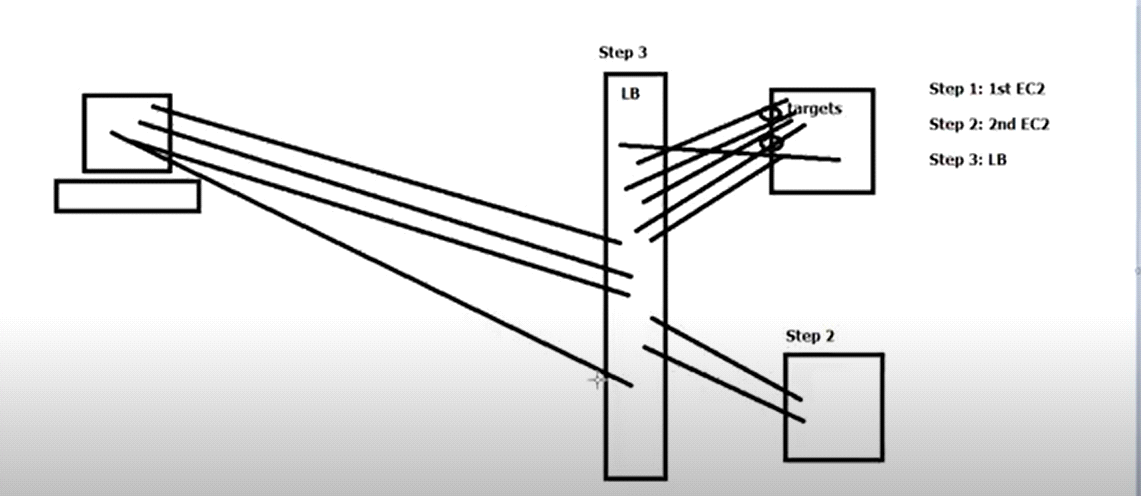
**Building Monitoring for application**

* **Creating Load Balancer and attaching EC2 machines**



A load balancer accepts incoming traffic from clients and routes requests to EC2 instances

( Targets).

The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets.

When the load balancer detects an unhealthy target, it stops routing traffic to that target. It then resumes routing traffic to that target when it detects that the target is healthy again.

Step 1: Create Linux Machine

Launch instance --- Amazon Linux -- No of intances - 1 --- Name Tag- Lin-1 --- Security Group - LinSG09

Description - LinSG

Add Rule

HTTP

Launch

----------

Step 2: COnvert pem to ppk file

Step 3: Access the machine

Step 4: Run the commands to install web package

sudo su

yum update -y

yum install httpd -y

cd /var/www/html

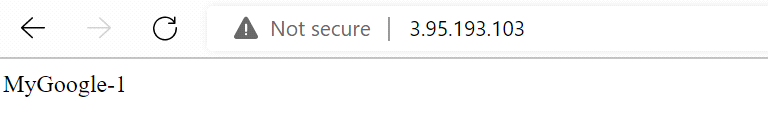
echo "MyGoogle-1" > index.html

ls

service httpd start

chkconfig httpd on

Step 5: Access the webserver by using public\_ip



+++++++++++++++++++++++++++++++++++++++++

Create second ec2 instance

Bootstrap scripts

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks.

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

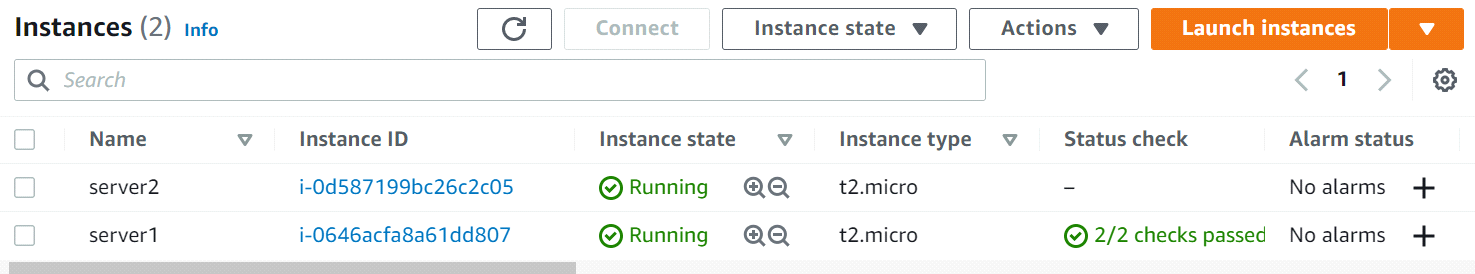
cd /var/www/html

echo "MyGoogle-2" > index.html

service httpd start

chkconfig httpd on

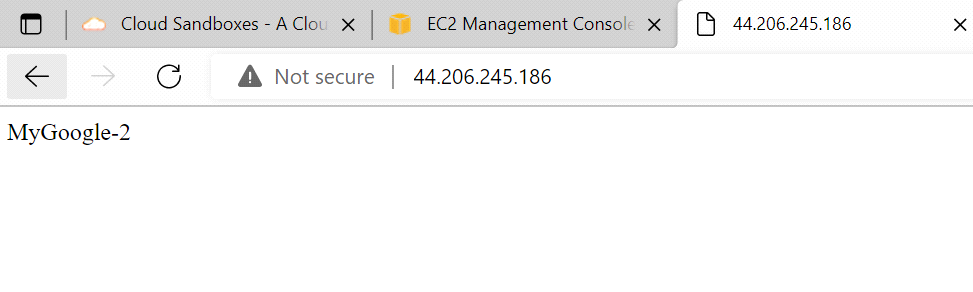
++++++++++++++++++++++++++



Step 6: Launch one more Linux Machine and install Web package

AmazonLinux --

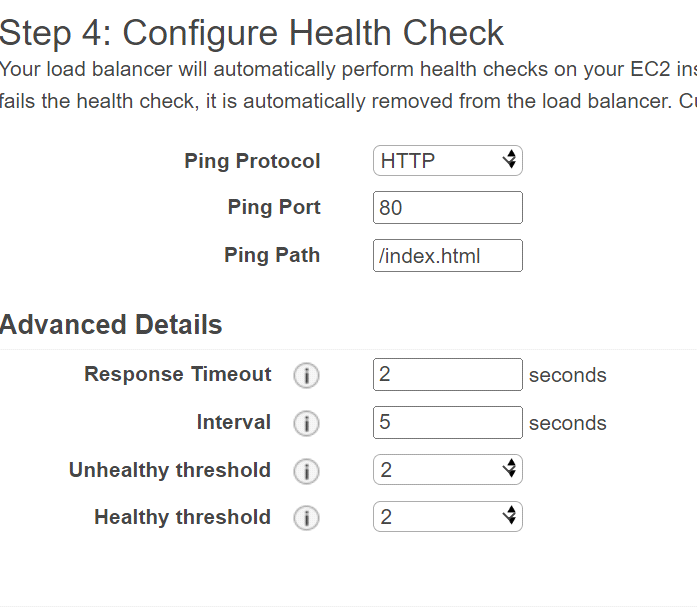
Next -- Add Name Tag -- Step 6: Select existing security group -- Choose existing key pair -- Launch instance.



Step 7: Create load balancers

Select classic load balancer

Load Balancer Name - MyLB --> Next ----> select existing security group ---> NExt -- Step 4:



Response Timeout - 2 Seconds

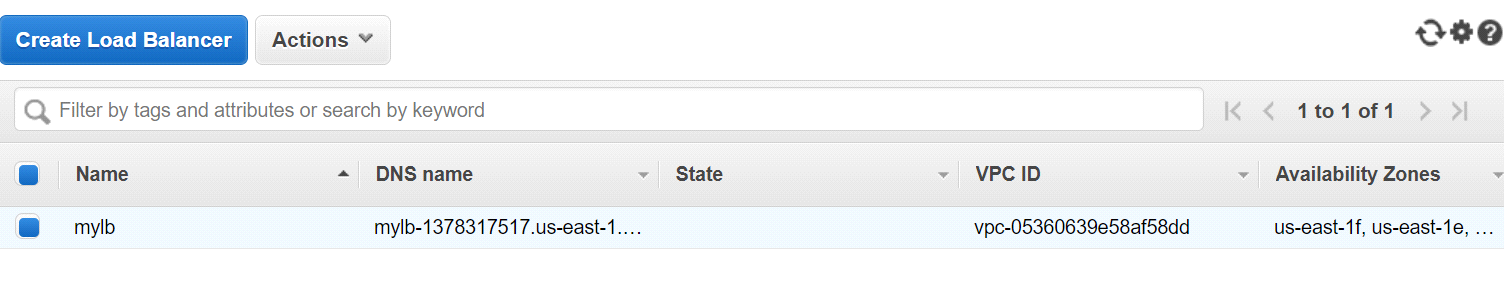
Interval - 5 Seconds

Unhealthy threshold - 2

Healthy threshold - 2

Next -- Attach both the instances

Next -- Next --Create

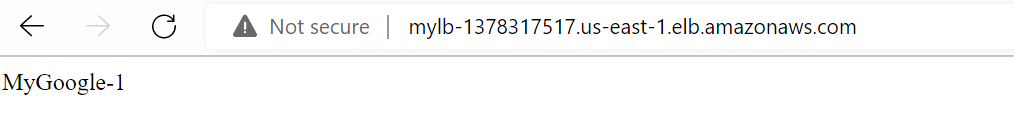


Step 8: Lest verify the 2nd instance manually

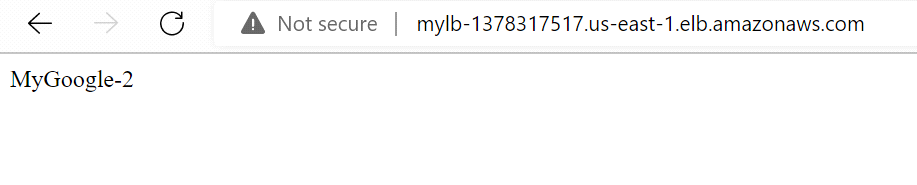
Select public ip and paste in browser

Step 9: Access the load balance by using DNS

and experience the load balancer.



Step 10: If one server is down, it should redirect the traffic to another server.



Auto Scaling

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

Auto Scaling helps you ensure that you have the correct number of EC2 instances available

to handle the load for your application.

You create collections of EC2 instances, called Auto Scaling groups.

You can specify the minimum number of instances and the maximum number of instances in Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.

+++++++++++++++++++++++++

Step 1: Create load balancer.

Step 2: Create Launch configuration

Step 3: Create Topic in SNS ( Simple Notification Service )

Step 4: Create Auto scaling group.

Step 5: Create Alarm in CloudWatch

Step 6: Add Policy in Auto Scaling

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

Step 1: Create load balancer

Select Mumbai location

EC 2 dashboard -- load balancer -- create load balancer

Load balancer Name - SampleLB14

Next -- Create new Security group

Security group name - Sample-Sg14

Description - Sample-Sg14

Lets open two ports SSH and HTTP

Next --> Configure health check

Response timeout -2

Interval -5

Uhealthy threshold - 2

Healthy threshold -2

Next -- Next -- Review and create --> Create---> Close

+++++++++++++++++++++++

Step 2:

Creating launch configuration

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

Under Auto Scaling

Select Launch configurations ---> Create launch configuration

Name: SampleLCG14

AMI - ami-08e0ca9924195beba

Step 2: t2 micro

In advanced Details User data

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

cd /var/www/html

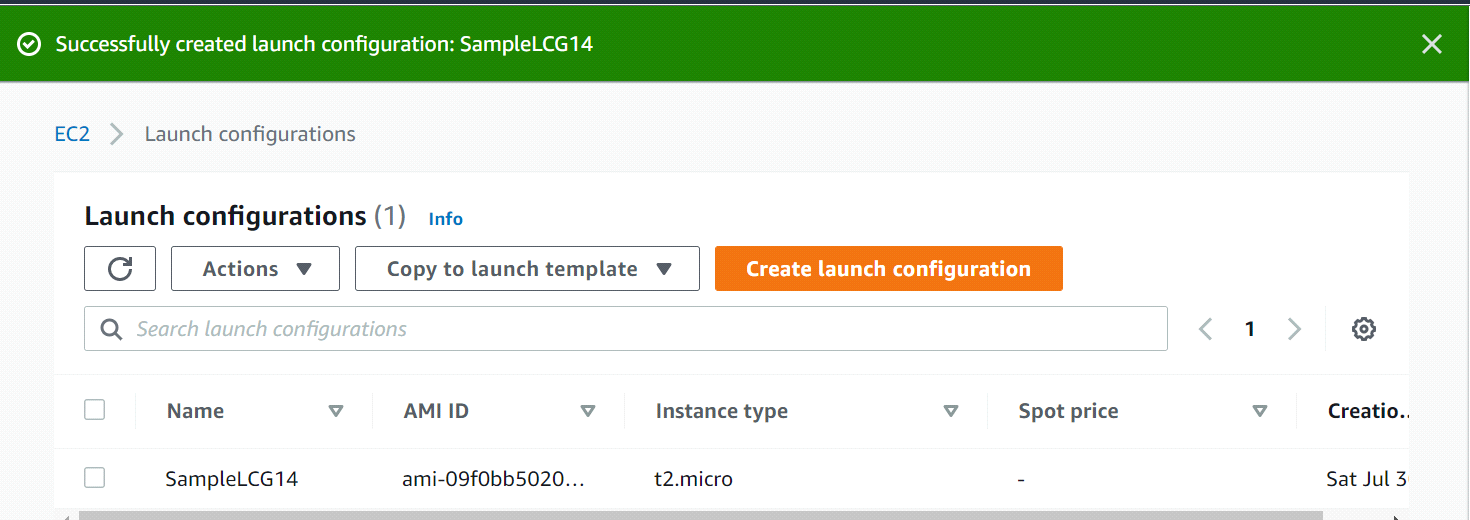
echo "MyGoogle-2" > index.html

service httpd start

chkconfig httpd on

Next --> Select existing security group : Sample-Sg14

Create new key pair -- Key Pair name - SampleKP14 -- download key pair --> create launch configuration.



+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Step 3: Create Topic in SNS

Services -- Application Integration -- Simple Notification Service

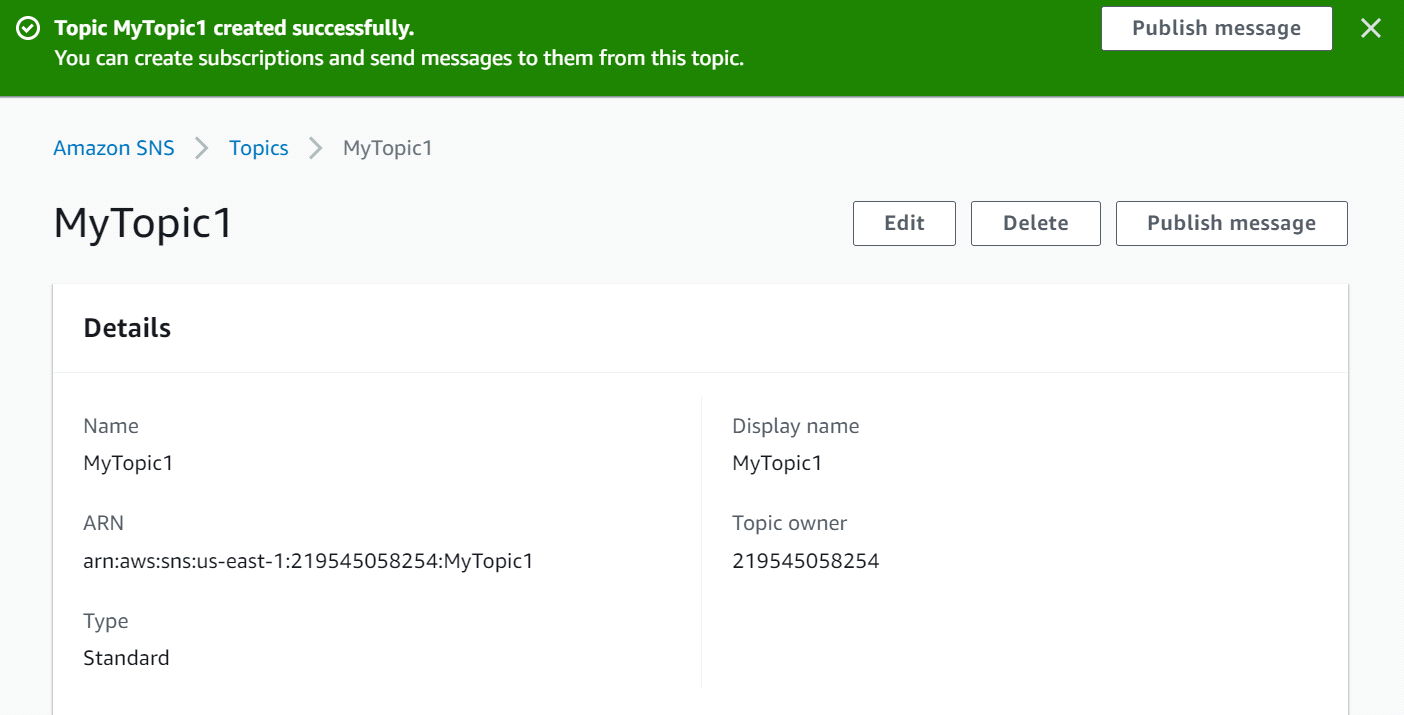
Create Topic

Type - Standard

Name - MyTopic1

Display Name - MyTopic1

Create topic.



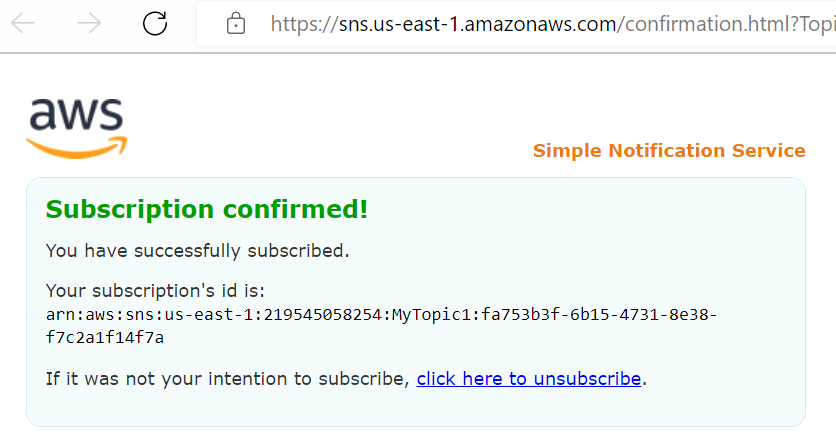
Add subscriptions to the topic

Create Subscription

Protocol - Email

Endpoint - Vikas\_Kolekar@epam.com

Create Subscription.



+++++++++++++++++++++++

Step 4: Create Auto Scaling Group

Select the Launch Configuration ---> Actions --- Create Auto Scaling Group

Step 1: Auto Scaling group name - SampleASG14

Step 2: Subnet -- us-east-1a

Step 3: Attach Existing Load Balancer ---> Choose from Classic Load Balancers -- Select Load Balancers

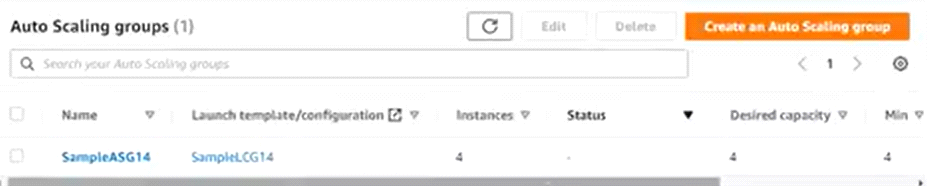
Step 4:Configure Group size ( Take Defaults )

Step 5: Next

Step 6: Next -- Provide Name Tag

Step 7: Review and Create

As the desired capacity is 1 , By this time one EC2 Machine could have been created.





+++++++++++++++++++++++++++++++++++++++

Step 5: Create Alarm in CloudWatch

Services ---- Management & Governance --- CloudWatch

Alarm --- Create Alarm

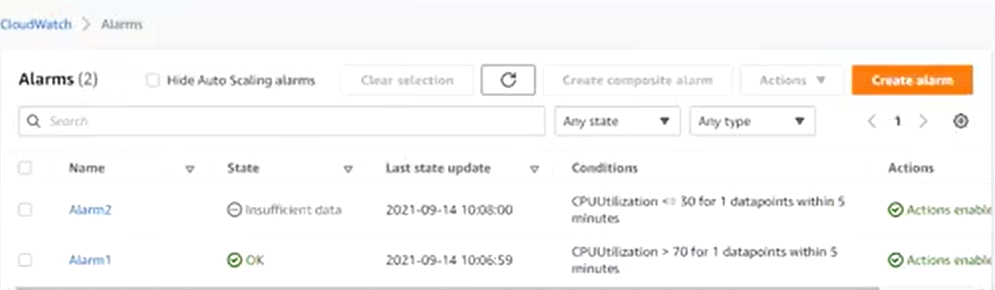
Select Metric ---> EC2 --- By Auto Scaling Group ---- Select Auto Scaling Group Name , Metric Name - CPUUtilization

-- Select Metric ---Conditions --- Static -- Greater than 80 -- Next ---Select existing SNS topic ( MyTopic1 ) -- Next ---

Alarm Name - My\_Alarm1 -- Next -- Create Alarm.

Similarly Create another Alarm for CPU Utilization <30

Name - My\_Alarm2



++++++++++++++++++++++++

Step 6: Add Policy in Auto Scaling

Select AutoScaling groups -- Automatic Scaling Tab --- Add Policy

Policy Type - Simple Scaling

Scaling policy name -- Increase\_Policy

CloudWatch Alarm - My\_alarm1

Take the Action - Add - 1 Unit ( EC2 Machine )

Create

Similary, we need to create another Decrease Policy

Scaling policy name -- Decrease\_Policy

CloudWatch Alarm - My\_alarm2

Take the Action - Remove - 1 Unit ( EC2 Machine )

Create