DESIGNING AND IMPLEMENTING A PROOF-OF-CONCEPT CLOUD SOLUTION BASED ON A CLIENT’S NEEDS

Cloud Services

Rayen Bentemessek
2021378

Table of Contents

Task 1:

Customer requirements:

- Allow patients to upload documents and images, looking for automating the process of extracting text from documents and converting images into multiple formats.
- If the database goes down, the application will fail to retrieve data (SPOF).
- Very slow application, especially during high traffic (scalability issues), auto scaling group because they are scaling manually atm.
- Application servers are based in Ireland, causing latency to customers in USA and around the world.
- Requires five nines availability.
- Reduce expenses (cost optimization).

Proposed solutions:

Medi-Advice first of all needs to change the whole infrastructure, they are having loads of issues with outages and data retrieval issues because the diagram shows the database as a single point of failure which is a very common wrong approach as you should never put all you eggs in the same basket, which means we require adding other database tiers in the infrastructure, also we can implement multiple availability zones so we'll have different versions throughout the zones. Adding a load balancer and an auto scaling group is a must nowadays and Medi-Advice needs them as they will take care of the scalability issues and automate the process instead of manual scaling. This will ensure the high availability they are looking for, also adding backups and disaster recovery is highly recommended as we're dealing with patients' information, which is critical information, Medi-Advice wants to make sure nothing is lost.

Here I've put a summary of the solutions that can be helpful for Medi-Advice:

Amazon VPC:

VPC will be a layer of security, as we are dealing with sensitive information about patients and their disease, I consider VPC highly important here to protect the layers of architecture and create an isolated environment. There are other measures of security that need to be taken but they will be all explained later in this assignment. For now, we are just starting with the VPC.

Elastic Load Balancer:

- Load Balancer will distribute incoming traffic across the available instances making sure each one handles a reasonable amount of traffic without overusing one of them and making sure the application is highly available.

EC2 Auto scaling:

- Auto scaling will automatically adapt the number of instances based on the initial configuration (desired number of instances, maximum number of instances etc..) which will make sure the application can upscale and downscale based on the current demand, which in our case will manage to handle the high traffic in covid and then returns to normal state when the rush has ended, this works like a retail shop, you want to have a maximum number of staff on the floor (instances in our case) during rush hours and holidays, then during low season, staff number will be reduced.

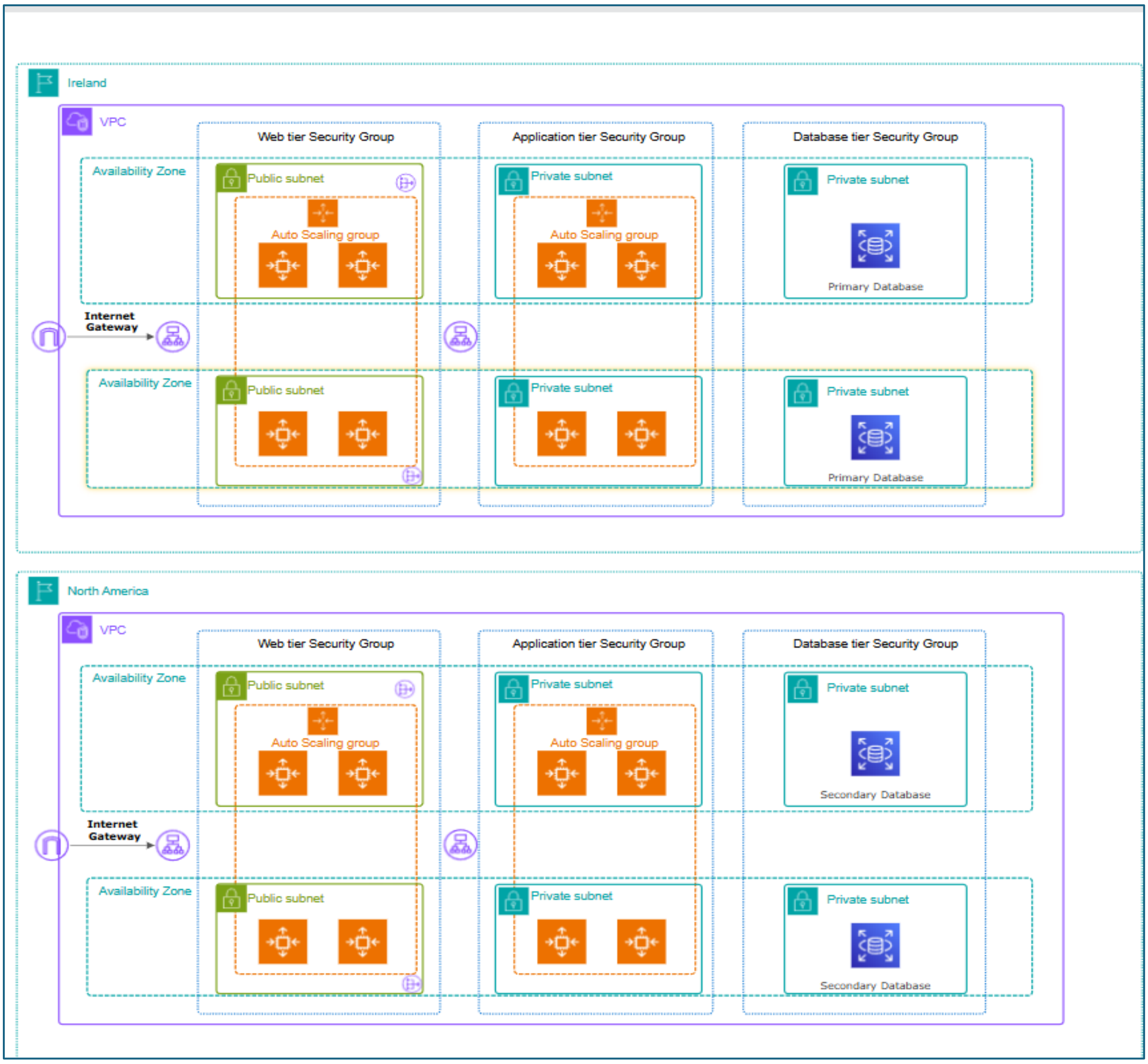
Amazon Relational Database service (RDS) with Multi-AZ:

- This service will handle the database tier that was a single point of failure, with multi-AZ enabled, the database will be replicated across the availability zones ensuring that the database is backed up and synchronized.

Amazon S3, Amazon Textract:

- Amazon Textract will ensure that text is extracted from documents using machine learning techniques which will automate the process of extracting text from documents (Amazon Web Services, Inc., 2019).
- Amazon S3 will be used to upload the documents and store the extracted text, will also store the brochure and any further additional websites (Amazon Web Services, Inc., 2019).

Task 2:



This is the architecture diagram, it will ensure that the previously mentioned services are there, Load balancer in purple, avoiding Single point of failure by using primary and secondary database across the two regions, also I've added a North America region to avoid the latencies happening, I've also created different security groups inside the VPC to enhance the security of the architecture, this was accompanied by adding a NAT gateway inside the public subnets and an internet gateway in the VPC. In addition to that, I've attached autoscaling groups to automatically scale up and down base on the current traffic. To do this diagram, I've used smartdraw website, I've followed the best practices by (Krassowski, 2022) to make sure I deliver a highly available architecture.

Task 3:

Task 3a:

Medi-Advice current infrastructure has currently few anti-patterns that need to be addressed to avoid any potential issues, two of these anti-patterns are:

- **The absence of an auto scaling group:** For a medical company that can be a last resort to some patients with high urgency demands, the application has to be highly available and provide a smooth user experience at any time, because users do not really want to see the application failing as this will result in going for other competitors and the company will no longer be in business. That is why getting rid of manual scaling and adding auto scaling groups and policies is a must with the current covid-19 situation where more users are expected to be visiting. The proposed solution will make sure failures don't happen as I've attached auto scaling in all the tiers combined with a Load balancer to ensure the availability of the application.
- **Single point of failure:** Another big issue with the current infrastructure, there is only one source of database retrieval, if that server fails, the whole application will fail and can be even worse by losing database records that are one of the most important assets of the company, that is why this has to be rectified by implementing Amazon RDS with multi Availability zones as part of my proposed solution. I've also made a primary database in the primary region, Ireland, and a secondary database in North America known as read replica which will guarantee high availability and constant backups that are useful in case of failures, they also communicate together to make sure records are updated and synchronized (docs.aws.amazon.com, n.d.).

Task 3b:

Reducing costs while maintaining quality is every business's goal, and in our context, Medi-Advice is looking to reduce costs and maintain a highly available application that robust and reliable. This can be achieved by following the following 6 pillars of the AWS Well-architected framework:

- **Operational excellence:** This pilar requires building a solid software that can deliver a great customer experience, one of the key points here is automation, which is missing in the current working environment, as mentioned earlier using Amazon Textract can be very helpful to automate the process of text recognition. Also implementing an auto scaling group can serve the automation purpose by reducing the manual interaction with the instances as explained earlier. Another key point is anticipating failure, because application failure means losing more money which is the opposite of reducing costs and high availability. This can be done by simply testing the application and the changes before deploying them as a way of doing things (AWS, n.d.).

- **Security:** Working with patients' data requires high level of security, which implicates the implementation of good security measures to protect the data and systems from possible vulnerabilities and breaches. This can be done through adding different Identity and access management (IAM) roles and policies to grant minimum access or least privilege to other tiers, adding encryption techniques is quite beneficial, and reducing direct access or manual process of data. Also making sure security is applied at all layers from the client end to database tier (docs.aws.amazon.com, n.d.).
- **Reliability:** This pillar ensures the correctness of the application functionality by automatically recovering from failures thanks to the automation process we discussed in the operational excellence pillar, scaling based on the current demand, avoiding single points of failure and stop guessing capacity, instead let the auto scaling group policies and configuration do the job. By following all these practices, the application will be steadily moving forward toward the five nines.
- **Performance efficiency:** This pilar is highly useful because now we are operating in two regions, Ireland (Europe-west) and North America with few issues here. This pilar consists of working between different Regions with much lower latency, which is the issue patients and doctors in the US are facing. This will make the company developers have more time to innovate, create new features and potentially scale the business by exploring other markets in different regions, generate more clients and even add new features and functionalities despite focusing only on trying to recover from the current issues (docs.aws.amazon.com, n.d.).
- **Cost optimization:** This pilar consists of the ability to run the application with the desired functionalities and objectives (five nines for example) without breaking the bank. At the moment, the company has only on-premises facilities, so it's time to go to the cloud, as this will save tons of money when it comes to scaling up and down without the need to use physical and permanent resources, so pay-as-you-go is very helpful here as it may save around 75% of costs. Also, the features on AWS, will allow the relevant management team to monitor the expenses records and even forecast the upcoming periods. And finally, as mentioned in the performance efficiency pilar, AWS will look after the infrastructure, the employees will focus on the customers while saving costs at the same time. You can grow your vegetables and buy a cow for the milk if you want, but you also can find them in the grocery store, saving you money and time, the choice is yours (docs.aws.amazon.com, n.d.).
- **Sustainability:** This pillar focuses on reducing the waste based on dynamic resource allocation which will result in less energy consumption, contributing to more eco-friendly system and less costs. The process will ensure that the performance and business goals are also met while

thinking of sustainability by using most of the pillars all together, for example, having an auto scaling group will reduce the workload in general resulting in a less energy consumption, same thing for S3 storage and the other services. Maximising utilization without affecting the workload can potentially save more energy as well, 1 instance running at 60% wouldn't likely to fail and also is more efficient than 2 instances running at 30% (Amazon.com, 2024).

Finally, these six pillars work much better all together as they will significantly help in reducing costs and ensuring high availability.

Task 4:

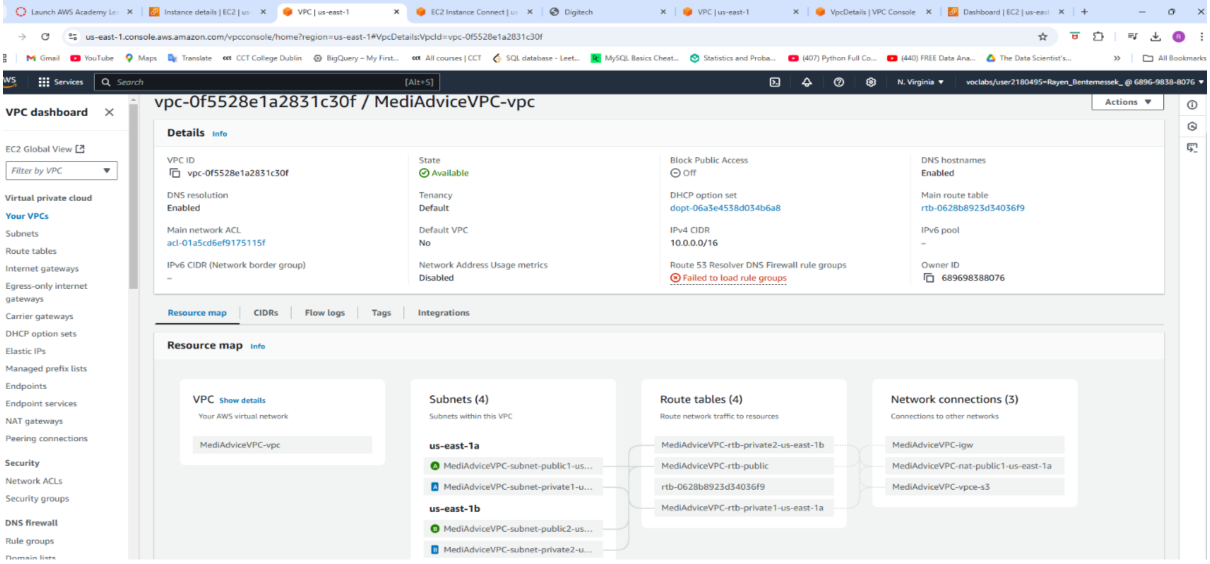
Task 4a:

A virtual Private Cloud, known as VPC, is an isolated virtual network inside the cloud that allows to create an isolated virtual network, in other words it is more like a department inside a big organization. It offers a wide range of scalability and security thanks to its features such as subnets that reside in availability zones, these subnets can be either public that have direct route to internet, or private that require a NAT gateway to access the internet offering the option to stay local or use the internet from private subnets. IP addresses for the VPC are represented using Classless Inter-Domain Routing (CIDR) associated with IPv4 CIDR blocks which can be in a range of /16 netmask (65536 IP addresses) and /28 netmask (16 IP addresses) and every resource in the VPC requires an IP address. Also, VPC's, offering a good level of security, they have security groups that can be associated with EC2 instances which allows the control of inbound and outbound traffic for the instance. As VPC's offer isolation for the network, this doesn't mean they cannot communicate as it's possible if needed by using VPC peering which is basically a connection between Two VPCs (or more) allowing the instances inside the VPC's to communicate with each other as if they are withing the same network even if they are not in the same region (Amazon Web Services, 2023).

Task 4b:

In order to do this task, I started by creating the custom VPC, set up the public and private subnets as shown in the next figure, I have also added an Internet Gateway to be able to access the internet.

Creating the VPC:



vpc-0f5528e1a2831c30f / MediAdviceVPC-vpc

Details Info

VPC ID vpc-0f5528e1a2831c30f	State Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy Default	DHCP option set dopt-06a3e4538d034b6a8	Main route table rtb-0628b8923d34036f9
Main network ACL acl-01a5cd6ef9175115f	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool –
IPv6 CIDR (Network border group) –	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 689698388076

Resource map Info

VPC Show details
Your AWS virtual network
MediAdviceVPC-vpc

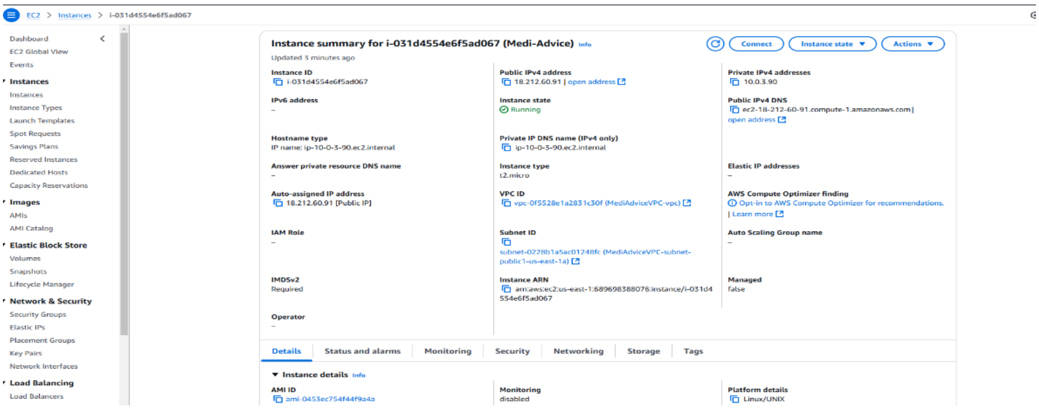
Subnets (4)
Subnets within this VPC
us-east-1a
MediAdviceVPC-subnet-public1-us...
MediAdviceVPC-subnet-private1-u...
us-east-1b
MediAdviceVPC-subnet-public2-us...
MediAdviceVPC-subnet-private2-u...

Route tables (4)
Route network traffic to resources
MediAdviceVPC-rtb-private2-us-east-1b
MediAdviceVPC-rtb-public
rtb-0628b8923d34036f9
MediAdviceVPC-rtb-private1-us-east-1a

Network connections (3)
Connections to other networks
MediAdviceVPC-igw
MediAdviceVPC-nat-public1-us-east-1a
MediAdviceVPC-vpce-s3

The I created the EC2 instance and made sure I used the custom VPC as shown in the upcoming figure.

Creating the EC2 instance using the created VPC (using a public subnet):



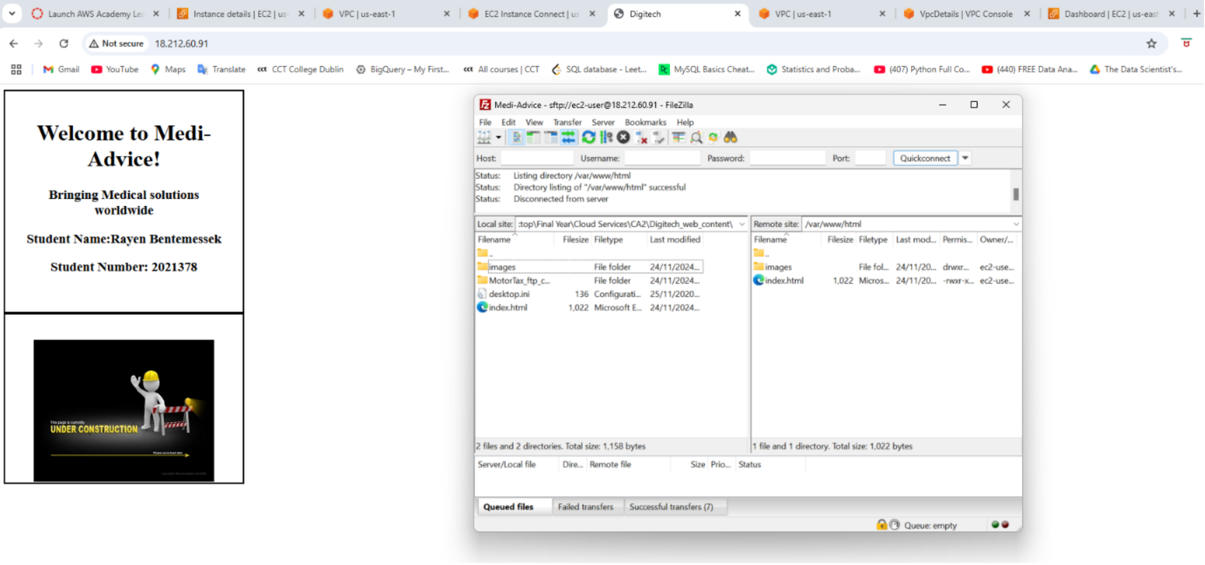
Instance summary for i-031d4554e6f5ad067 (Medi-Advice) Info
Updated 3 minutes ago

Instance ID i-031d4554e6f5ad067	Public IPv4 address 18.212.60.91 | open address	Private IPv4 addresses 10.0.3.90
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-18-212-60-91.compute-1.amazonaws.com | open address
Hostname type IP name: ip-10-0-3-90.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-3-90.ec2.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 18.212.60.91 [Public IP]	VPC ID vpc-0f5528e1a2831c30f (MediAdviceVPC-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	Subnet ID subnet-0228b1a5ac01248fc (MediAdviceVPC-subnet-public1-us-east-1a)	Auto Scaling Group name –
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:689698388076:instance/i-031d4554e6f5ad067	Managed false
Operator –		

Instance details Info

AMI ID ami-0453ec754f44f9a4a	Monitoring disabled	Platform details Linux/UNIX

Then I applied what we covered in class, used FileZilla FTP program to upload the Medi-Advice website.



Welcome to Medi-Advice!

Bringing Medical solutions worldwide

Student Name:Rayen Bentemessek

Student Number: 2021378

Medi-Advice - sftp://ec2-user@18.212.60.91 - FileZilla

Host: Username: Password: Port:

```
Status:   Listing directory /var/www/html
Status:   Directory listing of "/var/www/html" successful
Status:   Disconnected from server
```

Local site: :top\Final Year\Cloud Services\CA2\Digitech_web_content\

Filename	Filesize	Filetype	Last modified
..			
images		File folder	24/11/2024...
MotorTax_ftp_c...		File folder	24/11/2024...
desktop.ini	136	Configurati...	25/11/2020...
index.html	1,022	Microsoft E...	24/11/2024...

2 files and 2 directories. Total size: 1,158 bytes

Remote site: /var/www/html

Filename	Filesize	Filetype	Last mod...	Permis...	Owner/...
..					
images		File fol...	24/11/20...	drwxr...	ec2-use...
index.html	1,022	Micros...	24/11/20...	-rwxr-x...	ec2-use...

1 file and 1 directory. Total size: 1,022 bytes

Server/Local file	Dire...	Remote file	Size	Prio...	Status

Task 5:

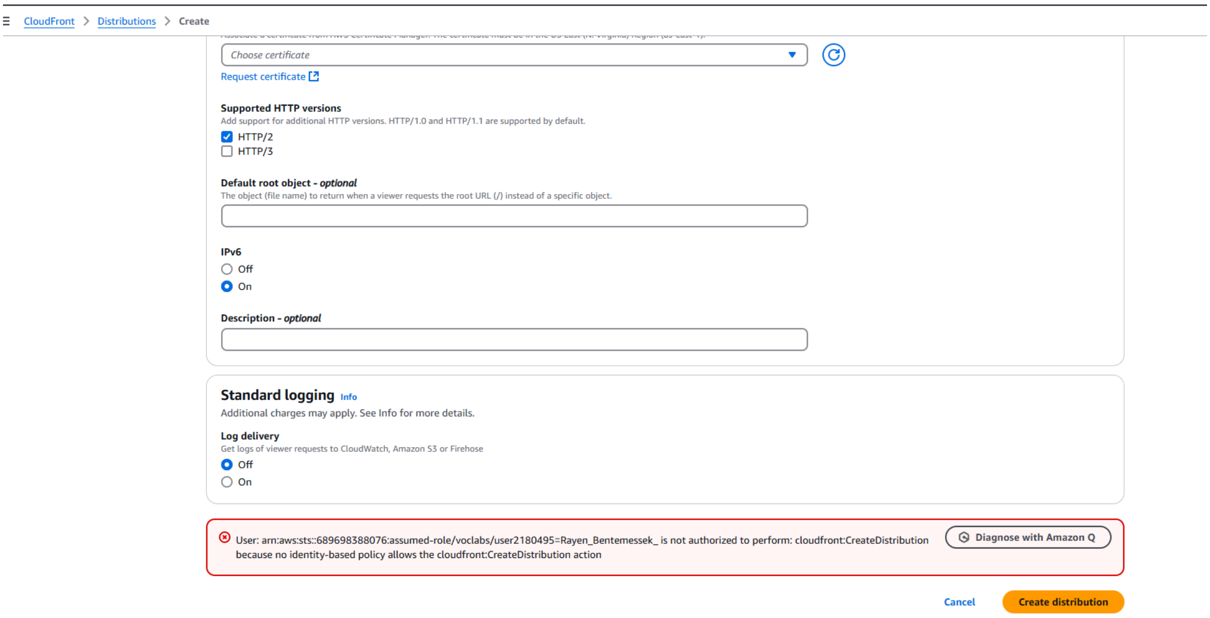
In 2022, Instagram had over 100 million photos and videos uploaded every day, around 800 million reels accessed, and there are 1 billion of them shared every day. In order to have these numbers on a daily basis and managing to deliver a smooth user experience – at least from my own perspective – helping everyone to catch up with the latest friends' pictures and viral news, Instagram follows many techniques. One of these techniques is Content Delivery Network (CDN) which consists of caching frequently accessed images and videos in geographically distributed edge locations resulting in reducing the load on the servers and managing to deliver very little latency for users no matter where you are (Appstudio, 2023).

In other words, CDN relies on caching the resources for temporary access giving the opportunity to access the desired content from the nearest server, which will result in loading pages quicker with a much better performance (Akamai, 2020). This means happy end-users, as most users – including myself – wouldn't really wait too much for the page to load as there are many other options for any content, I'm looking for whether it's a shopping website or football matches results or even medical advice. Speaking of medical advice, the use of CDN for Medi-Advice will work perfectly to deliver the brochures and potentially more to customers in the US and Ireland with very similar performance, solving the issue of slow response times and generating more income because of the improvement of the users' experience. I personally am more likely to use and rely on websites that give me better response and interactivity with very low latency, that is why companies are heavily spending in this particular point to catch and keep users as much as possible, cookies can be part of it but we'll talk about this in another day.

CDN can reduce latency from over 2800 ms to less than 900 ms which really counts for keeping users attached, which eventually will benefit Medi-Advice in generating more income, and at least good user experience and possible eventual customers due to the awareness of the services (CDN Performance | Cloudflare, n.d.).

Challenge TASK 1:

In order to achieve this task, I created an s3 bucket where I've uploaded the brochure in PDF format, then I went to CloudFront to create a distribution from my S3 bucket so I can display it, but I got a permission error that didn't let me do the task and create the distribution.



CloudFront > Distributions > Create

Choose certificate

Request certificate

Supported HTTP versions
Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.
☑ HTTP/2
☐ HTTP/3

Default root object - *optional*
The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

IPv6
○ Off
● On

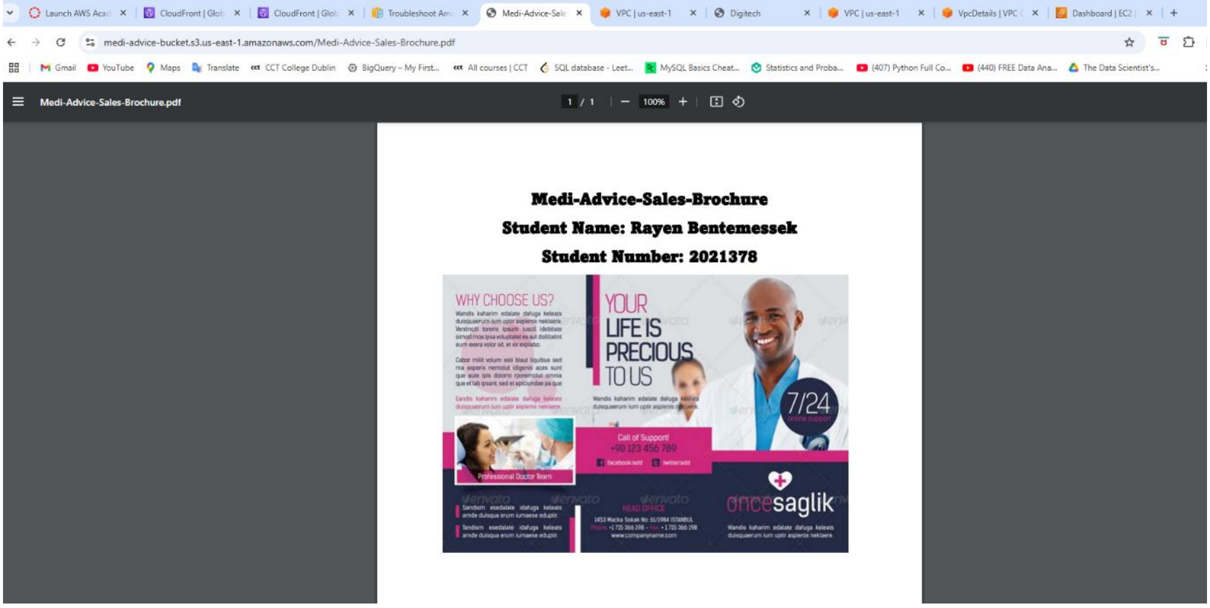
Description - *optional*

Standard logging Info
Additional charges may apply. See Info for more details.

Log delivery
Get logs of viewer requests to CloudWatch, Amazon S3 or Firehose
● Off
○ On

```
User: arn:aws:sts::689698388076:assumed-role/voclabs/user2180495=Rayen_Bentemessek_ is not authorized to perform: cloudfront:CreateDistribution
because no identity-based policy allows the cloudfront:CreateDistribution action
```

Alternatively, this is brochure link from the S3 bucket as a proof of a successful S3 upload



medi-advice-bucket.s3.us-east-1.amazonaws.com/Medi-Advice-Sales-Brochure.pdf

Medi-Advice-Sales-Brochure.pdf

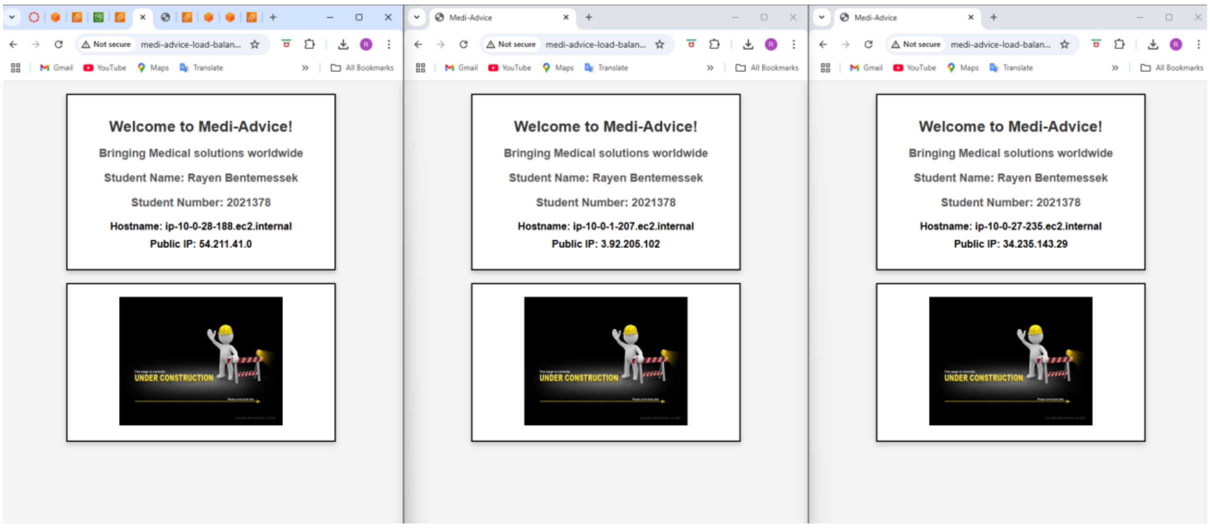
Medi-Advice-Sales-Brochure

Student Name: Rayen Bentemessek

Student Number: 2021378

Challenge TASK 2:

3 website pages (3 desired instances), I slightly modified the HTML provided to show both the private and public instance's IP address as shown in the picture below.

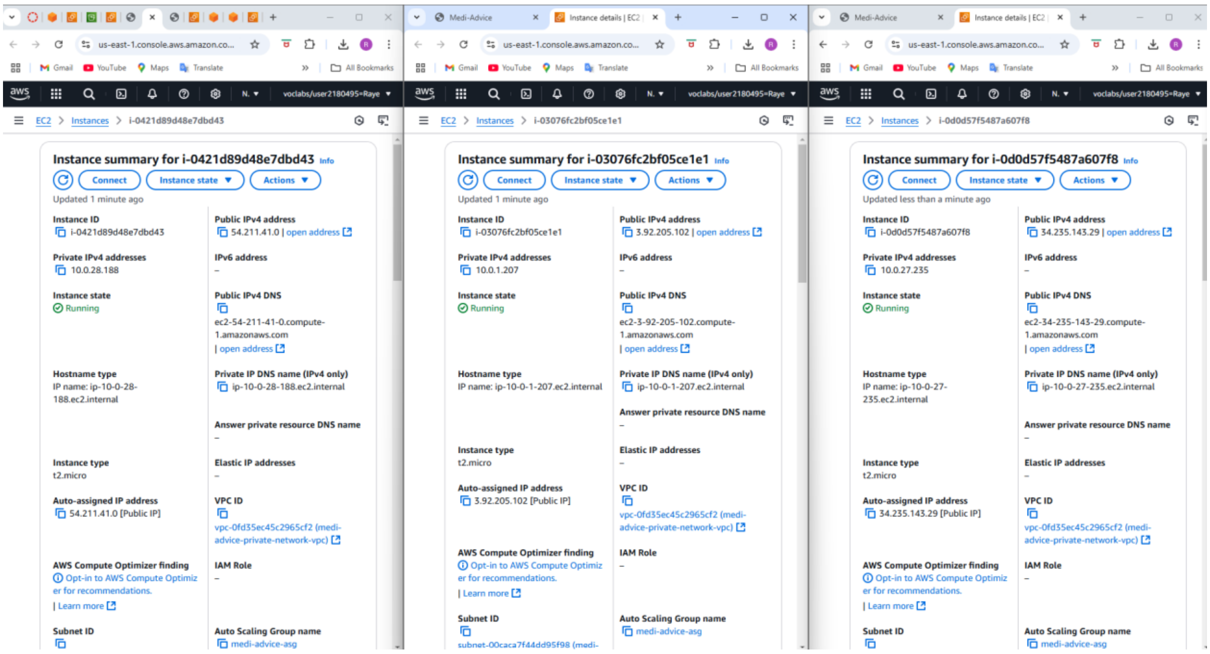


Welcome to Medi-Advice!
Bringing Medical solutions worldwide
Student Name: Rayen Bentemessek
Student Number: 2021378
Hostname: ip-10-0-28-188.ec2.internal
Public IP: 54.211.41.0

Welcome to Medi-Advice!
Bringing Medical solutions worldwide
Student Name: Rayen Bentemessek
Student Number: 2021378
Hostname: ip-10-0-1-207.ec2.internal
Public IP: 3.92.205.102

Welcome to Medi-Advice!
Bringing Medical solutions worldwide
Student Name: Rayen Bentemessek
Student Number: 2021378
Hostname: ip-10-0-27-235.ec2.internal
Public IP: 34.235.143.29

Instances Information



EC2 > Instances > i-0421d89d48e7dbd43

Instance summary for i-0421d89d48e7dbd43
Updated 1 minute ago

Instance ID: i-0421d89d48e7dbd43
Public IPv4 address: 54.211.41.0 | open address
Private IPv4 addresses: 10.0.28.188
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-54-211-41-0.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-10-0-28-188.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-28-188.ec2.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 54.211.41.0 [Public IP]
VPC ID: vpc-0fd35ec45c2965cf2 (medi-advice-private-network-vpc)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID
Auto Scaling Group name: medi-advice-asg

EC2 > Instances > i-03076fc2bf05ce1e1

Instance summary for i-03076fc2bf05ce1e1
Updated 1 minute ago

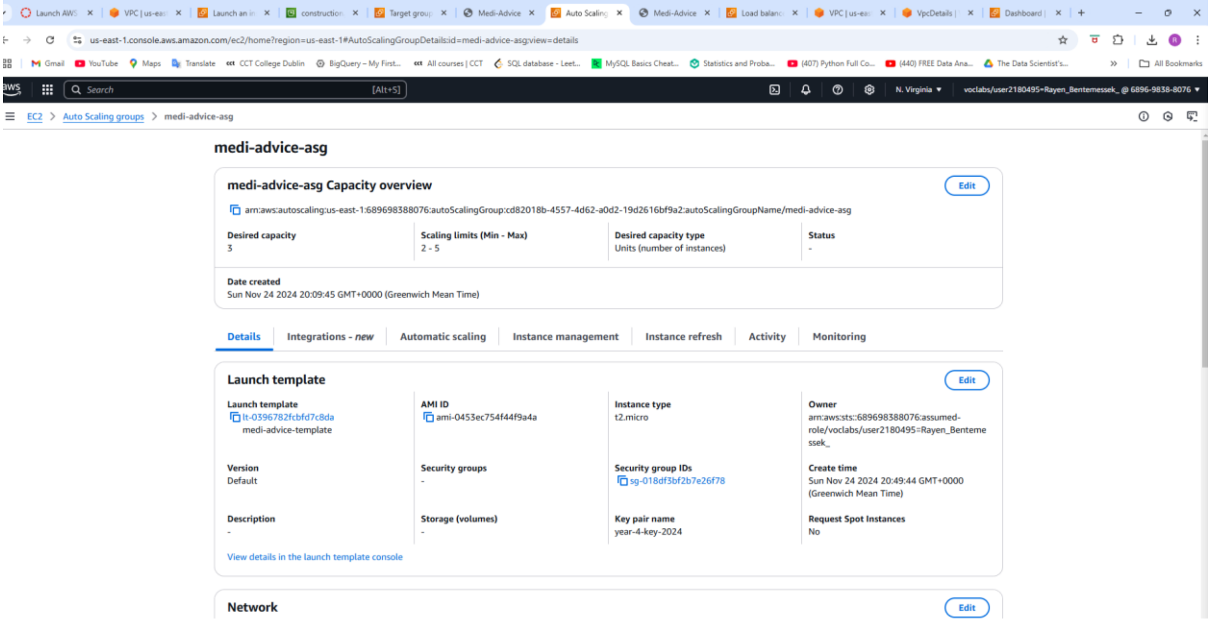
Instance ID: i-03076fc2bf05ce1e1
Public IPv4 address: 3.92.205.102 | open address
Private IPv4 addresses: 10.0.1.207
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-3-92-205-102.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-10-0-1-207.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-1-207.ec2.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 3.92.205.102 [Public IP]
VPC ID: vpc-0fd35ec45c2965cf2 (medi-advice-private-network-vpc)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID: subnet-00caca7f44dd95f98 (medi-
Auto Scaling Group name: medi-advice-asg

EC2 > Instances > i-0d0d57f5487a607f8

Instance summary for i-0d0d57f5487a607f8
Updated less than a minute ago

Instance ID: i-0d0d57f5487a607f8
Public IPv4 address: 34.235.143.29 | open address
Private IPv4 addresses: 10.0.27.235
IPv6 address: –
Instance state: Running
Public IPv4 DNS: ec2-34-235-143-29.compute-1.amazonaws.com | open address
Hostname type: IP name: ip-10-0-27-235.ec2.internal
Private IP DNS name (IPv4 only): ip-10-0-27-235.ec2.internal
Answer private resource DNS name: –
Instance type: t2.micro
Elastic IP addresses: –
Auto-assigned IP address: 34.235.143.29 [Public IP]
VPC ID: vpc-0fd35ec45c2965cf2 (medi-advice-private-network-vpc)
AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –
Subnet ID
Auto Scaling Group name: medi-advice-asg

Auto scaling group



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroupDetails:id=medi-advice-asg;view=details
Search [Alt+S]
N. Virginia ▼
voclabs/user2180495=Rayen_Bentemessek_ @ 6896-9838-8076 ▼
EC2 > Auto Scaling groups > medi-advice-asg
Edit
Details | Integrations - *new* | Automatic scaling | Instance management | Instance refresh | Activity | Monitoring

medi-advice-asg

medi-advice-asg Capacity overview

arn:aws:autoscaling:us-east-1:689698388076:autoScalingGroup:cd82018b-4557-4d62-a0d2-19d2616bf9a2:autoScalingGroupName/medi-advice-asg

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
3	2 - 5	Units (number of instances)	-

Date created
Sun Nov 24 2024 20:09:45 GMT+0000 (Greenwich Mean Time)

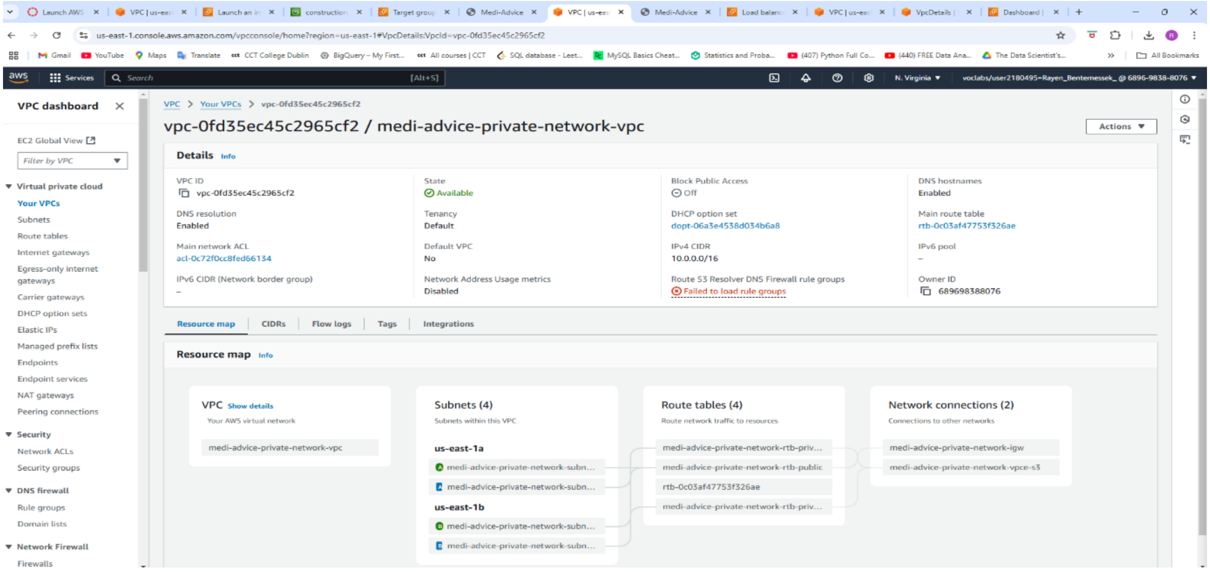
Launch template

Launch template	AMI ID	Instance type	Owner
lt-0396782fcbfd7c8da medi-advice-template	ami-0453ec754f44f9a4a	t2.micro	arn:aws:sts::689698388076:assumed-role/voclabs/user2180495=Rayen_Bentemessek_
Version	**Security groups**	**Security group IDs**	**Create time**
Default	-	sg-018df3bf2b7e26f78	Sun Nov 24 2024 20:49:44 GMT+0000 (Greenwich Mean Time)
Description	**Storage (volumes)**	**Key pair name**	**Request Spot Instances**
-	-	year-4-key-2024	No

View details in the launch template console

Network

VPC configuration



vpc-0fd35ec45c2965cf2 / medi-advice-private-network-vpc

Details Info

VPC ID	State	Block Public Access	DNS hostnames
vpc-0fd35ec45c2965cf2	Available	Off	Enabled
DNS resolution	Tenancy	DHCP option set	Main route table
Enabled	Default	dopt-06a3e4538d034b6a8	rtb-0c03af47753f326ae
Main network ACL	Default VPC	IPv4 CIDR	IPv6 pool
acl-0c72f0cc8fed66134	No	10.0.0.0/16	–
IPv6 CIDR (Network border group)	Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID
–	Disabled	Failed to load rule groups	689698388076

Resource map Info

VPC Show details
Your AWS virtual network
medi-advice-private-network-vpc

Subnets (4)
Subnets within this VPC
us-east-1a
medi-advice-private-network-subn...
medi-advice-private-network-subn...
us-east-1b
medi-advice-private-network-subn...
medi-advice-private-network-subn...

Route tables (4)
Route network traffic to resources
medi-advice-private-network-rtb-priv...
medi-advice-private-network-rtb-public
rtb-0c03af47753f326ae
medi-advice-private-network-rtb-priv...

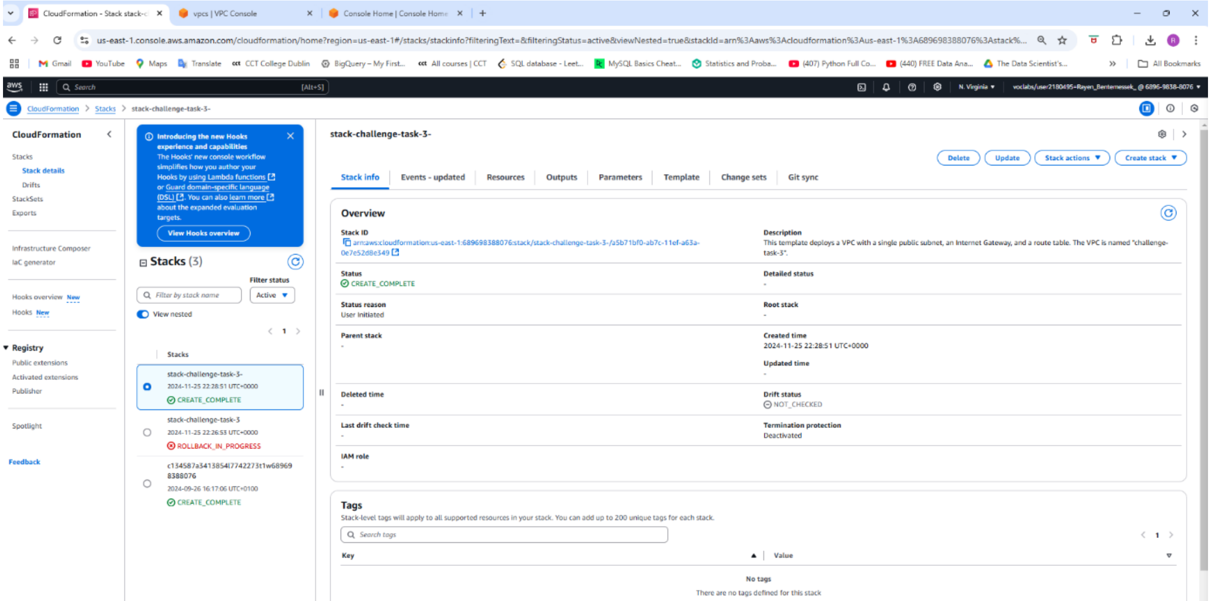
Network connections (2)
Connections to other networks
medi-advice-private-network-igw
medi-advice-private-network-vpce-s3

Load balancer when combined with an autoscaler is very helpful for Medi-Advice as it prevents manual scaling that is currently done and replaces it with auto scaling. In the previous example we put 3 desired instances so that would be the normal state. When more traffic comes in, The ASG scales up and changes the number of instances to 5 instances. Once this rush is over, it goes back to the normal state. If the website is having very little traffic, late night times for example, it scales down to 2 instances. This approach will benefit Medi-Advice by using only the needed number of instances contributing to reducing the costs.

Challenge TASK 3:

A1:

This screenshot below shows the creation of the stack, which was a very straightforward process with no assistance apart from the template, shown in the next screenshot.



Introducing the new Hooks experience and capabilities
The Hooks' new console workflow simplifies how you author your Hooks by using Lambda functions or Guard domain-specific language (DSL). You can also learn more about the expanded evaluation targets.

Stacks (3)

Stacks

stack-challenge-task-3-
2024-11-25 22:28:51 UTC+0000
CREATE_COMPLETE

stack-challenge-task-3
2024-11-25 22:26:53 UTC+0000
ROLLBACK_IN_PROGRESS

c134587a341385417742273t1w68969 8388076
2024-09-26 16:17:06 UTC+0100
CREATE_COMPLETE

stack-challenge-task-3-

Overview

Stack ID
arn:aws:cloudformation:us-east-1:689698388076:stack/stack-challenge-task-3-/a5b71bf0-ab7c-11ef-a63a-0e7e52d8e349

Description
This template deploys a VPC with a single public subnet, an Internet Gateway, and a route table. The VPC is named "challenge-task-3".

Status
CREATE_COMPLETE

Detailed status
-

Status reason
User Initiated

Root stack
-

Parent stack
-

Created time
2024-11-25 22:28:51 UTC+0000

Updated time
-

Deleted time
-

Drift status
NOT_CHECKED

Last drift check time
-

Termination protection
Deactivated

IAM role
-

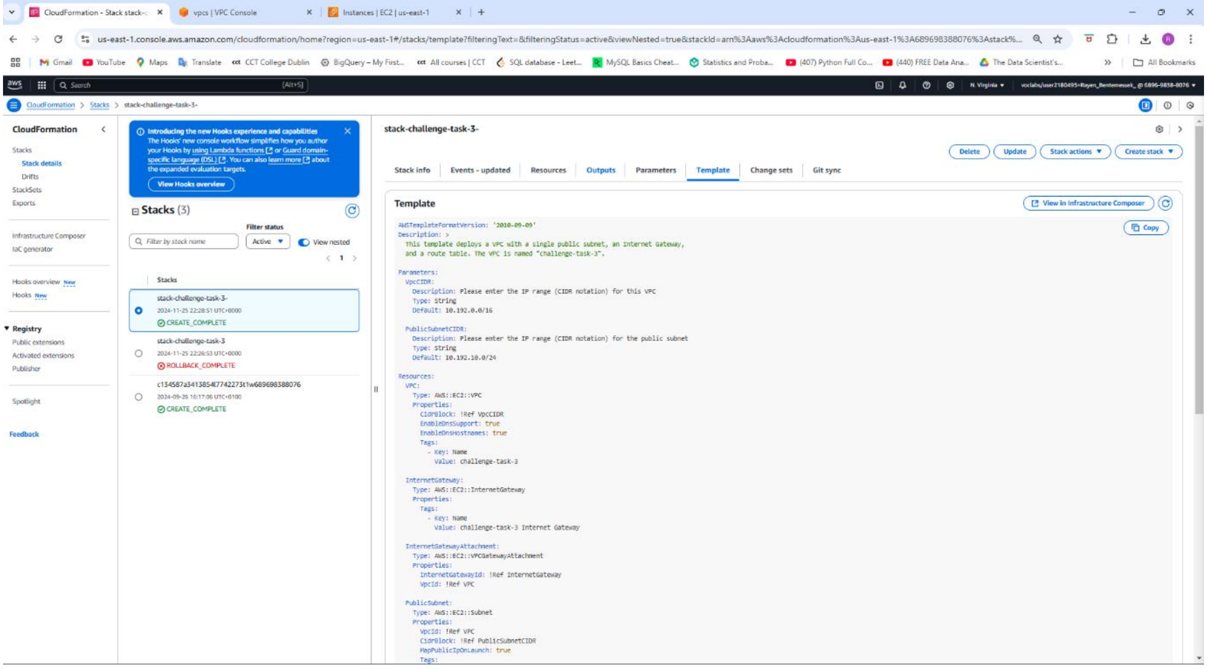
Tags
Stack-level tags will apply to all supported resources in your stack. You can add up to 200 unique tags for each stack.

Key | Value

No tags
There are no tags defined for this stack

I used the AWS template (docs.aws.amazon.com, n.d.) where I did few changes, such as reducing the unused subnets, the rest remains the same.

As we can see the yaml version is very human-readable, it highlights the VPC CIDR notation in the parameters section, which was 10.192.0.0/16 (unchanged), then it selects the IP range of the public, you can also change the name of the VPC as I did, then it adds the internet gateway and it attaches it to the VPC with the public subnet and updates the route table in the resources section, which is pretty similar to the manual approach, then finally there the outputs section which contains description for the different components.



stack-challenge-task-3-

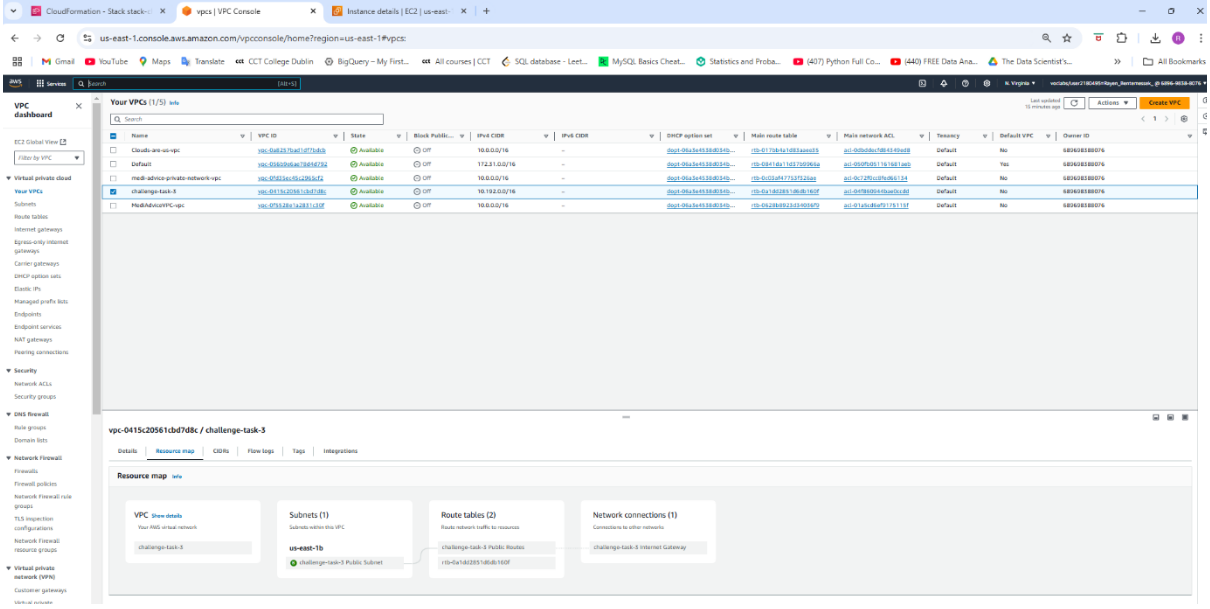
Stacks (3)

Stacks
stack-challenge-task-3- 2024-11-25 22:28:51 UTC+0000 CREATE_COMPLETE
stack-challenge-task-3 2024-11-25 22:26:53 UTC+0000 ROLLBACK_COMPLETE
c134587a3413854l7742273t1w689698388076 2024-09-26 16:17:06 UTC+0100 CREATE_COMPLETE

Template

```
AWSTemplateFormatVersion: '2010-09-09'
Description: >
  This template deploys a VPC with a single public subnet, an Internet Gateway,
  and a route table. The VPC is named "challenge-task-3".

Parameters:
  VpcCIDR:
    Description: Please enter the IP range (CIDR notation) for this VPC
    Type: String
    Default: 10.192.0.0/16

  PublicSubnetCIDR:
    Description: Please enter the IP range (CIDR notation) for the public subnet
    Type: String
    Default: 10.192.10.0/24

Resources:
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCIDR
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: challenge-task-3

  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: challenge-task-3 Internet Gateway

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC

  PublicSubnet:
    Type: AWS::EC2::Subnet
    Properties:
      VpcId: !Ref VPC
      CidrBlock: !Ref PublicSubnetCIDR
      MapPublicIpOnLaunch: true
      Tags:
```

The following screenshot shows the creation of the VPC



Your VPCs (1/5)

	Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL	Tenancy	Default VPC	Owner ID
☐	Clouds-are-us-vpc	vpc-0a8257bad1df7bdcb	Available	Off	10.0.0.0/16	–	dopt-06a3e4538d034b...	rtb-017bb4a1d83aaee35	acl-0dbddecfd84349ed8	Default	No	689698388076
☐	Default	vpc-056b9e6ae78d4d792	Available	Off	172.31.0.0/16	–	dopt-06a3e4538d034b...	rtb-0841da11d37b9966a	acl-050fb051161681aeb	Default	Yes	689698388076
☐	medi-advice-private-network-vpc	vpc-0fd35ec45c2965cf2	Available	Off	10.0.0.0/16	–	dopt-06a3e4538d034b...	rtb-0c03af47753f326ae	acl-0c72f0cc8fed66134	Default	No	689698388076
☑	challenge-task-3	vpc-0415c20561cbd7d8c	Available	Off	10.192.0.0/16	–	dopt-06a3e4538d034b...	rtb-0a1dd2851d6db160f	acl-04f860944bae0ccdd	Default	No	689698388076
☐	MediAdviceVPC-vpc	vpc-0f5528e1a2831c30f	Available	Off	10.0.0.0/16	–	dopt-06a3e4538d034b...	rtb-0628b8923d34036f9	acl-01a5cd6ef9175115f	Default	No	689698388076

vpc-0415c20561cbd7d8c / challenge-task-3

Resource map

VPC: challenge-task-3

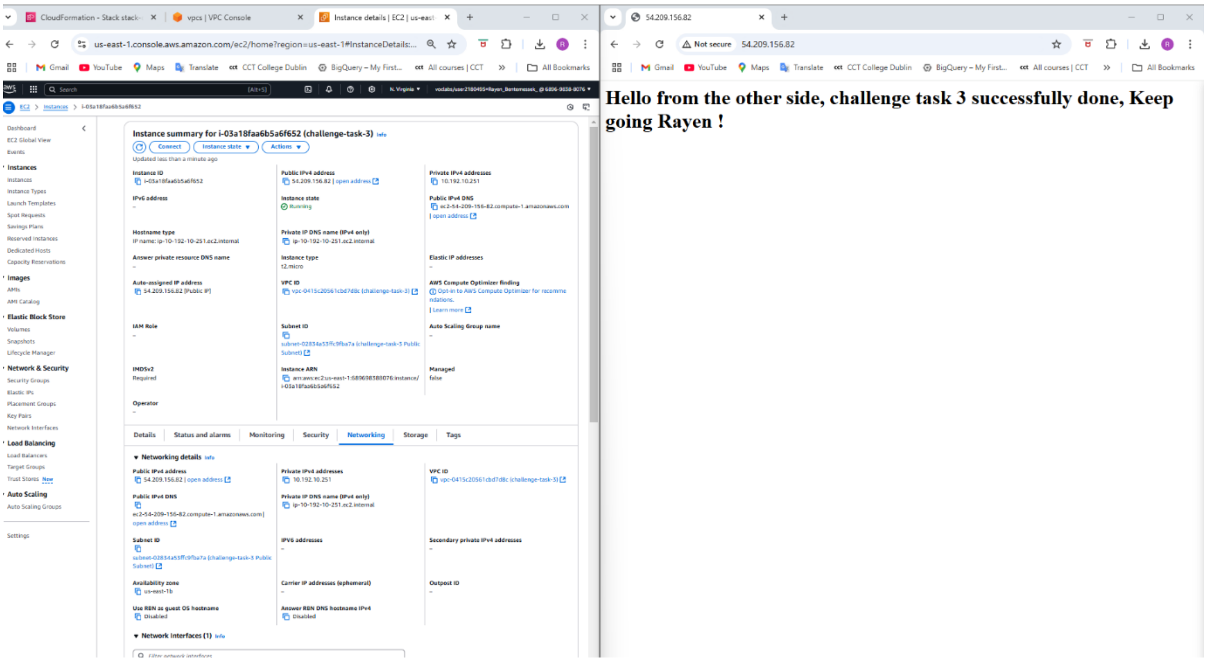
Subnets (1): us-east-1b, challenge-task-3 Public Subnet

Route tables (2): challenge-task-3 Public Routes, rtb-0a1dd2851d6db160f

Network connections (1): challenge-task-3 Internet Gateway

A2:

And this is the proof that the EC2 instance works perfectly on the created VPC



Instance summary for i-03a18faa6b5a6f652 (challenge-task-3)

Updated less than a minute ago

Instance ID: i-03a18faa6b5a6f652

Public IPv4 address: 54.209.156.82 | open address

Private IPv4 addresses: 10.192.10.251

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-54-209-156-82.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-10-192-10-251.ec2.internal

Private IP DNS name (IPv4 only): ip-10-192-10-251.ec2.internal

Answer private resource DNS name: –

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 54.209.156.82 [Public IP]

VPC ID: vpc-0415c20561cbd7d8c (challenge-task-3)

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-02834a53ffc9fba7a (challenge-task-3 Public Subnet)

Auto Scaling Group name: –

IMDSv2: Required

Instance ARN: arn:aws:ec2:us-east-1:689698388076:instance/i-03a18faa6b5a6f652

Managed: false

Operator: –

Networking details

Public IPv4 address: 54.209.156.82 | open address

Private IPv4 addresses: 10.192.10.251

VPC ID: vpc-0415c20561cbd7d8c (challenge-task-3)

Public IPv4 DNS: ec2-54-209-156-82.compute-1.amazonaws.com | open address

Private IP DNS name (IPv4 only): ip-10-192-10-251.ec2.internal

Subnet ID: subnet-02834a53ffc9fba7a (challenge-task-3 Public Subnet)

IPV6 addresses: –

Secondary private IPv4 addresses: –

Availability zone: us-east-1b

Carrier IP addresses (ephemeral): –

Outpost ID: –

Use RBN as guest OS hostname: Disabled

Answer RBN DNS hostname IPv4: Disabled

Network interfaces (1)

54.209.156.82

Hello from the other side, challenge task 3 successfully done, Keep going Rayen !

B1: Advantage of using a CloudFormation to create a custom VPC:

Creating a custom VPC using CloudFormation is highly beneficial as it can be saved through different templates through stacks instead of manually creating them, this process allows plenty of customisation and automation as it only takes few seconds to do so. The bigger the infrastructure becomes, the harder it is to modify VPC's manually; by grouping them under different templates, it keeps work more organized and efficient as it literally text files whether it is JSON or YAML (GeeksforGeeks, 2023).

B2: Example of how beneficial IaC can be:

In large companies, with different departments, it is highly recommended to use infrastructure as code, as it relieves developers from spending hours of deploying and troubleshooting VPC's, because all of that can be replaced with few lines of code. Also, while implementing the stack, I've seen that it's connected to an S3 bucket which adds more benefits regarding storage and access from anywhere, also I've seen uploading from git which allows version control and collaboration among the team making it easy to track the changes and restore previous versions if required, this will be overly complicated if it's done manually. In addition to that, you can implement the templates inside the AWS as well. So, you've got the option to do it this way all in one place -AWS in our case- for the whole team, that is going to relieve developers and make them focus on working on other projects rather than solving unnecessary technical issues.

References:

Amazon Web Services, Inc. (2019). *Amazon Textract | Extract Text & Data | AWS*. [online] Available at: https://aws.amazon.com/textract/.

Krassowski, T. (2022). *Highly Available 3-Tier Architecture Web Application in AWS*. [online] Medium. Available at: https://awstip.com/highly-available-3-tier-architecture-web-application-in-aws-2cbe90580f57.

SmartDraw (2019). *SmartDraw - Create Flowcharts, Floor Plans, and Other Diagrams on Any Device*. [online] Smartdraw.com. Available at: https://www.smartdraw.com/.

docs.aws.amazon.com. (n.d.). *Working with read replicas - Amazon Relational Database Service*. [online] Available at: https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ReadRepl.html.

AWS (n.d.). *Design Principles - AWS Well-Architected Framework*. [online] docs.aws.amazon.com. Available at: https://docs.aws.amazon.com/wellarchitected/latest/framework/oe-design-principles.html.

docs.aws.amazon.com. (n.d.). *Design Principles - AWS Well-Architected Framework*. [online] Available at: https://docs.aws.amazon.com/wellarchitected/latest/framework/sec-design.html.

docs.aws.amazon.com. (n.d.). *Design Principles - AWS Well-Architected Framework*. [online] Available at: https://docs.aws.amazon.com/wellarchitected/latest/framework/perf-dp.html.

docs.aws.amazon.com. (n.d.). *Design Principles - AWS Well-Architected Framework*. [online] Available at: https://docs.aws.amazon.com/wellarchitected/latest/framework/cost-dp.html.

Amazon.com. (2024). *Design principles - AWS Well-Architected Framework*. [online] Available at: https://docs.aws.amazon.com/wellarchitected/latest/framework/sus-design-principles.html.

Amazon Web Services (2023). *What Is Amazon VPC? - Amazon Virtual Private Cloud*. [online] Amazon.com. Available at: https://docs.aws.amazon.com/vpc/latest/userguide/what-is-amazon-vpc.html.

Appstudio (2023). *This Is How Instagram Is Delighting 2 Billion+ Users: Decoding System Architecture & Design*. [online] AppStudio. Available at: https://www.appstudio.ca/blog/this-is-how-instagram-is-delighting-2-billion-users-decoding-system-architecture-design/.

Akamai. (2020). *What Is a CDN (Content Delivery Network)? | How Do CDNs Work? | Akamai*. [online] Available at: https://www.akamai.com/glossary/what-is-a-cdn.

CDN Performance | Cloudflare. (n.d.). *Cloudflare*. [online] Available at: https://www.cloudflare.com/learning/cdn/performance/.

docs.aws.amazon.com. (n.d.). *AWS CloudFormation VPC template - AWS CodeBuild*. [online] Available at: https://docs.aws.amazon.com/codebuild/latest/userguide/cloudformation-vpc-template.html.

GeeksforGeeks (2023). *Build a VPC with CloudFormation*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/build-a-vpc-with-cloudformation/.