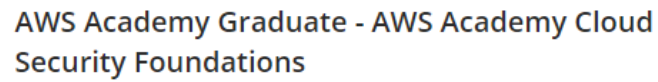


Congratulations on your most recent badge 🏆



Share



The diagram illustrates a multi-availability zone AWS architecture designed for high availability and disaster recovery. It features two Availability Zones, each containing a Private Subnet and a Public Subnet. Key components include:

- VPC:** A single Virtual Private Cloud spanning both Availability Zones.
- Internet Gateway:** Connects the VPC to the Internet.
- Route 53:** DNS service for domain resolution.
- CloudFront:** Content Delivery Network for static content.
- S3:** Amazon S3 bucket for static content.
- EC2:** Amazon EC2 instances (Application) in both Private Subnets, connected to the Internet via NAT Gateways in the Public Subnets.
- Amazon RDS:** Amazon Relational Database Service instance in the Private Subnet of the first Availability Zone, with a Read Replica in the Private Subnet of the second Availability Zone for disaster recovery.
- Amazon ElastiCache:** Amazon ElastiCache instance in the Private Subnet of the first Availability Zone.
- Amazon IAM:** Amazon Identity and Access Management for user authentication and authorization.
- Amazon CloudWatch:** Amazon CloudWatch for monitoring and logging.
- Amazon SNS:** Amazon Simple Notification Service for messaging.
- Amazon SSM:** Amazon Systems Manager for managing instances.
- Amazon EBS:** Amazon Elastic Block Store for persistent storage.
- Amazon EFS:** Amazon Elastic File System for shared file storage.
- Amazon EMR:** Amazon Elastic MapReduce for big data processing.
- Amazon Athena:** Amazon Athena for querying data in S3.
- Amazon Kinesis:** Amazon Kinesis for real-time data streaming.
- Amazon Lambda:** Amazon Lambda for serverless computing.
- Amazon Step Functions:** Amazon Step Functions for orchestrating workflows.
- Amazon Rekognition:** Amazon Rekognition for image and video analysis.
- Amazon Comprehend:** Amazon Comprehend for natural language processing.
- Amazon Forecast:** Amazon Forecast for time series forecasting.
- Amazon SageMaker:** Amazon SageMaker for machine learning.
- Amazon Bedrock:** Amazon Bedrock for generative AI.
- Amazon Lex:** Amazon Lex for building conversational interfaces.
- Amazon Polly:** Amazon Polly for text-to-speech.
- Amazon Transcribe:** Amazon Transcribe for speech-to-text.
- Amazon Textract:** Amazon Textract for document analysis.
- Amazon Rekognition:** Amazon Rekognition for image and video analysis.
- Amazon Comprehend:** Amazon Comprehend for natural language processing.
- Amazon Forecast:** Amazon Forecast for time series forecasting.
- Amazon SageMaker:** Amazon SageMaker for machine learning.
- Amazon Bedrock:** Amazon Bedrock for generative AI.
- Amazon Lex:** Amazon Lex for building conversational interfaces.
- Amazon Polly:** Amazon Polly for text-to-speech.
- Amazon Transcribe:** Amazon Transcribe for speech-to-text.
- Amazon Textract:** Amazon Textract for document analysis.

## Task 2: Developing a cost estimate

The screenshot shows the AWS Pricing Calculator interface for a 'My Estimate'. The top section displays the cost breakdown: Upfront cost is 0.00 USD, Monthly cost is 58.65 USD, and the Total 12 months cost is 703.56 USD (including upfront cost). The 'Getting Started with AWS' section offers buttons for 'Get started for free' and 'Contact Sales'. Below this is a table of services included in the estimate:

Service Name	Status	Upfront cost	Monthly cost	Description	Region	Config Summary
AWS Application Migration Se...	-	0.00 USD	0.00 USD	-	US East (Ohio)	Number of server/s (2), Numb...
Amazon EC2	-	0.00 USD	3.80 USD	-	US East (Ohio)	Tenancy (Shared Instances), O...
Elastic Load Balancing	-	0.00 USD	31.03 USD	-	US East (Ohio)	Number of Application Load B...
Amazon Aurora MySQL Comp...	-	0.00 USD	25.80 USD	-	US East (Ohio)	Aurora MySQL Cluster Config...

## Phase 2: Creating a basic functional web application

### Task 1: Creating a virtual network

The screenshot shows the AWS Management Console VPC dashboard for a VPC named 'project1-vpc'. The 'Details' section provides information about the VPC, including its ID, state (Available), DHCP option set, IPv4 CIDR, and DNS settings. The 'Resource map' section shows the VPC's internal structure, including subnets, route tables, and network connections.

**VPC Details:**

- VPC ID: vpc-0ebd274cfb648c0ad
- State: Available
- Tenancy: Default
- DHCP option set: dopt-0cc7f54533a2d796a
- Default VPC: No
- IPv4 CIDR: 10.0.0.0/16
- Network Address Usage metrics: Disabled
- DNS hostnames: Enabled
- Main route table: rtb-05e5d99a7111e9f1e
- IPv6 pool: -
- Owner ID: 411641690222
- DNS resolution: Enabled
- Main network ACL: acl-07c98bb06d50403d0
- IPv6 CIDR (Network border group): -

**Resource map:**

- VPC:** project1-vpc
- Subnets (4):**
  - us-east-1a
    - project1-subnet-public1-us-east-1a
    - project1-subnet-private1-us-east-1a
  - us-east-1b
    - project1-subnet-public2-us-east-1b
    - project1-subnet-private2-us-east-1b
- Route tables (4):**
  - project1-rtb-public
  - project1-rtb-private1-us-east-1a
  - rtb-05e5d99a7111e9f1e
  - project1-rtb-private2-us-east-1b
- Network connections (1):** project1-igw

## Task 2: Creating a virtual machine

The screenshot displays the AWS Management Console interface for a VPC. The top navigation bar shows the AWS logo, a search bar, and the user's profile. The left sidebar contains the 'VPC dashboard' and a list of services including Virtual private cloud, Security, DNS firewall, and Network Firewall. The main content area shows the details of the VPC 'vpc-0ebd274cfb648c0ad / project1-vpc'. The details are organized into a table with four columns: VPC ID, State, DNS hostnames, and DNS resolution. Below the details, there is a 'Resource map' section that provides a visual overview of the VPC's components, including subnets, route tables, and network connections.

**VPC dashboard** x

EC2 Global View [↗](#)

Filter by VPC ▼

▼ Virtual private cloud

▼ Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet gateways

Carrier gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

Endpoint services

NAT gateways

Peering connections

▼ Security

Network ACLs

Security groups

▼ DNS firewall

Rule groups

Domain lists

▼ Network Firewall

cloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

VPC > Your VPCs > vpc-0ebd274cfb648c0ad

vpc-0ebd274cfb648c0ad / project1-vpc

Actions ▼

**Details** info

VPC ID vpc-0ebd274cfb648c0ad	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-0cc7f54533a2d796a	Main route table rtb-05e5d99a7111e9f1e	Main network ACL acl-07c98bb06d50403d0
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 411641690222	

**Resource map** info

Subnets (4)  
Subnets within this VPC

us-east-1a

- project1-subnet-public1-us-east-1a
- project1-subnet-private1-us-east-1a

us-east-1b

- project1-subnet-public2-us-east-1b
- project1-subnet-private2-us-east-1b

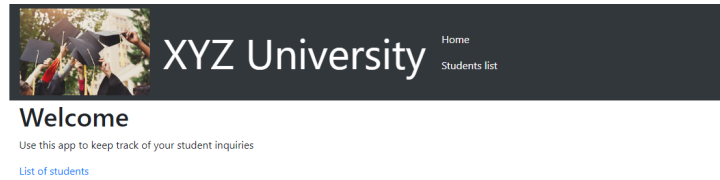
Route tables (4)  
Route network traffic to resources

- project1-rtb-public
- project1-rtb-private1-us-east-1a
- rtb-05e5d99a7111e9f1e
- project1-rtb-private2-us-east-1b

Network connections (1)  
Connections to other networks

- project1-igw

## Task 3: Testing the deployment



---

## Phase 3: Decoupling the application components

### Task 1: Changing the VPC configuration

```
voclabs:~/environment $  
voclabs:~/environment $ aws --version  
aws-cli/2.17.55 Python/3.12.6 Linux/6.8.0-1015-aws exe/x86_64.ubuntu.22
```

Made sure that aws installed in the machine

### Task 2: Creating and configuring the Amazon RDS database

OverviewRotationVersionsReplicationTags

Secret value Info

CloseEdit

Retrieve and view the secret value.

Key/valuePlaintext

Secret key	Secret value
username	admin
password	.-%iJRcwOT6!p7D*fr6Ksp.680xH
engine	mysql
host	database-project1.cgjsmhm4wmn2.us-east-1.rds.amazonaws.com
port	3306
dbInstanceIdentifier	database-project1

RWS Services Search [Alt+S]

N. Virginia vodlabs/user2801417~Rowan\_Mohamed\_Hussein @ 4116-4169-0222

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.

View detailsSee sample code

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	-	-

Madea new secret with username and password that i already have

### Task 3: Configuring the development environment

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	-	-
rdsldb-7a3d7525-1d77-44fb-bf1e-5a78de6a2496	The secret associated with the primary RDS DB instance: arn:aws:rds:us-east-1:007403005255:db:habiladatabase-1	September 20, 2024

## Task 5: Provisioning a new instance for the web server

i-034782508e4704016 (Rowanwebserver)

Details

Status and alarms

Monitoring

Security

Networking

Storage

Tags

▼ Instance summary Info

Instance ID

i-034782508e4704016 (Rowanwebserver)

IPv6 address

—

Hostname type

IP name: ip-10-0-14-233.ec2.internal

Answer private resource DNS name

—

Auto-assigned IP address

18.209.5.107 [Public IP]

IAM Role

LabRole

Public IPv4 address

18.209.5.107 | open address

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-14-233.ec2.internal

Instance type

t2.micro

VPC ID

vpc-0ebd274cfb648c0ad (project1-vpc)

Subnet ID

subnet-0d11ed3de145cef37 (project1-subnet-public1-us-east-1a)

Private IPv4 addresses

10.0.14.233

Public IPv4 DNS

ec2-18-209-5-107.compute-1.amazonaws.com | open address

Elastic IP addresses

—

AWS Compute Optimizer finding

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


Auto Scaling Group name

—

making a new instance called Rowanwebserver with ubuntu and t2.micro

## Task 6: Migrating the database

migrate the data that is in Project1 to RDS try to add data and then check if the data moved or not will add some students



XYZ University

Home

Students list

All students

Name	Address	City	State	Email	Phone	
testtesttest	testtesttesttest	test	testtesttest	test@gmail.com	0123456789	edit
testtesttesttesttesttesttest	testtesttesttesttesttest	testtest	testtesttest	testtesttest@gmail.com	1234567897	edit
test	test	test	test	testtest@gmail	2314567985	edit

Add a new student

deleting secret key without-recovery with credential aws secretsmanager delete-secret  
--secret-id Mydbsecret --force-delete-without-recovery --region us-east-1 AND CREATE A NEW  
SECRET KEY

```
ubuntu@ip-10-0-15-87:~$ aws secretsmanager create-secret \
  --name Mydbsecret \
  --description "Database secret for web app" \
  --secret-string "{\"user\":\"admin\", \"password\":\"3GaQr2W0z3E{t@uOBmyGdItt+6:y\", \"host\":\"habibadatabase-1.c5u8wmg20ua4.us-east-1.rds.amazonaws.com\", \"db\":\"STUDENTS\"}"
{
  "ARN": "arn:aws:secretsmanager:us-east-1:007403005255:secret:Mydbsecret-hoD36U",
  "Name": "Mydbsecret",
  "VersionId": "603ff507-eb0c-40eb-b054-75c0aca74c08"
}
ubuntu@ip-10-0-15-87:~$
```

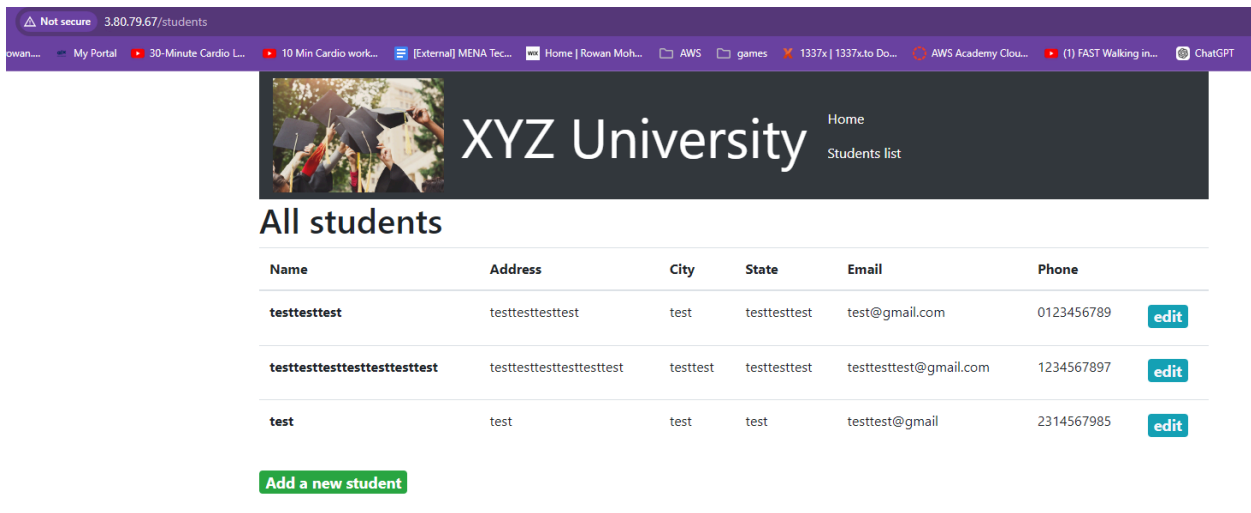
See the data in the DB

```
*!40000 ALTER TABLE `students` DISABLE KEYS *;/
INSERT INTO `students` VALUES (1,'Habiba','Alexandria','Alexandria','Egypt','1@y.com','123456789'),
('Habibaa Hossam','wabour el maya Bastour','Alexandria','الإسكندرية','Habibahossam.hh@gmail.com','1
345454878'), (3,'abc','abc','abc','abc','abc@y.com','1245545454');
*!40000 ALTER TABLE `students` ENABLE KEYS *;/
UNLOCK TABLES;
*!40103 SET TIME_ZONE=@OLD_TIME_ZONE *;/
```

Then adding the data to the RDS

```
mysql> show databases;
+-----+
| Database |
+-----+
| STUDENTS |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

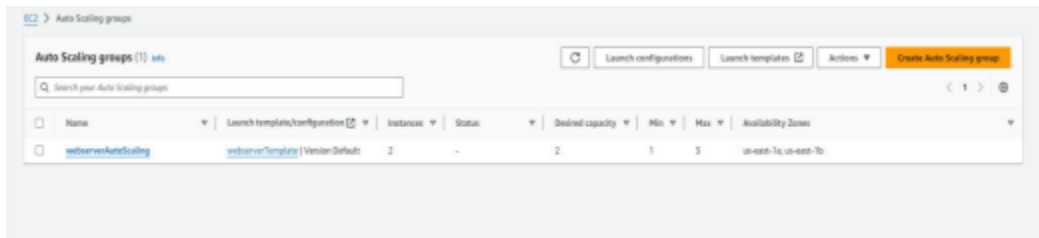
# Task 7: Testing the application



## Phase 4: Implementing high availability and scalability

### Task 1: Creating an Application Load Balancer

Made an image for The instance,create a template for auto scaling of launch template then create the auto scaling group





## Creating load Balancer by creating Target Group

EC2 > Target groups > webserverTarget

### webserverTarget

Details

arn:aws:elasticloadbalancing:us-east-1:007403005255:targetgroup/webserverTarget/21b63754a0b58731

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC <a href="#">vpc-0194e06f2aa35b37a</a>
IP address type IPv4	Load balancer <a href="#">None associated</a>		

2 Total targets

0 Healthy

0 Unhealthy

0 Anomalous

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.

Targets | Monitoring | Health checks | Attributes | Tags

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.

Anomaly mitigation: **Not applicable**

Deregister | Register targets

Filter targets

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection result
<input type="checkbox"/>	<a href="#">i-0be6f3c3d5ea280c3</a>		80	us-east-1a	Unused	Target group is not co...	Septembe...	Normal
<input type="checkbox"/>	<a href="#">i-075c8b86996096d59</a>		80	us-east-1b	Unused	Target group is not co...	Septembe...	Normal

Activate Windows  
Go to Settings to activate Windows

## Then create load balancer

EC2 > Load balancers > habibaWebserver-lb

### habibaWebserver-lb

Details

Load balancer type Application	Status Provisioning	VPC <a href="#">vpc-0194e06f2aa35b37a</a>	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z35SXDOTRQ7K7K	Availability Zones <a href="#">subnet-009b85b525e2e525</a> us-east-1a (use1-ap4) <a href="#">subnet-0c9e342794783102b</a> us-east-1b (use1-as6)	Date created September 21, 2024, 23:28 (UTC+03:00)
Load balancer ARN <a href="#">arn:aws:elasticloadbalancing:us-east-1:007403005255:loadbalancer/app/habibaWebserver-lb/735115de478ae50b</a>	DNS name info <a href="#">habibaWebserver-lb-386462410.us-east-1.elb.amazonaws.com</a> [A Record]		

Listeners and rules | Network mapping | Resource map - new | Security | Monitoring | Integrations | Attributes | Tags

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.

Filter listeners

<input type="checkbox"/>	Protocol/Port	Default action	Rules	ARN	Security policy	Default SSL/TLS certificate	mTLS	Trust store	Trust
<input type="checkbox"/>	<a href="#">HTTP:80</a>	Forward to target group • <a href="#">webserverTarget</a> 1 (100%) • Target group stickiness: Off	<a href="#">1 rule</a>	<a href="#">ARN</a>	Not applicable	Not applicable	Not applicable	Not applicable	Not applicable

## webserverTarget

Actions ▾

### Details

arn:aws:elasticloadbalancing:us-east-1:007403005255:targetgroup/webserverTarget/21683754oxy58751

Target type Instance	Protocol : Port HTTP-80	Protocol version HTTP1	VPC <a href="#">vpc-0194e0d6f2aa15b57a</a>
IP address type IPv4	Load balancer <a href="#">None associated</a>		

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

Targets Monitoring Health checks Attributes Tags

### 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 2 healthy targets.

Anomaly mitigation: **Not applicable** [Refresh](#) [Deregister](#) [Register targets](#)

< 1 > [Refresh](#)

<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Launch...	Anomaly detection result
<input type="checkbox"/>	<a href="#">i-0befc03d8aaa280d5</a>		80	us-east-1a	Healthy	-	September...	Normal
<input type="checkbox"/>	<a href="#">i-075c8b8d996916459</a>		80	us-east-1b	Healthy	-	September...	Normal

Activate Windows  
Go to Settings to activate Windows.