



MALAYSIAN INSTITUTE OF INFORMATION TECHNOLOGY

OCTOBER 2025

IBB43203 – CLOUD COMPUTING

GROUP PROJECT ASSIGNMENT

PREPARED BY

Name	ID	LAB	Course
MUHAMMAD ASRI BIN AZIZAN	52222122034	L01-B01	BCE
MUHAMAD KHAIRUL IKHWAN BIN SAMSUDIN	52222122346	L01-B01	BCE
NUR MUNAWWARAH BINTI MUZAMIL	52222122129	L01-B01	BCE

PREPARED FOR

Lecturer	MEGAT NORULAZMI MEGAT MOHAMED NOOR – ASSOC. PROF. DR.
----------	-------------------------------------------------------

Table of Contents

TITLE	3
INTRODUCTION	3
OBJECTIVES	3
SYSTEM ARCHITECTURE.....	4
METHODOLOGY	4
STEP 0: Login and Set AWS Region.....	4
STEP 1: Create EC2 Instance	5
STEP 2: Install Web Server (Apache & PHP)	8
STEP 3: Create S3 Bucket (Storage)	9
STEP 4: Create DynamoDB Table (Database)	11
STEP 5: Create and Attach IAM Role	13
STEP 6: Networking Configuration.....	16
STEP 7: Deploy Website Files.....	16
STEP 8: Install AWS SDK for PHP	17
STEP 9: Access the Web Application	19
STEP 10: Verify Data in DynamoDB	21
RESULTS AND DISCUSSION.....	21
TEAM CONTRIBUTION	21
CONCLUSION.....	22

TITLE

Group Project Assignment

INTRODUCTION

Cloud computing is now an important technology in modern information systems. It enables users and organizations to utilize computing resources like servers, storage, databases, and applications over the Internet without having to own or manage physical hardware. Utilizing cloud services allows organizations to reduce expenses and implement applications more effectively. Cloud computing is extensively utilized in education, commerce, and various other fields.

Amazon Web Services (AWS) is a widely used cloud platform. AWS offers a variety of on-demand cloud services that facilitate computing, storage, networking, and database administration. These services are adaptable and can be scaled up or down according to user needs. AWS enables students and beginners to explore and learn cloud technologies for little to no expense.

This project utilizes AWS to create and launch a basic web application. Amazon EC2 hosts the web server, Amazon S3 provides storage, and Amazon DynamoDB serves as the database. AWS Identity and Access Management (IAM) is utilized to guarantee safe communication between services. Completing this project provides hands-on experience in launching a cloud-based application and understanding how various AWS services collaborate in a real-world context.

OBJECTIVES

1. To deploy a web application using AWS services.
2. To utilize AWS Compute, Storage, Networking, and Database services
3. To provide a clear, step-by-step guide that allows replication of the project.

SYSTEM ARCHITECTURE

The AWS services used in this project are Amazon EC2 which hosts the apache web server and PHP application. Amazon S3 is used to provide cloud storage for project-related files. Amazon DynamoDB is used as the database for storing project records. AWS IAM is used to manage secure access between AWS services. Lastly, this project uses default VPC, security group and connectivity from the free tier setup.

The system architecture consists of a user accessing the web application through a browser. Requests are handled by an Apache web server running on an EC2 instance. Application data is stored in DynamoDB, while static files may be stored in an S3 bucket. IAM roles allow secure communication between EC2, DynamoDB, and S3.

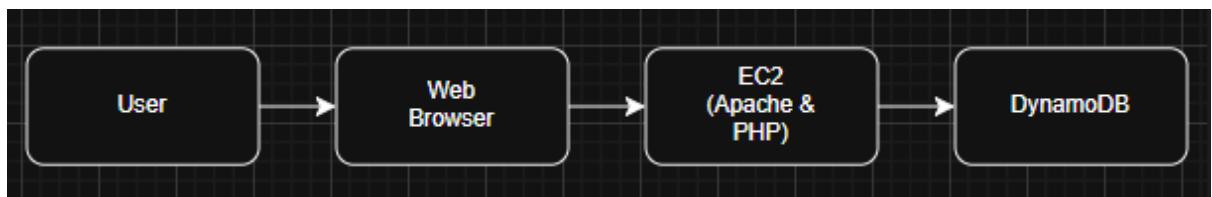
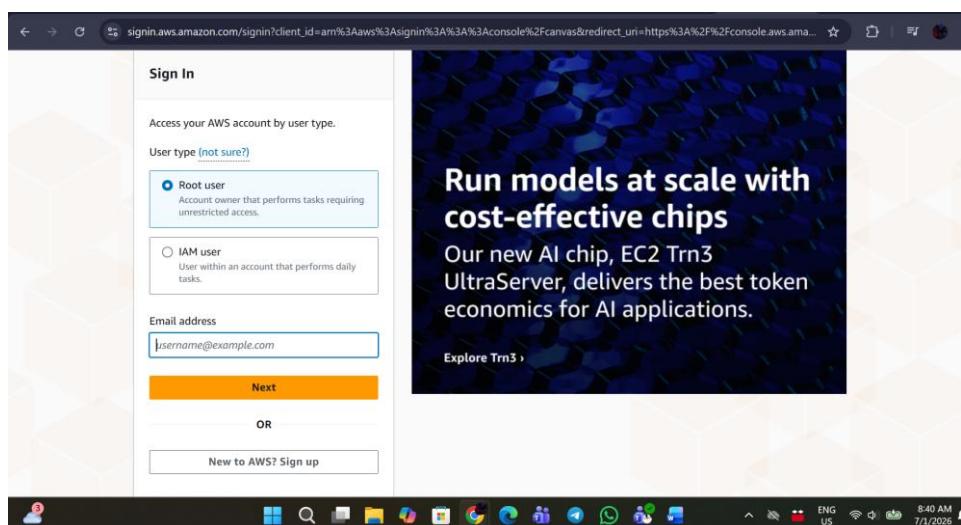


Figure 1 System architecture

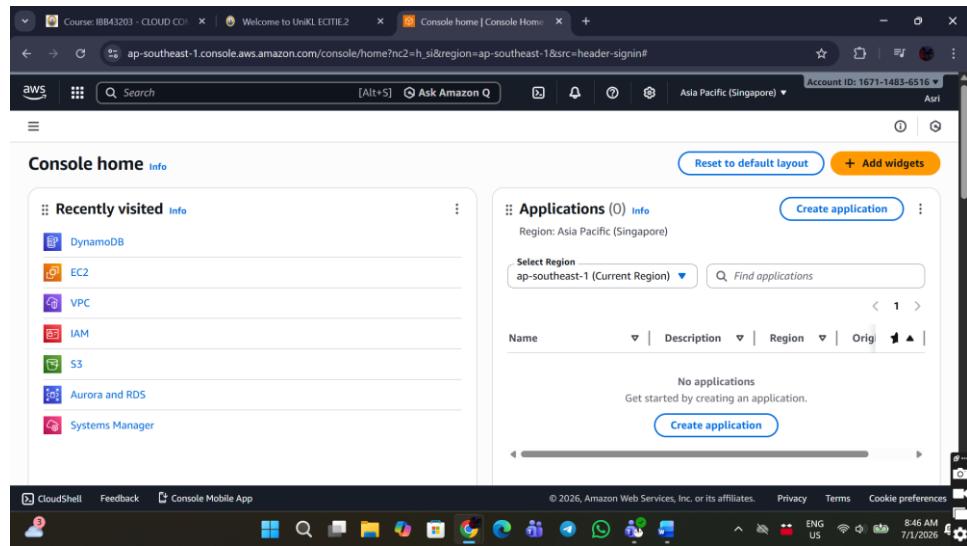
METHODOLOGY

STEP 0: Login and Set AWS Region

1. Login to the AWS Management Console

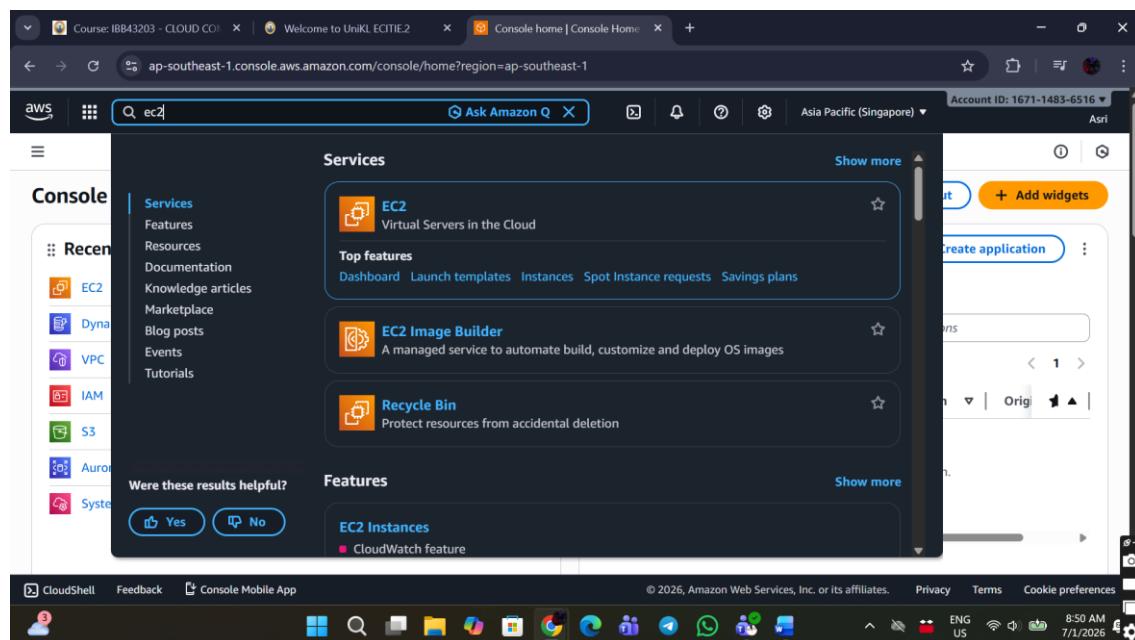


2. Set the Region to Asia Pacific (Singapore), ap-southeast-1.

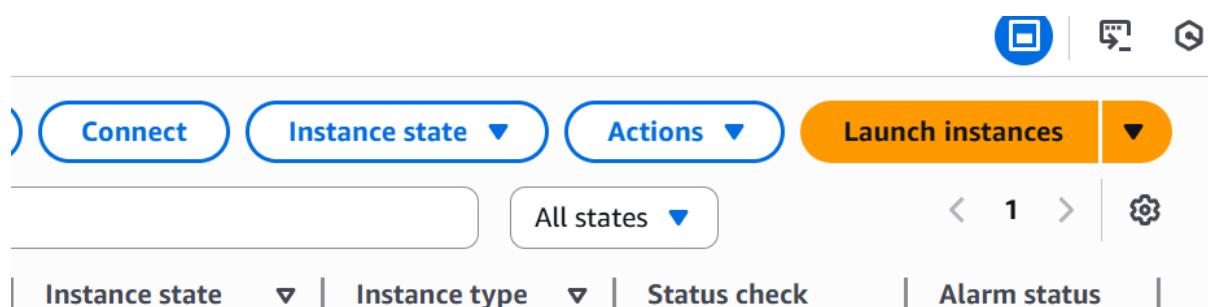


STEP 1: Create EC2 Instance

1. Search EC2 in AWS Console



2. Click Launch instances



3. Configure Instance

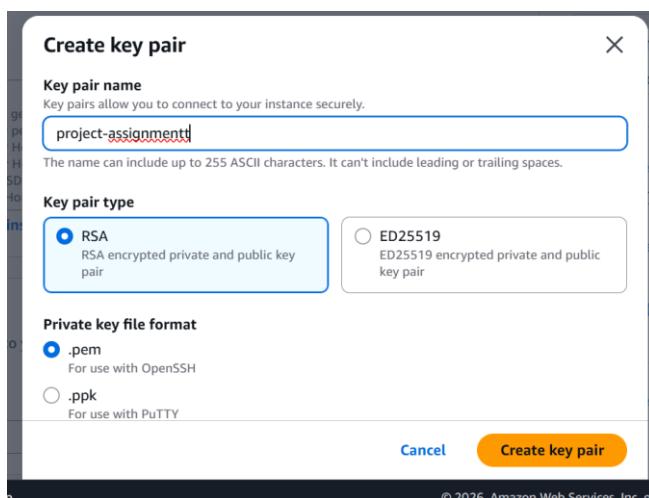
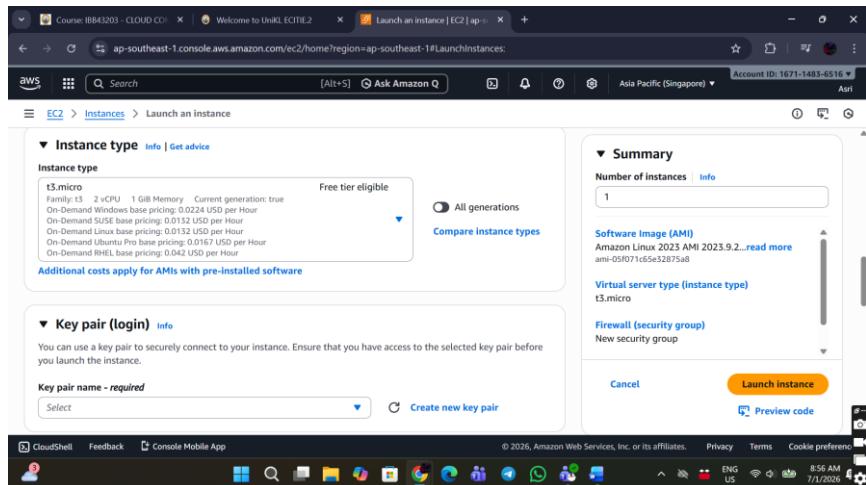
- Name: project-assignment-ec2

The screenshot shows the 'Launch an instance' page in the AWS EC2 console. In the 'Name and tags' section, the 'Name' field is filled with 'project-assignment-ec2'. There is also a link to 'Add additional tags'.

- AMI: Amazon Linux 2

The screenshot shows the 'Application and OS Images (Amazon Machine Image)' section. It displays a grid of recent and quick-start AMIs, including Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar at the top right allows users to search for more AMIs. To the right, there is a sidebar with navigation links for 'Instances', 'AMI', 'Software', 'Virtualization', and 'Firewall'.

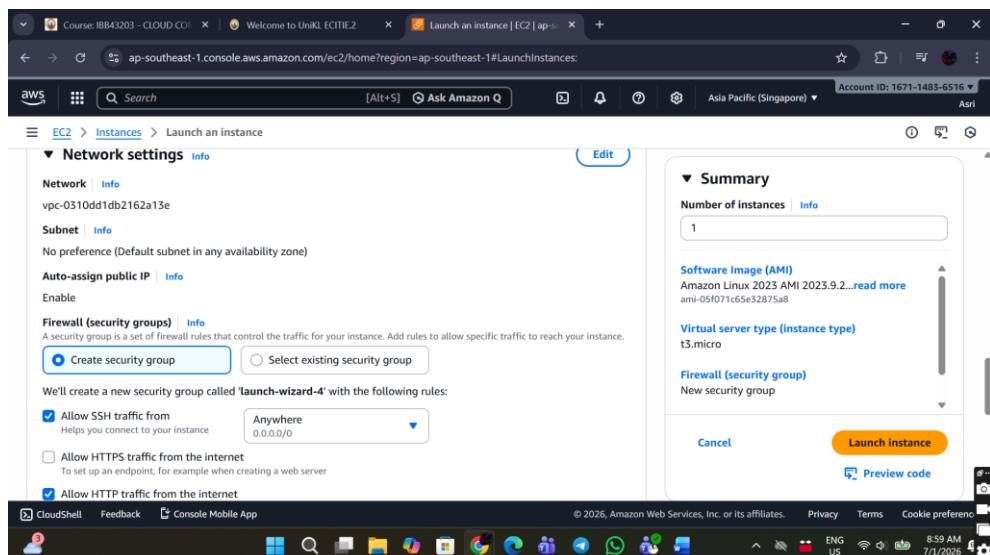
- Instance type: t3.micro
- Key pair: Create a new key pair and download it



4. Network Settings

Allow SSH traffic from (Port 22)

Allow HTTP (Port 80)



5. Click Launch instance at the bottom right corner.
6. Make sure the instance is running

The screenshot shows the AWS EC2 Instances page. At the top, there are three tabs: 'Course: IBB43203 - CLOUD COMPUTING' (active), 'Welcome to UniKL ECITIE.2', and 'Instances | EC2 | ap-southeast-1'. Below the tabs, the URL is 'ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:'. The page header includes the AWS logo, a search bar, and navigation links for 'EC2 > Instances'. On the right, it shows the account ID '1671-1483-6516' and the region 'Asia Pacific (Singapore)'. The main content area is titled 'Instances (1/2) Info' and shows a table with one row. The table columns are: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The single instance listed is 'project-assign...', with instance ID 'i-04ef75e363025e33e', state 'Running', type 't3.micro', status '3/3 checks passed', alarm status 'View alarms +', availability zone 'ap-southeast-1a', and public IP 'ec2-18-1-'. There are buttons for 'Connect', 'Actions', and 'Launch instances'.

STEP 2: Install Web Server (Apache & PHP)

1. Connect to EC2 using SSH, click the Connect button located besides the refresh button.

This screenshot shows the same EC2 Instances page as the previous one, but now with two instances listed. The first instance is a 't3.micro' running on 'ap-southeast-1a' with public IP 'ec2-18-1-'. The second instance is another 't3.micro' running on 'ap-southeast-1c' with public IP 'ec2-18-1-'. The 'Connect' button is visible next to the first instance's status bar.

2. Run the following commands:

```
sudo yum update -y
```

```
sudo yum install httpd php -y
```

```
sudo systemctl start httpd
```

```
sudo systemctl enable httpd
```

3. Verify Apache status, must show active (running)

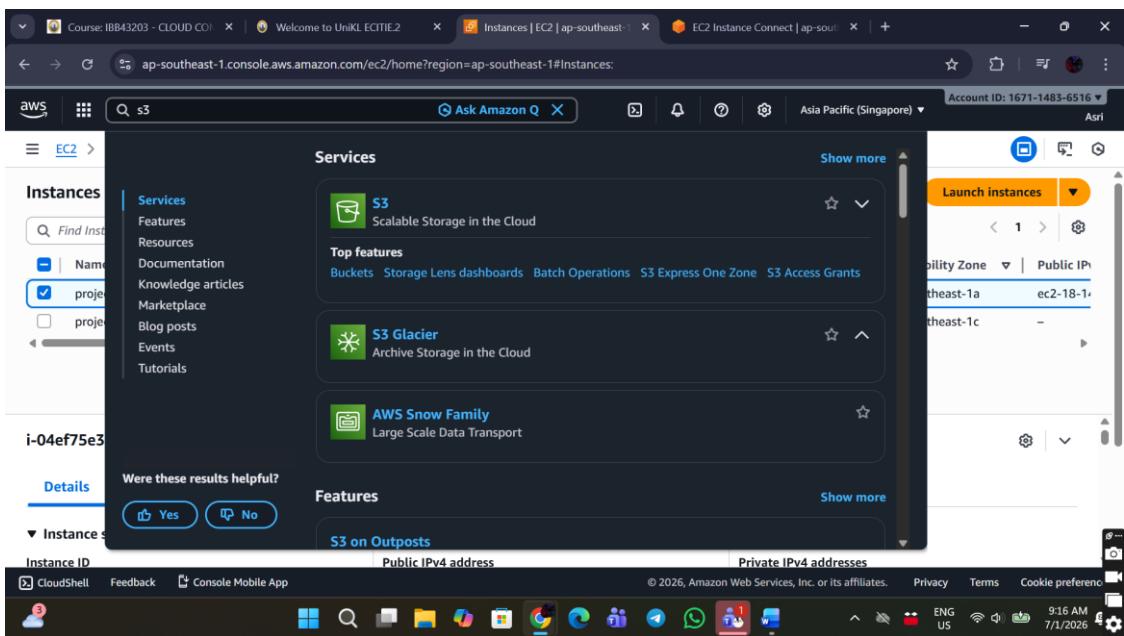
```
sudo systemctl status httpd
```

```
[ec2-user@ip-172-31-34-159 ~]$ sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Drop-In: /usr/lib/systemd/system/httpd.service.d
             └─php-fpm.conf
     Active: active (running) since Mon 2026-01-05 11:32:53 UTC; 1 day 13h ago
       Docs: man:httpd.service(8)
   Main PID: 26766 (httpd)
     Status: "Total requests: 418; Idle/Busy workers 100/0;Requests/sec: 0.00308; Bytes served/sec: 2 B/sec"
      Tasks: 230 (limit: 1067)
     Memory: 19.3M
        CPU: 2min 20.097s
       CGroup: /system.slice/httpd.service
               ├─26766 /usr/sbin/httpd -DFOREGROUND
               ├─26768 /usr/sbin/httpd -DFOREGROUND
               ├─26769 /usr/sbin/httpd -DFOREGROUND
               ├─26770 /usr/sbin/httpd -DFOREGROUND
               ├─26771 /usr/sbin/httpd -DFOREGROUND
               ├─27504 /usr/sbin/httpd -DFOREGROUND
               └─26767 /usr/sbin/httpd -DFOREGROUND

Jan 05 11:32:53 ip-172-31-34-159.ap-southeast-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
```

STEP 3: Create S3 Bucket (Storage)

1. Search S3
2. Click Create bucket



3. Bucket Configuration

- Bucket name: project-assignment-files-123
- Region: Asia Pacific (Singapore)

The screenshot shows the 'Create bucket' page in the AWS S3 console. Under 'General configuration', the 'AWS Region' is set to 'Asia Pacific (Singapore) ap-southeast-1'. The 'Bucket type' is set to 'General purpose', which is described as recommended for most use cases and access patterns. The bucket name is 'project-assignment-files-123'. There is a note about bucket names being 3 to 63 characters long and unique. A 'Copy settings from existing bucket - optional' section is present, with a 'Choose bucket' button and a note about copied settings. The status bar at the bottom shows various icons and the date/time.

- Public access: Untick Block all public access

The screenshot shows the 'Create bucket' page with the 'Block Public Access settings for this bucket' section expanded. It contains five checkboxes under the heading 'Block all public access':

- Block all public access**: Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLs)**: S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**: S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**: S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**: S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

A warning message at the bottom states: '⚠️ Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' The status bar at the bottom shows various icons and the date/time.

- Click Create bucket, wait until the bucket is successfully created.

The screenshot shows the AWS S3 buckets page. A green success message at the top states: "Successfully created bucket 'project-assignment-files-1234'". Below it, a table lists three general purpose buckets:

Name	AWS Region	Creation date
my-test-bucket-asri	US East (N. Virginia) us-east-1	December 31, 2025, 08:57:45 (UTC+08:00)
project-assignment-123	Asia Pacific (Singapore) ap-southeast-1	January 5, 2026, 19:18:10 (UTC+08:00)
project-assignment-files-1234	Asia Pacific (Singapore) ap-southeast-1	January 7, 2026, 09:19:30 (UTC+08:00)

On the right side, there are two sections: "Account snapshot" and "External access summary - new". The bottom of the screen shows the Windows taskbar with various pinned icons.

STEP 4: Create DynamoDB Table (Database)

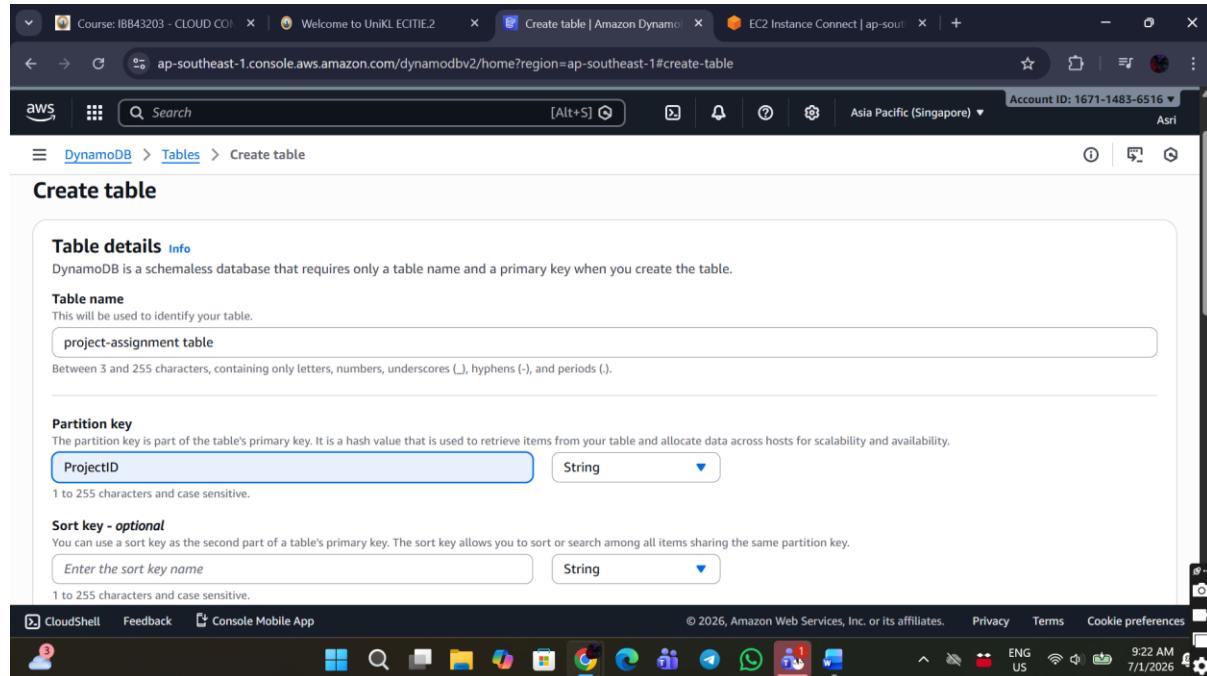
- Search DynamoDB

- Click Create table

The screenshot shows the AWS search results for "dynamoDB". The left sidebar has a "Success To upload" message and a "General" section with links like "Buckets", "Features", "Resources", etc. The main content area shows the "Services" section with "DynamoDB" highlighted as the top feature. Other services listed are "Amazon DocumentDB" and "CloudFront". The "Features" section below shows "Settings" and "DynamoDB feature". The bottom of the screen shows the Windows taskbar.

3. Table Configuration

- Table name: project-assignment-table
- Partition key: ProjectID (String)

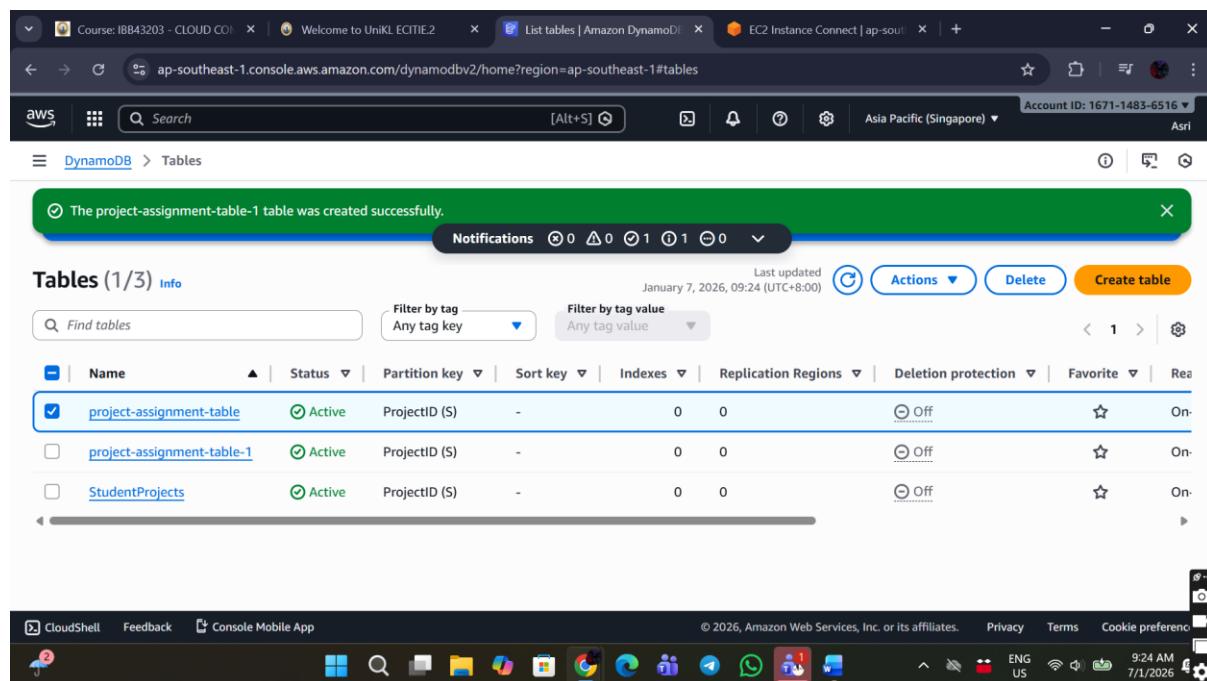


The screenshot shows the 'Create table' wizard in the AWS DynamoDB console. The 'Table details' section is open, showing the table name 'project-assignment-table' and the partition key 'ProjectID' of type String. The sort key field is empty. The status bar at the bottom indicates the table was created successfully on January 7, 2026.

- Use Default table settings, make sure Capacity mode is On-demand.

4. Click Create table

5. Wait until status is Active



The screenshot shows the 'Tables' page in the AWS DynamoDB console. A green notification bar at the top states 'The project-assignment-table-1 table was created successfully.' The table list shows three tables: 'project-assignment-table' (Active, 1 item, 0 bytes), 'project-assignment-table-1' (Active, 0 items, 0 bytes), and 'StudentProjects' (Active, 0 items, 0 bytes). The status bar at the bottom indicates the table was last updated on January 7, 2026.

STEP 5: Create and Attach IAM Role

This allows EC2 to securely access DynamoDB and S3.

1. Create IAM Role. Search and open IAM

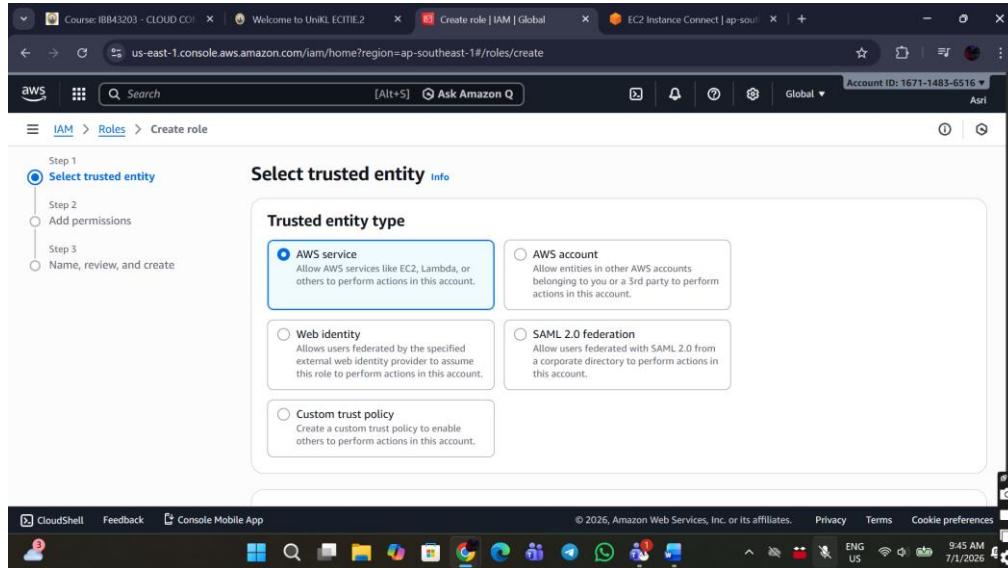
The screenshot shows the AWS Lambda service page. The left sidebar lists services like Features, Documentation, and Marketplace. The main content area displays the Lambda logo and the text "Lambda - Serverless compute for your code". Below this, there are sections for "Top features" (Groups, Users, Roles, Policies, Access Analyzer) and "Features" (Lambda Functions, Lambda@Edge, Lambda VPC). A search bar at the top right contains the term "iam". The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App.

2. Go to Roles → Create role

The screenshot shows the AWS IAM Roles page. The left sidebar under "Access Management" includes options like User groups, Users, Roles, Policies, Identity providers, Account settings, Root access management, and Temporary delegation requests. The main content area shows a table titled "Roles (5)" with columns for "Role name", "Trusted entities", and "Last activity". The roles listed are: AWSServiceRoleForResourceExplorer (AWS Service: resource-explorer-2 (Service-Linker), Last activity: 19 minutes ago), AWSServiceRoleForSupport (AWS Service: support (Service-Linker), Last activity: -), AWSServiceRoleForTrustedAdvisor (AWS Service: trustedadvisor (Service-Linker), Last activity: -), AWSSystemsManagerDefaultEC2InstanceManagementRole (AWS Service: ssm, Last activity: 55 days ago), and project-assignment-ec2-role (AWS Service: ec2, Last activity: 14 minutes ago). A "Create role" button is located at the top right of the table. The bottom navigation bar includes links for CloudShell, Feedback, and Console Mobile App.

3. Select:

- Trusted entity: AWS service
- Use case: EC2, Allows EC2 instances to call AWS services on your behalf.



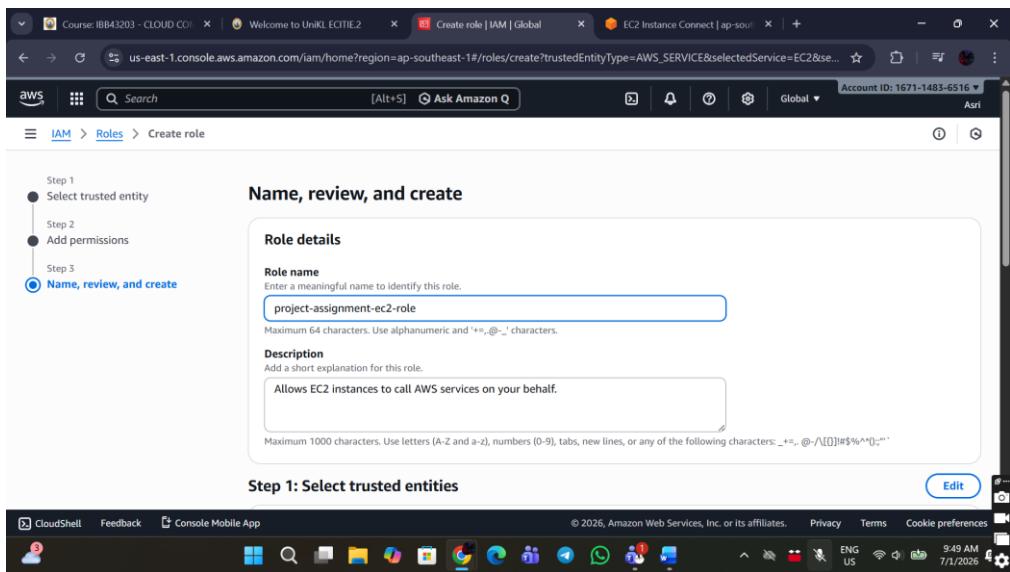
4. Add permissions. Attach Policies

Attach:

- AmazonDynamoDBFullAccess
- AmazonS3FullAccess

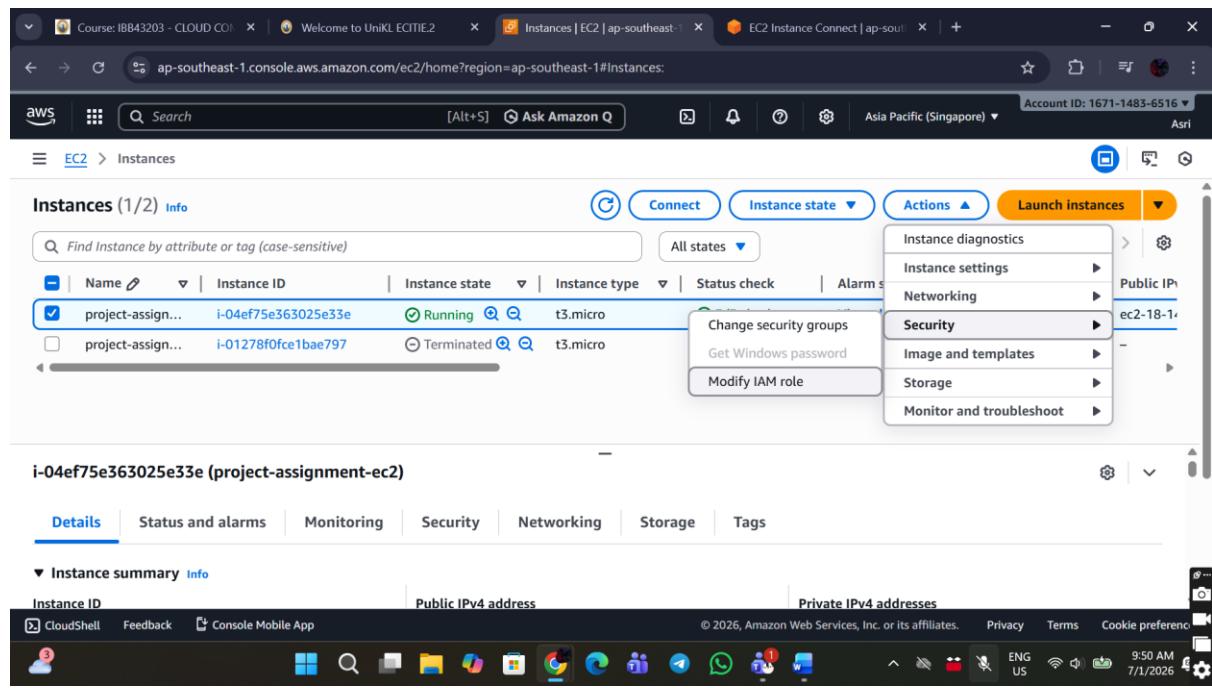
5. Role Details

- Role name: project-assignment-ec2-role
- Click Create role

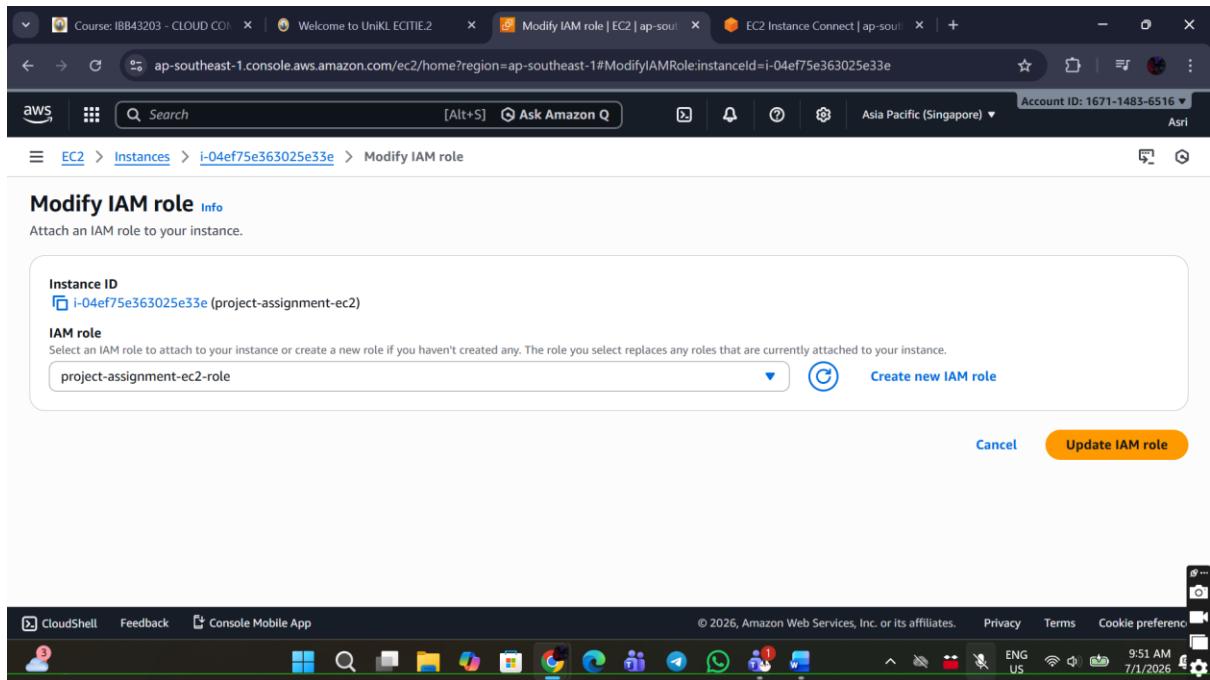


6. Attach Role to EC2

- Go to EC2 → Instances
- Select project-assignment-ec2
- Click Actions → Security → Modify IAM role



- Select project-assignment-ec2-role
- Click Update IAM role



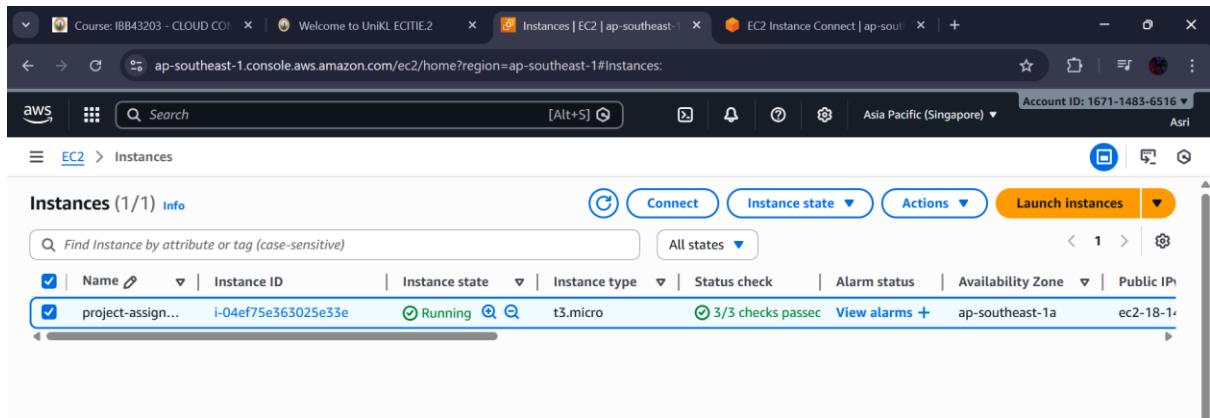
STEP 6: Networking Configuration

1. Use AWS default networking:

- Default VPC
- Public Subnet
- Internet Gateway
- Security Group

STEP 7: Deploy Website Files

1. Confirm EC2 Status, EC2 status must be Running



2. Create the index.html, save.php and view.php files. The codes are at the Github repository and APPENDICES section. Paste the codes into the respective files. Press **ctrl + O** to save the files. Press **ctrl + X** to exit. Run the following commands:

```
sudo nano index.html
```

```
sudo nano save.php
```

```
sudo nano view.php
```

Website directory for the project is:

```
/var/www/html/
```

3. Move files to the website directory

```
sudo mv index.html save.php view.php /var/www/html/
```

```
sudo chown apache:apache /var/www/html/*
```

STEP 8: Install AWS SDK for PHP

1. Install Required Packages

```
sudo yum install php-cli unzip -y
```

2. Install Composer

```
cd ~
```

```
curl -sS https://getcomposer.org/installer | php
```

```
[ec2-user@ip-172-31-34-159 ~]$ cd ~  
[ec2-user@ip-172-31-34-159 ~]$ curl -sS https://getcomposer.org/installer | php  
er | php  
All settings correct for using Composer  
Downloading...  
  
Composer (version 2.9.3) successfully installed to: /home/ec2-user/composer.phar  
Use it: php composer.phar
```

```
sudo mv composer.phar /usr/local/bin/composer
```

```
composer --version
```

```
[ec2-user@ip-172-31-34-159 ~]$ composer --version  
Composer version 2.9.3 2025-12-30 13:40:17  
PHP version 8.4.14 (/usr/bin/php)  
Run the "diagnose" command to get more detailed diagnostics output.  
[ec2-user@ip-172-31-34-159 ~]$ █
```

3. Install AWS SDK

```
cd /var/www/html
```

```
composer require aws/aws-sdk-php
```

4. Restore permissions:

```
sudo chown -R apache:apache /var/www/html
```

```
sudo chmod -R 755 /var/www/html
```

STEP 9: Access the Web Application

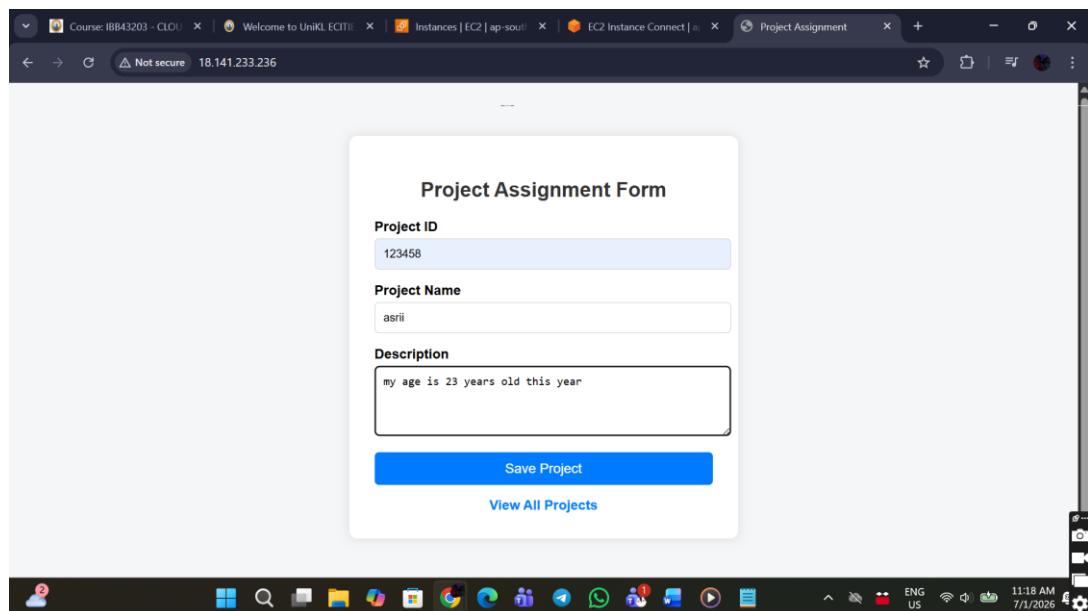
1. Go to EC2 → Instances
2. Copy Public IPv4 address

The screenshot shows the AWS Management Console with the EC2 Instances page open. A single instance, 'project-assign...', is listed as 'Running' with the Public IPv4 address 18.141.233.236. The browser's address bar shows the URL: ap-southeast-1.console.aws.amazon.com/ec2/home?region=ap-southeast-1#Instances:.

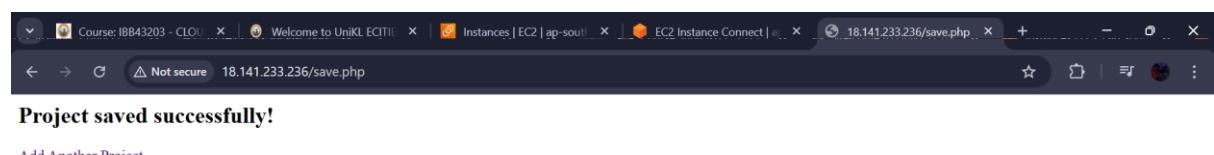
3. Open browser:

The screenshot shows a web browser window with the URL 18.141.233.236. The page displays a 'Project Assignment Form' with fields for 'Project ID', 'Project Name', and 'Description', along with 'Save Project' and 'View All Projects' buttons.

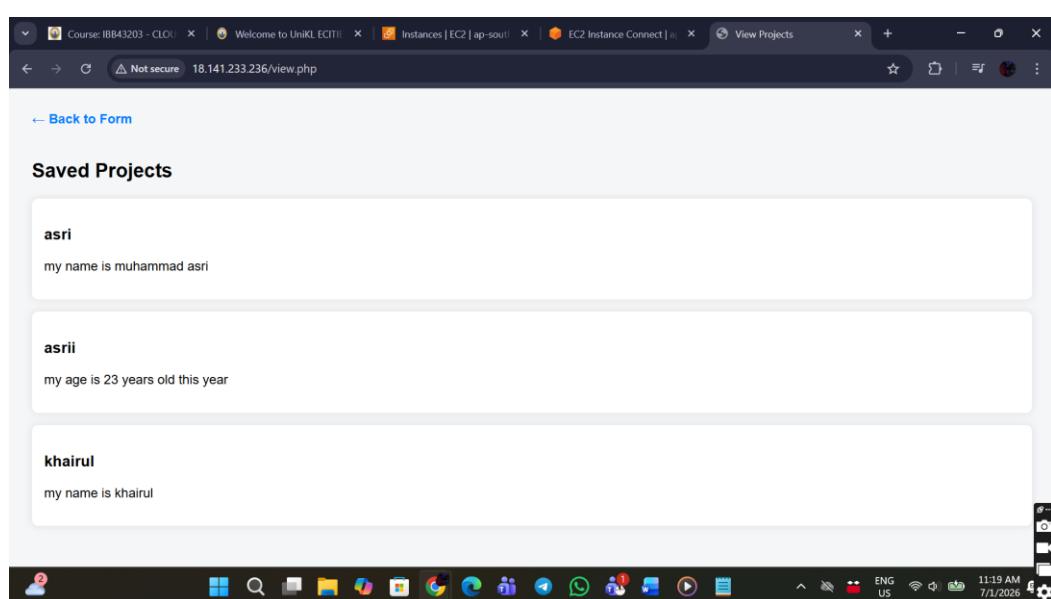
Test the project with project ID, name and description.



save.php page where it displays the project is saved successfully



view.php page where it shows the saved projects



STEP 10: Verify Data in DynamoDB

1. Open DynamoDB
2. Click Tables
3. Select project-assignment-table
4. Click Explore table items
5. Click Scan

The screenshot shows the AWS Lambda console with multiple tabs open. The active tab is 'Items | Amazon DynamoDB'. The URL in the address bar is 'ap-southeast-1.console.aws.amazon.com/dynamodbv2/home?region=ap-southeast-1#item-explorer'. The sidebar on the left shows 'DynamoDB' selected under 'Explore items', with options like Dashboard, Tables, and PartiQL editor. The main content area displays a table titled 'Table: project-assignment-table - Items returned (3)'. The table has columns for ProjectID (String), Description, and ProjectName. The data is as follows:

ProjectID (String)	Description	ProjectName
123456	my name is m...	asri
123458	my age is 23 ...	asrii
123457	my name is k...	khairul

RESULTS AND DISCUSSION

The web application was successfully deployed and accessed via the public IP address of the EC2 instance. Data submitted through the web form was correctly stored and retrieved from DynamoDB. The use of IAM roles ensured secure access without embedding AWS credentials in the code. All components operated within the AWS Free Tier limits.

TEAM CONTRIBUTION

This project was developed collaboratively by all team members. Each member worked together throughout the planning, development, and testing phases to ensure the system was

successfully deployed. Although the project was completed as a team, each member was assigned specific focus areas based on their roles and responsibilities.

Nur Munawwarah Binti Muzamil focused on the setup of the EC2 instance, networking configuration, and DynamoDB configuration. These tasks ensured that the cloud infrastructure and database services were properly created and accessible.

Muhamad Khairul Ikhwan Bin Samsudin was responsible for IAM configuration, web application development, and coding. This role ensured secure access between AWS services and the correct implementation of application functionality.

Muhammad Asri Bin Azizan contributed to coding, system testing, documentation, and GitHub management. This included verifying system functionality, preparing the project report, and managing version control to support team collaboration.

GITHUB AND VIDEO PRESENTATION

- Github repository: <https://github.com/asriazizan930-svg/cloud-computing-project-assignment>
- Youtube link: <https://youtu.be/NS-dTs1NuyI>

CONCLUSION

This project successfully demonstrates the deployment of a web application using AWS cloud services. A secure and scalable solution was achieved by integrating EC2, S3, DynamoDB, IAM, and networking components. The project fulfills all assignment requirements and provides a clear reference for AWS cloud deployment learning.

APPENDICES

```
<!-- index.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Project Assignment</title>




<style>

body {

    font-family: Arial, sans-serif;

    background: #f4f6f8;

    display: flex;

    justify-content: center;

    align-items: center;

    height: 100vh;

}

.container {

    background: white;

    padding: 30px;

    width: 400px;

    border-radius: 10px;

    box-shadow: 0 0 15px rgba(0,0,0,0.1);

}
```

```
}
```

```
h2 {  
    text-align: center;  
    color: #333;  
}
```

```
label {  
    font-weight: bold;  
}
```

```
input, textarea {  
    width: 100%;  
    padding: 10px;  
    margin-top: 5px;  
    margin-bottom: 15px;  
    border: 1px solid #ccc;  
    border-radius: 5px;  
}
```

```
button {  
    width: 100%;  
    background: #007BFF;  
    color: white;
```

```
padding: 10px;  
border: none;  
border-radius: 5px;  
font-size: 16px;  
cursor: pointer;  
}  
  
}
```

```
button:hover {  
background: #0056b3;  
}
```

```
.link {  
text-align: center;  
margin-top: 15px;  
}
```

```
a {  
text-decoration: none;  
color: #007BFF;  
font-weight: bold;  
}
```

```
</style>
```

```
</head>
```

```
<body>

<div class="container">

    <h2>Project Assignment Form</h2>

    <form action="save.php" method="post">

        <label>Project ID</label>
        <input type="text" name="id" required>

        <label>Project Name</label>
        <input type="text" name="name" required>

        <label>Description</label>
        <textarea name="desc" rows="4" required></textarea>

        <button type="submit">Save Project</button>

    </form>

    <div class="link">
        <a href="view.php">View All Projects</a>
    </div>
</div>

</body>
```

</html>

```
// save.php

<?php

ini_set('display_errors', 1);

error_reporting(E_ALL);

require 'vendor/autoload.php';

use Aws\DynamoDb\DynamoDbClient;

if (!isset($_POST['id'], $_POST['name'], $_POST['desc'])) {

    die("Invalid input.");

}

$client = new DynamoDbClient([
    'region' => 'ap-southeast-1',
    'version' => 'latest'
]);

$client->putItem([
    'TableName' => 'project-assignment-table',
    'Item' => [
        'ProjectID' => ['S' => $_POST['id']],
        'ProjectName' => ['S' => $_POST['name']],
        'Description' => ['S' => $_POST['desc']]
    ]
])
```

```
]);  
  
echo "<h2>Project saved successfully!</h2>";  
  
echo "<a href='index.html'>Add Another Project</a><br>";  
  
echo "<a href='view.php'>View Projects</a>";  
  
?>  
  
// view.php  
  
<?php  
  
require 'vendor/autoload.php';  
  
  
use Aws\DynamoDb\DynamoDbClient;  
  
  
  
$client = new DynamoDbClient([  
  
    'region' => 'ap-southeast-1',  
  
    'version' => 'latest'  
  
]);  
  
  
$result = $client->scan([  
  
    'TableName' => 'project-assignment-table'  
  
]);  
  
?>  
  
  
  
<!DOCTYPE html>  
  
<html>  
  
<head>
```

```
<title>View Projects</title>

<style>

body {
    font-family: Arial, sans-serif;
    background: #f4f6f8;
    padding: 20px;
}

.card {
    background: white;
    padding: 15px;
    margin-bottom: 15px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

a {
    display: inline-block;
    margin-bottom: 20px;
    text-decoration: none;
    font-weight: bold;
    color: #007BFF;
}

</style>

</head>
```

```
<body>
```

```
<a href="index.html">← Back to Form</a>
```

```
<h2>Saved Projects</h2>
```

```
<?php
```

```
if (count($result['Items']) === 0) {
```

```
    echo "<p>No projects found.</p>";
```

```
}
```

```
foreach ($result['Items'] as $item) {
```

```
    echo "<div class='card'>";
```

```
    echo "<h3>" . $item['ProjectName'][0] . "</h3>";
```

```
    echo "<p>" . $item['Description'][0] . "</p>";
```

```
    echo "</div>";
```

```
}
```

```
?>
```

```
</body>
```

```
</html>
```