

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).

The screenshot shows the AWS Load Balancers console. At the top, there are two buttons: 'Create Load Balancer' (blue) and 'Actions'. Below these are 'Filter' and a search bar ('Search Load Balancers'). A message states 'You do not have any load balancers in this region.' Below this, a button says 'Click the button below to create a load balancer for distributing traffic across your instances.' At the bottom, there is a link 'Select a Load Balancer'.

3. Click **Create Load Balancer**



© Paul Fremantle 2015. Licensed under the Creative Commons 3.0 BY-SA (Attribution-Sharealike) license.
See <http://creativecommons.org/licenses/by-sa/3.0/>

4. In the screen following:
 - a. Set name to *userid-elb* (e.g. oxclo02-elb)
 - b. 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.

The screenshot shows the 'Create a new load balancer' wizard. Step 1: Set Load Balancer configuration. The 'Load Balancer name' field contains 'oxclo02-elb'. The 'Create LB Inside' dropdown is set to 'My Default VPC (172.31.0.0/16)'. Under 'Listener Configuration', the 'Load Balancer Protocol' is 'HTTP' and 'Load Balancer Port' is '80'. The 'Instance Protocol' is 'HTTP' and 'Instance Port' is '80'. An 'Add' button is visible.

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

The screenshot shows the 'Create a new security group' wizard. Step 2: Set security group rules. A single rule is listed: Type 'HTTP', Protocol 'TCP', Port Range '80', Source 'Anywhere', and Destination '0.0.0.0/0'. An 'Add Rule' button is visible.

9. Click **Next: Configure Security Settings**

10. Ignore the warning and click: **Next: Configure Health Check**



11. Change the settings as follows:

- a. Ping Protocol: HTTP
- b. Ping Port: 8080
- c. Ping Path: /
- d. Response Timeout: 5
- e. Health Check interval: 10
- f. Unhealthy threshold: 5
- g. Healthy threshold: 5

The screenshot shows the 'Advanced Details' section of an AWS Auto Scaling configuration. It includes fields for Response Timeout (5 seconds), Health Check Interval (10 seconds), Unhealthy Threshold (5), and Healthy Threshold (5). The Ping Protocol is set to HTTP, Ping Port is 8080, and Ping Path is /. The 'Advanced Details' section is collapsed.

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:

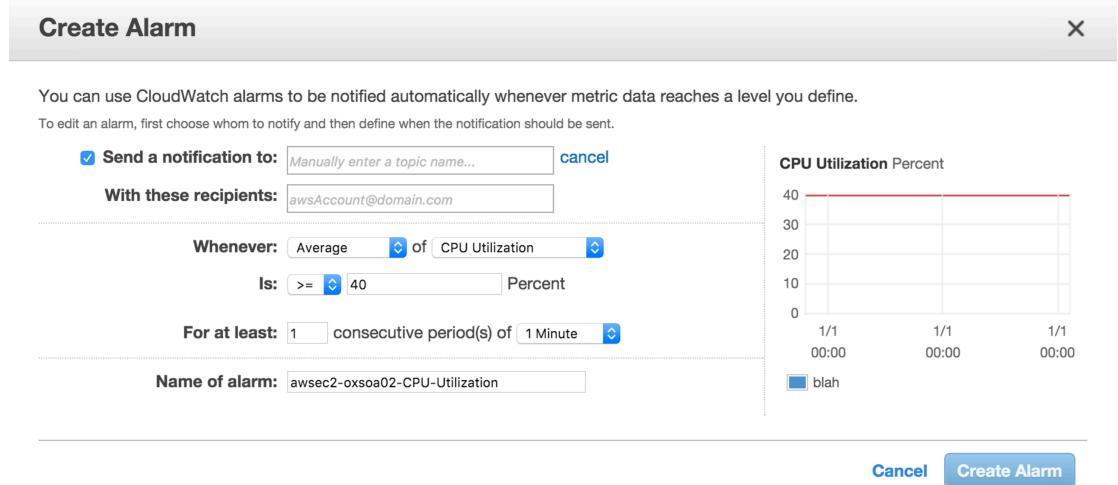
- a. Give it a group name of *userid*-asg (e.g. oxclo02-asg)
- b. Add one or more subnets as before
- c. Expand the **Advanced Details**
- d. Click **Receive Traffic from Elastic Load Balancers**



- e. Select your own Load Balancer from the options
- f. Change the Health Check type to ELB
- g. Leave the Grace period as 300 seconds
- h. Click **Next: Configure Scaling Policies**

21. On the following screen

- a. Select **Use scaling policies....**
- b. Change it to support scaling between 1 and 4 instances
- c. Click Add New Alarm
- d. If you want notifications, choose your own topic that you defined before.
- e. Change the Alarm to fire when the CPU utilization is $\geq 40\%$ for more than 1 minute (we want to see scaling, so this is deliberately low)



- f. 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 $\leq 30\%$ for 2 minutes, and change the rule to Remove 1 instance.



25. It should look like:

The screenshot shows two scaling policies for an Auto Scaling group. The top policy, named 'Increase Group Size', triggers when CPUUtilization >= 40 for 60 seconds. It adds 1 instance when CPUUtilization < +infinity. The bottom policy, named 'Decrease Group Size', triggers when CPUUtilization <= 30 for 2 consecutive periods of 60 seconds. It removes 1 instance when CPUUtilization > -infinity. Both policies have a 'Create a simple scaling policy' link.

Increase Group Size

Name: Increase Group Size

Execute policy when: awsec2-oxsoa02-CPU-Utilization Edit Remove
breaches the alarm threshold: CPUUtilization >= 40 for 60 seconds
for the metric dimensions AutoScalingGroupName = blah

Take the action: Add 1 instances when 40 <= CPUUtilization < +infinity

Add step ⓘ

Instances need: 300 seconds to warm up after each step

Create a simple scaling policy ⓘ

Decrease Group Size

Name: Decrease Group Size

Execute policy when: awsec2-oxclo03-asg-CPU-Utilization Add new alarm C
breaches the alarm threshold: CPUUtilization <= 30 for 2 consecutive periods of 60 seconds
for the metric dimensions AutoScalingGroupName = oxclo03-asg

Take the action: Remove 1 instances when 30 >= CPUUtilization > -infinity

Add step ⓘ

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:

The screenshot shows the ELB dashboard for 'oxclo02-elb'. The 'Description' tab is selected. The DNS name listed is 'oxclo02-elb-137471382.eu-west-1.elb.amazonaws.com (A Record)'.

Load balancer: oxclo02-elb

Description Instances Health Check Monitoring Security Listeners Tags

DNS Name: oxclo02-elb-137471382.eu-west-1.elb.amazonaws.com (A Record)

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:



Load balancer: **oxclo02-elb**

Description Instances Health Check Monitoring Security Listeners Tags

Connection Draining: Enabled, 300 seconds ([Edit](#))

[Edit Instances](#)

Instance ID	Name	Availability Zone	Status
i-3483498d	oxclo02-asg	eu-west-1a	InService (i)

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.

35. We are going to create a new instance in the same subnet to stress test the servers from. We could do it from here, but we will take out network delays if we can do it within the Amazon EC2 network.



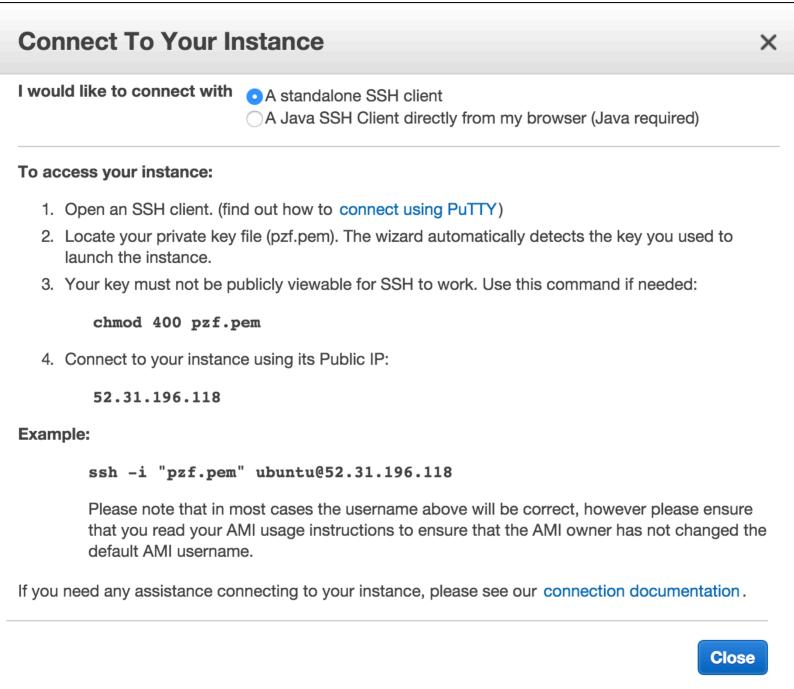
36. Using the EC2 Launch wizard like before, start a new instance with the following settings:

- a. **Ubuntu Server 14.04 LTS (HVM)**
- b. **t2.medium** (we want a beefier machine to be able to drive our nodes hard)
- c. User Data: (this is available in <http://freo.me/oxclo-siege-ud>)

```
#!/bin/bash
# verbosity
set -e -x
# update the package list
apt-get update
# install node, node package manager and git.
apt-get -y install siege
# set more file descriptors
echo "* hard nofile 64000" >> /etc/security/limits.conf
echo "* soft nofile 64000" >> /etc/security/limits.conf
```

- d. Tag Name: *userid-siege*
- e. Security Group: node-security-group
- f. Your existing SSH Key

37. Once the instance is started, right-click on it and select “Connect”
You will see a screen like:



38. You should be able to cut and paste the SSH line to your Terminal window and SSH into the server. Alternatively try out the built-in Java SSH client.

39. Accept the fingerprint as before.

40. In the SSH session type:

```
siege -c200 -t10m http://your-lb-dns-goes-here &
```

e.g

```
siege -c200 -t10m http://oxclo02-elb-137471382.eu-west-1.elb.amazonaws.com &
```

41. You should see something like:

```
** SIEGE 3.0.5
** Preparing 200 concurrent users for battle.
The server is now under siege...
```

42. This is basically hitting your Load Balancer with 200 concurrent clients for 10 minutes.

43. Unfortunately we are using a slightly old version of siege that has a few bugs, and it doesn't support high concurrency without crashing. We need to ramp up the stress on the cluster to cause it to scale up. This is why we added the & to the end of the line. That causes siege to run in the background. Simply pushing the up arrow will retrieve the same command line and we can start another siege.

Start 3 or 4 this way.

44. Unless we run out of network bandwidth, this should push the instances's average CPU above 40% and cause the Scaling Group to start another server.

45. Assuming all is well you should see a new instance spawned shortly.

46. You can also check the Auto Scaling Group's Activity History

The screenshot shows the AWS CloudWatch Metrics Activity History interface for an Auto Scaling Group named 'oxclo03-asg'. The top navigation bar includes tabs for 'Details', 'Activity History' (which is selected), 'Scaling Policies', 'Instances', 'Notifications', and 'Tags'. Below the navigation is a search bar labeled 'Filter scaling history...'. The main content area displays a table of activity items. The columns are 'Status', 'Description', 'Start Time', and 'End Time'. There are two entries: one for 'Waiting for instance warmup' (status orange) and one for 'Successful' (status green). Both entries describe launching a new EC2 instance with specific IDs and times.

Status	Description	Start Time	End Time
Waiting for instance warmup	Launching a new EC2 instance: i-fbd8ef42	2015 November 17 14:16:55 UTC	
Successful	Launching a new EC2 instance: i-f2bf754b	2015 November 17 13:24:22 UTC	2015 November 17 13:25:26 UTC



47. And the Elastic Load Balancer's instances

The screenshot shows the AWS Elastic Load Balancer Instances page. At the top, it says "Load balancer: oxclo02-elb". Below that is a navigation bar with tabs: Description, Instances (which is selected), Health Check, Monitoring, Security, Listeners, and Tags. A note below the tabs says "Connection Draining: Enabled, 300 seconds (Edit)". There is a "Edit Instances" button. The main table has columns: Instance ID, Name, Availability Zone, Status, and Actions. It lists two instances:

Instance ID	Name	Availability Zone	Status	Actions
i-f2bf754b	oxclo02-asi	eu-west-1a	InService ⓘ	Remove from Load Balancer
i-fbd8ef42	oxclo02-asi	eu-west-1b	OutOfService ⓘ	Remove from Load Balancer

48. Once the siege has ended, you should see the spare instance removed:

The screenshot shows the AWS Auto Scaling Group Activity History page. At the top, it says "Auto Scaling Group: oxclo03-asg". Below that is a navigation bar with tabs: Details, Activity History (which is selected), Scaling Policies, Instances, Notifications, and Tags. A filter bar at the top says "Filter: Any Status" and has a search input "Filter scaling history...". The main table has columns: Status, Description, and Start Time. It lists three events:

Status	Description	Start Time
Successful	Terminating EC2 instance: i-fbd8ef42	2015 November 17 14:24:24 UTC
Successful	Launching a new EC2 instance: i-fbd8ef42	2015 November 17 14:16:55 UTC
Successful	Launching a new EC2 instance: i-f2bf754b	2015 November 17 13:24:22 UTC

49. Once you have finished, **delete** the autoscaling group and **terminate** the siege instance. Make sure that you have no further instances running in your name!

50. You have completed the exercise. Well done.

51. As an **extension**, come up with a plan to secure the cloud instances better through improved configuration of the security groups. Identify which systems need to talk to which, and then suggest a set of security groups that would allow this.

