

Exercise 3

Configuring a Load Balancer, Autoscaling and 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 *wrk* command

Software Requirements

Browser and AWS account, previous configuration from Exercise 2

Part A: Starting an instance to do a stress test from

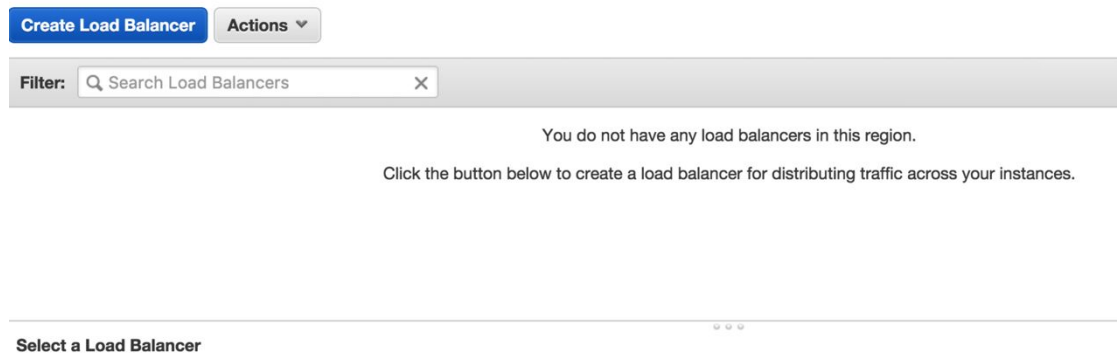
1. We are going to create a new instance in the same subnet to stress test the servers from. We could do it from our desktops, but we will take out network delays if we can do it within the Amazon EC2 network.
2. Because this takes a bit of time to start, we will get this running first and then come back and use it later.
3. Using the EC2 Launch wizard like before, start a new instance with the following settings:
 - a. **Ubuntu Server 18.04 LTS (HVM)**
 - b. **t2.medium** (we want a beefier machine to be able to drive our nodes hard)
 - c. User Data: please cut and paste from <https://freo.me/wrk-userdata>
 - d. This simply installs the latest version of *wrk* and sets correct parameters for the OS to handle this

(<https://github.com/wg/wrk>)
 - e. Tag Name: *userid-wrk*
 - f. Security Group: *node-security-group*

- g. Your existing SSH Key

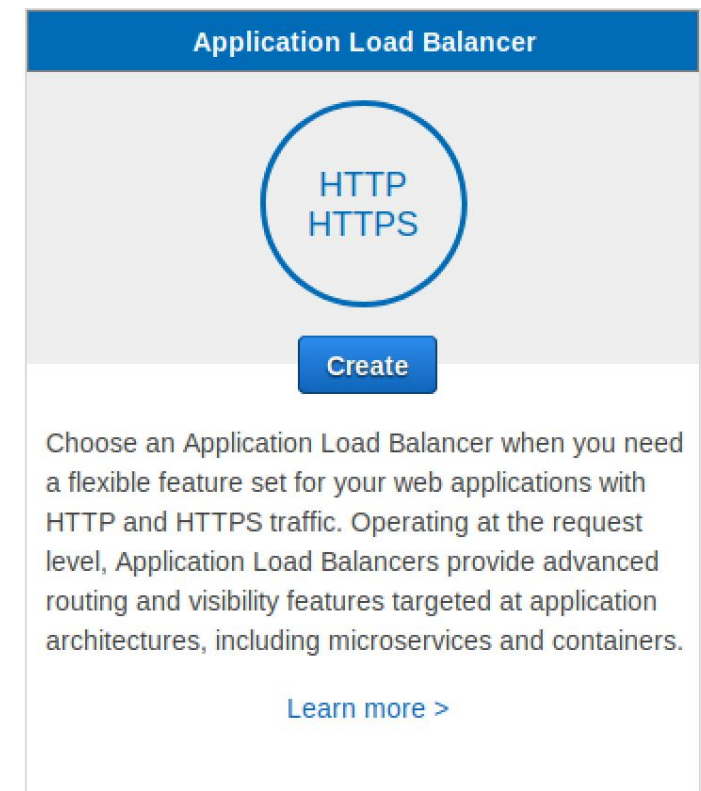
Part B: Setting up a Load Balancer and ELB Auto Scale Group

4. Go to the AWS Console and then the EC2 Console.
5. 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).



6. Click **Create Load Balancer**
- 7.

Choose **Application Load Balancer**



8. Set **Name** to *userid-elb* (e.g. *oxclo02-elb*), and leave the other fields the same.

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 1: Configure Load Balancer

Name ⓘ oxclo01-elb

Scheme ⓘ ☒ internet-facing ☐ internal

IP address type ⓘ ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

| Load Balancer Protocol | Load Balancer Port |
|------------------------|--------------------|
| HTTP | 80 |

Add listener

9. Click on all three of the **Availability Zones**:

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC ⓘ vpc-42fb9527 (172.31.0.0/16) (default) ▼

Availability Zones

☒ eu-west-1a subnet-36f9a653 ▼
IPv4 address ⓘ Assigned by AWS

☒ eu-west-1b subnet-79bbfc0e ▼
IPv4 address ⓘ Assigned by AWS

☒ eu-west-1c subnet-52d8410b ▼
IPv4 address ⓘ Assigned by AWS

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

Ignore the warning!

11. Click **Next: Configure Security Groups**

12. Select **Create a New Security Group**

13. Give it the name *userid-elb-sg* (e.g. oxclo02-elb-sg)

14. Make sure the rule says:

Custom TCP 80 Anywhere 0.0.0.0/0

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name: oxclo01-elb-sg

Description: load-balancer-wizard-1 created on 2019-07-01T08:03:18.750+01:00

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ |
|-------------|------------|--------------|---------------------------|
| Custom TCF▼ | TCP | 80 | Anywhere▼ 0.0.0.0/0, ::/0 |

Add Rule

15.

Click **Next: Configure Routing**

Make sure it says “New Target Group”, and then use

userid-target (e.g. oxclo01-target)

Choose **instance**

Change the port to **8080**

Leave the rest as-is:

Create target group

Target group name ⓘ

oxclo01-target

Target type

☒ Instance

☐ IP

☐ Lambda function

Protocol ⓘ

HTTP ▼

Port ⓘ

8080

VPC ⓘ

vpc-42fb9527 (172.31.0.0/16) (My Default V▼

Health check settings

Protocol ⓘ

HTTP ▼

Path ⓘ

/

16. Ignore the warning and click: **Next: Register Targets**

17. Don't add any instances yet. Just click **Review**

18. It should look like:

Step 6: Review

Please review the load balancer details before continuing

▼ Load balancer

Name

oxclo01-elb

Scheme

internet-facing

Listeners

Port:80 - Protocol:HTTP

IP address type

ipv4

VPC

vpc-42fb9527

Subnets

subnet-36f9a653, subnet-79bbfc0e, subnet-52d8410b

Tags

▼ Security groups

Security groups

oxclo01-elb-sg1

▼ Routing

Target group

New target group

Target group name

oxclo01-target

Port

8080

Target type

instance

Protocol

HTTP

Health check protocol

HTTP

Path

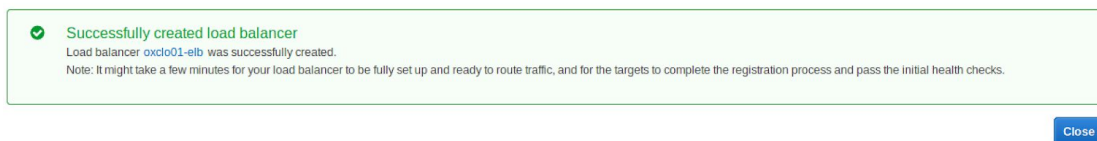
/

Health check port

traffic port

19. Click **Create**

20. You should see something like:



Now let's create our AutoScaling Group

21. Go back to creating an Auto Scale Group like last time. (**Auto Scaling Groups -> Create Auto Scaling Group**)

22. Create from an existing Launch Configuration and choose your own launch config that you previously created. (Scroll down)

Create Auto Scaling Group

You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)

[Create a new launch configuration](#)

☐ Launch Template New

Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)

[Create new launch template](#)

| Filter launch configurations... | | | |
|--|-----------------------|---------------|------------|
| Name | AMI ID | Instance Type | Spot Price |
| <input checked="" type="checkbox"/> oxclo01-lc | ami-08d658f84a6d84a80 | t2.micro | |

Click **Next Step**

23. On the following screen:

- Give it a group name of *userid-asg* (e.g. oxclo01-asg)
- Add one or more subnets as before
- Expand the **Advanced Details**
- Click **Load Balancing**
- Select your own **Target Group**
- Change the Health Check type to ELB
- Leave the Grace period as 300 seconds

It should look like this:

Create Auto Scaling Group

Group name ⓘ oxclo01-asg

Launch Configuration ⓘ oxclo01-lc

Group size ⓘ Start with 1 instances

Network ⓘ vpc-42fb9527 (172.31.0.0/16) (default) [Create new VPC](#)

Subnet ⓘ

| | |
|--|---|
| subnet-36fa653 (172.31.0.0/20) Default in eu-west-1a | X |
| subnet-79bbf0e (172.31.16.0/20) Default in eu-west-1b | X |
| subnet-52d8410b (172.31.32.0/20) Default in eu-west-1c | X |

[Create new subnet](#)

Each instance in this Auto Scaling group will be assigned a public IP address. ⓘ

▼ Advanced Details

Load Balancing ⓘ ☒ Receive traffic from one or more load balancers [Learn about Elastic Load Balancing](#)

Classic Load Balancers ⓘ

Target Groups ⓘ oxclo01-target X

Health Check Type ⓘ ☒ ELB ☐ EC2

Health Check Grace Period ⓘ 300 seconds

Monitoring ⓘ Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration oxclo01-lc. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency.

24. Click **Next: Configure Scaling Policies**

25. On the following screen

- Select **Use scaling policies....**
- Change it to support scaling between **1** and **4** instances
- Set the target CPU value to be **30%**
(we want this low enough to see scaling happen)

26. It should look like:

Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy chooses to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size.

☐ Keep this group at its initial size

☒ Use scaling policies to adjust the capacity of this group

Scale between and instances. These will be the minimum and maximum size of your group.

Scale Group Size

Name:

Metric type:

Target value:

Instances need: seconds to warm up after scaling

Disable scale-in: ☐

[Scale the Auto Scaling group using step or simple scaling policies](#)

27. Click **Next: Configure Notifications**

If you want to configure notifications you can, but you have to figure it out yourself 😊

28. Click **Next: Configure Tags**

29. Add the tag: Name / *userid*-autoscaled

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

A tag consists of a case sensitive key-value pair that you can use to identify your group. For example, you could define a tag with Key = Environment and Value = Production. You can optionally choose to apply these tags to instances in the group.

| Key | Value | Tag New Instances |
|------|--------------------|-------------------------------------|
| Name | oxclo01-autoscaled | <input checked="" type="checkbox"/> |

49 remaining

30. Click **Review**

31. Click **Create Autoscaling Group**

32. Go and see if an instance is being started. You should see something like:

| | | | | | | |
|--|--------------------|---------------------|----------|------------|---------|--------------|
| | oxclo01-autoscaled | i-0c15bf8433118511a | t2.micro | eu-west-1c | pending | Initializing |
|--|--------------------|---------------------|----------|------------|---------|--------------|

33. We need to give the new instance 300 seconds (5 minutes) before it is deemed healthy. This was a setting on a previous screen.

34. You can check the status of the instance in the target group.

Create target group

Actions

Filter by tags and attributes or search by keyword

1 to 1 of

| Name | Port | Protocol | Target type | Load Balanc | VPC ID | Monitoring |
|----------------|------|----------|-------------|-------------|--------------|------------|
| oxclo01-target | 80 | HTTP | instance | oxclo01-elb | vpc-42fb9527 | |

Description

Targets

Health checks

Monitoring

Tags

The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.

Edit

None of these Availability Zones contains a healthy target. Requests are being routed to all targets.

Registered targets

| Instance ID | Name | Port | Availability Zone | Status |
|---------------------|--------------------|------|-------------------|-----------|
| i-0c15bf8433118511a | oxclo01-autoscaled | 80 | eu-west-1c | unhealthy |

Wait until the instance is healthy before the next step.

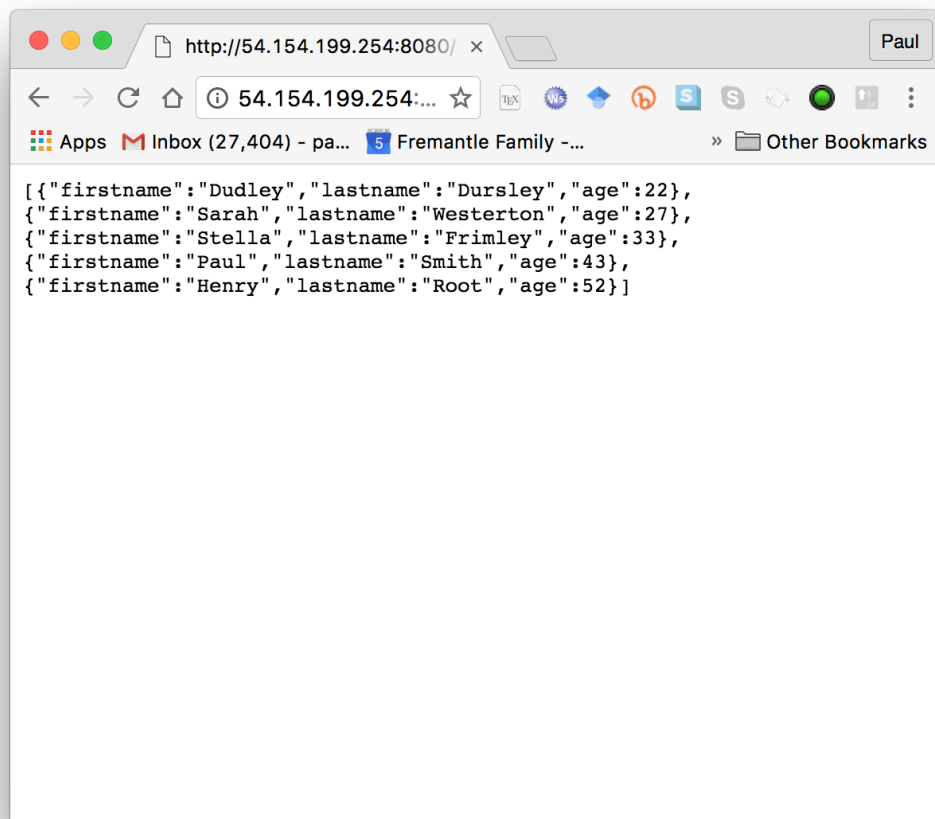
PART C – Stress testing

35. Navigate to view your ELB's dashboard page. You can find the DNS address of your ELB this way:

Basic Configuration

| | |
|----------|---|
| Name | oxclo01-elb |
| ARN | arn:aws:elasticloadbalancing:eu-west-1:775785745523:loadbalancer/app/oxclo01-elb/70c5afdf29a0437d  |
| DNS name | oxclo01-elb-423576883.eu-west-1.elb.amazonaws.com  (A Record) |

36. 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, because the load balancer listens on 80.

37. Remember the “wrk” instance you created earlier? Check the instance is running in the EC2 dashboard and then SSH into the instance as in Exercise 1.
38. Accept the fingerprint as before.

39. In the SSH session type: wrk

```
ubuntu@ip-172-31-44-70:~$ wrk
Usage: wrk <options> <url>
Options:
  -c, --connections <N>  Connections to keep open
  -d, --duration      <T>  Duration of test
  -t, --threads       <N>  Number of threads to use

  -s, --script        <S>  Load Lua script file
  -H, --header        <H>  Add header to request
      --latency              Print latency statistics
      --timeout             <T>  Socket/request timeout
  -v, --version              Print version details

  Numeric arguments may include a SI unit (1k, 1M, 1G)
  Time arguments may include a time unit (2s, 2m, 2h)
ubuntu@ip-172-31-44-70:~$
```

40. We want to call our load balancer with 100 concurrent connections, two threads, for around 15 minutes (enough time to see scaling).

e.g:

```
wrk -c 100 -t 2 -d 15m http://clo01-elb-1355165567.eu-west-1.elb.amazonaws.com
```

But with your ELB address

41. You should see something like:

```
Running 15m test @ http://oxclo01-elb-1355165567.eu-west-1.elb.amazonaws.com
2 threads and 100 connections
```

42. This is basically hitting your Load Balancer with a significant number of hits for 15 minutes. This should be long enough to see the behaviour we want.

43. Unless we run out of network bandwidth, this should push the instances's average CPU above 30% and cause the Scaling Group to start another server. Ideally it will push it to 99% and we will see at least two instances created.

44. Assuming all is well you should see a new instance spawned in a few minutes when there is enough CPU history to capture.

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

Auto Scaling Group: oxclo01-asg

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks

Filter: Any Status 1 to 2 of 2 History Items

| Status | Description | Start Time | End Time |
|-----------------------------|---|----------------------------|----------------------------|
| Waiting for instance warmup | Launching a new EC2 instance: i-0165443620e008a8f | 2019 July 1 16:07:34 UTC+1 | |
| Successful | Launching a new EC2 instance: i-00b9d0c7669d87e11 | 2019 July 1 15:39:46 UTC+1 | 2019 July 1 15:40:20 UTC+1 |

46. Once you have seen one or more new instances started, you can end the wrk if you like, by hitting Ctrl-C in the command line window:

```
Thread Stats Avg Stdev Max +/- Stdev
Latency 34.03ms 2.80ms 271.51ms 86.69%
Req/Sec 1.48k 60.62 1.72k 80.03%
339319 requests in 1.92m, 156.30MB read
Requests/sec: 2938.99
Transfer/sec: 1.35MB
```

47. Start wrk up again with the same parameters. Wait until the new server is in service and then stop/start wrk, so you get some new data with more instances running.

48. You should see the request count goes up a lot once the new server(s) are in service, compared to the data with only one server running.

In my tests I saw around 3000 tps from one server, 6000 from 2 and 8500 from 4. Why might this dropoff in scaling happen?

49. If you leave wrk running you may see even more servers launched over time.

50. Once the stress test has ended, you should see the spare instance removed after enough time.

Auto Scaling Group: oxclo03-asg

Details Activity History Scaling Policies Instances Notifications Tags

Filter: Any Status

| | 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 |

51. Once you have finished:

- Delete the autoscaling group
- Delete the load balancer
- Delete the target.
- Terminate the wrk instance.
- Make sure that you have no further instances running in your name!

52. You have completed the exercise. Well done.

53. 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.