رواد مصر الرقمية

وزارة الاتصـــــالات
وتكنولوجيا المعلومات

Project 1: Building a Highly Available, Scalable Web Application

Name: Yousef Hassan Yousef Eltobgy

Student ID: 21031713

Email: eltobgyy@gmail.com

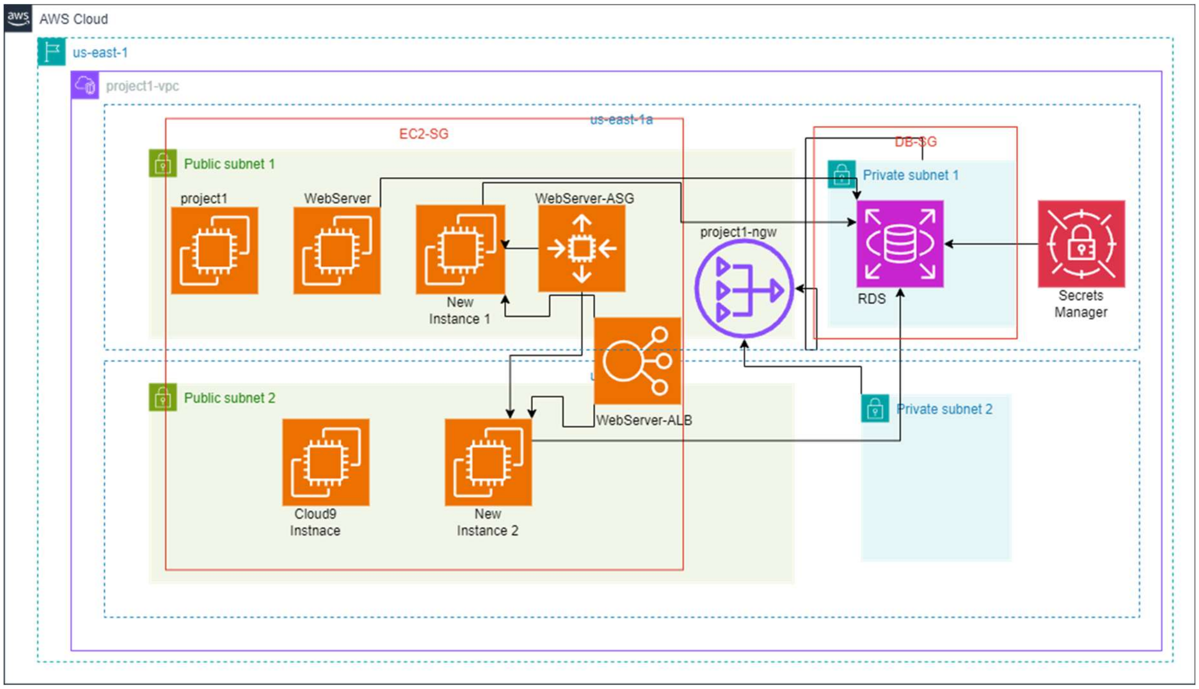
Track: AWS Cloud Solution Admin & Architect

Group Code: ALX1_ISS4_M1e

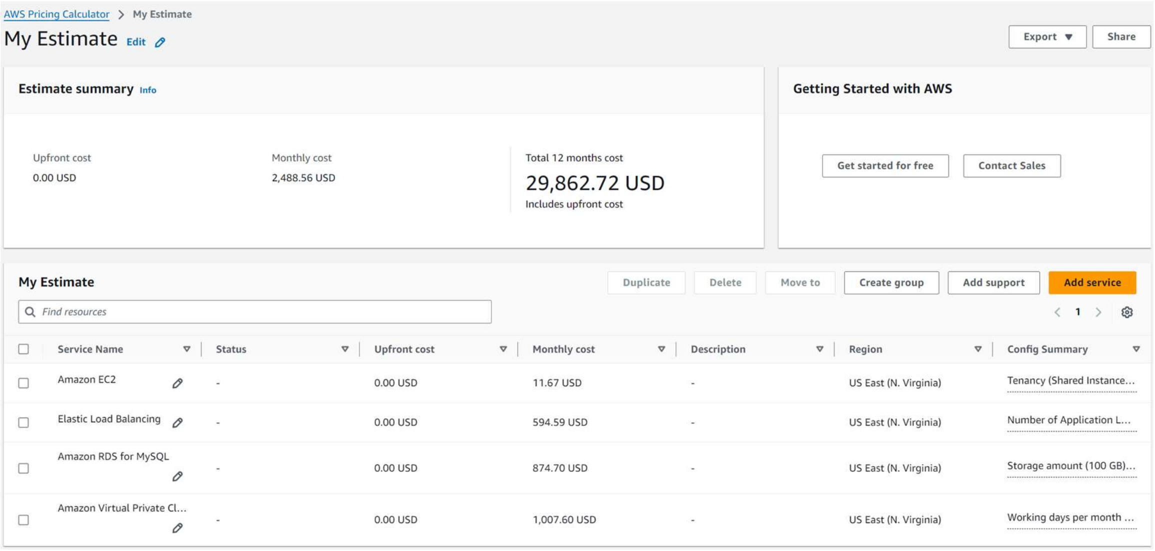
Project 1: Building a Highly Available, Scalable Web Application

Phase 1: Planning the design and estimating cost

Task 1: Creating an architectural diagram



Task 2: Developing a cost estimate



My Estimate

Estimate summary

Upfront cost
0.00 USD

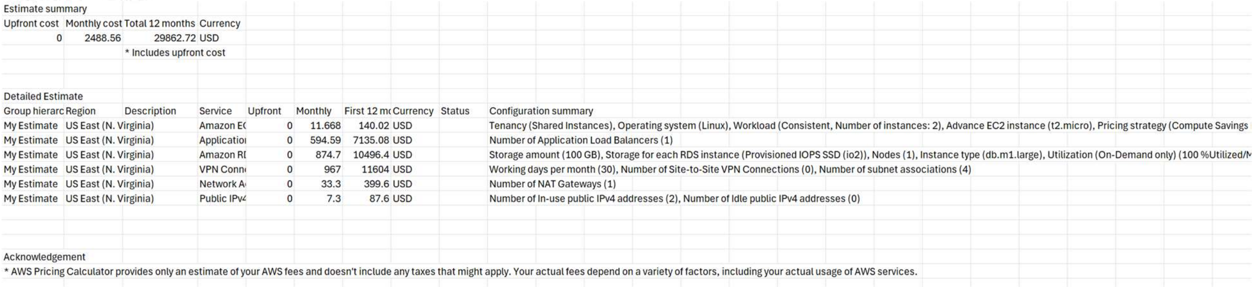
Monthly cost
2,488.56 USD

Total 12 months cost
29,862.72 USD
Includes upfront cost

Getting Started with AWS

My Estimate

□	Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
□	Amazon EC2	-	0.00 USD	11.67 USD	-	US East (N. Virginia)	Tenancy (Shared Instance…
□	Elastic Load Balancing	-	0.00 USD	594.59 USD	-	US East (N. Virginia)	Number of Application L…
□	Amazon RDS for MySQL	-	0.00 USD	874.70 USD	-	US East (N. Virginia)	Storage amount (100 GB)…
□	Amazon Virtual Private Cl…	-	0.00 USD	1,007.60 USD	-	US East (N. Virginia)	Working days per month …



Estimate summary

Upfront cost	Monthly cost	Total 12 months	Currency
0	2488.56	29862.72	USD
		* Includes upfront cost	

Detailed Estimate

Group hierarc	Region	Description	Service	Upfront	Monthly	First 12 mc	Currency	Status	Configuration summary
My Estimate	US East (N. Virginia)		Amazon E(	0	11.668	140.02	USD		Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 2), Advance EC2 instance (t2.micro), Pricing strategy (Compute Savings
My Estimate	US East (N. Virginia)		Applicatio	0	594.59	7135.08	USD		Number of Application Load Balancers (1)
My Estimate	US East (N. Virginia)		Amazon RI	0	874.7	10496.4	USD		Storage amount (100 GB), Storage for each RDS instance (Provisioned IOPS SSD (io2)), Nodes (1), Instance type (db.m1.large), Utilization (On-Demand only) (100 %Utilized/M
My Estimate	US East (N. Virginia)		VPN Conn	0	967	11604	USD		Working days per month (30), Number of Site-to-Site VPN Connections (0), Number of subnet associations (4)
My Estimate	US East (N. Virginia)		Network A	0	33.3	399.6	USD		Number of NAT Gateways (1)
My Estimate	US East (N. Virginia)		Public IPv4	0	7.3	87.6	USD		Number of In-use public IPv4 addresses (2), Number of Idle public IPv4 addresses (0)

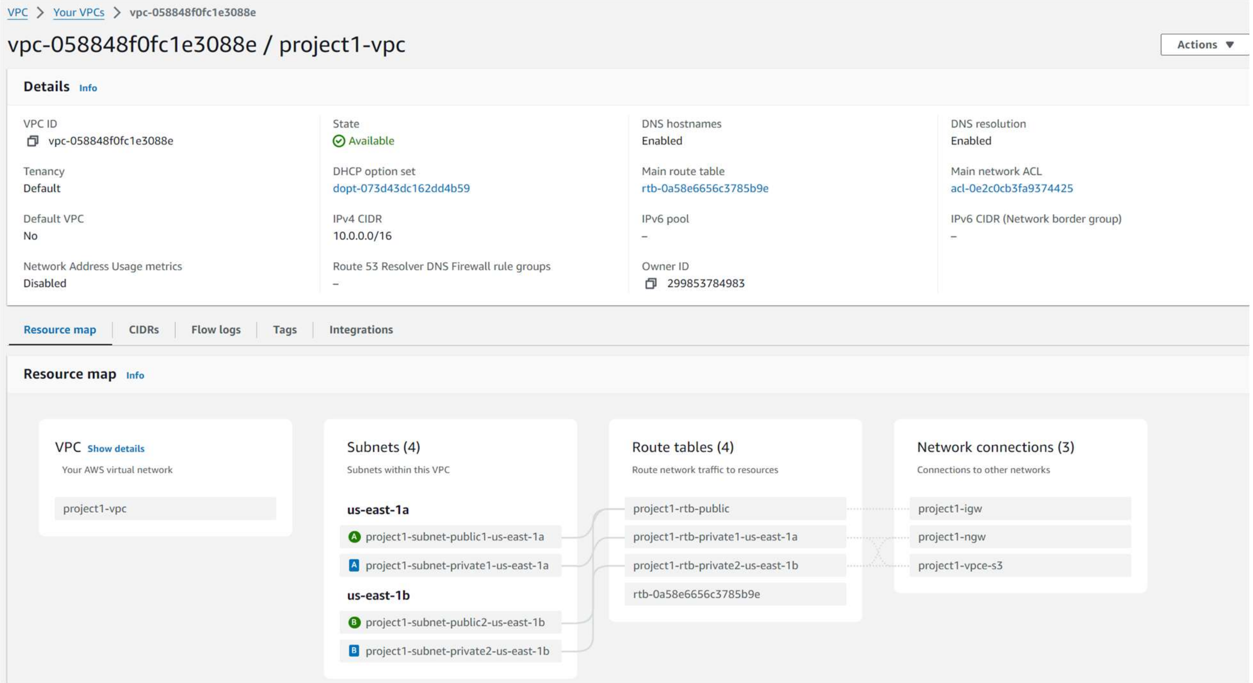
Acknowledgement

* AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services.

Phase 2: Creating a basic functional web application

Task 1: Creating a virtual network

We created VPC.



VPC > Your VPCs > vpc-058848f0fc1e3088e

vpc-058848f0fc1e3088e / project1-vpc

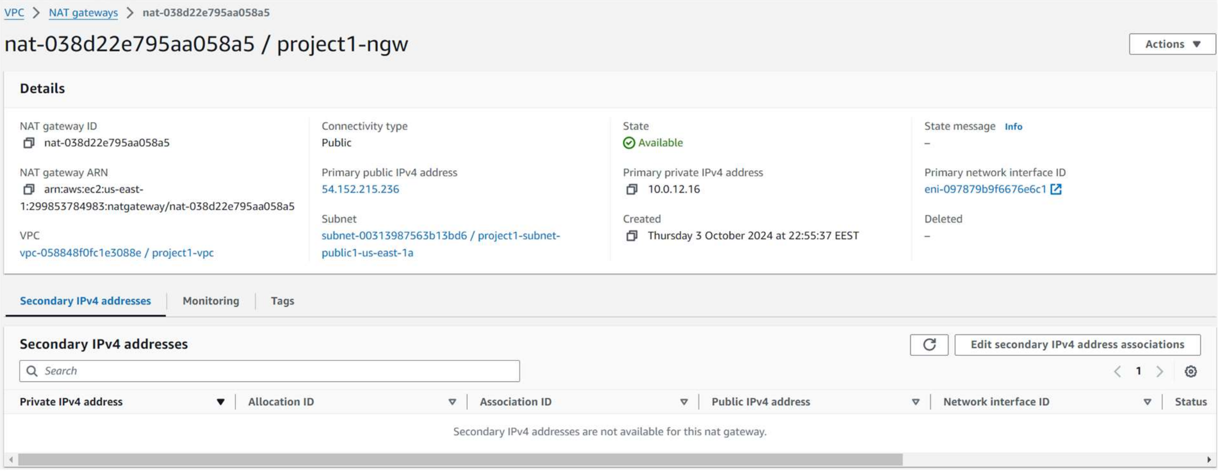
Details Info

VPC ID vpc-058848f0fc1e3088e	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-073d43dc162dd4b59	Main route table rtb-0a58e6656c3785b9e	Main network ACL acl-0e2c0cb3fa9374425
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 –	Owner ID 299853784983	

Resource map Info

VPC Show details
Your AWS virtual network
project1-vpc
Subnets (4)
Subnets within this VPC
us-east-1a
A project1-subnet-public1-us-east-1a
A project1-subnet-private1-us-east-1a
us-east-1b
B project1-subnet-public2-us-east-1b
B project1-subnet-private2-us-east-1b
Route tables (4)
Route network traffic to resources
project1-rtb-public
project1-rtb-private1-us-east-1a
project1-rtb-private2-us-east-1b
rtb-0a58e6656c3785b9e
Network connections (3)
Connections to other networks
project1-igw
project1-ngw
project1-vpce-s3

We then create NAT Gateway in the public subnet 1.



VPC > NAT gateways > nat-038d22e795aa058a5
Actions ▼
Secondary IPv4 addresses
Monitoring
Tags
Edit secondary IPv4 address associations
Search
1

nat-038d22e795aa058a5 / project1-ngw

Details

NAT gateway ID: nat-038d22e795aa058a5

NAT gateway ARN: arn:aws:ec2:us-east-1:299853784983:natgateway/nat-038d22e795aa058a5

VPC: vpc-058848f0fc1e3088e / project1-vpc

Connectivity type: Public

Primary public IPv4 address: 54.152.215.236

Subnet: subnet-00313987563b13bd6 / project1-subnet-public1-us-east-1a

State: Available

Primary private IPv4 address: 10.0.12.16

Created: Thursday 3 October 2024 at 22:55:37 EEST

State message Info: –

Primary network interface ID: eni-097879b9f6676e6c1

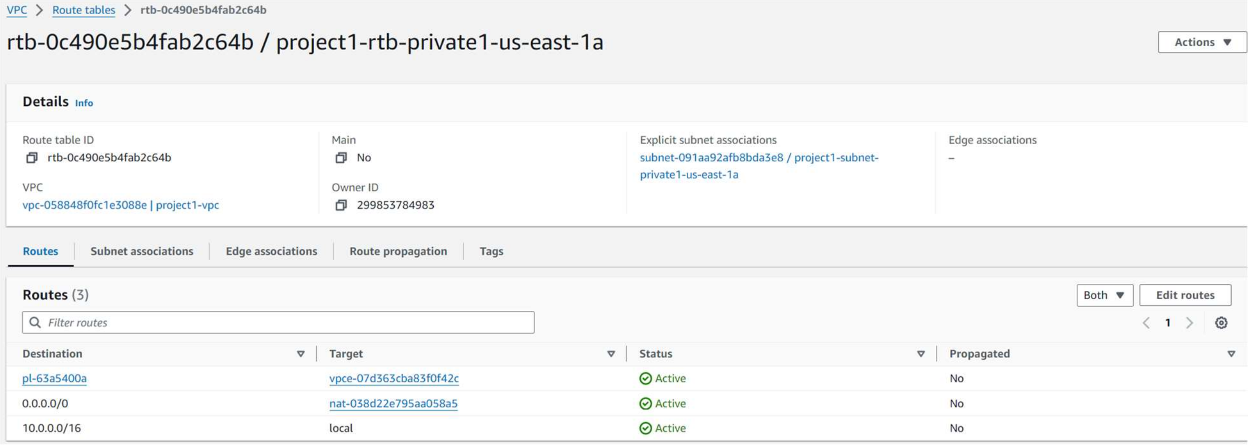
Deleted: –

Secondary IPv4 addresses

Private IPv4 address	Allocation ID	Association ID	Public IPv4 address	Network interface ID	Status

Secondary IPv4 addresses are not available for this nat gateway.

We then go to route tables and add new route for anywhere (0.0.0.0/0) to exit from NAT gateway in the private subnet 1.



rtb-0c490e5b4fab2c64b / project1-rtb-private1-us-east-1a

Details Info

Route table ID: rtb-0c490e5b4fab2c64b

VPC: vpc-058848f0fc1e3088e | project1-vpc

Main: No

Owner ID: 299853784983

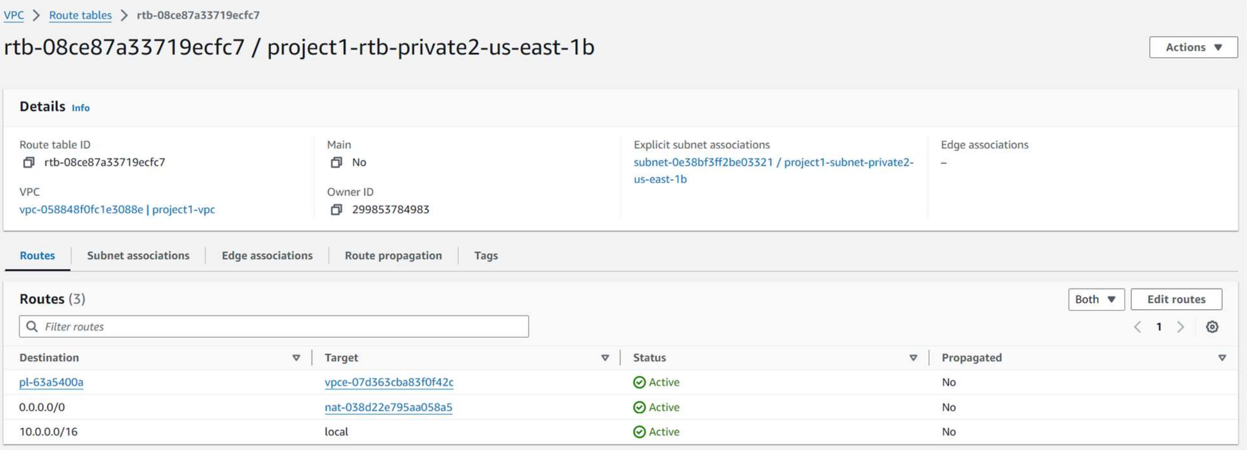
Explicit subnet associations: subnet-091aa92afb8bda3e8 / project1-subnet-private1-us-east-1a

Edge associations: –

Routes (3)

Destination	Target	Status	Propagated
pl-63a5400a	vpce-07d363cba83f0f42c	Active	No
0.0.0.0/0	nat-038d22e795aa058a5	Active	No
10.0.0.0/16	local	Active	No

Then do the same for private subnet 2.



rtb-08ce87a33719ecfc7 / project1-rtb-private2-us-east-1b

Details Info

Route table ID: rtb-08ce87a33719ecfc7

VPC: vpc-058848f0fc1e3088e | project1-vpc

Main: No

Owner ID: 299853784983

Explicit subnet associations: subnet-0e38bf3ff2be03321 / project1-subnet-private2-us-east-1b

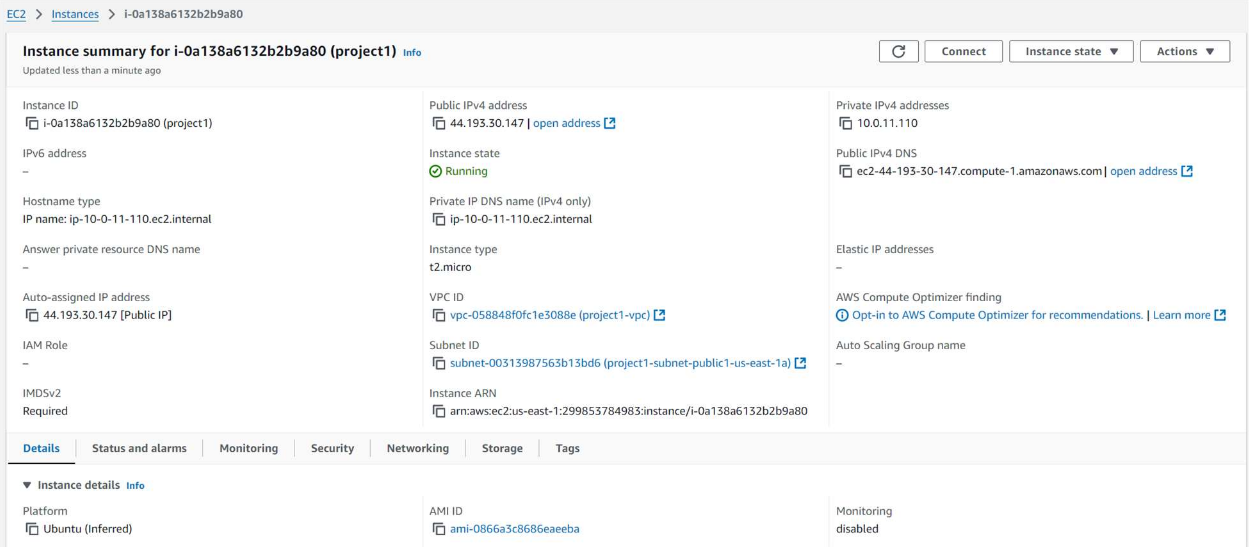
Edge associations: –

Routes (3)

Destination	Target	Status	Propagated
pl-63a5400a	vpce-07d363cba83f0f42c	Active	No
0.0.0.0/0	nat-038d22e795aa058a5	Active	No
10.0.0.0/16	local	Active	No

Task 2: Creating a virtual machine

We will create an EC2 instance. We will select Ubuntu as Operating system and we will create a security group to allow SSH and HTTP connections from anywhere (0.0.0.0/0).



Instance summary for i-0a138a6132b2b9a80 (project1) Info
Updated less than a minute ago

Instance ID
i-0a138a6132b2b9a80 (project1)

Public IPv4 address
44.193.30.147 | open address

Private IPv4 addresses
10.0.11.110

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-44-193-30-147.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-10-0-11-110.ec2.internal

Private IP DNS name (IPv4 only)
ip-10-0-11-110.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
44.193.30.147 [Public IP]

VPC ID
vpc-058848f0fc1e3088e (project1-vpc)

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

IAM Role
–

Subnet ID
subnet-00313987563b13bd6 (project1-subnet-public1-us-east-1a)

Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:us-east-1:299853784983:instance/i-0a138a6132b2b9a80

▼ Instance details Info

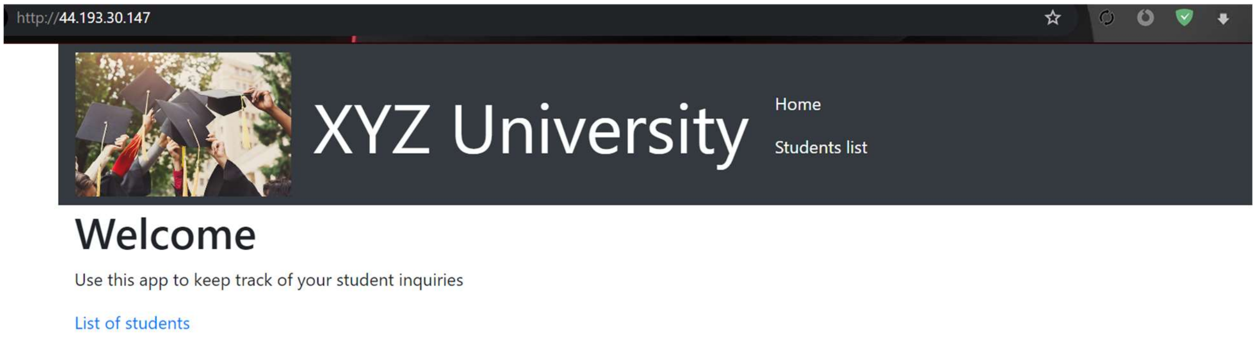
Platform
Ubuntu (Inferred)

AMI ID
ami-0866a3c8686eaeeba

Monitoring
disabled

Task 3: Testing the deployment

And we can test that the web application is running successfully by going to the public IP in new browser window.



XYZ University

Welcome

Use this app to keep track of your student inquiries

List of students

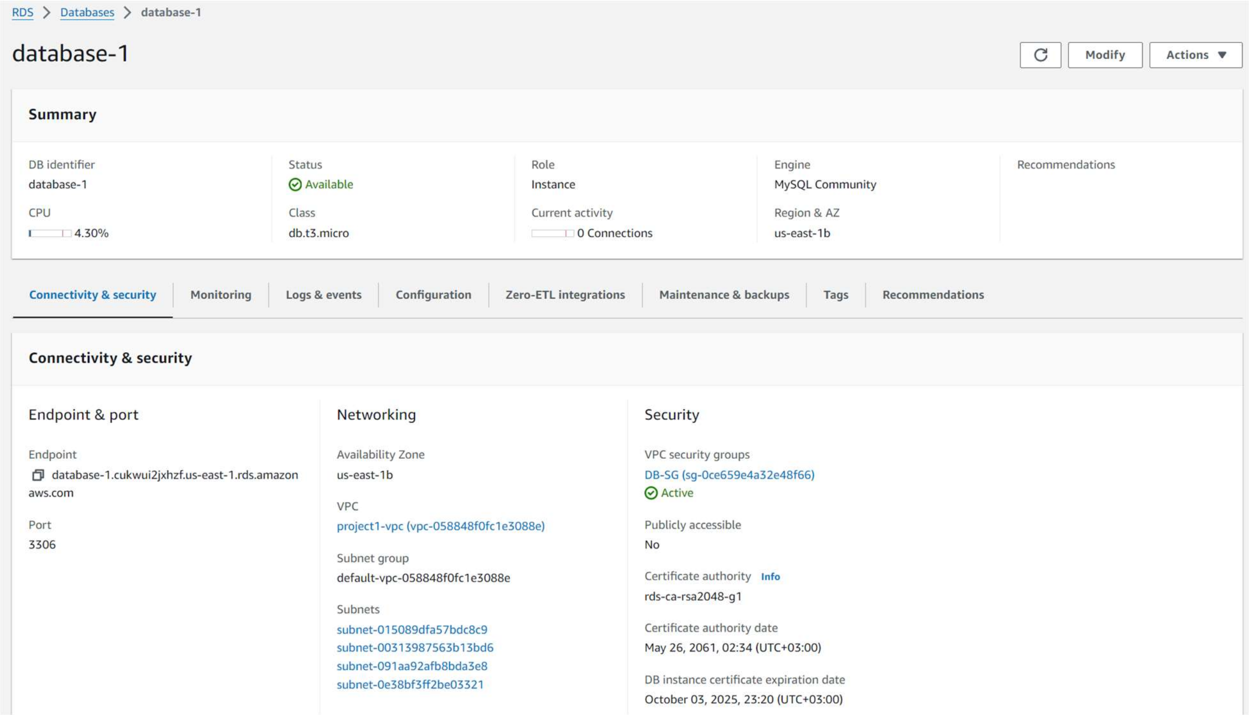
Phase 3: Decoupling the application components

Task 1: Changing the VPC configuration

We already created the private subnets and in two Availability Zones.

Task 2: Creating and configuring the Amazon RDS database

We create new database from RDS dashboard. We will choose MySQL engine.



database-1

Summary

DB identifier
database-1

CPU
4.30%

Status
✓ Available

Class
db.t3.micro

Role
Instance

Current activity
0 Connections

Engine
MySQL Community

Region & AZ
us-east-1b

Recommendations

Connectivity & security

Endpoint & port

Endpoint
database-1.cukwui2jxhzf.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1b

VPC
project1-vpc (vpc-058848f0fc1e3088e)

Subnet group
default-vpc-058848f0fc1e3088e

Subnets
subnet-015089dfa57bdc8c9
subnet-00313987563b13bd6
subnet-091aa92afb8bda3e8
subnet-0e38bf3ff2be03321

Security

VPC security groups
DB-SG (sg-0ce659e4a32e48f66)
✓ Active

Publicly accessible
No

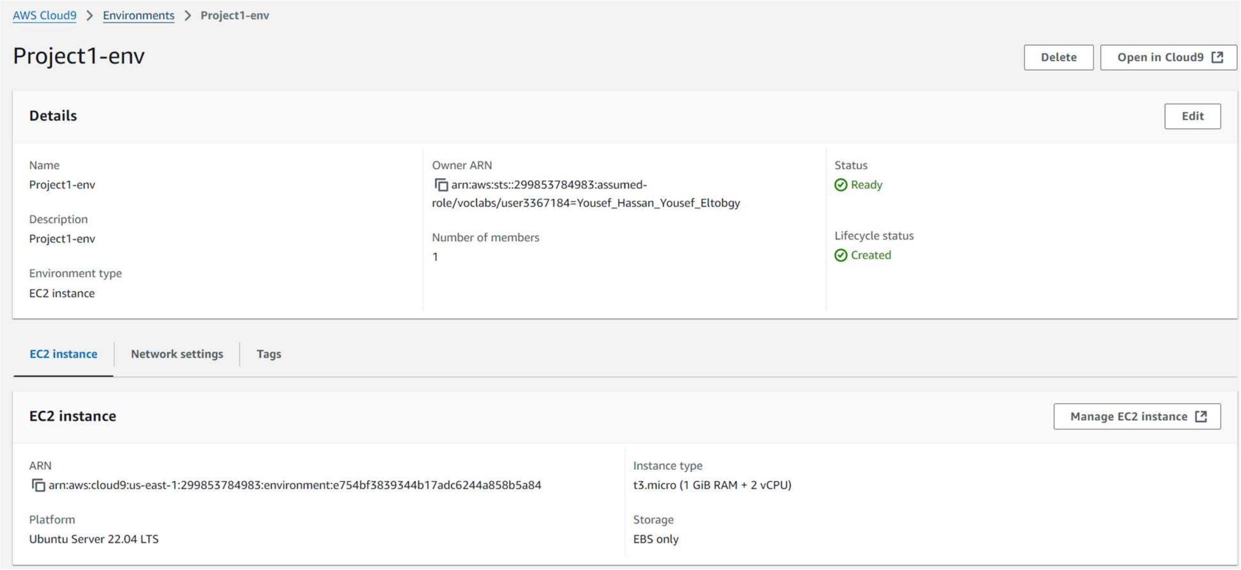
Certificate authority Info
rds-ca-rsa2048-g1

Certificate authority date
May 26, 2061, 02:34 (UTC+03:00)

DB instance certificate expiration date
October 03, 2025, 23:20 (UTC+03:00)

Task 3: Configuring the development environment

Created cloud9 environment on a new t3.micro instance and the platform is Ubuntu and the access is SSH.



Project1-env

Details

Name
Project1-env

Description
Project1-env

Environment type
EC2 instance

Owner ARN
arn:aws:sts::299853784983:assumed-role/voclabs/user3367184=Yousef_Hassan_Yousef_Eltobgy

Number of members
1

Status
✓ Ready

Lifecycle status
✓ Created

EC2 instance

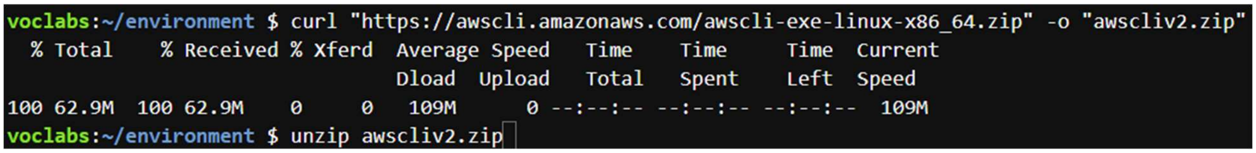
ARN
arn:aws:cloud9:us-east-1:299853784983:environment:e754bf3839344b17adc6244a858b5a84

Platform
Ubuntu Server 22.04 LTS

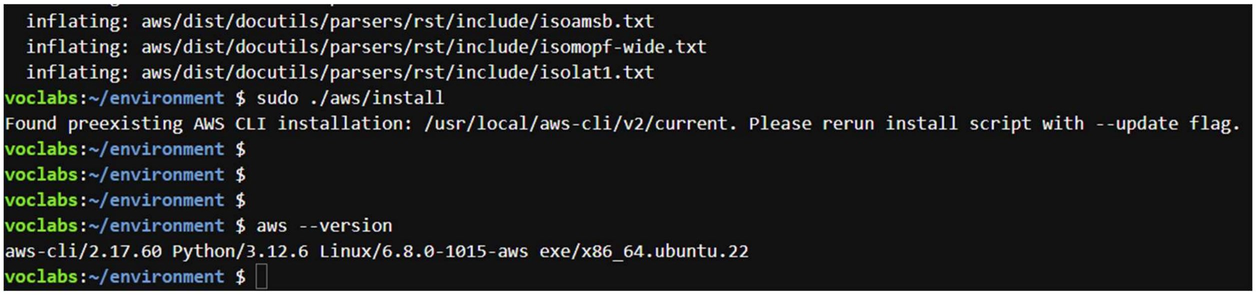
Instance type
t3.micro (1 GiB RAM + 2 vCPU)

Storage
EBS only

Installed AWS CLI on it.



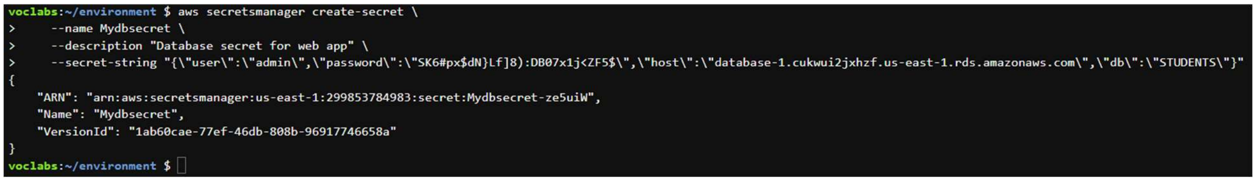
```
voclabs:~/environment $ curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 62.9M  100 62.9M    0     0   109M      0 --:--:-- --:--:-- --:--:--  109M
voclabs:~/environment $ unzip awscliv2.zip
```



```
  inflating: aws/dist/docutils/parsers/rst/include/isoamsb.txt
  inflating: aws/dist/docutils/parsers/rst/include/isomopf-wide.txt
  inflating: aws/dist/docutils/parsers/rst/include/isolat1.txt
voclabs:~/environment $ sudo ./aws/install
Found preexisting AWS CLI installation: /usr/local/aws-cli/v2/current. Please rerun install script with --update flag.
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $
voclabs:~/environment $ aws --version
aws-cli/2.17.60 Python/3.12.6 Linux/6.8.0-1015-aws exe/x86_64.ubuntu.22
voclabs:~/environment $
```

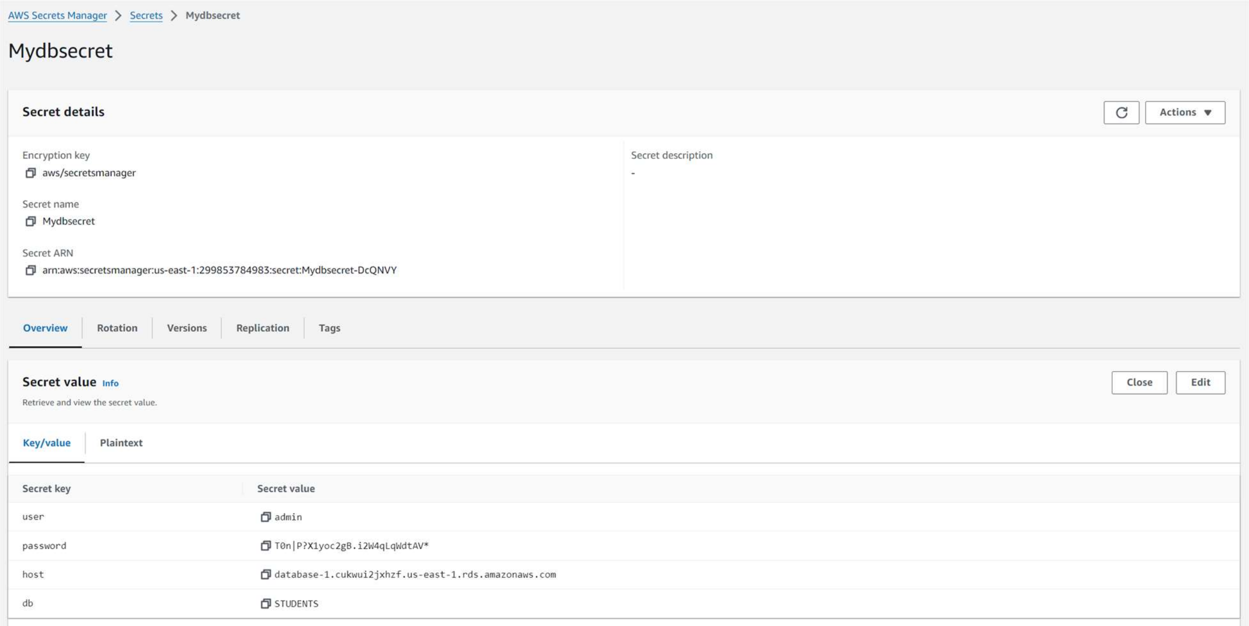
Task 4: Provisioning Secrets Manager

Created new secret using the below command in the cloud9 session.



```
voclabs:~/environment $ aws secretsmanager create-secret \
>     --name Mydbsecret \
>     --description "Database secret for web app" \
>     --secret-string "{\"user\":\"admin\",\"password\":\"SK6#px$dN}Lf]8):DB07x1j<ZF5$\",\"host\":\"database-1.cukwui2jxhzf.us-east-1.rds.amazonaws.com\",\"db\":\"STUDENTS\"}"
{
    "ARN": "arn:aws:secretsmanager:us-east-1:299853784983:secret:Mydbsecret-ze5uiW",
    "Name": "Mydbsecret",
    "VersionId": "1ab60cae-77ef-46db-808b-96917746658a"
}
voclabs:~/environment $
```

And we can confirm the data from the secret manager dashboard.



Mydbsecret

Secret details

Encryption key
aws/secretsmanager

Secret name
Mydbsecret

Secret ARN
arn:aws:secretsmanager:us-east-1:299853784983:secret:Mydbsecret-DcQNVY

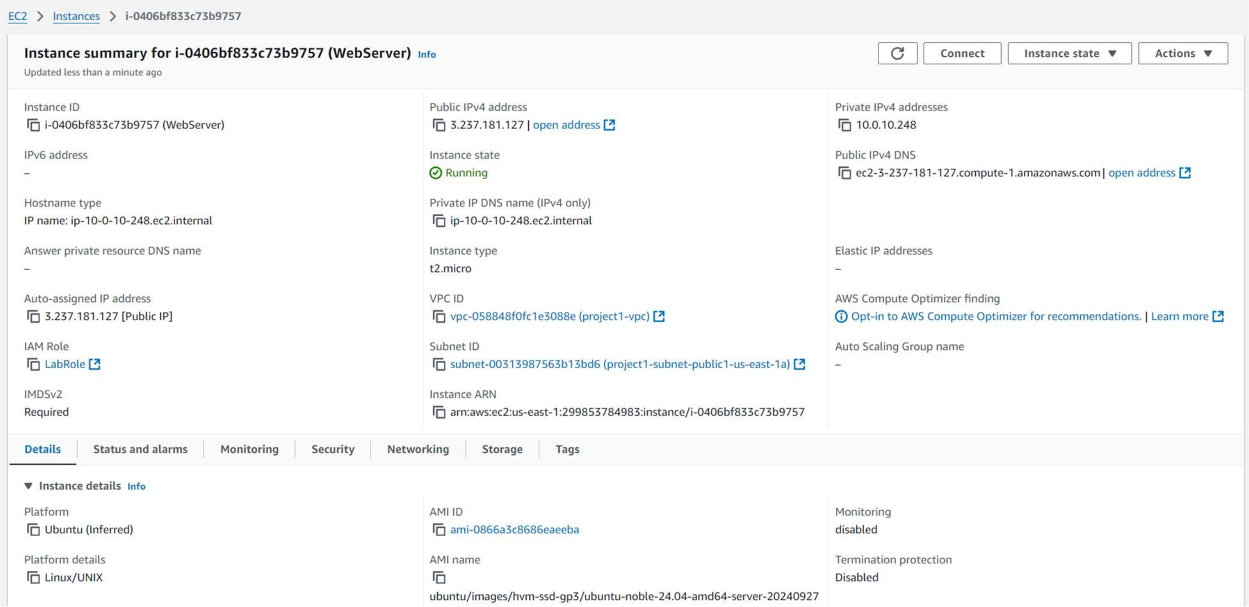
Secret description
-

Secret value Info
Retrieve and view the secret value.

Secret key	Secret value
user	admin
password	T0n|P?X1yoc2gB.i2W4qLqWdtAV*
host	database-1.cukwui2jxhzf.us-east-1.rds.amazonaws.com
db	STUDENTS

Task 5: Provisioning a new instance for the web server

Created new instance and added the script of the web application



EC2 > Instances > i-0406bf833c73b9757

Instance summary for i-0406bf833c73b9757 (WebServer) Info

Updated less than a minute ago

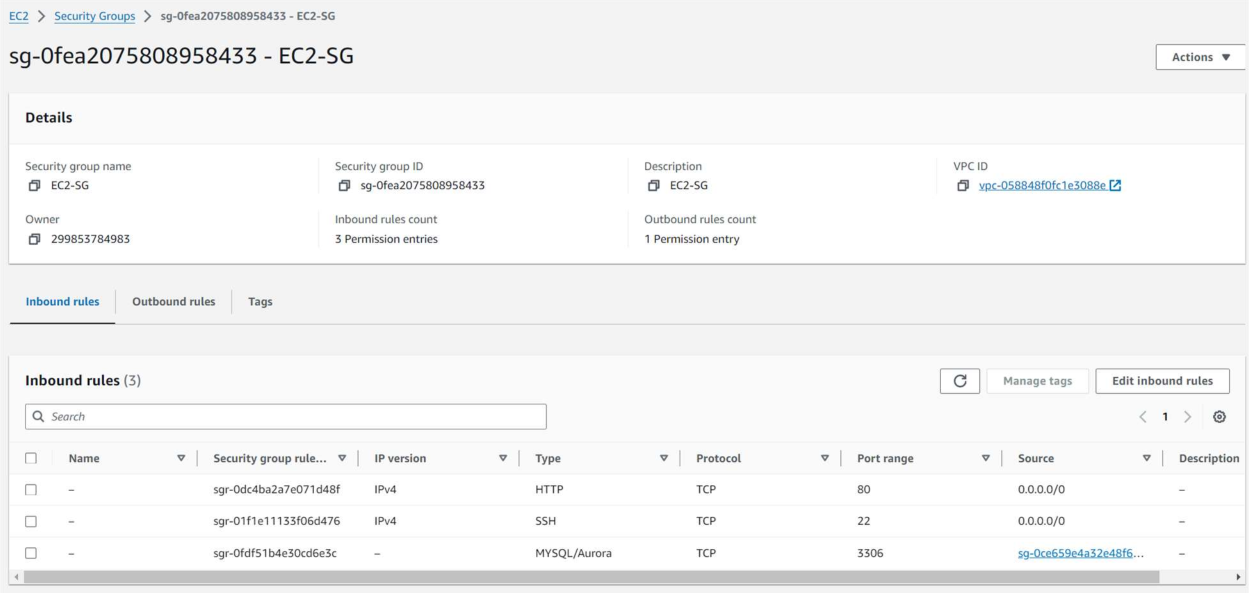
Instance ID i-0406bf833c73b9757 (WebServer)	Public IPv4 address 3.237.181.127 | open address	Private IPv4 addresses 10.0.10.248
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-3-237-181-127.compute-1.amazonaws.com | open address
Hostname type IP name: ip-10-0-10-248.ec2.internal	Private IP DNS name (IPv4 only) ip-10-0-10-248.ec2.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 3.237.181.127 [Public IP]	VPC ID vpc-058848f0fc1e3088e (project1-vpc)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role LabRole	Subnet ID subnet-00313987563b13bd6 (project1-subnet-public1-us-east-1a)	Auto Scaling Group name –
IMDSv2 Required	Instance ARN arn:aws:ec2:us-east-1:299853784983:instance/i-0406bf833c73b9757	

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Instance details Info

Platform Ubuntu (Inferred)	AMI ID ami-0866a3c8686eaeeba	Monitoring disabled
Platform details Linux/UNIX	AMI name ubuntu/images/hvm-ssd-gp3/ubuntu-noble-24.04-amd64-server-20240927	Termination protection Disabled

Edited the security group for EC2 instances to allow traffic through port 3306



EC2 > Security Groups > sg-0fea2075808958433 - EC2-SG

sg-0fea2075808958433 - EC2-SG

Details

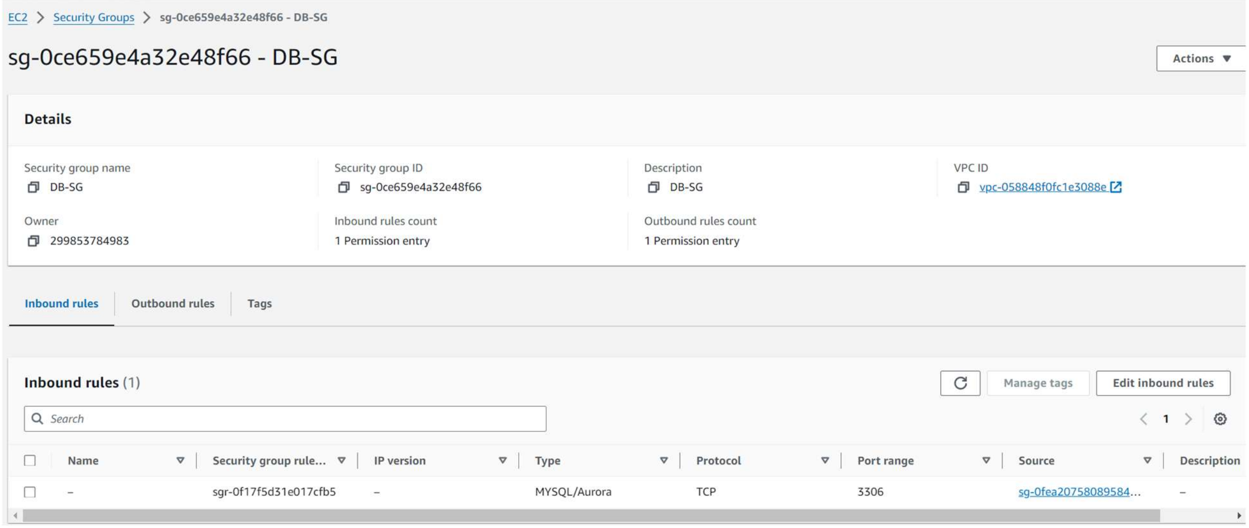
Security group name EC2-SG	Security group ID sg-0fea2075808958433	Description EC2-SG	VPC ID vpc-058848f0fc1e3088e
Owner 299853784983	Inbound rules count 3 Permission entries	Outbound rules count 1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (3)

□	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-0dc4ba2a7e071d48f	IPv4	HTTP	TCP	80	0.0.0.0/0	–
□	–	sgr-01f1e11133f06d476	IPv4	SSH	TCP	22	0.0.0.0/0	–
□	–	sgr-0fdf51b4e30cd6e3c	–	MYSQL/Aurora	TCP	3306	sg-0ce659e4a32e48f6...	–

Also created a security group for the RDS instance and allowed inbound traffic on port 3306



EC2 > Security Groups > sg-0ce659e4a32e48f66 - DB-SG

sg-0ce659e4a32e48f66 - DB-SG

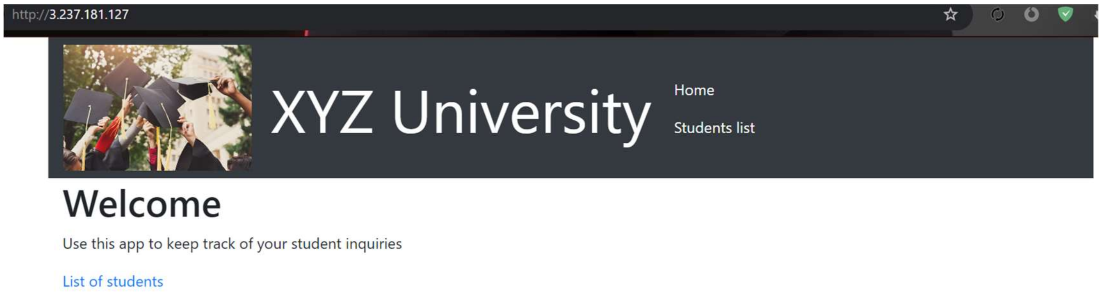
Details

Security group name	Security group ID	Description	VPC ID
DB-SG	sg-0ce659e4a32e48f66	DB-SG	vpc-058848f0fc1e3088e
Owner	**Inbound rules count**	**Outbound rules count**	
299853784983	1 Permission entry	1 Permission entry	

Inbound rules (1)

□	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
□	–	sgr-0f17f5d31e017cfb5	–	MYSQL/Aurora	TCP	3306	sg-0fea20758089584...	–

Tested the web app is working fine



http://3.237.181.127

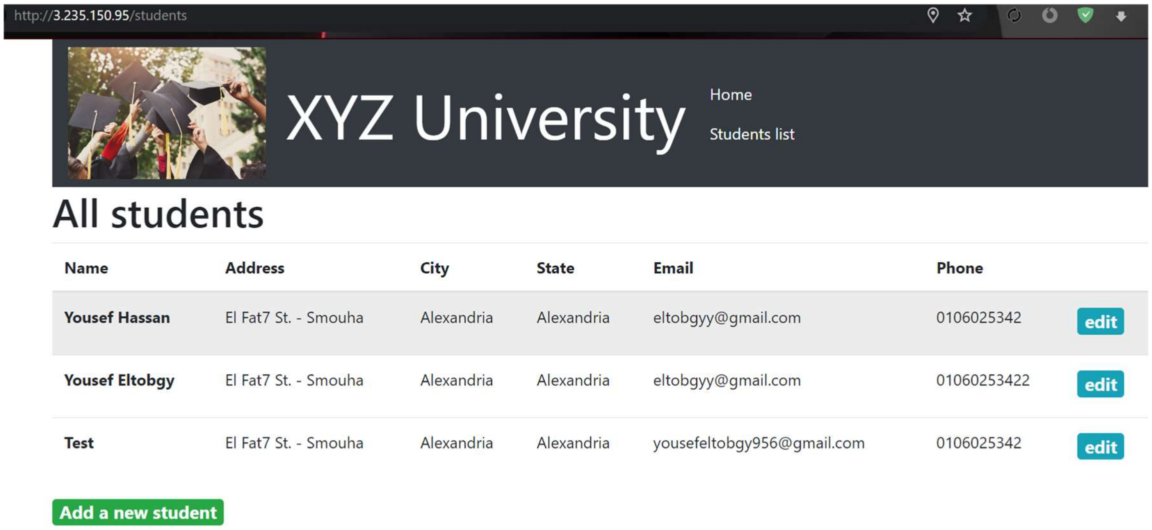
Welcome

Use this app to keep track of your student inquiries

List of students

Task 6: Migrating the database

First add some data on the website (on the old EC2 instance's public IP) like below:

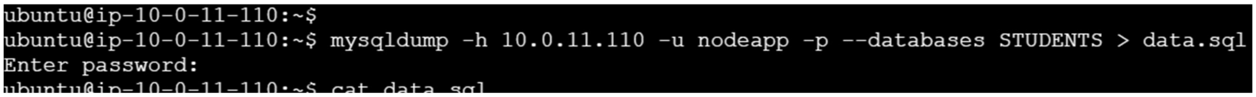


http://3.235.150.95/students

All students

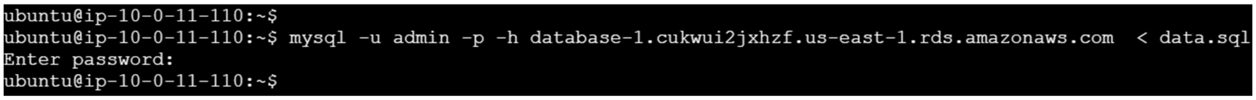
Name	Address	City	State	Email	Phone	
Yousef Hassan	El Fat7 St. - Smouha	Alexandria	Alexandria	eltobgyy@gmail.com	0106025342	edit
Yousef Eltobgy	El Fat7 St. - Smouha	Alexandria	Alexandria	eltobgyy@gmail.com	01060253422	edit
Test	El Fat7 St. - Smouha	Alexandria	Alexandria	yousefeltobgy956@gmail.com	0106025342	edit

Then, we will export the data from the old EC2 instance



```
ubuntu@ip-10-0-11-110:~$
ubuntu@ip-10-0-11-110:~$ mysqldump -h 10.0.11.110 -u nodeapp -p --databases STUDENTS > data.sql
Enter password:
ubuntu@ip-10-0-11-110:~$ cat data.sql
```

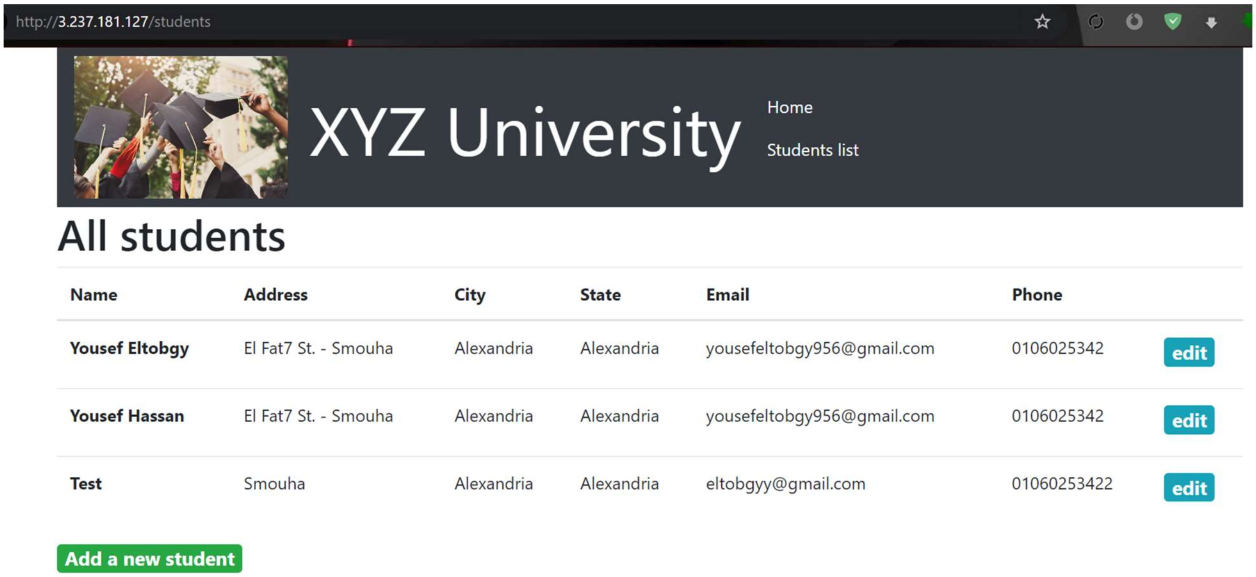
Then we will put it in the RDS instance



```
ubuntu@ip-10-0-11-110:~$
ubuntu@ip-10-0-11-110:~$ mysql -u admin -p -h database-1.cukwui2jxhzf.us-east-1.rds.amazonaws.com  < data.sql
Enter password:
ubuntu@ip-10-0-11-110:~$
```

Task 7: Testing the application

I went to the newly created EC2 instance and the found the data migrated which means the application can access it from RDS instance.



All students

Name	Address	City	State	Email	Phone	
Yousef Eltobgy	El Fat7 St. - Smouha	Alexandria	Alexandria	yousefeltobgy956@gmail.com	0106025342	edit
Yousef Hassan	El Fat7 St. - Smouha	Alexandria	Alexandria	yousefeltobgy956@gmail.com	0106025342	edit
Test	Smouha	Alexandria	Alexandria	eltobgyy@gmail.com	01060253422	edit

Phase 4: Implementing high availability and scalability

Task 1: Creating an Application Load Balancer

Created image from the Webserver instance.

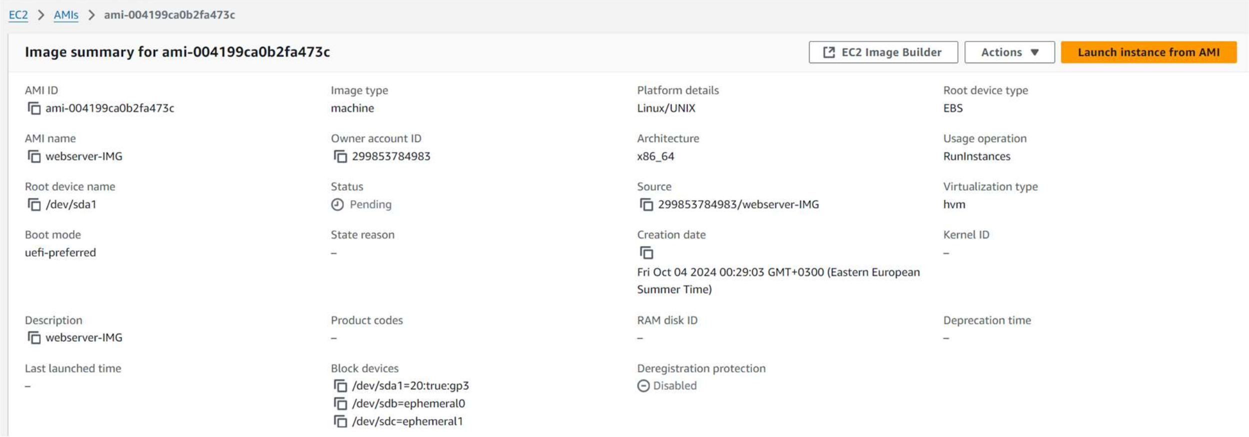
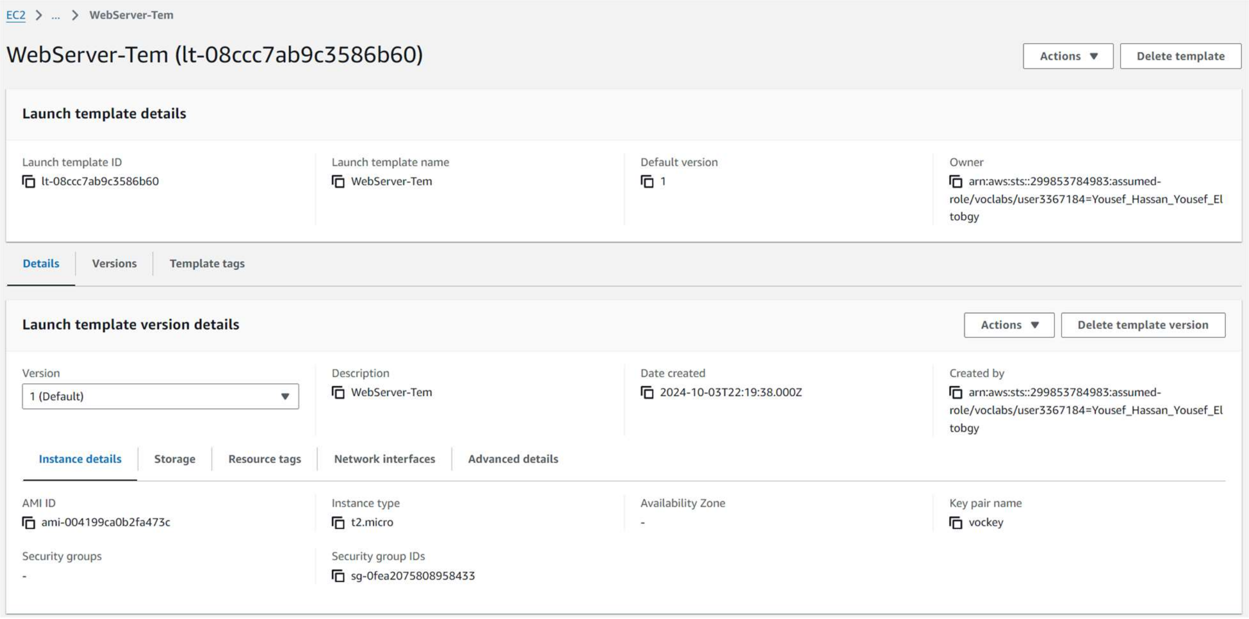


Image summary for ami-004199ca0b2fa473c

AMI ID: ami-004199ca0b2fa473c
Image type: machine
Platform details: Linux/UNIX
Root device type: EBS
AMI name: webserver-IMG
Owner account ID: 299853784983
Architecture: x86_64
Usage operation: RunInstances
Root device name: /dev/sda1
Status: Pending
Source: 299853784983/webserver-IMG
Virtualization type: hvm
Boot mode: uefi-preferred
State reason: –
Creation date: Fri Oct 04 2024 00:29:03 GMT+0300 (Eastern European Summer Time)
Kernel ID: –
Description: webserver-IMG
Product codes: –
RAM disk ID: –
Deprecation time: –
Last launched time: –
Block devices: /dev/sda1=20:true:gp3, /dev/sdb=ephemeral0, /dev/sdc=ephemeral1
Deregistration protection: Disabled

Create a launch template.



WebServer-Tem (lt-08ccc7ab9c3586b60)

Launch template details

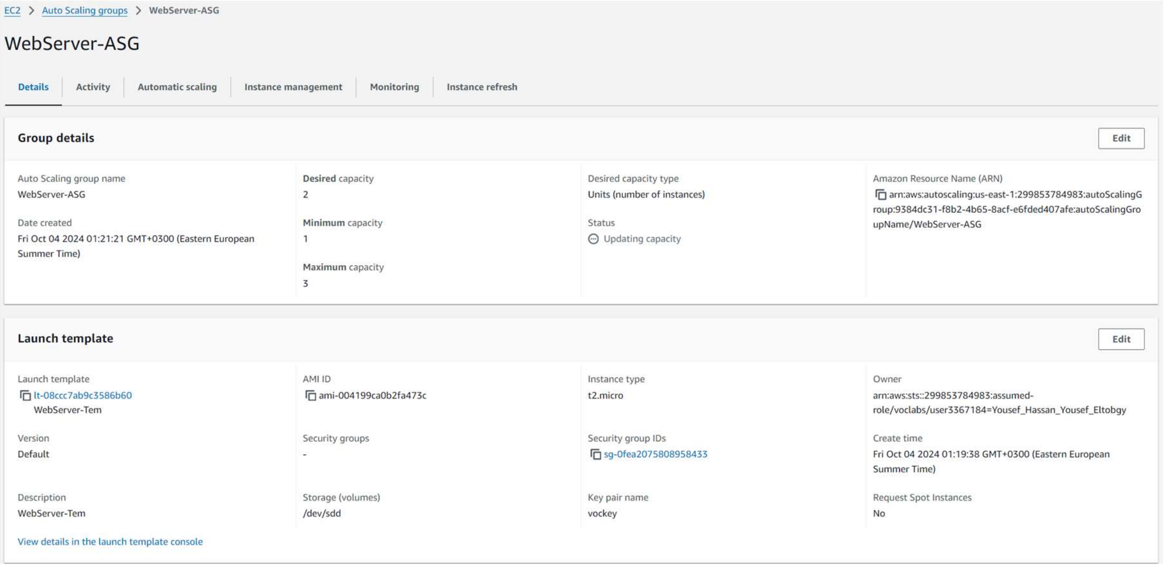
Launch template ID: lt-08ccc7ab9c3586b60
Launch template name: WebServer-Tem
Default version: 1
Owner: arn:aws:sts::299853784983:assumed-role/voclabs/user3367184=Yousef_Hassan_Yousef_Eltobgy

Launch template version details

Version: 1 (Default)
Description: WebServer-Tem
Date created: 2024-10-03T22:19:38.000Z
Created by: arn:aws:sts::299853784983:assumed-role/voclabs/user3367184=Yousef_Hassan_Yousef_Eltobgy

AMI ID: ami-004199ca0b2fa473c
Instance type: t2.micro
Availability Zone: -
Key pair name: vockey
Security groups: -
Security group IDs: sg-0fea2075808958433

Then create the Automatic Scaling Group. I set the desired capacity to two as I want two instances to run in the two public subnets (one per each).



EC2 > Auto Scaling groups > WebServer-ASG

WebServer-ASG

Group details

Auto Scaling group name
WebServer-ASG

Date created
Fri Oct 04 2024 01:21:21 GMT+0300 (Eastern European Summer Time)

Desired capacity
2

Minimum capacity
1

Maximum capacity
3

Desired capacity type
Units (number of instances)

Status
Updating capacity

Amazon Resource Name (ARN)
arn:aws:autoscaling:us-east-1:299853784983:autoScalingGroup:9384dc31-f8b2-4b65-8acf-e6fded407afe:autoScalingGroupName/WebServer-ASG

Launch template

Launch template
lt-08ccc7ab9c3586b60
WebServer-Tem

Version
Default

Description
WebServer-Tem

AMI ID
ami-004199ca0b2fa473c

Security groups
-

Storage (volumes)
/dev/sdd

Instance type
t2.micro

Security group IDs
sg-0fea2075808958433

Key pair name
vockey

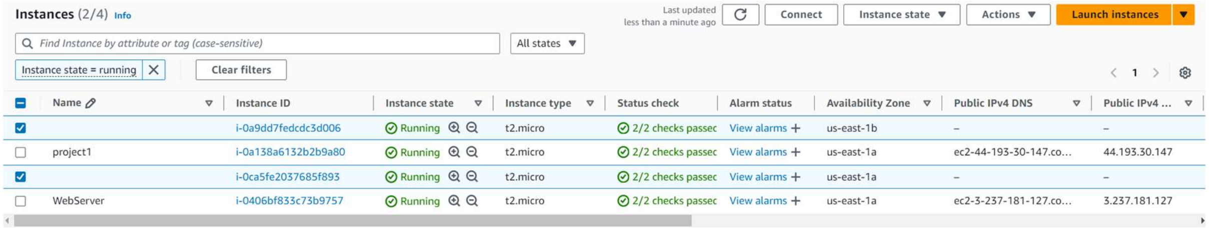
Owner
arn:aws:sts::299853784983:assumed-role/voclabs/user3367184=Yousef_Hassan_Yousef_Eltobgy

Create time
Fri Oct 04 2024 01:19:38 GMT+0300 (Eastern European Summer Time)

Request Spot Instances
No

View details in the launch template console

You can see that there two new instances have been created in the EC2 dashboard (because we set the desired capacity to 2 instances).

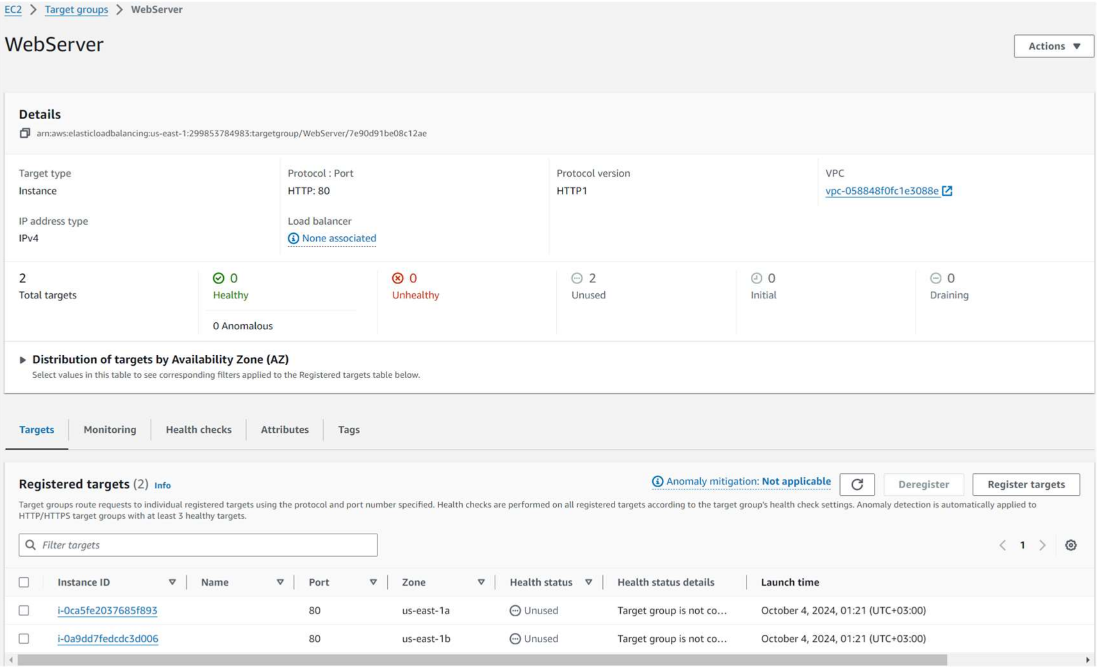


Instances (2/4) Info

Instance state = running

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
☑		i-0a9dd7fedcdc3d006	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1b	–	–
☐	project1	i-0a138a6132b2b9a80	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-44-193-30-147.co...	44.193.30.147
☑		i-0ca5fe2037685f893	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	–	–
☐	WebServer	i-0406bf833c73b9757	Running	t2.micro	2/2 checks passec	View alarms +	us-east-1a	ec2-3-237-181-127.co...	3.237.181.127

Create a target group and include the two created instances.



EC2 > Target groups > WebServer

WebServer

Details
arn:aws:elasticloadbalancing:us-east-1:299853784983:targetgroup/WebServer/7e90d91be08c12ae

Target type
Instance

IP address type
IPv4

Protocol : Port
HTTP: 80

Load balancer
None associated

Protocol version
HTTP1

VPC
vpc-058848f0fc1e3088e

2 Total targets
0 Healthy
0 Anomalous
0 Unhealthy
2 Unused
0 Initial
0 Draining

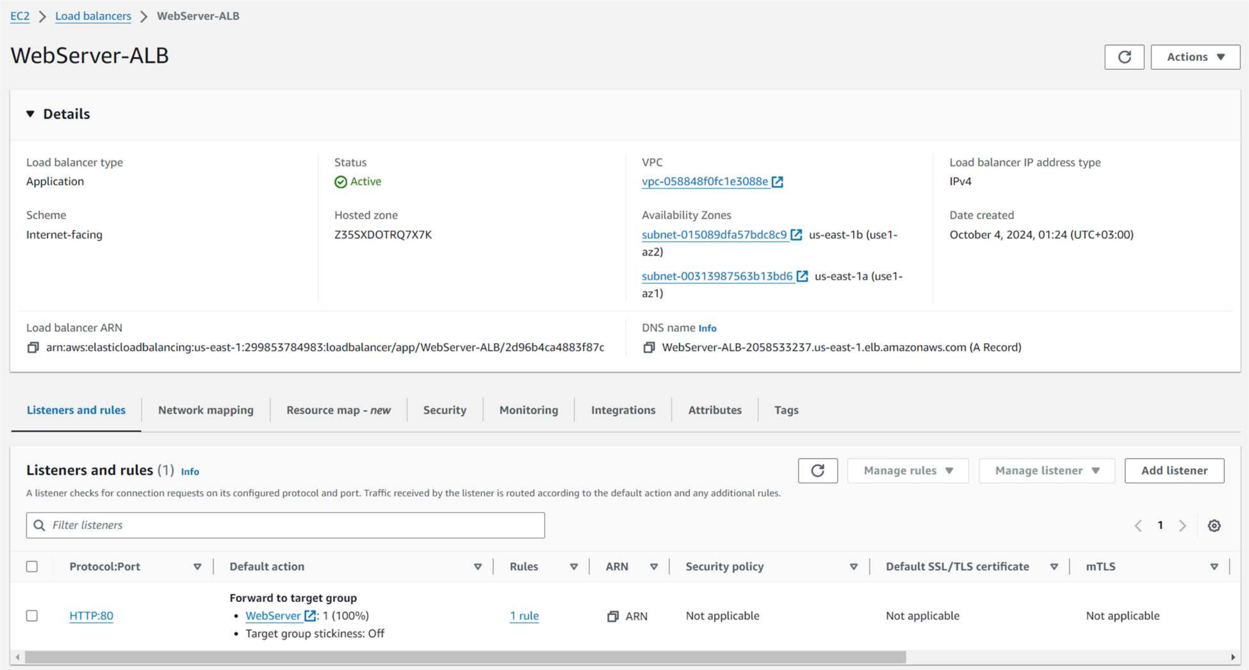
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2) Info

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

☐	Instance ID	Name	Port	Zone	Health status	Health status details	Launch time
☐	i-0ca5fe2037685f893		80	us-east-1a	Unused	Target group is not co...	October 4, 2024, 01:21 (UTC+03:00)
☐	i-0a9dd7fedcdc3d006		80	us-east-1b	Unused	Target group is not co...	October 4, 2024, 01:21 (UTC+03:00)

Then we will create the Application Load Balancer.



EC2 > Load balancers > WebServer-ALB

WebServer-ALB

▼ Details

Load balancer type
Application

Status
Active

VPC
vpc-058848f0fc1e3088e

Load balancer IP address type
IPv4

Scheme
Internet-facing

Hosted zone
Z35SXDOTRQ7X7K

Availability Zones
subnet-015089dfa57bdc8c9 us-east-1b (use1-az2)
subnet-00313987563b13bd6 us-east-1a (use1-az1)

Date created
October 4, 2024, 01:24 (UTC+03:00)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:299853784983:loadbalancer/app/WebServer-ALB/2d96b4ca4883f87c

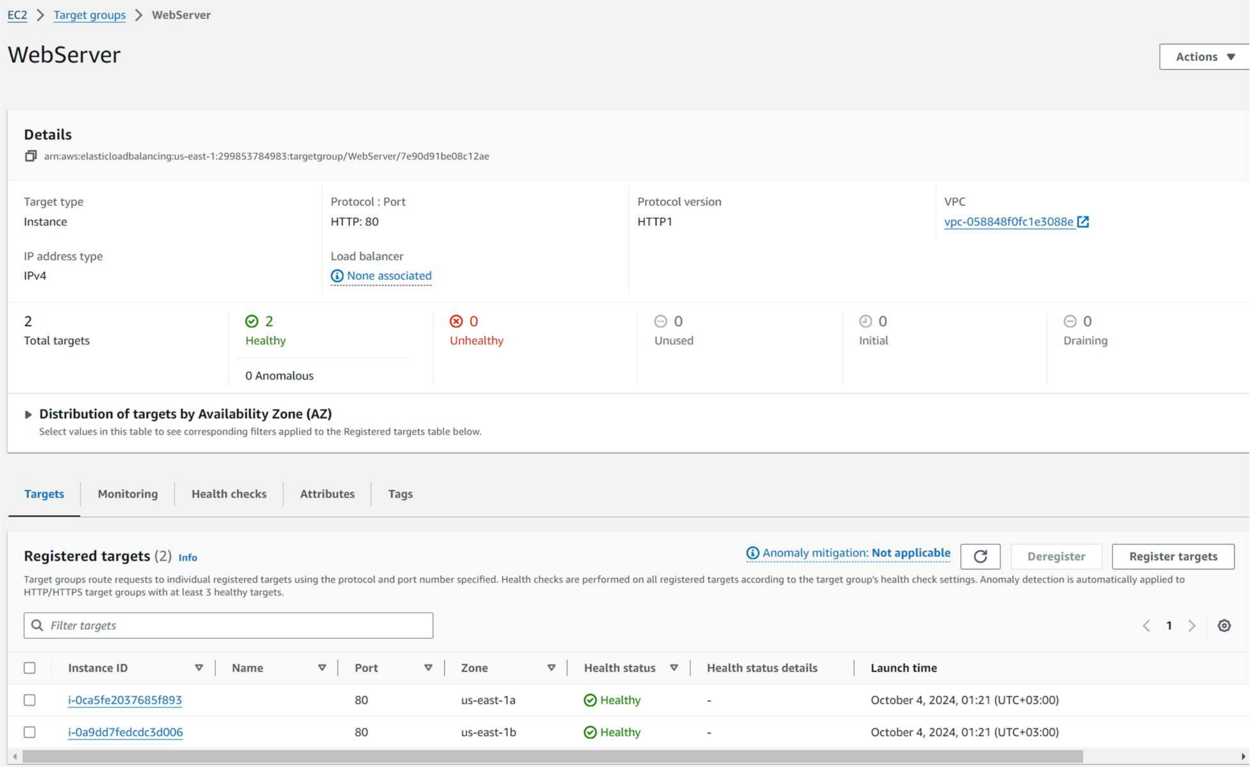
DNS name Info
WebServer-ALB-2058533237.us-east-1.elb.amazonaws.com (A Record)

Listeners and rules (1) Info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

□	Protocol:Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS
□	HTTP:80	**Forward to target group** • WebServer: 1 (100%) • Target group stickiness: Off	1 rule	ARN	Not applicable	Not applicable	Not applicable

And now the targets are receiving monitoring and their status became healthy



EC2 > Target groups > WebServer

WebServer

Details
arn:aws:elasticloadbalancing:us-east-1:299853784983:targetgroup/WebServer/7e90d91be08c12ae

Target type
Instance

Protocol : Port
HTTP: 80

Protocol version
HTTP1

VPC
vpc-058848f0fc1e3088e

IP address type
IPv4

Load balancer
None associated

2
Total targets

2
Healthy

0 Anomalous

0
Unhealthy

0
Unused

0
Initial

0
Draining

▶ Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Registered targets (2) Info

Anomaly mitigation: Not applicable

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

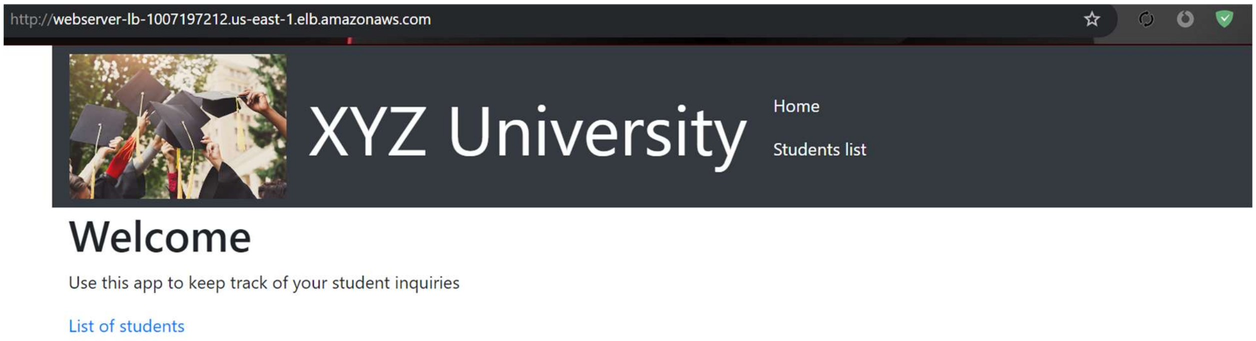
□	Instance ID	Name	Port	Zone	Health status	Health status details	Launch time
□	i-0ca5fe2037685f893		80	us-east-1a	✓ Healthy	-	October 4, 2024, 01:21 (UTC+03:00)
□	i-0a9dd7fedcdc3d006		80	us-east-1b	✓ Healthy	-	October 4, 2024, 01:21 (UTC+03:00)

Task 2: Implementing Amazon EC2 Auto Scaling

Created the Auto Scaling Group in the previous step.

Task 3: Accessing the application

By using the public DNS name of the Application Load Balancer, we can test that we can access the application.



http://webserver-lb-1007197212.us-east-1.elb.amazonaws.com

XYZ University

Home

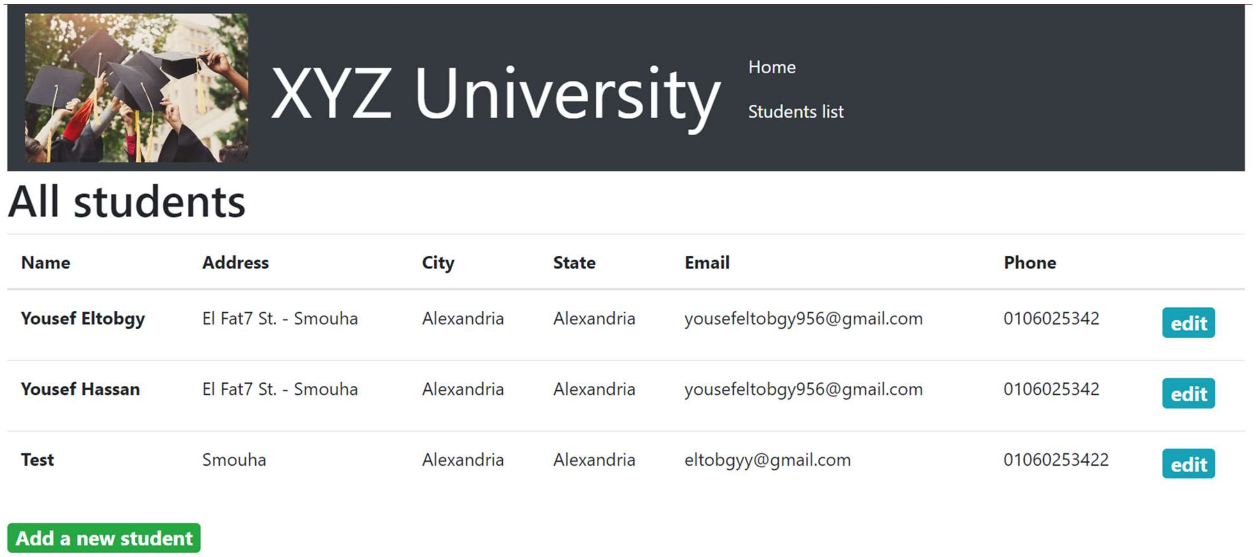
Students list

Welcome

Use this app to keep track of your student inquiries

List of students

And we will find the data from the RDS instance



XYZ University

Home

Students list

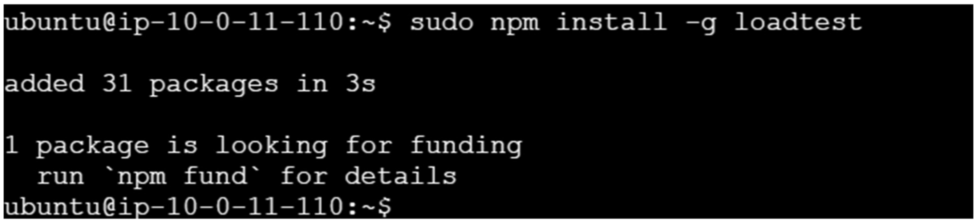
All students

Name	Address	City	State	Email	Phone	
Yousef Eltobgy	El Fat7 St. - Smouha	Alexandria	Alexandria	yousefeltobgy956@gmail.com	0106025342	edit
Yousef Hassan	El Fat7 St. - Smouha	Alexandria	Alexandria	yousefeltobgy956@gmail.com	0106025342	edit
Test	Smouha	Alexandria	Alexandria	eltobgyy@gmail.com	01060253422	edit

Add a new student

Task 4: Load testing the application

We will now try doing load testing on the Application Load Balancer that we created. So we will install the loadtest package using the below command.

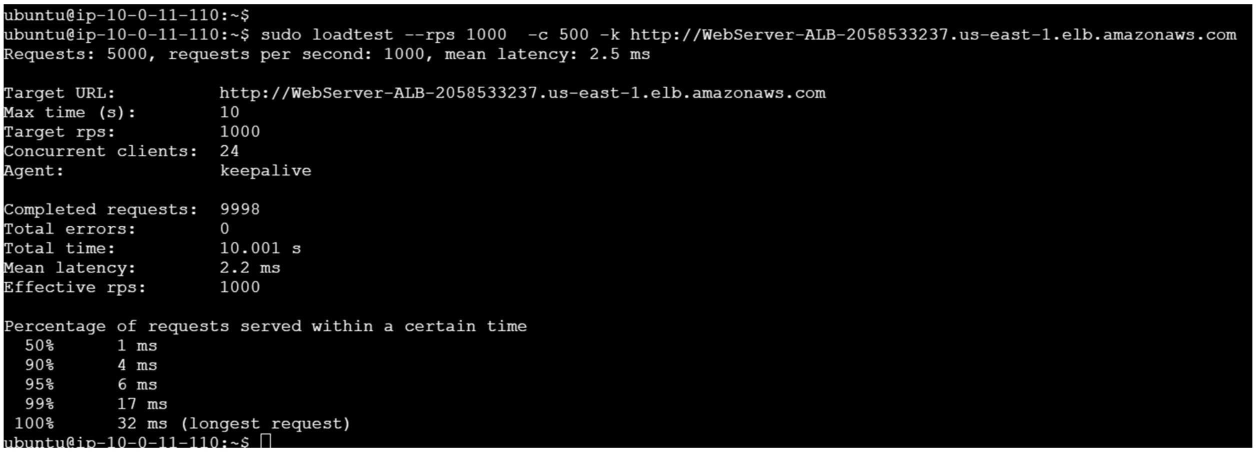


```
ubuntu@ip-10-0-11-110:~$ sudo npm install -g loadtest

added 31 packages in 3s

1 package is looking for funding
  run `npm fund` for details
ubuntu@ip-10-0-11-110:~$
```

Then we will run the loadtest command on the ALB.



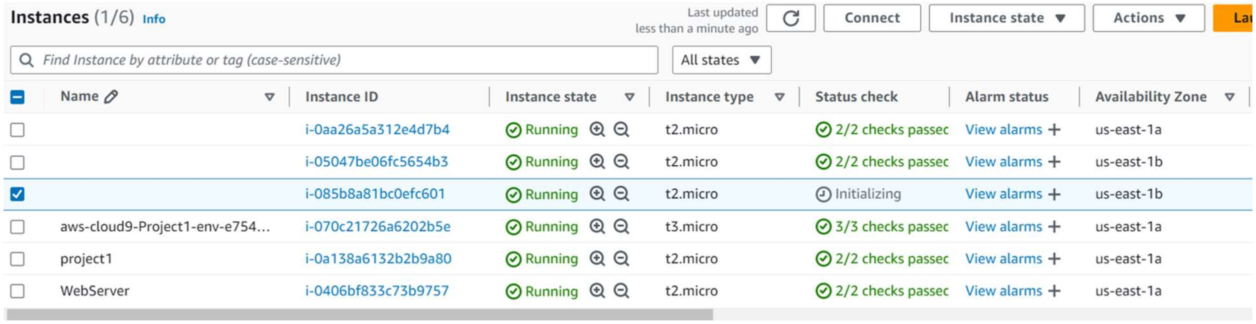
```
ubuntu@ip-10-0-11-110:~$
ubuntu@ip-10-0-11-110:~$ sudo loadtest --rps 1000  -c 500 -k http://WebServer-ALB-2058533237.us-east-1.elb.amazonaws.com
Requests: 5000, requests per second: 1000, mean latency: 2.5 ms

Target URL:          http://WebServer-ALB-2058533237.us-east-1.elb.amazonaws.com
Max time (s):        10
Target rps:          1000
Concurrent clients:  24
Agent:               keepalive

Completed requests:  9998
Total errors:        0
Total time:          10.001 s
Mean latency:        2.2 ms
Effective rps:       1000

Percentage of requests served within a certain time
  50%      1 ms
  90%      4 ms
  95%      6 ms
  99%      17 ms
 100%      32 ms (longest request)
ubuntu@ip-10-0-11-110:~$ 
```

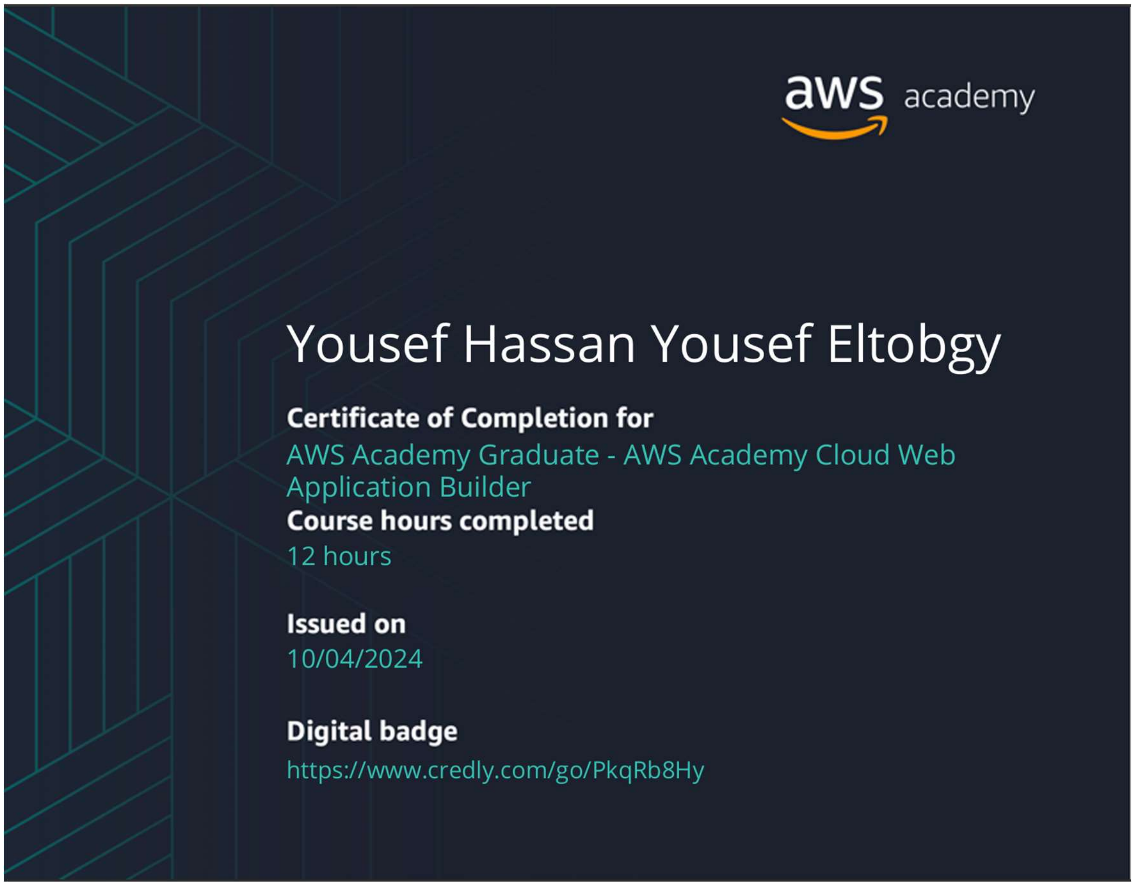
Then we will wait till the Auto Scaling Group create a new instance.



Instances (1/6) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
La
Find Instance by attribute or tag (case-sensitive)
All states ▼

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐		i-0aa26a5a312e4d7b4	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1a
☐		i-05047be06fc5654b3	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1b
☑		i-085b8a81bc0efc601	✓ Running	t2.micro	Initializing	View alarms +	us-east-1b
☐	aws-cloud9-Project1-env-e754...	i-070c21726a6202b5e	✓ Running	t3.micro	✓ 3/3 checks passec	View alarms +	us-east-1a
☐	project1	i-0a138a6132b2b9a80	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1a
☐	WebServer	i-0406bf833c73b9757	✓ Running	t2.micro	✓ 2/2 checks passec	View alarms +	us-east-1a

The Badge:



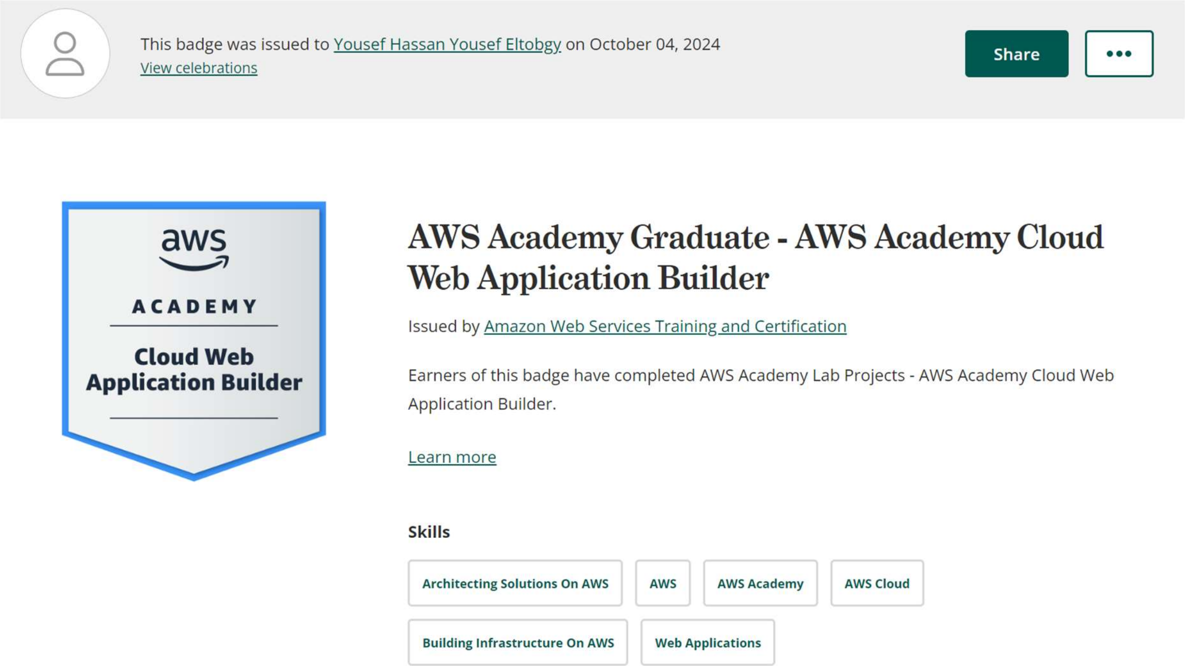
aws academy

Yousef Hassan Yousef Eltobgy

Certificate of Completion for
AWS Academy Graduate - AWS Academy Cloud Web Application Builder
Course hours completed
12 hours

Issued on
10/04/2024

Digital badge
https://www.credly.com/go/PkqRb8Hy



This badge was issued to Yousef Hassan Yousef Eltobgy on October 04, 2024
View celebrations

AWS Academy Graduate - AWS Academy Cloud Web Application Builder

Issued by Amazon Web Services Training and Certification

Earners of this badge have completed AWS Academy Lab Projects - AWS Academy Cloud Web Application Builder.

Learn more

Skills

- Architecting Solutions On AWS
- AWS
- AWS Academy
- AWS Cloud
- Building Infrastructure On AWS
- Web Applications