Auto-Scaling for Application Load Balancer

	Name	Roll No	Seat no.
1.	Pranav Hatwar	6	T214132

1. Target Group

Code:-

```
x=1

while [ $x -ne 0 ]

do
   echo "-----------------------------"

   echo "****Target Group******"

   echo 1. Create Target Group

   echo 2. Register-targets

   echo 3. Describe-target-groups

   echo 4. Modify-target-group

   echo 5. Delete-target-group

   echo 6. Deregister-targets

   echo "-----------------------------"

   echo "Enter the choice from above:-"

   read ch

   case "$ch" in

 1)echo "Enter the  Target Group Name: "
```

```
read nam
echo "Enter Protocol : "
read protocol
echo "Enter Port : "
 read port
 echo "Enter  Type : "
read typet
echo "Enter vpc-id: "
read vpcid
echo "Enter Protocol  Version: "
read vprotocol
echo "Enter  health-check-protocol "
read protocol
echo "Enter health-check-port "
read port
echo "Enter  health-check-interval-seconds "
read hint
echo "Enter  health-check-timeout-seconds "
read htime
echo "Enter  healthy-threshold-count  "
```

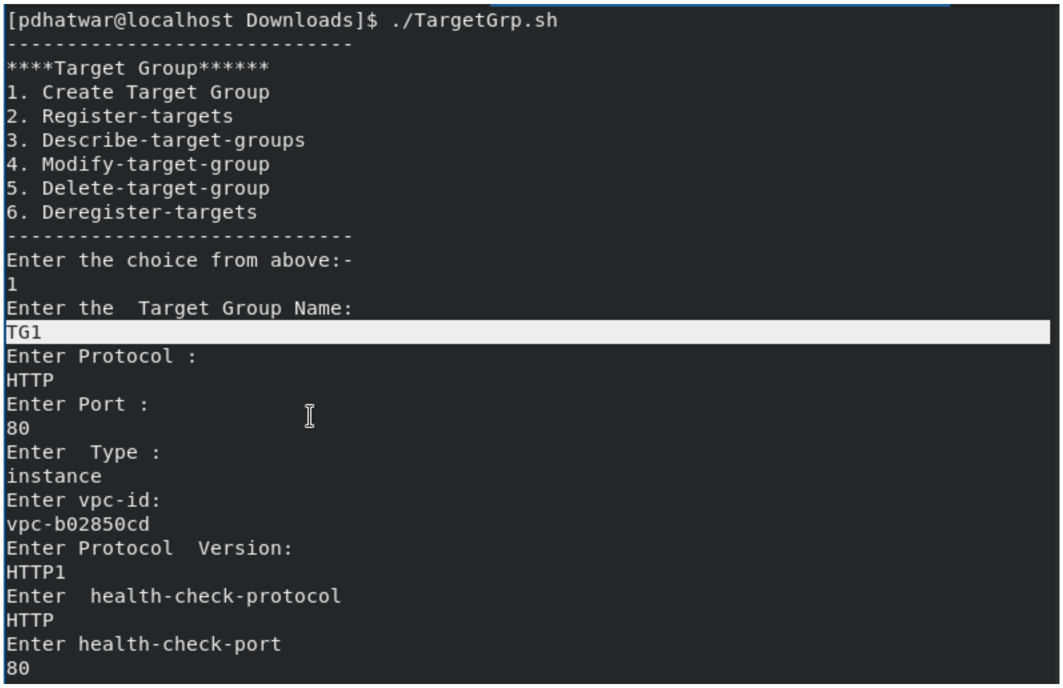
```
read hcount

echo "Enter  unhealthy-threshold-count "

read unhcount

aws elbv2 create-target-group --name $nam --protocol $protocol --port $port
--target-type $typet --vpc-id $vpcid --protocol-version $vprotocol
--health-check-protocol $protocol --health-check-port $port
--health-check-interval-seconds $hint --health-check-timeout-seconds $htime
--healthy-threshold-count $hcount --unhealthy-threshold-count $unhcount
echo " You have create-target-group Successfully";;

2) echo "Enter Amazon Resource Name "

read arn

echo "Enter instance/targets Id "

read tid

aws elbv2 register-targets --target-group-arn $arn --targets $tid

 echo " You have register-target-group Successfully " ;;

3) echo "Enter Amazon Resource Name "

read arn

aws elbv2 describe-target-groups --target-group-arn $arn

echo "Describe Sucessfully";;

4)echo "Enter Amazon Resource Name "

read arn

echo "Enter  health-check-protocol "
```

```
read protocol

echo "Enter health-check-port "

read port

echo "Enter  health-check-interval-seconds "

read hint

echo "Enter  health-check-timeout-seconds "

read htime

echo "Enter  healthy-threshold-count  "

read hcount

echo "Enter  unhealthy-threshold-count "

read unhcount

aws elbv2 modify-target-group --target-group-arn $arn --health-check-protocol
$protocol --health-check-port $port --health-check-interval-seconds $hint
--health-check-timeout-seconds $htime --healthy-threshold-count  $hcount
--unhealthy-threshold-count $unhcount
echo "Modify Sucessfully";;

5) echo "Enter Amazon Resource Name "

read arn

aws elbv2 delete-target-group --target-group-arn $arn

echo "Deleted Sucessfully";;

6) echo "Enter Amazon Resource Name "

read arn
```

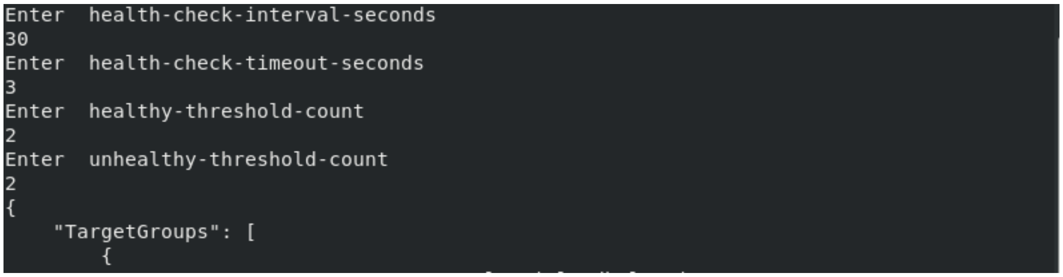
```
echo "Enter targets Id "

read tid

aws elbv2 deregister-targets --target-group-arn $arn --targets $tid
echo "The Deregister-targets Sucessfully";;

            *)echo "Invalid choice."

             esac

             echo "----------------------------------"

             echo "Enter 1 for continue and 0 for exit."

             read x

done

clear
```

Creating 2 Target Group Name TG1, TG2

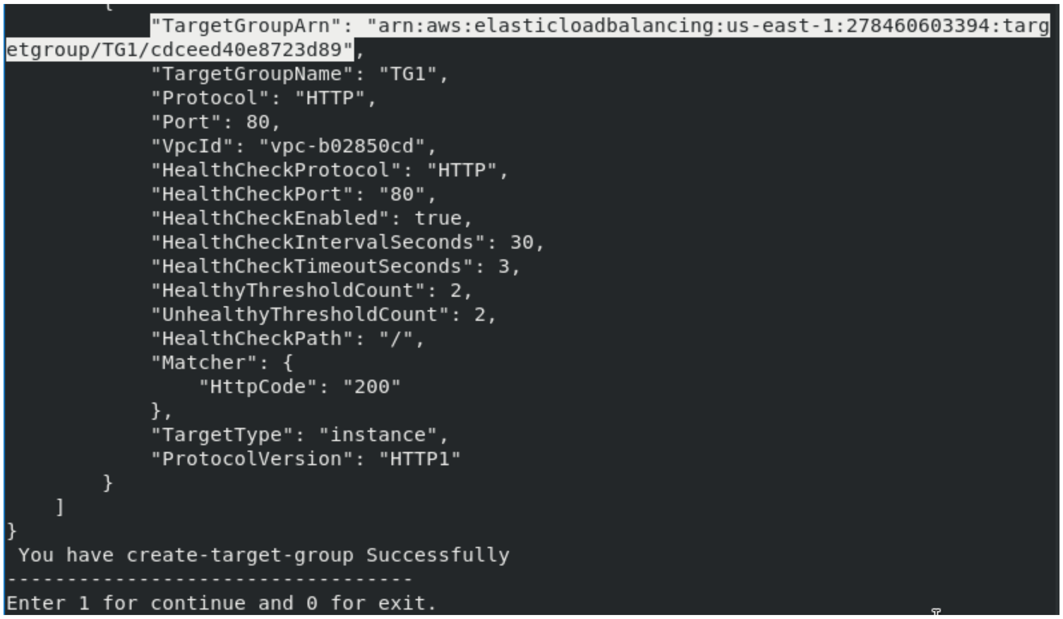
● TG1



```
[pdhatwar@localhost Downloads]$ ./TargetGrp.sh
-----------------------------
****Target Group******
1. Create Target Group
2. Register-targets
3. Describe-target-groups
4. Modify-target-group
5. Delete-target-group
6. Deregister-targets
-----------------------------
Enter the choice from above:-
1
Enter the  Target Group Name:
TG1
Enter Protocol :
HTTP
Enter Port :
80
Enter  Type :
instance
Enter vpc-id:
vpc-b02850cd
Enter Protocol  Version:
HTTP1
Enter  health-check-protocol
HTTP
Enter health-check-port
80
```

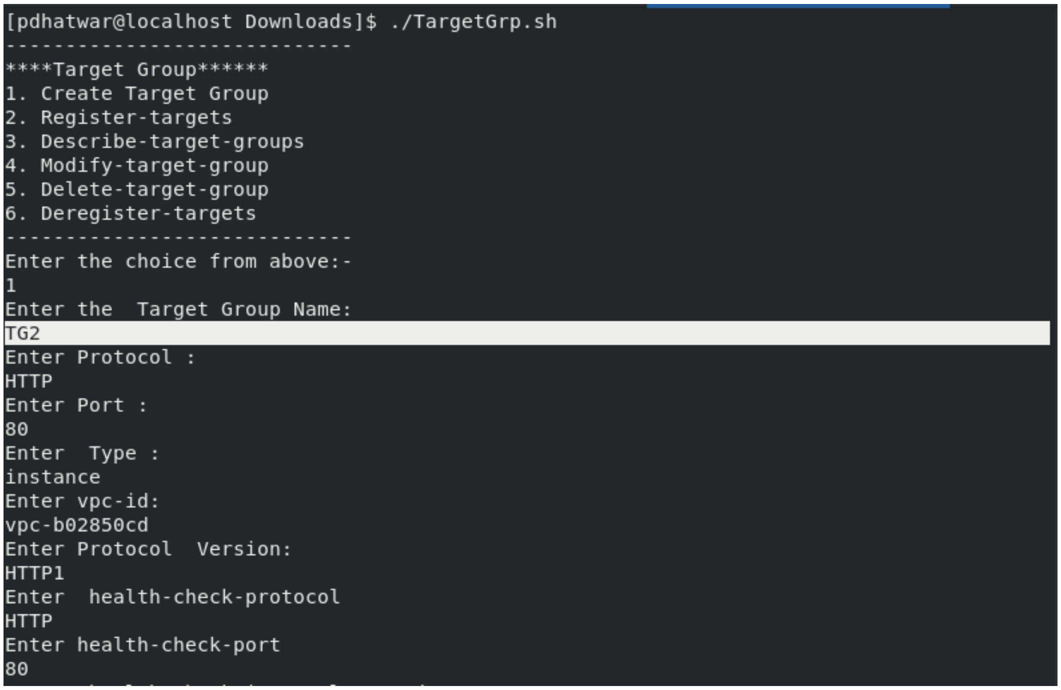


```
Enter  health-check-interval-seconds
30
Enter  health-check-timeout-seconds
3
Enter  healthy-threshold-count
2
Enter  unhealthy-threshold-count
2
{
    "TargetGroups": [
        {
```

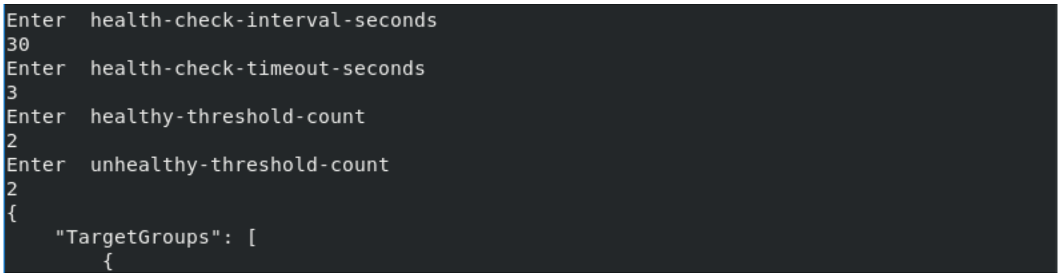


```
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:targ
etgroup/TG1/cdceed40e8723d89",
            "TargetGroupName": "TG1",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-b02850cd",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "80",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 3,
            "HealthyThresholdCount": 2,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1"
        }
    ]
}
 You have create-target-group Successfully
----------------------------------
Enter 1 for continue and 0 for exit.
```

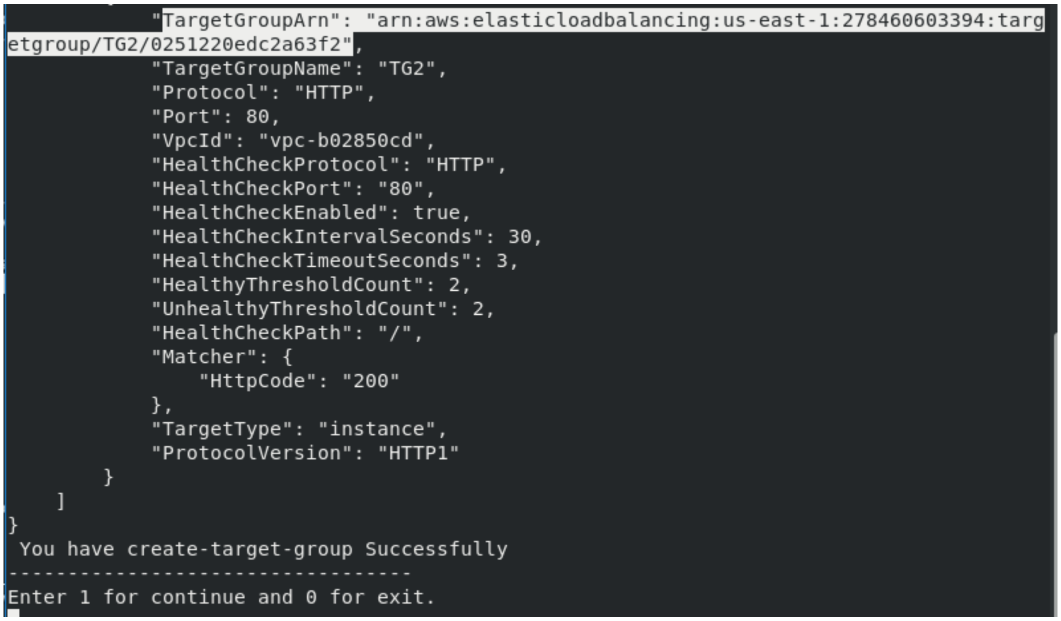
- **TG2**



```
[pdhatwar@localhost Downloads]$ ./TargetGrp.sh
-----------------------------
****Target Group******
1. Create Target Group
2. Register-targets
3. Describe-target-groups
4. Modify-target-group
5. Delete-target-group
6. Deregister-targets
-----------------------------
Enter the choice from above:-
1
Enter the  Target Group Name:
TG2
Enter Protocol :
HTTP
Enter Port :
80
Enter  Type :
instance
Enter vpc-id:
vpc-b02850cd
Enter Protocol  Version:
HTTP1
Enter  health-check-protocol
HTTP
Enter health-check-port
80
```



```
Enter  health-check-interval-seconds
30
Enter  health-check-timeout-seconds
3
Enter  healthy-threshold-count
2
Enter  unhealthy-threshold-count
2
{
    "TargetGroups": [
        {
```



```
            "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:targ
etgroup/TG2/0251220edc2a63f2",
            "TargetGroupName": "TG2",
            "Protocol": "HTTP",
            "Port": 80,
            "VpcId": "vpc-b02850cd",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckPort": "80",
            "HealthCheckEnabled": true,
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckTimeoutSeconds": 3,
            "HealthyThresholdCount": 2,
            "UnhealthyThresholdCount": 2,
            "HealthCheckPath": "/",
            "Matcher": {
                "HttpCode": "200"
            },
            "TargetType": "instance",
            "ProtocolVersion": "HTTP1"
        }
    ]
}
 You have create-target-group Successfully
---------------------------------
Enter 1 for continue and 0 for exit.
```

2. Load Balance

Code:-

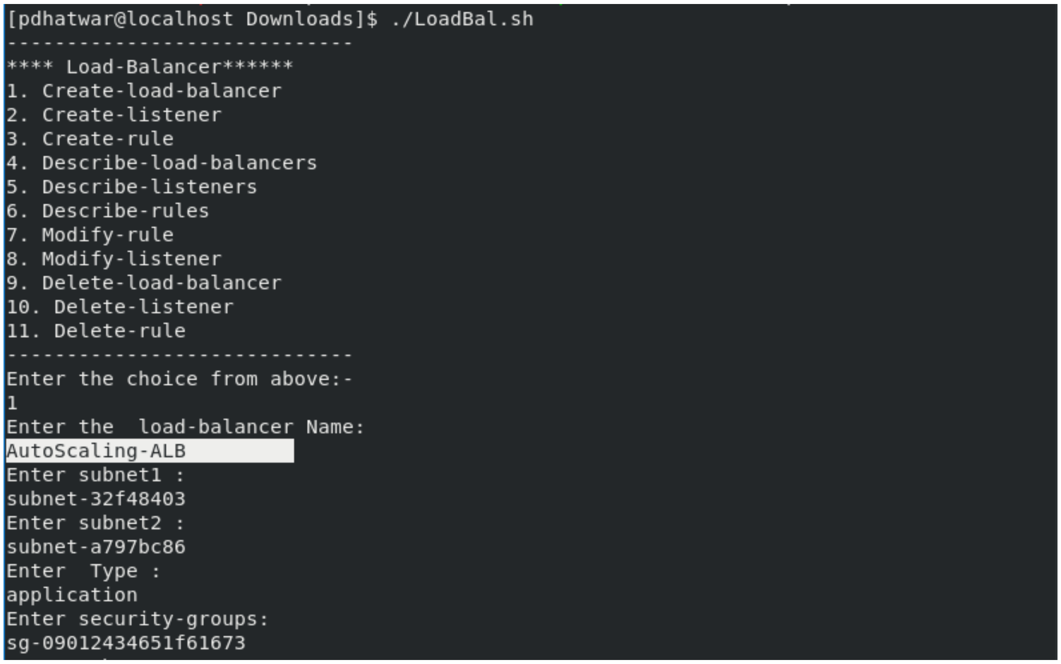
```
x=1

while [ $x -ne 0 ]

do
 echo "-----------------------------"

    echo "***** Load-Balancer*******"

    echo 1. Create-load-balancer

    echo 2. Create-listener

    echo 3. Create-rule

    echo 4. Describe-load-balancers

    echo 5. Describe-listeners

    echo 6. Describe-rules

   echo 7. Modify-rule

   echo 8. Modify-listener

   echo 9. Delete-load-balancer

   echo 10. Delete-listener

   echo 11. Delete-rule

    echo "-----------------------------"

    echo "Enter the choice from above:-"

    read ch
    case "$ch" in

 1)echo "Enter the  load-balancer Name: "

read nam

echo "Enter subnet1 : "

read sub1

echo "Enter subnet2 : "

read sub2

 echo "Enter  Type : "

read typ

echo "Enter security-groups: "

read sg

echo "Enter scheme: "

read sch

echo " Ip-address-type "

read ip

aws elbv2 create-load-balancer --name $nam --subnets $sub1 $sub2 --security-groups
$sg --scheme  $sch --type $typ --ip-address-type $ip

echo " You have Created-load-balancer Successfully " ;;

2) echo "Enter Amazon Resource Name for LB "

read arn

echo "Enter Amazon Resource Name for TG "
```

```
read tarn

echo "Enter  Protocol "

read protocol

echo "Enter Port "

read port

aws elbv2 create-listener --load-balancer-arn $arn --protocol $protocol --port $port --default-actions Type=forward,TargetGroupArn=$tarn

echo "You have created-listener Sucessfully";;

 3) echo "Enter Amazon Resource Name for Listener "

read larn

echo "Enter Amazon Resource Name for TG "

read tarn

echo "Enter  Priority "

read prio

echo "Enter Path "

read path

aws elbv2 create-rule --listener-arn $larn --priority $prio --conditions Field=path-pattern,Values='/$path/*' --actions Type=forward,TargetGroupArn=$tarn

echo "You have Create-rule Sucessfully";;

4)echo "Enter Amazon Resource Name for LB "

read arn
```

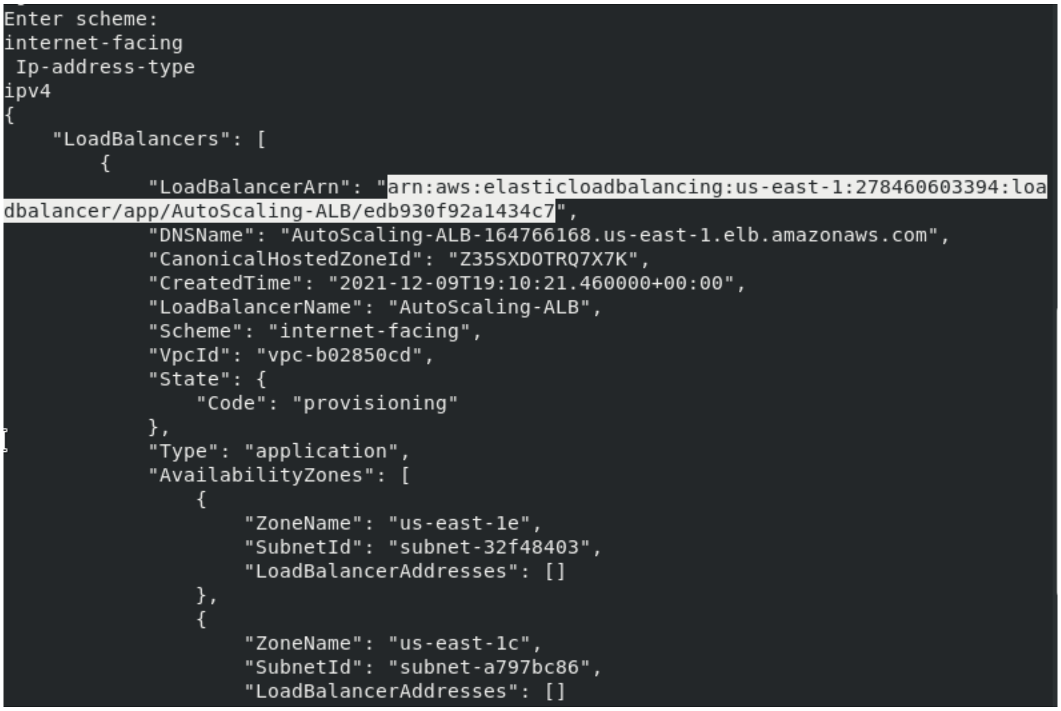
```
aws elbv2 describe-load-balancers --load-balancer-arns  $arn

echo "Describe-load-balancers Sucessfully";;

5)echo "Enter Amazon Resource Name for Listeners "

read arn

aws elbv2 describe-listeners --listener-arns $arn

echo "Describe-listeners Sucessfully";;

6)echo "Enter Amazon Resource Name  for Rule"

read arn

aws elbv2 describe-rules  --rule-arns $arn

echo " Describe-rules Sucessfully";;

7) echo "Enter Amazon Resource Name for Rule "

read rarn

echo "Enter Amazon Resource Name for TG "

read tarn

echo "Enter Path "

read path

aws elbv2 modify-rule --actions Type=forward,TargetGroupArn=$tarn --conditions
Field=path-pattern,Values='/$path/*' --rule-arn $rarn

echo " Modify-rule Sucessfully";;

8) echo "Enter Amazon Resource Name for Listener "
```

```
read larn

echo "Enter Amazon Resource Name for TG "

read tarn
aws elbv2 modify-listener --listener-arn $larn --default-actions
Type=forward,TargetGroupArn=$tarn

echo " Modify-listener Sucessfully";;

9) echo "Enter Amazon Resource Name for LB "

read arn

aws elbv2 delete-load-balancer  --load-balancer-arn $arn
echo " load-balancer Deleted Sucessfully";;

10) echo "Enter Amazon Resource Name  for Listener "

read arn

aws elbv2 delete-listener --listener-arn $arn
echo "Delete-listener  Sucessfully";;

11) echo "Enter Amazon Resource Name for Rule "

read arn

aws elbv2 delete-rule --rule-arn $arn
echo " Delete-rule Sucessfully";;

   *)echo "Invalid choice."
         esac
         echo "---------------------------------"
         echo "Enter 1 for continue and 0 for exit."
         read x

done

clear
```

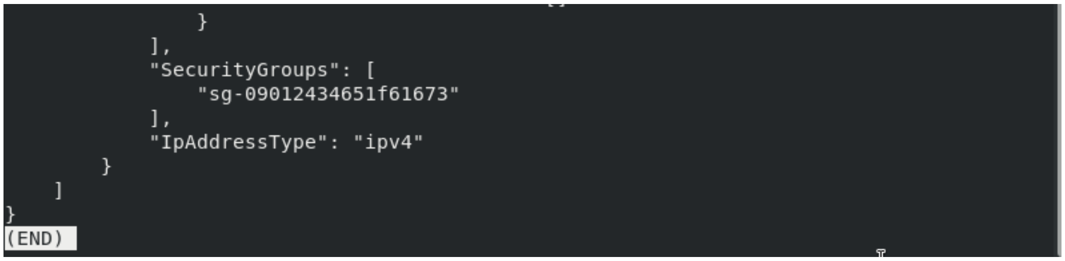
Creating Application Load Balance



```
[pdhatwar@localhost Downloads]$ ./LoadBal.sh
-----------------------------
**** Load-Balancer******
1. Create-load-balancer
2. Create-listener
3. Create-rule
4. Describe-load-balancers
5. Describe-listeners
6. Describe-rules
7. Modify-rule
8. Modify-listener
9. Delete-load-balancer
10. Delete-listener
11. Delete-rule
-----------------------------
Enter the choice from above:-
1
Enter the  load-balancer Name:
AutoScaling-ALB
Enter subnet1 :
subnet-32f48403
Enter subnet2 :
subnet-a797bc86
Enter  Type :
application
Enter security-groups:
sg-09012434651f61673
```

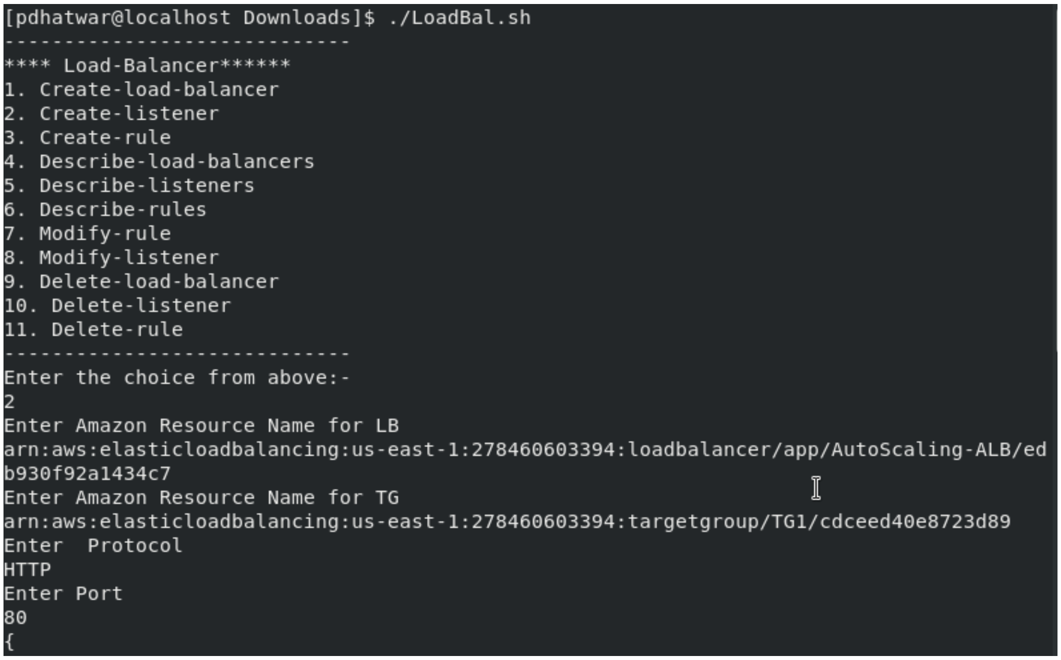


```
Enter scheme:
internet-facing
 Ip-address-type
ipv4
{
    "LoadBalancers": [
        {
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:loa
dbalancer/app/AutoScaling-ALB/edb930f92a1434c7",
            "DNSName": "AutoScaling-ALB-164766168.us-east-1.elb.amazonaws.com",
            "CanonicalHostedZoneId": "Z35SXDOTRQ7X7K",
            "CreatedTime": "2021-12-09T19:10:21.460000+00:00",
            "LoadBalancerName": "AutoScaling-ALB",
            "Scheme": "internet-facing",
            "VpcId": "vpc-b02850cd",
            "State": {
                "Code": "provisioning"
            },
            "Type": "application",
            "AvailabilityZones": [
                {
                    "ZoneName": "us-east-1e",
                    "SubnetId": "subnet-32f48403",
                    "LoadBalancerAddresses": []
                },
                {
                    "ZoneName": "us-east-1c",
                    "SubnetId": "subnet-a797bc86",
                    "LoadBalancerAddresses": []
```

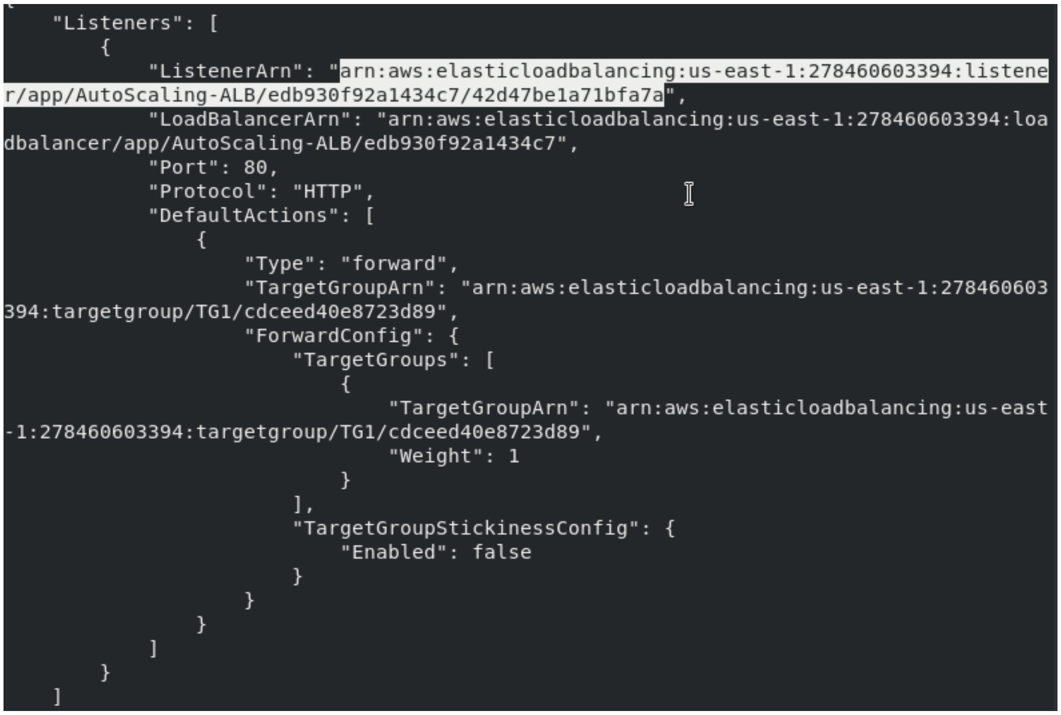


```
                }
            ],
            "SecurityGroups": [
                "sg-09012434651f61673"
            ],
            "IpAddressType": "ipv4"
        }
    ]
}
(END)
```

Creating Listener



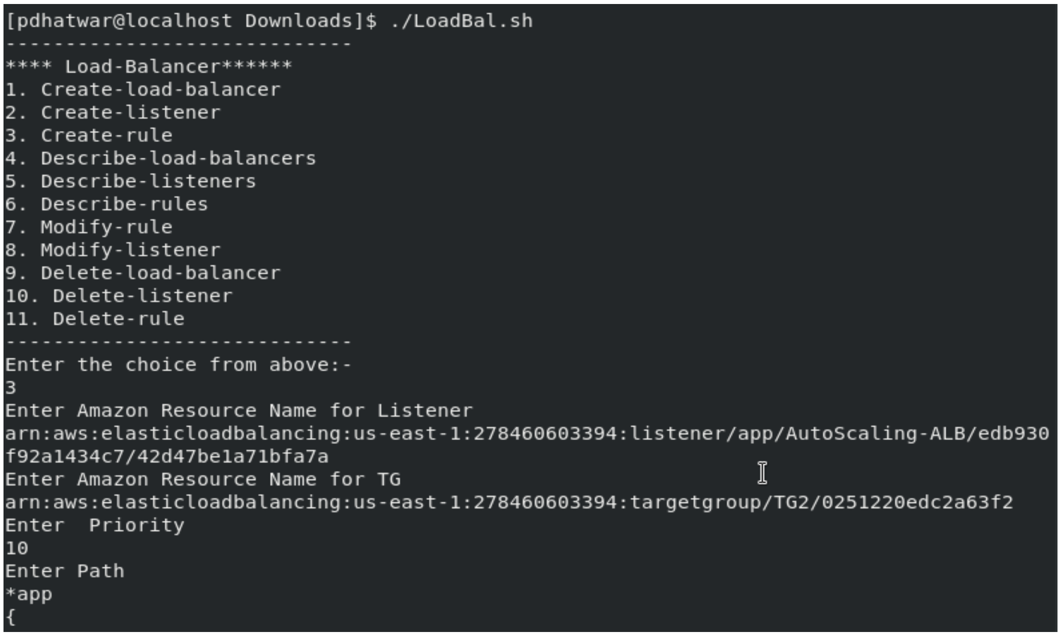
```
[pdhatwar@localhost Downloads]$ ./LoadBal.sh
-----------------------------
**** Load-Balancer******
1. Create-load-balancer
2. Create-listener
3. Create-rule
4. Describe-load-balancers
5. Describe-listeners
6. Describe-rules
7. Modify-rule
8. Modify-listener
9. Delete-load-balancer
10. Delete-listener
11. Delete-rule
-----------------------------
Enter the choice from above:-
2
Enter Amazon Resource Name for LB
arn:aws:elasticloadbalancing:us-east-1:278460603394:loadbalancer/app/AutoScaling-ALB/ed
b930f92a1434c7
Enter Amazon Resource Name for TG
arn:aws:elasticloadbalancing:us-east-1:278460603394:targetgroup/TG1/cdceed40e8723d89
Enter  Protocol
HTTP
Enter Port
80
{
```



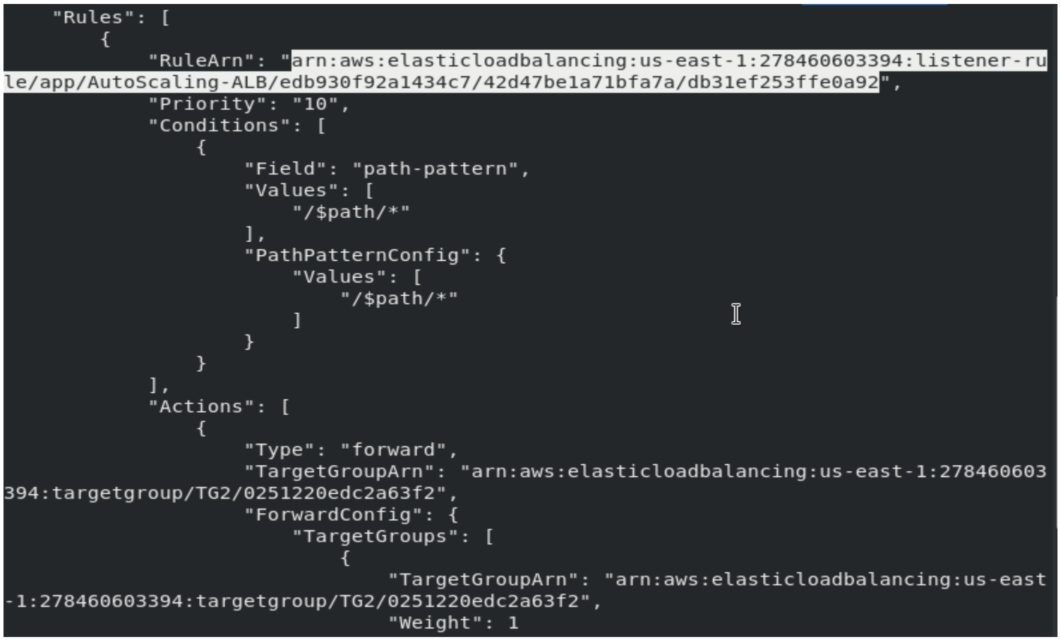
```
    "Listeners": [
        {
            "ListenerArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:listene
r/app/AutoScaling-ALB/edb930f92a1434c7/42d47be1a71bfa7a",
            "LoadBalancerArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:loa
dbalancer/app/AutoScaling-ALB/edb930f92a1434c7",
            "Port": 80,
            "Protocol": "HTTP",
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:278460603
394:targetgroup/TG1/cdceed40e8723d89",
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east
-1:278460603394:targetgroup/TG1/cdceed40e8723d89",
                                "Weight": 1
                            }
                        ],
                        "TargetGroupStickinessConfig": {
                            "Enabled": false
                        }
                    }
                }
            ]
        }
    ]
```

Adding Rule to Load Balancer

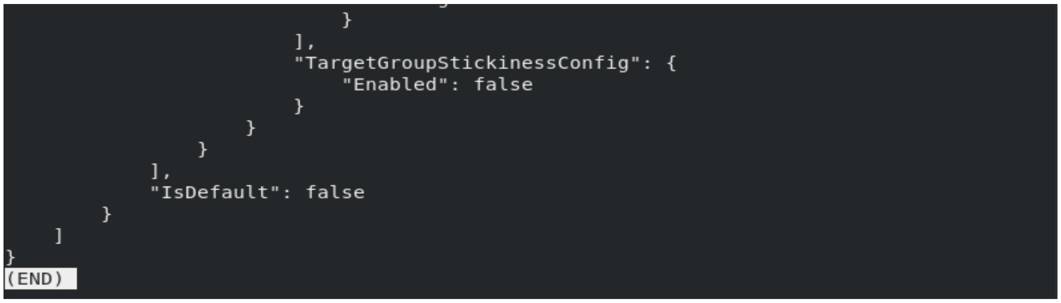
Rule for Target Group TG2



```
[pdhatwar@localhost Downloads]$ ./LoadBal.sh
-----------------------------
**** Load-Balancer******
1. Create-load-balancer
2. Create-listener
3. Create-rule
4. Describe-load-balancers
5. Describe-listeners
6. Describe-rules
7. Modify-rule
8. Modify-listener
9. Delete-load-balancer
10. Delete-listener
11. Delete-rule
-----------------------------
Enter the choice from above:-
3
Enter Amazon Resource Name for Listener
arn:aws:elasticloadbalancing:us-east-1:278460603394:listener/app/AutoScaling-ALB/edb930
f92a1434c7/42d47be1a71bfa7a
Enter Amazon Resource Name for TG
arn:aws:elasticloadbalancing:us-east-1:278460603394:targetgroup/TG2/0251220edc2a63f2
Enter  Priority
10
Enter Path
*app
{
```



```
    "Rules": [
        {
            "RuleArn": "arn:aws:elasticloadbalancing:us-east-1:278460603394:listener-ru
le/app/AutoScaling-ALB/edb930f92a1434c7/42d47be1a71bfa7a/db31ef253ffe0a92",
            "Priority": "10",
            "Conditions": [
                {
                    "Field": "path-pattern",
                    "Values": [
                        "/$path/*"
                    ],
                    "PathPatternConfig": {
                        "Values": [
                            "/$path/*"
                        ]
                    }
                }
            ],
            "Actions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east-1:278460603
394:targetgroup/TG2/0251220edc2a63f2",
                    "ForwardConfig": {
                        "TargetGroups": [
                            {
                                "TargetGroupArn": "arn:aws:elasticloadbalancing:us-east
-1:278460603394:targetgroup/TG2/0251220edc2a63f2",
                                "Weight": 1
```



```
                            }
                        ],
                        "TargetGroupStickinessConfig": {
                            "Enabled": false
                        }
                    }
                }
            ],
            "IsDefault": false
        }
    ]
}
(END)
```

3. Template

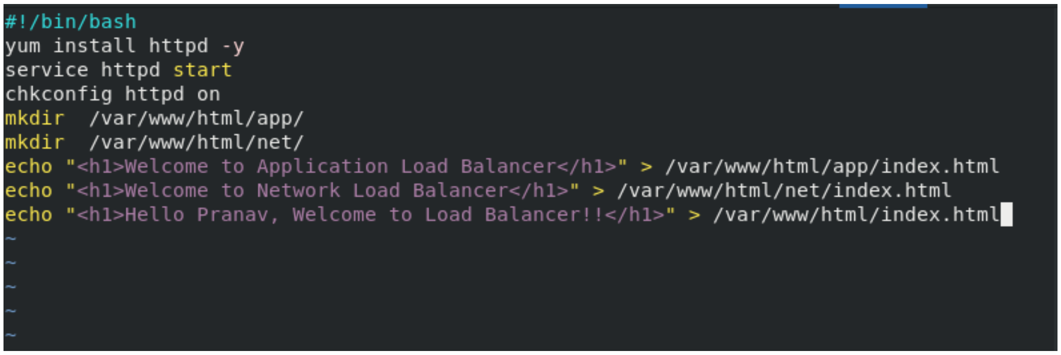
Code:-

```
x=1

while [ $x -ne 0 ]

do

   echo "-----------------------------"

   echo "**** Launch-Template ******"

   echo 1. Create-launch-template

   echo 2. Describe-launch-template

   echo 3. Delete-launch-template

   echo "-----------------------------"

   echo "Enter the choice from above:-"

   read ch

   case "$ch" in

1)echo "Enter Launch-Template Name: "

read nam

echo "Enter Version-description: "

read ver

echo "Enter Region : "

read reg

echo "Enter ImageId (AMI): "

read ami
```

```
echo "Enter Instance Type: "

read typ

echo "Enter Key Name: "

read key

echo "Enter Security Group Id: "

read sg

aws ec2 create-launch-template \
        --launch-template-name $nam \
        --version-description $ver \
        --region $reg \
        --launch-template-data
'{"ImageId":"'$ami'","InstanceType":"'$typ'","KeyName":"'$key'","UserData":"IyEvYmluL
2Jhc2ggIAp5dW0gaW5zdGFsbCBodHRwZCAteSAgCnNlcnZpY2UgaHR0cGQgc3RhcnQgIA
pjaGtjb25maWcgaHR0cGQgb24KbWtkaXIgIC92YXIvd3d3L2h0bWwvYXBwLwpta2RpciAg
L3Zhci93d3cvaHRtbC9uZXQvCmVjaG8gIjxoMT5XZWxjb21lIHRvIEFwcGxpY2F0aW9uIExv
YWQgQmFsYW5jZXI8L2gxPiIgPiAvdmFyL3d3dy9odG1sL2FwcC9pbmRleC5odG1sICAKZW
NobyAiPGgxPldlbGNvbWUgdG8gTmV0d29yayBMb2FkIEJhbGFuY2VyPC9oMT4iID4gL3Zh
ci93d3cvaHRtbC9uZXQvaW5kZXguaHRtbAplY2hvICI8aDE+SGVsbG8gUHJhbmF2LCBXZW
xjb21lIHRvIExvYWQgQmFsYW5jZXIhITwvaDE+IiA+IC92YXIvd3d3L2h0bWwvaW5kZXguaH
RtbCAK","SecurityGroupIds":["'$sg'"]}'

echo " Create-launch-template Sucessfully";;

2) echo "Enter launch-template-id "

read id

aws ec2 describe-launch-template-versions --launch-template-id $id

echo " Describe-launch-template Sucessfully";;

3) echo "Enter launch-template-id "
```

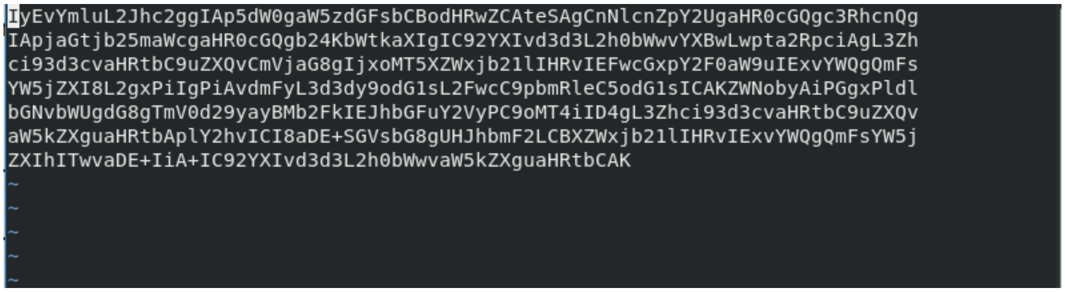
```
read id

echo "Enter Version"

read ver

aws ec2 delete-launch-template-versions --launch-template-id $id --versions $ver

 echo " Delete-launch-template Sucessfully";;


        *)echo "Invalid choice."

         esac

         echo "----------------------------------"

         echo "Enter 1 for continue and 0 for exit."

         read x
done

clear
```

User Data



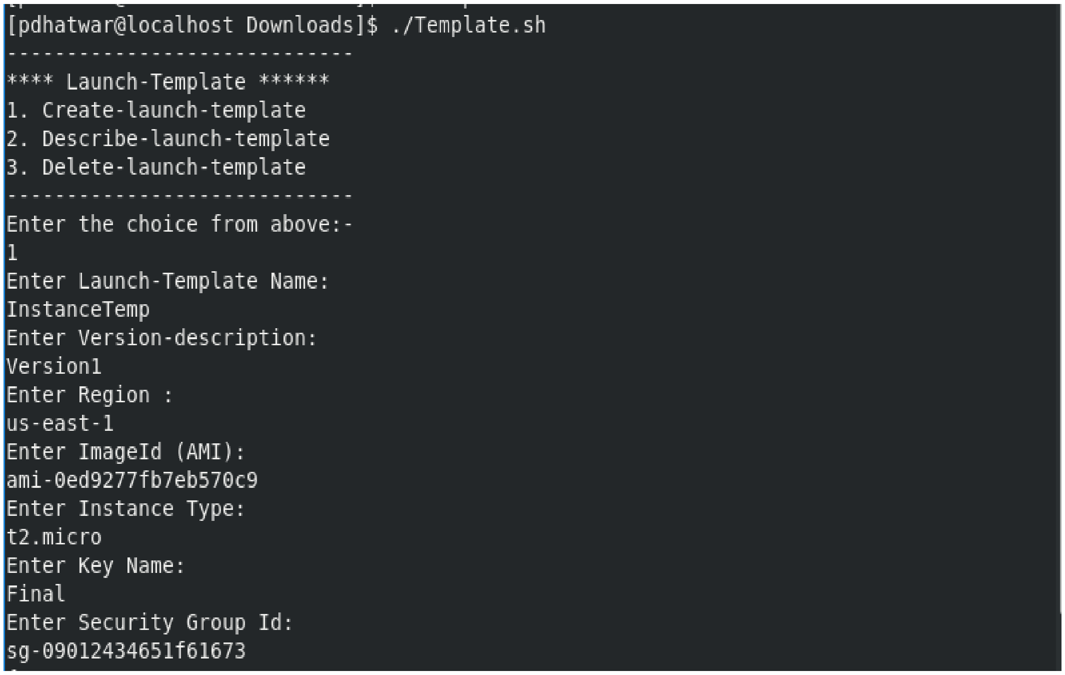
```
#!/bin/bash
yum install httpd -y
service httpd start
chkconfig httpd on
mkdir  /var/www/html/app/
mkdir  /var/www/html/net/
echo "<h1>Welcome to Application Load Balancer</h1>" > /var/www/html/app/index.html
echo "<h1>Welcome to Network Load Balancer</h1>" > /var/www/html/net/index.html
echo "<h1>Hello Pranav, Welcome to Load Balancer!!</h1>" > /var/www/html/index.html
~
~
~
~
~
```

Encoded User Data in Base64 Format

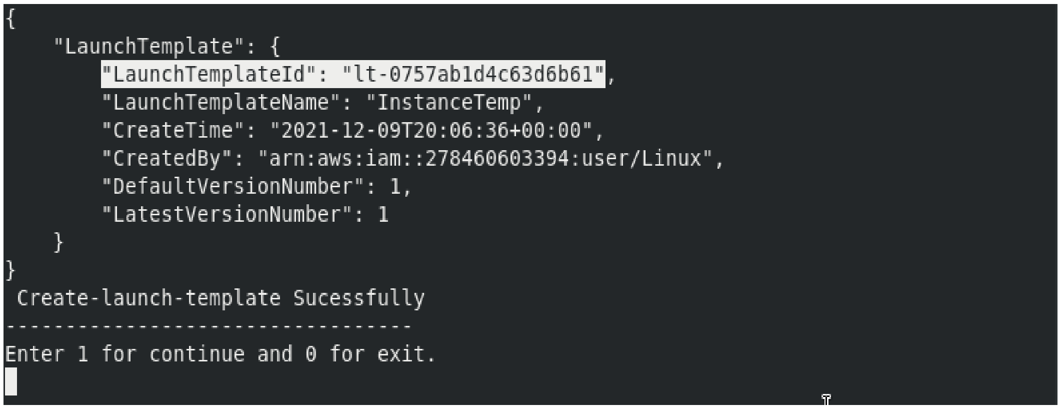


```
IyEvYmluL2Jhc2ggIAp5dW0gaW5zdGFsbCBodHRwZCAteSAgCnNlcnZpY2UgaHR0cGQgc3RhcnQg
IApjaGtjb25maWcgaHR0cGQgb24KbWtkaXIgIC92YXIvd3d3L2h0bWwvYXBwLwpta2RpciAgL3Zh
ci93d3cvaHRtbC9uZXQvCmVjaG8gIjxoMT5XZWxjb21lIHRvIEFwcGxpY2F0aW9uIExvYWQgQmFs
YW5jZXI8L2gxPiIgPiAvdmFyL3d3dy9odG1sL2FwcC9pbmRleC5odG1sICAKZWNobyAiPGgxPldl
bGNvbWUgdG8gTmV0d29yayBMb2FkIEJhbGFuY2VyPC9oMT4iID4gL3Zhci93d3cvaHRtbC9uZXQv
aW5kZXguaHRtbApleY2hvICI8aDE+SGVsbG8gUHJhbmF2LCBXZWxjb21lIHRvIExvYWQgQmFsYW5j
ZXIhITwvaDE+IiA+IC92YXIvd3d3L2h0bWwvaW5kZXguaHRtbCAK
~
~
~
~
~
```

Creating Launch Template



```
[pdhatwar@localhost Downloads]$ ./Template.sh
-----------------------------
**** Launch-Template ******
1. Create-launch-template
2. Describe-launch-template
3. Delete-launch-template
-----------------------------
Enter the choice from above:-
1
Enter Launch-Template Name:
InstanceTemp
Enter Version-description:
Version1
Enter Region :
us-east-1
Enter ImageId (AMI):
ami-0ed9277fb7eb570c9
Enter Instance Type:
t2.micro
Enter Key Name:
Final
Enter Security Group Id:
sg-09012434651f61673
```



```
{
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-0757ab1d4c63d6b61",
        "LaunchTemplateName": "InstanceTemp",
        "CreateTime": "2021-12-09T20:06:36+00:00",
        "CreatedBy": "arn:aws:iam::278460603394:user/Linux",
        "DefaultVersionNumber": 1,
        "LatestVersionNumber": 1
    }
}
 Create-launch-template Sucessfully
---------------------------------
Enter 1 for continue and 0 for exit.
```

4. Auto-Scaling

Code:-

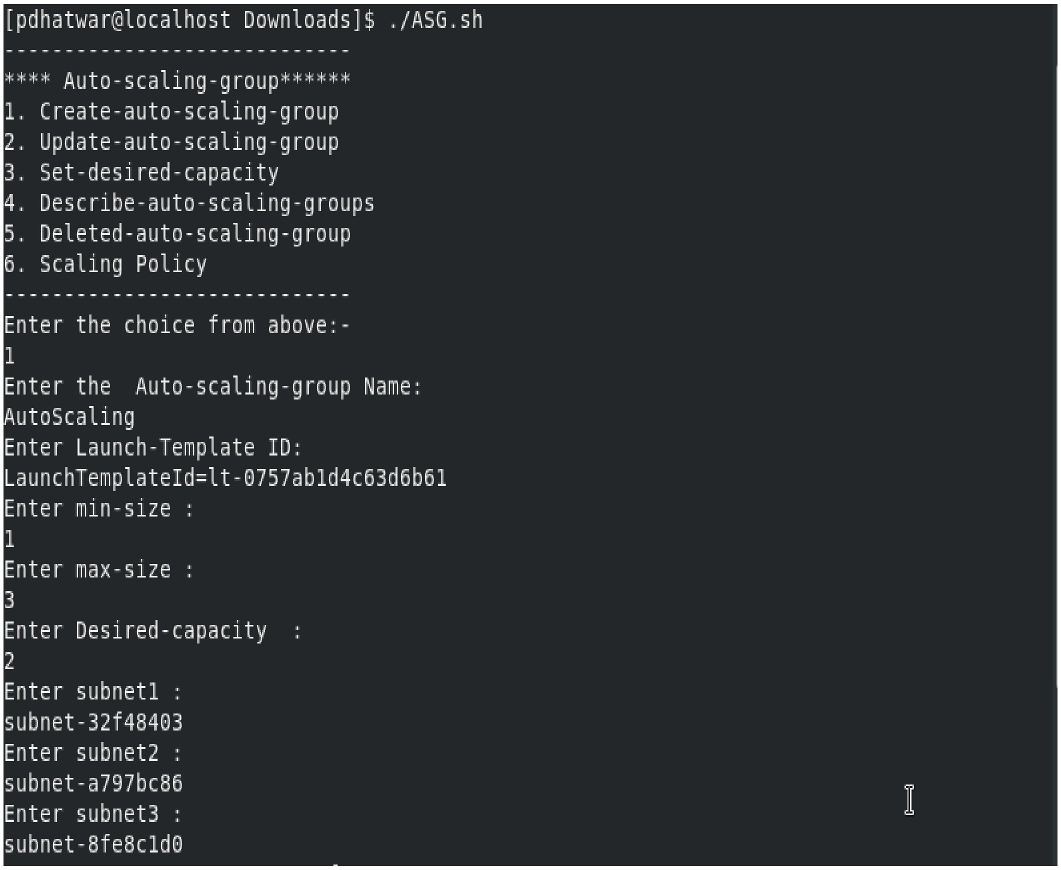
```
x=1

while [ $x -ne 0 ]

do
   echo "-----------------------------"

   echo "**** Auto-scaling-group******"

   echo 1. Create-auto-scaling-group

   echo 2. Update-auto-scaling-group

   echo 3. Set-desired-capacity

   echo 4. Describe-auto-scaling-groups

   echo 5. Deleted-auto-scaling-group

   echo 6. Scaling Policy

   echo "-----------------------------"

   echo "Enter the choice from above:-"

   read ch

   case "$ch" in

 1)echo "Enter the  Auto-scaling-group Name: "

read nam

echo "Enter Launch-Template ID: "

read temp
```

```
echo "Enter min-size : "
read min
echo "Enter max-size : "
read max
echo "Enter Desired-capacity  : "
read desire
echo "Enter subnet1 : "
read subnet1
echo "Enter subnet2 : "
read subnet2
echo "Enter subnet3 : "
read  subnet3
echo "Enter Amazon Resource Name for TG "
read tarn
echo "Enter  Health-check-type  : "
read typ
echo "Enter  Health-check-grace-period: "
read gp
aws autoscaling create-auto-scaling-group --auto-scaling-group-name $nam
--launch-template $temp --min-size $min --max-size $max --vpc-zone-identifier
"$subnet1,$subnet2,$subnet3" --desired-capacity  $desire --target-group-arns $tarn
--health-check-type  $typ --health-check-grace-period $gp\
echo " You have Create-auto-scaling-group Successfully " ;;

2)echo "Enter the  Auto-scaling-group Name: "

read nam

echo "Enter Launch-Template ID: "

read temp

echo "Enter min-size : "

read min

echo "Enter max-size : "

read max

echo "Enter Desired-capacity  : "

read desire

echo "Enter subnet1 : "

read subnet1

echo "Enter subnet2 : "

read subnet2

echo "Enter subnet2 : "

read  subnet3

echo "Enter  Health-check-type  : "

read typ
```

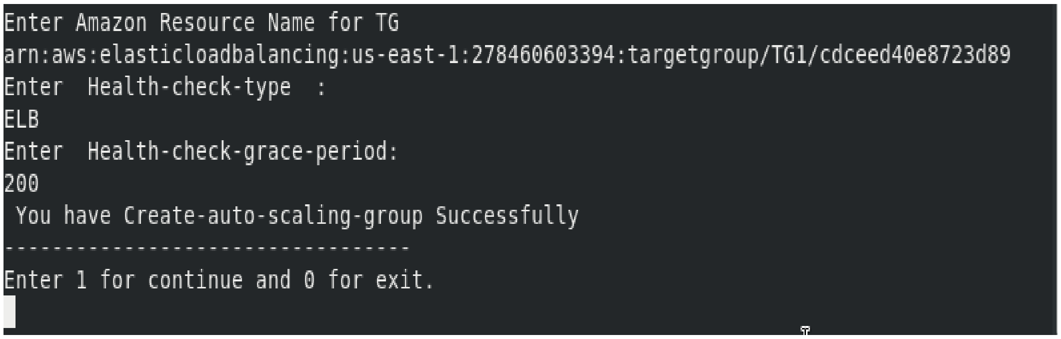
```
echo "Enter  Health-check-grace-period: "

read gp

aws autoscaling update-auto-scaling-group --auto-scaling-group-name $nam
--launch-template $temp --min-size $min --max-size $max --vpc-zone-identifier
"$subnet1,$subnet2,$subnet3" --availability-zones "us-east-1a""us-east-1b"
"us-east-1c" --desired-capacity  $desire --health-check-type  $typ
--health-check-grace-period $gp\

echo " You have Update-auto-scaling-group Successfully " ;;

3) echo "Enter Autoscaling Group Name  "

read nam

echo "Enter Desired-capacity "

read count

aws autoscaling set-desired-capacity --auto-scaling-group-name $nam
--desired-capacity $count \

echo " Set-desired-capacity Sucessfully";;

4) echo "Enter Autoscaling Group Name  "

read nam

aws autoscaling describe-auto-scaling-groups --auto-scaling-group-name $nam

echo " Describe-auto-scaling-groups Sucessfully";;

5) echo "Enter Autoscaling Group Name  "

read nam

aws autoscaling delete-auto-scaling-group --auto-scaling-group-name $nam

echo " Deleted-auto-scaling-group Sucessfully";;
```

```
6)echo "Enter the  Auto-scaling-group Name: "

read nam

echo "Enter Policy-name: "

read poly

echo "Enter Policy-type: "

read ptyp

aws autoscaling put-scaling-policy --policy-name $poly --auto-scaling-group-name $nam
--policy-type $ptyp\

echo "You have put-scaling-policy Sucessfully" ;;

         *)echo "Invalid choice."

          esac

          echo "---------------------------------"

          echo "Enter 1 for continue and 0 for exit."

          read x

done

clear
```

Creating Auto-Scaling Group for Target Group(TG1)

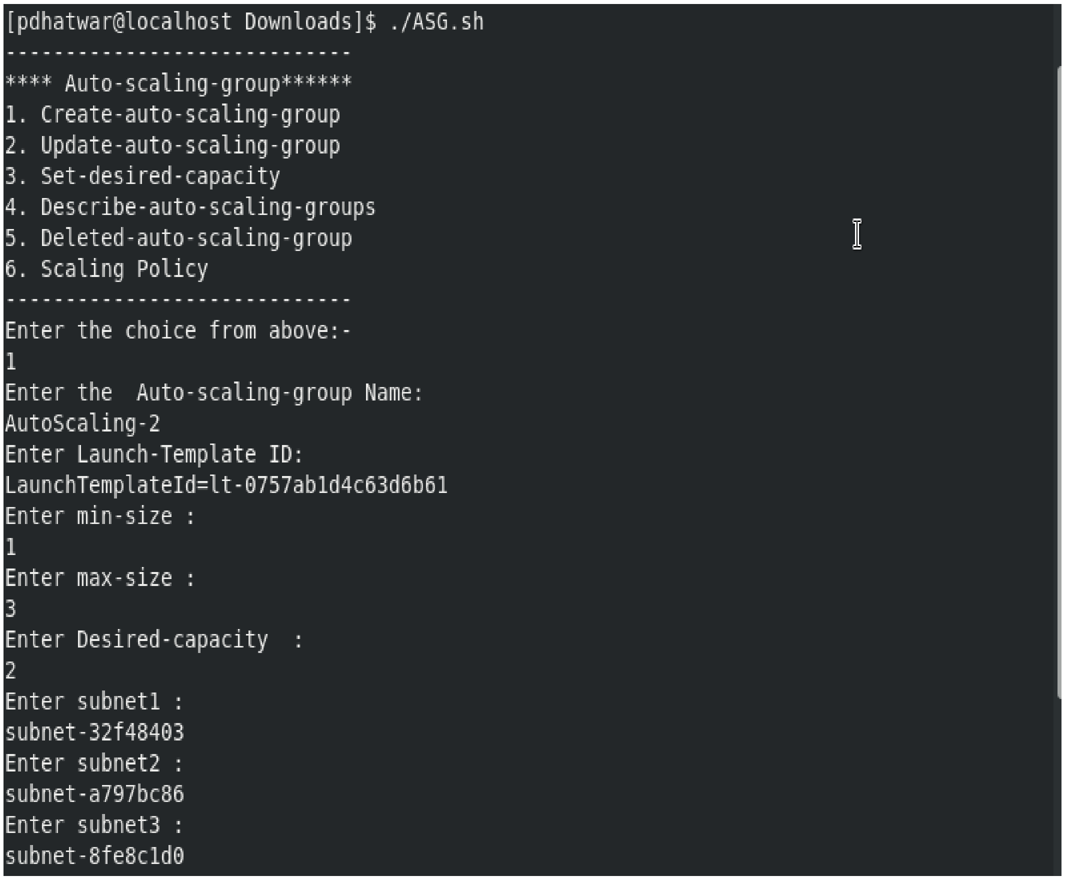


```
[pdhatwar@localhost Downloads]$ ./ASG.sh
-----------------------------
**** Auto-scaling-group******
1. Create-auto-scaling-group
2. Update-auto-scaling-group
3. Set-desired-capacity
4. Describe-auto-scaling-groups
5. Deleted-auto-scaling-group
6. Scaling Policy
-----------------------------
Enter the choice from above:-
1
Enter the  Auto-scaling-group Name:
AutoScaling
Enter Launch-Template ID:
LaunchTemplateId=lt-0757ab1d4c63d6b61
Enter min-size :
1
Enter max-size :
3
Enter Desired-capacity  :
2
Enter subnet1 :
subnet-32f48403
Enter subnet2 :
subnet-a797bc86
Enter subnet3 :
subnet-8fe8c1d0
```

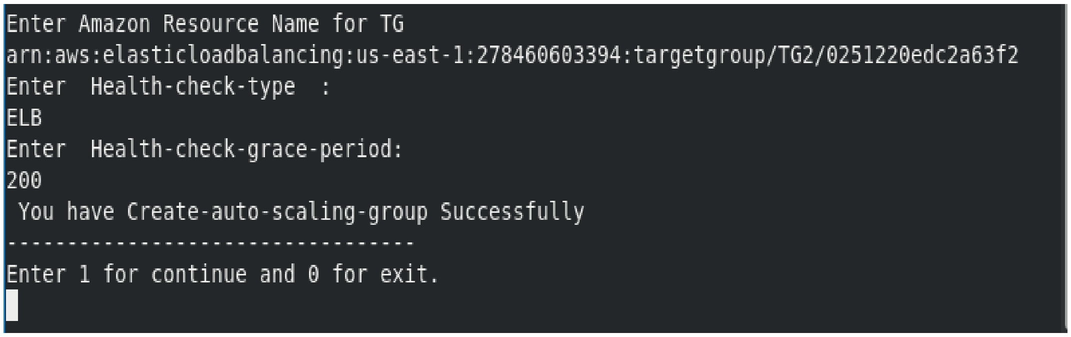


```
Enter Amazon Resource Name for TG
arn:aws:elasticloadbalancing:us-east-1:278460603394:targetgroup/TG1/cdceed40e8723d89
Enter  Health-check-type  :
ELB
Enter  Health-check-grace-period:
200
 You have Create-auto-scaling-group Successfully
----------------------------------
Enter 1 for continue and 0 for exit.
```

Creating Auto-Scaling Group for Target Group(TG2)



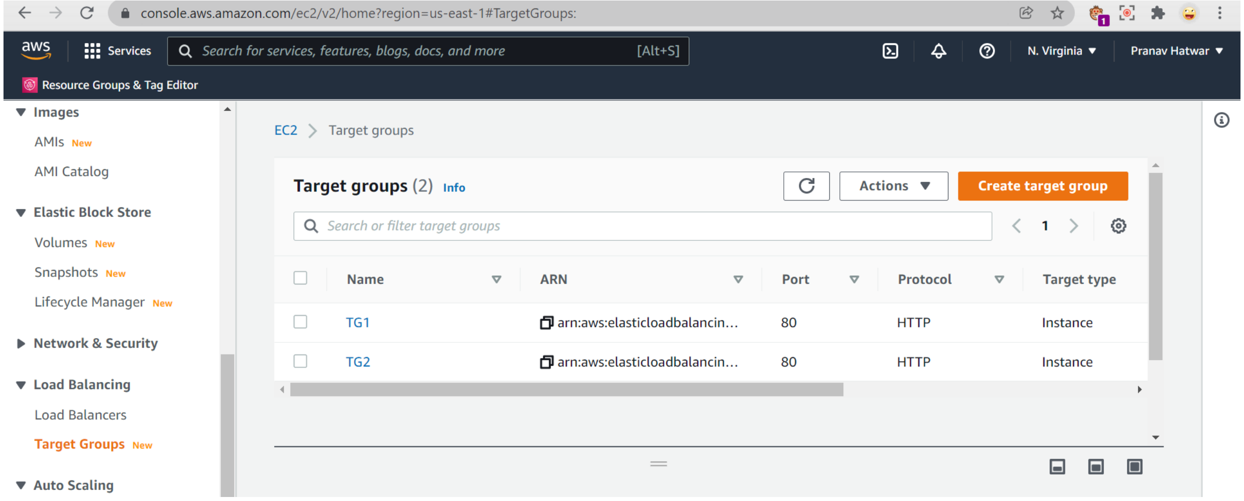
```
[pdhatwar@localhost Downloads]$ ./ASG.sh
-----------------------------
**** Auto-scaling-group******
1. Create-auto-scaling-group
2. Update-auto-scaling-group
3. Set-desired-capacity
4. Describe-auto-scaling-groups
5. Deleted-auto-scaling-group
6. Scaling Policy
-----------------------------
Enter the choice from above:-
1
Enter the  Auto-scaling-group Name:
AutoScaling-2
Enter Launch-Template ID:
LaunchTemplateId=lt-0757ab1d4c63d6b61
Enter min-size :
1
Enter max-size :
3
Enter Desired-capacity  :
2
Enter subnet1 :
subnet-32f48403
Enter subnet2 :
subnet-a797bc86
Enter subnet3 :
subnet-8fe8c1d0
```



```
Enter Amazon Resource Name for TG
arn:aws:elasticloadbalancing:us-east-1:278460603394:targetgroup/TG2/0251220edc2a63f2
Enter  Health-check-type  :
ELB
Enter  Health-check-grace-period:
200
 You have Create-auto-scaling-group Successfully
----------------------------------
Enter 1 for continue and 0 for exit.
```

5. Management Console(Target Groups, LoadBalancer, Template, ASG)

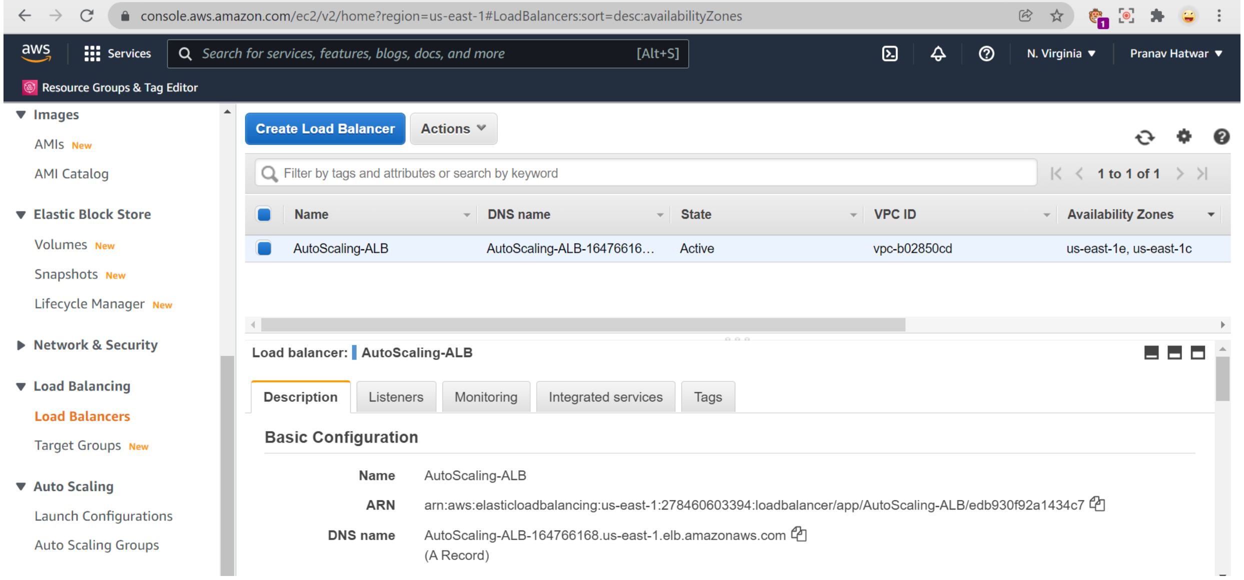
- ## Target Groups



Target groups (2) Info

	Name	ARN	Port	Protocol	Target type
□	TG1	arn:aws:elasticloadbalancin…	80	HTTP	Instance
□	TG2	arn:aws:elasticloadbalancin…	80	HTTP	Instance

- ## LoadBalancer



	Name	DNS name	State	VPC ID	Availability Zones
☑	AutoScaling-ALB	AutoScaling-ALB-16476616…	Active	vpc-b02850cd	us-east-1e, us-east-1c

Load balancer: AutoScaling-ALB

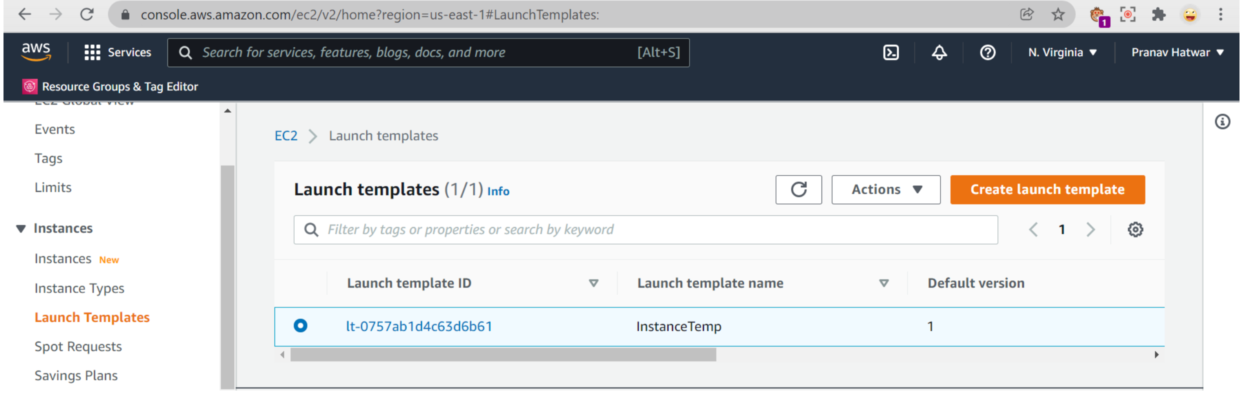
Basic Configuration

Name AutoScaling-ALB

ARN arn:aws:elasticloadbalancing:us-east-1:278460603394:loadbalancer/app/AutoScaling-ALB/edb930f92a1434c7

DNS name AutoScaling-ALB-164766168.us-east-1.elb.amazonaws.com
(A Record)

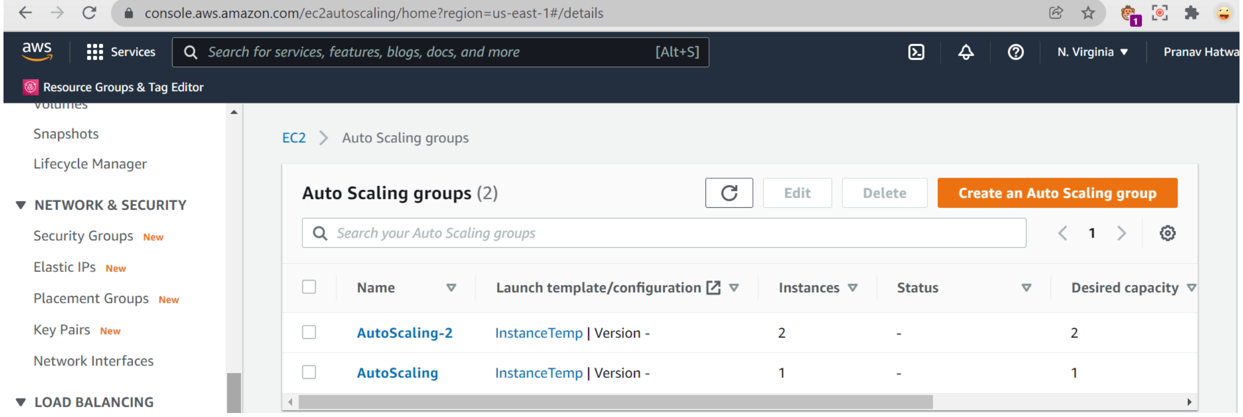
• Launch Template


console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchTemplates:
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Pranav Hatwar
Resource Groups & Tag Editor
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
EC2 > Launch templates
Actions
Create launch template
Filter by tags or properties or search by keyword
1

Launch templates (1/1) Info

	Launch template ID	Launch template name	Default version
●	lt-0757ab1d4c63d6b61	InstanceTemp	1

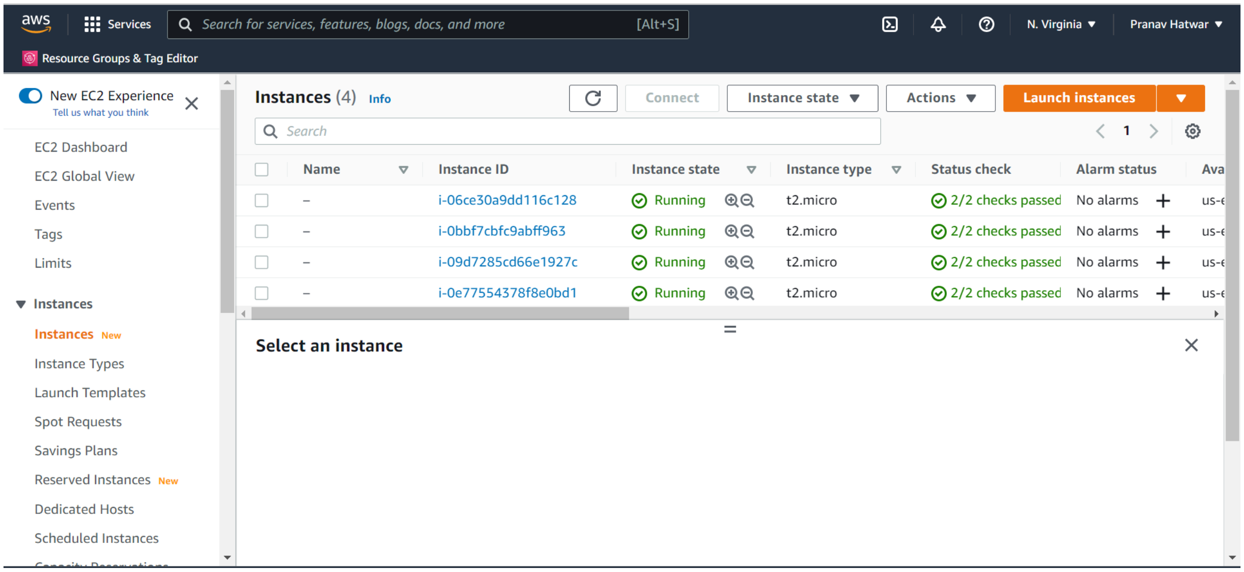
• Auto Scaling Group


console.aws.amazon.com/ec2autoscaling/home?region=us-east-1#/details
aws
Services
Search for services, features, blogs, docs, and more
[Alt+S]
N. Virginia
Pranav Hatwa
Resource Groups & Tag Editor
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
LOAD BALANCING
EC2 > Auto Scaling groups
Edit
Delete
Create an Auto Scaling group
Search your Auto Scaling groups
1

Auto Scaling groups (2)

□	Name	Launch template/configuration	Instances	Status	Desired capacity
□	AutoScaling-2	InstanceTemp | Version -	2	-	2
□	AutoScaling	InstanceTemp | Version -	1	-	1

• Instance

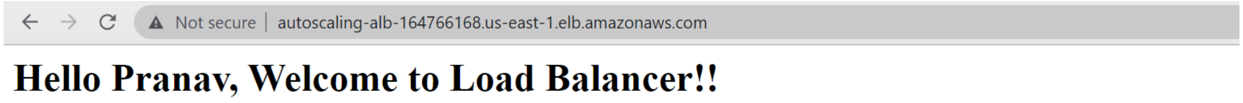


Instances (4) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	–	i-06ce30a9dd116c128	Running	t2.micro	2/2 checks passed	No alarms +	us-e
□	–	i-0bbf7cbfc9abff963	Running	t2.micro	2/2 checks passed	No alarms +	us-e
□	–	i-09d7285cd66e1927c	Running	t2.micro	2/2 checks passed	No alarms +	us-e
□	–	i-0e77554378f8e0bd1	Running	t2.micro	2/2 checks passed	No alarms +	us-e

Output:-

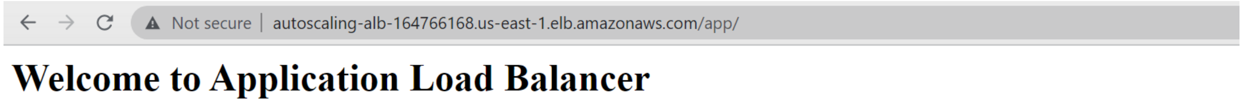
Home Page



Not secure | autoscaling-alb-164766168.us-east-1.elb.amazonaws.com

Hello Pranav, Welcome to Load Balancer!!

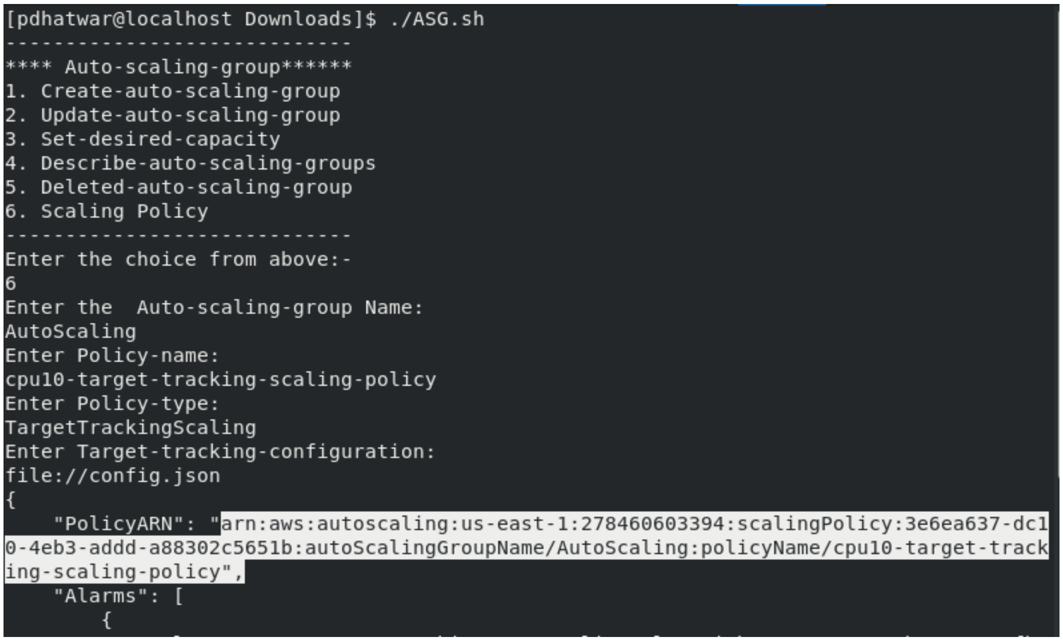
App Component



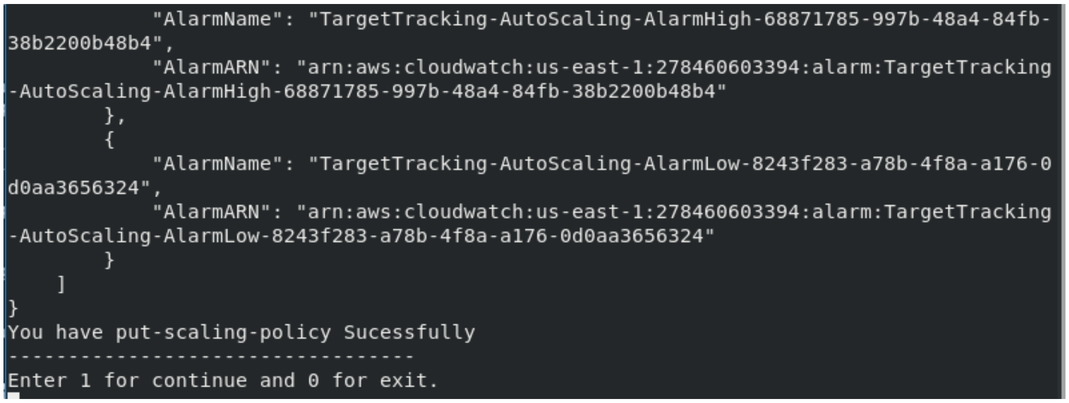
Not secure | autoscaling-alb-164766168.us-east-1.elb.amazonaws.com/app/

Welcome to Application Load Balancer

6. Adding Scaling Policy To Auto-Scaling Group

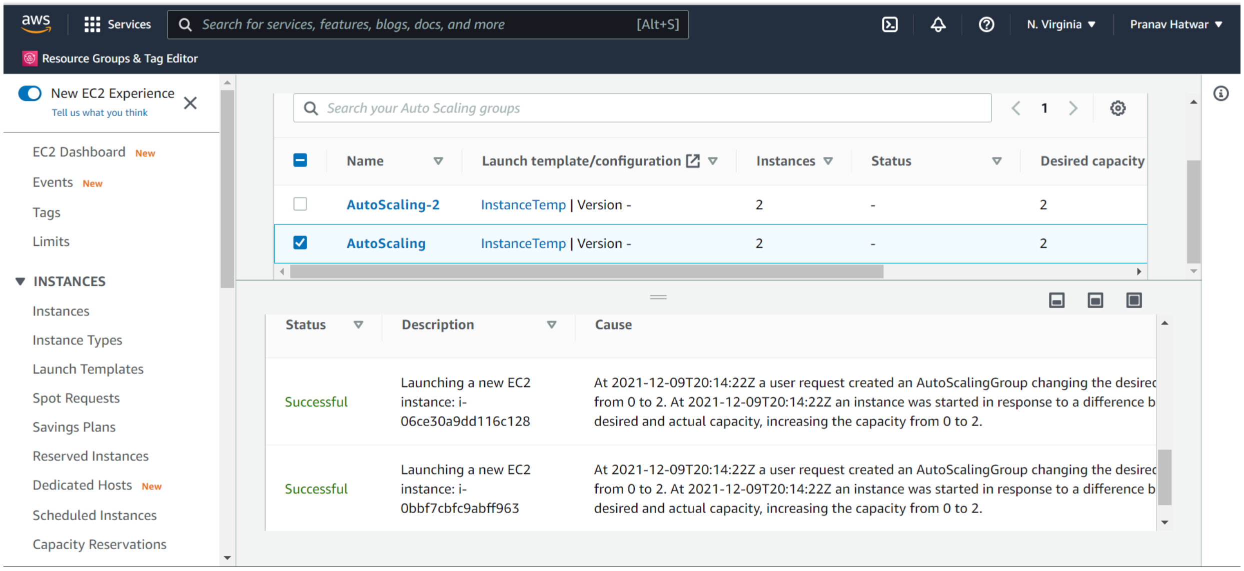


```
[pdhatwar@localhost Downloads]$ ./ASG.sh
-----------------------------
**** Auto-scaling-group******
1. Create-auto-scaling-group
2. Update-auto-scaling-group
3. Set-desired-capacity
4. Describe-auto-scaling-groups
5. Deleted-auto-scaling-group
6. Scaling Policy
-----------------------------
Enter the choice from above:-
6
Enter the  Auto-scaling-group Name:
AutoScaling
Enter Policy-name:
cpu10-target-tracking-scaling-policy
Enter Policy-type:
TargetTrackingScaling
Enter Target-tracking-configuration:
file://config.json
{
    "PolicyARN": "arn:aws:autoscaling:us-east-1:278460603394:scalingPolicy:3e6ea637-dc1
0-4eb3-addd-a88302c5651b:autoScalingGroupName/AutoScaling:policyName/cpu10-target-track
ing-scaling-policy",
    "Alarms": [
        {
```



```
            "AlarmName": "TargetTracking-AutoScaling-AlarmHigh-68871785-997b-48a4-84fb-
38b2200b48b4",
            "AlarmARN": "arn:aws:cloudwatch:us-east-1:278460603394:alarm:TargetTracking
-AutoScaling-AlarmHigh-68871785-997b-48a4-84fb-38b2200b48b4"
        },
        {
            "AlarmName": "TargetTracking-AutoScaling-AlarmLow-8243f283-a78b-4f8a-a176-0
d0aa3656324",
            "AlarmARN": "arn:aws:cloudwatch:us-east-1:278460603394:alarm:TargetTracking
-AutoScaling-AlarmLow-8243f283-a78b-4f8a-a176-0d0aa3656324"
        }
    ]
}
You have put-scaling-policy Sucessfully
-----------------------------------
Enter 1 for continue and 0 for exit.
```

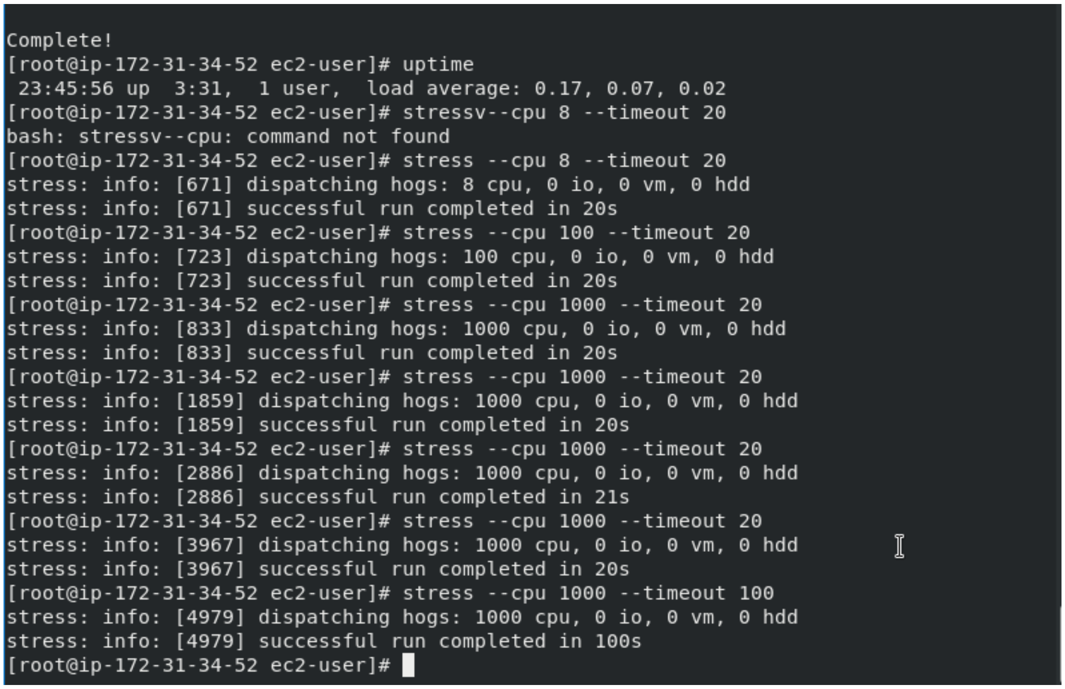
As Desired Capacity is 2 and Max Size is 3. So, to Check whether Autoscaling is working or not let’s Stress the Instance.



☐	Name	Launch template/configuration	Instances	Status	Desired capacity
☐	AutoScaling-2	InstanceTemp | Version -	2	-	2
☑	AutoScaling	InstanceTemp | Version -	2	-	2

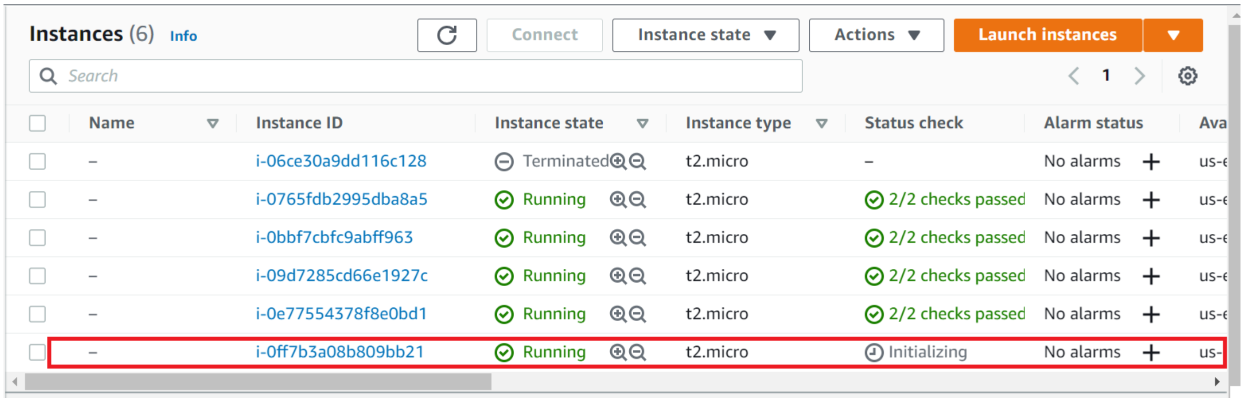
Status	Description	Cause
Successful	Launching a new EC2 instance: i-06ce30a9dd116c128	At 2021-12-09T20:14:22Z a user request created an AutoScalingGroup changing the desirec from 0 to 2. At 2021-12-09T20:14:22Z an instance was started in response to a difference b desired and actual capacity, increasing the capacity from 0 to 2.
Successful	Launching a new EC2 instance: i-0bbf7cbfc9abff963	At 2021-12-09T20:14:22Z a user request created an AutoScalingGroup changing the desirec from 0 to 2. At 2021-12-09T20:14:22Z an instance was started in response to a difference b desired and actual capacity, increasing the capacity from 0 to 2.

Giving Stress to Instance



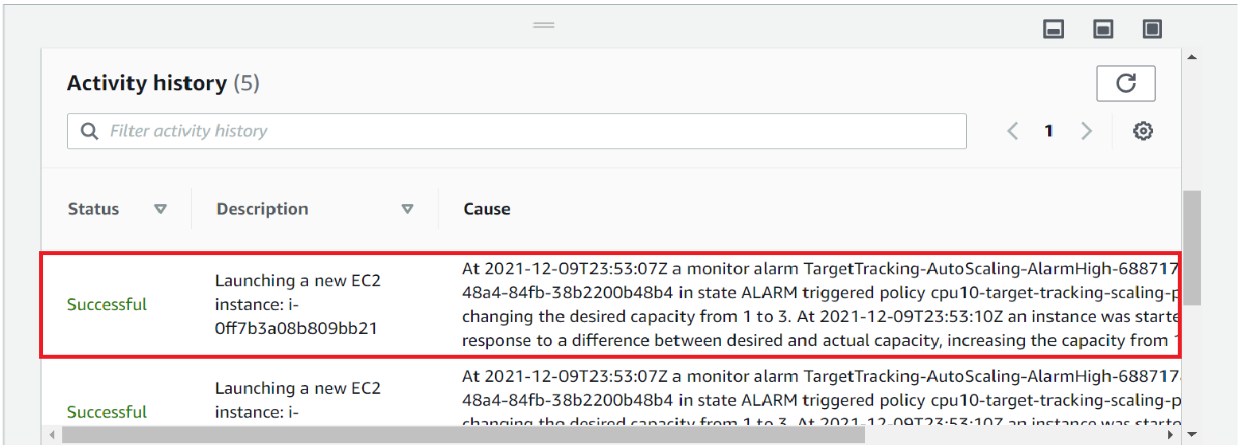
```
Complete!
[root@ip-172-31-34-52 ec2-user]# uptime
 23:45:56 up  3:31,  1 user,  load average: 0.17, 0.07, 0.02
[root@ip-172-31-34-52 ec2-user]# stressv--cpu 8 --timeout 20
bash: stressv--cpu: command not found
[root@ip-172-31-34-52 ec2-user]# stress --cpu 8 --timeout 20
stress: info: [671] dispatching hogs: 8 cpu, 0 io, 0 vm, 0 hdd
stress: info: [671] successful run completed in 20s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 100 --timeout 20
stress: info: [723] dispatching hogs: 100 cpu, 0 io, 0 vm, 0 hdd
stress: info: [723] successful run completed in 20s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 1000 --timeout 20
stress: info: [833] dispatching hogs: 1000 cpu, 0 io, 0 vm, 0 hdd
stress: info: [833] successful run completed in 20s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 1000 --timeout 20
stress: info: [1859] dispatching hogs: 1000 cpu, 0 io, 0 vm, 0 hdd
stress: info: [1859] successful run completed in 20s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 1000 --timeout 20
stress: info: [2886] dispatching hogs: 1000 cpu, 0 io, 0 vm, 0 hdd
stress: info: [2886] successful run completed in 21s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 1000 --timeout 20
stress: info: [3967] dispatching hogs: 1000 cpu, 0 io, 0 vm, 0 hdd
stress: info: [3967] successful run completed in 20s
[root@ip-172-31-34-52 ec2-user]# stress --cpu 1000 --timeout 100
stress: info: [4979] dispatching hogs: 1000 cpu, 0 io, 0 vm, 0 hdd
stress: info: [4979] successful run completed in 100s
[root@ip-172-31-34-52 ec2-user]#
```

New Instance started Launching



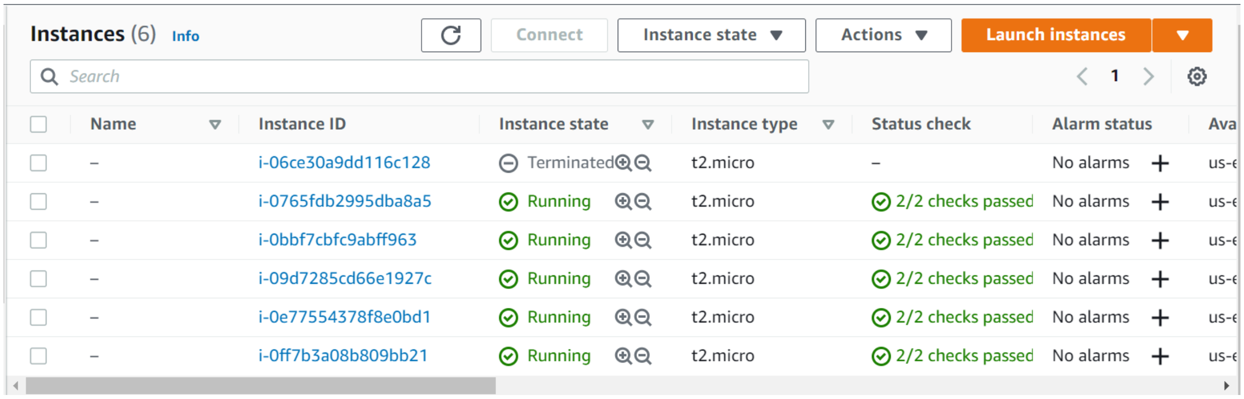
□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	–	i-06ce30a9dd116c128	⊖ Terminated	t2.micro	–	No alarms +	us-e
□	–	i-0765fdb2995dba8a5	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0bbf7cbfc9abff963	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-09d7285cd66e1927c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0e77554378f8e0bd1	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0ff7b3a08b809bb21	✓ Running	t2.micro	Initializing	No alarms +	us-

Activity History



Status	Description	Cause
Successful	Launching a new EC2 instance: i-0ff7b3a08b809bb21	At 2021-12-09T23:53:07Z a monitor alarm TargetTracking-AutoScaling-AlarmHigh-688717 48a4-84fb-38b2200b48b4 in state ALARM triggered policy cpu10-target-tracking-scaling-p changing the desired capacity from 1 to 3. At 2021-12-09T23:53:10Z an instance was starte response to a difference between desired and actual capacity, increasing the capacity from 1
Successful	Launching a new EC2 instance: i-	At 2021-12-09T23:53:07Z a monitor alarm TargetTracking-AutoScaling-AlarmHigh-688717 48a4-84fb-38b2200b48b4 in state ALARM triggered policy cpu10-target-tracking-scaling-p changing the desired capacity from 1 to 3. At 2021-12-09T23:53:10Z an instance was starte

Finally, Instance is Launched and Ready to Serve Traffic



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
□	–	i-06ce30a9dd116c128	⊖ Terminated	t2.micro	–	No alarms +	us-e
□	–	i-0765fdb2995dba8a5	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0bbf7cbfc9abff963	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-09d7285cd66e1927c	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0e77554378f8e0bd1	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e
□	–	i-0ff7b3a08b809bb21	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-e

****