

ICT & Infra S3 S/NO week 11: AWS Elastic load balancing setup

Date: Nov 2022
Version 1.0

Class: CB01

Student number: 4961854

Student name: Heiko Morales

Introduction

S/NO: Following these exercises you will learn how to:

- Create the Web Tier / the App Tier / Database Tier
- Use Network Load Balancer or Application Load Balancer for Web Tier
- Configure Sticky sessions / Idle timeouts
- Configure Path-based routing with load balancer

How to deliver your assignments?

Fill in this document with required information. Answer questions and upload the document to Canvas at most one week after the assignment is given.

Assignment 1: Create initial AWS Web/App/Database Tiers of the example web service

- Follow the demo from the lecture. Create necessary entities / configurations in AWS with provided Powershell scripts.
- Use “SNO_week11_scripts.zip” scripts

Provide screenshots and descriptions of the steps above

Assignment 2: Use Network Load Balancer for the App Tier

- Configure the Network Load Balancer for the App Tier
- Explain what are pro's/co's of the Network Load Balancer compared to Application Load Balancer for the App Tier.
- Where would you use Application and/or Network Load Balancer in your case-study project? And why?

Provide screenshots and descriptions of the steps above

Assignment 3: Configure HTTPS (self-signed) for the Web Tier

- Enrich the Web Tier (load balancer, target groups) to support HTTPS(self-signed), port 443.
- (Bonus) Brainstorm with your case-study members, how can you upgrade the HTTPS connection to web instances with domain-validated certificate.

In our case the certificates are obtained from letsencrypt and therefore we can associate a server to that certificate to obtain SSL encryption and have HTTPS.

Provide screenshots and descriptions of the steps above

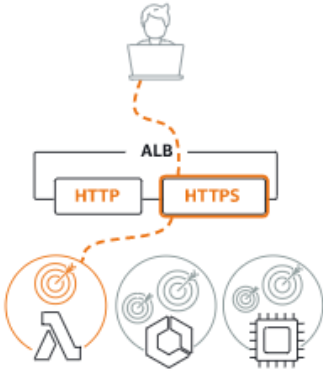
Assignment 4: Path-based routing via Application Load Balancer

- Brainstorm with your case-study members, how can you benefit from the path-based routing in your case-study project.
- In our case we do not use it since we are satisfied with the domains and subdomains we use with nginx but it is perfectly possible to replace the nginx service with the path-based one.**

First we will create an HTTP and HTTPS load balancer.

Load balancer types

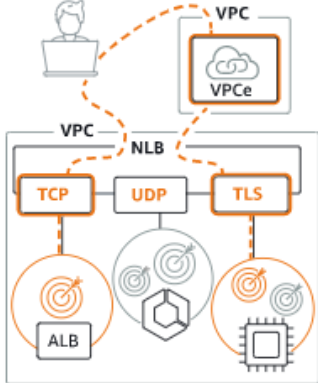
Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create


Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

► Classic Load Balancer - previous generation

We configure the basic configuration by selecting the internet-facing option.

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

☒ **Internet-facing**

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ **Internal**

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ **IPv4**

Recommended for internal load balancers.

☐ **Dualstack**

Includes IPv4 and IPv6 addresses.

We configure the network options.

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

vpc

vpc-025b41945b783a001
IPv4: 172.31.0.0/16



Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ **eu-central-1a (euc1-az2)**

Subnet

subnet-00f1f8cbdf712f9f8

IPv4 settings

Assigned by AWS

We configure the listeners which are the port that the balancer must listen to to redirect the traffic.

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol

HTTP ▼

Port

: 80

1-65535

Default action

Forward to

tgDemo

Target type: Instance, IPv4

HTTP ▼

⌂

Create target group [↗](#)

Listener tags - optional

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

We edit the inbound rules to the ports we require

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
-	Custom TCP ▼	TCP	3000	Custom ▼ <input type="text" value="Q"/>		Delete

Add rule

sg-0119ff6d6f0bea0e4

✕

Cancel

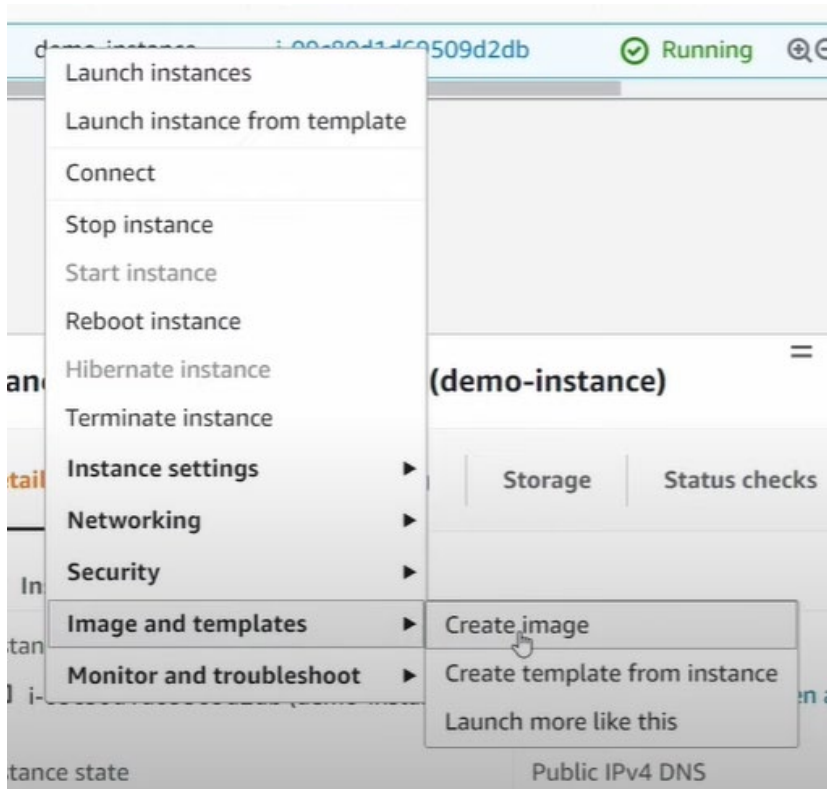
Preview changes

Save rules

And as we can see we have already created the load balancer.

<input type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
<input type="checkbox"/>	demo-lb	demo-lb-1646116872.eu-c...	⌂ Provisioning	vpc-025b41945b783a001	2 Availability Zones

Then we go to the instances and generate an image so that we can later deploy it automatically.



Once created, it can be found in the AMIS section.

Amazon Machine Images (AMIs) (1/2) [Info](#)

Owned by me < 1 > ⚙

	Name	AMI ID	Source	Owner	Visibility
<input type="checkbox"/>	-	ami-0e66699a4ae37d0a0	m... 807013657668/moniserv	807013657668	Private
<input checked="" type="checkbox"/>	webdemo	ami-073152177b74cbf96	D... 807013657668/DBserver	807013657668	Private

later on we create an auto scaling group by selecting the newly created ami

Q Search our full catalog including 1000s of application and OS images

Recents

My AMIs

Quick Start

☒ Owned by me

☐ Shared with me

Q

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

webdemo

ami-073152177b74cbf96

2022-11-16T10:07:00.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

▼

Description

tmp image

Architecture

AMI ID

x86_64

ami-073152177b74cbf96

We configure the instance type and select the key pair

▼ Instance type Info

Advanced

Instance type

t2.micro

Family: t2 1 vCPU 1 GiB Memory

On-Demand Linux pricing: 0.0134 USD per Hour

On-Demand Windows pricing: 0.018 USD per Hour

Free tier eligible

▼

Compare instance types

▼ Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

webserver

▼

↻

Create new key pair

Finally, we define the group size of our instances.

Group size - *optional* [Info](#)

Specify the size of the Auto Scaling group by changing the desired capacity limits. Your desired capacity must be within the limit range

Desired capacity

Minimum capacity

Maximum capacity



demo-asg, 1 Scaling policy created successfully

EC2 > Auto Scaling groups > demo-asg

DetailsActivityAutomatic scalingInstance managementMonitoring

Group details

Desired capacity	2	Auto Scaling group name	demo-asg
Minimum capacity	1	Date created	Sun Jul 25 2021 09:48:43 GMT-0300 (Uruguay Standard Time)
Maximum capacity	4	Amazon Resource Name (ARN)	arn:aws:autoscaling:us-east-1:025284601468:autoScalingGroup:a50d0

Once created, we can see how it starts to act in the activity history.

Activity history (2)

Filter activity history

Status	Description	Cause
PreInService	Launching a new EC2 instance: i-09ff179c15a39bc9f	At 2021-07-25T12:48:43Z a user request from 0 to 2. At 2021-07-25T12:49:10Z ar desired and actual capacity, increasing th
PreInService	Launching a new EC2 instance: i-02955a1ef154a9559	At 2021-07-25T12:48:43Z a user request from 0 to 2. At 2021-07-25T12:49:10Z ar desired and actual capacity, increasing th

As we can see, two instances are being created that our auto scaling load balancer needs.

Instances (3) Info							
<input type="text" value="Filter instances"/>		Refresh Connect		Instance state ▼	Actions ▼	Launch instances	▼
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Alarm
<input type="checkbox"/>	demo-instance	i-09c80d1d69509d2db	✓ Running 🔍	t2.micro	✓ 2/2 checks passed	•	US
<input type="checkbox"/>	-	i-02955a1ef154a9559	✓ Running 🔍	t2.micro	⌚ Initializing	•	US
<input type="checkbox"/>	-	i-09ff179c15a39bc9f	✓ Running 🔍	t2.micro	⌚ Initializing	•	US

demo-tg

[🔗](#) arn:aws:elasticloadbalancing:us-east-1:025284601468:targetgroup/demo-tg/d056732b4571fc52

Details

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 3000	HTTP1	vpc-l
Load balancer	demo-lb 🔗		

Total targets	Healthy	Unhealthy	Unused	Initial
2	✓ 2	✗ 0	⋮ 0	⌚ 0

In this way whenever a request comes in through the load balancer the auto scaling will detect if there is too much load and will create at most 4 instances at the same time to give service.