group-id sg-0e8e789d76e1d6c89
--protocol tcp --port 22 --cidr 0.0.0.0/0
```

Allow PostgreSQL (Port 5432)

```
aws ec2 authorize-security-group-ingress --group-id sg-0e8e789d76e1d6c89
--protocol tcp --port 5432 --cidr 0.0.0.0/0
```

Allow Custom TCP (Port 8000)

```
aws ec2 authorize-security-group-ingress --group-id sg-0e8e789d76e1d6c89
--protocol tcp --port 8000 --cidr 0.0.0.0/0
```

Attach the Security Group to Backend Instance

```
aws ec2 modify-instance-attribute --instance-id i-0b914903718557bf4 --groups sg-
0abe4db7856c0aaca
```

Database Security Group

```
aws ec2 create-security-group --group-name BackendSecurityGroup --description
"Database Security group using CLI" --vpc-id vpc-0ac8ec3d7ac633b3f
```

"GroupId": " sg-069a1abb1e848f7e8"

Allow SSH (Port 22)

```
aws ec2 authorize-security-group-ingress --group-id <your-security-group-id> --
protocol tcp --port 22 --cidr 0.0.0.0/0
```

Allow PostgreSQL (Port 5432)

```
aws ec2 authorize-security-group-ingress --group-id sg-069a1abb1e848f7e8
 --protocol tcp --port 5432 --cidr 0.0.0.0/0
```

Attach the Security Group to Database Instance

```
aws ec2 modify-instance-attribute --instance-id i-08dccb206e26a3143 --groups sg-
069a1abb1e848f7e8
```

Display the Security groups

```
aws ec2 describe-security-groups --group-ids sg-0abe4db7856c0aaca sg-0e8e789d76e1d6c89
sg-069a1abb1e848f7e8 --query 'SecurityGroups[*].[GroupId,GroupName,Description,VpcId]'
--output table
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 describe-security-groups --group-ids sg-0abe4db7856c0aaca sg-0e8e789d76e1d6c89 sg-069a1abb1e848f7e8 --query 'SecurityGroups[
*].[GroupId,GroupName,Description,VpcId]' --output table
-------------------------------------------------------------------------------------------------------------------------
|                                                DescribeSecurityGroups                                                 |
+----------------------+------------------+-----------------------------------------------------+-----------------------+
|  sg-0abe4db7856c0aaca|  Frontend-sg-CLI |  Security group for frontend instance using CLI     |  vpc-0ac8ec3d7ac633b3f  |
|  sg-069a1abb1e848f7e8|  Database-SG-CLI |  Database Security group using CLI                  |  vpc-0ac8ec3d7ac633b3f  |
|  sg-0e8e789d76e1d6c89|  Backend-sg-CLI  |  Security group for Backend instance using CLI      |  vpc-0ac8ec3d7ac633b3f  |
+----------------------+------------------+-----------------------------------------------------+-----------------------+

DELL@DESKTOP-OHVN2RA MINGW64 ~
$
```

Note:

Now I have tested that my application is running with Launched instances. Now I have create Load balancer to distribute traffic & ASG to auto create instances based on desired capacity.

Target Groups

Backend Target Group

```
aws elbv2 create-target-group --name Backend-TG-CLI --protocol HTTP --port 8000 --vpc-
id vpc-0ac8ec3d7ac633b3f --target-type instance --ip-address-type ipv4 --health-check-
path /home/
```

To display the target groups

```
aws elbv2 describe-target-groups --query
"TargetGroups[*].[TargetGroupName,TargetGroupArn]" --output table
```


```
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c",
            "TargetGroupName": "Backend-TG-CLI",
            "Protocol": "HTTP",
            "Port": 8000,
            "VpcId": "vpc-0ac8ec3d7ac633b3f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/home",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}


C:\Users\DELL>aws elbv2 describe-target-groups --query "TargetGroups[*].[TargetGroupName,TargetGroupArn]" --output table
-------------------------------------------------------------------------------------------------------------------------
|                                                DescribeTargetGroups                                                   |
+----------------------+------------------------------------------------------------------------------------------------+
|  Backend-TG-CLI      |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c  |
|  backend-target-group|  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/backend-target-group/2d5bc58d35bad6aa  |
|  frontend-target-group|  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/frontend-target-group/88a16c1476078db7  |
+----------------------+------------------------------------------------------------------------------------------------+
```

Register Backend instance to the Backend target group

```
aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c --targets Id=i-0e9b010f04e999327
```

Display the health of target group

```
aws elbv2 describe-target-health --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c --query "TargetHealthDescriptions[*].[Target.Id, TargetHealth.State]" --output table
```

```
C:\Users\DELL>aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c --targets Id=i-0e9b010f04e999327

C:\Users\DELL>aws elbv2 describe-target-health --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c --query "TargetHealthDescriptions[*].[Target.Id, TargetHealth.State]" --output table
---------------------------------
|      DescribeTargetHealth      |
+---------------------+----------+
|  i-0e9b010f04e999327 |  unused  |
+---------------------+----------+
```

To derigester the targets

```
aws elbv2 deregister-targets –target-group-arn <arn> --targets Id=<id>
```

Frontend Target Group

```
aws elbv2 create-target-group --name Frontend-TG-CLI --protocol HTTP --port 80 --vpc-id vpc-0ac8ec3d7ac633b3f --target-type instance --ip-address-type ipv4 --health-check-path /home/
```

```
C:\Users\DELL>aws elbv2 create-target-group --name Frontend-TG-CLI --protocol HTTP --port 80 --vpc-id vpc-0ac8ec3d7ac633b3f --target-type instance --ip-address-type ipv4 --health-check-path /home/
{
    "TargetGroups": [
        {
            "TargetGroupArn": "arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-TG-CLI/93007ad0aa208777",
            "TargetGroupName": "Frontend-TG-CLI",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-0ac8ec3d7ac633b3f",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "traffic-port",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 5,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/home/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1",
            "IpAddressType": "ipv4"
        }
    ]
}
```

To display the target groups

```
aws elbv2 describe-target-groups --query "TargetGroups[*].[TargetGroupName,TargetGroupArn]" --output table
```

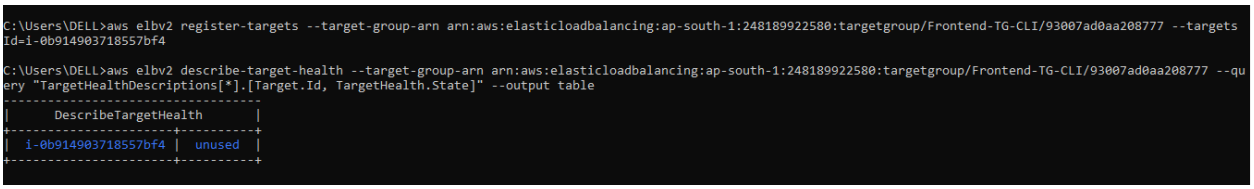
```
C:\Users\DELL>aws elbv2 describe-target-groups --query "TargetGroups[*].[TargetGroupName,TargetGroupArn]" --output table
-----------------------------------------------------------------------------------------------------------------------------------
|                                                    DescribeTargetGroups                                                         |
+----------------------+----------------------------------------------------------------------------------------------------------+
|  Backend-TG-CLI      |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c        |
|  Frontend-TG-CLI     |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-TG-CLI/93007ad0aa208777       |
|  backend-target-group |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/backend-target-group/2d5bc58d35bad6aa  |
|  frontend-target-group|  arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/frontend-target-group/88a16c1476078db7 |
+----------------------+----------------------------------------------------------------------------------------------------------+
```

Register Frontend instance to Frontend targets

```
aws elbv2 register-targets --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-TG-CLI/93007ad0aa208777 --targets Id=i-0b914903718557bf4
```

Display the health of target group

```
aws elbv2 describe-target-health --target-group-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-TG-CLI/93007ad0aa208777 --query "TargetHealthDescriptions[*].[Target.Id, TargetHealth.State]" --output table
```



Frontend Load Balancer

Creating Frontend Load Balancer

```
aws elbv2 create-load-balancer --name Frontend-LB-CLI --subnets subnet-0a5083f882cf209c0 subnet-0e1603ed47a79c6a4 --security-groups sg-0abe4db7856c0aaca --scheme internet-facing --type application
```

Creating listener for Frontend Load balancer

```
aws elbv2 create-listener --load-balancer-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Frontend-LB-CLI/f9532af2cba6ac1c --protocol HTTP --port 80 --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Frontend-TG-CLI/93007ad0aa208777
```

Display the Frontend Load balancer

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Frontend-LB-CLI/f9532af2cba6ac1c --output table
```

```
DELL@DESKTOP-OHVN2KA MINGW64 ~
$ aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Frontend-LB-CLI/f9532a
f2cba6ac1c --output table
-----------------------------------------------------------------------------------------------------------------------------------------------
|                                                                DescribeLoadBalancers                                                        |
+---------------------------------------------------------------------------------------------------------------------------------------------+
||                                                                 LoadBalancers                                                             ||
|+-------------------------------+-------------------------------------------------------------------------------------------------------------+|
||  CanonicalHostedZoneId        |  ZP97RAFLXTNZK                                                                                              ||
||  CreatedTime                  |  2024-11-17T14:37:16.997000+00:00                                                                           ||
||  DNSName                      |  Frontend-LB-CLI-1583095497.ap-south-1.elb.amazonaws.com                                                    ||
||  EnablePrefixForIpv6SourceNat|  off                                                                                                        ||
||  IpAddressType                |  ipv4                                                                                                       ||
||  LoadBalancerArn              |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Frontend-LB-CLI/f9532af2cba6ac1c     ||
||  LoadBalancerName             |  Frontend-LB-CLI                                                                                            ||
||  Scheme                       |  internet-facing                                                                                            ||
||  Type                         |  application                                                                                                ||
||  VpcId                        |  vpc-0ac8ec3d7ac633b3f                                                                                      ||
|+-------------------------------+-------------------------------------------------------------------------------------------------------------+|
|||                                                            AvailabilityZones                                                            |||
||+-----------------------------------------+---------------------------------------------------------------------------------------------------+||
|||  SubnetId                               |  subnet-0a5083f882cf209c0                                                                         |||
|||  ZoneName                               |  ap-south-1a                                                                                      |||
||+-----------------------------------------+---------------------------------------------------------------------------------------------------+||
|||                                                            AvailabilityZones                                                            |||
||+-----------------------------------------+---------------------------------------------------------------------------------------------------+||
|||  SubnetId                               |  subnet-0e1603ed47a79c6a4                                                                         |||
|||  ZoneName                               |  ap-south-1b                                                                                      |||
||+-----------------------------------------+---------------------------------------------------------------------------------------------------+||
|||                                                             SecurityGroups                                                              |||
||+---------------------------------------------------------------------------------------------------------------------------------------------+||
|||  sg-0abe4db7856c0aaca                                                                                                                       |||
||+---------------------------------------------------------------------------------------------------------------------------------------------+||
|||                                                                 State                                                                   |||
||+-------------------------------------------------------+-------------------------------------------------------------------------------------+||
|||  Code                                                 |  active                                                                             |||
||+-------------------------------------------------------+-------------------------------------------------------------------------------------+||
```

Backend Load Balancer

Creating Backend Load Balancer

```
aws elbv2 create-load-balancer --name Backend-LB-CLI --subnets subnet-0c8e31d0e6e06b0ee subnet-086c667c572d8c3fe --security-groups sg-0e8e789d76e1d6c89 --scheme internal --type application
```

Creating listener for Backend Load balancer

```
aws elbv2 create-listener --load-balancer-arn arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Backend-LB-CLI/75d6ea5ee02b267c --protocol HTTP --port 8000 --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c
```

Display the Backend Load balancer

```
aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Backend-LB-CLI/75d6ea5ee02b267c --output table
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws elbv2 describe-load-balancers --load-balancer-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Backend-LB-CLI/75d6ea5
ee02b267c --output table
```

```
-----------------------------------------------------------------------------------------------------------------------------------------------
|                                                        DescribeLoadBalancers                                                                |
+---------------------------------------------------------------------------------------------------------------------------------------------+
||                                                          LoadBalancers                                                                    ||
|+-----------------------------+-------------------------------------------------------------------------------------------------------------+|
||  CanonicalHostedZoneId      |  ZP97RAFLXTNZK                                                                                              ||
||  CreatedTime                |  2024-11-17T15:54:49.262000+00:00                                                                           ||
||  DNSName                    |  internal-Backend-LB-CLI-2045494817.ap-south-1.elb.amazonaws.com                                            ||
||  EnablePrefixForIpv6SourceNat|  off                                                                                                       ||
||  IpAddressType              |  ipv4                                                                                                       ||
||  LoadBalancerArn            |  arn:aws:elasticloadbalancing:ap-south-1:248189922580:loadbalancer/app/Backend-LB-CLI/75d6ea5ee02b267c      ||
||  LoadBalancerName           |  Backend-LB-CLI                                                                                             ||
||  Scheme                     |  internal                                                                                                   ||
||  Type                       |  application                                                                                                ||
||  VpcId                      |  vpc-0ac8ec3d7ac633b3f                                                                                      ||
|+-----------------------------+-------------------------------------------------------------------------------------------------------------+|
|||                                                    AvailabilityZones                                                                  |||
||+-----------------------------------+-------------------------------------------------------------------------------------------------+||
|||  SubnetId                         |  subnet-086c667c572d8c3fe                                                                       |||
|||  ZoneName                         |  ap-south-1c                                                                                    |||
||+-----------------------------------+-------------------------------------------------------------------------------------------------+||
|||                                                    AvailabilityZones                                                                  |||
||+-----------------------------------+-------------------------------------------------------------------------------------------------+||
|||  SubnetId                         |  subnet-0c8e31d0e6e06b0ee                                                                       |||
|||  ZoneName                         |  ap-south-1b                                                                                    |||
||+-----------------------------------+-------------------------------------------------------------------------------------------------+||
|||                                                     SecurityGroups                                                                    |||
||+---------------------------------------------------------------------------------------------------------------------------------------+||
|||  sg-0e8e789d76e1d6c89                                                                                                               |||
||+---------------------------------------------------------------------------------------------------------------------------------------+||
|||                                                        State                                                                          |||
||+-----------------------------------------------------+-------------------------------------------------------------------------------+||
|||  Code                                               |  active                                                                       |||
||+-----------------------------------------------------+-------------------------------------------------------------------------------+||
```

Launch Templates

Frontend Launch Template

```
aws ec2 create-launch-template --launch-template-name Frontend-lt-CLI --launch-template-data "{\"ImageId\": \"ami-0258e417b14bd44ee\", \"InstanceType\": \"t2.micro\", \"KeyName\": \"project-1\", \"SecurityGroupIds\": [\"sg-0abe4db7856c0aaca\"]}"
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-launch-template --launch-template-name Frontend-lt-CLI --launch-template-data "{\"ImageId\": \" ami-0258e417b14bd44ee
\", \"InstanceType\": \"t2.micro\", \"KeyName\": \"project-1\", \"SecurityGroupIds\": [\"sg-0abe4db7856c0aaca\"]}"
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-025bff741152440b8",
        "LaunchTemplateName": "Frontend-lt-CLI",
        "CreateTime": "2024-11-18T05:59:57+00:00",
        "CreatedBy": "arn:aws:iam::248189922580:root",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Backend Launch Template

```
aws ec2 create-launch-template --launch-template-name Backend-lt-CLI --launch-template-data "{\"ImageId\": \"ami-063665082b90579c7\", \"InstanceType\": \"t2.micro\", \"KeyName\": \"project-1\", \"SecurityGroupIds\": [\"sg-0e8e789d76e1d6c89\"]}"
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 create-launch-template --launch-template-name Backend-lt-CLI --launch-template-data "{\"ImageId\": \"ami-063665082b90579c7\"
, \"InstanceType\": \"t2.micro\", \"KeyName\": \"project-1\", \"SecurityGroupIds\": [\"sg-0e8e789d76e1d6c89\"]}"
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-05fcd34d9b5017115",
        "LaunchTemplateName": "Backend-lt-CLI",
        "CreateTime": "2024-11-18T06:01:42+00:00",
        "CreatedBy": "arn:aws:iam::248189922580:root",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
```

Display the Launch Templates

```
aws ec2 describe-launch-templates --launch-template-ids lt-025bff741152440b8 lt-05fcd34d9b5017115 --query "LaunchTemplates[*].[LaunchTemplateName,LaunchTemplateId]" --output table
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws ec2 describe-launch-templates --launch-template-ids lt-025bff741152440b8 lt-05fcd34d9b5017115 --query "LaunchTemplates[*].[Launc
hTemplateName,LaunchTemplateId]" --output table
-----------------------------------------------
|          DescribeLaunchTemplates            |
+-----------------+-----------------------+
|  Backend-lt-CLI  |  lt-05fcd34d9b5017115  |
|  Frontend-lt-CLI |  lt-025bff741152440b8  |
+-----------------+-----------------------+
```

Auto Scaling Group

Frontend Auto scaling

Create Frontend ASG

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name Frontend-auto-CLI --launch-template "LaunchTemplateName=Frontend-lt-CLI,Version=1" --min-size 1 --max-size 4 --desired-capacity 2 --vpc-zone-identifier subnet-0a5083f882cf209c0,subnet-0e1603ed47a79c6a4 --availability-zones ap-south-1a ap-south-1b
```

Create a Scaling Policy

```
aws autoscaling put-scaling-policy --auto-scaling-group-name Frontend-auto-CLI --policy-name TargetScalingPolicy --policy-type TargetTrackingScaling --target-tracking-configuration "{\"TargetValue\":70.0, \"PredefinedMetricSpecification\": {\"PredefinedMetricType\": \"ASGAverageCPUUtilization\"}, \"DisableScaleIn\": false}"
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws autoscaling create-auto-scaling-group --auto-scaling-group-name Frontend-auto-CLI --launch-template "LaunchTemplateName=Frontend
-lt-CLI,Version=1" --min-size 1 --max-size 4 --desired-capacity 2 --vpc-zone-identifier subnet-0a5083f882cf209c0,subnet-0e1603ed47a79c
6a4 --availability-zones ap-south-1a ap-south-1b

DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws autoscaling put-scaling-policy --auto-scaling-group-name Frontend-auto-CLI --policy-name TargetScalingPolicy --policy-type Targe
tTrackingScaling --target-tracking-configuration "{\"TargetValue\":70.0, \"PredefinedMetricSpecification\": {\"PredefinedMetricType\":
 \"ASGAverageCPUUtilization\"}, \"DisableScaleIn\": false}"
{
    "PolicyARN": "arn:aws:autoscaling:ap-south-1:248189922580:scalingPolicy:b776c0a1-bcae-446a-b065-bd3cea5fa69b:autoScalingGroupName/
Frontend-auto-CLI:policyName/TargetScalingPolicy",
    "Alarms": [
        {
            "AlarmName": "TargetTracking-Frontend-auto-CLI-AlarmHigh-e6da4b3d-6f58-4a4a-a89a-fed42cde6e44",
            "AlarmARN": "arn:aws:cloudwatch:ap-south-1:248189922580:alarm:TargetTracking-Frontend-auto-CLI-AlarmHigh-e6da4b3d-6f58-4a4
a-a89a-fed42cde6e44"
        },
        {
            "AlarmName": "TargetTracking-Frontend-auto-CLI-AlarmLow-5fff09a1-d0e5-488c-a708-2b35f3e31fe1",
            "AlarmARN": "arn:aws:cloudwatch:ap-south-1:248189922580:alarm:TargetTracking-Frontend-auto-CLI-AlarmLow-5fff09a1-d0e5-488c
-a708-2b35f3e31fe1"
        }
    ]
}
```

Display policy for Auto scaling Group

```
aws autoscaling describe-policies --auto-scaling-group-name Frontend-auto-CLI --query "ScalingPolicies[?PolicyName=='TargetScalingPolicy']" --output table
```

Attach load balancer to Auto scaling group

```
aws autoscaling attach-load-balancer-target-groups --auto-scaling-group-name Frontend-auto-CLI --target-group-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/frontend-target-group/88a16c1476078db7 --region ap-south-1
```

Display the info related to Frontend Auto scaling group

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names Frontend-auto-CLI --region ap-south-1 --output table --query "AutoScalingGroups[*].[DesiredCapacity,MinSize,MaxSize,LaunchTemplate.LaunchTemplateName,AutoScalingGroupARN]"
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names Frontend-auto-CLI --region ap-south-1 --output table --query "AutoScalingGr
oups[*].[DesiredCapacity,MinSize,MaxSize,LaunchTemplate.LaunchTemplateName,AutoScalingGroupARN]"
-------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------
|                                                                DescribeAutoScalingGroups
                        |
+---+----+----+-----------------+-----------------------------------------------------------------------------------------------------------------
-------------------------+
|  2|  1 |  4 |  Frontend-lt-CLI |  arn:aws:autoscaling:ap-south-1:248189922580:autoScalingGroup:1dc03d04-bc89-4f0c-8010-2f591c3cadaf:autoScalingGrou
pName/Frontend-auto-CLI   |
+---+----+----+-----------------+-----------------------------------------------------------------------------------------------------------------
-------------------------+
```

Update capacity

```
aws autoscaling update-auto-scaling-group --auto-scaling-group-name Frontend-auto-CLI --desired-capacity 2
```

Backend Auto scaling

Create Backend ASG

```
aws autoscaling create-auto-scaling-group --auto-scaling-group-name Backend-auto-CLI --launch-template "LaunchTemplateName=Backend-lt-CLI,Version=1" --min-size 1 --max-size 4 --desired-capacity 2 --vpc-zone-identifier subnet-0ee3aa017f3c459ad,subnet-086c667c572d8c3fe --availability-zones ap-south-1b ap-south-1c
```

Create a Scaling Policy

```
aws autoscaling put-scaling-policy --auto-scaling-group-name Backend-auto-CLI --policy-name TargetScalingPolicy --policy-type TargetTrackingScaling --target-tracking-configuration "{\"TargetValue\":70.0, \"PredefinedMetricSpecification\": {\"PredefinedMetricType\": \"ASGAverageCPUUtilization\"}, \"DisableScaleIn\": false}"
```

```
DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws autoscaling create-auto-scaling-group --auto-scaling-group-name Backend-auto-CLI --launch-template "LaunchTemplateName=Backend-lt-CLI,Version=1
" --min-size 1 --max-size 4 --desired-capacity 2 --vpc-zone-identifier subnet-0ee3aa017f3c459ad,subnet-086c667c572d8c3fe --availability-zones ap-sout
h-1b ap-south-1c

DELL@DESKTOP-OHVN2RA MINGW64 ~
$ aws autoscaling put-scaling-policy --auto-scaling-group-name Backend-auto-CLI --policy-name TargetScalingPolicy --policy-type TargetTrackingScaling
 --target-tracking-configuration "{\"TargetValue\":70.0, \"PredefinedMetricSpecification\": {\"PredefinedMetricType\": \"ASGAverageCPUUtilization\"},
 \"DisableScaleIn\": false}"

{
    "PolicyARN": "arn:aws:autoscaling:ap-south-1:248189922580:scalingPolicy:4bb093e8-6f59-4560-b5d0-138df143699a:autoScalingGroupName/Backend-auto-CL
I:policyName/TargetScalingPolicy",
    "Alarms": [
        {
            "AlarmName": "TargetTracking-Backend-auto-CLI-AlarmHigh-aa618c18-87c6-47f8-ab2b-a7e75a255a67",
            "AlarmARN": "arn:aws:cloudwatch:ap-south-1:248189922580:alarm:TargetTracking-Backend-auto-CLI-AlarmHigh-aa618c18-87c6-47f8-ab2b-a7e75a255
a67"
        },
        {
            "AlarmName": "TargetTracking-Backend-auto-CLI-AlarmLow-162c8007-de13-4d63-8932-eb98f7b6eb79",
            "AlarmARN": "arn:aws:cloudwatch:ap-south-1:248189922580:alarm:TargetTracking-Backend-auto-CLI-AlarmLow-162c8007-de13-4d63-8932-eb98f7b6eb
79"
        }
    ]
}
```

Display policy for Auto scaling Group

```
aws autoscaling describe-policies --auto-scaling-group-name Backend-auto-CLI --query "ScalingPolicies[?PolicyName=='TargetScalingPolicy']" --output table
```

Attach load balancer to Auto scaling group

```
aws autoscaling attach-load-balancer-target-groups --auto-scaling-group-name Backend-auto-CLI --target-group-arns arn:aws:elasticloadbalancing:ap-south-1:248189922580:targetgroup/Backend-TG-CLI/718b64ee08450b0c --region ap-south-1
```

Display the info related to Frontend Auto scaling group

```
aws autoscaling describe-auto-scaling-groups --auto-scaling-group-names Backend-auto-CLI --region ap-south-1 --output table --query "AutoScalingGroups[*].[DesiredCapacity,MinSize,MaxSize,LaunchTemplate.LaunchTemplateName,AutoScalingGroupARN]"
```