Configuration Manual

MSc Research Project
MSc Cloud Computing

Anbu Arivu Selvi Raviselvam
Student ID: 22170901

School of Computing
National College of Ireland

Supervisor: Dr Giovani Estrada

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anbu Arivu Selvi Raviselvam
Student ID:	22170901
Programme:	MSc Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr Giovani Estrada
Submission Due Date:	29/01/2025
Project Title:	A cost-benefit analysis of AWS and Alibaba services for cloud migration
Word Count:	2049
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Anbu Arivu Selvi Raviselvam
Date:	28th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Anbu Arivu Selvi Raviselvam
22170901

1 Introduction

To replicate the same objectives as the original work, this configuration manual can be used. It consists of the device configuration that was used during the research, the methods used during the implementation process.

2 System Requirements

The below section as in Table 1 illustrates the hardware configurations, device requirements and the library packages required to reproduce the study.

Feature	**Specification**
Environment	Visual Studio Code, AWS & Alibaba Cloud
Operating System	Windows 11 Pro 64-bit operating system
RAM	32.0 GB
Processor	12th Gen Intel(R) Core (TM) i7-1265U 1.80 GHz
Hard disk Storage	476.92 GB

Table 1: System Specifications

3 Required Software

The below software had been used during the research

- GitHub
- VS code
- Node.js 16.14.2
- XAMPP
- DBeaver
- Apache Jmeter
- AWS services – EC2, S3, AWS RDS
- Alibaba Services – ECS, OSS, PolarDB

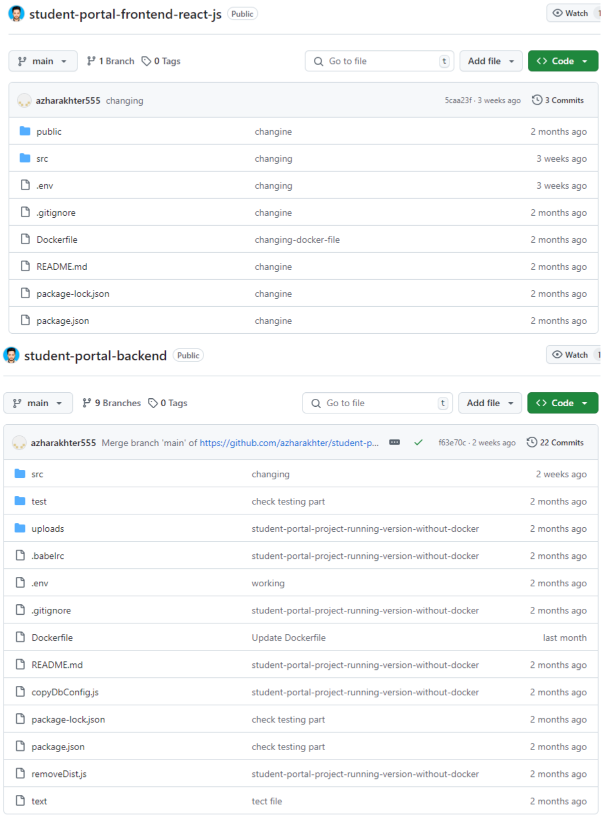
The study clones the existing application from GitHub, VS code was the developer environment for the node.js application, XAMPP mirrors production-like environment, Database is managed by DBeaver tool, while Apache Jmeter was the load testing tool that assisted with performance assessment. AWS and Alibaba were the two cloud platforms where the application was migrated to.

4 Application Replication (GitHub repository)

An existing application from GitHub was used during the research to deploy on multiple cloud platforms. Figure 1 a and b show the front-end and back-end application repositories.

GitHub Link:

- `https://github.com/azharakhter/student-portal-backend.git`
- `https://github.com/azharakhter/student-portal-frontend-react-js`



azharakhter555 changing · 5caa23f · 3 weeks ago · 3 Commits

public	changine	2 months ago
src	changing	3 weeks ago
.env	changing	3 weeks ago
.gitignore	changine	2 months ago
Dockerfile	changing-docker-file	2 months ago
README.md	changine	2 months ago
package-lock.json	changine	2 months ago
package.json	changine	2 months ago

azharakhter555 Merge branch 'main' of https://github.com/azharakhter/student-p... ✓ f63e70c · 2 weeks ago · 22 Commits

src	changing	2 weeks ago
test	check testing part	2 months ago
uploads	student-portal-project-running-version-without-docker	2 months ago
.babelrc	student-portal-project-running-version-without-docker	2 months ago
.env	working	2 months ago
.gitignore	student-portal-project-running-version-without-docker	2 months ago
Dockerfile	Update Dockerfile	last month
README.md	student-portal-project-running-version-without-docker	2 months ago
copyDbConfig.js	student-portal-project-running-version-without-docker	2 months ago
package-lock.json	check testing part	2 months ago
package.json	check testing part	2 months ago
removeDist.js	student-portal-project-running-version-without-docker	2 months ago
text	tect file	2 months ago

Figure 1: (a) Front-end Repository and (b) Back-end Repository

5 Research Workflow

The steps followed during the study are discussed in this section. A test application, which is the workload, is employed in this migration study. An existing application (Student Hub) is cloned from GitHub to a local setup (laptop). Secondly, the workload is deployed on two cloud environments, AWS and Alibaba. Followed by applying two migration strategies (Rehosting and Replatforming) to the replicated application to study cloud features that affect the performance of the application. The next step is assessment of key indicators on how these approaches affect performance and cost. Key metrics calculated in the research are Scalability, Response time, Throughput, Bandwidth, Error rate and Cost. Figure 2 shows the workflow on how the research was carried out.

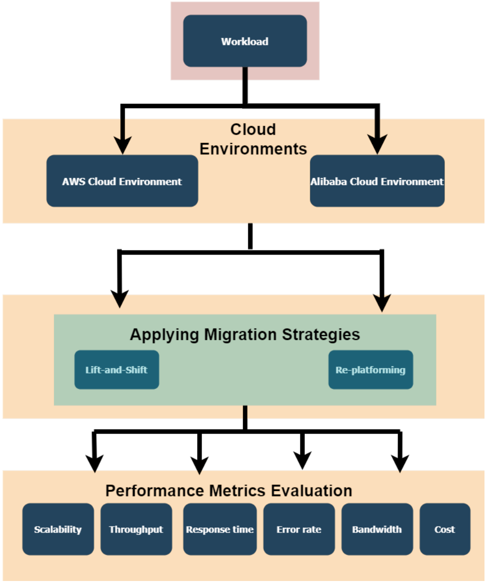


Figure 2: Research Workflow

6 Baseline Deployment- Local Server

- The first step is the replication of an application from GitHub. To replicate the application, the GitHub repository URL is copied. The Node.js application from GitHub is cloned to the local environment.

- Download Visual Studio Code [1], the developer environment chosen for the research, now a folder is created, where the application is to be cloned. The folder is opened in Visual Studio code and proceeded with project replication.

- To replicate the application, the command below is used ensuring that the local copy is identical to the one in GitHub and directed to the parent folder. The repository URL can be found in section 4.

  ```
  git clone repo_url
  cd repo_folder
  ```

- Once the repository is cloned, the necessary dependencies for the node.js application are to be installed using Node Package Manager (NPM). The below step installs all the packages and library files mentioned in the package.json file that are required for the proper functioning of the application. This is done by the command:

  ```
  npm install
  ```

[1] https://code.visualstudio.com/

- The application is configured to run on local host (The application's code is attached along the code zip files).
  Backend - 8080
  Database - 3306
  Frontend - 3000

- A build file is created for the application for running the tasks using the below command:

  npm run build

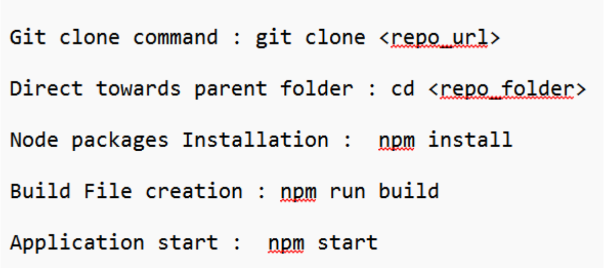
- An .env file is created with all the keys, configuration and database credentials, preventing sensitive information from being exposed. Figure 3 shows the commands that were executed to deploy the application on local environment.

- The application is run using the command:

  npm start

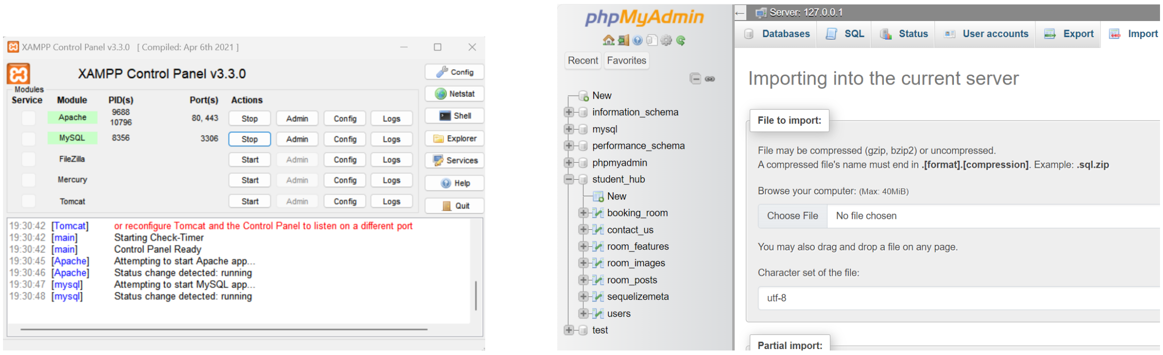
- To replicate a production-like atmosphere, XAMPP is used as the local server, Install XAMPP[2] and within XAMPP, Apache (Web server) and MySQL(database) are enabled as shown in figure 4a.

- MySQL acts as the local server database and the PHPMyAdmin tool is utilized to interact with the database. The application's database schema (relationships and tables) was imported into MySQL using the tool as shown in figure 4b, thus mirroring the local environment to production setup (The table schema file is attached along the code zip files).



```
Git clone command : git clone <repo_url>

Direct towards parent folder : cd <repo_folder>

Node packages Installation :  npm install

Build File creation : npm run build

Application start :  npm start
```

Figure 3: Commands for local host deployment



XAMPP Control Panel v3.3.0   [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions			
	Apache	9688 10796	80, 443	Stop	Admin	Config	Logs
	MySQL	8356	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
19:30:42  [Tomcat]    or reconfigure Tomcat and the Control Panel to listen on a different port
19:30:42  [main]      Starting Check-Timer
19:30:42  [main]      Control Panel Ready
19:30:45  [Apache]    Attempting to start Apache app...
19:30:46  [Apache]    Status change detected: running
19:30:47  [mysql]     Attempting to start MySQL app...
19:30:48  [mysql]     Status change detected: running
```

Server: 127.0.0.1

Importing into the current server

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

Choose File No file chosen

You may also drag and drop a file on any page.

Character set of the file: utf-8

Partial import:

Figure 4: (a) XAMPP: Apache and MySQL (b) Importing Schema to phpMyAdmin

[2] https://www.apachefriends.org/download.html

6.1 Application Connection

The application is configured as below to run on local host (The application's code is attached along the code zip files). The Front-end and back-end deployed in local host is shown in figure 5 a and b.
Backend - 8080
Database - 3306
Frontend - 3000

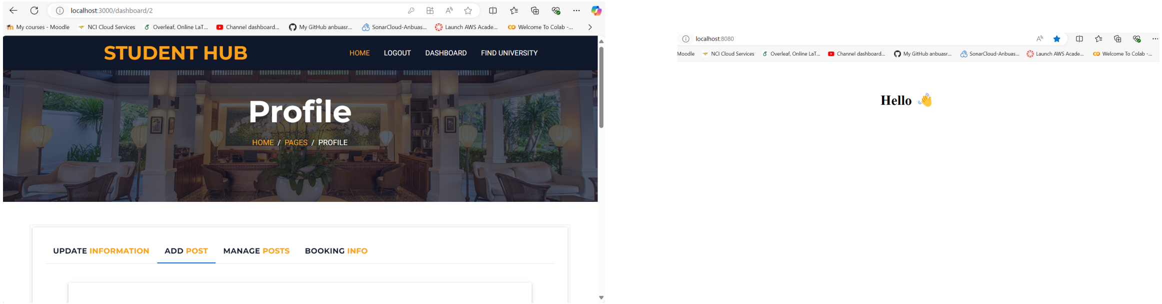


Figure 5: (a) Local Host Front-end (b)Local Host Back-end

7 Database Migration

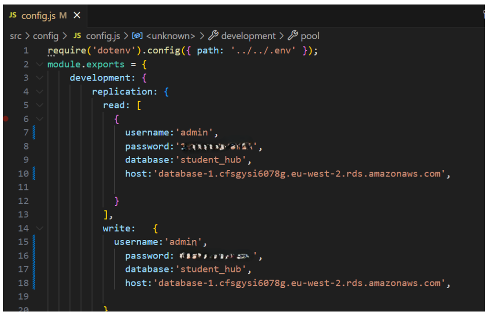
Database is migrated to AWS and Alibaba Cloud Services using migration scripts. The below steps are followed to migrate the local database to cloud environments.

- Install Sequelize CLI using the below command in VS code for MySQL database drivers.
  npm install sequelize mysql2

- Configure the config.js file including the database endpoint, credentials and port details for AWS, local host and Alibaba cloud individually. An example with Alibaba cloud database details are shown in figure 6.

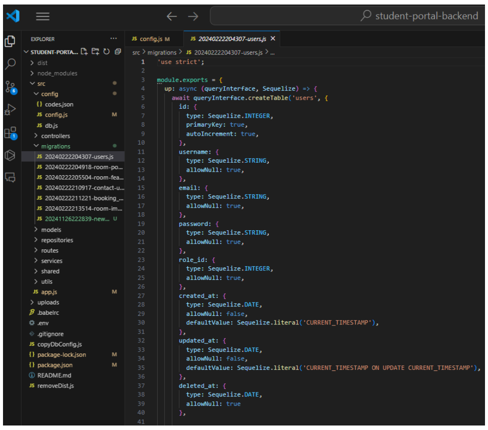
- The models are created already as shown in figure 7 and is attached along the code zip files.

- Now run the below command to migrate the database using the sequelize scripts.
  npx sequelize-cli db:migrate



```
require('dotenv').config({ path: '../../.env' });
module.exports = {
    development: {
        replication: {
          read: [
            {
              username:'admin',
              password:'[redacted]',
              database:'student_hub',
              host:'database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com',

            }
          ],
          write:   {
            username:'admin',
              password: [redacted]',
              database:'student_hub',
              host:'database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com',

          }
```

Figure 6: Database Migration- Config.js



```
'use strict';

module.exports = {
  up: async (queryInterface, Sequelize) => {
    await queryInterface.createTable('users', {
      id: {
        type: Sequelize.INTEGER,
        primaryKey: true,
        autoIncrement: true,
      },
      username: {
        type: Sequelize.STRING,
        allowNull: true,
      },
      email: {
        type: Sequelize.STRING,
        allowNull: true,
      },
      password: {
        type: Sequelize.STRING,
        allowNull: true,
      },
      role_id: {
        type: Sequelize.INTEGER,
        allowNull: true,
      },
      created_at: {
        type: Sequelize.DATE,
        allowNull: false,
        defaultValue: Sequelize.literal('CURRENT_TIMESTAMP'),
      },
      updated_at: {
        type: Sequelize.DATE,
        allowNull: false,
        defaultValue: Sequelize.literal('CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP'),
      },
      deleted_at: {
        type: Sequelize.DATE,
        allowNull: true
      },
```

Figure 7: Database Migration- Sequelize Scripts

8 Database Management

DBeaver is used for database management as the study involves multiple databases from AWS and Alibaba. The steps below assist in managing the databases in DBeaver.

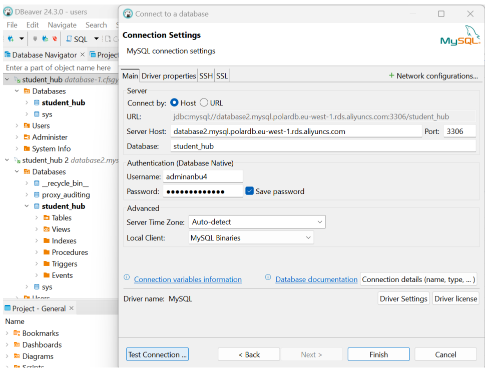
- Install DBeaver[3] and start it.

- Create a new connection using the + icon at the top left corner and choose MySQL database.

- Provide the following details for each use case (The details vary with local host, AWS, Alibaba) as shown in figure 8 and click on test connection.
  Host: Database URL
  Port: 3306
  Database: student_hub
  Username: the database username: admin
  Password: the database password: **********

- If the connection succeeds, Click on finish. The tables and data will be now visible within DBeaver as seen in figure 9.

- If the connection fails, make sure the details are entered correct and test again.



Connection Settings
MySQL connection settings

Server
Connect by: ● Host ○ URL
URL: jdbc:mysql://database2.mysql.polardb.eu-west-1.rds.aliyuncs.com:3306/student_hub
Server Host: database2.mysql.polardb.eu-west-1.rds.aliyuncs.com Port: 3306
Database: student_hub

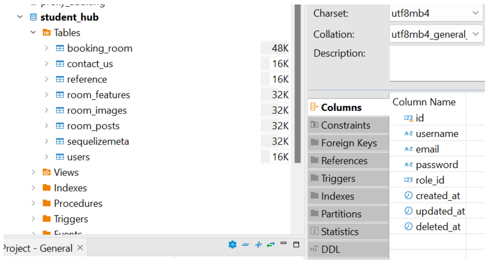
Authentication (Database Native)
Username: adminanbu4
Password: ●●●●●●●●●●●●● ☑ Save password

Advanced
Server Time Zone: Auto-detect
Local Client: MySQL Binaries

Driver name: MySQL

Figure 8: DBeaver- Database Management

[3] https://dbeaver.io/download/



student_hub

Tables

booking_room	48K
contact_us	16K
reference	16K
room_features	32K
room_images	32K
room_posts	32K
sequelizemeta	32K
users	16K

Views

Indexes

Procedures

Triggers

Charset: utf8mb4

Collation: utf8mb4_general_

Description:

Column Name

- id
- username
- email
- password
- role_id
- created_at
- updated_at
- deleted_at

Figure 9: DBeaver- Migrated Tables

9 AWS Deployment

9.1 Rehosting

AWS Rehosting involves 3 cloud services, AWS RDS for database, EC2 for backend and S3 for frontend.

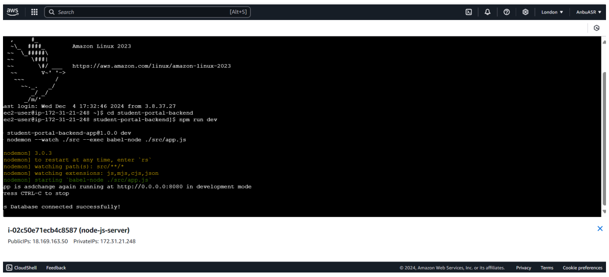
9.1.1 Backend Setup at EC2

- Launch AWS console[4] and search for EC2 and click on launch Instance.
- Choose the operating system (Linux) and select the instance type based on requirements. (e.g., t2.micro since our application is small).
- Ensure the security groups allow inbound HTTP/HTTPS requests. (port 80/443)
- For SSH access (https) configure key pairs and make a note of the endpoint.
- Now connect to EC2 instance by clicking on Connect option.
- Once the instance is launched, git clone the application's backend using the below command. The repository URL can be found in section 4.
  git clone repo_url
- Install the node packages for the application using the below command.
  npm install
- A build file is created for the application for running the tasks using the below command:

  npm run build
- The application's backend is run using the below command as shown in figure 10.

  npm start
- The application's backend will be running on the EC2 endpoint.
  Endpoint: `http://ec2-18-175-123-129.eu-west-2.compute.amazonaws.com:8080/`

[4] `https://aws.amazon.com/`



```
Amazon Linux 2023

https://aws.amazon.com/linux/amazon-linux-2023

ast login: Wed Dec  4 17:32:46 2024 from 3.8.37.27
ec2-user@ip-172-31-21-248 ~]$ cd student-portal-backend
ec2-user@ip-172-31-21-248 student-portal-backend]$ npm run dev

 student-portal-backend-app@1.0.0 dev
 nodemon --watch ./src --exec babel-node ./src/app.js

nodemon] 3.0.3
nodemon] to restart at any time, enter `rs`
nodemon] watching path(s): src/**/*
nodemon] watching extensions: js,mjs,cjs,json
nodemon] starting `babel-node ./src/app.js`
pp is asdchange again running at http://0.0.0.0:8080 in development mode
ress CTRL-C to stop

s Database connected successfully!
```

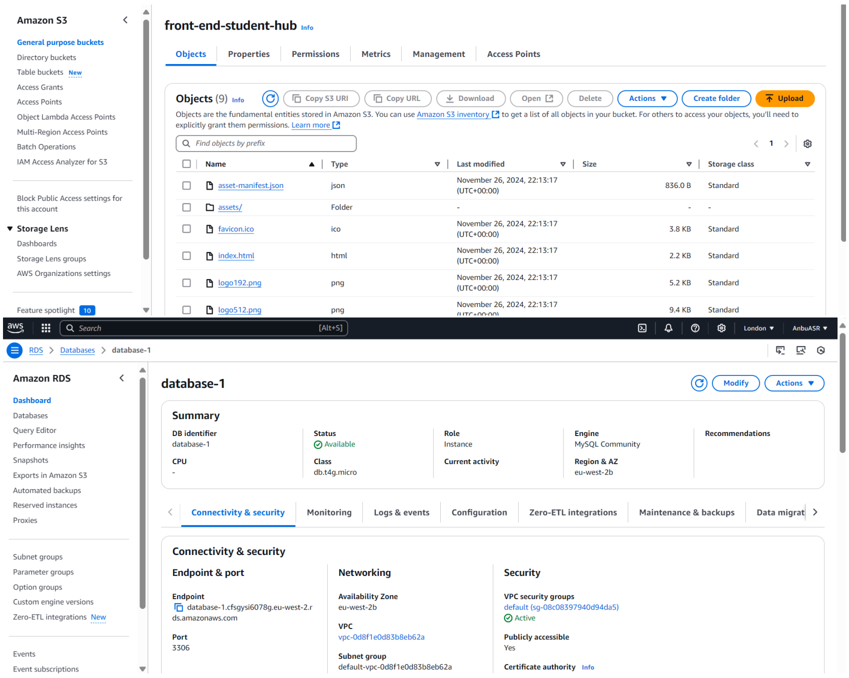
i-02c50e71ecb4c8587 (node-js-server)

PublicIPs: 18.169.163.50 PrivateIPs: 172.31.21.248

Figure 10: AWS EC2 instance

9.1.2 Frontend Setup at S3

- Launch AWS console[5] and search for S3 and click on Create bucket.
- Enable the option of static Website hosting.
- Upload the build (Static files include HTML, JS, CSS and image files) from the application's frontend folder into the bucket as shown in figure 11a.
- Set the permissions to public-read, ensuring the files are accessible to the public.
- After uploading the files, configure the bucket for static website hosting.
- Set the index.html as index document and error.html as error document.
- The URL to frontend will be provided as follows.
  URL: `http://front-end-student-hub.s3-website.eu-west-2.amazonaws.com/`



front-end-student-hub Info

Objects (9) Info

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
asset-manifest.json	json	November 26, 2024, 22:13:17 (UTC+00:00)	836.0 B	Standard
assets/	Folder	-	-	-
favicon.ico	ico	November 26, 2024, 22:13:17 (UTC+00:00)	3.8 KB	Standard
index.html	html	November 26, 2024, 22:13:17 (UTC+00:00)	2.2 KB	Standard
logo192.png	png	November 26, 2024, 22:13:17 (UTC+00:00)	5.2 KB	Standard
logo512.png	png	November 26, 2024, 22:13:17	9.4 KB	Standard

database-1

Summary

DB identifier: database-1
CPU: -
Status: Available
Class: db.t4g.micro
Role: Instance
Current activity
Engine: MySQL Community
Region & AZ: eu-west-2b
Recommendations

Connectivity & security

Endpoint & port
Endpoint: database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com
Port: 3306

Networking
Availability Zone: eu-west-2b
VPC: vpc-0d8f1e0d83b8eb62a
Subnet group: default-vpc-0d8f1e0d83b8eb62a

Security
VPC security groups: default (sg-08c08397940d94da5) Active
Publicly accessible: Yes
Certificate authority Info

Figure 11: (a) S3 Bucket(b) Amazon RDS database

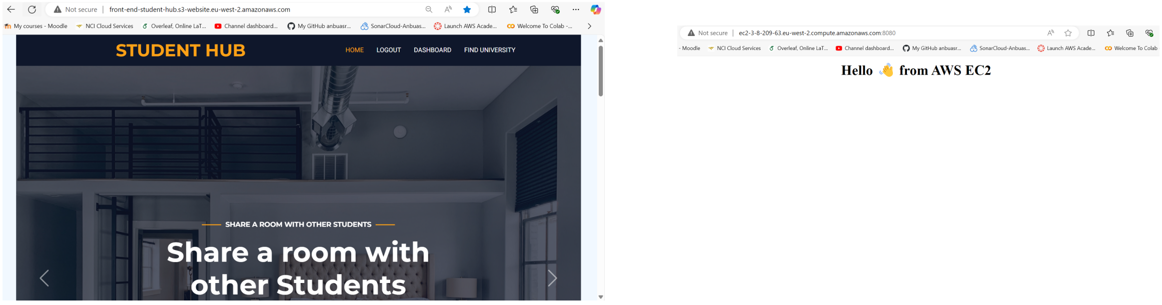
[5] `https://aws.amazon.com/`

9.1.3 Database Setup at AWS RDS

- Launch AWS console[6] and search AWS RDS and click on Create database.
- Select SQL database and instance type (db.t2.micro)
- Ensure the security groups allow inbound EC2 requests. (port 3306 for MySQL)
- Create a database username and password and note it down for connecting the database with Dbeaver as in section 8.
- The End point can be seen in figure 11b.
  Endpoint: `database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com`

9.1.4 Application Connection

Now use the above RDS endpoint in backend code of EC2 for database connection and also mention the EC2 endpoint at .env file of Frontend for entire application to work. The Front-end and back-end deployed in AWS are shown in figure 12 a and b.

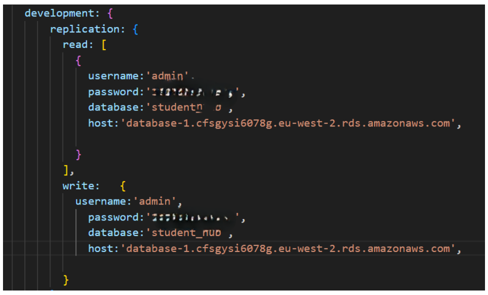


Hello 👋 from AWS EC2

Figure 12: (a) AWS Front-end (b) AWS back-end

9.2 Replatforming

- The application runs on local host, while the database alone is replatformed to AWS by configuring the config.js file with Amazon RDS URL in section 9 as shown in figure 13.
- The application will be running on local host with connection to AWS RDS database on providing the below command in VS code.
  npm start



```
development: {
    replication: {
      read: [
        {
          username:'admin'.
          password:'[redacted]',
          database:'student[redacted] ,
          host:'database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com',

        }
      ],
      write:   {
        username:'admin',
          password:'[redacted]',
          database:'student_[redacted] ,
          host:'database-1.cfsgysi6078g.eu-west-2.rds.amazonaws.com',

      }
```

Figure 13: AWS Replatforming Configuration

[6] `https://aws.amazon.com/`

10 Alibaba Deployment

10.1 Rehosting

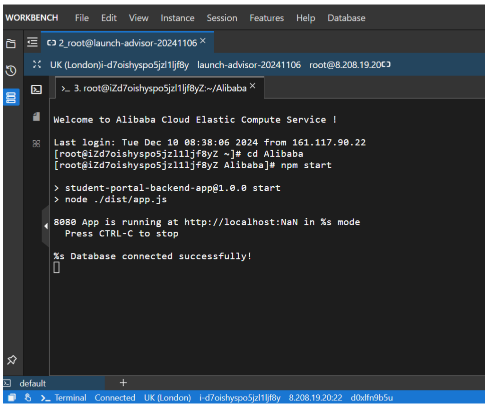
Alibaba Rehosting involves 3 cloud services, PolarDB for database, ECS for backend and OSS for frontend.

10.1.1 Backend Setup at ECS

- Launch Alibaba console[7] and search for ECS and click on launch Instance.
- Choose the operating system (Linux) and select the instance type based on requirements. (e.g., t2.micro since our application is small).
- Ensure the security groups allow inbound HTTP/HTTPS requests. (port 80/443)
- For SSH access (https) configure key pairs and make a note of the endpoint.
- Now connect to ECS instance by clicking on Connect option.
- Once the instance is launched, git clone the application's backend using the below command. The repository URL can be found in section 4.
  git clone repo_url
- Install the node packages for the application using the below command.
  npm install
- A build file is created for the application for running the tasks using the below command:

  npm run build
- The application's backend is run using the below command as shown in figure 14.

  npm start
- The application's backend will be running on the ECS endpoint.
  Endpoint: `http://8.208.19.20:8080/`


WORKBENCH
File
Edit
View
Instance
Session
Features
Help
Database
2_root@launch-advisor-20241106
UK (London)i-d7oishyspo5jzl1ljf8y launch-advisor-20241106 root@8.208.19.20
3. root@iZd7oishyspo5jzl1ljf8yZ:~/Alibaba
default
Terminal Connected UK (London) i-d7oishyspo5jzl1ljf8y 8.208.19.20:22 d0xlfn9b5u

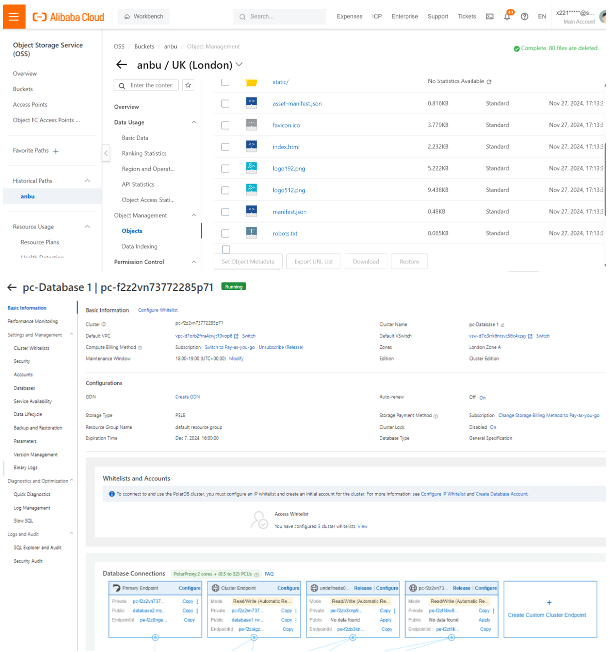
```
Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Tue Dec 10 08:38:06 2024 from 161.117.90.22
[root@iZd7oishyspo5jzl1ljf8yZ ~]# cd Alibaba
[root@iZd7oishyspo5jzl1ljf8yZ Alibaba]# npm start

> student-portal-backend-app@1.0.0 start
> node ./dist/app.js

8080 App is running at http://localhost:NaN in %s mode
  Press CTRL-C to stop

%s Database connected successfully!
```

Figure 14: Alibaba ECS instance

[7] `https://account.alibabacloud.com/`

10.1.2 Frontend Setup at OSS

- Launch Alibaba console[8] and search for OSS and click on Create bucket.
- Enable the option of static Website hosting.
- Upload the build files (Static files include HTML, JS, CSS and image files) from the application's frontend folder into the bucket as shown in figure 15a.
- Set the permissions to public-read, ensuring the files are accessible to the public.
- After uploading the files, configure the bucket for static website hosting.
- Set the index.html as index document and error.html as error document.
- The URL to frontend will be provided as follows.
  URL: `http://studenthubanbu.xyz/`



OSS / Buckets / anbu / Object Management

Complete. 80 files are deleted.

anbu / UK (London)

static/	No Statistics Available		
asset-manifest.json	0.816KB	Standard	Nov 27, 2024, 17:13:3
favicon.ico	3.779KB	Standard	Nov 27, 2024, 17:13:3
index.html	2.232KB	Standard	Nov 27, 2024, 17:13:3
logo192.png	5.222KB	Standard	Nov 27, 2024, 17:13:3
logo512.png	9.438KB	Standard	Nov 27, 2024, 17:13:3
manifest.json	0.48KB	Standard	Nov 27, 2024, 17:13:3
robots.txt	0.065KB	Standard	Nov 27, 2024, 17:13:3

pc-Database 1 | pc-f2z2vn73772285p71

Basic Information

Cluster ID	pc-f2z2vn73772285p71	Cluster Name	pc-Database 1
Default VPC	vpc-d7orb2frra4cwjt10vop6 Switch	Default VSwitch	vsw-d7o3rrk6nrsvc56c4xzey Switch
Compute Billing Method	Subscription Switch to Pay-as-you-go Unsubscribe (Release)	Zones	London Zone A
Maintenance Window	18:00-19:00 (UTC+00:00) Modify	Edition	Cluster Edition

Configurations

GDN	Create GDN	Auto-renew	Off On
Storage Type	PSL5	Storage Payment Method	Subscription Change Storage Billing Method to Pay-as-you-go
Resource Group Name	default resource group	Cluster Lock	Disabled On
Expiration Time	Dec 7, 2024, 16:00:00	Database Type	General Specification

Whitelists and Accounts

To cconnect to and use the PolarDB cluster, you must configure an IP whitelist and create an initial account for the cluster. For more information, see Configure IP Whitelist and Create Database Account.

Access Whitelist

You have configured 3 cluster whitelists, View

Database Connections PolarProxy:2 cores + (0.5 to 32) PCUs FAQ

Primary Endpoint Configure	Cluster Endpoint Configure	undefinede0... Release | Configure	pc-f2z2vn73... Release | Configure	Create Custom Cluster Endpoint
Private pc-f2z2vn737... Copy	Mode Read/Write (Automatic Re...	Mode Read/Write (Automatic Re...	Mode Read/Write (Automatic Re...	
Public database2.my... Copy	Private pc-f2z2vn737... Copy	Private pe-f2zb3knp9... Copy	Private pe-f2ztf4iw8... Copy	
EndpointId pe-f2z8nge... Copy	Public database1.rw... Copy	Public No data found Apply	Public No data found Apply	
	EndpointId pe-f2zoigz... Copy	EndpointId pe-f2zb3kn... Copy	EndpointId pe-f2ztf4i... Copy	

Figure 15: (a) OSS Bucket(b) PolarDB database

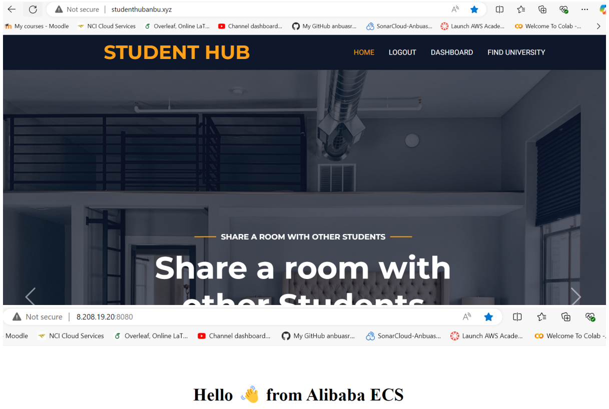
10.1.3 Database Setup at PolarDB

- Launch Alibaba console and search PolarDB and click on Create database.
- Select SQL database and instance type (db.t2.micro)
- Ensure the security groups allow inbound ECS requests. (port 3306 for MySQL)
- Create a database username and password and note it down for connecting the database with Dbeaver as in section 8.
- The End point can be seen in figure 15b.
  Endpoint: `database2.mysql.polardb.eu-west-1.rds.aliyuncs.com`

[8] `https://account.alibabacloud.com/`

10.1.4 Application Connection

Now use the above PolarDB endpoint in backend code of ECS for database connection and mention the ECS endpoint at .env file of Frontend for entire application to work. The Front-end and back-end deployed in Alibaba is shown in figure 16 a and b.


Not secure | studenthubanbu.xyz
STUDENT HUB
HOME
LOGOUT
DASHBOARD
FIND UNIVERSITY
Not secure | 8.208.19.20:8080

SHARE A ROOM WITH OTHER STUDENTS

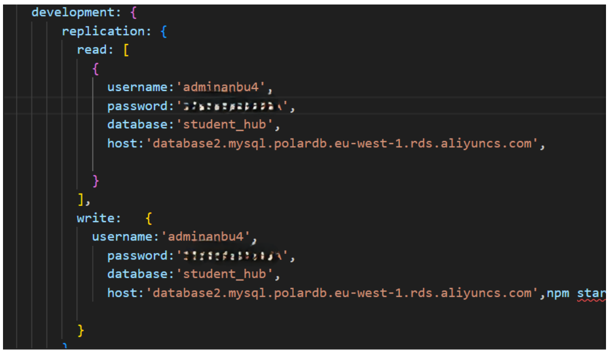
Share a room with other Students

Hello 👋 from Alibaba ECS

Figure 16: (a) Alibaba Front-end (b) Alibaba back-end

10.2 Replatforming

- The application runs on local host, while the database alone is replatformed to Alibaba by configuring the config.js file with PolarDB URL in section 10 as shown in figure 17.
- The application will be running on local host with connection to Alibaba PolarDB database on providing the below command in VS code.
  npm start



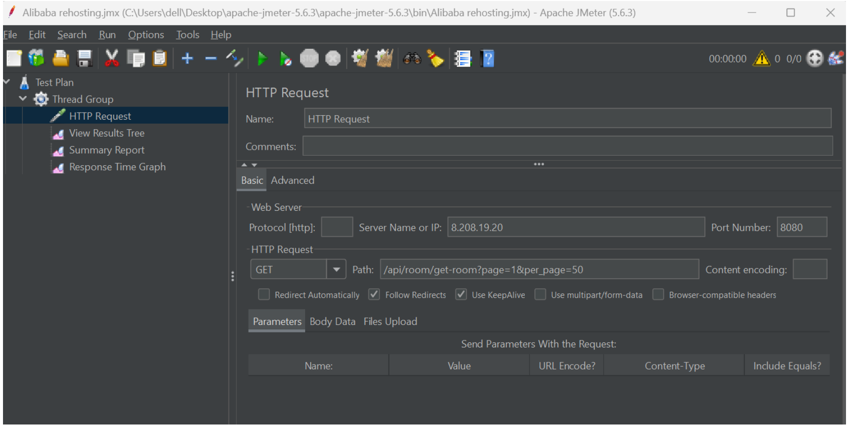
```
development: {
    replication: {
      read: [
        {
          username:'adminanbu4',
          password:'[redacted]',
          database:'student_hub',
          host:'database2.mysql.polardb.eu-west-1.rds.aliyuncs.com',

        }
      ],
      write:   {
        username:'adminanbu4',
          password:'[redacted]',
          database:'student_hub',
          host:'database2.mysql.polardb.eu-west-1.rds.aliyuncs.com',npm star

      }
```

Figure 17: Alibaba Replatforming Configuration

11 Performance Evaluation

Finally, the performance is monitored using the load testing tool called Apache Jmeter. The GET requests will be monitored to evaluate the performance indicators. The following steps below is followed to assess the performance indicators like response time, error rate, bandwidth and throughput.

- Download Apache Jmeter[9] and run to start the GUI.
- Create a test plan, with Thread Groups, Sampler and Listeners.
- A thread group is created defining the following:
  Threads: 20
  Ramp-Up time : 60
  Loop Count: infinite
- Simulate the user requests for each use case individually by providing the backend URL(local host, AWS rehosting and replatforming, Alibaba rehosting and replatforming).
- Configure the URL (Use case:1 - local host), request type (GET) and port number as shown in figure 18.
  Server name: localhost
  Port number: 8080
  Path: /api/room/get-room?page=1&per_page=50
- Collect the performance indicators information by adding Listeners for all 5 use cases individually (Listeners Used: Summary Report, Response Time Graph, View Results Tree).
- Save the plan and click on run.



Alibaba rehosting.jmx (C:\Users\dell\Desktop\apache-jmeter-5.6.3\apache-jmeter-5.6.3\bin\Alibaba rehosting.jmx) - Apache JMeter (5.6.3)

HTTP Request

Name: HTTP Request

Comments:

Web Server

Protocol [http]: Server Name or IP: 8.208.19.20 Port Number: 8080

HTTP Request

GET Path: /api/room/get-room?page=1&per_page=50 Content encoding:

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

Send Parameters With the Request:

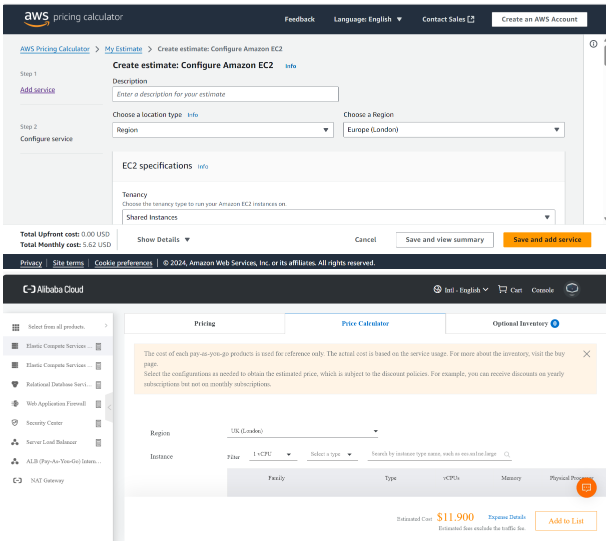
Name:	Value	URL Encode?	Content-Type	Include Equals?

Figure 18: Apache Jmeter Performance Evaluation

[9] https://jmeter.apache.org/download_jmeter.cgi

12 Cost Assessment

The Cost is assessed by the pricing calculators offered by AWS [10] and Alibaba Cloud vendors [11]. Figure 19 shows the pricing calculator of AWS and Alibaba cloud.



AWS Pricing Calculator > My Estimate > Create estimate: Configure Amazon EC2

Create estimate: Configure Amazon EC2

Step 1 Add service

Step 2 Configure service

Description: Enter a description for your estimate

Choose a location type: Region

Choose a Region: Europe (London)

EC2 specifications

Tenancy

Choose the tenancy type to run your Amazon EC2 instances on.

Shared Instances

Total Upfront cost: 0.00 USD

Total Monthly cost: 5.62 USD

© 2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The cost of each pay-as-you-go products is used for reference only. The actual cost is based on the service usage. For more about the inventory, visit the buy page.

Select the configurations as needed to obtain the estimated price, which is subject to the discount policies. For example, you can receive discounts on yearly subscriptions but not on monthly subscriptions.

Region: UK (London)

Instance: Filter: 1 vCPU, Select a type, Search by instance type name, such as ecs.sn1ne.large

Family	Type	vCPUs	Memory	Physical Processor

Estimated Cost $11.900

Estimated fees exclude the traffic fee.

Figure 19: (a)AWS Pricing Calculator (b) Alibaba Pricing Calculator

[10] https://calculator.aws/

[11] https://www.alibabacloud.com/en/pricing-calculator/