ELASTIC LOAD BALANCING

(MAPPING ELB with Auto scaling group and setting criteria along with cloud watch)

- Achieve fault tolerance for any application by ensuring scalability, performance, and security
- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, and Lambda functions. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant

  Types of load balancers

  1. Application Load balancer
  2. Network load balancer (which is recently brought up & working under progress)
  3. Classic load balancer

Application Load balancer (HTTP &HTTPS) for web application

- Application Load Balancer is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers. Operating at the individual request level (Layer 7), **Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request**.
- Incoming request will route based on HOSTNAME & PATHNAME (Ex:- www.icicibank.com/creditcard/signup) So it will route only to credit-card which is path name
- It read URL

Network Load balancer (PORT NUMBERS)

- It will route the traffic based on the port numbers
- Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic where extreme performance is required. Operating at the connection level (Layer 4)
- Ex:- www.icicibank.com://8080 -> so this will route only based on port numbers & Highly secured purpose
- Network Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) and is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is also optimized to handle sudden and volatile traffic patterns

Classic Load balancer (Old load balancer)

- It will not read URL
- It will re-direct the request through IP address & port numbers.It operates at both the request level and connection level
- Classic Load Balancer is intended for applications that were built within the EC2-Classic network

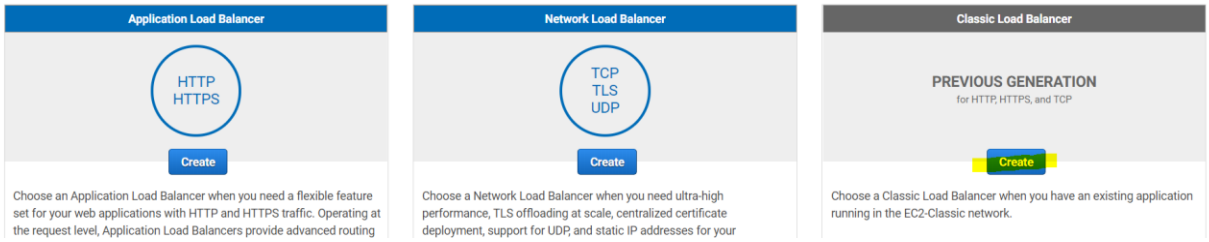
- Currently we are only this classic load balancer in real-time

  Step by step process along with screenshot will show below.

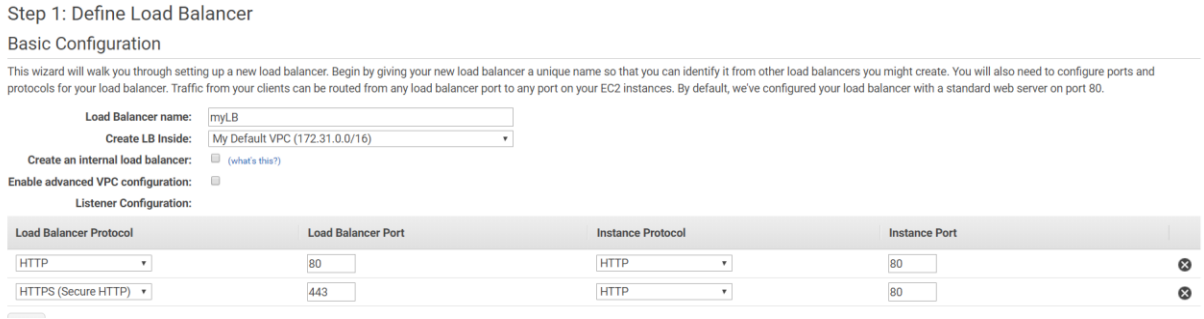
Step 1:-

First created one LOADBALANCER

Here we going to create classic loadbalancer

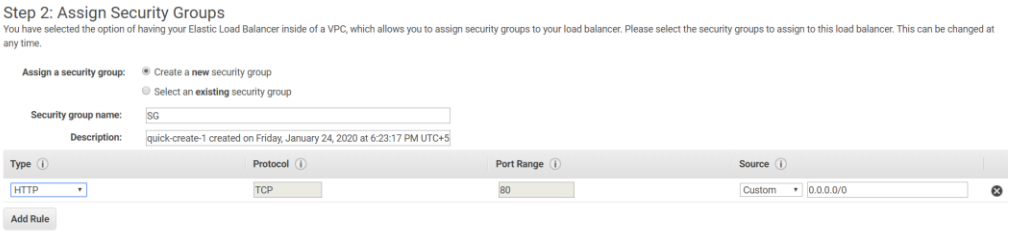


Creating loadbalancer name as myLB

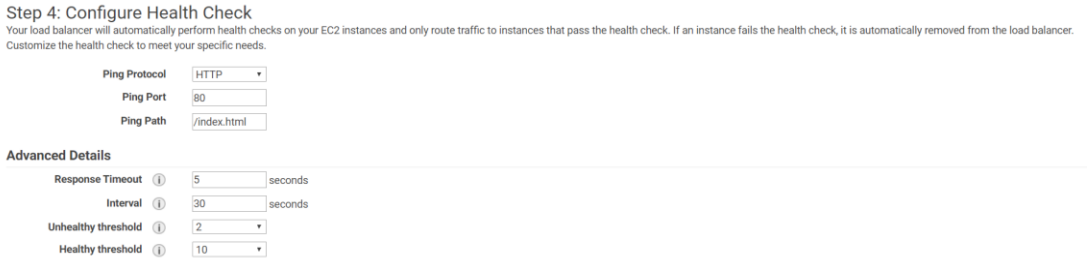


Assign security groups

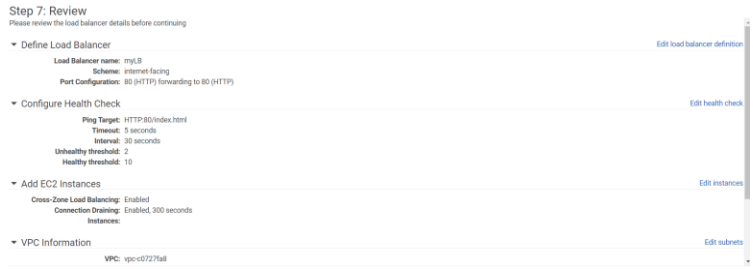
Create new group & named as SG



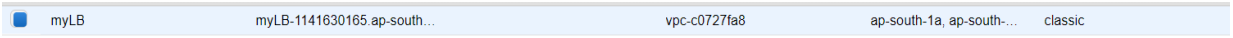
Health check on our ec2 instances



Review our loadbalancer before creating

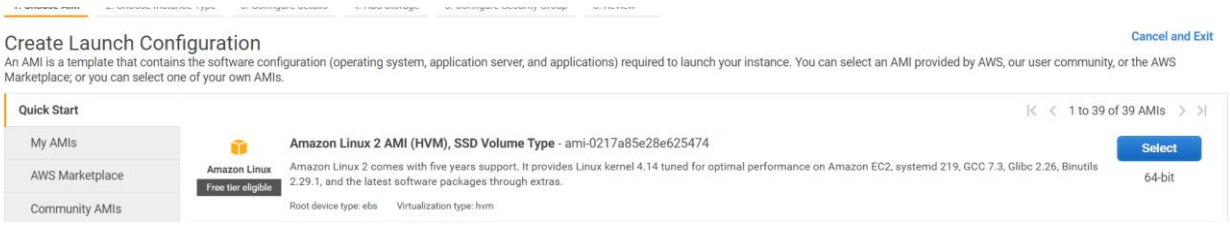


Then click create.

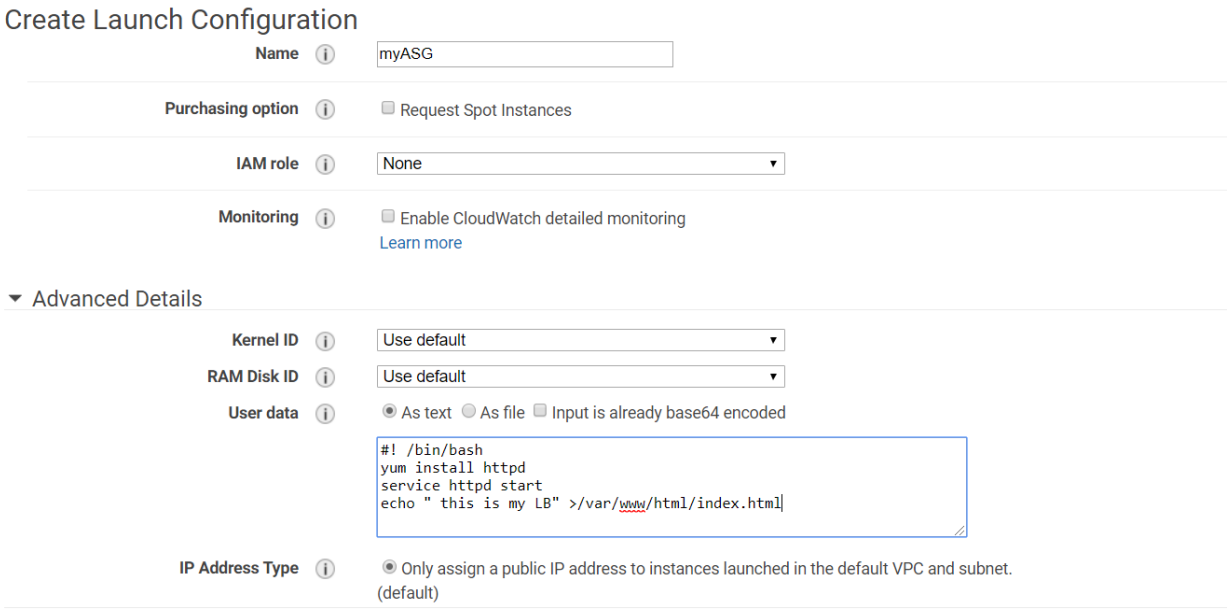


So load balancer created successfully.

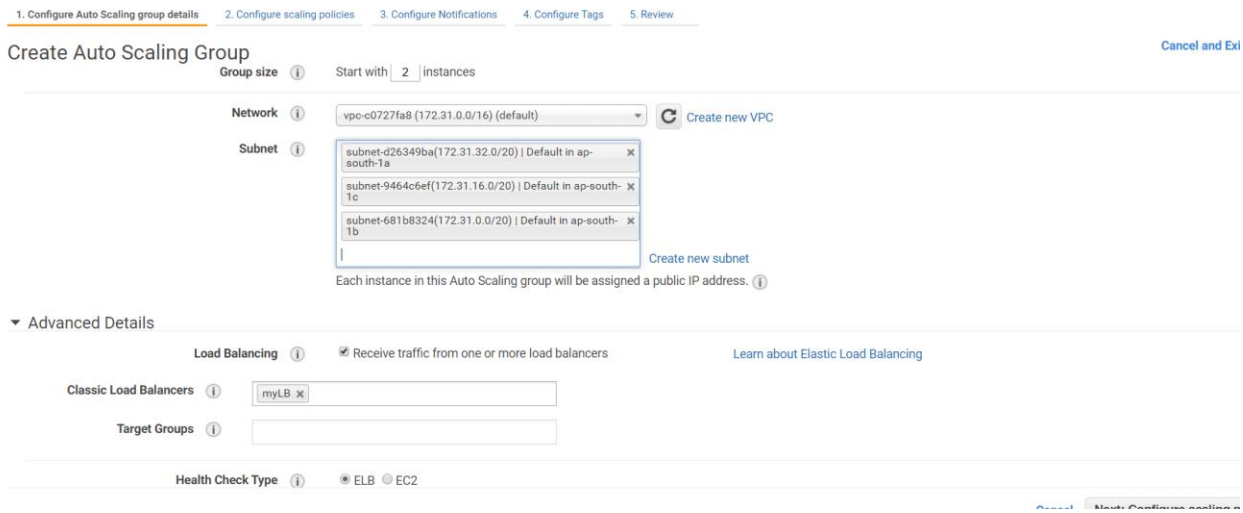
Then go to Auto scaling configurations



setting up pre configured user data & named as ASG for autoscaling

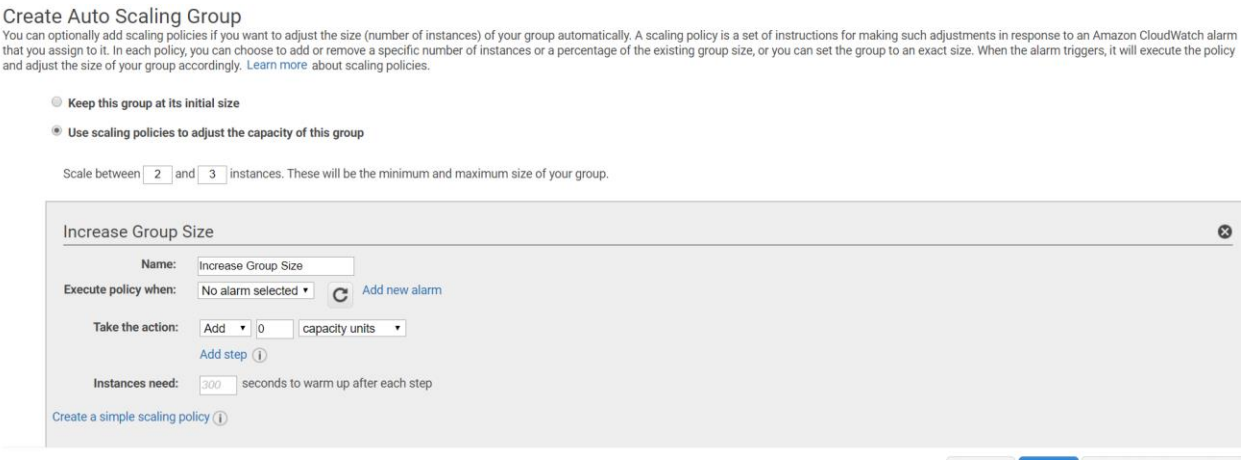


Once done, Go to auto-scaling group and enter the details below



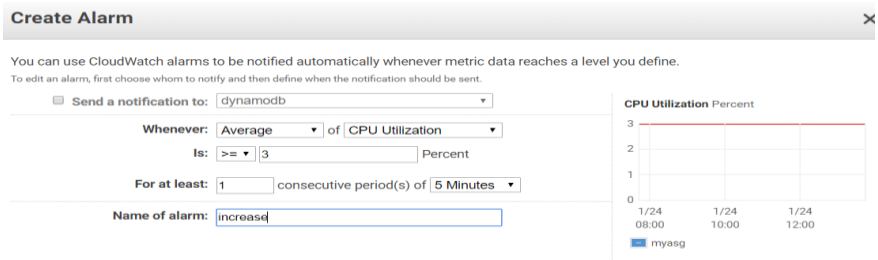
Once done, select use scaling policies to adjust the capacity of this group

In that select scale from 2 to instances, which means if once instance goes down,automatically another instance is to get created

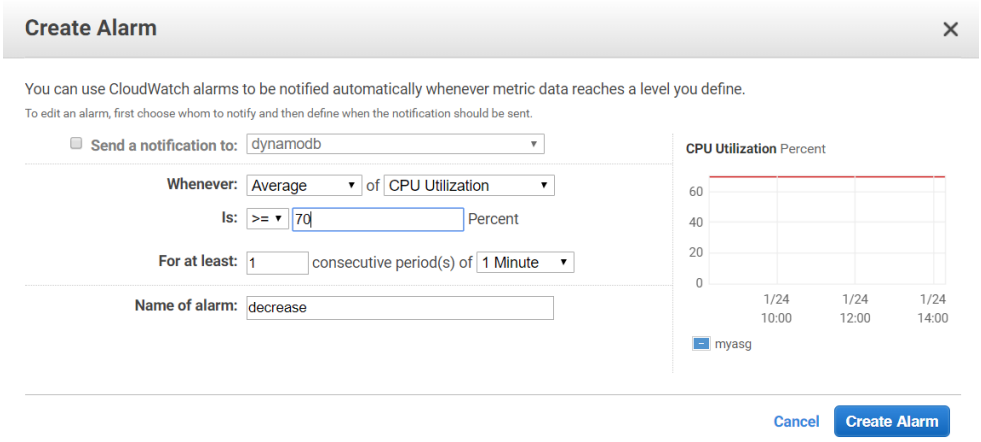


Now we going to set thresold, here shown below is just sample to get fast output for increments gng to 3 percent for utilization and for decrease 70% of utilizations

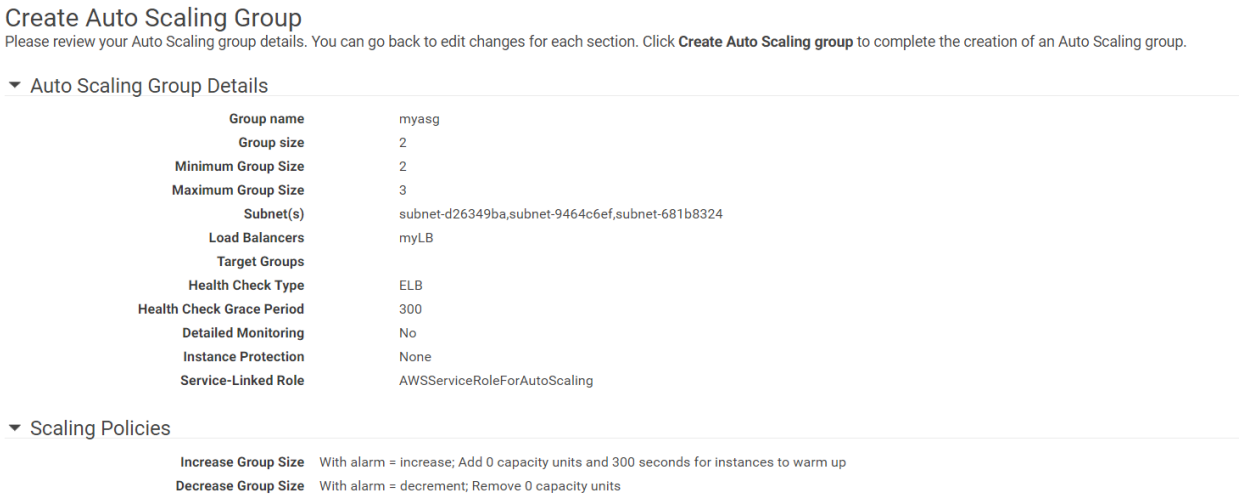
Go to add new alarm for increment and set consecutive period as 1 minute, I have set 5minutes, but make sure set 1 minute



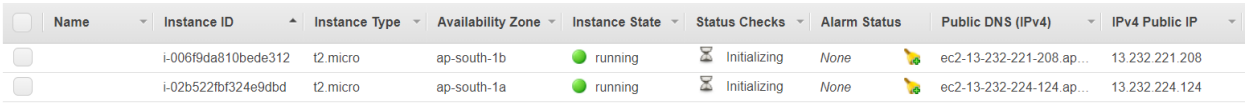
Go to add new alarm for decrement



Review:-



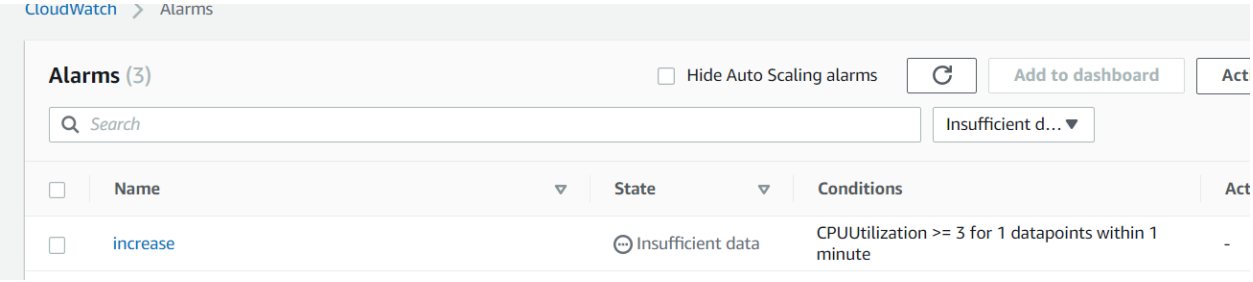
So now auto scaling group also done successfully and now we could see 2 instance running in EC2 dashbaord



So far we have done is created **LOAD BALANCER & CREATED AUTO-SCALING GROUP and set criteria as well**

After clicking our website multi times, obviously thresold level wil get increase

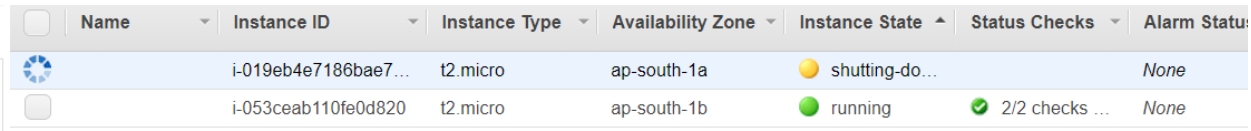
Then we get alarm in CLOUDWATCH



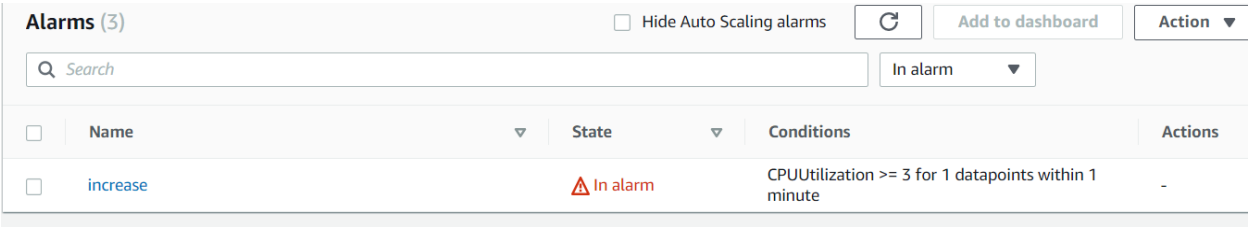
This is how it notified alert in cloud watch.

Now we going to terminate the instance, assume that one instance got accidently got terminated, lets how auto scaling works

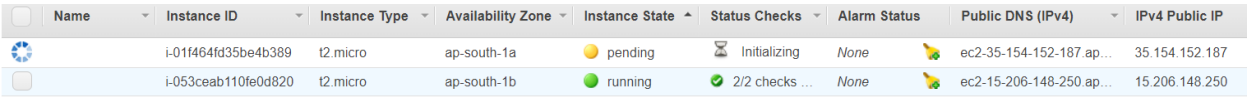
Im terminating one instance below



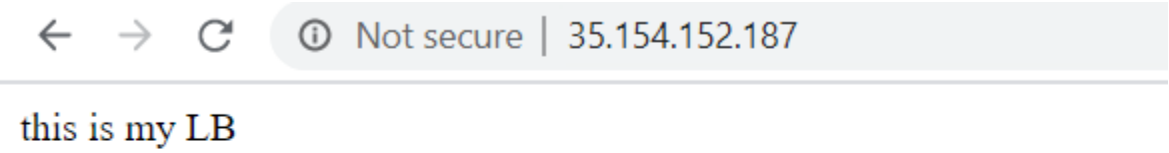
After termination received alarm in cloud watch as well, then



could see automatically instance is creating in ec2, if one instance got automatically terminated



Now I am hitting newly launched instance IP



YAAAH!!!! I can able to access my web server with out losing my connectivity

Note:-

In this notes, we covered,

- Created loadbalancer as my LB
- Then autoscaling configurations, we created 2 instances and added our Loadbalancer name as well, then configure security group set as http & ssh.
- Then we created auto scaling groups, in that we modified that 2 instance to 3 instance as per we have set thresold too,
- Then multi-time we hit our application browser,then we checked in cloud watch, it crossed thresold and we got alert as well,
- Then accidently deleted 1EC2 instance, then as per autoscaling scenario, immediately one EC2 instances got created with out losing productivy of web server application

Diagram