رواد مصر الرقمية

وزارة الاتصالات
وتكنولوجيا المعلومات

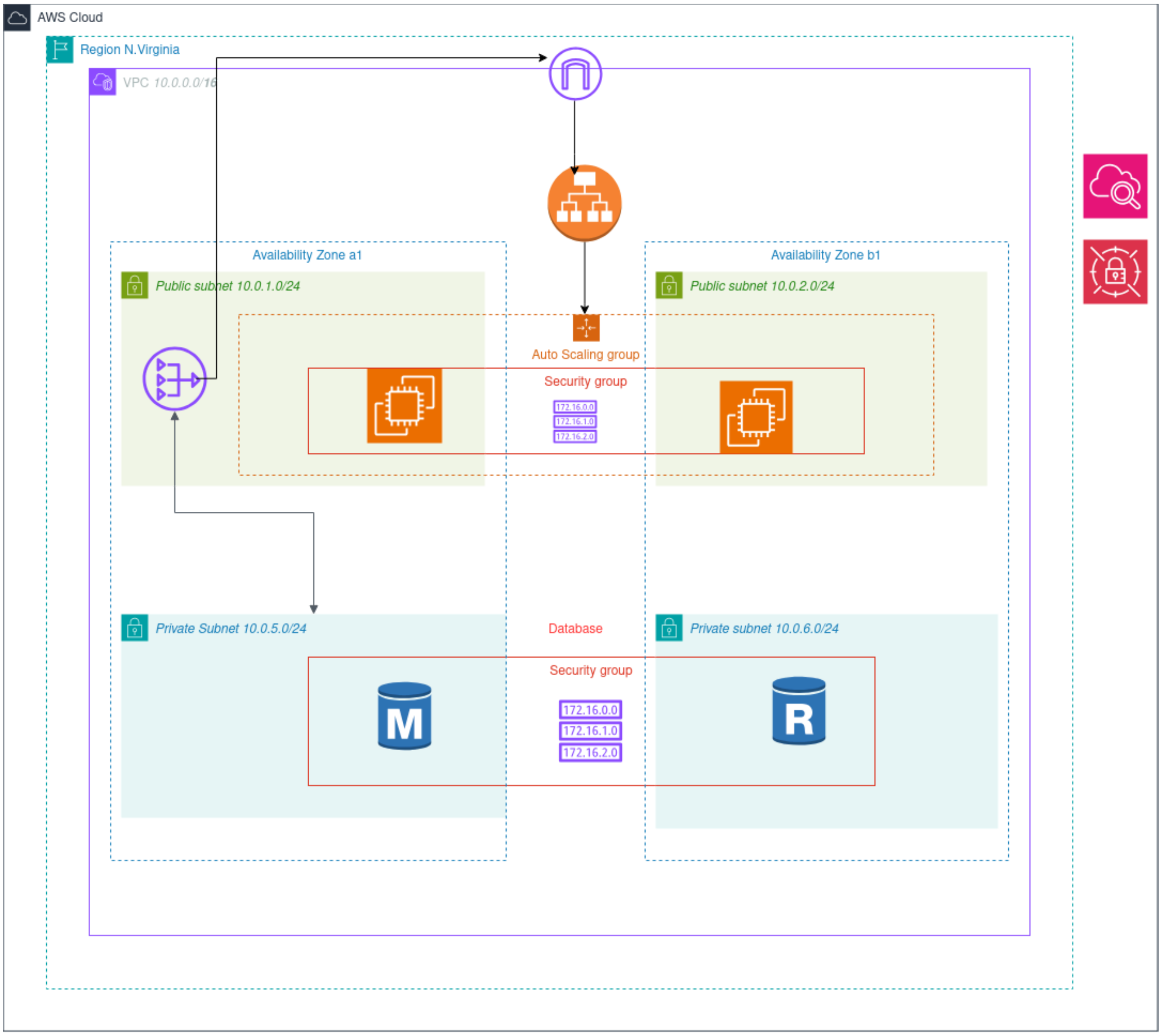
Building a Highly Available, Scalable Web Application Project

- Name: Menna Allah Ahmed Ibarhim Eldosky
- Group: CAI1_SWD8_M2d
- Email: mennaahmed0640@gmail.com
- Cell Phone: 01092593895 or 01151901449
- LinkedIn:www.linkedin.com/in/menna-ahmed-620151219

Requirements:

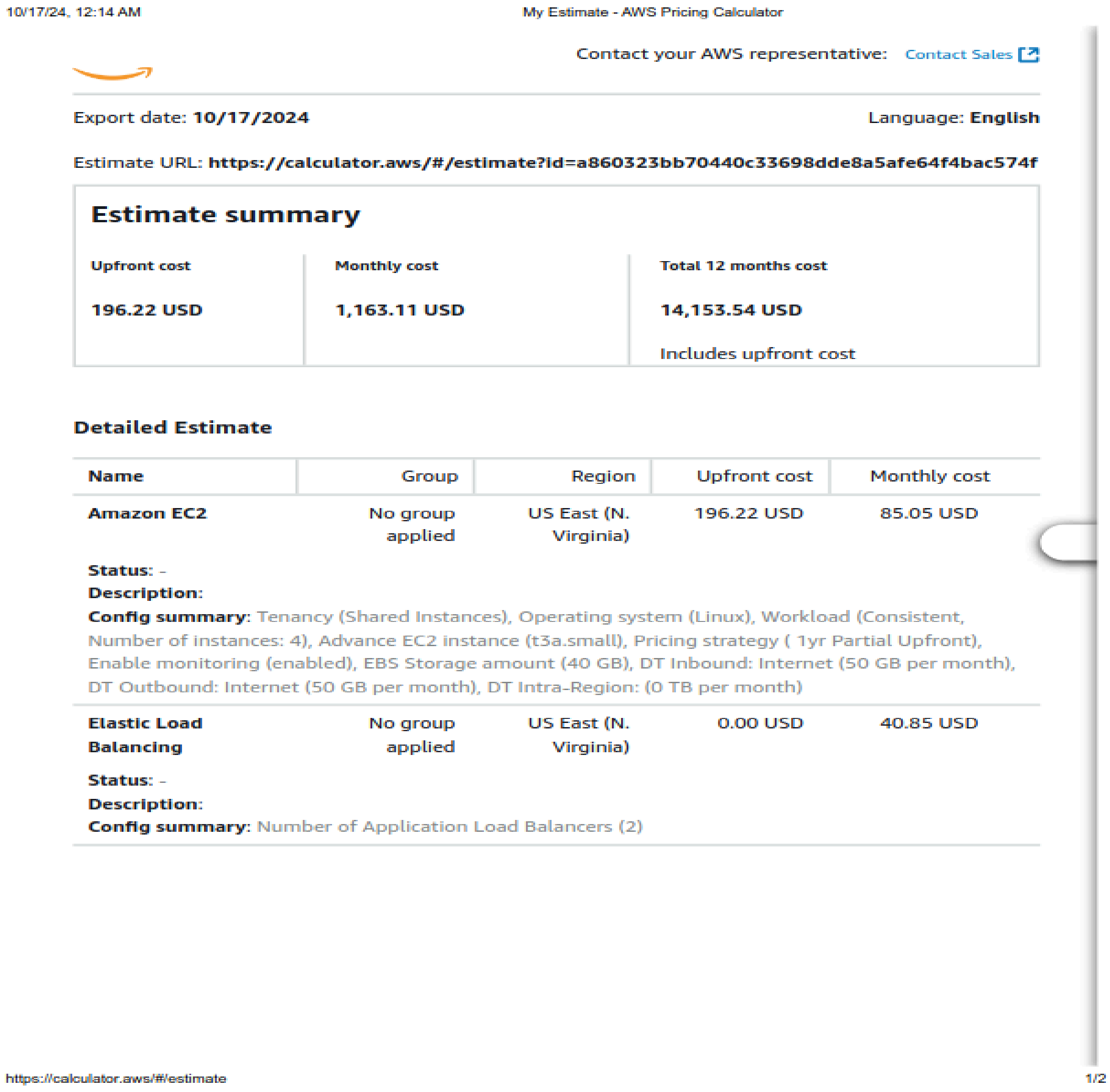
- Create an architectural diagram to depict various AWS services and their interactions with each other.
- Estimate the cost of using services by using the AWS Pricing Calculator.
- Deploy a functional web application that runs on a single virtual machine and is backed by a relational database.
- Architect a web application to separate layers of the application, such as the web server and database.
- Create a virtual network that is configured appropriately to host a web application that is publicly accessible and secure.
- Deploy a web application with the load distributed across multiple web servers.
- Configure the appropriate network security settings for the web servers and database.
- Implement high availability and scalability in the deployed solution.
- Configure access permissions between AWS services.

1- Architecture Diagram and Cost Estimation:



2-Estimate Costs:

A cost projection for operating the architecture on AWS for a 12-month period is generated using the AWS Pricing Calculator to ensure accurate budgeting.



10/17/24, 12:14 AM My Estimate - AWS Pricing Calculator

Contact your AWS representative: Contact Sales

Export date: **10/17/2024** Language: **English**

Estimate URL: **https://calculator.aws/#/estimate?id=a860323bb70440c33698dde8a5afe64f4bac574f**

Estimate summary

Upfront cost	Monthly cost	Total 12 months cost
196.22 USD	1,163.11 USD	14,153.54 USD Includes upfront cost

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon EC2	No group applied	US East (N. Virginia)	196.22 USD	85.05 USD

Status: -
Description:
Config summary: Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 4), Advance EC2 instance (t3a.small), Pricing strategy (1yr Partial Upfront), Enable monitoring (enabled), EBS Storage amount (40 GB), DT Inbound: Internet (50 GB per month), DT Outbound: Internet (50 GB per month), DT Intra-Region: (0 TB per month)

Elastic Load Balancing	No group applied	US East (N. Virginia)	0.00 USD	40.85 USD

Status: -
Description:
Config summary: Number of Application Load Balancers (2)

https://calculator.aws/#/estimate 1/2

Abstract:

The purpose of this project is to design, build, and deploy a highly available and scalable web application using Amazon Web Services (AWS). The project involves implementing a multi-tier architecture, where the web application and database are decoupled to ensure better performance and flexibility. In the initial phase, the project creates a virtual private cloud (VPC) with public and private subnets across two Availability Zones, ensuring redundancy and fault tolerance. The web application is hosted on an Amazon EC2 instance, while the database is provisioned separately using Amazon RDS to manage and store data efficiently.

To improve the security and efficiency of communication between the application and database, AWS Secrets Manager is used to manage database credentials securely. AWS Cloud9 provides a development environment for managing application code, running AWS CLI commands, and migrating databases. Auto Scaling and Load Balancer services are configured to ensure the application scales seamlessly based on traffic, offering both high availability and cost optimization.

This project demonstrates the use of cloud-native services for building a scalable, secure, and robust web application architecture, suitable for real-world deployment scenarios.

I. Introduction:

1. Overview of the project

This project focuses on developing a highly available, scalable, and secure web application hosted on the Amazon Web Services (AWS) cloud platform. The application architecture follows a 3-tier design with web, application, and database layers. The project is built using AWS services like EC2 for hosting the web application, RDS for managing the database, and Auto Scaling for handling traffic spikes.

2. Objectives of the solution

The primary objectives of this project include:

- **High Availability:** Ensure the web application is accessible even in case of failure or maintenance in one availability zone by distributing the resources across multiple zones.

- **Scalability:** Automatically scale the application based on traffic demand, ensuring the performance remains optimal during peak loads.

- **Security:** Protect the web application and database using AWS security features like security groups, private subnets, and IAM roles.
- **Cost-effectiveness:** Utilize AWS services efficiently to minimize costs without sacrificing performance or availability.

- Automation: Use Infrastructure as Code (IaC) principles for automating the deployment and scaling of the application.

3. Functional Requirements

- **Web Application Hosting:** The application should be hosted on an EC2 instance, which serves dynamic content to users via a web interface.

- **Database Management:** The backend database must be hosted separately on Amazon RDS (MySQL engine) to decouple the application components and improve performance.

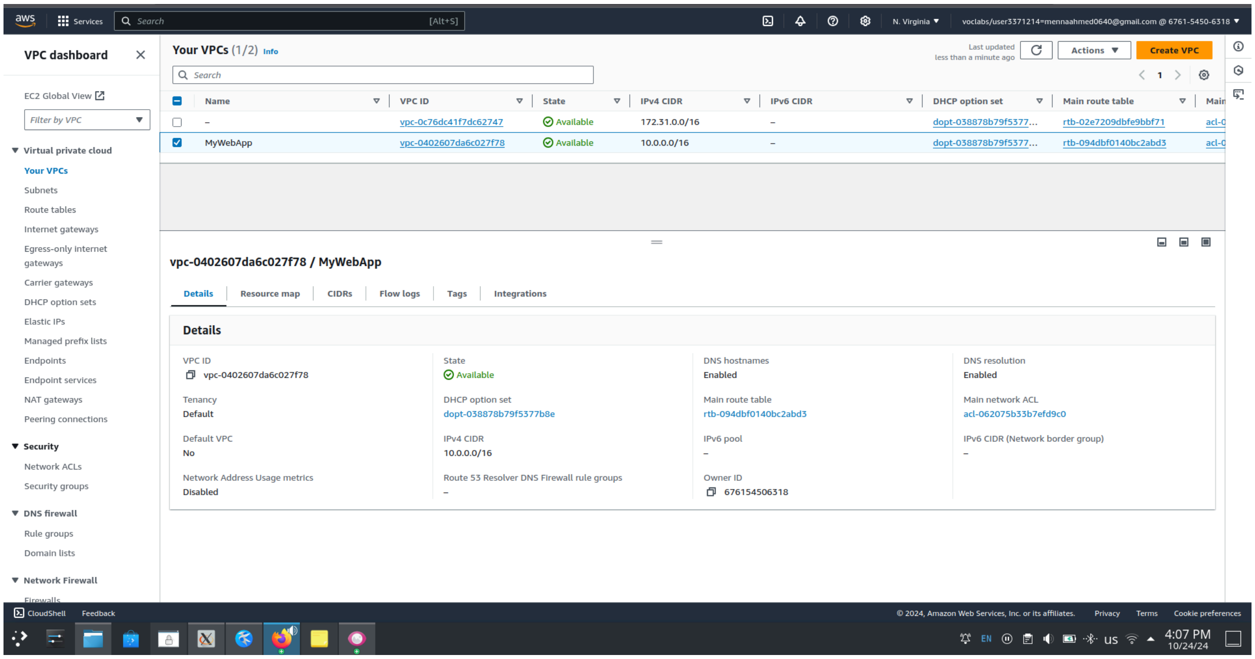
- **Scalability:** The system must support auto-scaling to handle increased loads dynamically and reduce resources during low usage.

- **Security Controls:** Implement security controls to restrict access to the database from the public internet, allowing only the web application to interact with it.

- **Monitoring and Logging:** Utilize AWS CloudWatch for monitoring application performance and health, and set up logging for tracking access and performance metrics.

2-Implementation Steps:

A.VPC



Your VPCs (1/2) Info

	Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Mai
□	–	vpc-0c76dc41f7dc62747	Available	172.31.0.0/16	–	dopt-038878b79f5377…	rtb-02e7209dbfe9bbf71	acl-0
☑	MyWebApp	vpc-0402607da6c027f78	Available	10.0.0.0/16	–	dopt-038878b79f5377…	rtb-094dbf0140bc2abd3	acl-0

vpc-0402607da6c027f78 / MyWebApp

Details | Resource map | CIDRs | Flow logs | Tags | Integrations

Details

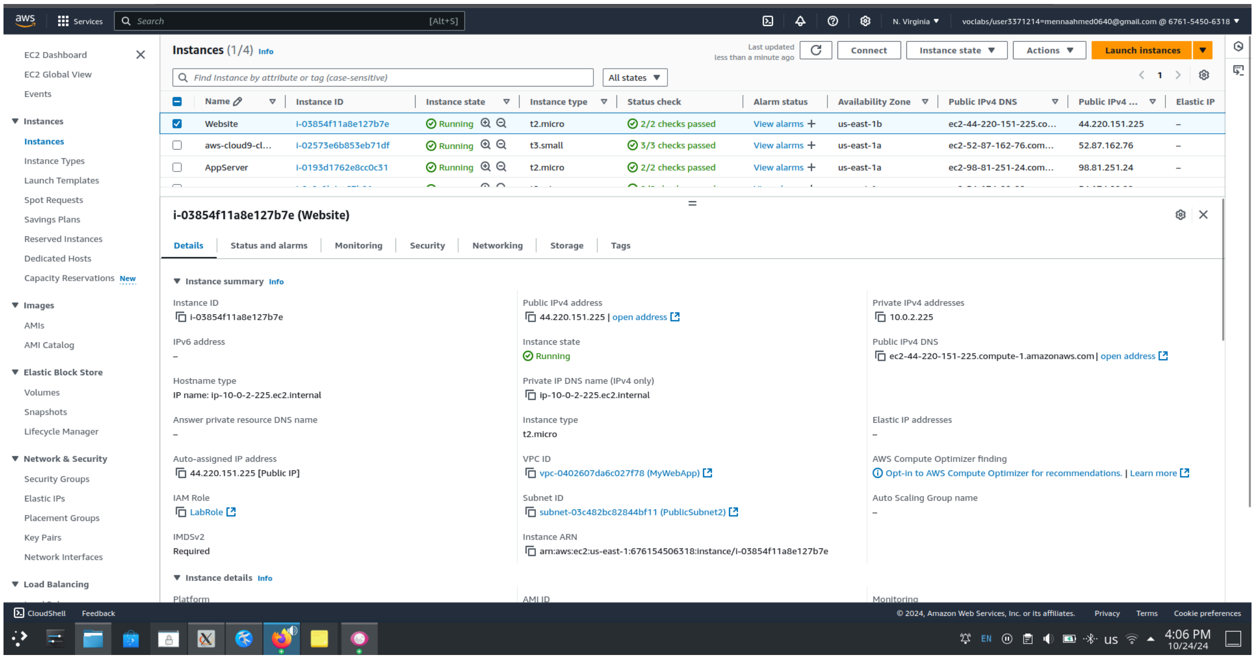
VPC ID: vpc-0402607da6c027f78
Tenancy: Default
Default VPC: No
Network Address Usage metrics: Disabled
State: Available
DHCP option set: dopt-038878b79f5377b8e
IPv4 CIDR: 10.0.0.0/16
Route 53 Resolver DNS Firewall rule groups: –
DNS hostnames: Enabled
Main route table: rtb-094dbf0140bc2abd3
IPv6 pool: –
Owner ID: 676154506318
DNS resolution: Enabled
Main network ACL: acl-062075b33b7efd9c0
IPv6 CIDR (Network border group): –

This screenshot shows the VPC (Virtual Private Cloud) dashboard in AWS. The selected VPC is named "MyWebApp," and its VPC ID is displayed. Key details include:

- **VPC CIDR block**: 10.0.0.0/16, defining the IP address range for the network.
- **State:** Available, indicating the VPC is active.
- **DNS resolution**: Enabled, allowing resources within the VPC to resolve domain names.
- **Main route table**: Displays the ID of the route table managing traffic within the VPC.
- **DHCP option set:** A set of rules to assign network configuration to instances in the VPC.

This VPC is a custom network where your web application and database will reside, ensuring isolation and security within AWS.

B. **EC2 Instance**



Instances (1/4) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
☑	Website	i-03854f11a8e127b7e	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-44-220-151-225.co...	44.220.151.225	–
□	aws-cloud9-cl...	i-02573e6b853eb71df	Running	t3.small	3/3 checks passed	View alarms +	us-east-1a	ec2-52-87-162-76.com...	52.87.162.76	–
□	AppServer	i-0193d1762e8cc0c31	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-98-81-251-24.com...	98.81.251.24	–

i-03854f11a8e127b7e (Website)

Instance summary Info

Instance ID: i-03854f11a8e127b7e

Public IPv4 address: 44.220.151.225 | open address

Private IPv4 addresses: 10.0.2.225

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-44-220-151-225.compute-1.amazonaws.com | open address

Hostname type: IP name: ip-10-0-2-225.ec2.internal

Private IP DNS name (IPv4 only): ip-10-0-2-225.ec2.internal

Answer private resource DNS name: –

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 44.220.151.225 [Public IP]

VPC ID: vpc-0402607da6c027f78 (MyWebApp)

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

IAM Role: LabRole

Subnet ID: subnet-03c482bc82844bf11 (PublicSubnet2)

Auto Scaling Group name: –

IMDSv2: Required

Instance ARN: arn:aws:ec2:us-east-1:676154506318:instance/i-03854f11a8e127b7e

Instance details Info

Platform

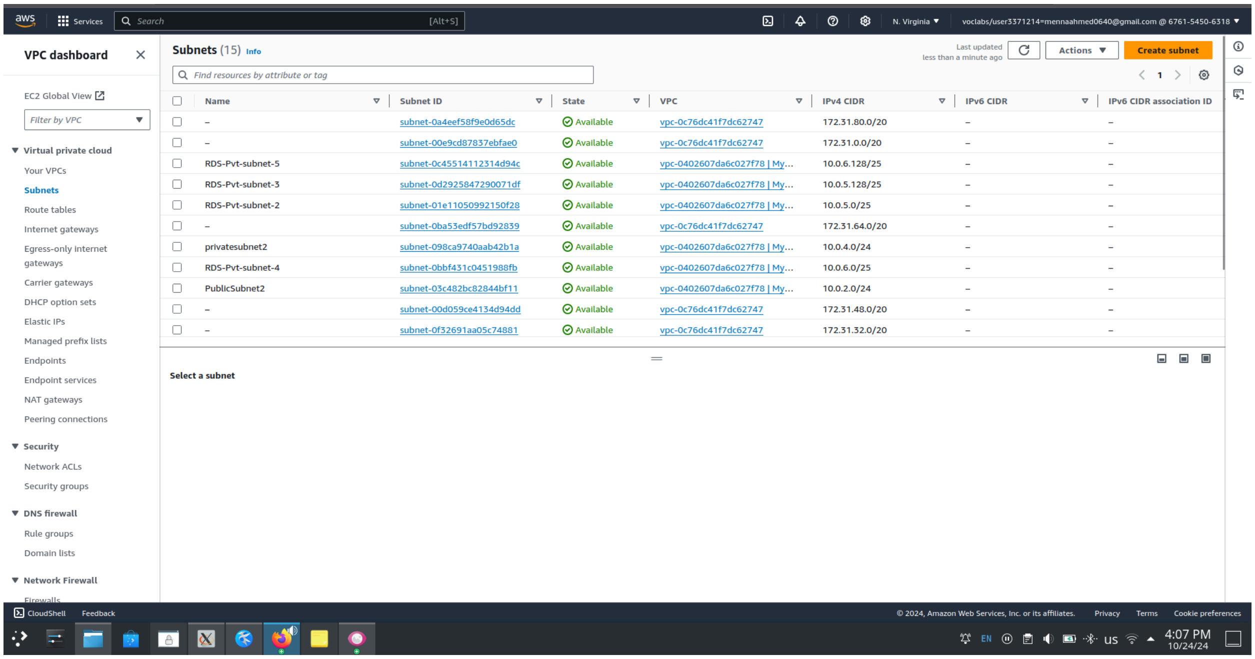
AMI ID

Monitoring

This screenshot shows details of an EC2 instance named "Website" running in AWS. Key information includes:

- **Instance ID:** I-03854f11a8e127b7e
- **Instance state:** Running (indicating it is currently active)
- **Instance type**: t2.micro (a small instance type used for low-cost general-purpose workloads)
- **Private IP**: 10.0.2.225 (used for internal communication within the VPC)
- **Public IP:** 44.220.151.225 (used to access the instance from the internet)
- **VPC ID**: vpc-0402607da6c027f78 (shows the instance belongs to the "MyWebApp" VPC)
- **Subnet ID**: subnet-03c482bc8284bff11 (indicates which subnet the instance is deployed in)
- **IAM Role:** LabRole (this role provides permissions for the instance to interact with AWS services)

C. Subnets



Subnets (15) Info

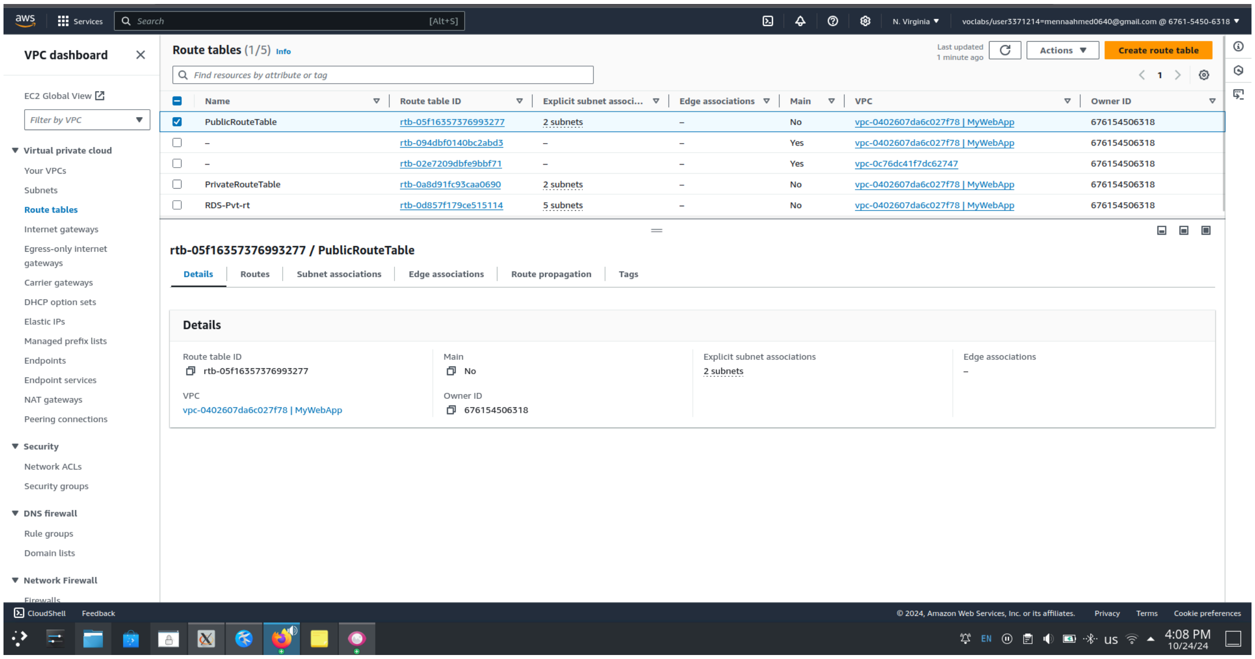
Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID
–	subnet-0a4eef58f9e0d65dc	Available	vpc-0c76dc41f7dc62747	172.31.80.0/20	–	–
–	subnet-00e9cd87837ebfae0	Available	vpc-0c76dc41f7dc62747	172.31.0.0/20	–	–
RDS-Pvt-subnet-5	subnet-0c45514112314d94c	Available	vpc-0402607da6c027f78 | My...	10.0.6.128/25	–	–
RDS-Pvt-subnet-3	subnet-0d2925847290071df	Available	vpc-0402607da6c027f78 | My...	10.0.5.128/25	–	–
RDS-Pvt-subnet-2	subnet-01e11050992150f28	Available	vpc-0402607da6c027f78 | My...	10.0.5.0/25	–	–
–	subnet-0ba53edf57bd92839	Available	vpc-0c76dc41f7dc62747	172.31.64.0/20	–	–
privatesubnet2	subnet-098ca9740aab42b1a	Available	vpc-0402607da6c027f78 | My...	10.0.4.0/24	–	–
RDS-Pvt-subnet-4	subnet-0bbf431c0451988fb	Available	vpc-0402607da6c027f78 | My...	10.0.6.0/25	–	–
PublicSubnet2	subnet-03c482bc82844bf11	Available	vpc-0402607da6c027f78 | My...	10.0.2.0/24	–	–
–	subnet-00d059ce4134d94dd	Available	vpc-0c76dc41f7dc62747	172.31.48.0/20	–	–
–	subnet-0f32691aa05c74881	Available	vpc-0c76dc41f7dc62747	172.31.32.0/20	–	–

Select a subnet

This screenshot shows the list of subnets within your Virtual Private Cloud (VPC) in AWS. Key points to note:

- **Subnet names**: You have subnets like `RDS-Pvt-subnet-3`, `PublicSubnet2`, and `privatesubnet2`, indicating both public and private subnets.
- **Subnet IDs**: Unique identifiers for each subnet (e.g., `subnet-04aef58f9e0d65dc` for the first subnet).
- **State**: All subnets are in the "Available" state, meaning they are ready for use.
- **VPCs**: Most subnets are associated with either your "MyWebApp" VPC (`vpc-0402607da6c027f78`) or another VPC (`vpc-0c76dc41f7dc62747`).
- **IPv4 CIDR blocks**: The IP range assigned to each subnet (e.g., `10.0.0.0/24`, `172.31.80.0/20`). Public subnets typically allow internet traffic, while private subnets are used for internal resources, like databases.

D. Route Tables



Route tables (1/5) Info

	Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC	Owner ID
☑	PublicRouteTable	rtb-05f16357376993277	2 subnets	–	No	vpc-0402607da6c027f78 | MyWebApp	676154506318
□	–	rtb-094dbf0140bc2abd3	–	–	Yes	vpc-0402607da6c027f78 | MyWebApp	676154506318
□	–	rtb-02e7209dbfe9bbf71	–	–	Yes	vpc-0c76dc41f7dc62747	676154506318
□	PrivateRouteTable	rtb-0a8d91fc93caa0690	2 subnets	–	No	vpc-0402607da6c027f78 | MyWebApp	676154506318
□	RDS-Pvt-rt	rtb-0d857f179ce515114	5 subnets	–	No	vpc-0402607da6c027f78 | MyWebApp	676154506318

rtb-05f16357376993277 / PublicRouteTable

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Details

Route table ID: rtb-05f16357376993277

Main: No

Explicit subnet associations: 2 subnets

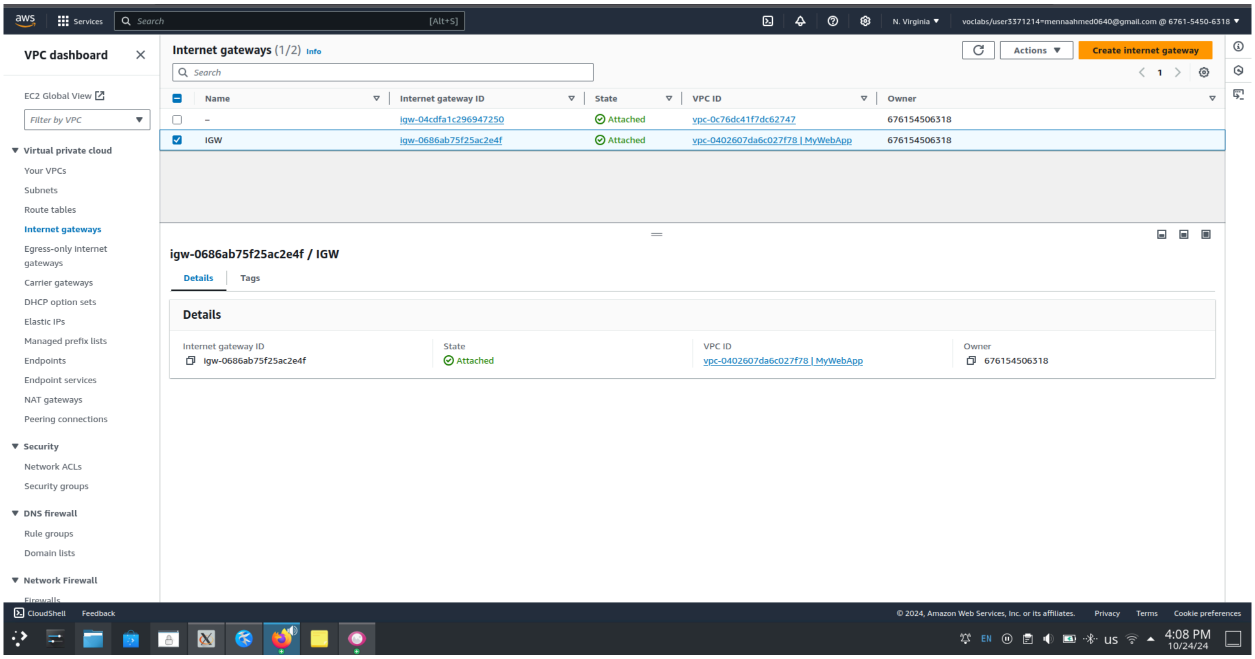
Edge associations: –

VPC: vpc-0402607da6c027f78 | MyWebApp

Owner ID: 676154506318

- **Route Tables:**These are network objects that control how traffic flows within a VPC. The list shows the route table ID, name, number of explicit subnet associations, and other relevant information.
- **Public Route Table:**This is a special type of route table that allows traffic to flow between the VPC and the public internet.
- **Details**:Clicking on a route table likely provides more detailed information about its configuration, including routes, subnet associations, and propagation settings.
- **Actions:**The "Create route table" button suggests that you can create new route tables within this VPC.

E. Internet Gateways



Internet gateways (1/2) Info

☐	Name	Internet gateway ID	State	VPC ID	Owner
☐	–	igw-04cdfa1c296947250	Attached	vpc-0c76dc41f7dc62747	676154506318
☑	IGW	igw-0686ab75f25ac2e4f	Attached	vpc-0402607da6c027f78 | MyWebApp	676154506318

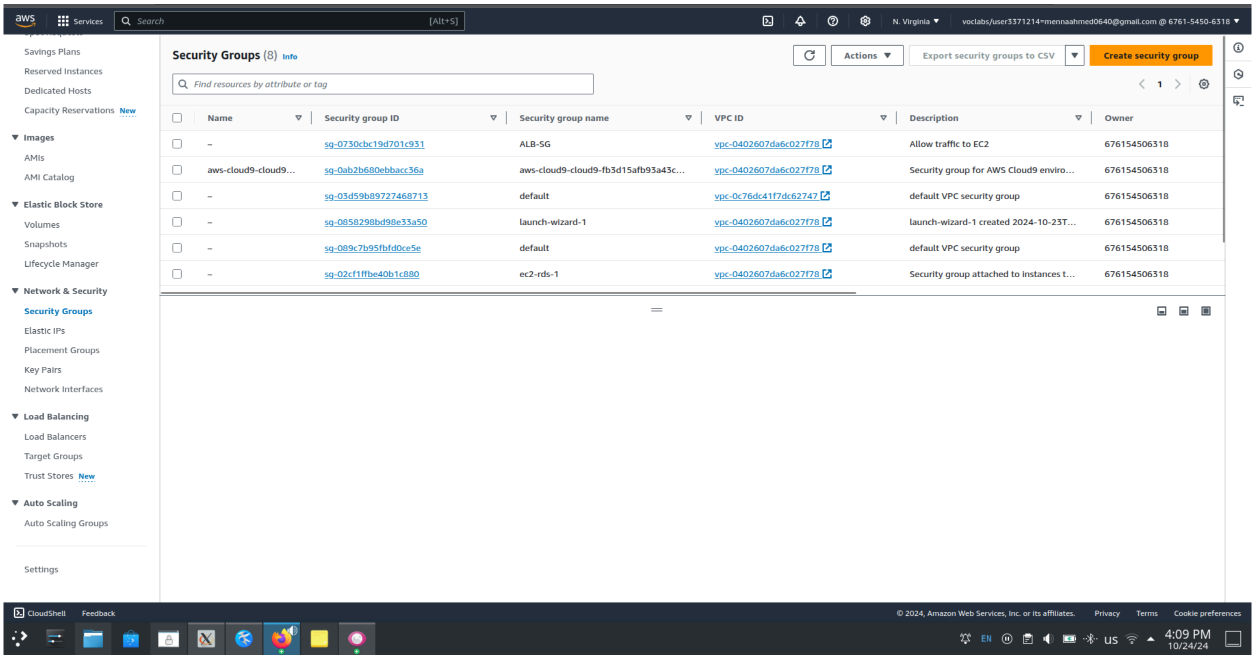
igw-0686ab75f25ac2e4f / IGW

Details

Internet gateway ID	State	VPC ID	Owner
igw-0686ab75f25ac2e4f	Attached	vpc-0402607da6c027f78 | MyWebApp	676154506318

- **Internet Gateways:**These are network objects that allow traffic to flow between your VPC and the public internet. The list shows the internet gateway ID, state, VPC ID, and owner information.
- **Details:**Clicking on an internet gateway likely provides more detailed information about its configuration, including the internet gateway ID, state, VPC ID, and owner.
- **Actions:**The "Create internet gateway" button suggests that you can create new internet gateways within this VPC.

F. Security Groups

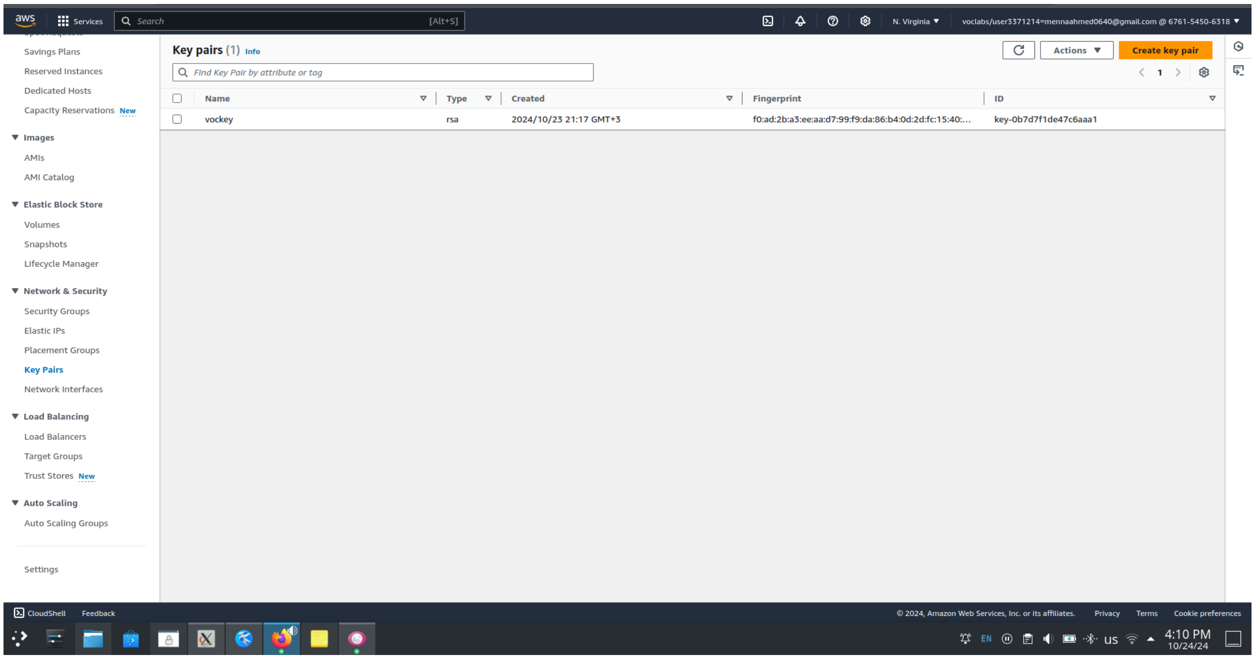


Security Groups (8) Info

Name	Security group ID	Security group name	VPC ID	Description	Owner
–	sg-0730cbc19d701c931	ALB-SG	vpc-0402607da6c027f78	Allow traffic to EC2	676154506318
aws-cloud9-cloud9…	sg-0ab2b680ebbacc36a	aws-cloud9-cloud9-fb3d15afb93a43c…	vpc-0402607da6c027f78	Security group for AWS Cloud9 enviro…	676154506318
–	sg-03d59b89727468713	default	vpc-0c76dc41f7dc62747	default VPC security group	676154506318
–	sg-0858298bd98e33a50	launch-wizard-1	vpc-0402607da6c027f78	launch-wizard-1 created 2024-10-23T…	676154506318
–	sg-089c7b95fbfd0ce5e	default	vpc-0402607da6c027f78	default VPC security group	676154506318
–	sg-02cf1ffbe40b1c880	ec2-rds-1	vpc-0402607da6c027f78	Security group attached to instances t…	676154506318

- **Security Groups:** These are network objects that act as virtual firewalls, controlling inbound and outbound traffic for resources within a VPC. The list shows the security group ID, name, VPC ID, and description.
- **Action:** The "Create security group" button suggests that you can create new security groups within this VPC.

G. Key Pairs



Key pairs (1) Info

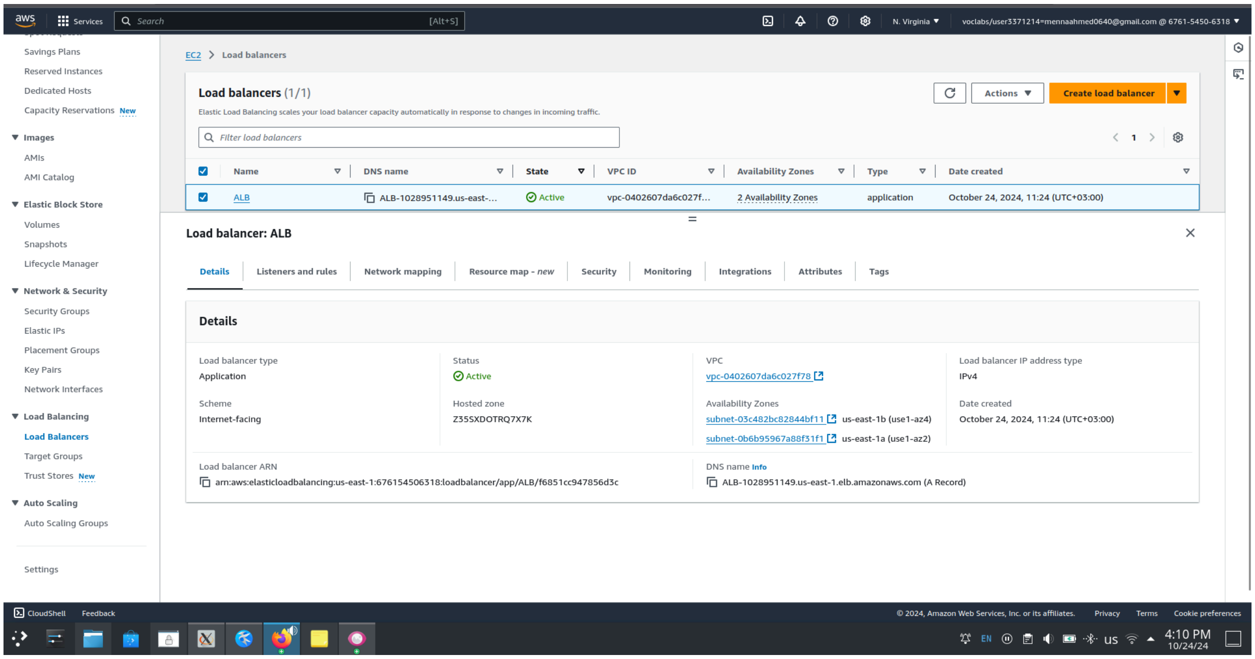
Find Key Pair by attribute or tag

□	Name	Type	Created	Fingerprint	ID
□	vockey	rsa	2024/10/23 21:17 GMT+3	f0:ad:2b:a3:ee:aa:d7:99:f9:da:86:b4:0d:2d:fc:15:40:...	key-0b7d7f1de47c6aaa1

- **Key Pairs:** These are cryptographic key pairs used for authentication and encryption. The list shows the key pair name, type, created date, and fingerprint.
- **Actions:** The "Create key pair" button suggests that you can create new key pairs within this VPC.

Overall, this image provides a basic overview of the key pairs present in the VPC and allows you to manage authentication and encryption for resources within your AWS environment.

H. Load Balancers



Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

☑	Name	DNS name	State	VPC ID	Availability Zones	Type	Date created
☑	ALB	ALB-1028951149.us-east-...	Active	vpc-0402607da6c027f...	2 Availability Zones	application	October 24, 2024, 11:24 (UTC+03:00)

Load balancer: ALB

Details

Load balancer type
Application

Status
Active

VPC
vpc-0402607da6c027f78

Load balancer IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-03c482bc82844bf11 us-east-1b (use1-az4)
subnet-0b6b95967a88f31f1 us-east-1a (use1-az2)

Date created
October 24, 2024, 11:24 (UTC+03:00)

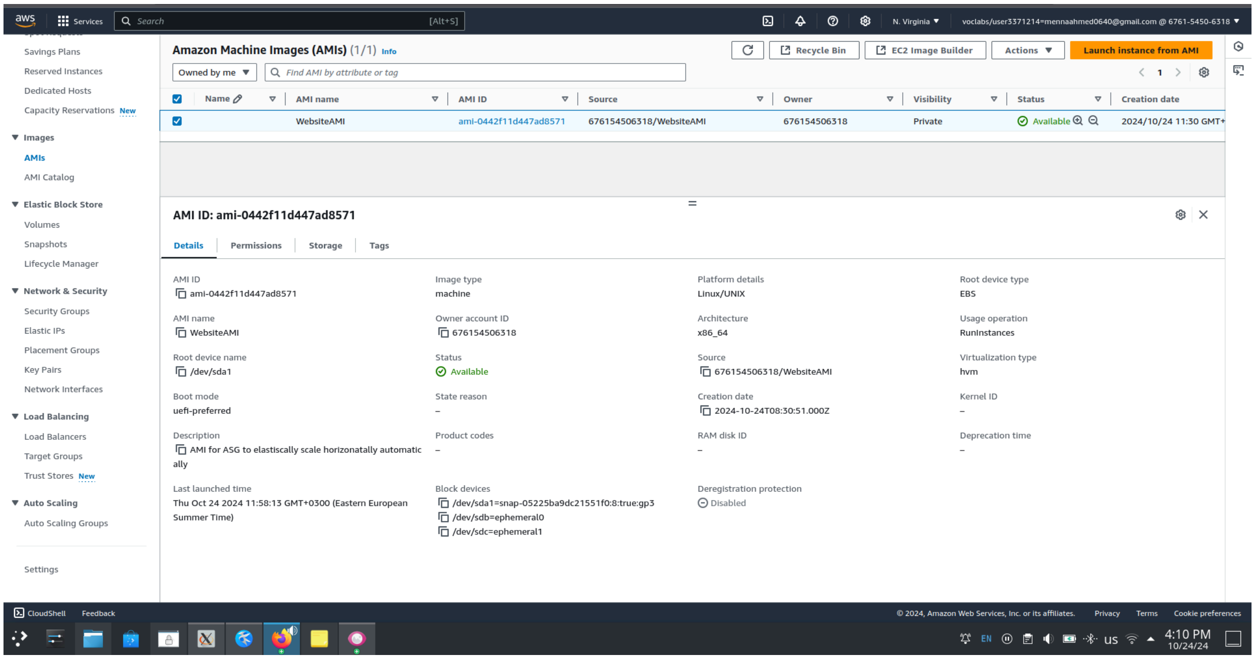
Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:676154506318:loadbalancer/app/ALB/f6851cc947856d3c

DNS name Info
ALB-1028951149.us-east-1.elb.amazonaws.com (A Record)

- **Load Balancers:** These are network objects that distribute incoming traffic across multiple EC2 instances. The list shows the load balancer name, DNS name, state, VPC ID, and other relevant information.
- **Details:** Clicking on a load balancer likely provides more detailed information about its configuration, including listeners, rules, security settings, and health checks.
- **Actions:** The "Create load balancer" button suggests that you can create new load balancers within this VPC.

Overall, this image provides a basic overview of the load balancers present in the VPC and allows you to manage traffic distribution across your EC2 instances.

I. AMIs



Amazon Machine Images (AMIs) (1/1) Info

☑	Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date
☑		WebsiteAMI	ami-0442f11d447ad8571	676154506318/WebsiteAMI	676154506318	Private	✓ Available	2024/10/24 11:30 GMT+

AMI ID: ami-0442f11d447ad8571

AMI ID
ami-0442f11d447ad8571

Image type
machine

Platform details
Linux/UNIX

Root device type
EBS

AMI name
WebsiteAMI

Owner account ID
676154506318

Architecture
x86_64

Usage operation
RunInstances

Root device name
/dev/sda1

Status
✓ Available

Source
676154506318/WebsiteAMI

Virtualization type
hvm

Boot mode
uefi-preferred

State reason
–

Creation date
2024-10-24T08:30:51.000Z

Kernel ID
–

Description
AMI for ASG to elastiscally scale horizonatally automatically

Product codes
–

RAM disk ID
–

Deprecation time
–

Last launched time
Thu Oct 24 2024 11:58:13 GMT+0300 (Eastern European Summer Time)

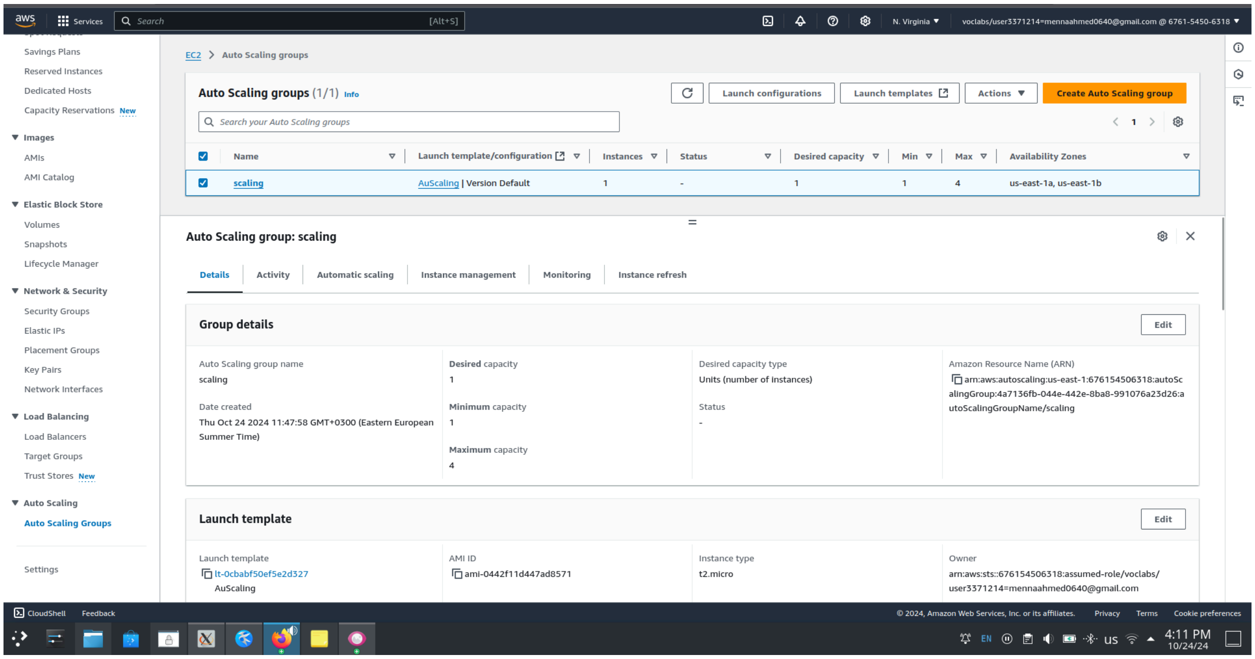
Block devices
/dev/sda1=snap-05225ba9dc21551f0:8:true:gp3
/dev/sdb=ephemeral0
/dev/sdc=ephemeral1

Deregistration protection
Disabled

- **AMIs:** These are templates that define the software configuration for an EC2 instance. The list shows the AMI ID, name, owner, and other relevant information.
- **Details:** Clicking on an AMI likely provides more detailed information about its configuration, including platform details, root device type, and kernel ID.
- **Actions:** The "Launch instance from AMI" button suggests that you can create new EC2 instances based on this AMI.

Overall, this image provides a basic overview of the AMIs available in the VPC and allows you to launch new EC2 instances with specific configurations.

J. Auto Scaling Groups



Auto Scaling groups (1/1) Info

☑	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
☑	scaling	AuScaling | Version Default	1	-	1	1	4	us-east-1a, us-east-1b

Auto Scaling group: scaling

Group details

Auto Scaling group name
scaling

Date created
Thu Oct 24 2024 11:47:58 GMT+0300 (Eastern European Summer Time)

Desired capacity
1

Minimum capacity
1

Maximum capacity
4

Desired capacity type
Units (number of instances)

Status
-

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:676154506318:autoScalingGroup:4a7136fb-044e-442e-8ba8-991076a23d26:autoScalingGroupName/scaling

Launch template

Launch template
lt-0cbabf50ef5e2d327
AuScaling

AMI ID
ami-0442f11d447ad8571

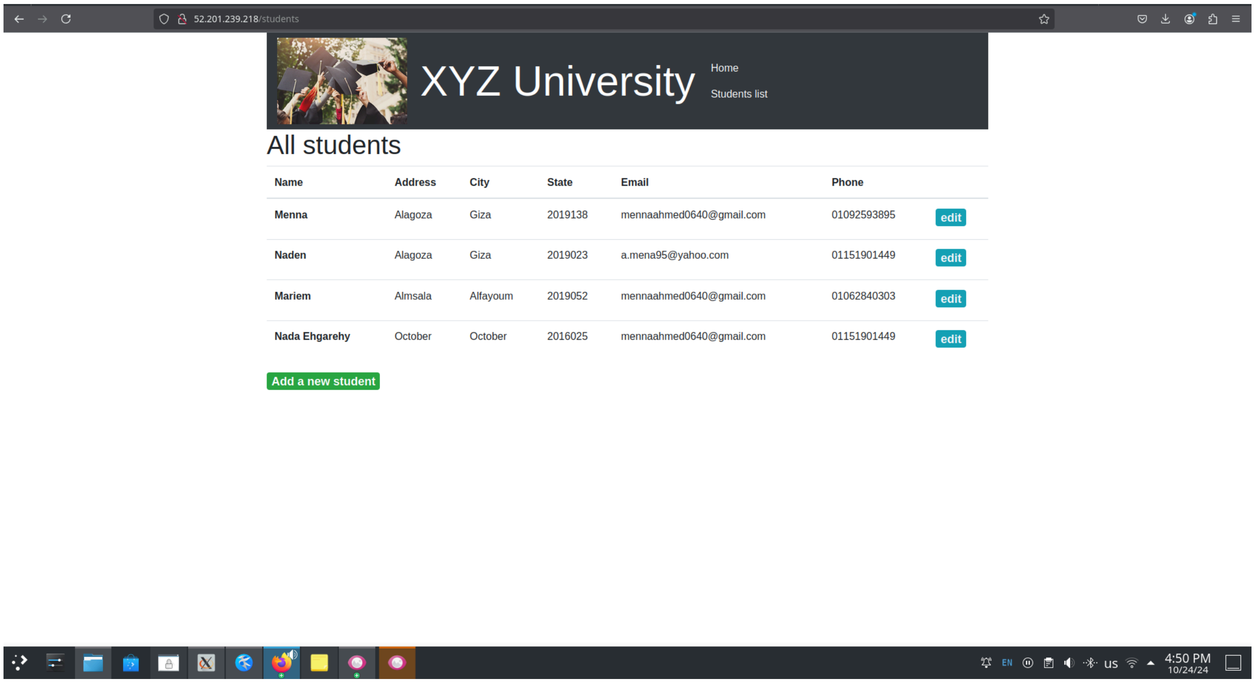
Instance type
t2.micro

Owner
arn:aws:sts::676154506318:assumed-role/voclabs/user3371214=mennaahmed0640@gmail.com

- **Auto Scaling Groups:** These are collections of EC2 instances that automatically scale up or down based on demand. The list shows the Auto Scaling group name, launch template, instances, and other relevant information.
- **Details:** Clicking on an Auto Scaling group likely provides more detailed information about its configuration, including desired capacity, minimum and maximum capacity, and health check settings.
- **Actions:** The "Create Auto Scaling group" button suggests that you can create new Auto Scaling groups within this VPC.

Overall, this image provides a basic overview of the Auto Scaling groups present in the VPC and allows you to manage the scaling of your EC2 instances based on demand.

K. Final Testing Deployment:



XYZ University

All students

Name	Address	City	State	Email	Phone	
Menna	Alagoza	Giza	2019138	mennaahmed0640@gmail.com	01092593895	edit
Naden	Alagoza	Giza	2019023	a.mena95@yahoo.com	01151901449	edit
Mariem	Almsala	Alfayoum	2019052	mennaahmed0640@gmail.com	01062840303	edit
Nada Ehgarehy	October	October	2016025	mennaahmed0640@gmail.com	01151901449	edit

Add a new student

In conclusion, this project successfully demonstrates the design and implementation of a highly available, scalable web application using AWS cloud infrastructure. By leveraging multiple AWS services such as VPC, EC2, RDS, Auto Scaling, and Cloud9, the application is designed to handle traffic fluctuations while ensuring performance and resilience. The architecture follows best practices for security, cost-effectiveness, and scalability, making it an ideal solution for production-level deployment. Additionally, AWS Secrets Manager ensures secure handling of sensitive data, enhancing the overall security of the system. This project highlights the power of cloud technology in building modern, efficient web applications.