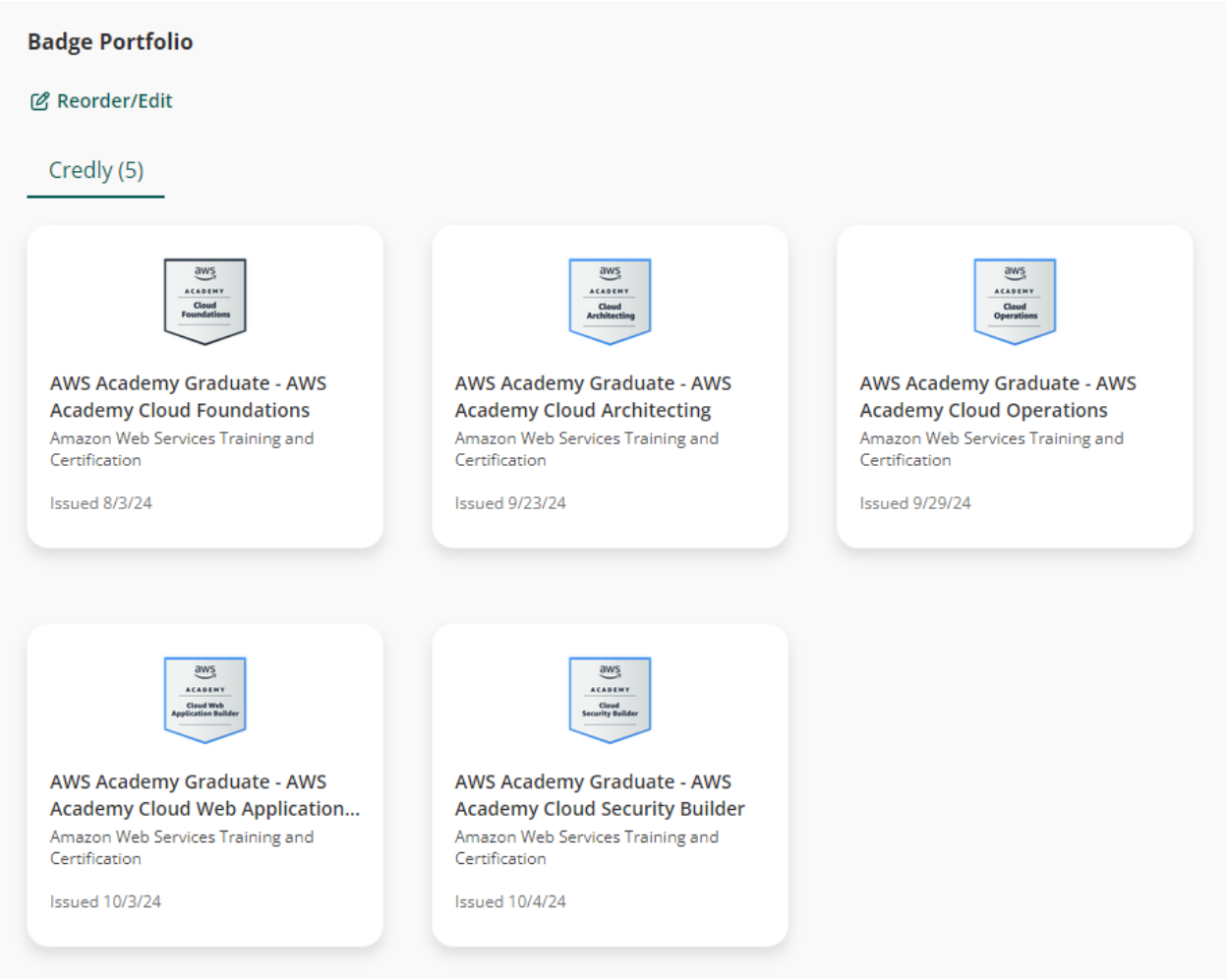
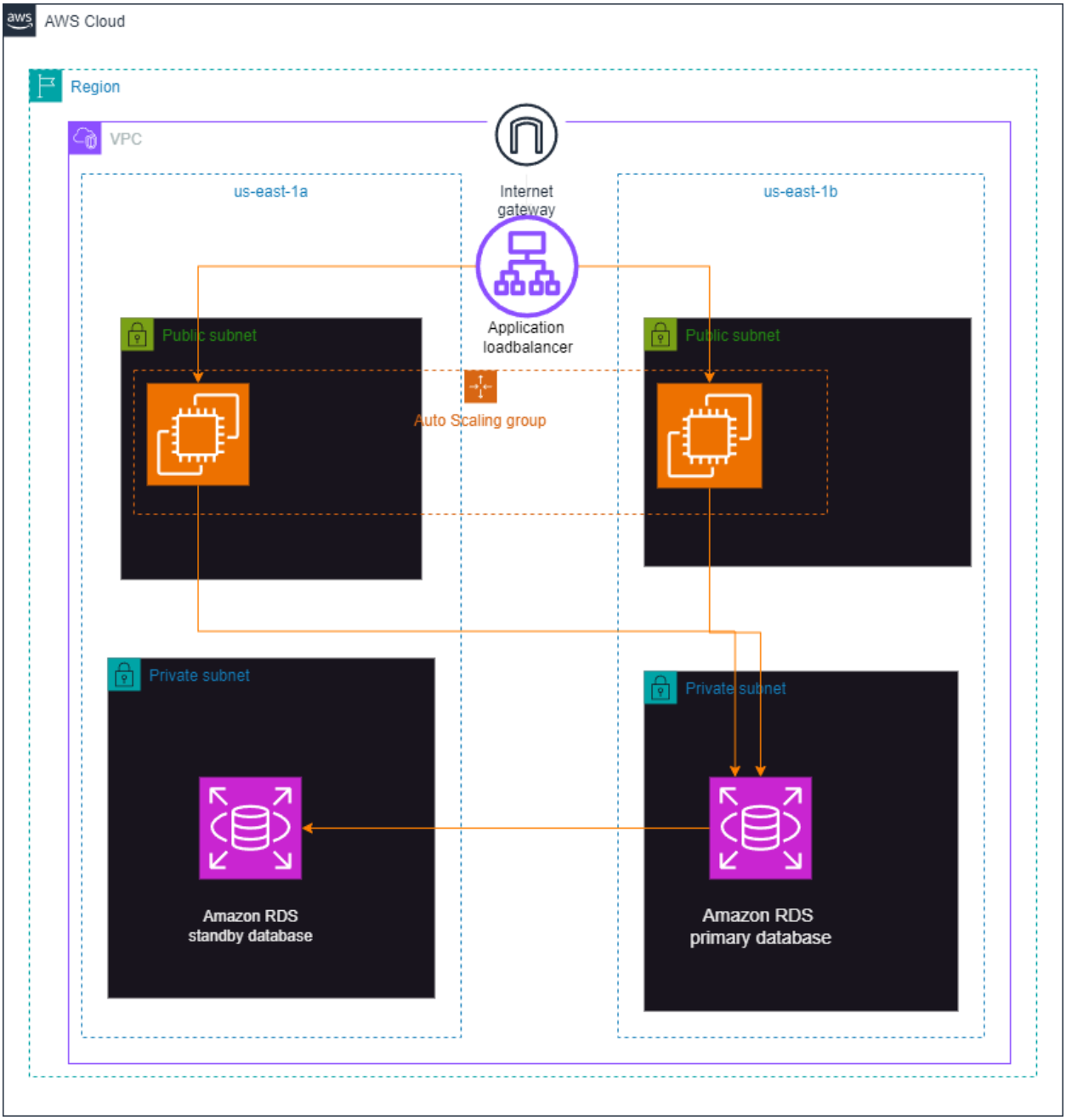
Building a Highly Available, Scalable Web Application

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



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
My Estimate	US East (N. Virginia)		Amazon RDS for M	0	19.492	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 E	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 (92.16)
My Estimate	US East (N. Virginia)		AWS Secrets Mana	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 Bl	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

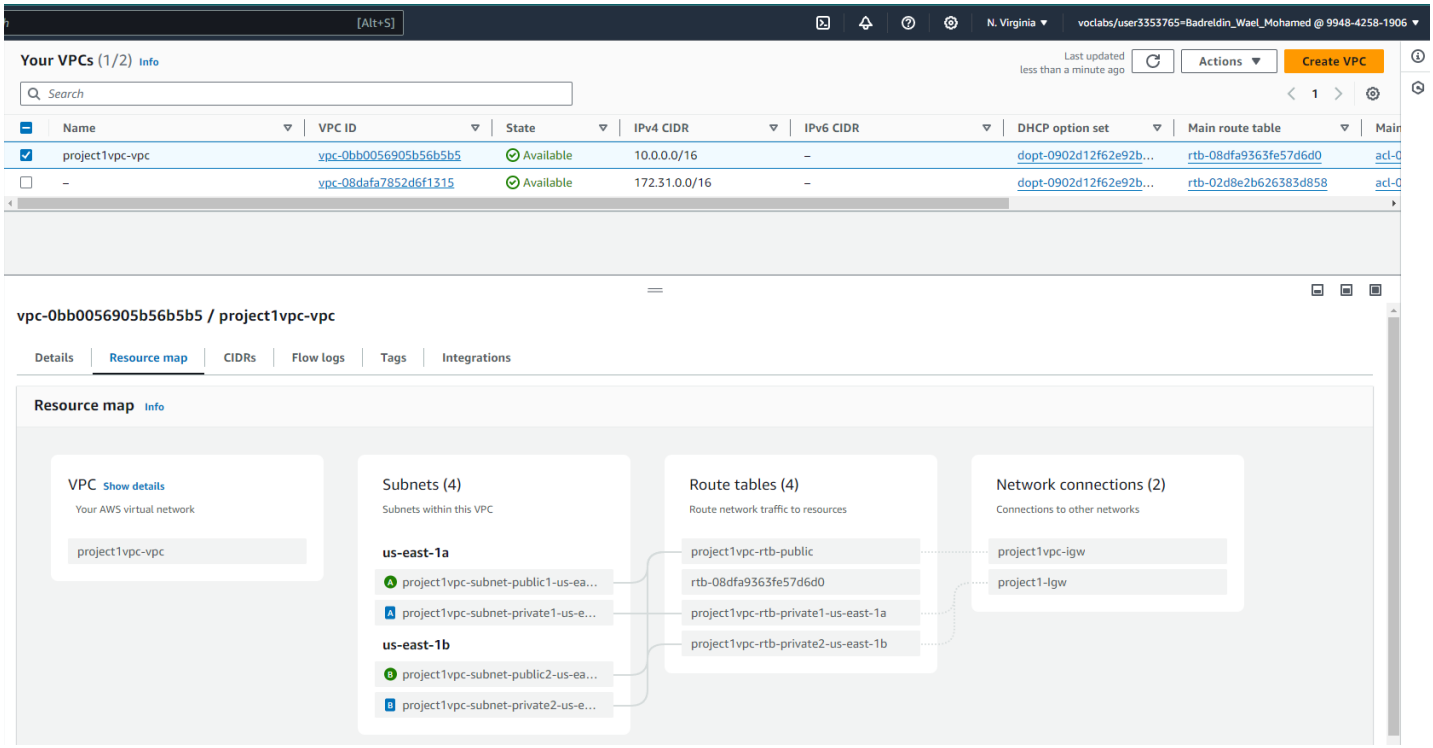
Acknowledgement

* 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

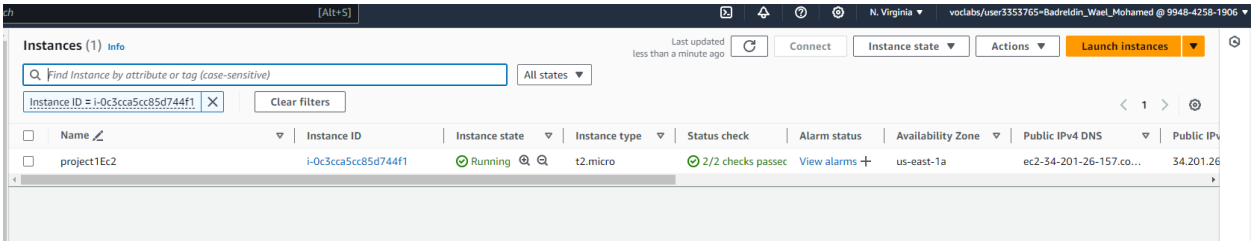
Task 1: Creating a virtual network

-Creating a Vpc for project



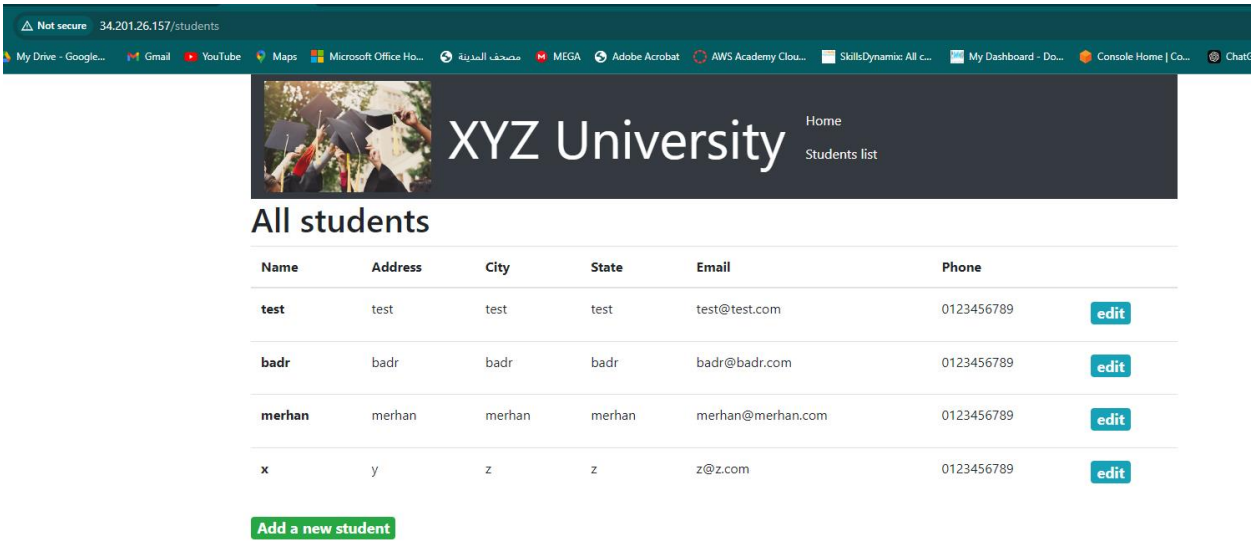
Task 2: Creating a virtual machine

-Creating EV2 instance for testing the deployment of website



Task 3: Testing the deployment

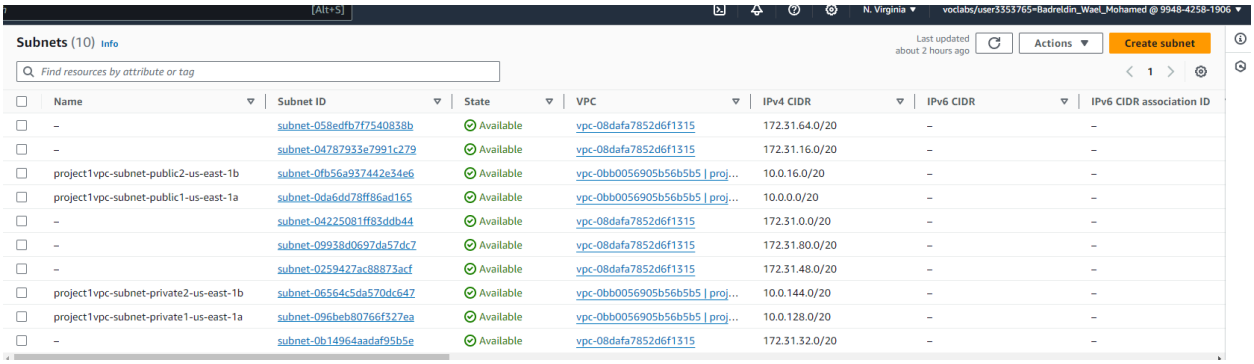
-The result from The first Instance



Phase 3: Decoupling the application components

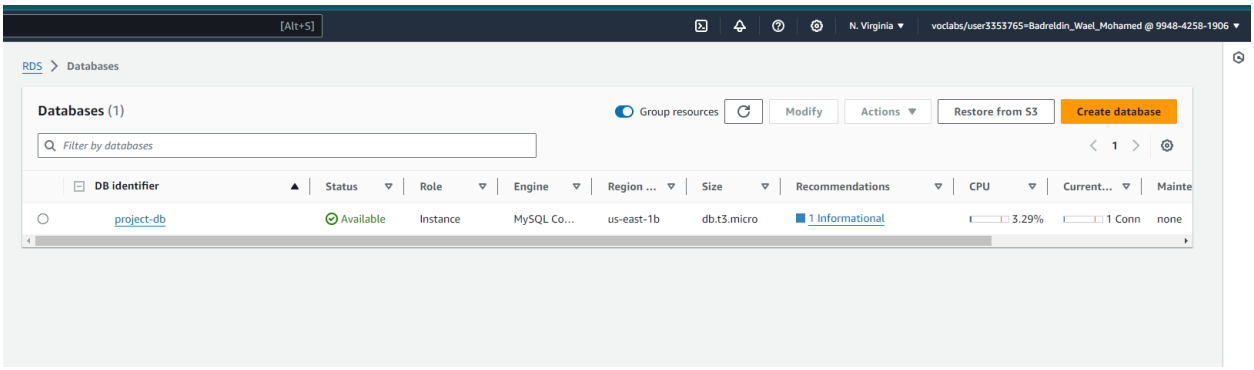
Task 1: Changing the VPC configuration

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



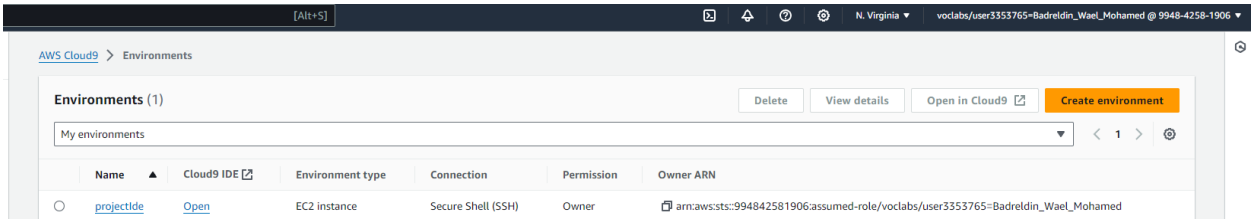
Task 2: Creating and configuring the Amazon RDS database

-RDS database is being configured



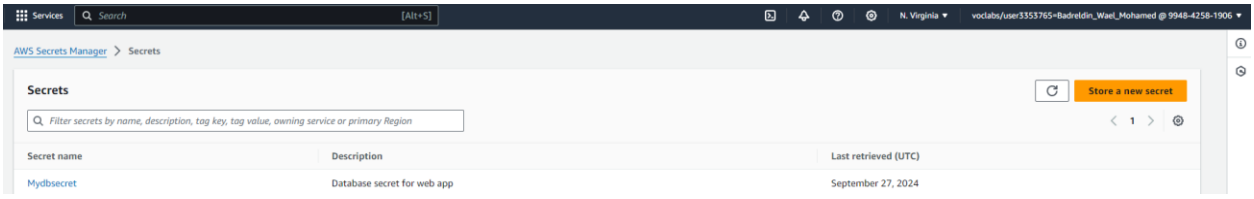
Task 3: Configuring the development environment

-Creating the Cloud9 for the later scripts and commands



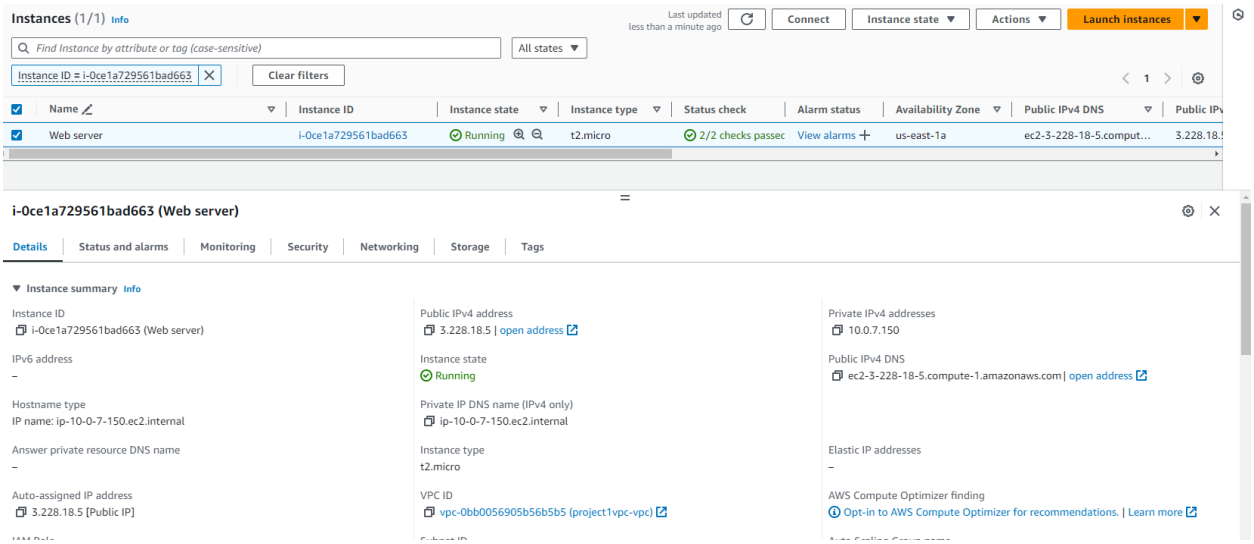
Task 4: Provisioning Secrets Manager

-Making Secrets using cli and scripts



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



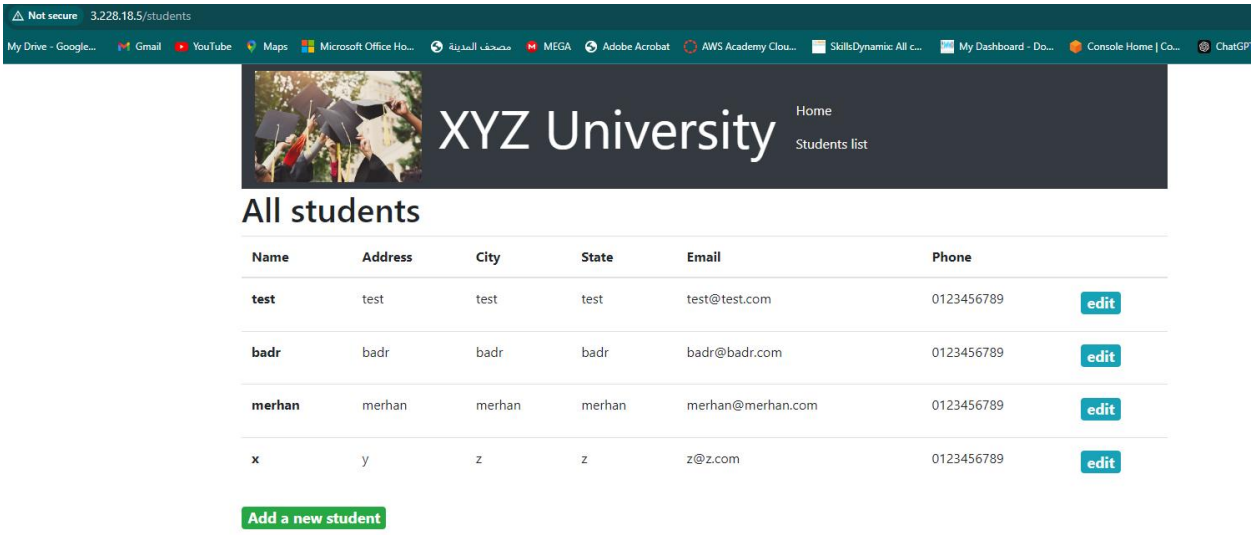
Task 6: Migrating the database

-The following commands wil 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

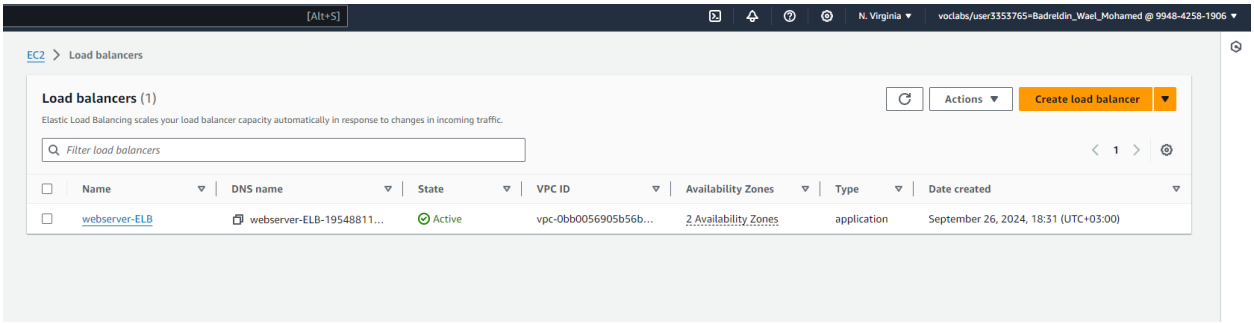
-in the webserver instance that we created



Phase 4: Implementing high availability and scalability

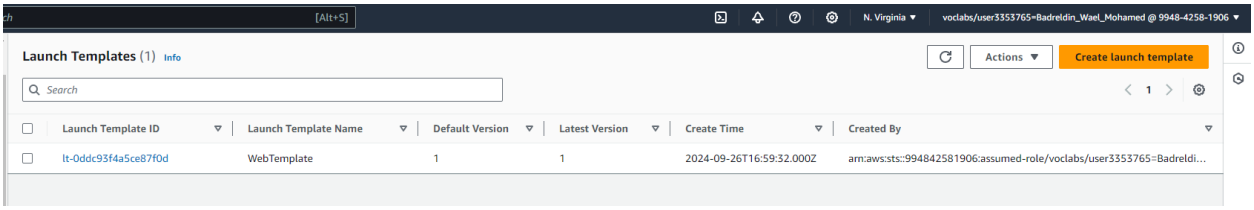
Task 1: Creating an Application Load Balancer

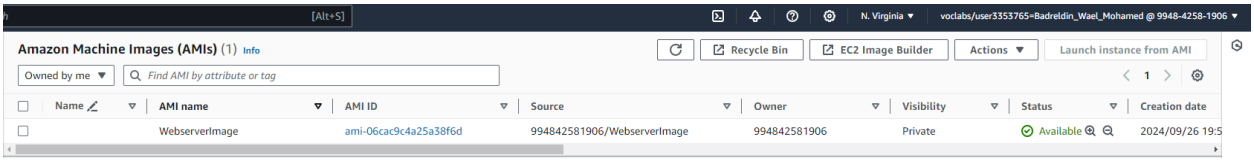
-To implement a high availability and scalability we need to setup application load balancer

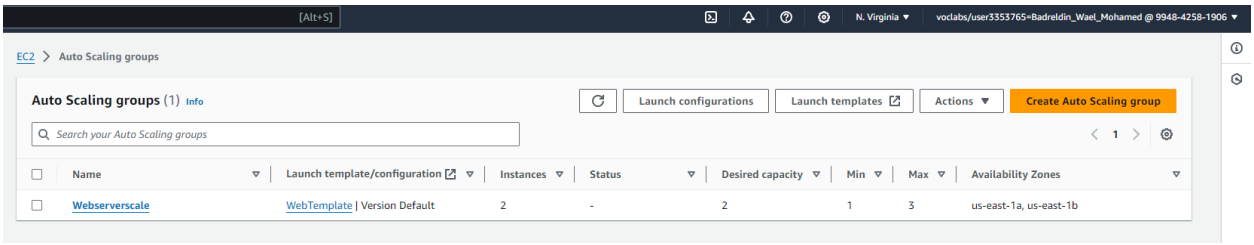


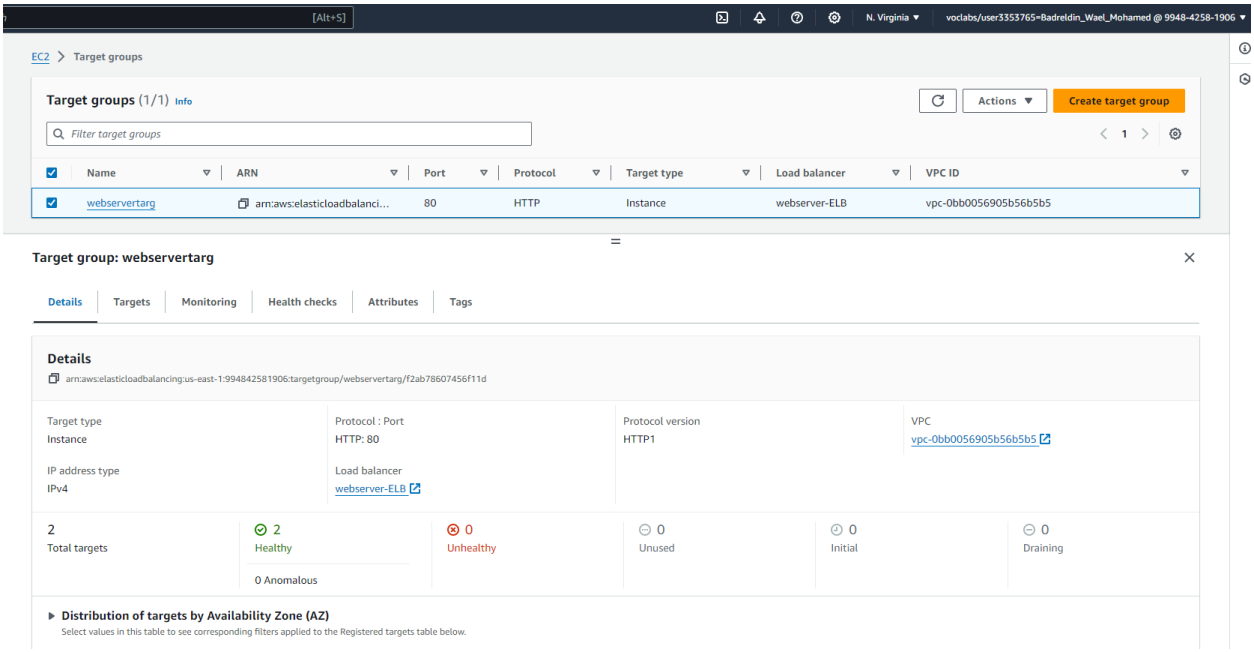
Task 2: Implementing Amazon EC2 Auto Scaling

-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
Launch Template ID
Launch Template Name
Default Version
Latest Version
Create Time
Created By
lt-0ddc93f4a5ce87f0d
WebTemplate
1
1
2024-09-26T16:59:32.000Z
arn:aws:sts::994842581906:assumed-role/voclabs/user3353765=Badreldi...

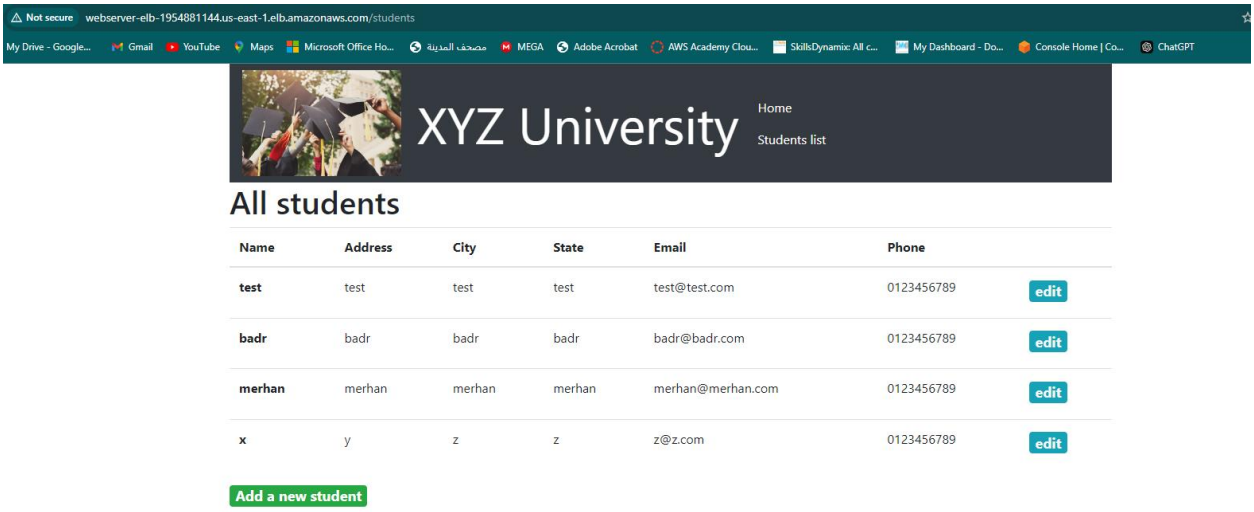

[Alt+S]
N. Virginia
voclabs/user3353765=Badreldin_Wael_Mohamed @ 9948-4258-1906
Amazon Machine Images (AMIs) (1) Info
Recycle Bin
EC2 Image Builder
Actions
Launch instance from AMI
Owned by me
Find AMI by attribute or tag
Name
AMI name
AMI ID
Source
Owner
Visibility
Status
Creation date
WebserverImage
ami-06cac9c4a25a38f6d
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
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
Availability Zones
Webserverscale
WebTemplate | Version Default
2
-
2
1
3
us-east-1a, us-east-1b

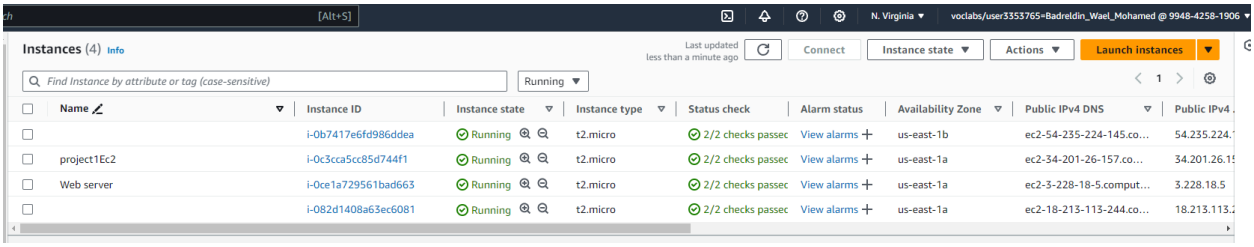

[Alt+S]
N. Virginia
voclabs/user3353765=Badreldin_Wael_Mohamed @ 9948-4258-1906
EC2 > Target groups
Target groups (1/1) Info
Actions
Create target group
Filter target groups
Name
ARN
Port
Protocol
Target type
Load balancer
VPC ID
webservertarg
arn:aws:elasticloadbalanci...
80
HTTP
Instance
webserver-ELB
vpc-0bb0056905b56b5b5
Target group: webservertarg
Details
Targets
Monitoring
Health checks
Attributes
Tags
Details
arn:aws:elasticloadbalancing:us-east-1:994842581906:targetgroup/webservertarg/f2ab78607456f11d
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-0bb0056905b56b5b5
IP address type
IPv4
Load balancer
webserver-ELB
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.

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