



**COLLEGE OF COMPUTING TECHNOLOGY - DUBLIN**  
**BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY**

**CLOUD VIRTUALIZATION FRAMEWORKS**

**Assignment 1**  
**Designing and Implementing a Cloud Solution**

Adelo Vieira  
Student Number: 2017279

Lecturer: Greg South

May 11, 2020

# Contents

<b>1</b>	<b>Technical solution</b>	<b>1</b>
1.1	Creating a VPC . . . . .	1
1.2	Subnet architecture . . . . .	2
1.2.1	Availability zone 1 . . . . .	2
1.2.1.1	Public subnet 1 . . . . .	2
1.2.1.2	Private Web-Tiers subnet 1 . . . . .	3
1.2.1.3	Private App-Tiers subnet 1 . . . . .	4
1.2.2	Replication of the subnet architecture in a second AZ . . . . .	5
1.3	Defining Servers Security Groups . . . . .	6
1.4	Launching a Web and App tiers instances and configuring the Web application . . . . .	6
1.5	Creating Amazon Machine Images for the Web-Tier and App-Tier instances . . . . .	7
1.6	Load Balancing . . . . .	8
1.7	Auto Scaling . . . . .	8
1.8	Database tier . . . . .	9
1.9	Storing Web-Accessible Content in Amazon S3 . . . . .	9
1.10	Caching with Amazon CloudFront . . . . .	9
<b>2</b>	<b>Architecture diagram</b>	<b>11</b>
<b>3</b>	<b>Implementation of a proof of concept solution</b>	<b>13</b>
3.1	Creating a VPC . . . . .	14
3.1.1	Configuring the VPC's ACL . . . . .	15
3.2	Subnet architecture . . . . .	16
3.2.1	Adding a target to the IGW in the public-named subnets to make them really publics . . . . .	18
3.3	Creating a NAT Gateway in each Public subnet . . . . .	20
3.4	Creating 2 new private route tables . . . . .	20
3.5	Defining Servers Security Groups . . . . .	22

3.6	Launching a Web and App tiers instances and configuring the Web application . . . . .	23
3.6.1	Launching the instances . . . . .	23
3.6.2	Accessing the instance using SSH and Installing the necessary packages . . . . .	25
3.6.3	Uploading the web app and testing it . . . . .	27
3.7	Creating Amazon Machine Images for the Web-Tier and App-Tier instances . . . . .	28
3.8	Load Balancing . . . . .	30
3.9	Auto Scaling . . . . .	32
3.10	Storing Web-Accessible Content in Amazon S3 . . . . .	33
3.11	Caching with Amazon CloudFront . . . . .	35
3.12	Cross-region disaster recovery . . . . .	36
3.12.0.1	Data replication . . . . .	36
<b>4</b>	<b>Proposed solution report</b>	<b>37</b>
	<b>Bibliography</b>	<b>40</b>

# List of Figures

- 1.1 ACL for the VPC . . . . . 2
- 1.2 Security Groups for the EC2 Instance (Web and App tiers servers) . . . . . 6
- 1.3 Security Groups for the Aurora DB . . . . . 9
  
- 2.1 Architecture diagram . . . . . 12

# Task 1

## Technical solution

### 1.1 Creating a VPC

First, we need to create a Virtual Private Cloud (VPC). A VPC is an Amazon Service that allows us to define a Virtual Network<sup>1</sup> where we can launch AWS resources, such as EC2 Instances.

The VPC will have the following features:

- **Region:** Europe (Ireland)

We have chosen this region because the largest part of our users is in Ireland.

- **Inter-Domain Routing (CIDR) block - IP range:** 10.100.0.0/20

This VPC includes 4,094 IPs between 10.100.0.1 and 10.100.15.254 (with some reserved).

- **Route table:** When you create a VPC, a routing table is automatically generated with a target to the Internet Gateway:

Destination	Target
10.100.0.0/20	Local
0.0.0.0/0	IGW

- **ACL:** We only need HTTP/HTTPS and SSH traffic. The following ACL will allow only inbound and outbound HTTP and HTTPS traffic from any IPv4 address. It also allows inbound SSH since we will need to remotely access

---

<sup>1</sup>A virtual network can be defined as a logically isolated section of the Amazon Web Services cloud

to our Linux instances via SSH. We haven't allowed RDP traffic because our environment will be based only on Linux instance.

### ACL for the VPS

Inbound			
Type	Port	Source	Allow/Deny
HTTP	80	0.0.0.0/0	Allow
HTTPS	433	0.0.0.0/0	Allow
SSH	22	192.0.1.0/24*	Allow

Outbound			
Type	Port	Destination	Allow/Deny
HTTP	80	0.0.0.0/0	Allow
HTTPS	433	0.0.0.0/0	Allow

\* Network from which we will access our VPC

Figure 1.1: ACL for the VPC. Notice that the source allows for SSH need to match the network from which we will access our VPC.

## 1.2 Subnet architecture

Now, in our VPC, we will be able to define our subnet architecture for high availability. Essentially, we will define 3 subnets that will be used to host our web and app servers. These 3 subnets will be created in a first availability zone and then replicated to another availability zone.

### 1.2.1 Availability zone 1

#### 1.2.1.1 Public subnet 1

We need a subnet that has a direct connection to the Internet gateway (a public subnet). This is necessary because our servers will need Internet access for patching and updates.

It is important to notice that we won't launch Instance in this subnet or any other public subnet. Instead, we will launch a **NAT Gateway** that will connect our Instance located in a **private subnet** with the Internet Gateway.

The **NAT Gateway** will be provisioned with an Elastic IP Address (EIP). “An Elastic IP address will remain unchanged over the life of the NAT Gateway”. [[AWS Academy \(2019a\)](#)]

Avoiding launching Instance in a public subnet is a security measure that makes sure that our servers are not exposed to direct Internet access.

### Features:

- **IP range (IPv4 CIDR block):** 10.100.1.0/24

This subnet includes 254 IPs<sup>2</sup> between 10.100.0.1 and 10.100.0.254

- **Public Route Table:**

A Route Table defines how traffic flows into and out of a Subnet.

When you create a Subnet, it will be provided with a default Route Table, but this Route Table does not have a connection to your Internet gateway. You will change it to use the Public Route Table.

**Local traffic:** The first entry specifies that traffic destined within the VPC’s CIDR range (10.100.0.0/20) will be routed within the VPC (*local*).

**Internet Gateway:** This specifies that any traffic destined for the Internet (0.0.0.0/0) is routed to the Internet Gateway (*igw*). **This setting makes it a Public Subnet.** That is to say, because we have a Route Table that has a connection to our Internet gateway, then, Public Subnet 1 is now Public, which means it can communicate directly with the Internet.

Destination	Target
10.100.1.0/24	Local
0.0.0.0/0	IGW

- **ACL:** The ACL of the VPN (Figure 1.1)

#### 1.2.1.2 Private Web-Tiers subnet 1

In this private subnet we will deploy our **Web** Tiers Instances.

---

<sup>2</sup>Notice that some of these IPs are reserved and unusable

As we already mentioned, the application will be hosted in private subnets. This way, we improve security since no direct access from the Internet is possible. [AWS Academy (2019a)]

**Features:**

- **Availability zone 1:** eu-west-1
- **IP range (IPv4 CIDR block):** 10.100.3.0/24

Range: 10.100.3.1 - 10.100.3.254

- **Private Route Table:**

This route table will be in charge of sending Internet-bound traffic **through the NAT Gateway**.

Destination	Target
10.100.11.0/24	Local
0.0.0.0/0	NAT-1

- **ACL:** The ACL of the VPN (Figure 1.1)

### 1.2.1.3 Private App-Tiers subnet 1

In this private subnet we will deploy our **App** Tiers Instances.

**Features:**

- **Availability zone 1:** eu-west-1
- **IP range (IPv4 CIDR block):** 10.100.5.0/24

- **Private Route Table:**

This route table will be in charge of sending Internet-bound traffic **through Private App-Tiers subnet 1 > NAT Gateway 1 > Internet Gateway**.

Destination	Target
10.100.12.0/24	Local
0.0.0.0/0	NAT-1



**ACL:** The ACL of the VPN (Figure 1.1)

## 1.2.2 Replication of the subnet architecture in a second AZ

Amazon EC2 instances are not inherently highly available. They need to be provisioned in at least 2 different Availability zones to make our environment highly available. So, if a failure causes the interruption of one of the availability zones, our system will still be available through the second AZ. This is why we will replicate our subnet configuration in a second AZ:

- **Public subnet 2:**

- **IP range (IPv4 CIDR block):** 10.100.2.0/24

- **ACL:** The ACL of the VPN (Figure 1.1)

- **Public Route Table:**

Destination	Target
10.100.0.0/20	Local
0.0.0.0/0	IGW

- **Private Web-Tiers subnet 2:**

- **IP range (IPv4 CIDR block):** 10.100.4.0/24

- **ACL:** The ACL of the VPN (Figure 1.1)

- **Public Route Table:**

Destination	Target
10.100.0.0/20	Local
0.0.0.0/0	NAT-2

- **Private App-Tiers subnet 2:**

- **IP range (IPv4 CIDR block):** 10.100.6.0/24

- **ACL:** The ACL of the VPN (Figure 1.1)
- **Public Route Table:**

Destination	Target
10.100.0.0/20	Local
0.0.0.0/0	NAT-2

### 1.3 Defining Servers Security Groups

In Figure 1.3 we show the Security Groups that will be applied to our App-Tier Instance. The ones will be assigned to the Web-Tier Instances are the same but without the rule that allows outbound traffic on port 3306. This is because only the App-Tier servers will connect to the database.

Security groups for the Web-tier servers				Security groups for the App-tier servers			
<b>Inbound</b>				<b>Inbound</b>			
Type	Port	Source	Allow/Deny	Type	Port	Source	Allow/Deny
HTTP	80	0.0.0.0/0	Allow	HTTP	80	10.100.3.0/24 10.100.4.0/24	Allow
HTTPS	443	0.0.0.0/0	Allow	HTTPS	443	10.100.3.0/24 10.100.4.0/24	Allow
SSH	22	192.0.1.0/24*	Allow	SSH	22	192.0.1.0/24	Allow
<b>Outbound</b>				<b>Outbound</b>			
Type	Port	Destination	Allow/Deny	Type	Port	Destination	Allow/Deny
HTTP	80	10.100.5.0/24 10.100.6.0/24	Allow	TCP	3306	Aurora DB IP	Allow
HTTPS	433	10.100.5.0/24 10.100.6.0/24	Allow	HTTP	80	0.0.0.0/0	Allow
				HTTPS	433	0.0.0.0/0	Allow

\* Network from which we will access our VPC      \* Network from which we will access our VPC

Figure 1.2: Security Groups for the EC2 Instance (Web and App tiers servers). 3306 is the default port used by Amazon Aurora, which is the database we are going to use

### 1.4 Launching a Web and App tiers instances and configuring the Web application

We will launch two EC2 instances. One for the Web-Tier servers and the other for the App-Tier servers.

The client wish to move to Linux based machines. So, we will propose the implementation of Ubuntu Server 18.04 LTS - 64 bits (x86) instances.

To respect the tech details required by the client, the following AWS instance must be used:

- **Web-Tier:**
  - **t2.small:** 1vCPU 2 GiB for \$0.023 per Hour<sup>3</sup>
- **App-Tier:**
  - **a1.larg :** 2vCPU, 4 BiB

However, Medi-Advice also would like to avail of the free tier as much as possible in the proof of concept design. This is why we will use **t2.micro** EC2 instance, which are eligible for the free tier.

## 1.5 Creating Amazon Machine Images for the Web-Tier and App-Tier instances

Now that the application has been configured and is correctly running in our instances, we will create two Amazon Machine Images (AMI), one for the Web-Tier instances and another for the App-Tier instances.

An AMI is a complete copy of the volumes of an instance. This way, when we launch a new instance from these AMI, they will be created containing the same data as the original instance so they will be ready to run the Web App. [[AWS Academy \(2019a\)](#)]

These images will be used later by the Auto Scaling group to create new instances that will be deployed to scale the resources of the web application when needed.

---

<sup>3</sup>This EC2 instance is not eligible for the AWS free usage tier

## 1.6 Load Balancing

A Load Balancer, is an essential component of a highly available design. The Load Balancer distributes traffic across several instances.

A Load Balancer will check if the server is online before sending the request. If a server is down because it reached the maximum number of requests that can manage, the Load Balancer will detect it and send the request to another server.

In our architecture, we will need two Load balancers. The first one will distribute requests coming from the Internet across Web-Tier instances. A second one will be configured to distribute requests coming from the Web-Tiers instance across App-Tiers instances.

## 1.7 Auto Scaling

One of the most important features of cloud architecture is the capability to automatically scale resources based on demand.

This feature, that is commonly referred to as Scalability, avoids downtimes and thus provides high availability by increasing resources when the system reaches its maximum capability. Also, it provides cost-efficiency by reducing resources when the capabilities provisioned are no longer needed.

Two auto scaling groups will be defined in our architecture. The first one will be in charge of provisioning or terminating Web-Tiers instances based on the demand. The second one will manage App-Tiers instances.

The auto-scaling groups will be defined so the minimum number of servers in each tier will be 2. Because Medi-Advice is expecting to double the number of users due to the coronavirus crisis, the auto-scaling group will automatically provision the necessary resources to manage the increasing demand, but it will scale back when the situation returns to 'normal'. This way, the company will only pay for the resources that are needed to manage the demand.

## 1.8 Database tier

**Security groups for the DB**

Inbound			
Type	Port	Source	Allow/Deny
TCP	3306	10.100.5.0/24 10.100.6.0/24	Allow

Figure 1.3: Security Groups for the Aurora DB

## 1.9 Storing Web-Accessible Content in Amazon S3

Storing static large files in an EC2 instance is not a good practice. It actually generates critical issues that must be avoided when designing a cloud infrastructure.

If we are building a highly available environment, we need common storage for all the instances of our infrastructure since replicating the data across all the instances is not a good strategy at all.

Amazon S3 is an object storage service that provides high availability in a region. The data in an S3 bucket is automatically replicated across 3 availability zones. [AWS Academy (2019c)]

Therefore, a good practice is to store files or content that is not going to change in an Amazon S3 bucket. Another important point is that the content in an S3 bucket can be easily cached using Amazon Cloud CloudFront. We will talk more about it in the next section.

In our design, we will create an Amazon S3 in our primary region and another one in the secondary region. We will of course implement cross-region data replication between both buckets.

## 1.10 Caching with Amazon CloudFront

Caching refers to temporary store data in an intermediate location between the source and the final user so it can easily be accessed from a closer location.

Caching is a critical element that should be implemented especially if our site is going to be used for people around the world.

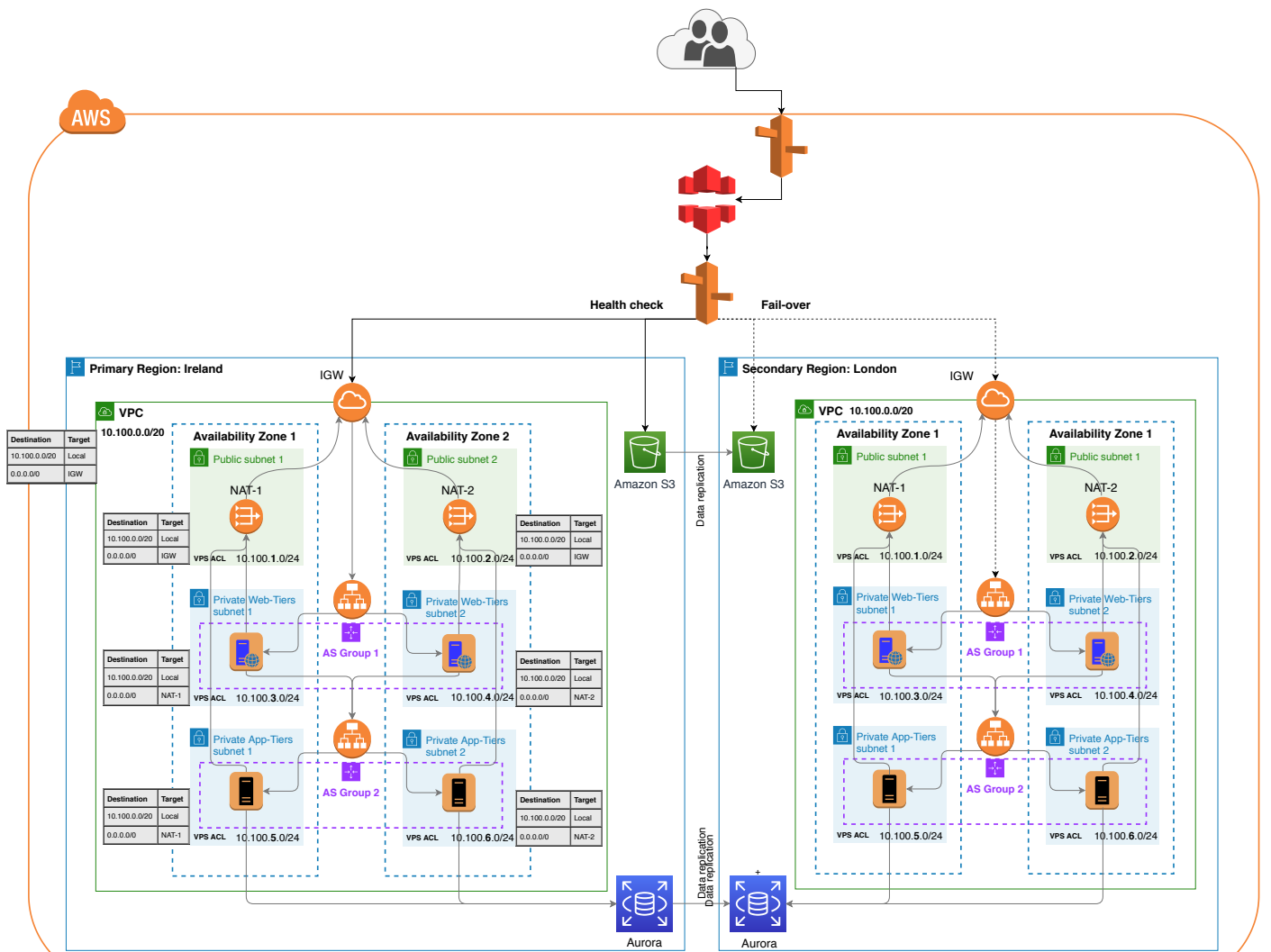
Amazon CloudFront is a caching service that distributes content worldwide.

In the case of Medi-Advice, they are expecting to have a big demand in both Ireland and in the US. Due to the remoteness of these main locations, the implementation of Caching become fundamental to the cloud infrastructure.

Because Medi-Advice is based in Ireland, it is logical to think the largest part of the customer lives in Ireland. This is why we have chosen Ireland as our primary location. This way, we are looking to provide optimal services by reducing Latency in our primary location.

## Task 2

# Architecture diagram



**ACL for the VPS and Subnets**

Inbound			
Type	Port	Source	Allow/Deny
HTTP	80	0.0.0.0/0	Allow
HTTPS	443	0.0.0.0/0	Allow
SSH	22	192.0.1.0/24*	Allow

Outbound			
Type	Port	Destination	Allow/Deny
HTTP	80	0.0.0.0/0	Allow
HTTPS	443	0.0.0.0/0	Allow

\* Network from which we will access our VPC

**Security groups for the Web-tier servers**

Inbound			
Type	Port	Source	Allow/Deny
HTTP	80	0.0.0.0/0	Allow
HTTPS	443	0.0.0.0/0	Allow
SSH	22	192.0.1.0/24*	Allow

Outbound			
Type	Port	Destination	Allow/Deny
HTTP	80	10.100.5.0/24 10.100.6.0/24	Allow
HTTPS	443	10.100.5.0/24 10.100.6.0/24	Allow

\* Network from which we will access our VPC

**Security groups for the App-tier servers**

Inbound			
Type	Port	Source	Allow/Deny
HTTP	80	10.100.3.0/24 10.100.4.0/24	Allow
HTTPS	443	10.100.3.0/24 10.100.4.0/24	Allow
SSH	22	192.0.1.0/24	Allow

Outbound			
Type	Port	Destination	Allow/Deny
TCP	3306	Aurora DB IP	Allow
HTTP	80	0.0.0.0/0	Allow
HTTPS	443	0.0.0.0/0	Allow

\* Network from which we will access our VPC

**ACL for the DB**

Inbound			
Type	Port	Source	Allow/Deny
TCP	3306	10.100.5.0/24 10.100.6.0/24	Allow

Figure 2.1: Architecture diagram



## Task 3

# Implementation of a proof of concept solution

We were going to implement the regions configuration we explained in the first task. However, because of the restriction of the AWS Starter Account we haven't been able to use European regions but only US regions.

## 3.1 Creating a VPC

The screenshot shows the AWS VPC Management Console in a Mozilla Firefox browser. The page title is "VPC Management Console - Mozilla Firefox". The URL is "https://console.aws.amazon.com/vpc/home?region=us-east-1". The page content is titled "Step 1: Select a VPC Configuration".

On the left, there are four VPC configuration options:

- VPC with a Single Public Subnet
- VPC with Public and Private Subnets** (selected)
- VPC with Public and Private Subnets and Hardware VPN Access
- VPC with a Private Subnet Only and Hardware VPN Access

The selected option, "VPC with Public and Private Subnets", is described as follows:

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

**Creates:**  
A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

A diagram illustrates the setup: an "Amazon Virtual Private Cloud" contains a "Public Subnet" and a "Private Subnet". The "Public Subnet" is connected to the "Internet, S3, DynamoDB, SNS, SQS, etc." cloud. A "NAT" device is shown between the "Public Subnet" and the "Private Subnet", indicating that traffic from the private subnet goes through the NAT to reach the Internet.

At the bottom right of the configuration area, there is a blue "Select" button and a "Cancel and Exit" link.

The footer of the console shows "Feedback", "English (US)", "© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy Policy", and "Terms of Use".

The screenshot shows the AWS VPC Management Console in a Mozilla Firefox browser. The page title is "VPC Management Console - Mozilla Firefox". The URL is "https://console.aws.amazon.com/vpc/home?region=us-east-1#wizardFullpage...". The page content is titled "Step 2: VPC with a Single Public Subnet".

The configuration form includes the following fields:

- IPv4 CIDR block:** 10.100.0.0/20 (4091 IP addresses available)
- IPv6 CIDR block:**  No IPv6 CIDR Block,  Amazon provided IPv6 CIDR block,  IPv6 CIDR block owned by me
- VPC name:** VPCMediAdvice
- Public subnet's IPv4 CIDR:** 10.100.1.0/24 (251 IP addresses available)
- Availability Zone:** us-east-1a
- Subnet name:** Public subnet

Below the subnet name, it says: "You can add more subnets after AWS creates the VPC."

**Service endpoints:**

**Enable DNS hostnames:**  Yes  No

**Hardware tenancy:** Default

At the bottom right of the configuration area, there are three buttons: "Cancel and Exit", "Back", and "Create VPC".

The footer of the console shows "Feedback", "English (US)", "© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.", "Privacy Policy", and "Terms of Use".

### 3.1.1 Configuring the VPC's ACL

Network ACL ac-0246011d1fe8817aa

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
110	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
120	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
130	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW

**\* Required** Cancel Save

Network ACL ac-0246011d1fe8817aa

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
110	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
120	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW

**\* Required** Cancel Save

## 3.2 Subnet architecture

Subnets > Create subnet

### Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Public subnet 2

VPC\*: vpc-09a493d53dd471610

Availability Zone: us-east-1b

VPC CIDRs	CIDR	Status	Status Reason
	10.100.0.0/20	associated	

IPv4 CIDR block\*: 10.100.2.0/24

\* Required

Cancel Create

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Create subnet | VPC Management Console - Mozilla Firefox

Correo: adelo aleman x Create subnet | VPC | x

https://console.aws.amazon.com/vpc/home?region=us-east-1

Subnets > Create subnet

## Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Private Web-Tiers subnet 2

VPC\*: vpc-09a493d53dd471610

Availability Zone: us-east-1b

VPC CIDRs	CIDR	Status	Status Reason
	10.100.0.0/20	associated	

IPv4 CIDR block\*: 10.100.4.0/24

\* Required

Cancel Create

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Subnets | VPC Management Console - Mozilla Firefox

Correo: adelo aleman x Subnets | VPC Manag | x

https://console.aws.amazon.com/vpc/home?region=us-east-1#subnets:sort=Subnet

Create subnet Actions

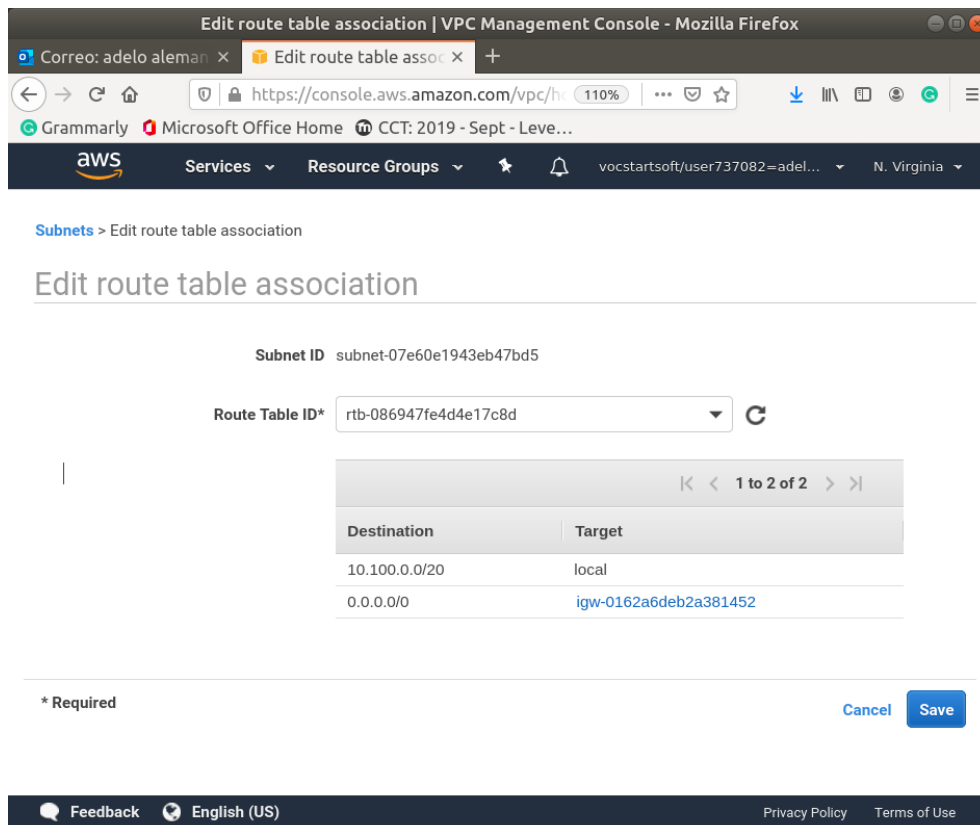
Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6	Availability Zone
Public subnet 2	subnet-07e60e1...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.2.0/24	251	-	us-east-1b
Private App-Tiers subnet 2	subnet-081d1b5...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.6.0/24	251	-	us-east-1b
Private App-Tiers subnet 1	subnet-0c8fd4...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.5.0/24	251	-	us-east-1a
Private Web-Tiers subnet 2	subnet-0e544af...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.4.0/24	251	-	us-east-1b
Public subnet 1	subnet-0f49c78...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.1.0/24	250	-	us-east-1a
Private Web-Tiers subnet 1	subnet-0ffa421c...	available	vpc-09a493d53dd471610   VPCMediAdvice	10.100.3.0/24	251	-	us-east-1a
	subnet-3eec2b1f	available	vpc-3debfe47	172.31.80.0/20	4091	-	us-east-1b
	subnet-5ce05252	available	vpc-3debfe47	172.31.64.0/20	4091	-	us-east-1f
	subnet-998800d4	available	vpc-3debfe47	172.31.16.0/20	4091	-	us-east-1c
	subnet-9ab673c5	available	vpc-3debfe47	172.31.32.0/20	4091	-	us-east-1d
	subnet-a13f339f	available	vpc-3debfe47	172.31.48.0/20	4091	-	us-east-1e

Subnets: subnet-0ffa421cd74b84f39, subnet-0f49c78597d4c7fe9, subnet-0e544af5694864620, subnet-0c8fd4a804bddc38, subnet-081d1b5fd45780c61, subnet-07e60e1943eh47hv15

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

### 3.2.1 Adding a target to the IGW in the public-named subnets to make them really publics



Subnets > Edit route table association

## Edit route table association

Subnet ID subnet-07e60e1943eb47bd5

Route Table ID\*

Destination	Target
10.100.0.0/20	local
0.0.0.0/0	<a href="#">igw-0162a6deb2a381452</a>

\* Required Cancel Save

Feedback English (US) Privacy Policy Terms of Use

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create subnet Actions

Filter by tags and attributes or search by keyword 1 to 12 of 12

<input type="checkbox"/>	Name	Subnet ID	State	VPC
<input checked="" type="checkbox"/>	Public subnet 2	subnet-07e60e1...	available	vpc-09a493d53dd471610   VPCMediAdvice
<input type="checkbox"/>	Private App-Tiers subnet 2	subnet-081d1b5...	available	vpc-09a493d53dd471610   VPCMediAdvice
<input type="checkbox"/>		subnet-0b0cf56d	available	vpc-3debfe47

Subnet: subnet-07e60e1943eb47bd5

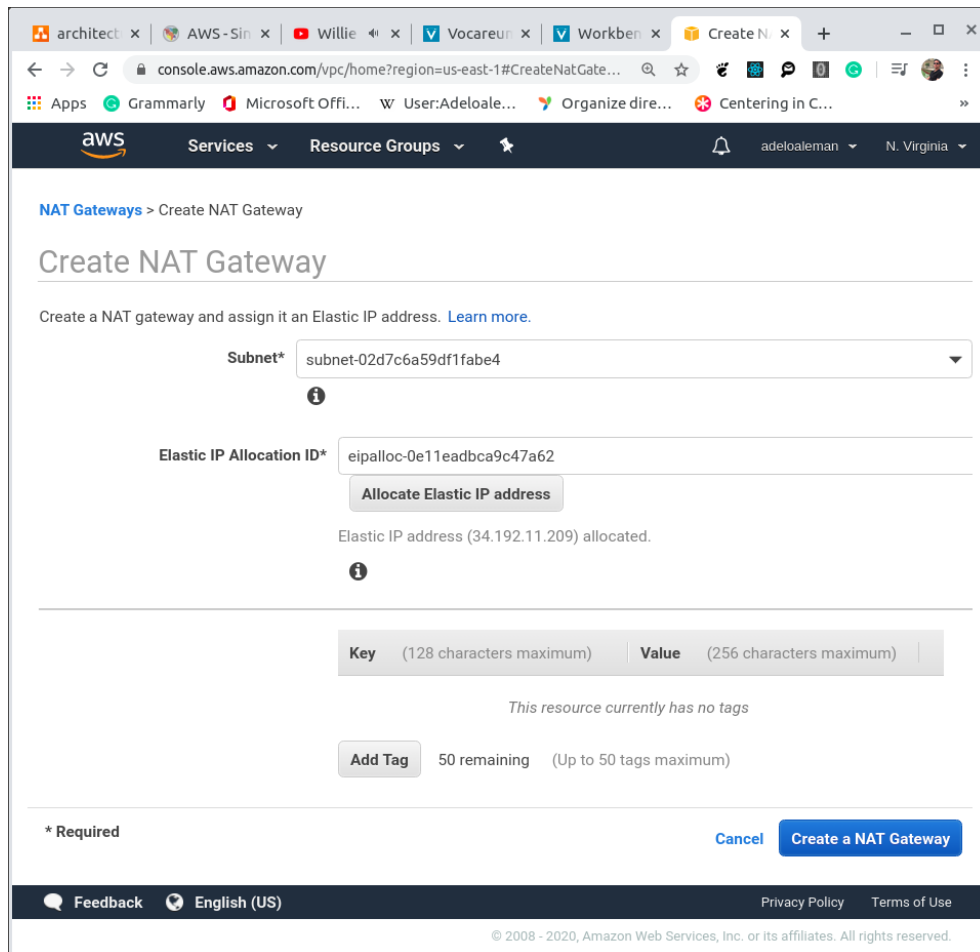
Description Flow Logs **Route Table** Network ACL Tags Sharing

Edit route table association

Route Table: [rtb-086947fe4d4e17c8d](#)

Destination	Target
10.100.0.0/20	local
0.0.0.0/0	<a href="#">igw-0162a6deb2a381452</a>

### 3.3 Creating a NAT Gateway in each Public subnet



### 3.4 Creating 2 new private route tables

These private route tables are going to be used so Private Web-Tiers subnet 1, Private Web-Tiers subnet 2, Private App-Tiers subnet 1, and Private App-Tiers subnet 2 are able to route Internet-bound traffic through the NAT Gateways created in the Public subnets.



Create route table | VPC

console.aws.amazon.com/vpc/home?region=us-east-1#CreateRouteTa...

aws Services Resource Groups adeloaleman N. Virginia

Route Tables > Create route table

## Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag

VPC\*

\* Required Cancel Create

Feedback English (US) Privacy Policy Terms of Use

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Edit routes | VPC Manage

console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:routeTab...

aws Services Resource Groups adeloaleman N. Virginia Support

Route Tables > Edit routes

## Edit routes

Destination	Target	Status	Propagated
<input type="text" value="10.100.0.0/20"/>	<input type="text" value="local"/>	active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="nat-0840d60b9cd75ce81"/>		No

Add route

\* Required Cancel Save routes

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 3.5 Defining Servers Security Groups

Create security group | VPC x New Tab

console.aws.amazon.com/vpc/home?region=us-east-1#CreateSecurityGroup

Services Resource Groups

Security Groups > Create security group

### Create security group

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group fill in the fields below.

Security group name\* Web-tier servers ⓘ

Description\* Web-tier servers ⓘ

VPC vpc-041fc1e97d0f33a26 ⓘ

\* Required

Cancel Create

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Edit inbound rules | VPC x New Tab

console.aws.amazon.com/vpc/home?region=us-east-1#ModifyInboundSecurityGroupRules:groupId=sg-06529a7c8ffff74f9

Services Resource Groups

Security Groups > Edit inbound rules

### Edit inbound rules

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

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
HTTP	TCP	80	Custom 10.100.3.0/24, 10.100.4.0/24	e.g. SSH for Admin Desktop	⊗
HTTPS	TCP	443	Custom 10.100.3.0/24, 10.100.4.0/24	e.g. SSH for Admin Desktop	⊗
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop	⊗

Add Rule

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

\* Required

Cancel Save rules

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 3.6 Launching a Web and App tiers instances and configuring the Web application

### 3.6.1 Launching the instances

Launch instance wizard | EC2 Management Console - Mozilla Firefox

Correo: adelo aleman x Launch instance wizard x +

https://console.aws.amazon.com/ec2/v2/home?re 110%

Grammarly Microsoft Office Home CCT: 2019 - Sept - Leve...

aws Services Resource Groups vocstartsoft/user737082=adel... N. Virginia Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group

### Step 1: Choose an Amazon Machine Image (AMI)

**SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type, Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.**

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

**Free tier eligible**

**Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-085925f297f89fce1 (64-bit x86) / ami-05d7ab19b28efa213 (64-bit Arm)**

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

**Are you launching a database instance? Try Amazon RDS.**

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy **Amazon Aurora, MariaDB, MySQL, Oracle, PostgreSQL, and SQL Server** databases on AWS. **Aurora** is a MySQL- and PostgreSQL-compatible, enterprise-class database

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

ec Highlight All Match Case Match Diacritics Whole Words 5 of 5 matches

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group

## Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

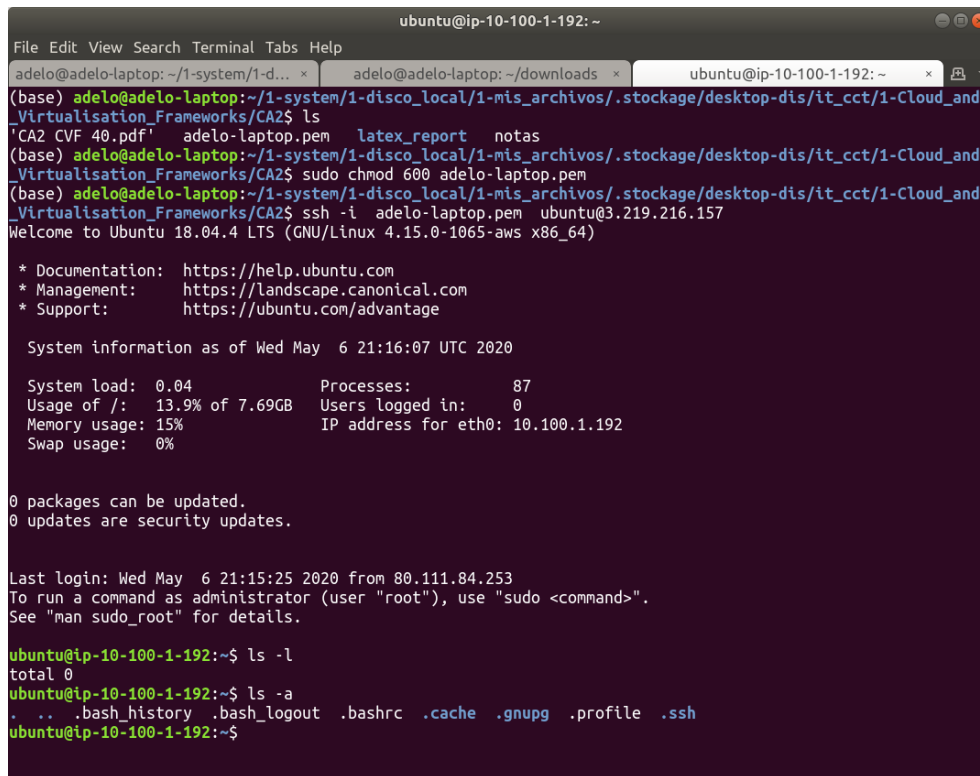
Filter by: All Instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Modera
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Modera
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Modera
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Modera
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Modera

Cancel Previous Review and Launch Next: Configure Instance Details

### 3.6.2 Accessing the instance using SSH and Installing the necessary packages



```
ubuntu@ip-10-100-1-192: ~
File Edit View Search Terminal Tabs Help
adelo@adelo-laptop: ~/1-system/1-d... x adelo@adelo-laptop: ~/downloads x ubuntu@ip-10-100-1-192: ~ x
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virtualisation_Frameworks/CA2$ ls
'CA2 CVF 40.pdf' adelo-laptop.pem latex_report notas
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virtualisation_Frameworks/CA2$ sudo chmod 600 adelo-laptop.pem
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virtualisation_Frameworks/CA2$ ssh -i adelo-laptop.pem ubuntu@3.219.216.157
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 4.15.0-1065-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May  6 21:16:07 UTC 2020

System load:  0.04          Processes:      87
Usage of /:   13.9% of 7.69GB  Users logged in:  0
Memory usage: 15%          IP address for eth0: 10.100.1.192
Swap usage:   0%

0 packages can be updated.
0 updates are security updates.

Last login: Wed May  6 21:15:25 2020 from 80.111.84.253
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-100-1-192:~$ ls -l
total 0
ubuntu@ip-10-100-1-192:~$ ls -a
.  ..  .bash_history  .bash_logout  .bashrc  .cache  .gnupg  .profile  .ssh
ubuntu@ip-10-100-1-192:~$
```

```
ubuntu@ip-10-100-1-192: ~  
File Edit View Search Terminal Tabs Help  
adelo@adelo-laptop: ~/1-system/1-d... x adelo@adelo-laptop: ~/downloads x ubuntu@ip-10-100-1-192: ~ x  
ubuntu@ip-10-100-1-192:~$ sudo apt update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]  
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [8570 kB]  
Get:5 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]  
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/universe Translation-en [4941 kB]  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [151 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic/multiverse Translation-en [108 kB]  
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [932 kB]  
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [318 kB]  
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [50.1 kB]  
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/restricted Translation-en [12.6 kB]  
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1068 kB]  
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/universe Translation-en [332 kB]  
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [15.5 kB]  
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/multiverse Translation-en [6352 B]  
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [7516 B]  
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/main Translation-en [4764 B]  
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [7484 B]  
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4436 B]  
Get:21 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [707 kB]  
Get:22 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [224 kB]  
Get:23 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [40.3 kB]  
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [10.2 kB]  
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [660 kB]  
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [219 kB]  
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [7392 B]  
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2788 B]  
Fetched 18.7 MB in 4s (5149 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
20 packages can be upgraded. Run 'apt list --upgradable' to see them.  
ubuntu@ip-10-100-1-192:~$
```

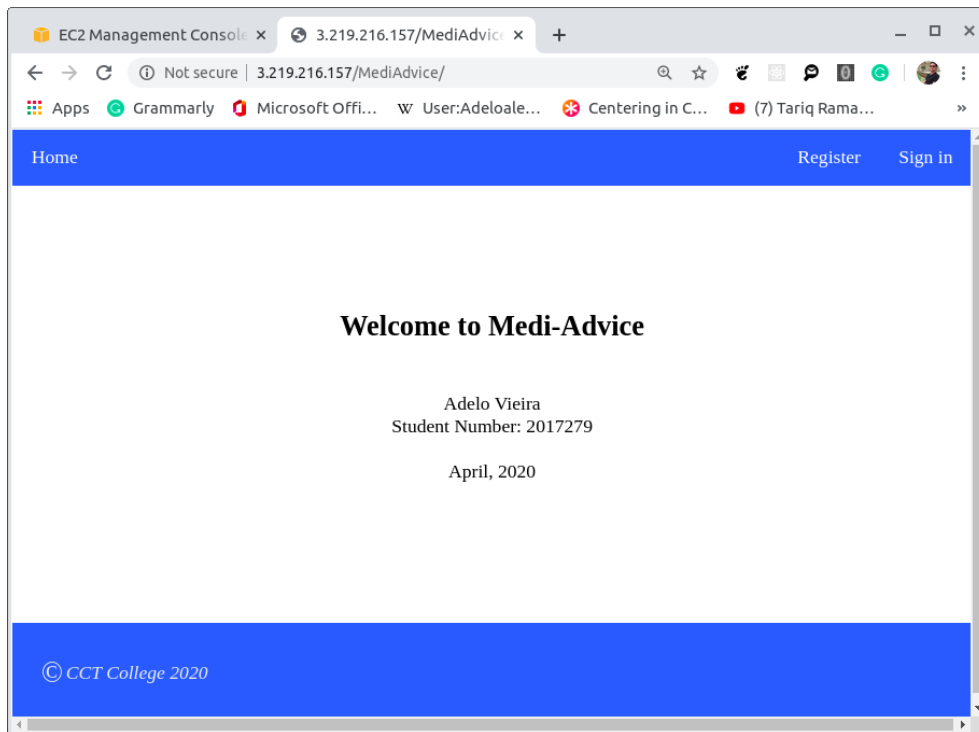
```
ubuntu@ip-10-100-1-192: ~  
File Edit View Search Terminal Tabs Help  
adelo@adelo-laptop: ~/1-system/1-d... x adelo@adelo-laptop: ~/downloads x ubuntu@ip-10-100-1-192: ~ x  
ubuntu@ip-10-100-1-192:~$ sudo apt install apache2  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap  
  liblua5.2-0 ssl-cert  
Suggested packages:  
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom openssl-blacklist  
The following NEW packages will be installed:  
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3  
  libaprutil1-ldap liblua5.2-0 ssl-cert  
0 upgraded, 10 newly installed, 0 to remove and 20 not upgraded.  
Need to get 1729 kB of archives.  
After this operation, 6986 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

```
ubuntu@ip-10-100-1-192: ~  
File Edit View Search Terminal Tabs Help  
adelo@adelo-laptop: ~/1-system/1-d... x adelo@adelo-laptop: ~/downloads x ubuntu@ip-10-100-1-192: ~ x  
ubuntu@ip-10-100-1-192:~$ sudo apt install php  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libapache2-mod-php7.2 libsodium23 php-common php7.2 php7.2-cli php7.2-common php7.2-json  
  php7.2-opcache php7.2-readline  
Suggested packages:  
  php-pear  
The following NEW packages will be installed:  
  libapache2-mod-php7.2 libsodium23 php php-common php7.2 php7.2-cli php7.2-common php7.2-json  
  php7.2-opcache php7.2-readline  
0 upgraded, 10 newly installed, 0 to remove and 20 not upgraded.  
Need to get 4010 kB of archives.  
After this operation, 17.6 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

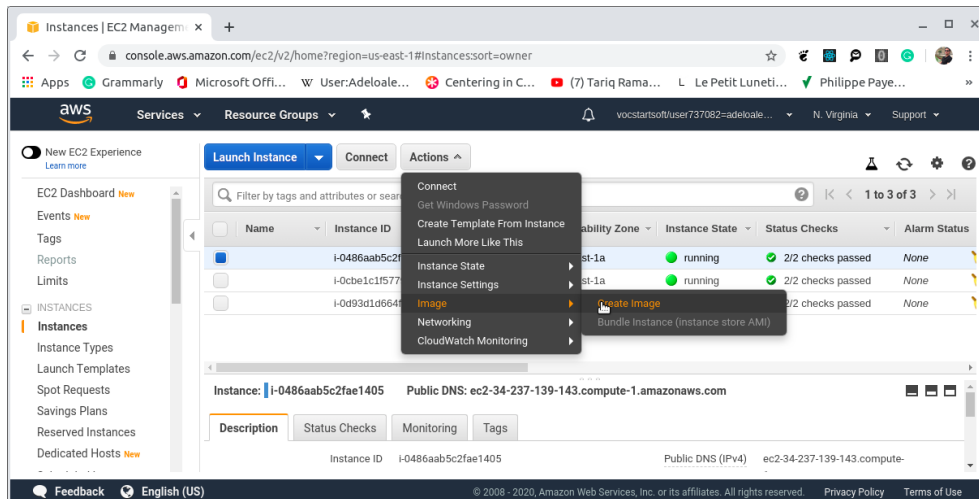
### 3.6.3 Uploading the web app and testing it

```
adelo@adelo-laptop: ~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virtualisation_Framework... x  
File Edit View Search Terminal Tabs Help  
adelo@adelo-laptop: ~/1-s... x adelo@adelo-laptop: ~/do... x adelo@adelo-laptop: ~/1-s... x ubuntu@ip-10-100-1-192: ~ x  
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virt  
ualisation_Frameworks/CA2$ ls  
'CA2 CVF 40.pdf' MediAdvice adelo-laptop.pem latex_report notas  
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virt  
ualisation_Frameworks/CA2$ scp -ri adelo-laptop.pem MediAdvice ubuntu@3.219.216.157:  
menu.css 100% 1167 5.6KB/s 00:00  
menu2.css 100% 1287 12.8KB/s 00:00  
style2.css 100% 128 1.3KB/s 00:00  
style.css 100% 1417 13.3KB/s 00:00  
index.html 100% 1013 10.1KB/s 00:00  
(base) adelo@adelo-laptop:~/1-system/1-disco_local/1-mis_archivos/.stockage/desktop-dis/it_cct/1-Cloud_and_Virt  
ualisation_Frameworks/CA2$
```

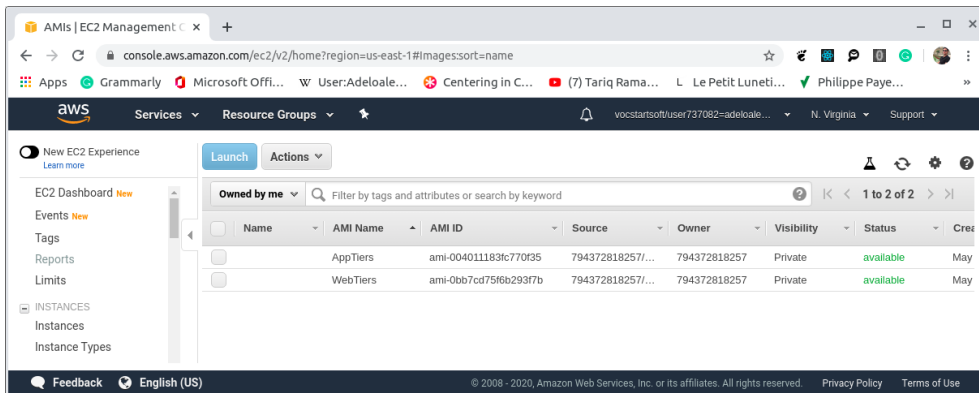
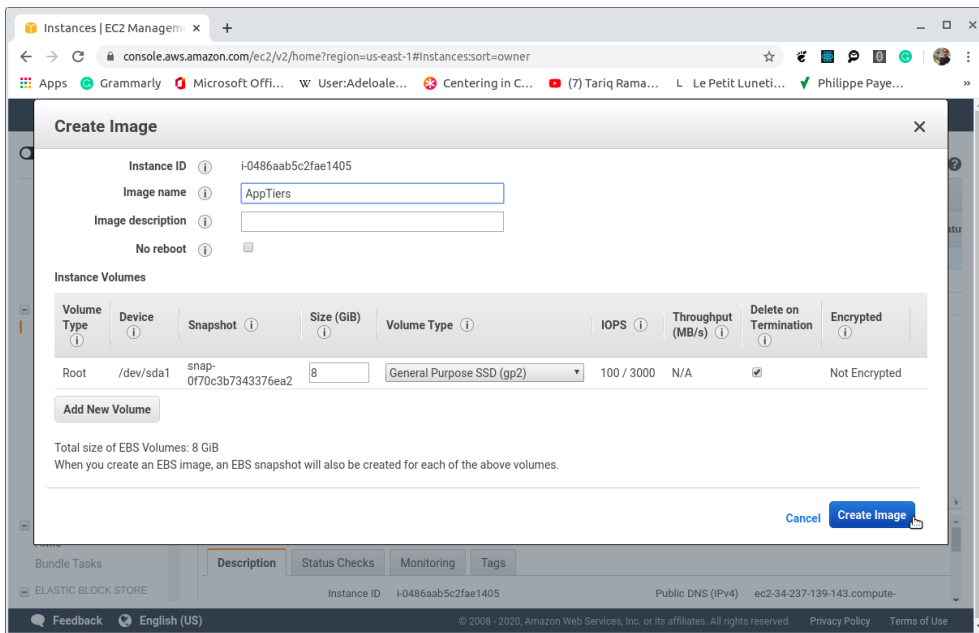
```
ubuntu@ip-10-100-1-192: ~  
File Edit View Search Terminal Tabs Help  
adelo@adelo-laptop: ~/1-s... x adelo@adelo-laptop: ~/do... x adelo@adelo-laptop: ~/1-s... x ubuntu@ip-10-100-1-192: ~ x  
ubuntu@ip-10-100-1-192:~$ ls  
MediAdvice  
ubuntu@ip-10-100-1-192:~$ sudo cp -r MediAdvice/ /var/www/html/  
ubuntu@ip-10-100-1-192:~$ ls /var/www/html/  
MediAdvice index.html  
ubuntu@ip-10-100-1-192:~$
```



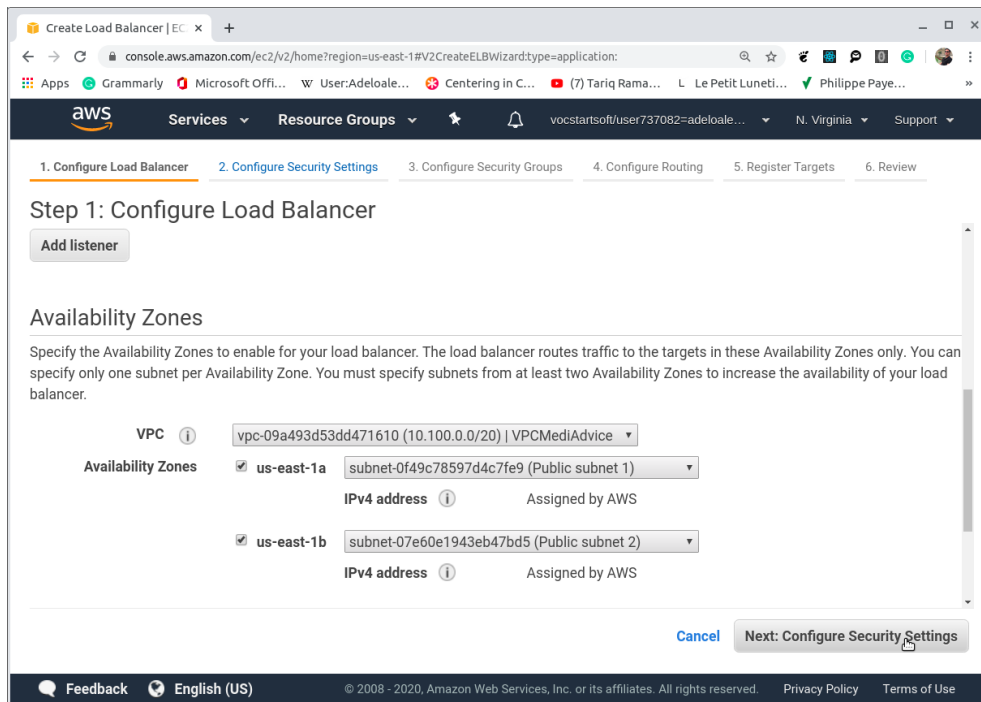
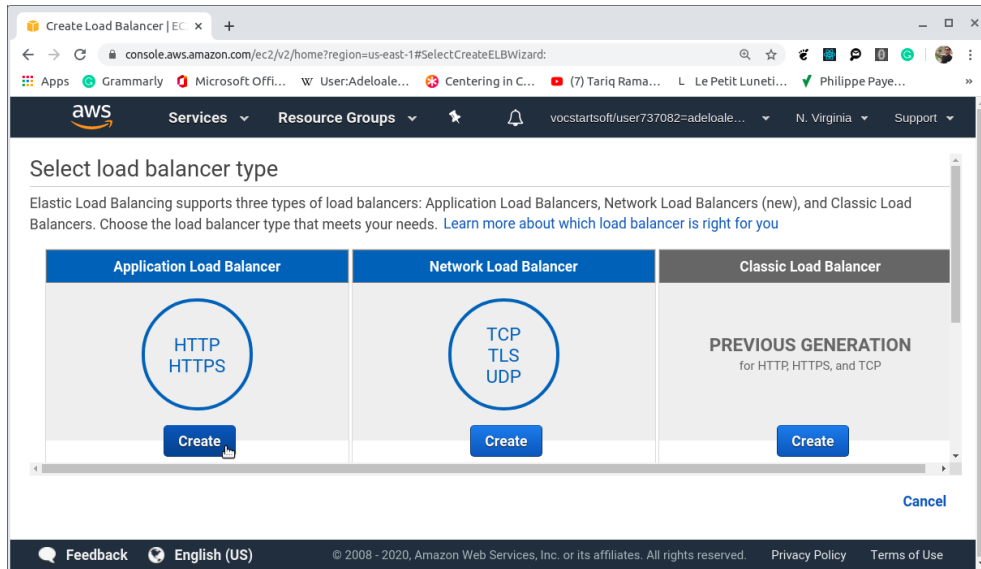
### 3.7 Creating Amazon Machine Images for the Web-Tier and App-Tier instances







## 3.8 Load Balancing



Create Load Balancer | EC2 x +

console.aws.amazon.com/ec2/v2/home?region=us-east-1#V2Cre...  
Apps Grammarly Microsoft Offi... W User:Adeloale... Centering in C... (7) Tariq Rama...

aws Services Resource Groups vocstartsoft/user737082=adeloale... N. Virginia Support

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

### Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.

#### Target group

Target group

Name

Target type  
 Instance  
 IP  
 Lambda function

Protocol

Port

#### Health checks

Protocol

Path

Advanced health check settings

Port  traffic port  
 override

Healthy threshold

Unhealthy threshold

Timeout  seconds

Interval  seconds

Success codes

[Cancel](#) [Previous](#) [Next: Register Targets](#)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

## 3.9 Auto Scaling

The screenshot shows the 'Create Auto Scaling Group' wizard in the AWS Management Console. The page is divided into several sections:

- Load Balancing:** Includes a checkbox for 'Receive traffic from one or more load balancers' and a dropdown for 'Target Groups' set to 'TG-AppTiers'.
- Health Check Type:** Radio buttons for 'ELB' (selected) and 'EC2'. Below it is a 'Health Check Grace Period' of 300 seconds.
- Monitoring:** A text block explaining that Amazon EC2 Detailed Monitoring metrics are not enabled for this configuration.
- Instance Protection:** A checkbox that is currently unchecked.
- Service-Linked Role:** A dropdown menu showing 'AWSServiceRoleForAutoScaling'.

At the bottom right, there is a 'Next: Configure scaling policies' button and a 'Cancel' button. The footer contains copyright information for Amazon Web Services, Inc. (© 2008 - 2020).

The screenshot displays the 'Auto Scaling Groups' page in the AWS Management Console. A table lists the existing groups:

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Ch
App Tier	App-Configuration	2	2	2	4	us-east-1a, us-east-1b	300	300
Web Tier	Web-Configuration	2	2	2	4	us-east-1a, us-east-1b	300	300

Below the table, the details for the 'App Tier' group are shown:

- Launch Configuration:** App-Configuration
- Desired Capacity:** 2
- Min:** 2
- Max:** 4
- Availability Zone(s):** us-east-1a, us-east-1b
- Subnet(s):** subnet-017ab1b97c30bb471ac182ecb1ac
- Classic Load Balancers:** TG-AppTiers
- Health Check Type:** ELB

The footer includes the URL 'https://console.aws.amazon.com/ec2sp/v1/si/home?regio...' and copyright information for Amazon Web Services, Inc. (© 2008 - 2020).

## 3.10 Storing Web-Accessible Content in Amazon S3

The screenshot displays the AWS Management Console interface for creating a new S3 bucket. The browser address bar shows the URL `s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1`. The page title is "Create bucket".

**General configuration**

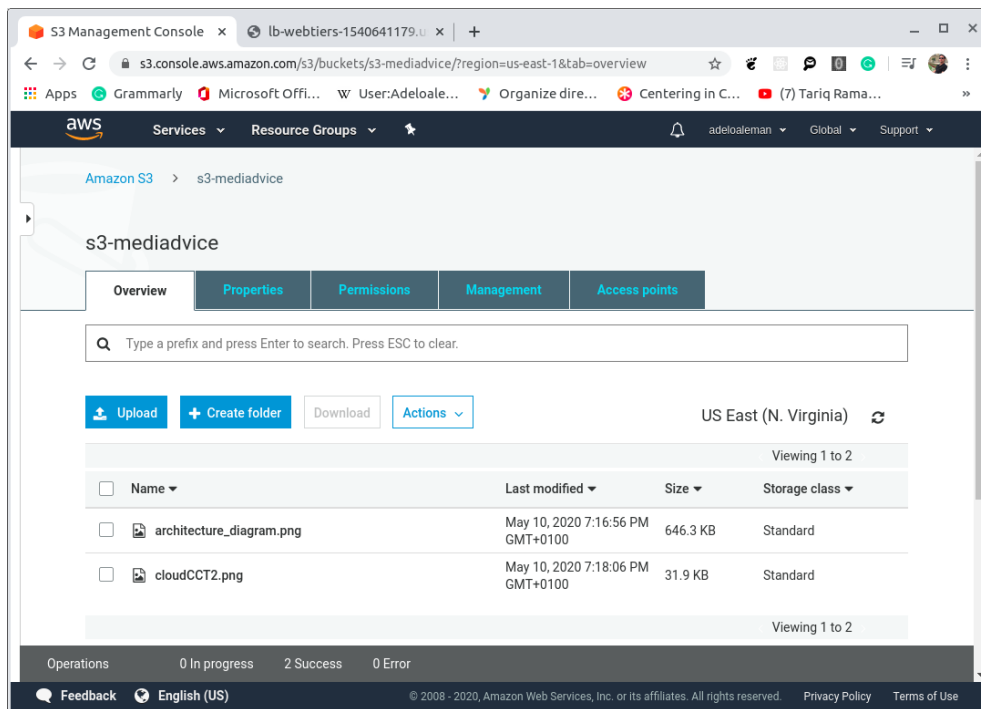
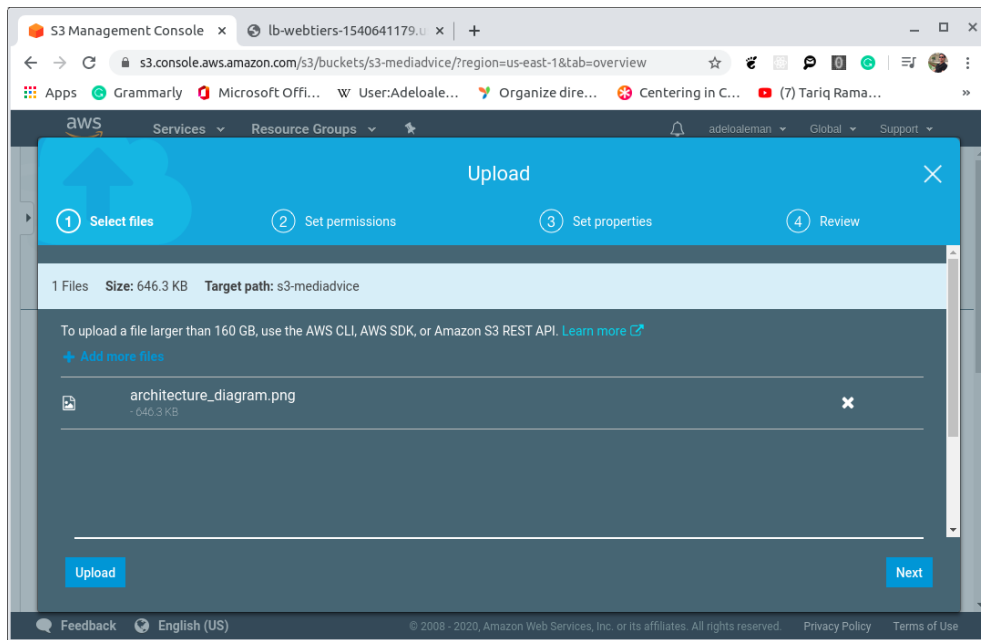
- Bucket name:** `s3-mediadvices`
- Region:** `US East (N. Virginia) us-east-1`

**Bucket settings for Block Public Access**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

- Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

Footer: Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use



## 3.11 Caching with Amazon CloudFront

The screenshot shows the 'Create Distribution' page in the AWS CloudFront console, specifically Step 2: Create distribution. The page is divided into two main sections: Origin Settings and Default Cache Behavior Settings.

**Origin Settings:**

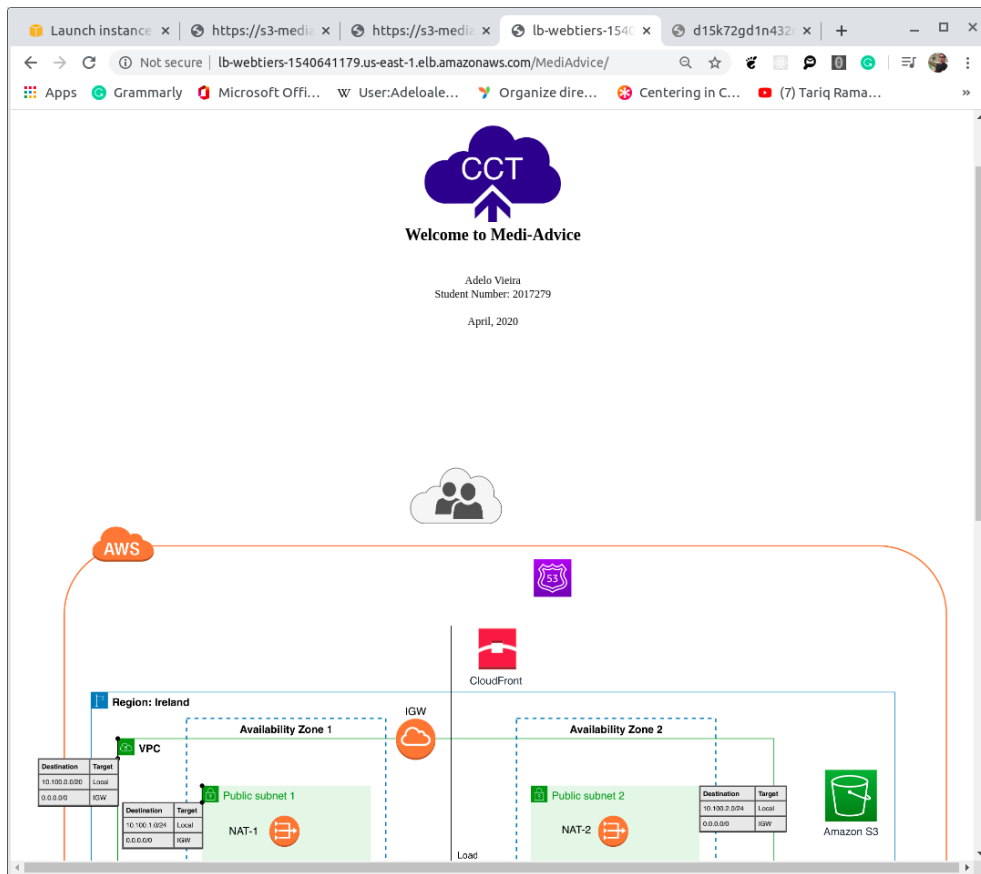
- Origin Domain Name: s3-mediadvice.s3.amazonaws.com
- Origin Path: (empty)
- Origin ID: S3-s3-mediadvice
- Restrict Bucket Access:  Yes,  No
- Origin Custom Headers: A table with columns for Header Name and Value.

**Default Cache Behavior Settings:**

- Path Pattern: Default (\*)
- Viewer Protocol Policy:  HTTP and HTTPS,  Redirect HTTP to HTTPS,  HTTPS Only
- Allowed HTTP Methods:  GET, HEAD

The screenshot shows the 'CloudFront Distributions' page in the AWS CloudFront console. The page displays a list of distributions and includes a table with the following data:

Delivery Method	ID	Domain Name	Comment	Origin
Web	E1BKRGGQMS6TUP	dc1d59pw6cbz.cloudfront.net	-	s3-mediadvices.s3.amazonaws.com



## 3.12 Cross-region disaster recovery

### 3.12.0.1 Data replication



## Task 4

# Proposed solution report

In this report, we explain the design of an architecture for a web application that will be hosted in the AWS. The design and solutions we provide are mainly based on the client's requirements but always keeping in mind the best practices from the AWS Well-Architected Framework.

In short, our goal is to propose the implementation of a secure and high available multitier architecture. In accordance with the current Medi-Advice environment, we designed an environment based on three tiers: Web Tier, App Tier, and Database Tier. To make our environment Highly Available, we have replicated the design in a second availability zone. In addition, as mentioned in the client's requirements, we will provide cross-region availability by replicating the design in another region and implementing a multi-region failover.

Below is presented a short but concrete review of the most important elements of our technical solution. In case further details are required at a particular point, we will provide links to the sections where more technical details of the proposal are provided. We also recommend to always review the Architecture diagram in Figure 2.1. This is without any doubt the best way to understand our solution.

1. **Creating a VPC:** A Virtual Private Cloud is an Amazon Service that will allow us to define a Virtual Network:
  - **ACL attached to th VPC:** After creating a VPC, we will configure its ACL. All the subnets created inside the VPC will inherit the ACL associate with the VPC; unless a particular ACL is configured on a specific subnet. Security is one of the main concerns in the design since Medi-Advice will manage very sensitive medical information. Please see the ACL in figure 1.1.
2. **Subnet architecture:**
  - **Primary region:** Ireland (eu-west-1):

- **Availability zone 1:** In our primary region, we will define a subnet architecture in a first AZ:
  - \* **Public subnet 1** We need a subnet that has a direct connection to the Internet gateway. This is necessary because our servers will need Internet access for patching and updates. In this Subnet, we will place a NAT Gateway that will provide outbound Internet connection to the private subnets where the application will be hosted.
  - \* **Private Web-Tiers subnet 1** In this layer, we will deploy our Web-Tiers servers. It is important to highlight that we won't launch Instance in this subnet or any other public subnet. Instead, we will launch a **NAT Gateway** that will connect our Instance hosted in **private subnet** with the Internet Gateway.
  - \* **Private App-Tiers subnet 1** This tier will be in charge of the App-Tiers instances.
- **Availability zone 2:** Then, we will replicate the subnet architecture in a second AZ:
  - \* **Public subnet 2**
  - \* **Private Web-Tiers subnet 2**
  - \* **Private App-Tiers subnet 2**

3. **Defining Servers Security Groups:** As an extra layer of security, We will configure strict Security groups that will only allow the necessary traffic. Please refer to Figure 1.3 if you want to review the security groups we will apply to our instances.
4. **Launching a Web and App tiers instances and configuring the Web application:** In this part we will create 2 base instances: one for the Web tier and another for the App tier. We will also install the packages we need and configure our web application.
5. **Creating Amazon Machine Images for the Web-Tier and App-Tier instances:** We will create an AMI from each of the 2 instances configured in the last step. These images will be used later for the Auto Scaling Group to create new instances in case more resources are needed.
6. **Load Balancing:** Two load balancers will be configured. One will distribute requests across the Web-tiers instance and the other across the App-tiers instances.
7. **Auto Scaling:** Two Auto Scaling Groups will be configured. One for the Web tier and the other for the App tier. They will launch or terminate instances automatically in response to the resources required by the application.
8. **Database tier:** We will manage our application's database using Amazon RDS. Amazon RDS is inherently highly available. Furthermore, by using Amazon RDS instead of a database hosted in an EC2 instance, we offload many operational and maintenance responsibilities. [[AWS documentation \(a\)](#)]
9. **Storing Web-Accessible Content in Amazon S3:** Static assets-content must be stored in Amazon S3 instead of in an EC2 instance. This is a good practice that provides benefits that will be described later in the corresponding section. [[AWS Academy \(2019b\)](#)]
10. **Caching with Amazon CloudFront:** The process of storing data in an intermediary location between the request and

the source is called caching. This process reduces cost and latency, so the requests are faster. [[AWS Academy \(2019b\)](#)]

11. Cross-region disaster recovery: Our design will provide cross-region availability by replicating the entire environment in another region and implementing a multi-region failover with Amazon Route 53.
  - Data replication from our primary Amazon S3 bucket to the Secondary S3 bucket in the other region.
  - Data replication from our primary Amazon RDB to the Secondary Amazon RDB in the other region.

# Bibliography

- AWS ACADEMY : *ACA Module 3 LAB: Making Your Environment Highly Available*, 2019a. Version A5L5. 3, 4, 7
- AWS ACADEMY : *Academy Cloud Architecting (ACA) - Module 07 Student Guide*. Amazon Web Services, Inc., 2019b. Version 1.1.5. 38, 39
- AWS ACADEMY : *Module 7: Designing Web Scale Media*. Amazon Web Services, Inc., 2019c. 9
- AWS DOCUMENTATION : Amazon rds features, a. URL <https://aws.amazon.com/rds/features/#ha>. 38
- AWS DOCUMENTATION : Network acls, b. URL [https://docs.amazonaws.cn/en\\_us/vpc/latest/userguide/vpc-network-acls.html](https://docs.amazonaws.cn/en_us/vpc/latest/userguide/vpc-network-acls.html).
- AWS DOCUMENTATION : *Security groups for your VPC*. Amazon Web Services, Inc., c. URL [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html).