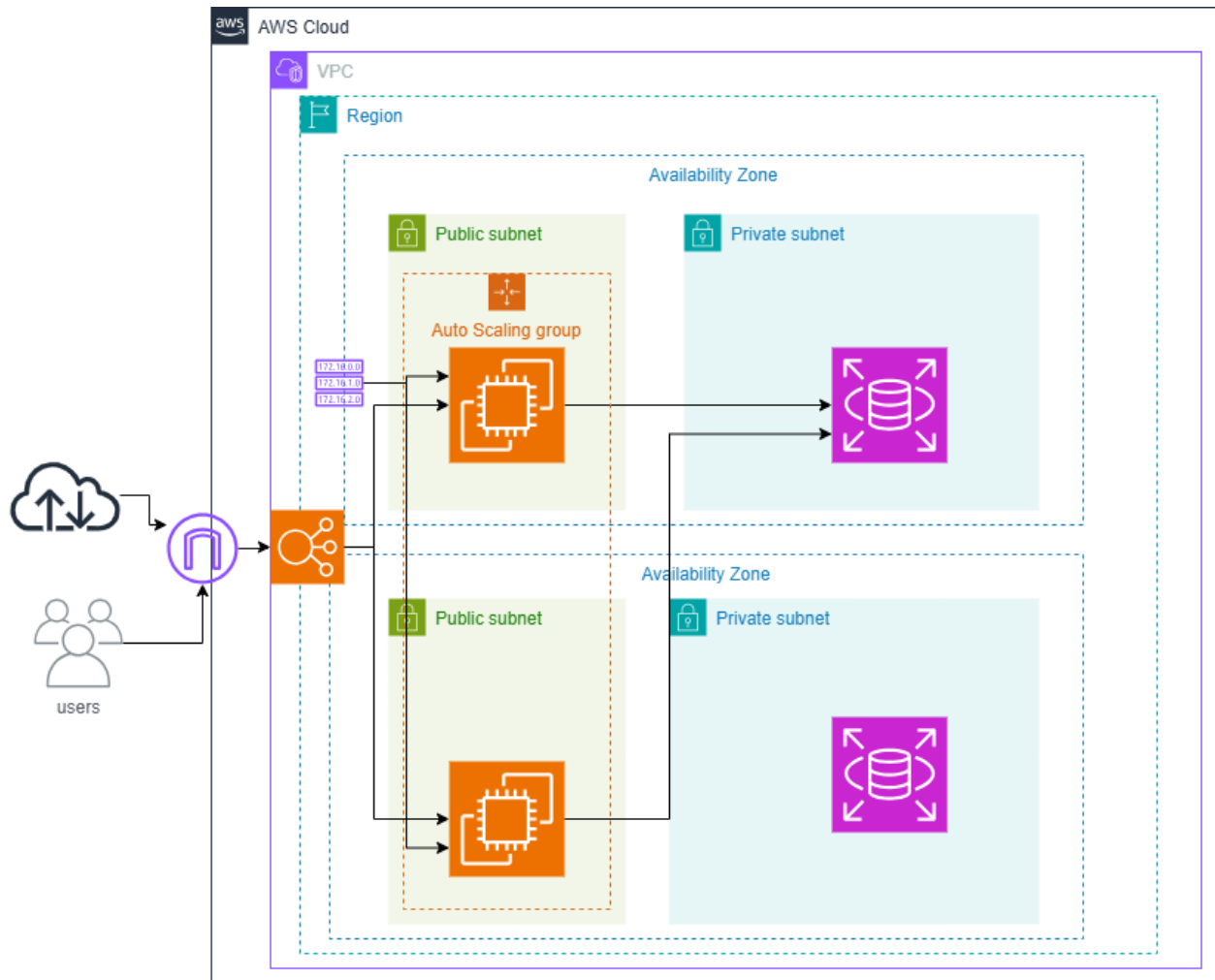


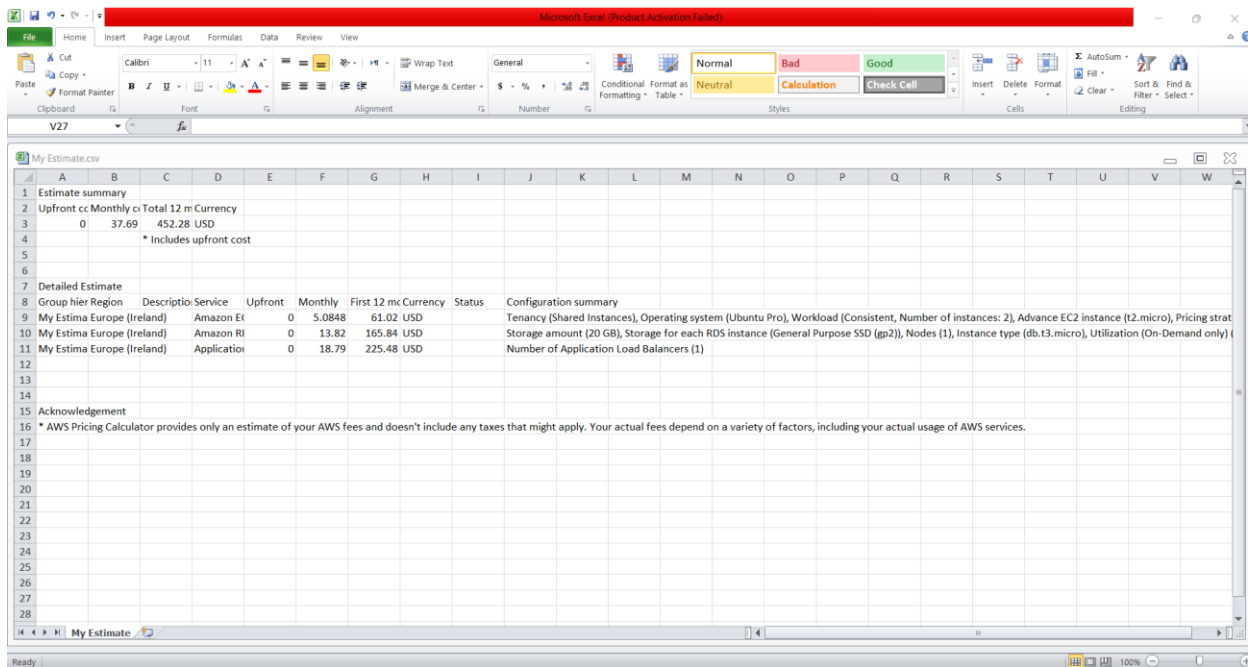
Report

Phase 1: Planning the design and estimating cost

Task 1: Creating an architectural diagram



Task 2: Developing a cost estimate

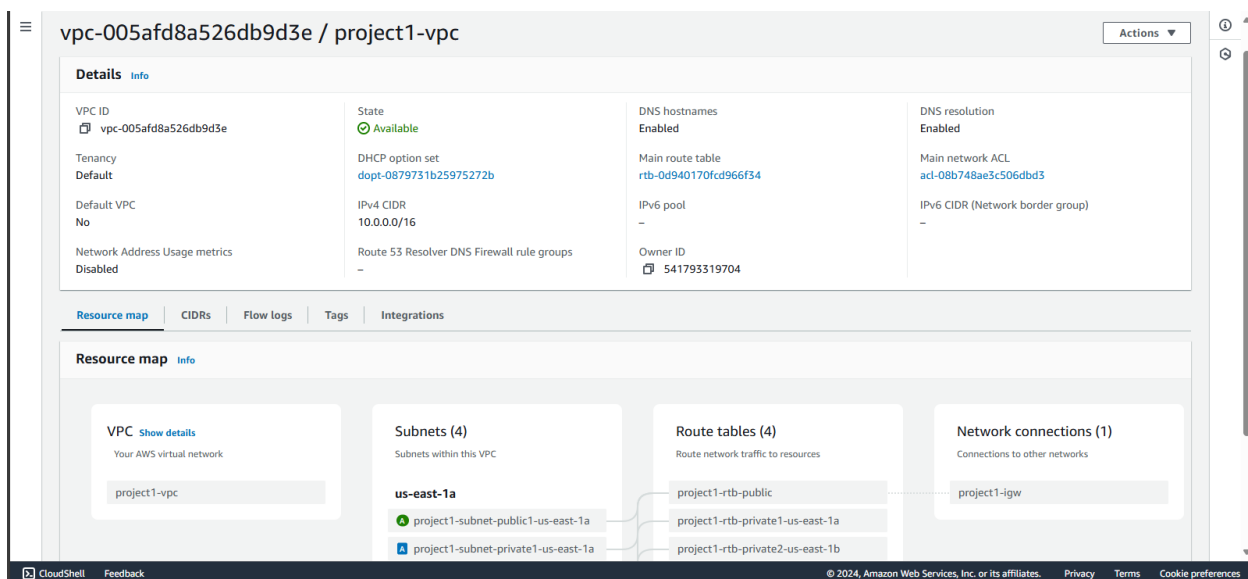


The screenshot shows a Microsoft Excel spreadsheet titled "My Estimate.csv". The spreadsheet is divided into several sections: "Estimate summary", "Detailed Estimate", and "Acknowledgement". The "Estimate summary" section includes a table with columns for "Group", "Region", "Description", "Service", "Upfront", "Monthly", "First 12 mo", "Currency", and "Status". The "Detailed Estimate" section includes a table with columns for "Group", "Region", "Description", "Service", "Upfront", "Monthly", "First 12 mo", "Currency", and "Status". The "Acknowledgement" section includes a text box with the following text: "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."

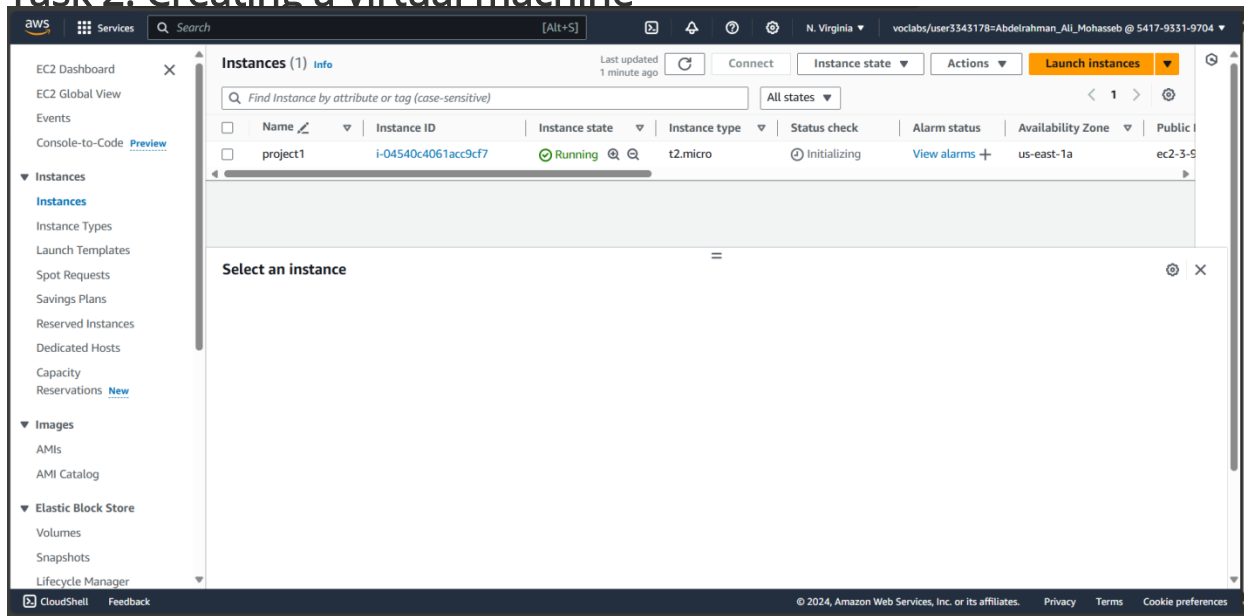
Group	Region	Description	Service	Upfront	Monthly	First 12 mo	Currency	Status
My Estima Europe (Ireland)	Amazon El	0	5.0848	61.02	USD	Tenancy (Shared Instances), Operating system (Ubuntu Pro), Workload (Consistent, Number of instances: 2), Advance EC2 instance (t2.micro), Pricing strat		
My Estima Europe (Ireland)	Amazon RI	0	13.82	165.84	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)		
My Estima Europe (Ireland)	Application	0	18.79	225.48	USD	Number of Application Load Balancers (1)		

Phase 2: Creating a basic functional web application

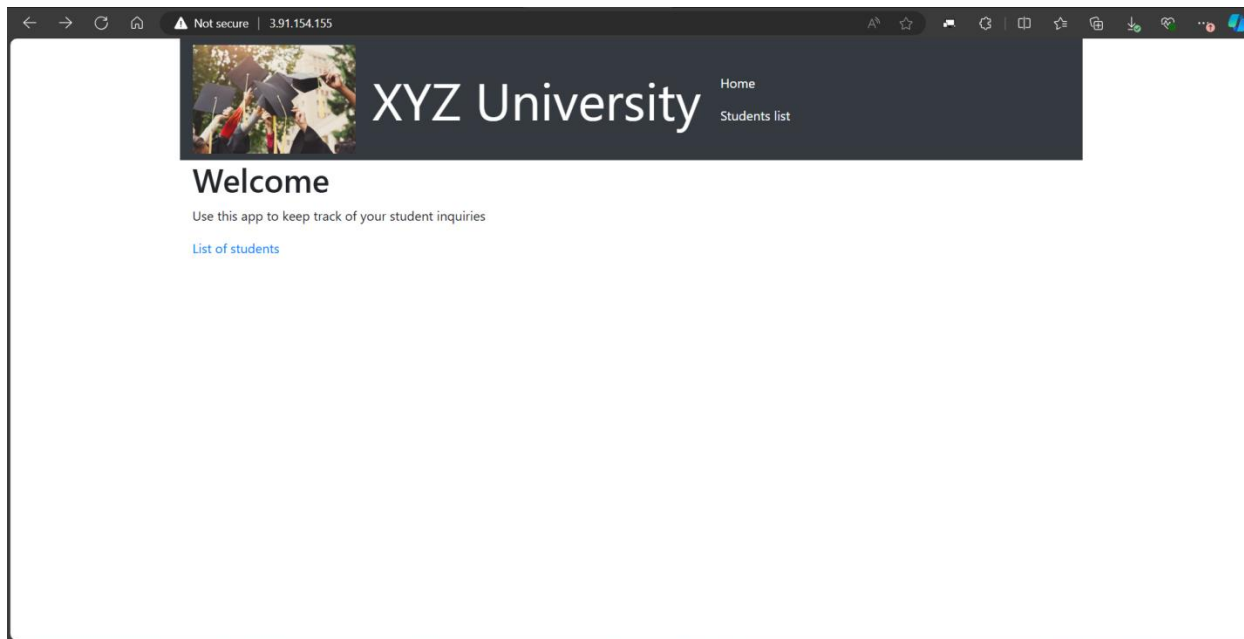
Task 1: Creating a virtual network



Task 2: Creating a virtual machine

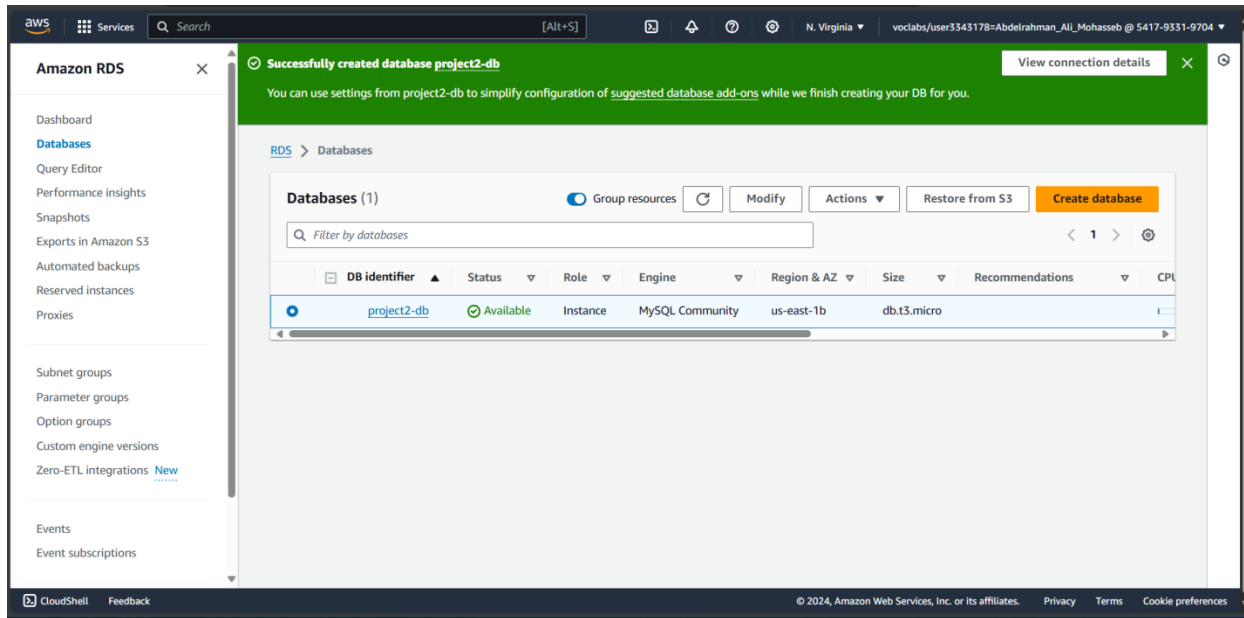


Task 3: Testing the deployment

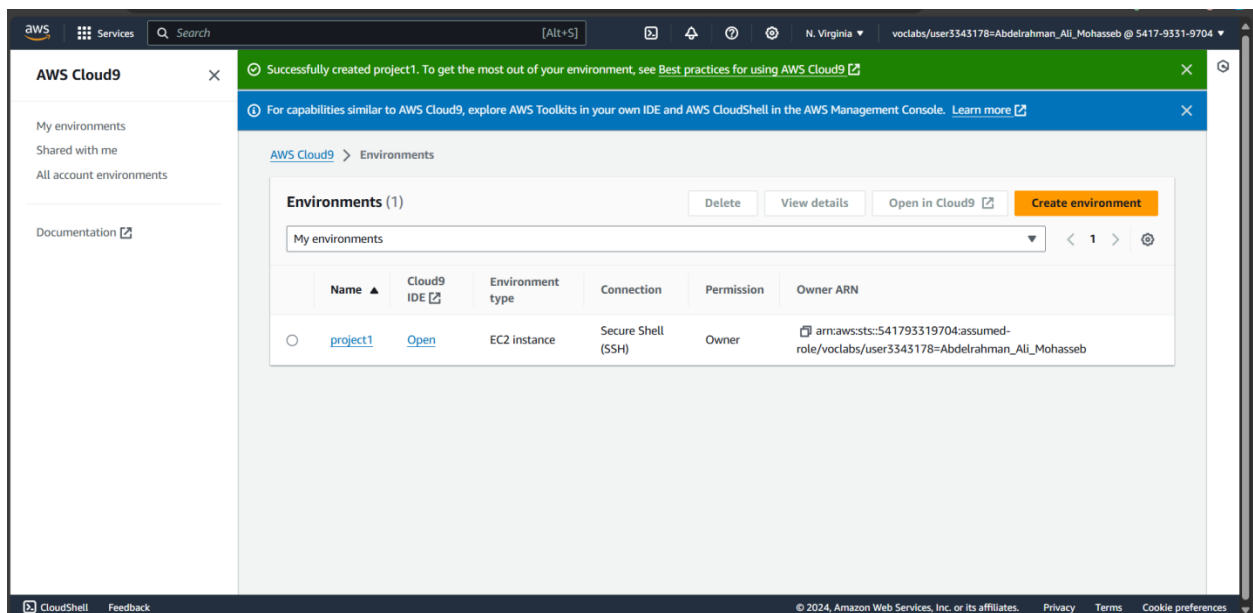
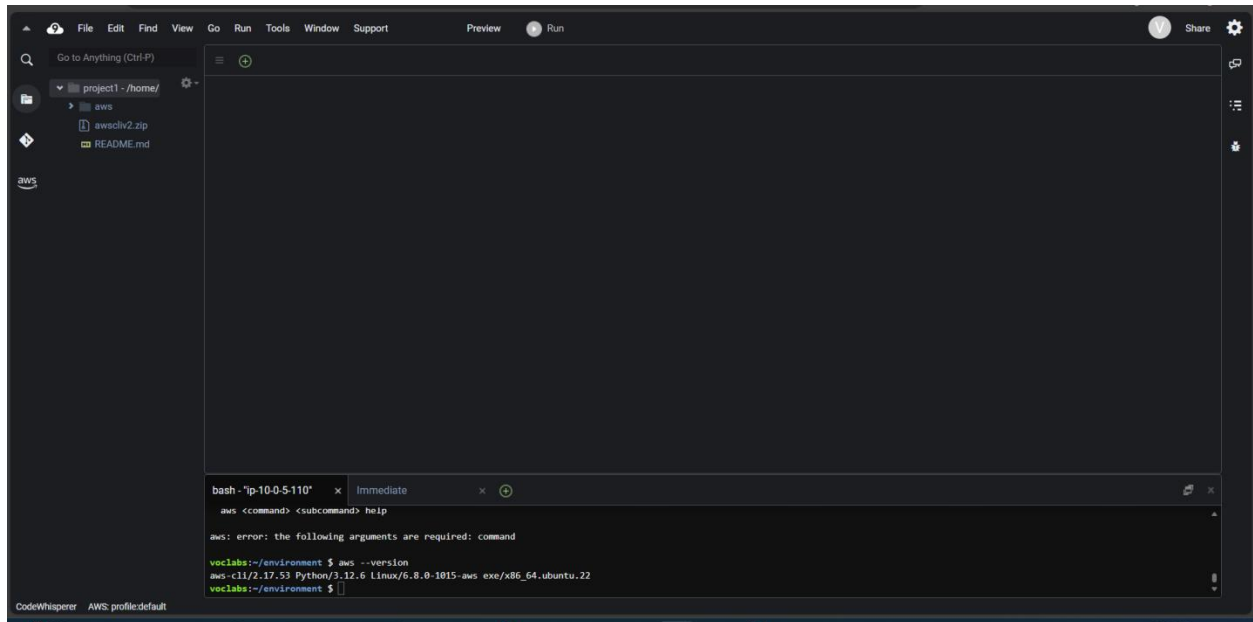


Phase 3: Decoupling the application components

Task 2: Creating and configuring the Amazon RDS database



Task 3: Configuring the development environment



Task 4: Provisioning Secrets Manager

The screenshot shows the AWS Secrets Manager console. The top navigation bar includes the AWS logo, a search bar, and the user's profile. The main content area has tabs for Overview, Rotation, Versions, Replication, and Tags. The 'Overview' tab is selected, showing the 'Secret value' page. A green notification bubble indicates that the secret value for the key 'dbInstanceIdentifier' has been copied. The table below lists the secret keys and their corresponding values.

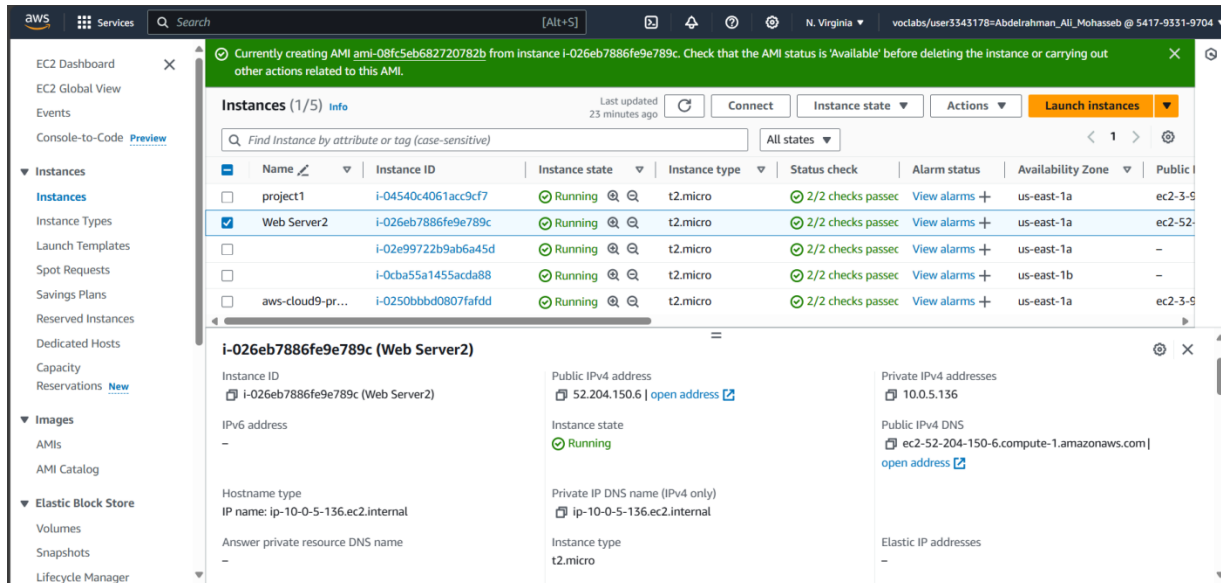
Secret key	Secret value
username	admin
password	h->p-en-h]GND]gu90h6e17]AxxQ
engine	mysql
host	cekovrbws5rm.us-east-1.rds.amazonaws.com
port	
dbInstanceIdentifier	project2-db

Below the table, there is a section for 'Resource permissions - optional' with an 'Edit permissions' button.

The screenshot shows the AWS Secrets Manager console. A green notification banner at the top states: 'You successfully stored the secret Mydbsecret. To show it in the list, choose Refresh. Use the sample code to update your applications to retrieve this secret.' The main content area has a breadcrumb trail 'AWS Secrets Manager > Secrets'. There is a search bar and a 'Store a new secret' button. Below these, a table lists the secrets.

Secret name	Description	Last retrieved (UTC)
Mydbsecret	-	September 23, 2024
rds/db-0ad5e5b2-b607-44d8-a46b-b5b3a5e25122	The secret associated with the primary RDS DB instance: arn:aws:rds:us-east-1:541793319704:db:project2-db	September 23, 2024

Task 5: Provisioning a new instance for the web server



Currently creating AMI ami-08fc5eb682720782b from instance i-026eb7886fe9e789c. Check that the AMI status is 'Available' before deleting the instance or carrying out other actions related to this AMI.

Instances (1/5) info

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
project1	i-04540c4061acc9cf7	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-5
Web Server2	i-026eb7886fe9e789c	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-52-
	i-02e99722b9ab6a45d	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-
	i-0c8a55a1455acda88	Running	t2.micro	2/2 checks passed	View alarms	us-east-1b	-
aws-cloud9-pr...	i-0250bbd0807fafdd	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-3-5

i-026eb7886fe9e789c (Web Server2)

Instance ID: i-026eb7886fe9e789c (Web Server2)

Public IPv4 address: 52.204.150.6 | [open address](#)

Private IPv4 addresses: 10.0.5.136

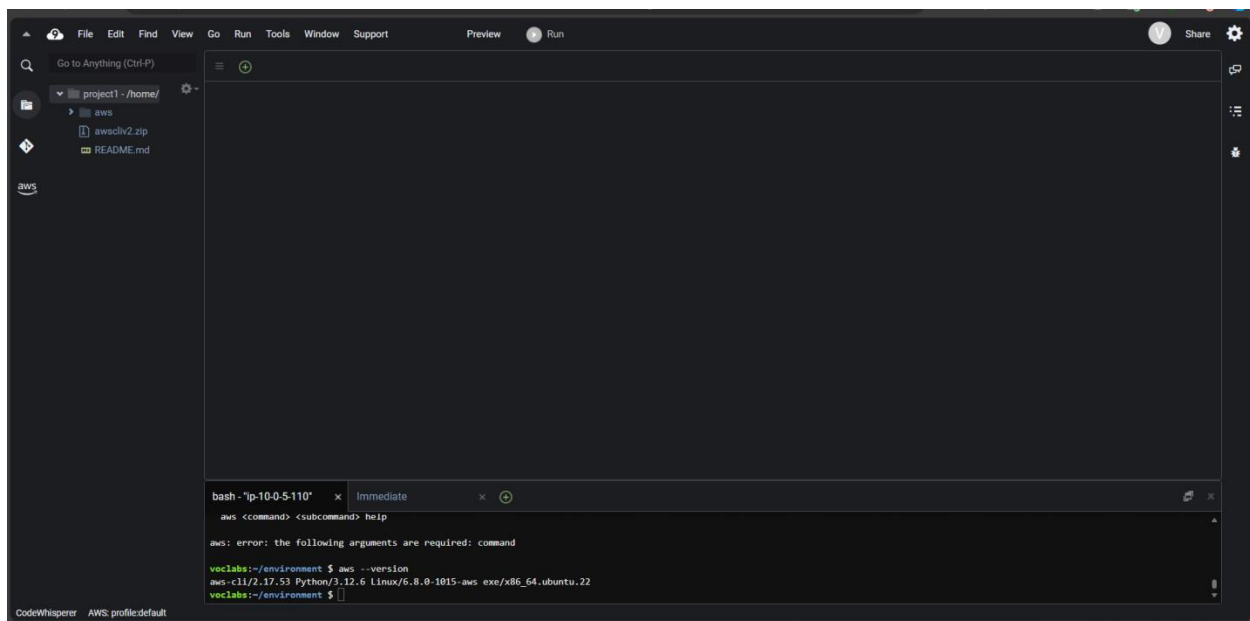
Instance state: Running

Public IPv4 DNS: ec2-52-204-150-6.compute-1.amazonaws.com | [open address](#)

Private IP DNS name (IPv4 only): ip-10-0-5-136.ec2.internal

Instance type: t2.micro

Task 6: Migrating the database



bash - "ip-10-0-5-110" x Immediate x

aws <command> <subcommand> help

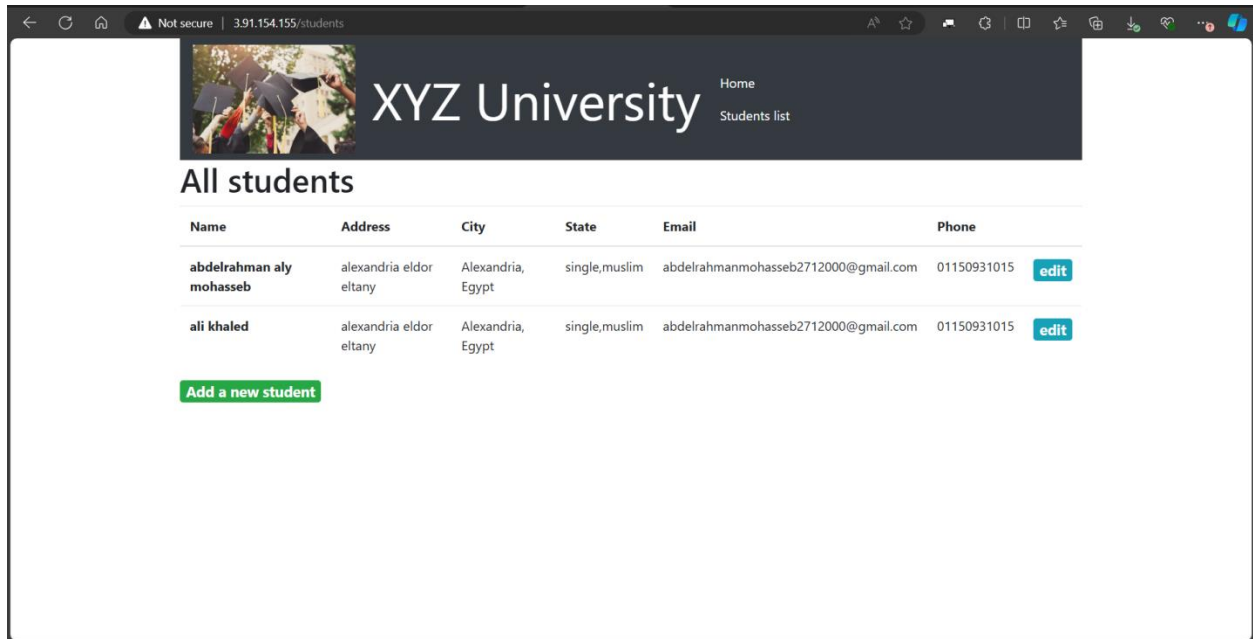
aws: error: the following arguments are required: command

voclabs:~/environment \$ aws --version

aws-cli/2.17.53 Python/3.12.6 Linux/x86_64-ubuntu.22

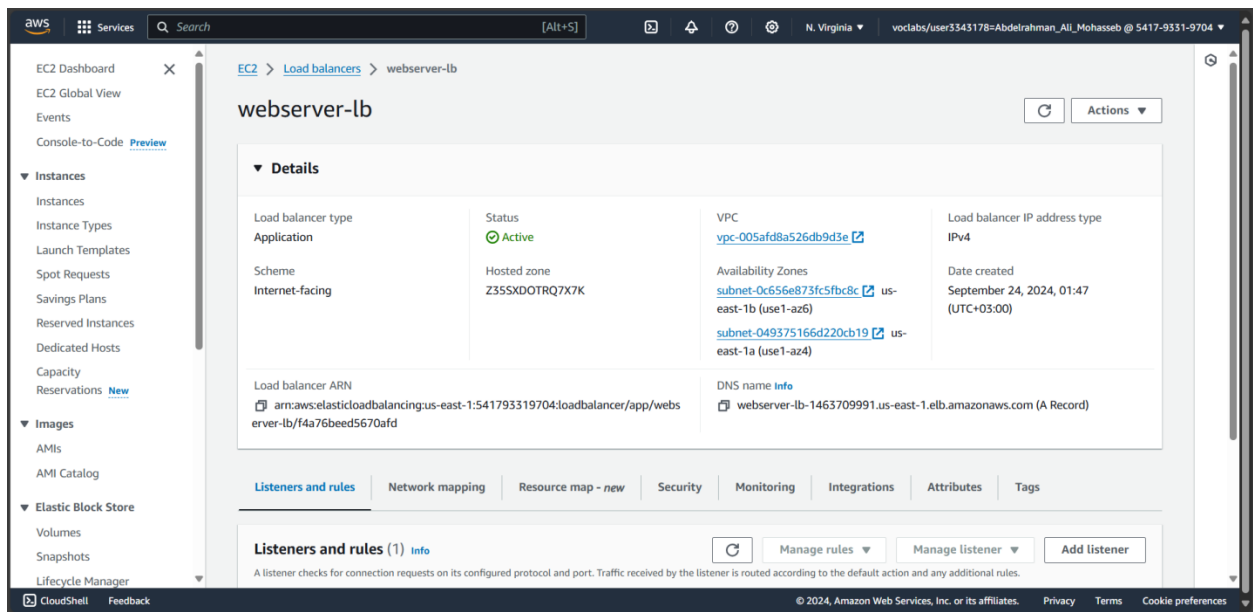
voclabs:~/environment \$

Task 7: Testing the application

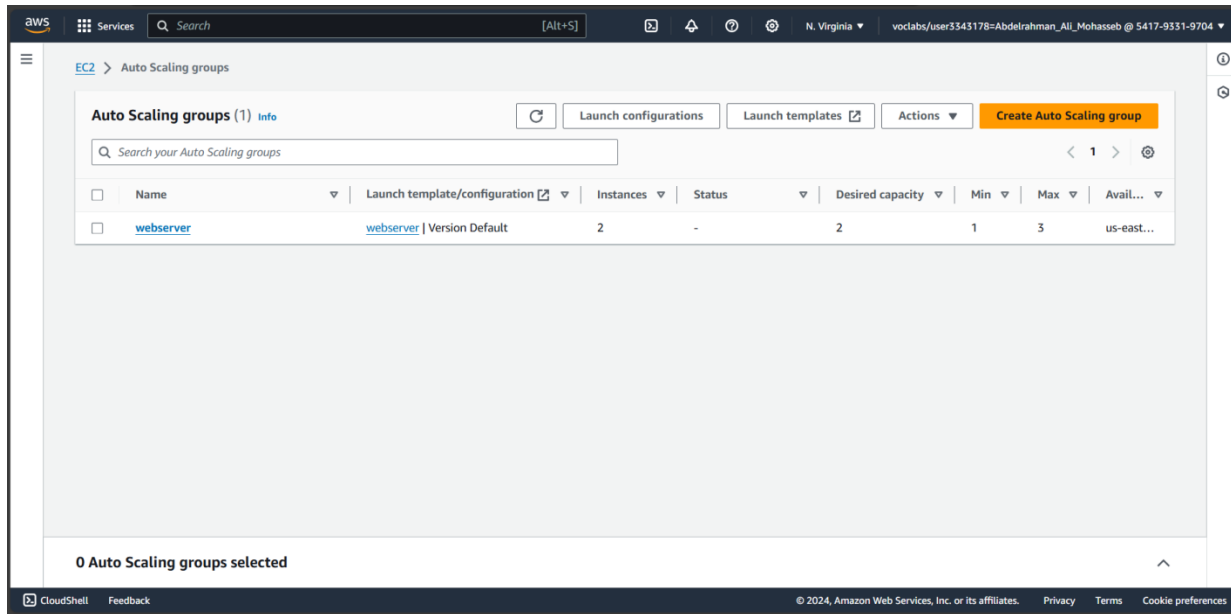


Phase 4: Implementing high availability and scalability

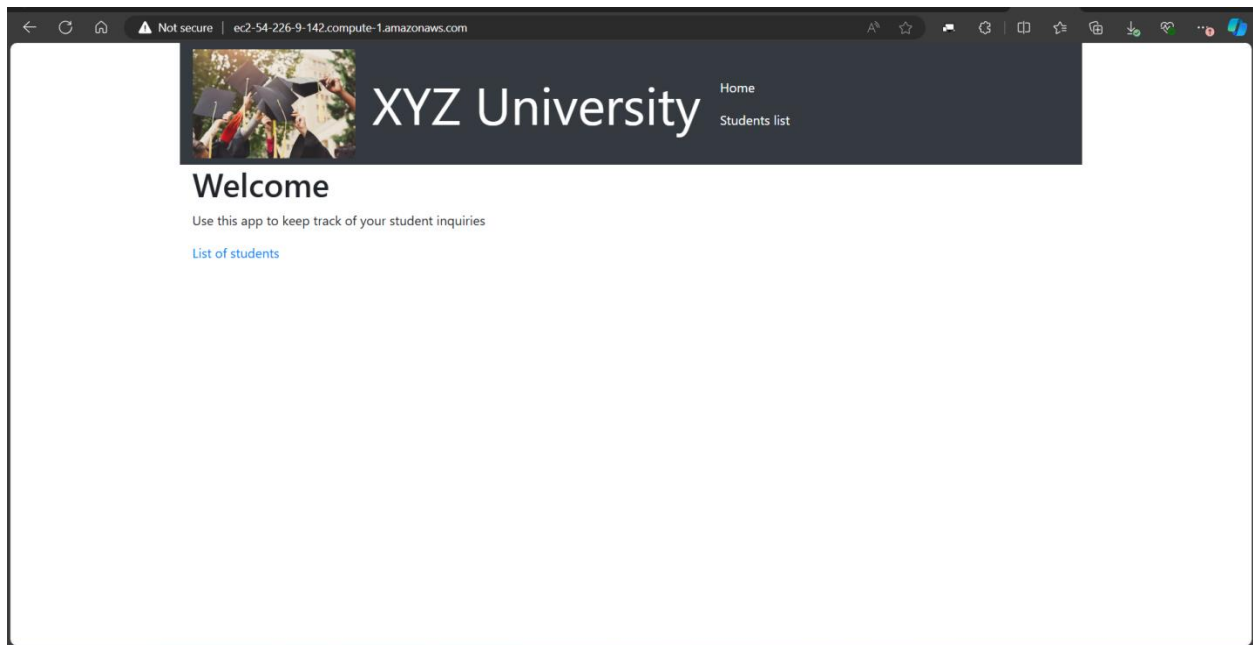
Task 1: Creating an Application Load Balancer



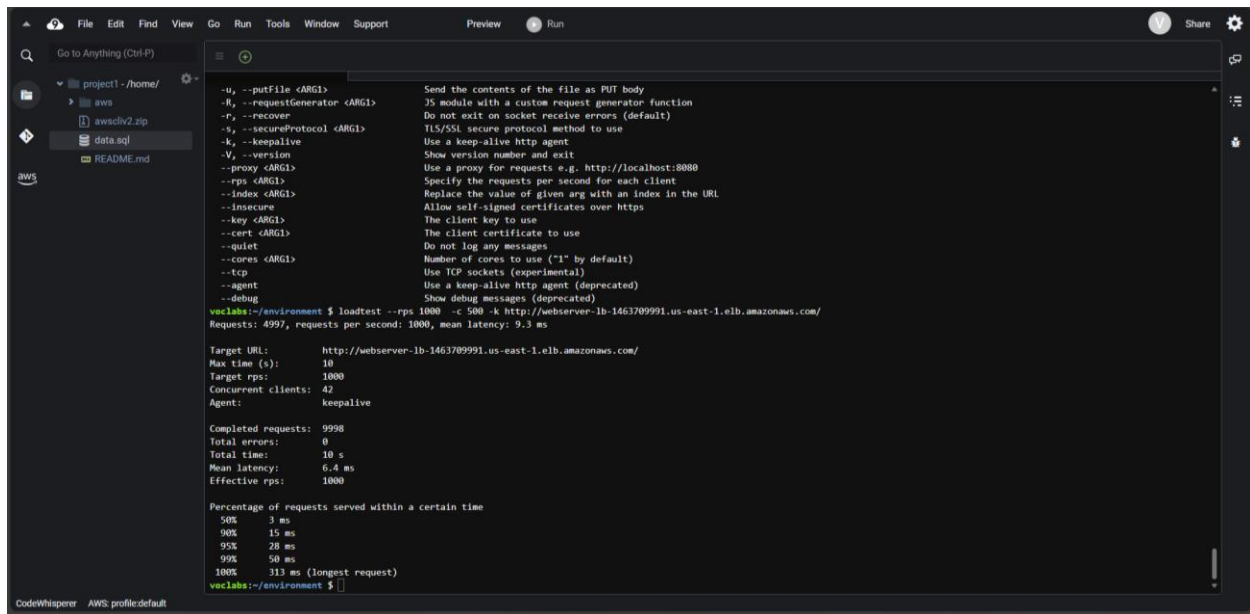
Task 2: Implementing Amazon EC2 Auto Scaling



Task 3: Accessing the application



Task 4: Load testing the application



```
-u, --putFile <ARG1>          Send the contents of the file as PUT body
-R, --requestGenerator <ARG1> JS module with a custom request generator function
-r, --recover                  Do not exit on socket receive errors (default)
-s, --secureProtocol <ARG1>   TLS/SSL secure protocol method to use
-k, --keepalive                Use a keep-alive http agent
-V, --version                  Show version number and exit
--proxy <ARG1>                Use a proxy for requests e.g. http://localhost:8080
--rps <ARG1>                  Specify the requests per second for each client
--index <ARG1>                Replace the value of given arg with an index in the URL
--insecure                    Allow self-signed certificates over https
--key <ARG1>                  The client key to use
--cert <ARG1>                 The client certificate to use
--quiet                       Do not log any messages
--cores <ARG1>                Number of cores to use ("1" by default)
--tcp                         Use TCP sockets (experimental)
--agent                        Use a keep-alive http agent (deprecated)
--debug                       Show debug messages (deprecated)

voclabs:~/environment $ wrk -rps 1000 -c 500 -k http://webserver-1b-1463709991.us-east-1.elb.amazonaws.com/
Requests: 4997, requests per second: 1000, mean latency: 9.3 ms

Target URL:      http://webserver-1b-1463709991.us-east-1.elb.amazonaws.com/
Max time (s):    10
Target rps:      1000
Concurrent clients: 42
Agent:           keepalive

Completed requests: 9998
Total errors:      0
Total time:        10 s
Mean latency:      6.4 ms
Effective rps:     1000

Percentage of requests served within a certain time
50%    3 ms
90%    15 ms
95%    28 ms
99%    50 ms
100%   313 ms (longest request)

voclabs:~/environment $
```

Badges and Completion Certificates - Cloud Web Application Builder

