**WRITE UP**

**Course-end Project 3**

**Scaling the EC2 Instance Based on Monitored CloudWatch Metrics**

**Description**

Use Auto Scaling to manage the EC2 instances and use EC2 instance and capture the metrics in the CloudWatch.

Let’s take the case of Hotstar — a platform that provides on-demand video streaming services. The more the users join the streaming service platform, the more the resources in terms of servers (EC2 in AWS) Hotstar needs to invest in. This way, the load is distributed across different servers and leads to jitter-free experience for the customers while watching the videos. Another example is Amazon Prime Day, where a bevy of customers access the amazon.com site. Depending on the number of customers logging into the amazon.com site, Amazon would like to add more servers for better customer experience.

Both the above actions lead to increased customer satisfaction, which will eventually boost profits for the companies. This feature of adding and removing servers is called Dynamic Scaling and is a unique feature of the Cloud. Simply put, the users of the Cloud can scale to thousands of servers and scale down when appropriate and pay for what they use. However, that flexibility to add/subtract servers does not come with the on-premise servers, which is why the cost is always fixed. Also, during the slack time, many resources remain under- utilized which is a wastage of CAPEX. One way of adding and deleting the EC2 instances is to do it manually which may lead to extra manual effort, increase in costs, and inaccurate results.

Another approach is to use Auto Scaling to manage the EC2 instances automatically. As Auto Scaling adds more EC2 instances, the software/application installation and configuration can be automated using the AMI (Amazon Machine Images). In the previous use case, we have seen how to capture custom metrics (number of users logged in) in the CloudWatch. Here, we would need the same metric to manage (add/delete) the EC2 instances depending on the number of users logged into the website.

**Tools required**: AWS Services – ELB, CloudWatch, Auto Scaling, EC2

Steps to Implement the solution:

Step 1: create Launch Template

Step 2: Create ALB

Step 3: Create Target Group

Step 4: Create ASG

Step 5: Test ALB

Step 6: Add Dynamic Scaling policy

Step 7: Test ASG scale-in

Step 8: Test ASG Scale-out