**Exercise 3**

*Configuring a Load Balancer*

*Stress Testing*

**Prior Knowledge**

Unix Command Line Shell

Exercise 2: Auto Scaling groups and Launch Configurations

**Learning Objectives**

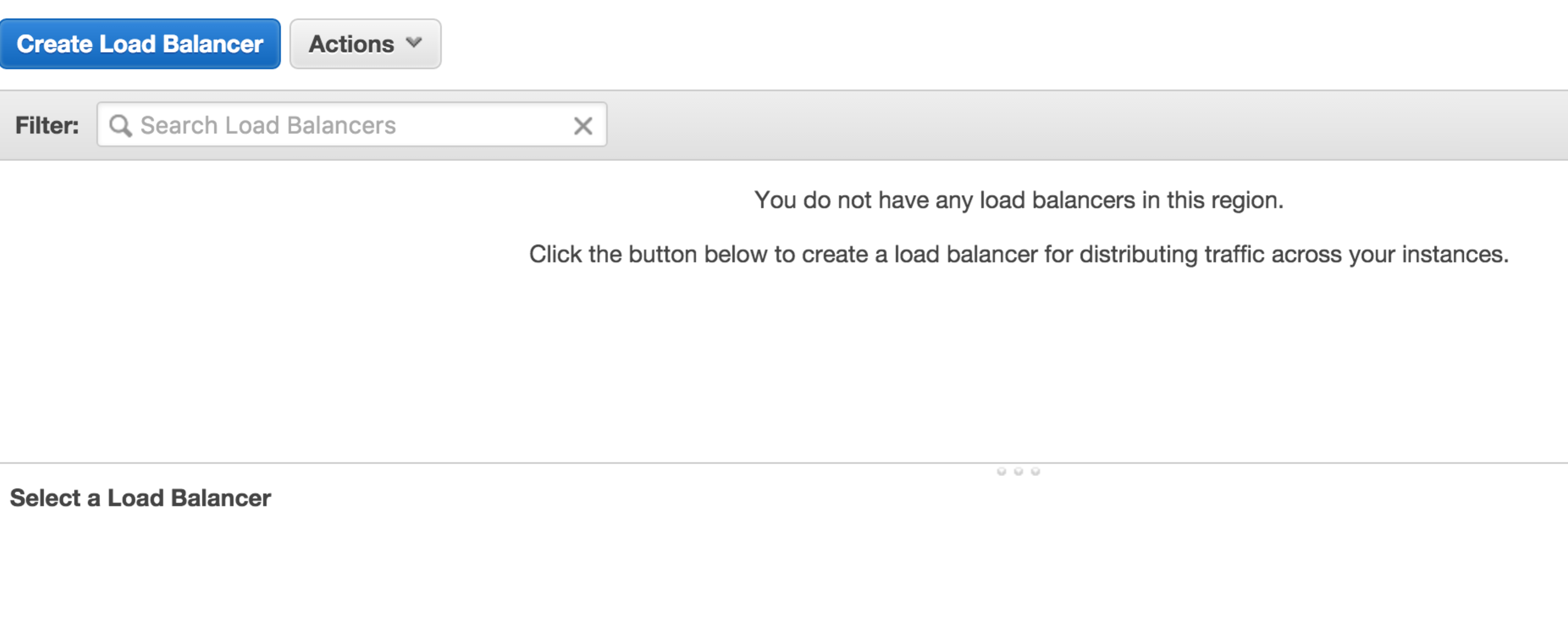
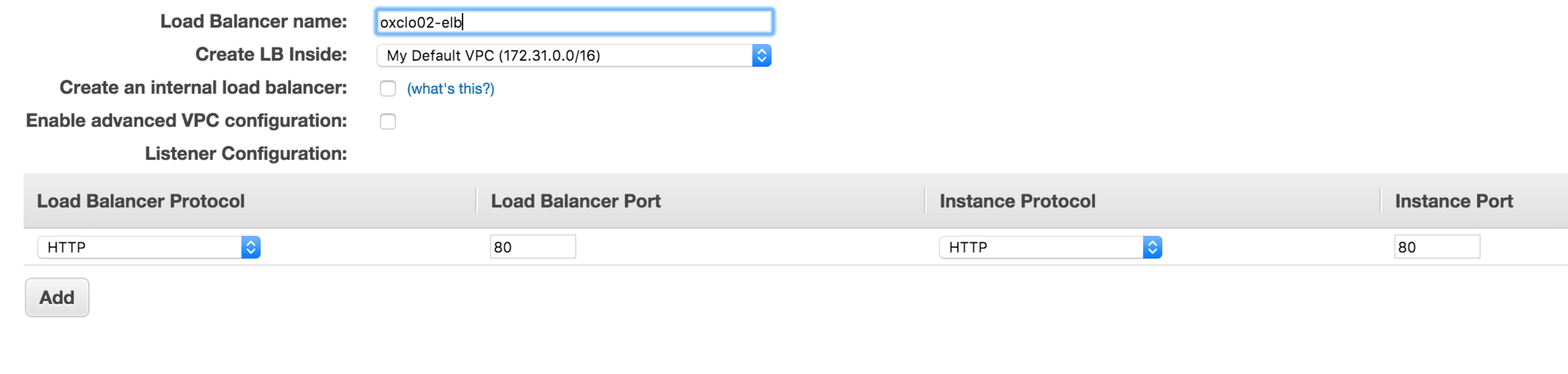
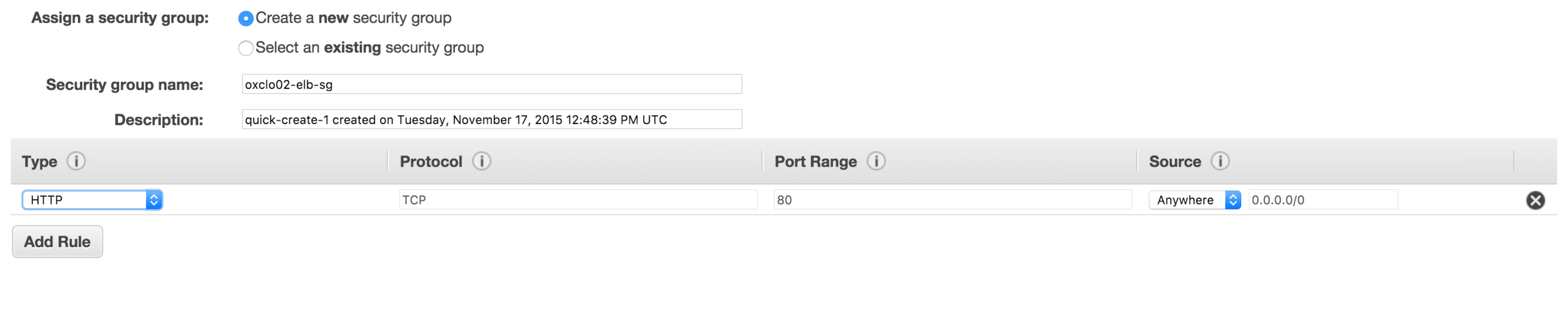
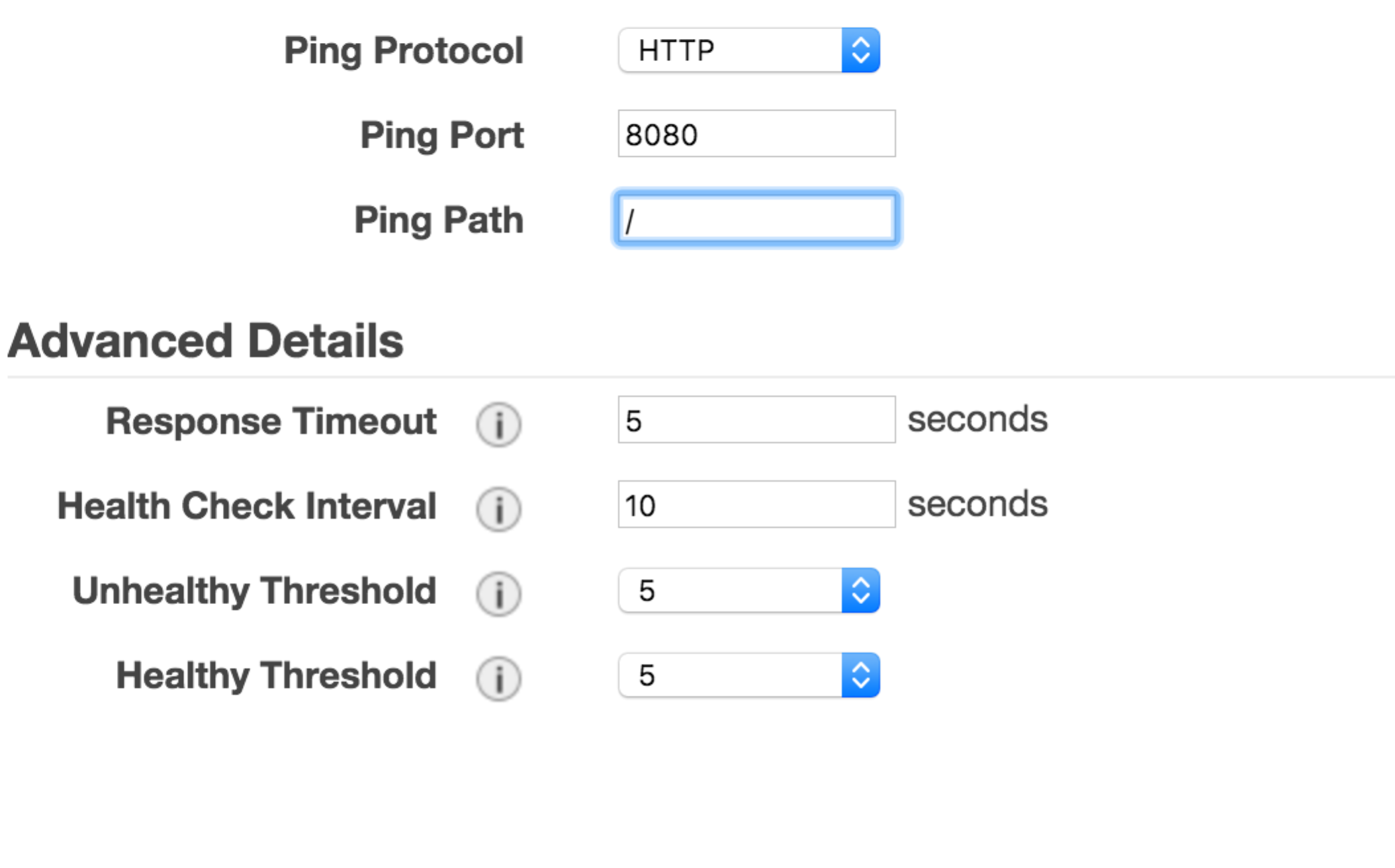
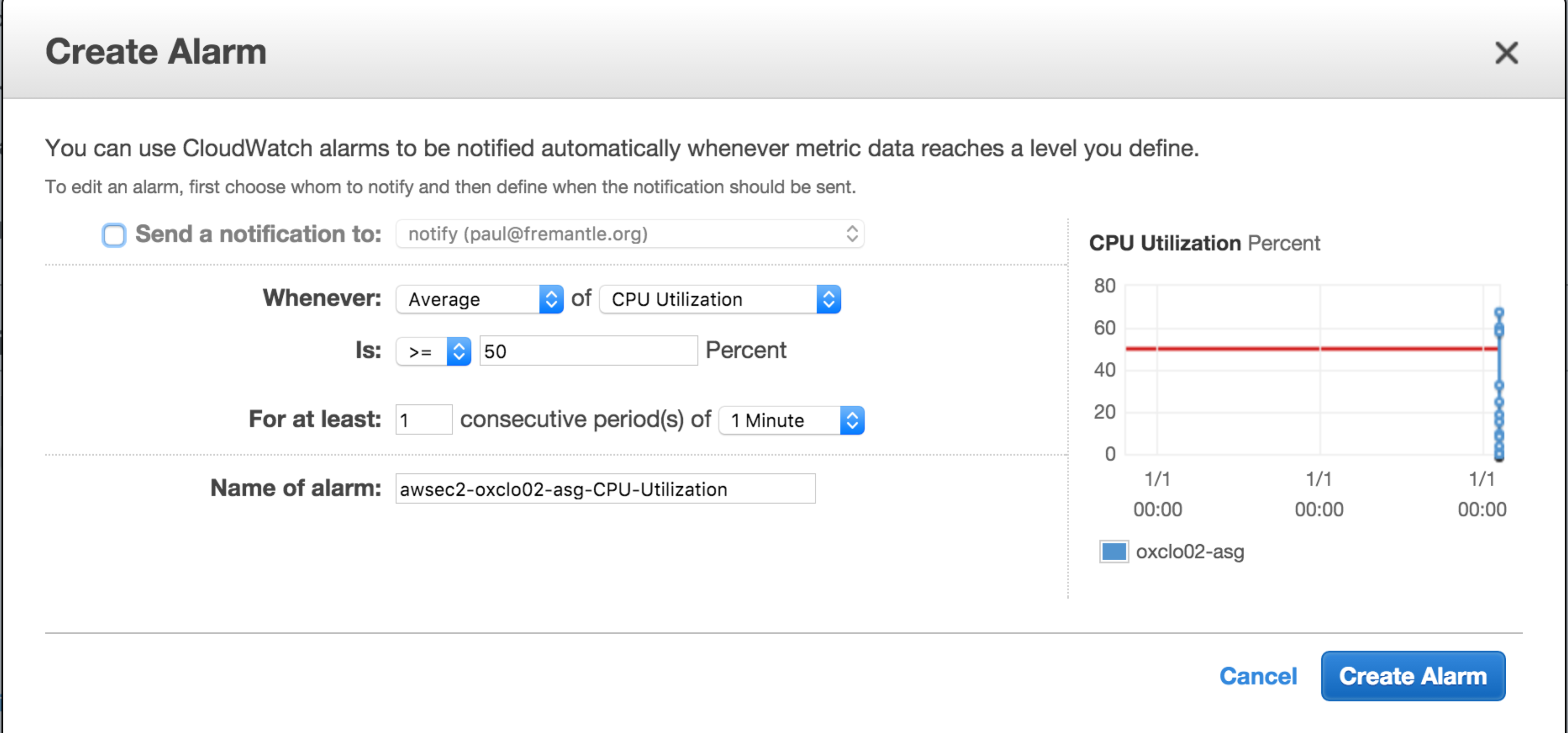
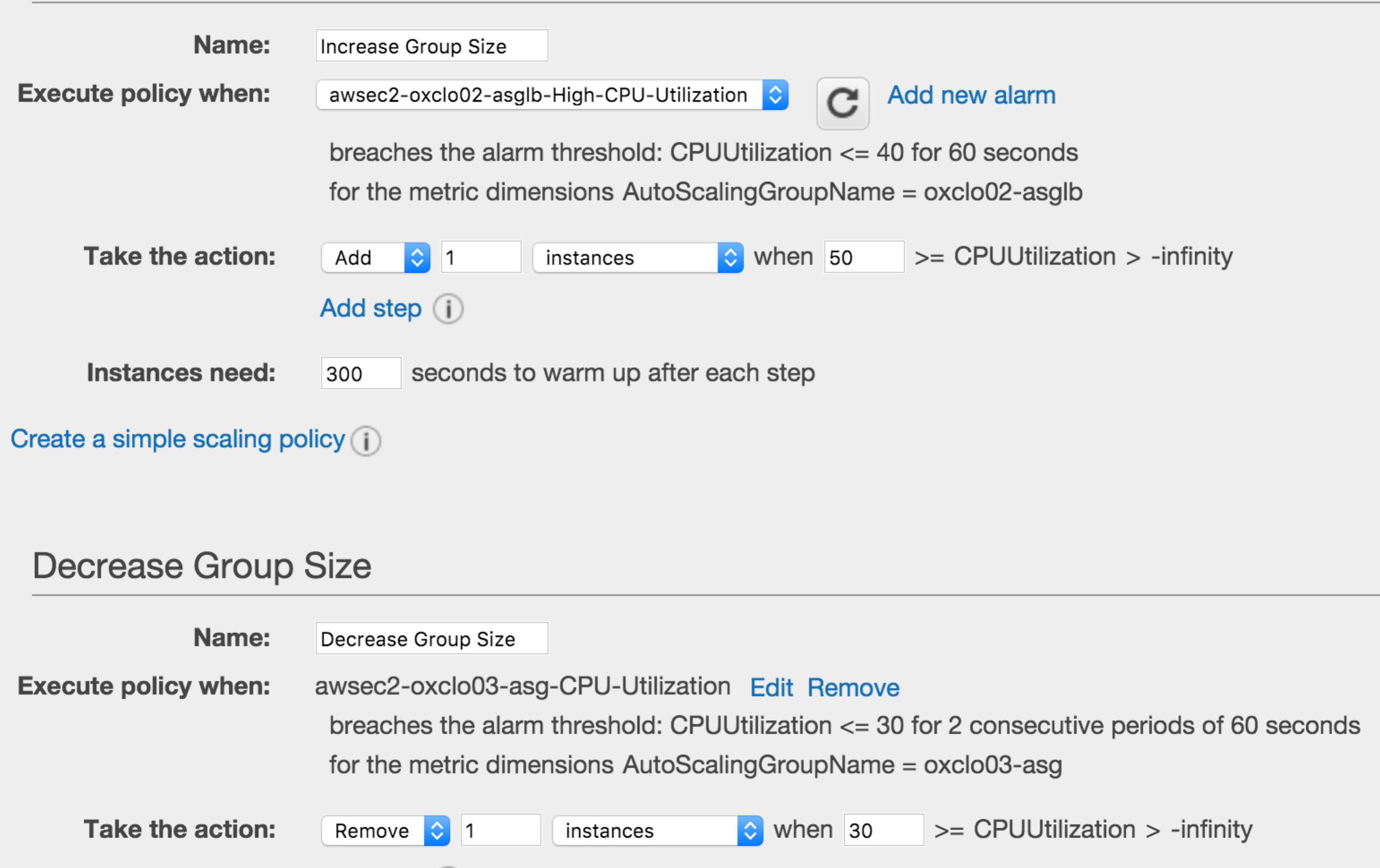
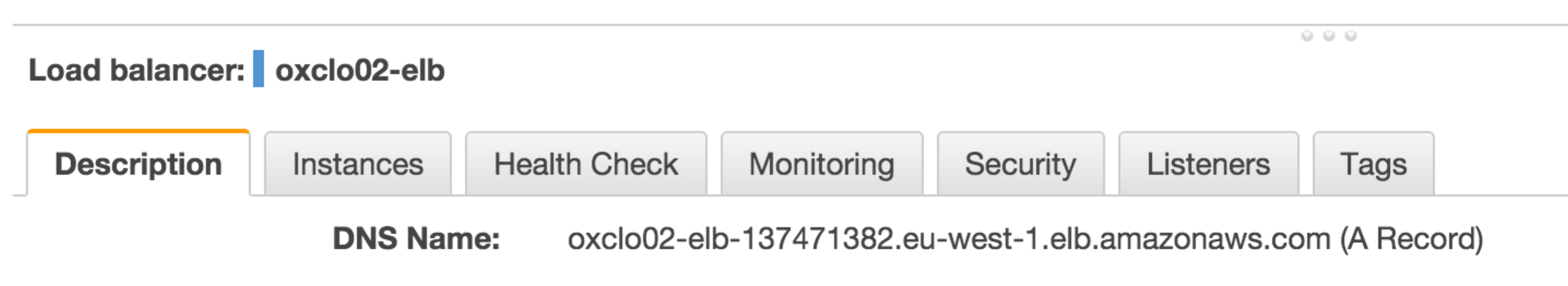
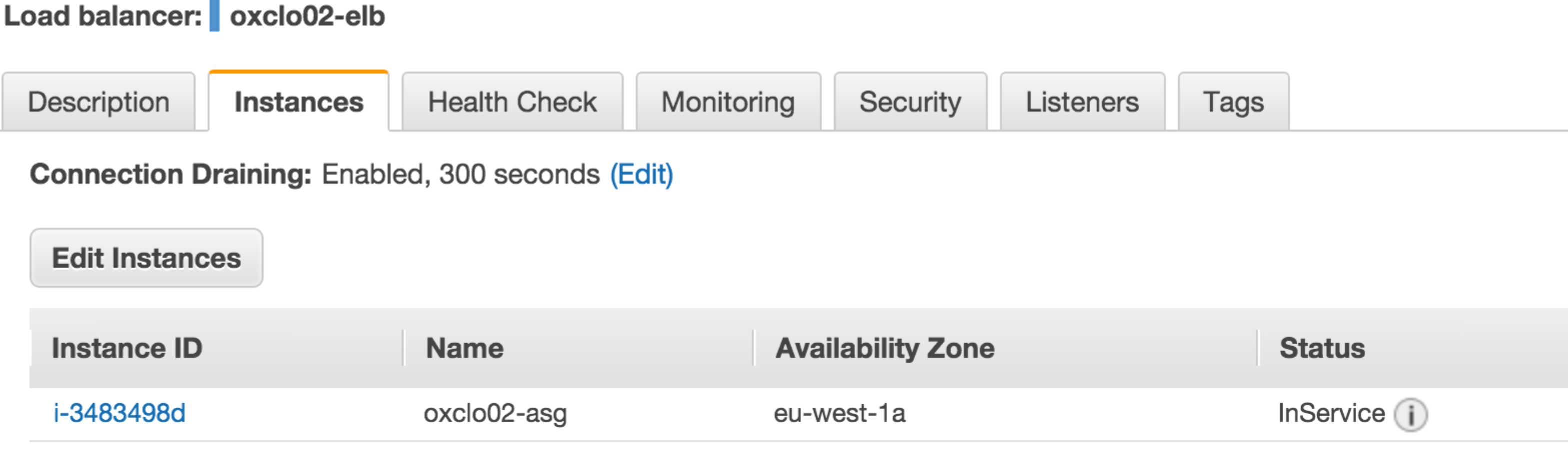
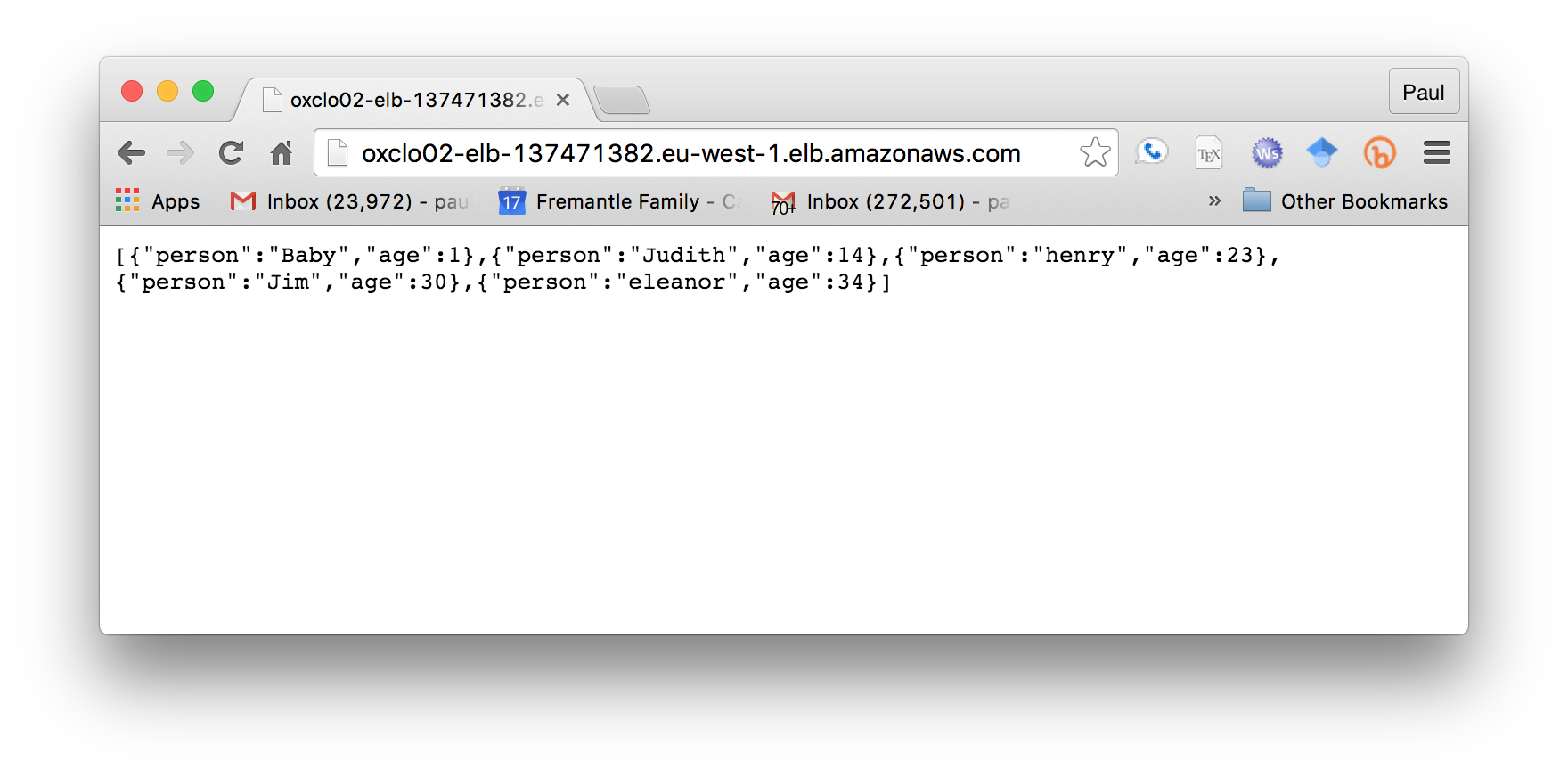
Creating an elastically scaled system in the cloud

How to stress test using Linux siege command

**Software Requirements**

Browser and AWS account, previous configuration from Exercise 2

**Part A: Setting up a Load Balancer and ELB Auto Scale Group**

1. Go to the AWS Console and then the EC2 Console.
2. Near the bottom of the left hand menu, find Load Balancers and Click on it. You will see something like this (although other students may have created load balancers that will show up).  
   
3. Click **Create Load Balancer**
4. In the screen following:
   1. Set name to *userid-*elb (e.g. oxclo02-elb)
   2. Leave the Load Balancer protocol as HTTP, etc, except change the **Instance Port** to 8080.   
      This will mean that traffic coming to the LB will be sent to port 8080 on the instance servers.  
      
5. Click **Next: Assign Security Groups**
6. Select **Create a New Security Group**
7. Give it the name *userid-*elb-sg (e.g. oxclo02-elb-sg)
8. Make sure the rule says:  
   HTTP TCP 80 Anywhere 0.0.0.0/0  
     
   
9. Click **Next: Configure Security Settings**
10. Ignore the warning and click: **Next: Configure Health Check**
11. Change the settings as follows:
    1. Ping Protocol: HTTP
    2. Ping Port: 8080
    3. Ping Path: /
    4. Response Timeout: 5
    5. Health Check interval: 10
    6. Unhealthy threshold: 5
    7. Healthy threshold: 5  
       
12. Click **Next: Add EC2 Instances**
13. Do NOT add any instances! Click **Next: Add Tags**
14. Add the tag with Key/Value: Name / *userid*-asi
15. Click **Review and Create** then **Create**
16. Click **Close**
17. Now let’s create our AutoScaling Group
18. Go back to creating an Auto Scale Group like last time. **(Auto Scaling Groups -> Create Auto Scaling Group)**
19. Create from an existing Launch Configuration and choose your own launch config that you previously created. Click **Next Step**
20. On the following screen:  
    1. Give it a group name of *userid-*asg (e.g. oxclo02-asg)
    2. Add one or more subnets as before
    3. Expand the **Advanced Details**
    4. Click **Receive Traffic from Elastic Load Balancers**
    5. Select your own Load Balancer from the options
    6. Change the Health Check type to ELB
    7. Leave the Grace period as 300 seconds
    8. Click **Next: Configure Scaling Policies**
21. On the following screen   
    1. Select **Use scaling policies….**
    2. Change it to support scaling between 1 and 4 instances
    3. Click Add New Alarm
    4. If you want notifications, choose your own topic that you defined before.
    5. Change the Alarm to fire when the CPU utilization is >= 50% for more than 1 minute (we want to see scaling!)  
       
    6. Click **Create Alarm**
22. Now update the rule to **Add 1 instance**
23. Set **Instances need 300 seconds to warm up after each step**
24. Create a similar Alarm for when CPU utilization is <= 30% for 2 minutes, and change the rule to Remove 1 instance.
25. It should look like:  
    
26. Click **Next: Configure Notifications**
27. Click **Next: Configure Tags**
28. Add the tag: Name / *userid*-asi
29. Click **Review**
30. Click **Create Autoscaling Group**
31. Go and see if your instances are being started.   
      
    **PART B – Stress testing**
32. Navigate to view your ELB’s dashboard page. You can find the DNS address of your ELB this way:  
    
33. After the system has warmed up and your instance is running, it will eventually be tested by the ELB and become **In-Service.** You should see something like this:
34. Once you have an InService instance, copy and paste the DNS name into the address bar of your browser. You should see JSON returned from the node.js app.   
      
      
    Notice this is now available on port 80 and no longer using 8080.