**Mini Home Work -4**

* **Elastic Load Balancer Created named “*MyProgramElasticLoadBalancer*”**

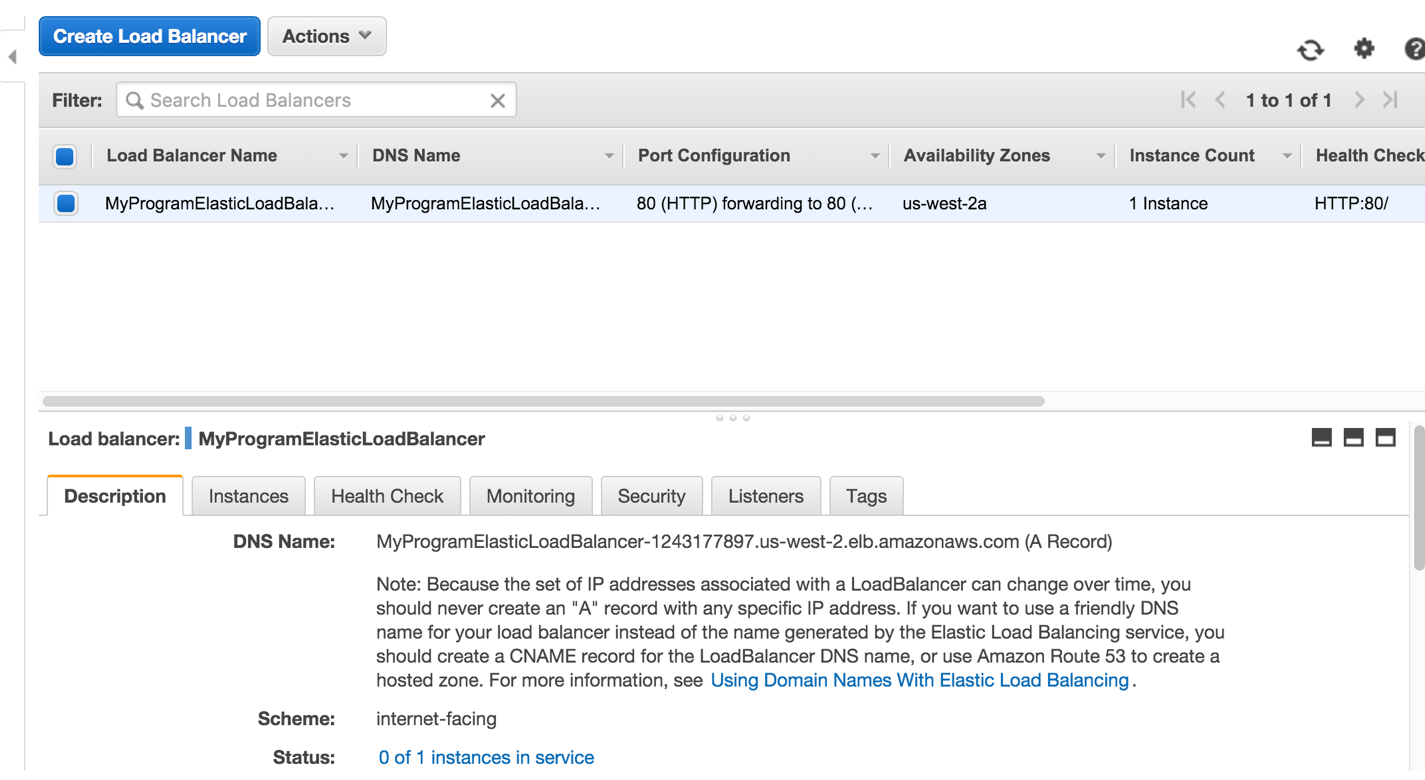


Figure 1- Screen shot of Console showing Elastic Load Balancer Created

* **Auto scale parameters for Auto Scale Group “*MyAutoScaleGroup*”**
  + Auto Scale minimum = 2
  + Auto Scale maximum = 6
* **Policy Parameters**
  + Policy named “**ScaleUpPolicy**”
    - Scale up by Adjusment 2
  + Policy named “**ScaleDownPolicy**”
    - Scale down by Adjustment -2
* **Cloud Watch Alarms**
  + Alarm named “alarm-scale-up”
    - Threshold average CPU Utilization = 40
  + Alarm named “alarm-scale-down”
    - Threshold average CPU Utilization = 40

Following loads where created on initial 2 machines using “***stress***” linux tool and running it on 4 CPU’s

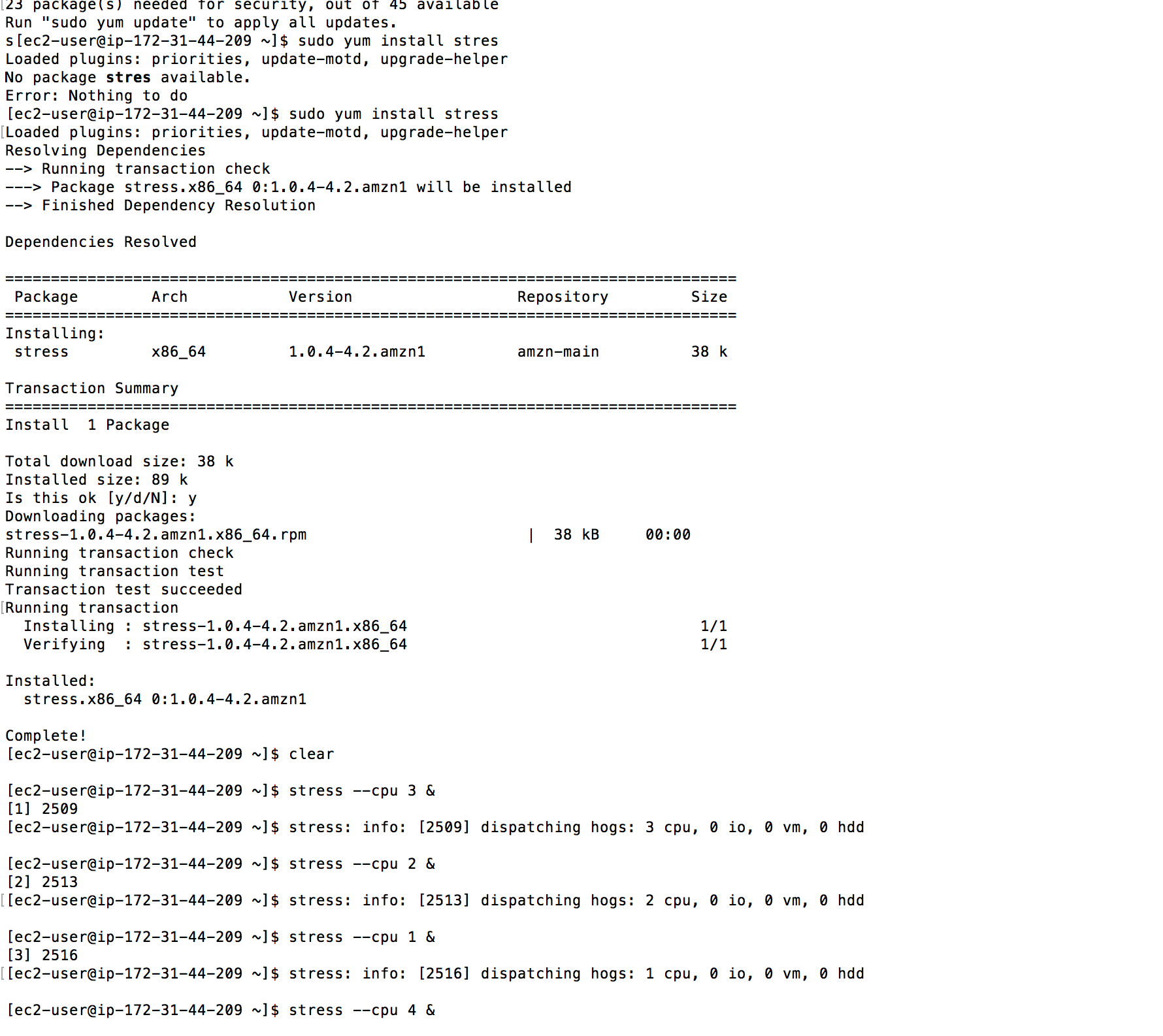


Figure 2 - Load Generated using “stress” tool

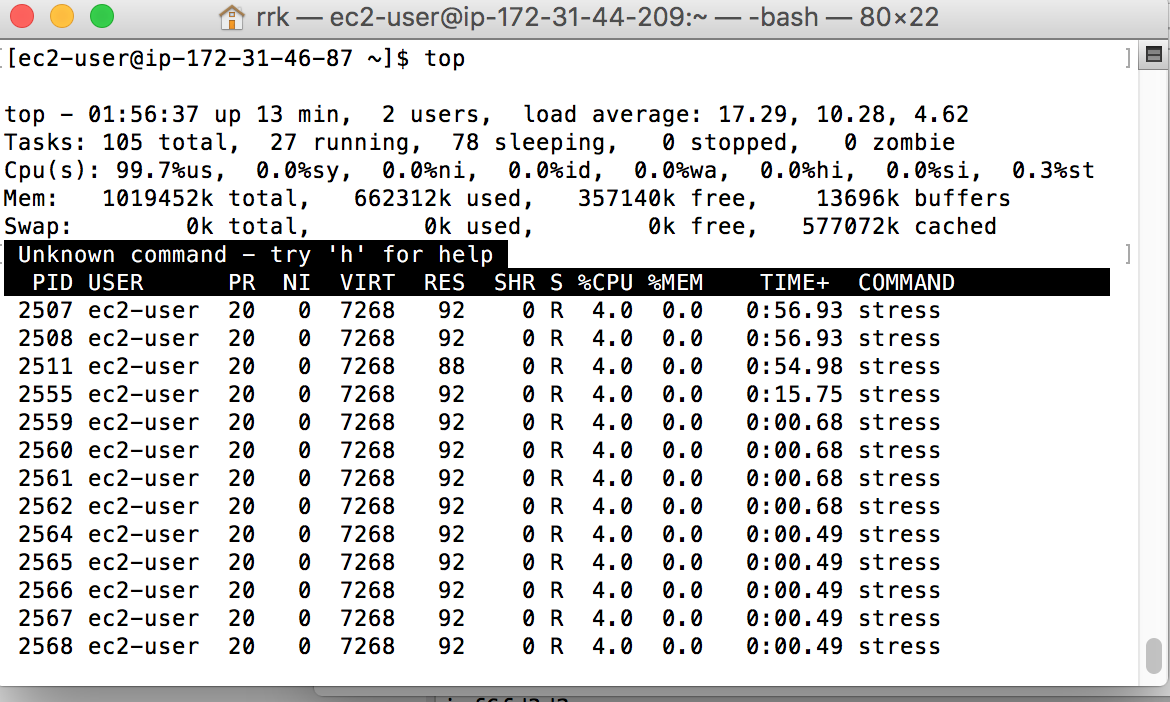


Figure 3- Running top command on server

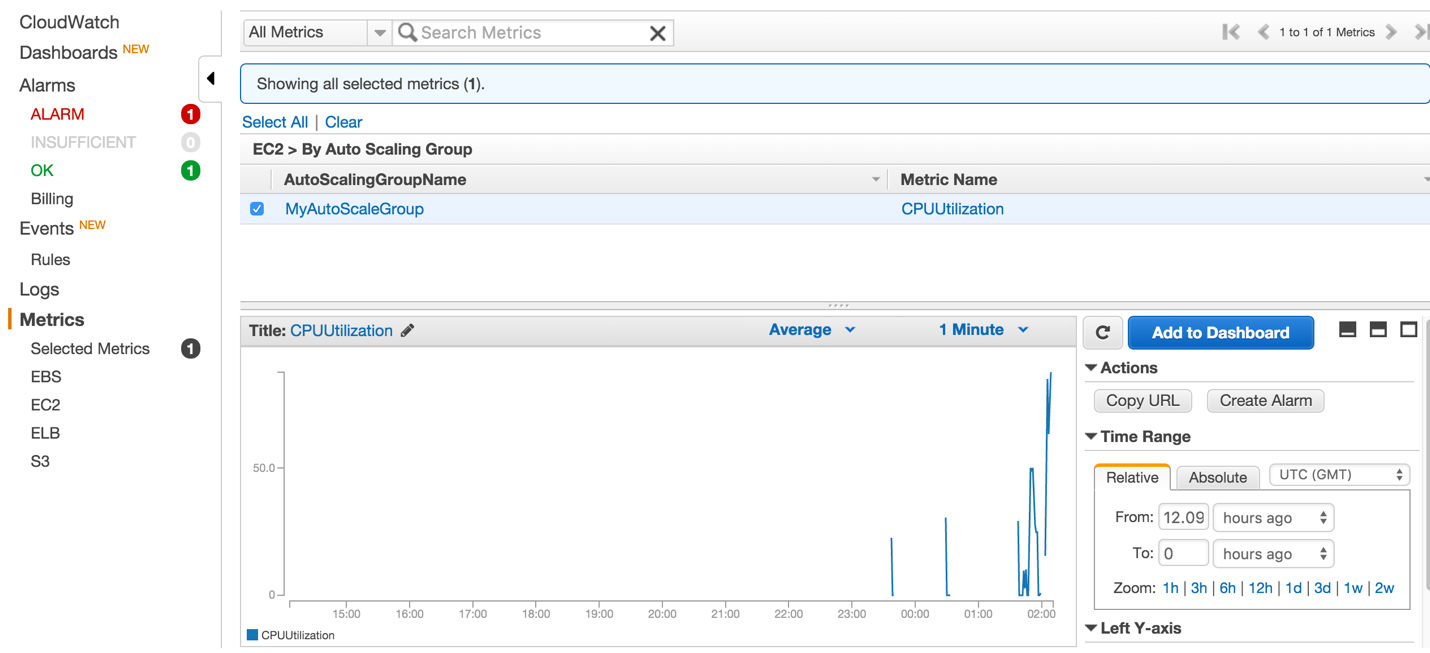


Figure 4- CPU Utilization exceeds 40%

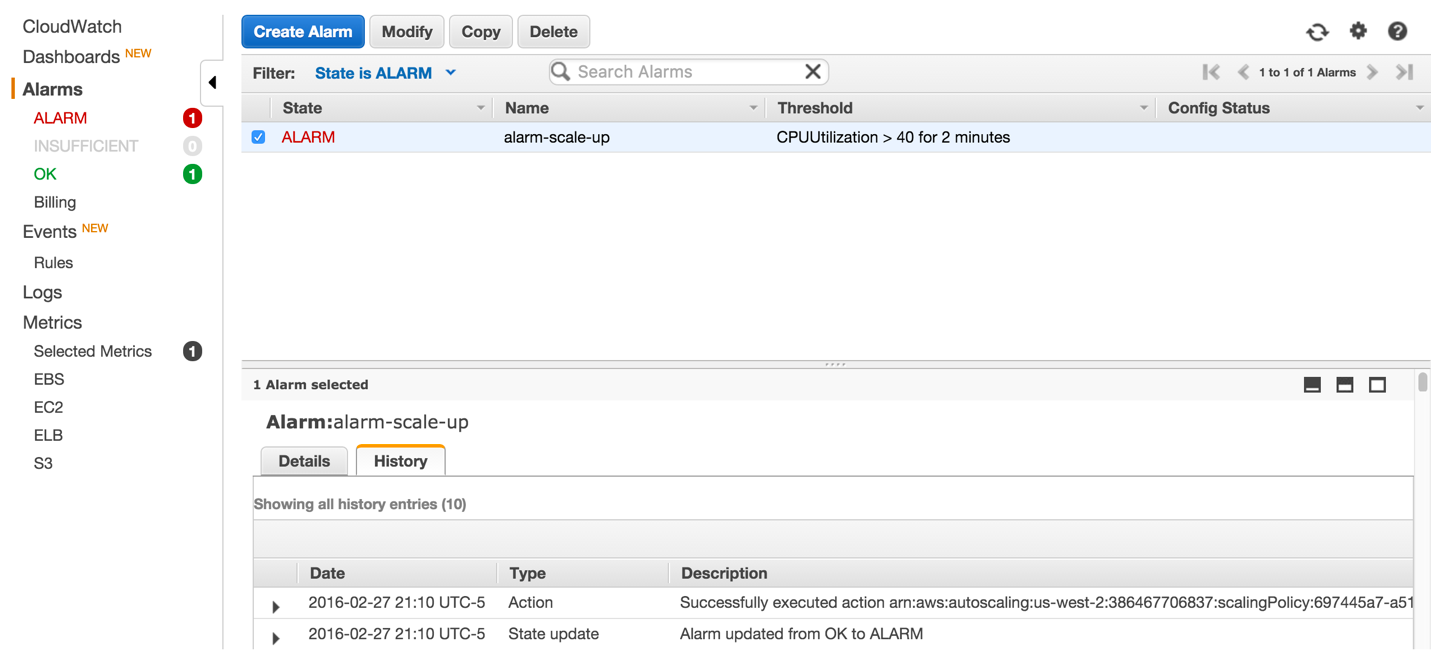


Figure 5 - Alarm “**alarm-scale-up**” is triggered

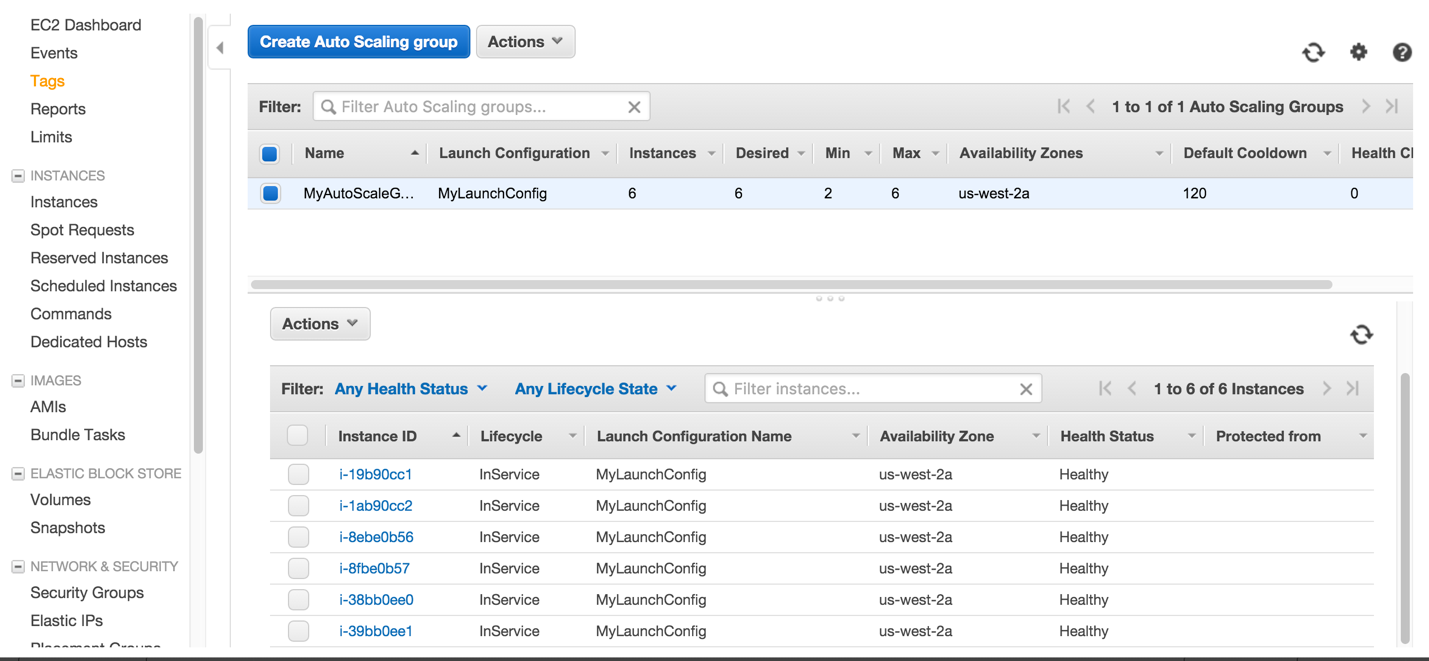


Figure 6 – New Instances added to “MyAutoScalingGroup”

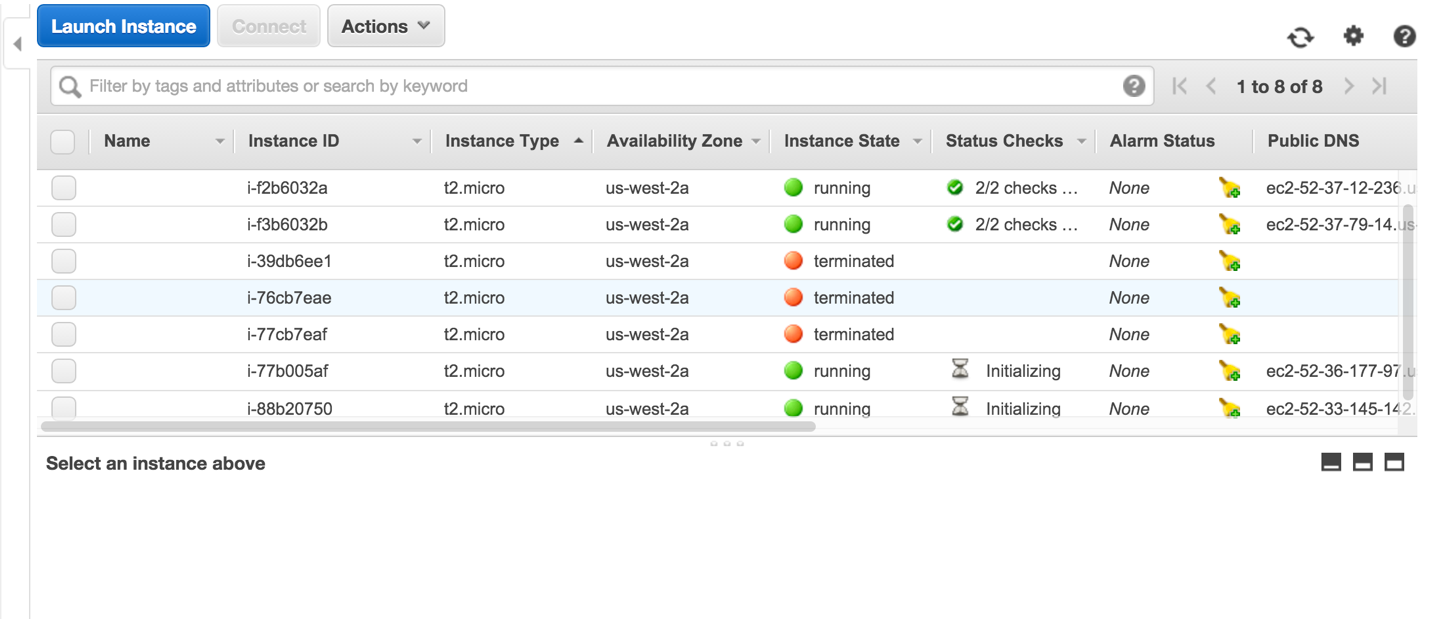


Figure 7 – Instance launched after Alarm

Following was done programmatically using AWS Java SDK

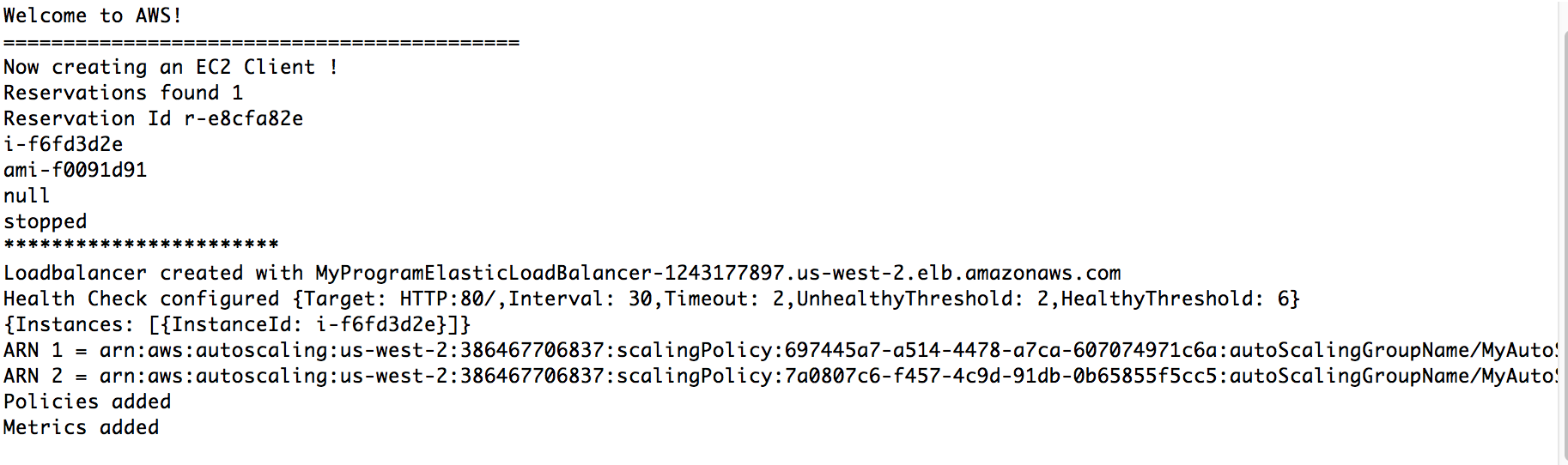


Figure 8 – Console in Eclipse