

Building a Highly Available, Scalable Web Application

Name	Badr Eldin Wael Mohamed Mohamed
Group	ALX1_ISS4_M1e
Student ID	21007401
Badges	https://www.credly.com/users/badr-eldin-wael

Badge Portfolio

 Reorder/Edit

Credly (5)



AWS Academy Graduate - AWS Academy Cloud Foundations

Amazon Web Services Training and Certification

Issued 8/3/24



AWS Academy Graduate - AWS Academy Cloud Architecting

Amazon Web Services Training and Certification

Issued 9/23/24



AWS Academy Graduate - AWS Academy Cloud Operations

Amazon Web Services Training and Certification

Issued 9/29/24



AWS Academy Graduate - AWS Academy Cloud Web Application...

Amazon Web Services Training and Certification

Issued 10/3/24



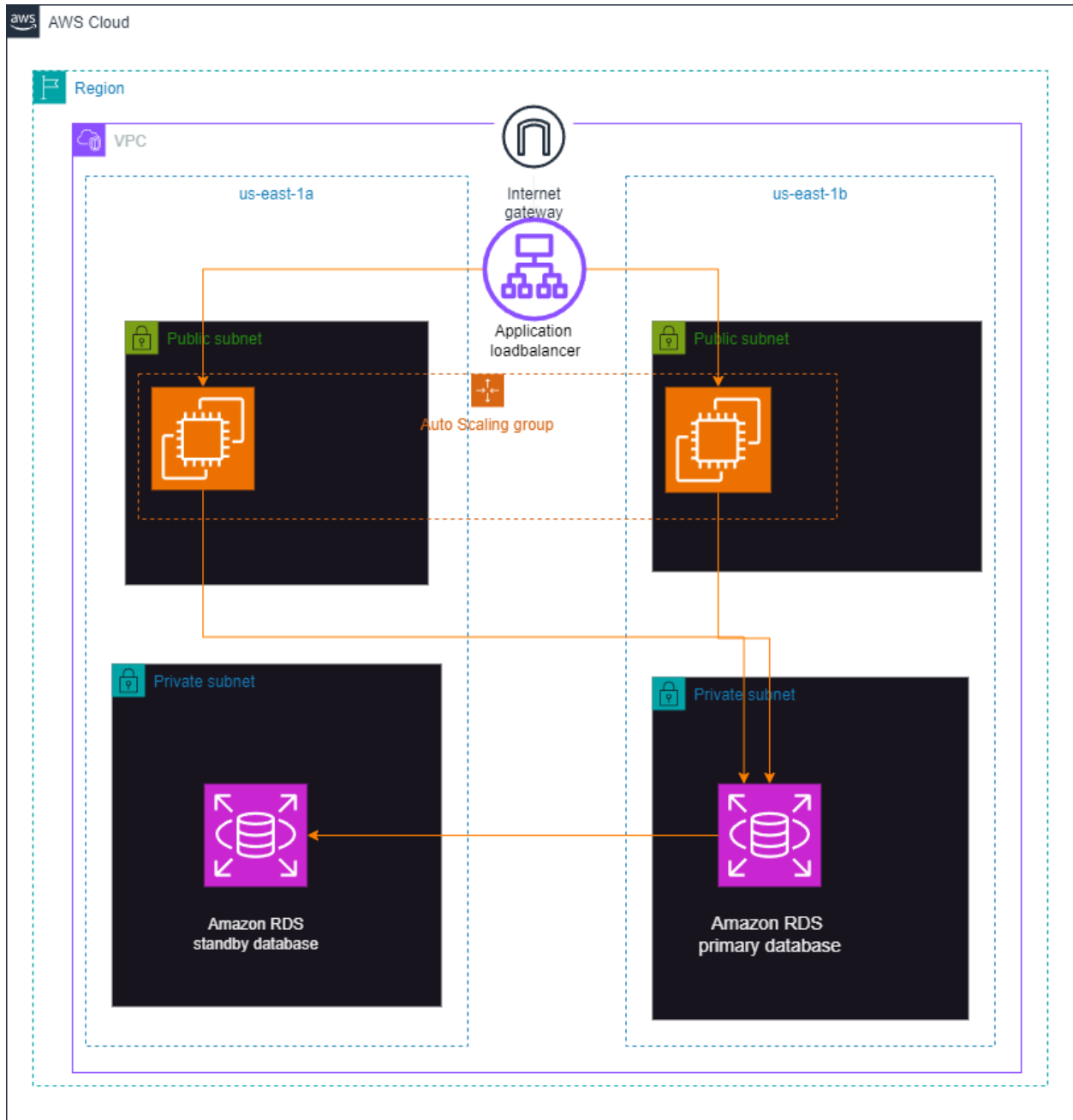
AWS Academy Graduate - AWS Academy Cloud Security Builder

Amazon Web Services Training and Certification

Issued 10/4/24

Phase 1: Planning the design and estimating cost

Task 1: Creating an architectural diagram



Task 2: Developing a cost estimate

Estimate summary								
Upfront cost	Monthly cost	Total 12 months c	Currency					
0	147.12	1765.44	USD					
	* Includes upfront cost							
Detailed Estimate								
Group hierarchy	Region	Description	Service	Upfront	Monthly	First 12 m	Currency	Status Configuration summary
My Estimate	US East (N. Virginia)		Amazon EC2	0	8.468	101.62	USD	Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 2), Advance EC2 instance (t2.micro), Pricing strategy (Compute Savings Plan) (On-Demand only) (60 %Utilized/Mo)
My Estimate	US East (N. Virginia)		Amazon RDS for MySQL	0	19.482	233.9	USD	Storage amount (20 GB), Storage for each RDS Instance (General Purpose SSD (gp2)), Nodes (1), Instance type (db.t3.micro), Utilization (On-Demand only) (60 %Utilized/Mo)
My Estimate	US East (N. Virginia)		Application Load Balancing	0	17.6	211.2	USD	Number of Application Load Balancers (1)
My Estimate	US East (N. Virginia)		AWS Data Transfer	0	92.16	1105.92	USD	DT Inbound: Not selected (100 TB per month), DT Outbound: Internet (1 TB per month), DT Intra-Region: (0 TB per month), Data transfer cost (\$0.09/GB)
My Estimate	US East (N. Virginia)		AWS Secrets Manager	0	0.4	4.8	USD	Number of secrets (1), Average duration of each secret (30 days), Number of API calls (50 per month)
My Estimate	US East (N. Virginia)		Amazon Elastic Block Store	0	9	108	USD	Number of volumes (2), Average duration each instance runs (730 hours per month), Storage amount per volume (20 GB), Snapshot Frequency (Daily), Amount changed per month (0 GB)
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

-Creating a Vpc for project

Your VPCs (1/2) [Info](#)

Last updated less than a minute ago [Refresh](#) [Actions](#) [Create VPC](#)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main
<input checked="" type="checkbox"/> project1vpc-vpc	vpc-0bb0056905b56b5b5	Available	10.0.0.0/16	-	dopt-0902d12f62e92b...	rtb-08dfa9363fe57d6d0	acl-
<input type="checkbox"/> -	vpc-08dafa7852d6f1315	Available	172.31.0.0/16	-	dopt-0902d12f62e92b...	rtb-02d8e2b626383d858	acl-

vpc-0bb0056905b56b5b5 / project1vpc-vpc

[Details](#) | [Resource map](#) | [CIDRs](#) | [Flow logs](#) | [Tags](#) | [Integrations](#)

Resource map [Info](#)

```

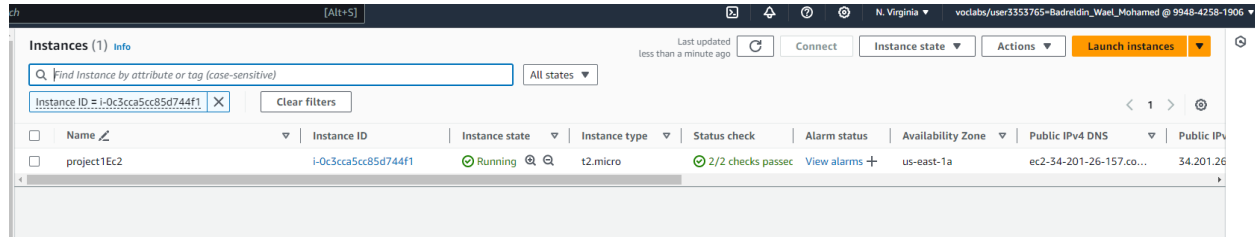
graph LR
    subgraph VPC [VPC]
        direction TB
        S1[us-east-1a  
project1vpc-subnet-public-1-us-ea...]
        S2[us-east-1a  
project1vpc-subnet-private1-us-e...]
        S3[us-east-1b  
project1vpc-subnet-public2-us-ea...]
        S4[us-east-1b  
project1vpc-subnet-private2-us-e...]
    end
    
    subgraph RTB [Route tables]
        direction TB
        RTB1[project1vpc-rtb-public  
rtb-08dfa9363fe57d6d0]
        RTB2[project1vpc-rtb-private1-us-east-1a]
        RTB3[project1vpc-rtb-private2-us-east-1b]
    end
    
    subgraph NC [Network connections]
        direction TB
        NC1[project1vpc-igw]
        NC2[project1-lgw]
    end
    
    S1 --- RTB1
    S2 --- RTB2
    S3 --- RTB3
    S4 --- RTB3
    RTB1 --- NC1
    RTB2 --- NC1
    RTB3 --- NC1
    RTB3 --- NC2
  
```

The resource map illustrates the components of the VPC:

- VPC:** project1vpc-vpc
- Subnets (4):**
 - us-east-1a: project1vpc-subnet-public-1-us-ea..., project1vpc-subnet-private1-us-e...
 - us-east-1b: project1vpc-subnet-public2-us-ea..., project1vpc-subnet-private2-us-e...
- Route tables (4):** project1vpc-rtb-public (rtb-08dfa9363fe57d6d0), project1vpc-rtb-private1-us-east-1a, project1vpc-rtb-private2-us-east-1b.
- Network connections (2):** project1vpc-igw, project1-lgw.

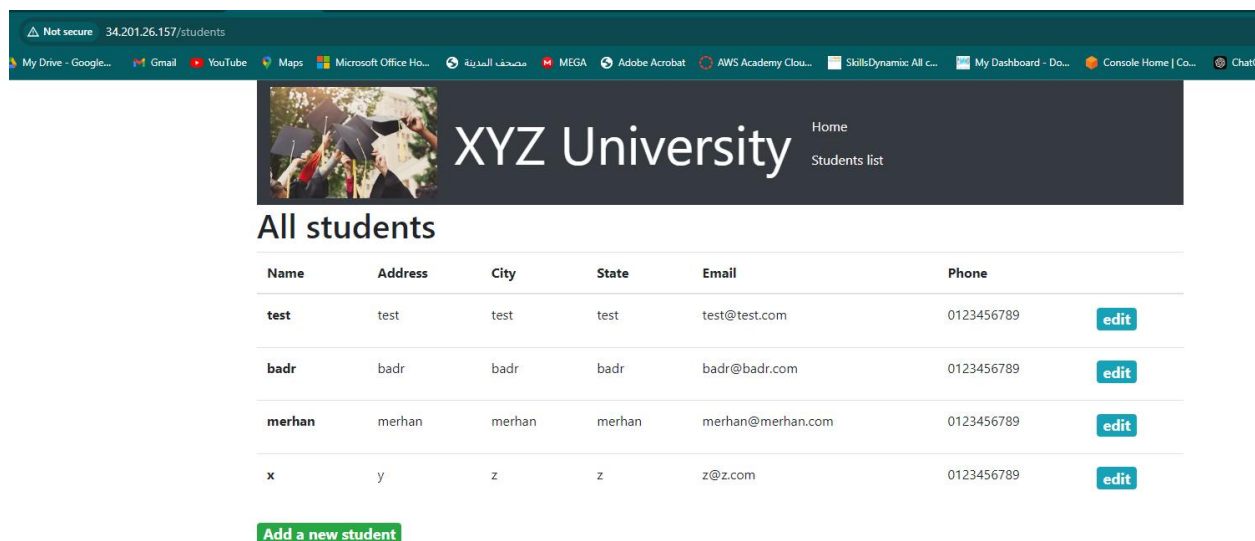
Task 2: Creating a virtual machine

-Creating EV2 instance for testing the deployment of website



Task 3: Testing the deployment

-The result from The first Instance



Phase 3: Decoupling the application components

Task 1: Changing the VPC configuration

-We already made the subnets in task number one using vpc console

Subnets (10) info

Find resources by attribute or tag

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR association ID
-	subnet-058edfb77540838b	Available	vpc-08dafa7852d6f1315	172.31.64.0/20	-	-
-	subnet-04787933e7991c279	Available	vpc-08dafa7852d6f1315	172.31.16.0/20	-	-
project1vpc-subnet-public2-us-east-1b	subnet-0fb56a937442e3de6	Available	vpc-0bb0056905b56b5b5 proj...	10.0.16.0/20	-	-
project1vpc-subnet-public1-us-east-1a	subnet-0da6dd78ff86ad165	Available	vpc-0bb0056905b56b5b5 proj...	10.0.0.0/20	-	-
-	subnet-04225081ff83ddb44	Available	vpc-08dafa7852d6f1315	172.31.0.0/20	-	-
-	subnet-09938d0697da57dc7	Available	vpc-08dafa7852d6f1315	172.31.80.0/20	-	-
-	subnet-0259427ac88873acf	Available	vpc-08dafa7852d6f1315	172.31.48.0/20	-	-
project1vpc-subnet-private2-us-east-1b	subnet-06564c5da570dc647	Available	vpc-0bb0056905b56b5b5 proj...	10.0.144.0/20	-	-
project1vpc-subnet-private1-us-east-1a	subnet-096b6eb80766f327ea	Available	vpc-0bb0056905b56b5b5 proj...	10.0.128.0/20	-	-
-	subnet-0b14964aada95b5e	Available	vpc-08dafa7852d6f1315	172.31.32.0/20	-	-

Task 2: Creating and configuring the Amazon RDS database

-RDS database is being configured

RDS > Databases

Databases (1)

Filter by databases

DB identifier	Status	Role	Engine	Region ...	Size	Recommendations	CPU	Current...	Mainte
project-db	Available	Instance	MySQL Co...	us-east-1b	db.t3.micro	1 Informational	3.29%	1 Conn	none

Task 3: Configuring the development environment

-Creating the Cloud9 for the later scripts and commands

AWS Cloud9 > Environments

Environments (1)

My environments

Name	Cloud9 IDE	Environment type	Connection	Permission	Owner ARN
projectide	Open	EC2 instance	Secure Shell (SSH)	Owner	arn:aws:sts::994842581906:assumed-role/voclabs/user3353765-Badreldin_Wael_Mohamed

Task 4: Provisioning Secrets Manager

-Making Secrets using cli and scripts

AWS Secrets Manager > Secrets

Secrets

Filter secrets by name, description, tag key, tag value, owning service or primary Region

Secret name	Description	Last retrieved (UTC)
Mydbsecret	Database secret for web app	September 27, 2024

Task 5: Provisioning a new instance for the web server

-We will take an ami from this instance later to use it in the launch template

The screenshot displays the AWS Management Console interface for an EC2 instance. At the top, a table lists instances, with one instance named 'Web server' (ID: i-0ce1a729561bad663) in a 'Running' state. Below this, the 'Details' tab for the selected instance is shown. The instance summary includes the following information:

- Instance ID:** i-0ce1a729561bad663 (Web server)
- IPv6 address:** -
- Hostname type:** IP name: ip-10-0-7-150.ec2.internal
- Answer private resource DNS name:** -
- Auto-assigned IP address:** 3.228.18.5 [Public IP]
- Public IPv4 address:** 3.228.18.5 | [open address](#)
- Instance state:** Running
- Private IP DNS name (IPv4 only):** ip-10-0-7-150.ec2.internal
- Instance type:** t2.micro
- VPC ID:** vpc-0bb0056905b56b5b5 (project1-vpc-vpc)
- Private IPv4 addresses:** 10.0.7.150
- Public IPv4 DNS:** ec2-3-228-18-5.compute-1.amazonaws.com | [open address](#)
- Elastic IP addresses:** -
- AWS Compute Optimizer finding:** Opt-in to AWS Compute Optimizer for recommendations. | [Learn more](#)

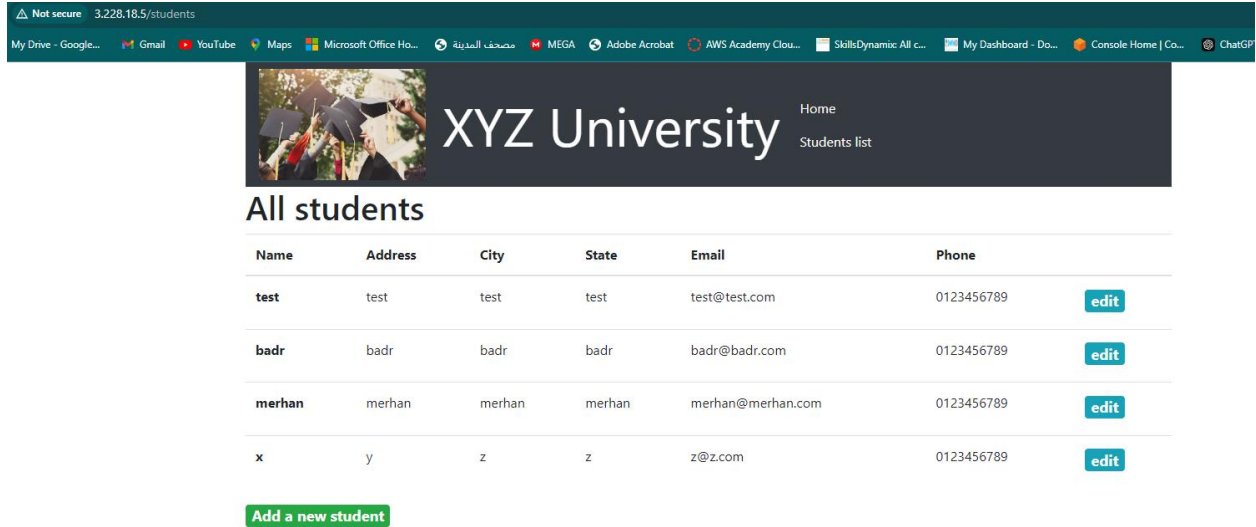
Task 6: Migrating the database

-The following commands will be written in the cloud 9 to migrate the sqldump n the instance in the data base

```
mysqldump -h 10.0.6.221 -u nodeapp -p --databases STUDENTS > data.sql
student12
mysql -u admin -p -h project-db.cergcsua0bxo.us-east-1.rds.amazonaws.com < data.sql
```

Task 7: Testing the application

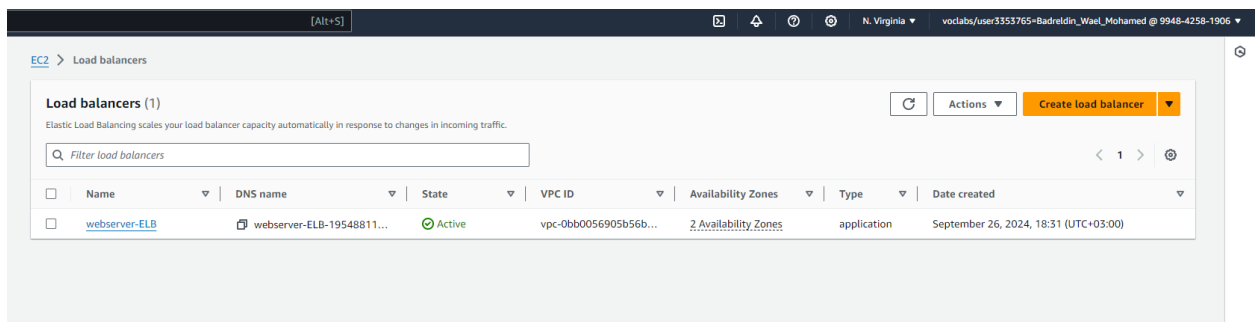
-in the webserver instance that we created



Phase 4: Implementing high availability and scalability

Task 1: Creating an Application Load Balancer

-To implement a high availability and scalability we need to setup application load balancer



Task 2: Implementing Amazon EC2 Auto Scaling

-In the following figures we setup up the launch template ,autoscaling group , ami and the target groups to make a high available and scalable web application

[Alt+S]

N. Virginia

voclabs/user3353765-Badreldin_Wael_Mohamed @ 9948-4258-1906

Launch Templates (1) Info

🔄

Actions

Create launch template

🔍 Search

< 1 > ⚙️

<input type="checkbox"/>	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
<input type="checkbox"/>	lt-Oddc93f4a5ce87f0d	WebTemplate	1	1	2024-09-26T16:59:32.000Z	arn:aws:sts:994842581906:assumed-role/voclabs/user3353765-Badreldi...

[Alt+S]

N. Virginia

voclabs/user3353765-Badreldin_Wael_Mohamed @ 9948-4258-1906

Amazon Machine Images (AMIs) (1) Info

🔄

Recycle Bin

EC2 Image Builder

Actions

Launch instance from AMI

Owned by me

🔍 Find AMI by attribute or tag

< 1 > ⚙️

<input type="checkbox"/>	Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation date
<input type="checkbox"/>	WebserverImage		ami-06ca9c4a25a38f6d	994842581906/WebserverImage	994842581906	Private	Available	2024/09/26 19:5

[Alt+S]

N. Virginia

voclabs/user3353765-Badreldin_Wael_Mohamed @ 9948-4258-1906

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info

🔄

Launch configurations

Launch templates

Actions

Create Auto Scaling group

🔍 Search your Auto Scaling groups

< 1 > ⚙️

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
<input type="checkbox"/>	WebserverScale	WebTemplate Version Default	2	-	2	1	3	us-east-1a, us-east-1b

[Alt+S]

N. Virginia

voclabs/user3353765-Badreldin_Wael_Mohamed @ 9948-4258-1906

EC2 > Target groups

Target groups (1/1) Info

🔄

Actions

Create target group

🔍 Filter target groups

< 1 > ⚙️

<input checked="" type="checkbox"/>	Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
<input checked="" type="checkbox"/>	webservertarget	arn:aws:elasticloadbalanci...	80	HTTP	Instance	webserver-ELB	vpc-0bb0056905b56b5b5

Target group: webservertarget

Details

Targets

Monitoring

Health checks

Attributes

Tags

Details

arn:aws:elasticloadbalancing:us-east-1:994842581906:targetgroup/webservertarget/f2ab78607456f11d

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-0bb0056905b56b5b5
IP address type	Load balancer		
IPv4	webserver-ELB		

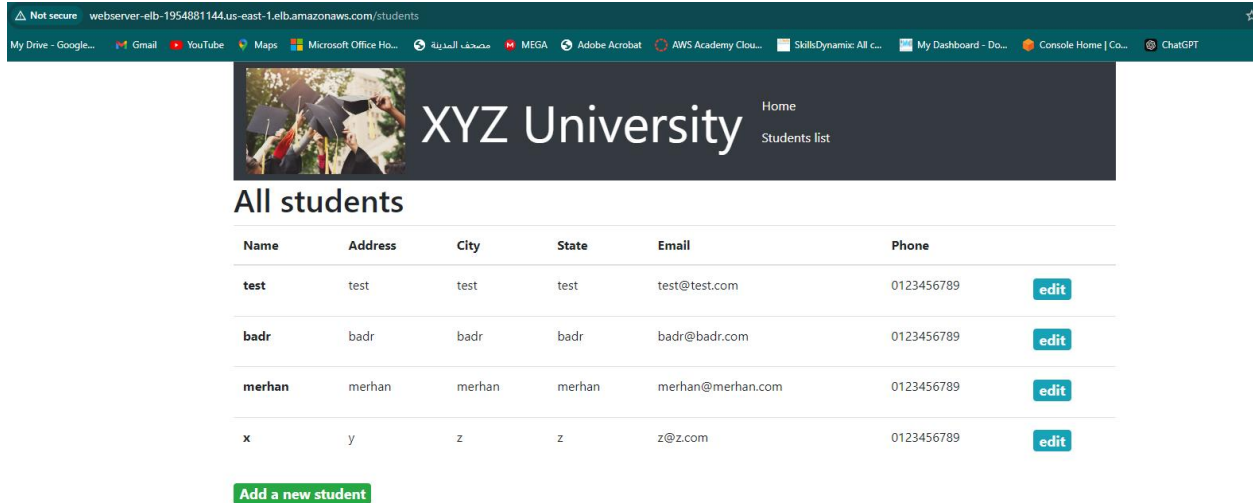
2	🟢 2 Healthy	🔴 0 Unhealthy	🟡 0 Unused	🟡 0 Initial	🟡 0 Draining
Total targets					
	0 Anomalous				

► Distribution of targets by Availability Zone (AZ)

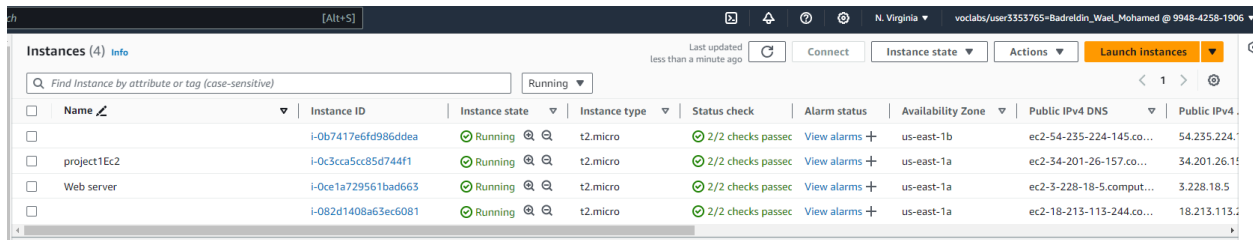
Select values in this table to see corresponding filters applied to the Registered targets table below.

Task 3: Accessing the application

-this figure is from dns of the load balancer that proof that the whole process is working



-the new instances that are created



Task 4: Load testing the application

-the Load test that profess that the load balancer work

```

voclabs:~/environment $ loadtest --rps 1000 -c 500 -k http://webserver-ELB-1954881144.us-east-1.elb.amazonaws.com
Requests: 4998, requests per second: 1000, mean latency: 5.5 ms

Target URL:      http://webserver-ELB-1954881144.us-east-1.elb.amazonaws.com
Max time (s):    10
Target rps:      1000
Concurrent clients: 34
Agent:           keepalive

Completed requests: 9997
Total errors:      0
Total time:        10 s
Mean latency:      4.2 ms
Effective rps:     1000

Percentage of requests served within a certain time
 50%    2 ms
 90%    9 ms
 95%   18 ms
 99%   40 ms
100%  111 ms (longest request)

```

-This is the security groups that are created in the whole process

Security Groups (6) Info							
Find resources by attribute or tag							
<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Owner	
<input type="checkbox"/>	-	sg-0159bb3dbd71a8c87	db-sg	vpc-0bb0056905b56b5b5	allow mysql	994842581906	
<input type="checkbox"/>	-	sg-0ed4945464217f853	default	vpc-08dafa7852d6f1315	default VPC security group	994842581906	
<input type="checkbox"/>	-	sg-0307717afb9687841	Ec2-sg	vpc-0bb0056905b56b5b5	Instance sg group	994842581906	
<input type="checkbox"/>	-	sg-04e40c8531914788	default	vpc-0bb0056905b56b5b5	default VPC security group	994842581906	