

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
<div><div></div> AWS Introductions</div>
<div><div></div> AWS Sign up</div>
<div><div></div> AWS Billing Alert</div>
<div><div></div> AWS EC2</div>

Elastic Load Balancer

Contents

- [Overview](#)
- [Tutorial](#)
- [Exercises](#)

Overview

Elastic Load Balancers (ELB's) evenly distribute traffic for our applications from a single point of ingress.

Tutorial

This task will guide us through creating and configuring a Classic Load Balancer. We want to create a system where the users can only access the application through the ELB.

Your first task is to create 2 EC2 instances in the same region, this guide will not take you through that. Please apply the same Security Group to both of these instances, which should allow communication over port 80 from all sources.

When you have created these machines run the following commands on each one. We are installing and setting up a **Nginx** server on each machine, Nginx is a simple proxy service that has many interesting uses, for the purpose of this task however we are using its default functionality of serving out an index.html file. We will be changing the index.html that each machine serves out so that we can differentiate which of the 2 machines are serving us our content.

```
sudo apt update

sudo apt install nginx -y

sudo systemctl status nginx

sudo systemctl start nginx

cd /var/www/html/

sudo nano index.nginx-debian.html
```

For the last step you need to edit one instance so that it displays the number 1 in the `<h1>` tags, and for the other the number 2. An example of how one of the final .html files should look can be seen below.

<div><div></div><div>Key Pairs</div></div>
<div><div></div><div>S3 Introduction</div></div>
<div><div></div><div>S3 Storage Options</div></div>
<div><div></div><div>AWS S3 bucket creation</div></div>
<div><div></div><div>S3 Bucket Policies</div></div>
<div><div></div><div>S3 Lifecycle Policies</div></div>
<div><div></div><div>S3 File Upload</div></div>
<div><div></div><div>S3 AWS-CLI Commands</div></div>
<div><div></div><div>S3 Glacier</div></div>
<div><div></div><div>Elastic Beanstalk Introduction</div></div>
<div><div></div><div>AWS IAM Intro</div></div>
<div><div></div><div>AWS IAM User Overview</div></div>
<div><div></div><div>AWS IAM Users</div></div>
<div><div></div><div>AWS IAM Policies</div></div>
<div><div></div><div>AWS Programmatic Access</div></div>
<div><div></div><div>AWS IAM Role CLI</div></div>
<div><div></div><div>AWS RDS</div></div>
<div><div></div><div>AWS Auto-Scaling Group CLI</div></div>
<div><div></div><div>Elastic Load Balancer</div></div>
<div><div></div><div>AWS Intermediate</div></div>
<div><div></div><div>Linux</div></div>
<div><div></div><div>DevOps</div></div>
<div><div></div><div>Jenkins Introduction</div></div>
<div><div></div><div>Jenkins Pipeline</div></div>
<div><div></div><div>Markdown</div></div>
<div><div></div><div>IDE Cheatsheet</div></div>

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>2</h1>
</body>
</html>
```

Navigate to the EC2 Console.

In the menu on the left hand side, click the **Load Balancers**.

Click **Create Load Balancer**.

As you may have seen you can select from 3 different kinds of ELBs, for the purpose of this task we will be using a **Classic Load Balancer**, click the blue **Create** in this box.

Create a name for your Load Balancer, for example:

```
classic-test-elb
```

Hit the **Enable advanced VPC configuration** checkbox.

You now need to select all the subnets that are available to you, based upon the region that you are building your ELB in. Even if you do not currently have EC2 instances running in these subnets, you need to select it.

Click the **Next select security groups** button.

create new security group, give it a name, for example.

```
elb-sg
```

next - configure security settings

next - health check

make the ping path /

next add instances, select the 2 instances that you created earlier in the Task.

next add tags

review and create

Click the blue **Create** button.

Click the blue **Close** button.

ensure that both instances are correctly attached to the Load Balancer and their status is InService, if it switches over to OutOfService check the question mark too see why, it may be because you have not correctly set the permissions in the Security Groups.

When both of the instances are InService select the DNS name from the Description tab of the Load Balancer. It will look something like this;

```
classic-test-elb-844687033.eu-west-1.elb.amazonaws.com
```

Paste this into browser, you should see either the number '1' or '2' displayed. This is because the Load Balancer, which you have accessed using the DNS name, has forwarded your request to one of the instances you have running.

Refresh the page a few times, as you refresh the page you should see that the number on the screen is changing, this is because the Load Balancer is sending the request to either of the 2 servers.

The above is great, however we can still directly access these instances by using its own individual IP address, this is because the Security Group for the instances allows communication over port 80 from **all sources**. The next step is to change the Security Group for these instances so that they only allow communication from the ELB.

Navigate to the ELB dashboard, with the ELB we created selected, go to the Description tab:

And scroll down until you see Security:

Copy the unique identifier of the ELB's Security Group, it will look like this;

sg-09b2dfde7ca8c32b4

Now click on Security Groups, in the menu on the left.

Select the Security Group that your 2 instances belong to that you created at the beginning of this task.

Click on Edit to change the current rules associated with this SG.

In the source column for communication over port 80, select custom and then replace the source with the ELB's security group identifier that you just copied.

Click Save.

Now let's test the rule. Click on Load Balancers to be able to copy the URL for the LB again.

Copy the DNS Name address and past it into a new tab in your browser and check your results. You should see that the ELB still manages traffic to the 2 instances.

However, if you try to copy and past the public IP address of the specific instance you are accessing, and paste it into your browser, you shouldn't be able to access the app - the connection will timeout.

Our final step in this Task is to stop one of the instances to see how the ELB responds, do this now.

When you access the DNS name of the ELB now (via your browser) you will see that you are always directed to the one surviving instance.

If you restart the stopped instance, traffic is once again balanced.

Exercises

There are no exercises for this module.