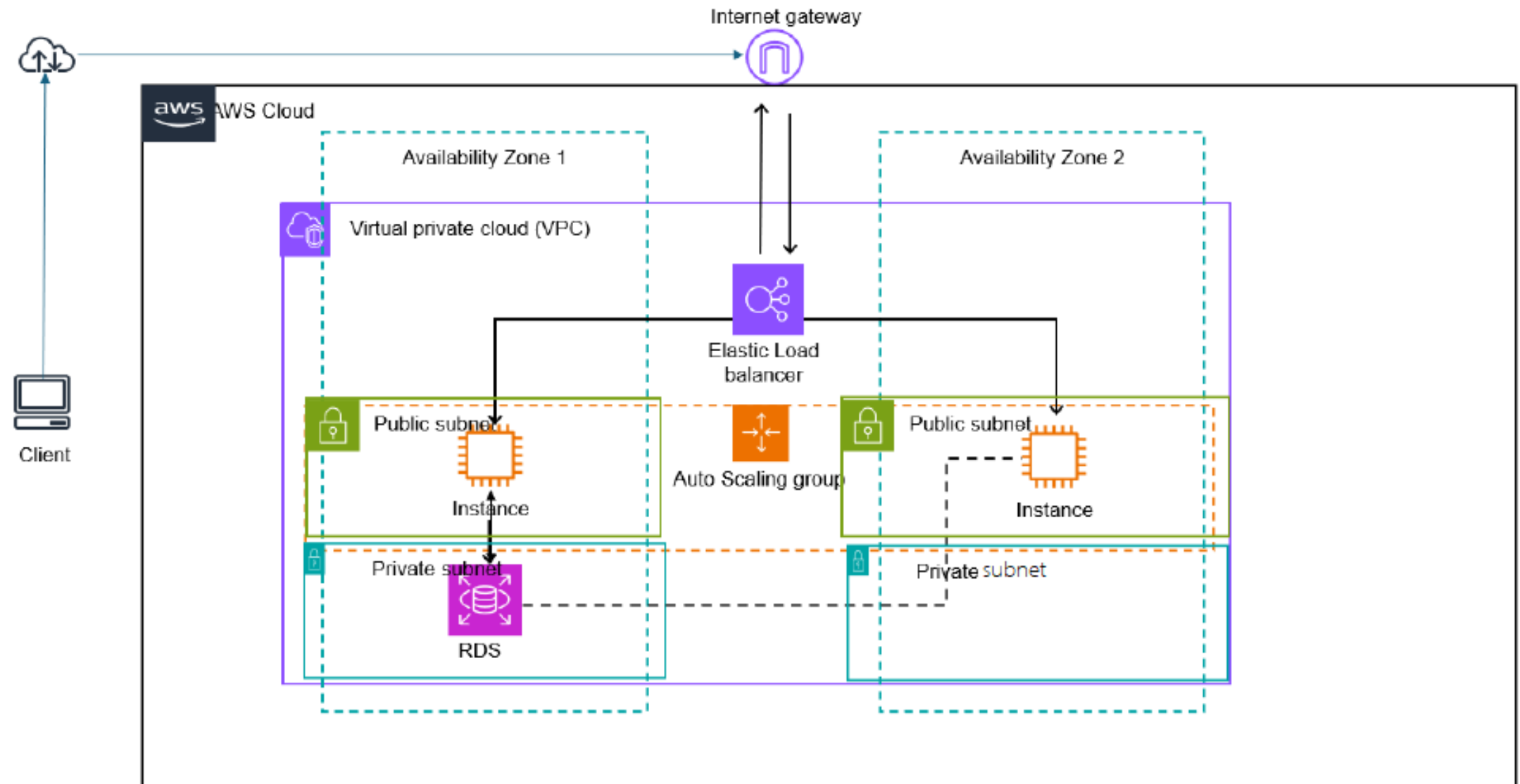


Lab Project - Cloud Web Application Builder

Adhm Ahmed Ali Alanssary

Task 1: Solution Diagram

Phase 1: Design and cost estimation



Task 2: Cost estimate

Use the AWS pricing calculator to estimate for 3 services: EC2, RDS, and Elastic Load Balancer

Estimate summary [Info](#)

Upfront cost

0.00 USD

Monthly cost

103.01 USD

Total 12 months cost

1,236.12 USD

Includes upfront cost

My Estimate

Duplicate

Delete

Find resources

<input type="checkbox"/>	Service Name ▾	Status ▾	Upfront cost ▾	Monthly cost ▾	Description ▾	Region ▾
<input type="checkbox"/>	Amazon RDS fo... <div><div></div></div>	-	0.00 USD	51.32 USD	-	US East (N. Virg...
<input type="checkbox"/>	Elastic Load Bal... <div><div></div></div>	-	0.00 USD	22.27 USD	-	US East (N. Virg...
<input type="checkbox"/>	Amazon EC2 <div><div></div></div>	-	0.00 USD	29.42 USD	-	US East (N. Virg...

Phase 2: Create a basic web app

Task 1: Create VPC

Set the IPv4 CIDR address for the VPC to 10.0.0.0/16 and create 2 Zones containing public and private subnets

VPC settings

Resources to create [Info](#)

Create only the VPC resource or the VPC and other networking resources.

☐ VPC only

☒ VPC and more

Name tag auto-generation [Info](#)

Enter a value for the Name tag. This value will be used to auto-generate Name tags for all resources in the VPC.

☒ Auto-generate

my-vpc

IPv4 CIDR block [Info](#)

Determine the starting IP and the size of your VPC using CIDR notation.

10.0.0.0/16

65,536 IPs

CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)

VPC [Show details](#)

Your AWS virtual network

my-vpc-vpc

Subnets (4)

Subnets within this VPC

us-east-1a

my-vpc-subnet-public1-us-east-1a

my-vpc-subnet-private1-us-east-1a

us-east-1b

my-vpc-subnet-public2-us-east-1b

my-vpc-subnet-private2-us-east-1b

Create 1 security group with inbound rules

Added a rule that accepts all IPs accessed via the HTTP protocol

Inbound rules

Info

Type	Protocol	Port range	Source
	Info		Info
<div>HTTP</div>	<div>TCP</div>	<div>80</div>	<div>An...</div>
			<div>0.0.0.0/0</div>

Add rule

Task 2: Create a virtual machine using EC2

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

Select an AMI

Setting up the operating system for the instance


▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

🔍 Search our full catalog including 1000s of application and OS images

Quick Start

Amazon
Linux



macOS


Ubuntu


Windows


Red Hat


⋮


[Browse more AMIs](#)
Including AMIs from
AWS, Marketplace and
the Community

Network Setup

Select the VPC in the first task you created and select the public subnet to deploy the instance.

Enable auto-assign public IP for automatic
create 1 public IPv4 address

Create a new or use security group built-in

▼ Network settings

Info

VPC - required

Info

vpc-055b007c11d0b9013 (my-vpc-vpc)

10.0.0.0/16

▼

↺

Subnet

Info

subnet-0a584b75cf90b6aa1

my-vpc-subnet-public1-us-east-1a

VPC: vpc-055b007c11d0b9013 Owner: 049948569672

Availability Zone: us-east-1a IP addresses available: 250 CIDR: 10.0.1.0/24

▼

↺ Create new subnet

Auto-assign public IP

Info

Enable

▼

Firewall (security groups)

Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group

☐ Select existing security group

Security group name - required

EC2-SG 1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and . -:/()#.,@!+=&:{}!\$*

Setting up a security group for an instance

▼ Security group rule 2 (TCP, 80, 0.0.0.0/0)

Remove

Type [Info](#)

HTTP ▼

Protocol [Info](#)

TCP

Port range [Info](#)

80

Source type [Info](#)

Anywhere ▼

Source [Info](#)

🔍 Add CIDR, prefix list or security

0.0.0.0/0 ✕

Description - optional [Info](#)

e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 3306, 0.0.0.0/16)

Remove

Type [Info](#)

MYSQL/Aurora ▼

Protocol [Info](#)

TCP

Port range [Info](#)

3306

Source type [Info](#)

Custom ▼

Source [Info](#)

🔍 Add CIDR, prefix list or security

0.0.0.0/16 ✕


Description - optional [Info](#)

e.g. SSH for admin desktop

Setting up user data for an instance

User data - *optional* [Info](#)


Upload a file with your user data or enter it in the field.

 Choose file

```
#!/bin/bash -xe
apt update -y
apt install nodejs unzip wget npm mysql-server -y
#wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-
200-ACCAP1-1-DEV/code.zip -P /home/ubuntu
wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-
200-ACCAP1-1-79581/1-lab-capstone-project-1/code.zip -P
/home/ubuntu
cd /home/ubuntu
unzip code.zip -x "resources/codebase_partner/node_modules/*"
cd resources/codebase_partner
npm install aws aws-sdk
mysql -u root -e "CREATE USER 'nodeapp' IDENTIFIED WITH
mysql_native_password BY 'student12'";
mysql -u root -e "GRANT all privileges on *.* to 'nodeapp'@'%';"
```

Task 3: Testing the deployment

After creating the instance, access the public IPv4 address of the instance to open the website below, do some more students to have data illustrating the transition to a new database using RDS



XYZ University

- Home
- Students list

All students

Name	Address	City	State	Email	Phone	
adhm	Egypt	Alexandria	Alexandria	adhmalanssary@gmail.com	01008107524	edit
dsda	asdds	asd	dsad	dasda@gmail.com	54516541	edit

[Add a new student](#)

Phase 3: Decoupling the application components

In this phase, the database will be separated and deployed separately on RDS.

Separating databases and deploying them separately will be easier to manage, using subnet groups to deploy across multiple AZs will ensure availability and security.

Task 1: Configure VPC

Create 2 private subnets in 2 different AZ zones (as done in phase 2)

Task 2: Create and configure an amazon RDS database

Create a subnet
group with the VPC
my-vpc

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

Name

You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

VPC

Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Create a subnet group of 2 private subnets of the selected VPC

Add subnets

Availability Zones

Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone ▼

us-east-1a ✕ us-east-1b ✕


Subnets

Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets ▼

subnet-0a649639772a4e793 (10.0.4.0/24) ✕

subnet-065ee9d312bc3c48e (10.0.3.0/24) ✕

 For Multi-AZ DB clusters, you must select 3 subnets in 3 different Availability Zones.

Create a database

Choosing an engine for the database

Engine options

Engine type [Info](#)

☐ Aurora (MySQL Compatible)



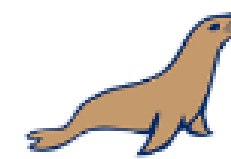
☐ Aurora (PostgreSQL Compatible)



☒ MySQL



☐ MariaDB



☐ PostgreSQL

☐ Oracle

Create DB instance information

Set Username and Password for DB

Password can use a password created by Secret Manager or created by yourself

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

STUDENTS

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

nodeapp

1 to 16 alphanumeric characters. The first character must be a letter.

☐ **Manage master credentials in AWS Secrets Manager**
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

☐ **Auto generate a password**
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

.....


Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Instance Setup and Storage

use the DB instance class db.t3.micro and use the SSD storage type with 20GiB provisioned.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

 **Amazon RDS Optimized Writes** - new [Info](#)
☐ Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)

- ☐ Standard classes (includes m classes)
- ☐ Memory optimized classes (includes r and x classes)
- ☒ Burstable classes (includes t classes)

db.t3.micro

2 vCPUs 1 GiB RAM Network: 2,085 Mbps

db.t3.micro

☐ Include previous generation classes

Storage



Storage type [Info](#)

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)

20 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

 After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
[Learn more](#) 

► Storage autoscaling

Task 3 : Configure the development environment

Prepare your development environment by configuring AWS Cloud9 with the t3.small virtual machine to execute AWS CLI commands. Use Secure Shell (SSH) connections to securely interact with your development environment.

Details

Name

name

Limit of 60 characters, alphanumeric, and unique per user.

Description - optional

hahaha

Limit 200 characters.

Environment type [Info](#)

Determines what the Cloud9 IDE will run on.

☒ New EC2 instance

Cloud9 creates an EC2 instance in your account. The configuration of your EC2 instance cannot be changed by Cloud9 after creation.

☐ Existing compute

You have an existing instance or server that you'd like to use.

New EC2 instance

Choose a configuration for Cloud9

New EC2 instance

Instance type [Info](#)

The memory and CPU of the EC2 instance that will be created for Cloud9 to run on.

☐ **t2.micro** (1 GiB RAM + 1 vCPU)
Free-tier eligible. Ideal for educational users and exploration.

☒ **t3.small** (2 GiB RAM + 2 vCPU)
Recommended for small web projects.

☐ **m5.large** (8 GiB RAM + 2 vCPU)
Recommended for production and most general-purpose development.

☐ **Additional instance types**
Explore additional instances to fit your need.

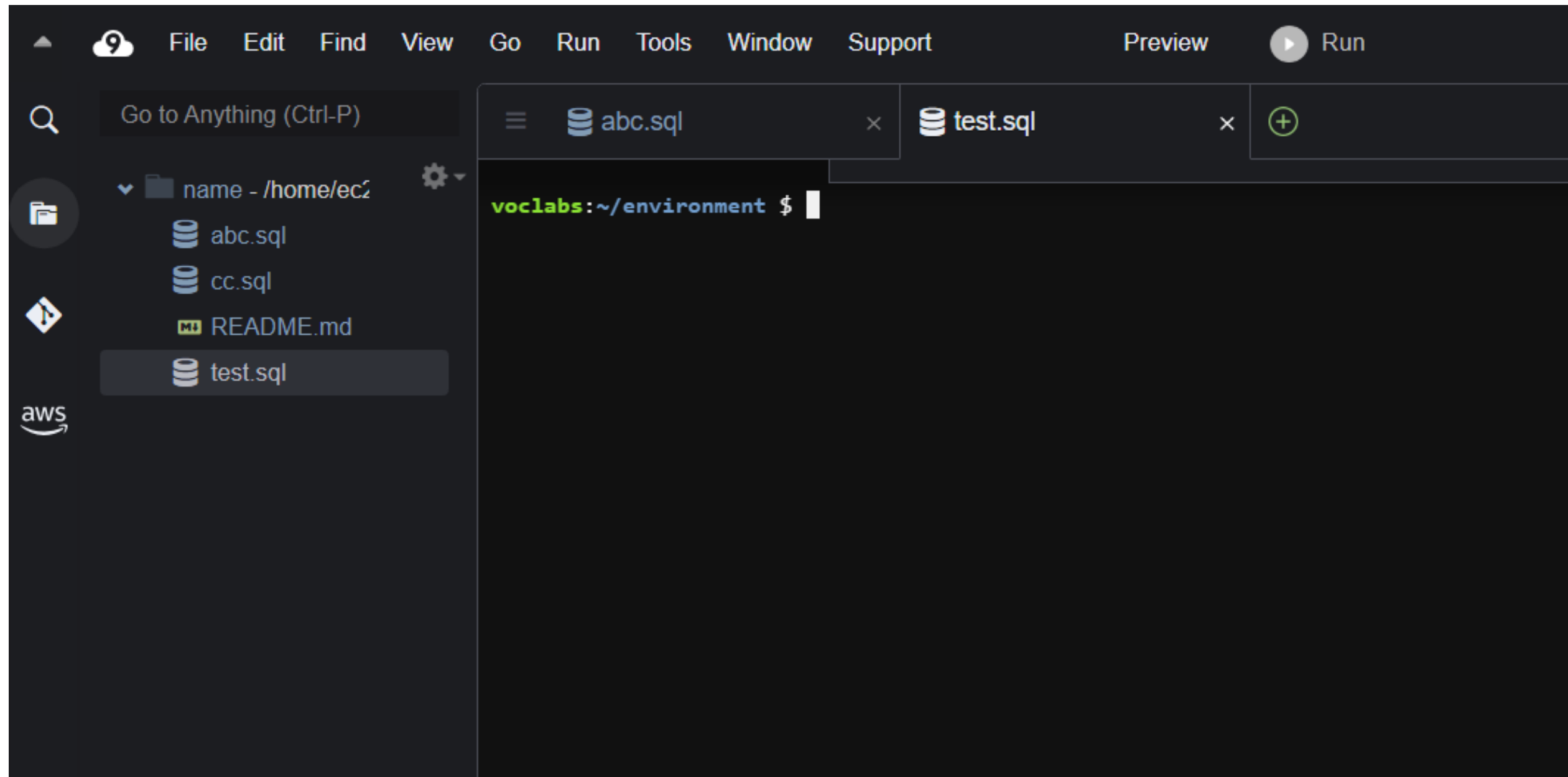
Platform [Info](#)

This will be installed on your EC2 instance. We recommend Amazon Linux 2023.

Amazon Linux 2



Once created, cloud9 will be able to access Ubuntu's CLI interface



Task 4: Provisioning Secrets Manager

Use AWS Secrets Manager to create a database repository of secret credentials.

Use Script-1 from a YAML file in AWS Cloud9 to perform secret creation and related tasks.

```
aws - "ip-10-0-1-14.ec2.in × (+)  
voclabs:~/environment $ aws secretsmanager create-secret \  
> --name Mydbsecret \  
> --description "Database secret for web app" \  
> --secret-string "{\"user\":\"nodeapp\",\"password\":\"12345678\",\"host\":\"<RDS Endpoint>\",\"db\":\"STUDENTS\"}"
```

Task 5: Create 1 more EC2 instance for the new web server

A new instance created in a different public subnet will use different user data than the previously created instance, for AWS IAM uses a role called LabRole to be able to access security information securely.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

[Add additional tags](#)

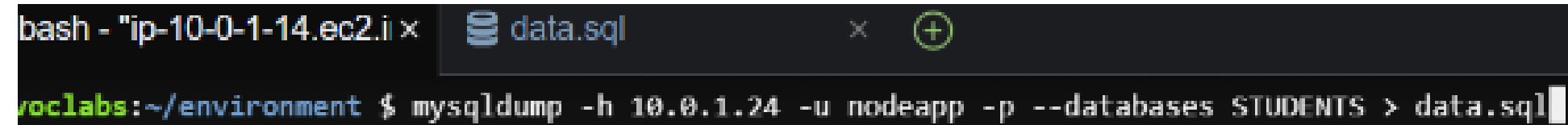
▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Task 6: Migrating the database

Step 1: Export database

Open cloud9 and run the command below to export 1 file data.sql, using the available password student12

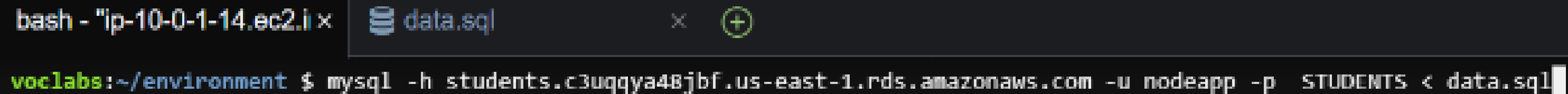


The screenshot shows a cloud9 IDE interface. At the top, there is a tab labeled 'data.sql' with a file icon, a close button (X), and a plus button (+). Below the tab, the terminal window shows a bash prompt with the command: `mysql -h 10.0.1.24 -u nodeapp -p --databases STUDENTS > data.sql`. The prompt is `voclabs:~/environment $`. The command is currently being entered, with a cursor at the end of the line.

```
bash - "ip-10-0-1-14.ec2.i x data.sql × +  
voclabs:~/environment $ mysql -h 10.0.1.24 -u nodeapp -p --databases STUDENTS > data.sql
```

Step 2: Import data

Use the following command to enter data into a database created on RDS, using the initial self-generated password at the step of creating a database on RDS




The image shows a terminal window with a dark background. At the top, there is a tab labeled 'data.sql' with a close button (X) and a plus sign (+). Below the tab, the terminal prompt is 'voclabs:~/environment \$'. The command being entered is 'mysql -h students.c3uqqya48jbf.us-east-1.rds.amazonaws.com -u nodeapp -p STUDENTS < data.sql'. The cursor is at the end of the command.

```
bash - "ip-10-0-1-14.ec2.i x data.sql x (+)  
voclabs:~/environment $ mysql -h students.c3uqqya48jbf.us-east-1.rds.amazonaws.com -u nodeapp -p STUDENTS < data.sql
```


Task 7: Testing

Results of a web run deployed on a new EC2 instance with the database decoupled to RDS



XYZ University

- Home
- Students list

All students

Name	Address	City	State	Email	Phone	
adhm	Egypt	Alexandria	Alexandria	adhmalanssary@gmail.com	01008107524	edit
dsda	asddsa	asd	dsad	dasda@gmail.com	54516541	edit

[Add a new student](#)

Phase 4: Implementing high availability and scalability

Task 1: Create Application Load Balancer

A load balancer is used to evenly distribute traffic to servers or components of a system. To ensure high performance and availability of the system.

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule is applicable, it selects a target from the target group for the rule action.

► How Elastic Load Balancing works

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

ALB-EC2

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

The "Network Mapping" section of the Application Load Balancer (ALB) allows mapping traffic from the ALB to target groups in the VPC "my-vpc-vpc".

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets or you can [create a new VPC](#). Only VPCs with an internet gateway are enabled for selection. The load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

my-vpc-vpc

vpc-055b007c11d0b9013

IPv4: 10.0.0.0/16

Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not associated with the VPC or the VPC are not available for selection.

☐ **us-east-1a (use1-az1)**

☐ **us-east-1b (use1-az2)**

Select the pre-configured ALB-SG Security Group to manage and control traffic to and from the Load Balancer.

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

ALB-SG

sg-053125b64223d94b1 VPC: vpc-055b007c11d0b9013

Set up listeners and routing to check connection requests using ports and network connection methods here are http port 80 methods and use the previously created target group

Listeners and routing [Info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

▼ Listener HTTP:80

Remove

Protocol

Port

Default action

[Info](#)

HTTP ▼

:

80

1-65535

Forward to

TG-EC2

Target type: Instance, IPv4

HTTP ▼

↻

[Create target group](#) [↗](#)

Listener tags - *optional*

Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Overview Results After Setting Up Application Load Balancer

<div><div>Summary</div><div>Review and confirm your configurations. Estimate cost</div></div>			
<div><div>Basic configuration</div><div>ALB-EC2<ul style="list-style-type: none">Internet-facingIPv4</div></div>	<div><div>Security groups</div><ul style="list-style-type: none">ALB-SG<ul style="list-style-type: none"></div>	<div><div>Network mapping</div><div>VPC<ul style="list-style-type: none"></div><div>my-vpc-vpc<ul style="list-style-type: none">us-east-1a<ul style="list-style-type: none">us-east-1b<ul style="list-style-type: none"></div></div>	<div><div>Listeners and routing</div><ul style="list-style-type: none">HTTP:80 defaults to<ul style="list-style-type: none"></div>

Task 2: Deploy Amazon EC2 Auto Scaling

The Auto Scaling group (ASG) automatically manages and scales EC2 instances deployed with the EC2 template in phase 3. Automatically changing the number of instances based on the given settings helps ensure that the system always maintains stable performance and high availability.

ASG helps the web system operate better when there is a fluctuation in traffic to the website.

Create Template

Create a template that uses an AMI from a previously created EC2 instance, and once used, the template will be configured identically to the configuration from that EC2 instance

Create launch template

Creating a launch template allows you to create a saved instance configuration that can be launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - *required*

myTemplate

Must be unique to this account. Max 128 chars. No spaces or special characters.

Template version description

template

Max 255 chars

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Recents


My AMIs

Quick Start

☐ Don't include in launch template

☒ Owned by me

☐ Shared with me


[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Web AMI

ami-04de259c44cf9216c

2024-01-07T11:29:17.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Set up the created template for auto scaling group to create an instance

Name


Auto Scaling group name

Enter a name to identify the group.

Auto Scale

Must be unique to this account in the current Region and no more than 255 characters.


Launch template [Info](#)


 For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

myTemplate ▼



[Create a launch template](#) 

select VPCs and set up the AZs and subnets that the auto scaling group will deploy

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-055b007c11d0b9013 (my-vpc-vpc)
10.0.0.0/16

[Create a VPC](#)

Availability Zones and subnets

Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets

us-east-1a | subnet-065ee9d312bc3c48e (my-vpc-subnet-private1-us-east-1a)
10.0.3.0/24

us-east-1b | subnet-0a649639772a4e793 (my-vpc-subnet-private2-us-east-1b)
10.0.4.0/24

--

Load balancer selection

Load balancing [Info](#)

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

☐ No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.

☒ Attach to an existing load balancer

Choose from your existing load balancers.

☐ Attach to a new load balancer

Quickly create a basic load balancer to attach to your Auto Scaling group.

Assign ALB to auto scaling group

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

- ☒ Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

- ☐ Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups▼

↻

TG-EC2 | HTTP

Application Load Balancer: ALB-EC2

✕

Select group size and scaling with a min of 2 instances and a max of 6 instances depending on the number of requests and tasks to be processed

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to match demand with automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported if the instances are configured with a set of instance attributes.

Units (number of instances) ▼

Desired capacity

Specify your group size.

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

Max desired capacity



Results after creating an auto scaling group

Instances created from the Auto Scaling Group will be placed in the Target Group

Target group: TG-EC2

target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 5 healthy targets.



Filter targets

< 1 >

<input type="checkbox"/>	Instance ID ▾	Name ▾	Port ▾	Zone ▾	Health status ▾	Health status c
<input type="checkbox"/>	i-0ce65c05cdbe86cc2	Lab Instance	80	us-east-1a	✔ Healthy	-
<input type="checkbox"/>	i-04a5c288936df95f6	Lab Instance	80	us-east-1b	✔ Healthy	-

Results when creating an auto scaling group


Once created, access to the load balancer service will have 1 DNS name to access the web

<input checked="" type="checkbox"/>	Name	DNS name	State	VPC ID	Availability Zones
<input checked="" type="checkbox"/>	ALB-EC2	 ALB-EC2-2106406089.us-...	 Active	vpc-055b007c11d0b9...	<u>2 Availability Zones</u>


Load balancer: ALB-EC2

Internet-facing

Z35SXDOTRQ7X7K


[subnet-0a584b75cf90b6aa1](#)

us-east-1a (use1-az1)


[subnet-0876033cfaaa97f3e](#)

us-east-1b (use1-az2)

Load balancer ARN

 arn:aws:elasticloadbalancing:us-east-1:049948569672:loadbalancer/app/ALB-EC2/35e3c7800e3f83fd


DNS name [Info](#)

 ALB-EC2-2106406089.us-east-1.elb.amazonaws.com (A Record)

January 7, 2024, 22:10
(UTC+07:00)

Task 3: Go to the Web

Go to the app and perform some tasks to check it. For example, view, add, delete, and modify student records.



XYZ University

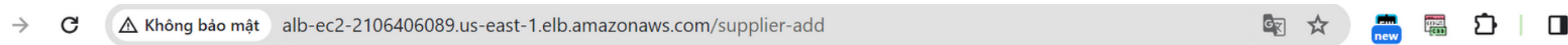
[Home](#)[Students list](#)

All students

Name	Address	City	State	Email	Phone	
adhm	Egypt	Alexandria	Alexandria	adhmalanssary@gmail.com	01008107524	edit
dsda	asdds	asd	dsad	dasda@gmail.com	54516541	edit

[Add a new student](#)

Testing additional functions



XYZ University

[Home](#)[Students list](#)

All fields are required

Name

Name of this student

Address

Address for this student

City

City for this student

Task 4: Load test the application

Download the load test package for testing using the following command "npm install -g loadtest" on the CLI interface of clould9

The image below performs a loadtest with 1000 requests per second and 500 connections while not checking the SSL certificate

```
voclabs:~/environment $ loadtest --rps 10000 -c 5000 -k http://ALB-EC2-2106406089.us-east-1.elb.amazonaws.com
Requests: 13534, requests per second: 2703, mean latency: 469.2 ms

Target URL:      http://ALB-EC2-2106406089.us-east-1.elb.amazonaws.com
Max time (s):    10
Target rps:      10000
Concurrent clients: 11413
Agent:           keepalive

Completed requests: 28664
Total errors:      0
Total time:        10.002 s
Mean latency:      820.7 ms
Effective rps:     2866

Percentage of requests served within a certain time
50%      558 ms
90%      986 ms
95%      2557 ms
99%      6973 ms
100%     7581 ms (longest request)
```