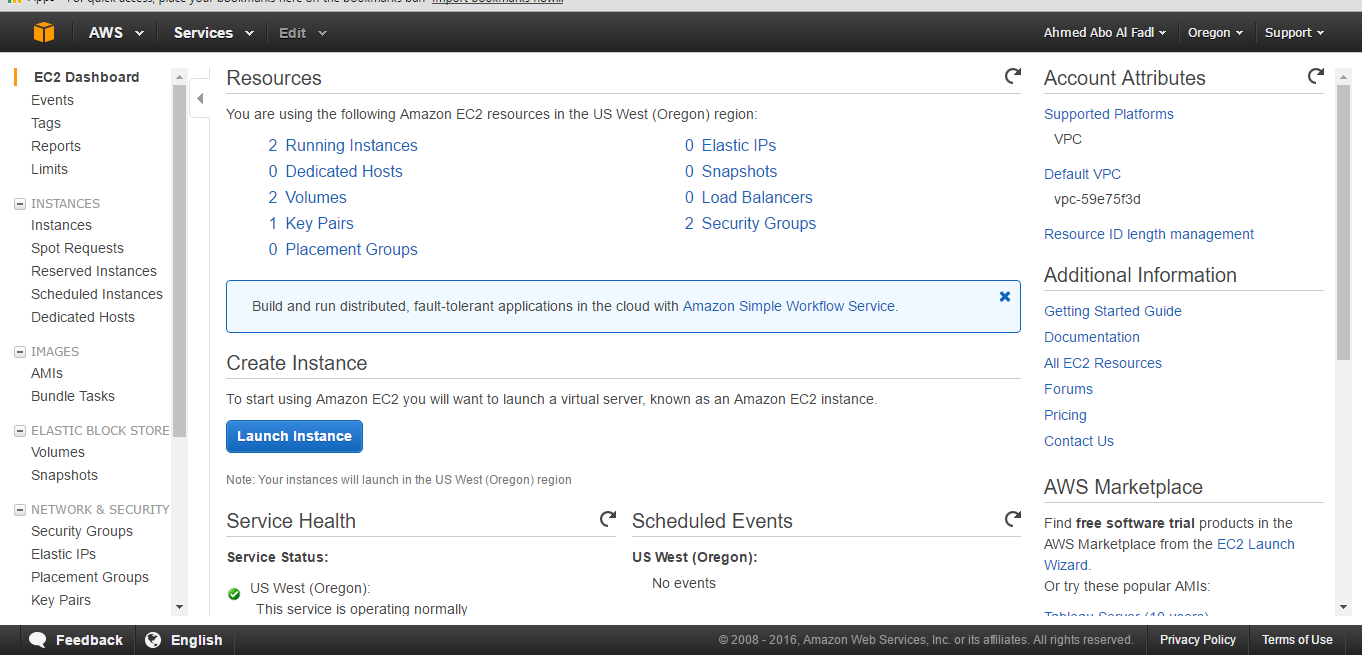
|  |
| --- |
|  |

|  |
| --- |
| **Project 1,2** |
| Cloud Computing Course Project |
|  |
|  |
| **Dr : Mohamed Shaheen**  **TA : Eng. Ahmed Saiid**  **Students : Amira Mahmoud**  **Ahmed Abo Alfadl** |

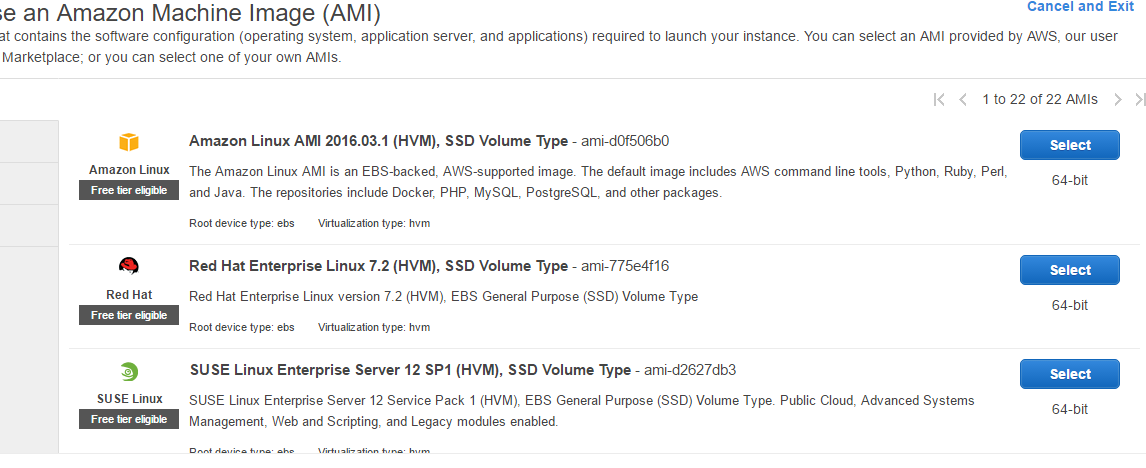
***Project 1 steps:***

First Create EC2 instance

Services ->EC2



Click on launch instance

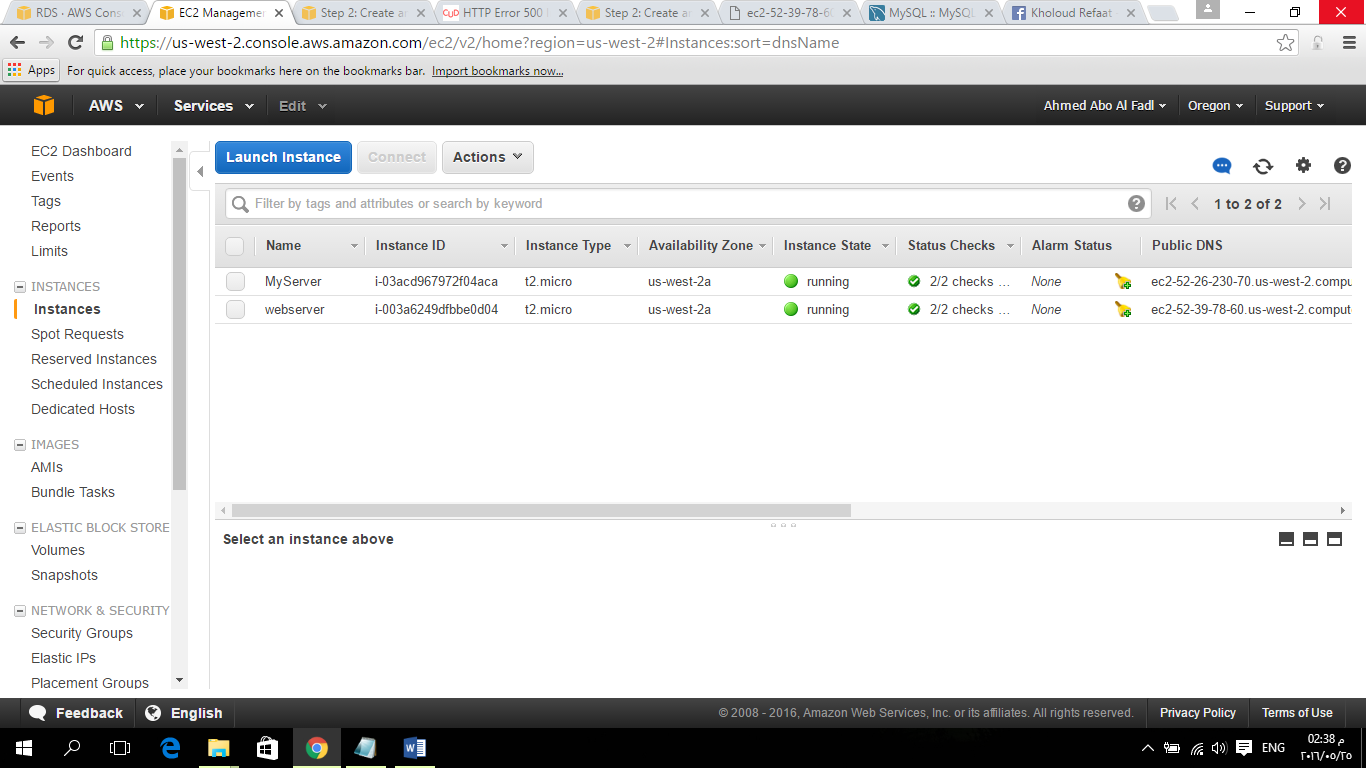


Choose the instance that you want to use we chose **Ubuntu Server 14.04 LTS (HVM), SSD Volume Type**

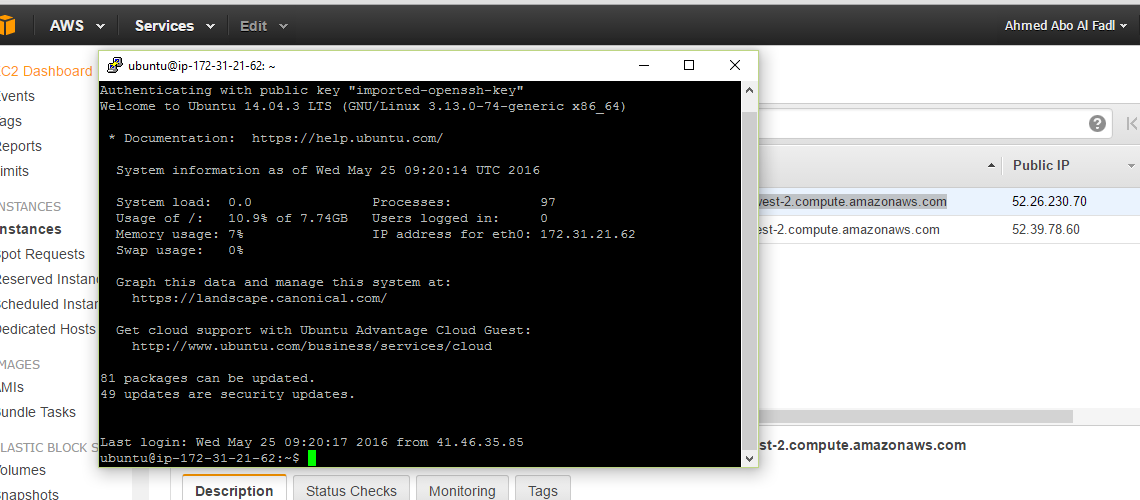
Choose the specifications that you need and click launch instance

Wait until the status change from initialization to 2/2 checks.

The instance MyServer is Ubuntu server but webserver is linux and it's part of project2



No Connect to Putty



***Project 2 steps:***

***Step 1: Create an RDS DB Instance***

In this step you create an Amazon RDS MySQL DB instance that maintains the data used by a web application.

1. Sign in to the AWS Management Console and open the Amazon RDS console at<https://console.aws.amazon.com/rds/>.
2. In the top-right corner of the AWS Management Console, choose the region in which you want to create the DB instance. This example uses the US West (Oregon) region.
3. Choose **Instances**.
4. Choose **Launch DB Instance**.
5. On the **Select Engine** page, shown following, choose the MySQL DB engine, and then choose **Select**.
6. On the Production page, below Dev/Test, choose MySQL This instance is intended for use outside of production, and then choose Next Step.
7. On the Specify DB Details page, shown following, set these values:

DB Engine Version: Use the default value.

DB Instance Class: db.t2.micro

Multi-AZ Deployment: No

Storage Type: General purpose

Allocated Storage: 5 GB

DB Instance Identifier: MyDatabase

Master Username: AboAlfadl

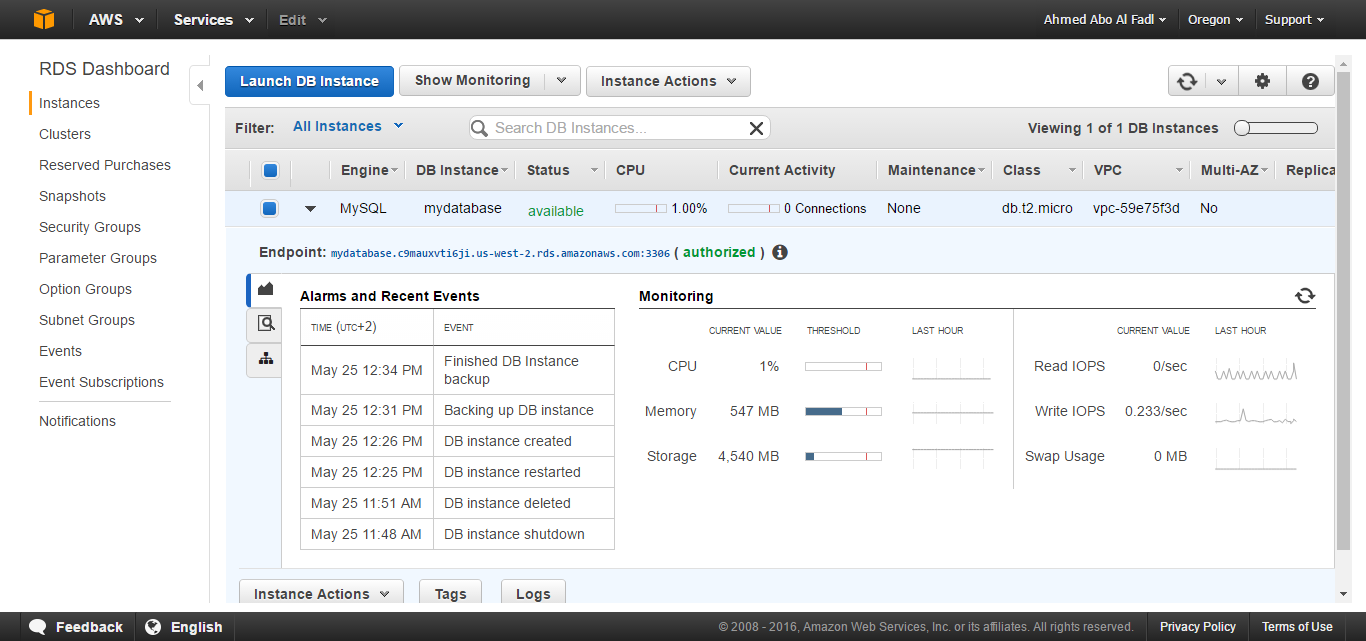
Master Password: Choose a password.

Confirm Password: Retype the password.

1. Choose **Next Step** and set the following values in the **Configure Advanced Settings**page, shown following:

* **VPC:** default
* **Subnet group:** default
* **Publicly Accessible:** Yes
* **Availability Zone:** No Preference
* **VPC Security Group(s):** MySG (the default)
* **Database Name:** MyDatabase

1. To create your Amazon RDS MySQL DB instance, choose Launch DB Instance.
2. On the next page, choose View Your DB Instances to view your RDS MySQL DB instance.
3. Wait for the status of your new DB instance to show as available. Then choose the selection box to the left of your DB instance to display the DB instance details.



***Step 2: Create web server and connection***

The steps of creating the server is like project1 but the instance used is Amazon Machine Image (AMI)

**Install an Apache web server with PHP**

Next you connect to your EC2 instance and install the web server.

**To connect to your EC2 instance and install the Apache web server with PHP**

1. To connect to the EC2 instance that you created earlier.
2. To get the latest bug fixes and security updates, update the software on your EC2 instance by using the following command:

[ec2-user ~]$ **sudo yum update –y**

1. After the updates complete, install the Apache web server with the PHP software package using the **yum install** command, which installs multiple software packages and related dependencies at the same time:
2. [ec2-user ~]$ **sudo yum install -y httpd24 php56 php56-mysqlnd**

1. Start the web server with the command shown following:
2. [ec2-user ~]$ **sudo service httpd start**

1. Configure the web server to start with each system boot using the **chkconfig**command:
2. [ec2-user ~]$ **sudo chkconfig httpd on**

To allow ec2-user to manage files in the default root directory for your Apache web server, you need to modify the ownership and permissions of the /var/www directory. In this tutorial, you add a group named www to your EC2 instance, and then you give that group ownership of the /var/www directory and add write permissions for the group. Any members of that group can then add, delete, and modify files for the web server.

**To set file permissions for the Apache web server**

1. Add the www group to your EC2 instance with the following command:
2. [ec2-user ~]$ **sudo groupadd www**

1. Add the ec2-user user to the www group:
2. [ec2-user ~]$ **sudo usermod -a -G www ec2-user**

1. To refresh your permissions and include the new www group, log out:
2. [ec2-user ~]$ **exit**

1. Log back in again and verify that the www group exists with the groups command:
2. [ec2-user ~]$ **groups**
3. ec2-user wheel www

1. Change the group ownership of the /var/www directory and its contents to the wwwgroup:
2. [ec2-user ~]$ **sudo chown -R root:www /var/www**

1. Change the directory permissions of /var/www and its subdirectories to add group write permissions and set the group ID on subdirectories created in the future:
2. [ec2-user ~]$ **sudo chmod 2775 /var/www**
3. [ec2-user ~]$ **find /var/www -type d -exec sudo chmod 2775 {} +**

1. Recursively change the permissions for files in the /var/www directory and its subdirectories to add group write permissions:
2. [ec2-user ~]$ **find /var/www -type f -exec sudo chmod 0664 {} +**

**Connect your Apache web server to your RDS DB instance**

Next, you add content to your Apache web server that connects to your Amazon RDS DB instance.

**To add content to the Apache web server that connects to your RDS DB instance**

1. While still connected to your EC2 instance, change the directory to /var/www and create a new subdirectory named inc:
2. [ec2-user ~]$ **cd /var/www**
3. [ec2-user ~]$ **mkdir inc**
4. [ec2-user ~]$ **cd inc**

1. Create a new file in the inc directory named dbinfo.inc, and then edit the file by calling nano (or the editor of your choice).
2. [ec2-user ~]$ **>dbinfo.inc**
3. [ec2-user ~]$ **nano dbinfo.inc**

1. Add the following contents to the dbinfo.inc file, where *endpoint* is the endpoint of your RDS MySQL DB instance, without the port, and *master password* is the master password for your RDS MySQL DB instance.

**Note**

Placing the user name and password information in a folder that is not part of the document root for your web server reduces the possibility of your security information being exposed.

<?php

define('DB\_SERVER', '*endpoint*');

define('DB\_USERNAME', 'tutorial\_user');

define('DB\_PASSWORD', '*master password*');

define('DB\_DATABASE', 'sample');

?>

1. Save and close the dbinfo.inc file.
2. Change the directory to /var/www/html:
3. [ec2-user ~]$ **cd /var/www/html**

1. Create a new file in the html directory named SamplePage.php, and then edit the file by calling nano (or the editor of your choice).
2. [ec2-user ~]$ **>SamplePage.php**
3. [ec2-user ~]$ **nano SamplePage.php**

1. Add the following contents to the SamplePage.php file:

**Note**

Placing the user name and password information in a folder that is not part of the document root for your web server reduces the possibility of your security information being exposed.

<?php include "../inc/dbinfo.inc"; ?>

<html>

<body>

<h1>Sample page</h1>

<?php

/\* Connect to MySQL and select the database. \*/

$connection = mysqli\_connect(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD);

if (mysqli\_connect\_errno()) echo "Failed to connect to MySQL: " . mysqli\_connect\_error();

$database = mysqli\_select\_db($connection, DB\_DATABASE);

/\* Ensure that the Employees table exists. \*/

VerifyEmployeesTable($connection, DB\_DATABASE);

/\* If input fields are populated, add a row to the Employees table. \*/

$employee\_name = htmlentities($\_POST['Name']);

$employee\_address = htmlentities($\_POST['Address']);

if (strlen($employee\_name) || strlen($employee\_address)) {

AddEmployee($connection, $employee\_name, $employee\_address);

}

?>

<!-- Input form -->

<form action="<?PHP echo $\_SERVER['SCRIPT\_NAME'] ?>" method="POST">

<table border="0">

<tr>

<td>Name</td>

<td>Address</td>

</tr>

<tr>

<td>

<input type="text" name="Name" maxlength="45" size="30" />

</td>

<td>

<input type="text" name="Address" maxlength="90" size="60" />

</td>

<td>

<input type="submit" value="Add Data" />

</td>

</tr>

</table>

</form>

<!-- Display table data. -->

<table border="1" cellpadding="2" cellspacing="2">

<tr>

<td>ID</td>

<td>Name</td>

<td>Address</td>

</tr>

<?php

$result = mysqli\_query($connection, "SELECT \* FROM Employees");

while($query\_data = mysqli\_fetch\_row($result)) {

echo "<tr>";

echo "<td>",$query\_data[0], "</td>",

"<td>",$query\_data[1], "</td>",

"<td>",$query\_data[2], "</td>";

echo "</tr>";

}

?>

</table>

<!-- Clean up. -->

<?php

mysqli\_free\_result($result);

mysqli\_close($connection);

?>

</body>

</html>

<?php

/\* Add an employee to the table. \*/

function AddEmployee($connection, $name, $address) {

$n = mysqli\_real\_escape\_string($connection, $name);

$a = mysqli\_real\_escape\_string($connection, $address);

$query = "INSERT INTO `Employees` (`Name`, `Address`) VALUES ('$n', '$a');";

if(!mysqli\_query($connection, $query)) echo("<p>Error adding employee data.</p>");

}

/\* Check whether the table exists and, if not, create it. \*/

function VerifyEmployeesTable($connection, $dbName) {

if(!TableExists("Employees", $connection, $dbName))

{

$query = "CREATE TABLE `Employees` (

`ID` int(11) NOT NULL AUTO\_INCREMENT,

`Name` varchar(45) DEFAULT NULL,

`Address` varchar(90) DEFAULT NULL,

PRIMARY KEY (`ID`),

UNIQUE KEY `ID\_UNIQUE` (`ID`)

) ENGINE=InnoDB AUTO\_INCREMENT=1 DEFAULT CHARSET=latin1";

if(!mysqli\_query($connection, $query)) echo("<p>Error creating table.</p>");

}

}

/\* Check for the existence of a table. \*/

function TableExists($tableName, $connection, $dbName) {

$t = mysqli\_real\_escape\_string($connection, $tableName);

$d = mysqli\_real\_escape\_string($connection, $dbName);

$checktable = mysqli\_query($connection,

"SELECT TABLE\_NAME FROM information\_schema.TABLES WHERE TABLE\_NAME = '$t' AND TABLE\_SCHEMA = '$d'");

if(mysqli\_num\_rows($checktable) > 0) return true;

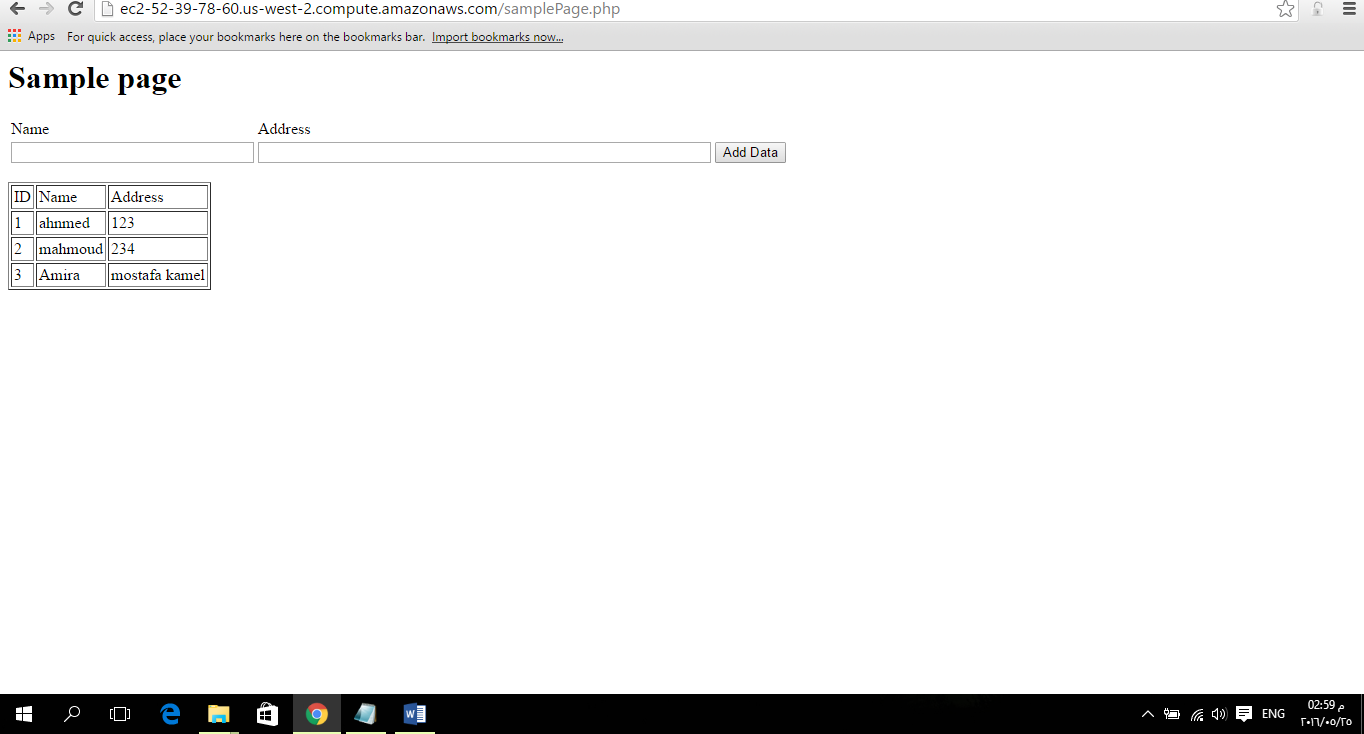
return false;

}

?>

1. Save and close the SamplePage.php file.
2. Verify that your web server successfully connects to your RDS MySQL DB instance by opening a web browser and browsing to http://*EC2 instance endpoint*/SamplePage.php, for example: http://ec2-55-122-41-31.us-west-2.compute.amazonaws.com/SamplePage.php.

You can use SamplePage.php to add data to your RDS MySQL DB instance. The data that you add is then displayed on the page.



http://ec2-52-39-78-60.us-west-2.compute.amazonaws.com/samplePage.php