**DEPLOYING MY PHP WEBSITE ON AWS WITH EC2, RDS, AND ROUTE 53**

**Overview**

This document explains how I deployed my PHP website on AWS using **EC2**, **MySQL RDS**, and **Route 53**. I have included every step I took, the problems I faced, and how I solved them. This deployment showcases my ability to work with AWS services efficiently.

1. **Launching EC2 Instance**

**Step 1: Creating EC2 Instance**

1. Logged into **AWS Management Console** → **EC2**.
2. Clicked **Launch Instance** and selected **Ubuntu 22.04**.
3. Chose **t2.micro** (Free Tier eligible) as the instance type.
4. Configured Security Group to allow:
   * **HTTP (Port 80)** for website access.
   * **SSH (Port 22)** for remote access.
5. Launched instance and **noted down the Public IP**.

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.1 : CREATED EC2 INSTANCE**

**Step 2: Connecting to EC2 via MobaXterm**

1. Opened **MobaXterm**.
2. Clicked **Session → SSH**.
3. Entered **Remote Host**: my-ec2-public-ip.
4. Under **Advanced SSH Settings**, selected my **.pem** key.
5. Clicked **OK** and successfully connected to my EC2 instance.

**2️. Installing Apache, PHP, and MySQL**

**Step 1: Installing Apache**

sudo apt update -y

sudo apt install apache2 -y

**Problem Faced: Apache Not Starting**

**Solution:**

sudo systemctl status apache2 # Checked service status

sudo systemctl restart apache2 # Restarted Apache

A screenshot of a computer program

AI-generated content may be incorrect.

**FIG.2 : INSTALLED APACHE**

**Step 2: Installing PHP and Required Modules**

sudo apt install php libapache2-mod-php php-mysql -y

**Step 3: Installing MySQL Client**

sudo apt install mysql-client -y

**3️. Setting Up MySQL RDS Database**

**Step 1: Creating RDS Instance**

1. Open AWS RDS Console → Create Database.
2. Select MySQL and Free Tier options.
3. Set DB Identifier, Username, and Password.
4. Enable Public Access.
5. Modify Security Group to allow MySQL (Port 3306).
6. Wait for the database to become available.

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.3 : CREATED RDS**

**Step 2: Creating Database**

CREATE DATABASE snabinterns\_db;

GRANT ALL PRIVILEGES ON snabinterns\_db.\* TO 'admin'@'%' IDENTIFIED BY 'my-password';

FLUSH PRIVILEGES;

**Step 3: Creating the Users Table**

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL,

email VARCHAR(100) NOT NULL,

password VARCHAR(255) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**Step 4: Viewing Output in MySQL Workbench on Windows**

1. Opened **MySQL Workbench**.
2. Clicked **New Connection**.
3. Entered **RDS Endpoint** and credentials.
4. Clicked **Test Connection** → Successfully connected.

**Problem Faced: RDS Connection Issue**

**Solution:**

* Allowed **Port 3306** in **RDS Security Group**.
* Verified my **username, password, and endpoint**.

**4️. Deploying My PHP Website**

**Step 1: Uploading PHP Files to EC2**

Migrated the PHP files from Windows to EC2 using the SCP command in Windows Command Prompt

scp -i my-key.pem C:\path\to\mywebsite\\* ubuntu@my-ec2-public-ip:/var/www/html/

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.3 :** **MIGRATED THE PHP FILES FROM WINDOWS TO EC2**

**Step 2: Changing Ownership and Permissions**

sudo chown -R www-data:www-data /var/www/html/

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

**Step 3: Editing config.php to Connect to RDS**

$host = "my-rds-endpoint";

$username = "admin";

$password = "my-password";

$database = "snabinterns\_db";

$conn = new mysqli($host, $username, $password, $database);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.4: EDITING CONFIG.PHP TO CONNECT TO RDS**

**Problem Faced: Database Connection Error**

**Solution:**

* Allowed **EC2 IP** in **RDS Security Group**.
* Double-checked credentials in config.php.

**5️. Configuring Route 53 to Point Domain to EC2**

**Step 1: Creating a Hosted Zone in Route 53**

1. Opened **Route 53** → **Hosted Zones** → **Create Hosted Zone**.
2. Entered **Domain Name** (snabinterns.info) and clicked **Create**.

**Step 2: Adding an A Record**

1. Clicked **Create Record**.
2. Set **Type:** A Record.
3. Entered **Value:** My EC2 Public IP.
4. Clicked **Create Record**.

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.5 : CREATED HOSTED ZONE AND ADDED RECORD**

**Step 3: Updating Name Servers in Domain Registrar**

* Copied **NS records** from Route 53.
* Updated them in my **domain provider settings**.
* Waited **5 mins to 24 hours** for DNS propagation.

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.6 : UPDATING NAME SERVERS IN GODADDY**

**Problem Faced: Website Not Resolving**

**Solution:**

nslookup snabinterns.info

dig snabinterns.info +short

Ensured **A Record** was correctly pointing to EC2 Public IP.

**6️.Troubleshooting Issues I Faced**

**Issue 1: Default Apache Page Showing**

**Fix:**

sudo rm /var/www/html/index.html

sudo systemctl restart apache2

**Issue 2: Permission Denied for Uploaded Files**

**Fix:**

sudo chown -R www-data:www-data /var/www/html/

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

**Issue 3: DNS Propagation Delay**

**Fix:**

nslookup snabinterns.info

dig snabinterns.info +short

**Issue 4: Database Connection Failing**

**Fix:**

* Allowed EC2 **Port 3306** in **RDS Security Group**.
* Verified credentials in **config.php**.

**7️. Result**

After successfully completing all the steps, my PHP website is now live on AWS. The EC2 instance is hosting the Apache server with PHP, and the RDS MySQL database is securely storing the user data. Through Route 53, the domain **snabinterns.info** is pointing to the EC2 instance, allowing users to access the website globally.

When users register on the website, their details are successfully inserted into the users table in the RDS database. I can view this data in MySQL Workbench, ensuring smooth functionality for user registration and database management.

This deployment demonstrates my ability to effectively utilize AWS services like EC2, RDS, and Route 53 to build and host a fully functional website. It also highlights my capability in troubleshooting issues such as DNS propagation delays, file permissions, and database connectivity.

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.7 : DOMAIN POINTING OUR REGISTARTION PAGE**

A screenshot of a computer

AI-generated content may be incorrect.

**FIG.8 : USER DETAILS ARE STORED IN RDS**

**8. Conclusion**

I successfully deployed my **PHP website** on AWS using **EC2, RDS, and Route 53**. I troubleshooted various issues, ensuring a smooth deployment. This project demonstrates my practical knowledge of AWS services and cloud-based hosting