

AWS Solutions Architect Project

Name: Vlasits Pitsios

Email: vlasits.pitsios@vodafone.com

Company: Vodafone

Main project (Description): You are working as a database administrator for an IT firm. You have been asked to create a new database instance on AWS cloud and connect it with the employee management portal hosted on a web server. Your organization wants to deploy a new multi-tier application. The application will take live inputs from the employees and it will be hosted on a web server running on the AWS cloud. The development team has asked you to set up the web server and configure it to scale automatically in cases of a traffic surge, to make the application highly available. They have also asked you to take the inputs from the employees and store them securely in the database.

Following requirements should be met:

- Follow the above-mentioned specifications
- Make sure that the Availability Zone is similar throughout the instances and volumes
- Ensure that the server scales automatically and the traffic is optimally routed among the scaled servers
- Document the step-by-step process involved in completing this task

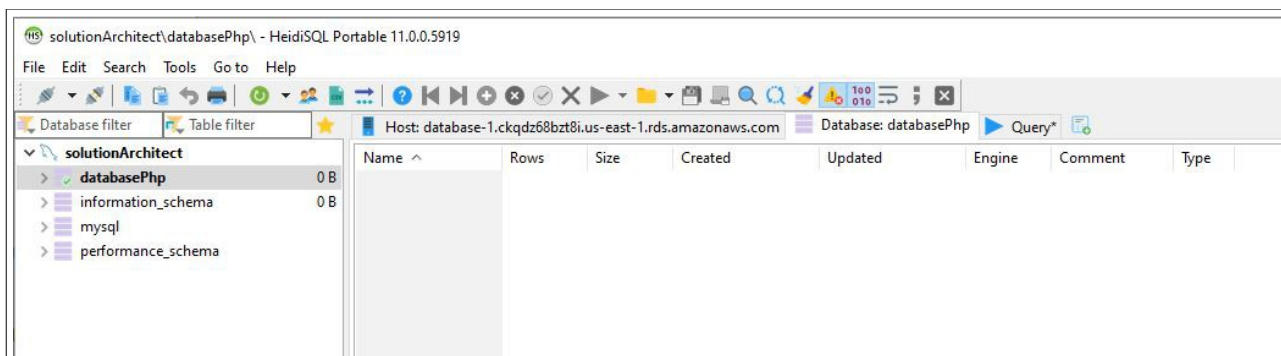
The screenshot displays the AWS Management Console interface for an Amazon RDS database instance. The top navigation bar includes the AWS logo, a search bar, and user information. The left sidebar shows the 'Amazon RDS' menu with various options like Dashboard, Databases, Query Editor, etc. The main content area shows the details for 'database-1' under the 'Databases' section. A 'Summary' table provides key information about the instance, including its DB identifier, CPU, status (Available), class (db.t2.micro), role (Instance), current activity, engine (MySQL Community), and region & AZ (us-east-1b). Below the summary, there are tabs for 'Connectivity & security', 'Monitoring', 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. The 'Connectivity & security' tab is active, showing details for 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' section shows the endpoint 'database-1.ckqd268bzt8i.us-east-1.rds.amazonaws.com' and port '3306'. The 'Networking' section shows the availability zone 'us-east-1b', VPC 'vpc-526d952f', subnet group 'default-vpc-526d952f', and subnets 'subnet-302dde01', 'subnet-c8532c85', and 'subnet-9e1c7d90'. The 'Security' section shows the VPC security groups 'default (sg-6ec5de52) (active)' and 'Public accessibility' set to 'Yes'. The 'Certificate authority' is 'rds-ca-2019' and the 'Certificate authority date' is 'Aug 22nd, 2024'.

Summary			
DB identifier database-1	CPU -	Status Available	Class db.t2.micro
Role Instance	Current activity	Engine MySQL Community	Region & AZ us-east-1b

Connectivity & security		
Endpoint & port	Networking	Security
Endpoint database-1.ckqd268bzt8i.us-east-1.rds.amazonaws.com	Availability zone us-east-1b	VPC security groups default (sg-6ec5de52) (active)
Port 3306	VPC vpc-526d952f	Public accessibility Yes
	Subnet group default-vpc-526d952f	Certificate authority rds-ca-2019
	Subnets subnet-302dde01 subnet-c8532c85 subnet-9e1c7d90	Certificate authority date Aug 22nd, 2024

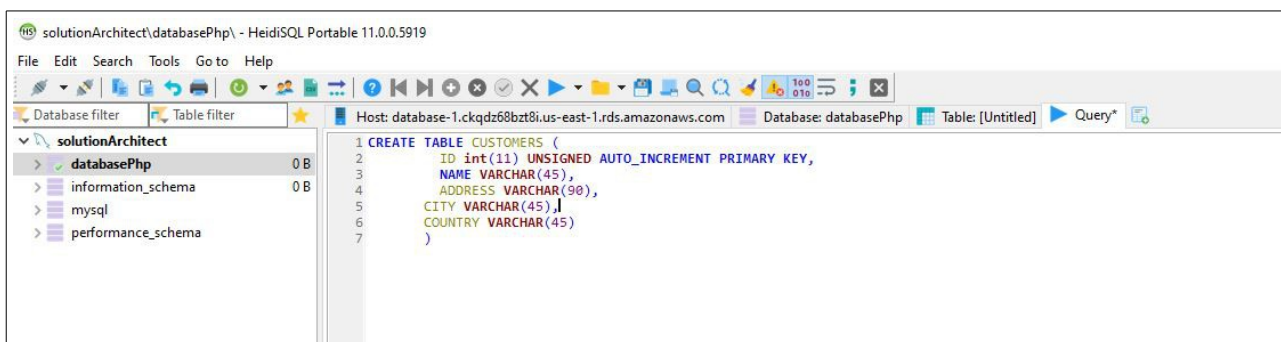
Screenshot 1

MySQL database created using Amazon RDS.



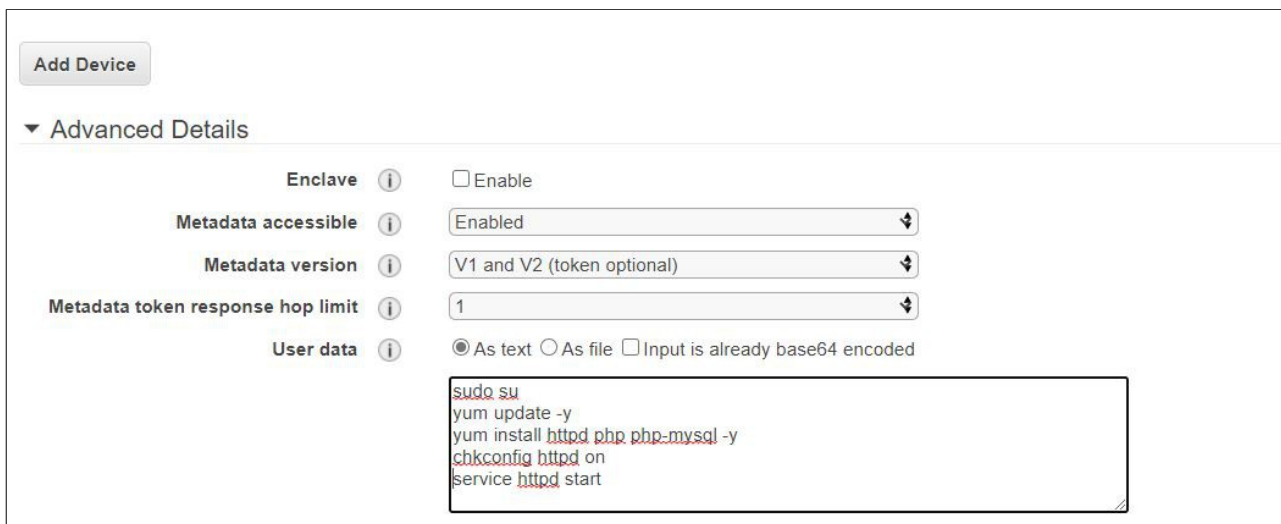
Screenshot 2

Connection established to MySQL by using an SQL client. Database "databasePhp" was created successfully.



Screenshot 3

Create CUSTOMERS table on "databasePhp" database.



Screenshot 4

Create two EC2 instances on different availability zones and run the commands in the screenshot in order to install php, php-mysql and start an apache server.

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Corestack_Role/vlasipitsios_vodafone @ 6847-4222-3434

N. Virginia

Support

1. Choose AMI

2. Choose Instance Type

3. Configure Instance

4. Add Storage

5. Add Tags

6. Configure Security Group

7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more about Amazon EC2 security groups.](#)

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP 5.54.212.237/32	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

Screenshot 5

Configure security groups on each EC2 instance. HTTP traffic must be permitted from anywhere.

aws

Services

Search for services, features, marketplace products, and docs

[Alt+S]

Corestack_Role/vlasipitsios_vodafone @ 6847-4222-3434

N. Virginia

Support

New EC2 Experience
Tell us what you think

EC2 Dashboard

Events

Tags

Limits

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Welcome to the new Instances experience!
We're redesigning the EC2 console to make it easier to use. To switch between the old console and the new console, use the New EC2 Experience toggle above the navigation panel. We'll release updates continuously based on customer feedback.

Instances (1/2) Info

Filter instances

Connect

Instance state

Actions

Launch instances

1

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
<input type="checkbox"/>	machine1	i-0645fbc2feab9d103	Running	t2.micro	Initializing	No alarms	us-east-1a	ec2-3-231-26-56.com...	3.231.26.56
<input checked="" type="checkbox"/>	machine2	i-05b50c430f06d5806	Running	t2.micro	-	No alarms	us-east-1b	ec2-3-85-43-54.comp...	3.85.43.54

Screenshot 6

The two EC2 instances (machine1 and machine2) are running on different availability zones (us-east-1a and us-east-1b).

ec2-user@ip-172-31-2-88-

login as: ec2-user

Authenticating with public key "imported-openssh-key"

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/

2 package(s) needed for security, out of 5 available

Run "sudo yum update" to apply all updates.

ec2-user@ip-172-31-2-88 ~\$

ec2-user@ip-172-31-81-144-

login as: ec2-user

Authenticating with public key "imported-openssh-key"

Amazon Linux 2 AMI

https://aws.amazon.com/amazon-linux-2/

2 package(s) needed for security, out of 5 available

Run "sudo yum update" to apply all updates.

ec2-user@ip-172-31-81-144 ~\$

Screenshot 7

Connect to EC2 instances (machine1 and machine2) with Putty.

```
root@ip-172-31-2-88:/var/www/html
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]#
[root@ip-172-31-2-88 www]# pwd
/var/www
[root@ip-172-31-2-88 www]# cd html/
[root@ip-172-31-2-88 html]# pwd
/var/www/html
[root@ip-172-31-2-88 html]#

root@ip-172-31-81-144:/var/www/html
login as: ec2-user
Authenticating with public key "imported-openssh-key"

 _ _ _ _ _
| | | | | / Amazon Linux 2 AMI
|_|_|_|_|_|

https://aws.amazon.com/amazon-linux-2/
2 package(s) needed for security, out of 5 available
Run "sudo yum update" to apply all updates.
ec2-user@ip-172-31-81-144 ~]$ cd /var/
ec2-user@ip-172-31-81-144 var]$ sudo -s
root@ip-172-31-81-144 var]# mkdir www
root@ip-172-31-81-144 var]# cd www
root@ip-172-31-81-144 www]# mkdir html
root@ip-172-31-81-144 www]# ls
html
root@ip-172-31-81-144 www]# cd html/
root@ip-172-31-81-144 html]# ls
root@ip-172-31-81-144 html]# pwd
/var/www/html
root@ip-172-31-81-144 html]#
```

Screenshot 8
Create folder "/var/www/html" on both EC2 instances (machine1 and machine2).

```

root@ip-172-31-2-88:/var/www/html
[root@ip-172-31-2-88 html]# cat index.php
<?php include "dbinfo.inc"; ?>
<html>
<body>

<?php
$instance_id = file_get_contents("http://instance-data/latest/meta-data/instance-id");
echo "<h1>Information about customers from ",$instance_id, "</h1>";

/* 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();

$dbdatabase = mysqli_select_db($connection, DB_DATABASE);

/* Ensure that the EMPLOYEES table exists. */
verifyCustomersTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$customer_name = htmlentities($ _POST['NAME']);
$customer_address = htmlentities($ _POST['ADDRESS']);
$customer_city = htmlentities($ _POST['CITY']);
$customer_country = htmlentities($ _POST['COUNTRY']);

if (strlen($customer_name) || strlen($customer_address) || strlen($customer_city) || strlen($customer_country) {
    addCustomer($connection, $customer_name, $customer_address, $customer_city, $customer_country);
}
?>

<!-- Input form -->
<form action="<?PHP echo $ _SERVER['SCRIPT_NAME'] ?>" method="POST">
    <table border="0">
        <tr>
            <td>NAME</td>
            <td>ADDRESS</td>
            <td>CITY</td>
            <td>COUNTRY</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="text" name="CITY" maxlength="90" size="60" />
            </td>
            <td>
                <input type="text" name="COUNTRY" 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>

```

Screenshot 9

Create on path "/var/www/html" the file "index.php" with the following PHP code.

Below there is the PHP code which is located in path "/var/www/html". The "index.php" file is a simple form to add some information about a new customer, for example customer name, customer address, customer city and customer country. The PHP script reads the inserted data from the user information and insert the data on the MySQL database. When the insertion of the data is done, the script runs a select query to the database on table CUSTOMERS and shows the result as a table. Also there is one more PHP script named as "dbinfo.inc" which contains information about the database to connect to, for example the database url, database username, database password and database name.

index.php

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<?php
$Instance_id = file_get_contents("http://instance-data/latest/meta-data/instance-id");
echo "<h1>Information about customers from ", $Instance_id, "</h1>";
/* 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();
$dbname = mysqli_select_db($connection, DB_DATABASE);
/* Ensure that the EMPLOYEES table exists. */
verifyCustomersTable($connection, DB_DATABASE);
/* If input fields are populated, add a row to the EMPLOYEES table. */
$customer_name = htmlentities($_POST['NAME']);
$customer_address = htmlentities($_POST['ADDRESS']);
$customer_city = htmlentities($_POST['CITY']);
$customer_country = htmlentities($_POST['COUNTRY']);
if (strlen($customer_name) || strlen($customer_address) || strlen($customer_city) || strlen($customer_country)) {
    addCustomer($connection, $customer_name, $customer_address, $customer_city, $customer_country);
}
?>
<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
<table border="0">
    <tr>
        <td>NAME</td>
        <td>ADDRESS</td>
        <td>CITY</td>
        <td>COUNTRY</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="text" name="CITY" maxlength="90" size="60" />
        </td>
        <td>
            <input type="text" name="COUNTRY" 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>
        <td>CITY</td>
        <td>COUNTRY</td>
    </tr>
<?php
$result = mysqli_query($connection, "SELECT * FROM CUSTOMERS");
while($query_data = mysqli_fetch_row($result)) {
    echo "<tr>";
    echo "<td>",$query_data[0], "</td>";
    echo "<td>",$query_data[1], "</td>";
    echo "<td>",$query_data[2], "</td>";
    echo "<td>",$query_data[3], "</td>";
    echo "<td>",$query_data[4], "</td>";
    echo "</tr>";
}
?>
</table>
<!-- Clean up. -->
<?php
mysqli_free_result($result);
mysqli_close($connection);
?>
</body>
</html>
<?php
/* Add an customer to the table. */
function addCustomer($connection, $name, $address, $city, $country) {
    $n = mysqli_real_escape_string($connection, $name);
    $a = mysqli_real_escape_string($connection, $address);
    $b = mysqli_real_escape_string($connection, $city);
    $c = mysqli_real_escape_string($connection, $country);
    $query = "INSERT INTO CUSTOMERS (NAME, ADDRESS, CITY, COUNTRY) VALUES ('$n', '$a', '$b', '$c');";
    if (mysqli_query($connection, $query)) echo("<p>Error adding customer data.</p>");
}
/* Check whether the table exists and, if not, create it. */
function verifyCustomersTable($connection, $dbName) {
    if (!TableExists("CUSTOMERS", $connection, $dbName))
    {
        $query = "CREATE TABLE CUSTOMERS (
            ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            NAME VARCHAR(45),
            ADDRESS VARCHAR(90),
            CITY VARCHAR(45),
            COUNTRY VARCHAR(45)
        )";
    }
}
```

```

    if(mysql_query($connection, $query)) echo("<p>Error creating table.</p>");
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
    $t = mysql_real_escape_string($connection, $tableName);
    $d = mysql_real_escape_string($connection, $dbName);
    $checktable = mysql_query($connection,
        "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t' AND TABLE_SCHEMA = '$d'");
    if(mysql_num_rows($checktable) > 0) return true;
    return false;
}
?>

```

dbinfo.php

```

<?php
define('DB_SERVER', 'database-2.ckqdz68bzt8i.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'admin123');
define('DB_DATABASE', 'databasePhp');
?>

```

```

root@ip-172-31-2-88:/var/www/html
$checktable = mysql_query($connection,
    "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t' AND TABLE_SCHEMA = '$d'");

if(mysql_num_rows($checktable) > 0) return true;

return false;

?>

root@ip-172-31-2-88 html#
root@ip-172-31-2-88 html# nano dbinfo.inc
root@ip-172-31-2-88 html# cat dbinfo.inc
<?php

define('DB_SERVER', 'database-1.ckqdz68bzt8i.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'admin123');
define('DB_DATABASE', 'databasePhp');

?>

root@ip-172-31-2-88 html# ls
dbinfo.inc index.php
root@ip-172-31-2-88 html#

root@ip-172-31-81-144:/var/www/html
root@ip-172-31-81-144 html#
root@ip-172-31-81-144 html#
root@ip-172-31-81-144 html#
root@ip-172-31-81-144 html#
root@ip-172-31-81-144 html# ls
dbinfo.inc index.php
root@ip-172-31-81-144 html# cat dbinfo.inc
<?php

define('DB_SERVER', 'database-1.ckqdz68bzt8i.us-east-1.rds.amazonaws.com');
define('DB_USERNAME', 'admin');
define('DB_PASSWORD', 'admin123');
define('DB_DATABASE', 'databasePhp');

?>

root@ip-172-31-81-144 html# ls
dbinfo.inc index.php
root@ip-172-31-81-144 html#

```

Screenshot 10

The two PHP (index.php and dbinfo.inc) scripts are available on path "/var/www/html". In a next step the dbinfo.inc will be moved to path "/var/www/inc".

← → ↻ ⚠ Not secure | 3.231.26.56/index.php

Information about customers from i-0645fbc2feab9d103

Failed to connect to MySQL: Server sent charset unknown to the client. Please, report to the developers

Error creating table.

NAME ADDRESS CITY COUNTRY Add Data

ID	NAME	ADDRESS	CITY	COUNTRY
----	------	---------	------	---------

Screenshot 11

Request to first EC2 instance (machine1). The user should add some information about a customer and the data will be stored in MySQL.

← → ↻ ⚠ Not secure | 3.85.43.54/index.php

Information about customers from i-05b50c430f06d5806

Failed to connect to MySQL: Server sent charset unknown to the client. Please, report to the developers

Error creating table.

NAME ADDRESS CITY COUNTRY Add Data

ID	NAME	ADDRESS	CITY	COUNTRY
----	------	---------	------	---------

Screenshot 12

Request to second EC2 instance (machine2).

3.231.26.56/index.php x 3.85.43.54/index.php x +

← → ↻ ⚠ Not secure | 3.231.26.56/index.php

Information about customers from i-0645fbc2feab9d103

NAME ADDRESS CITY COUNTRY Add Data

ID	NAME	ADDRESS	CITY	COUNTRY
6	Vlasis	Street1	Athens	Greece

Screenshot 13

A customer was inserted successfully on MySQL after pushing the "Add Data" button on the first EC2 instance (machine 1).

3.231.26.56/index.php x 3.85.43.54/index.php x +

← → ↻ ⚠ Not secure | 3.85.43.54/index.php

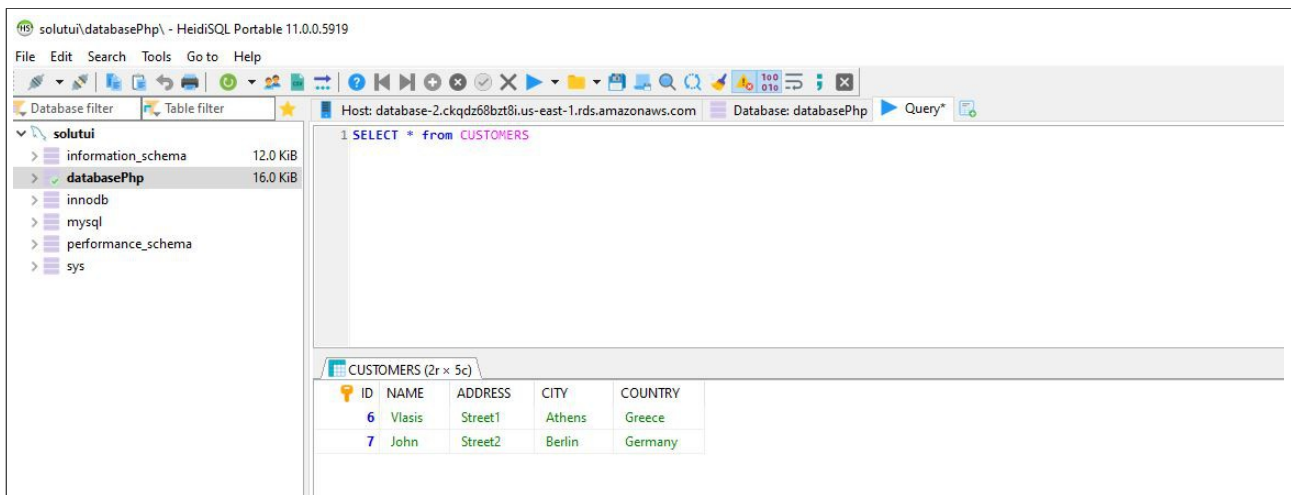
Information about customers from i-05b50c430f06d5806

NAME ADDRESS CITY COUNTRY Add Data

ID	NAME	ADDRESS	CITY	COUNTRY
6	Vlasis	Street1	Athens	Greece
7	John	Street2	Berlin	Germany

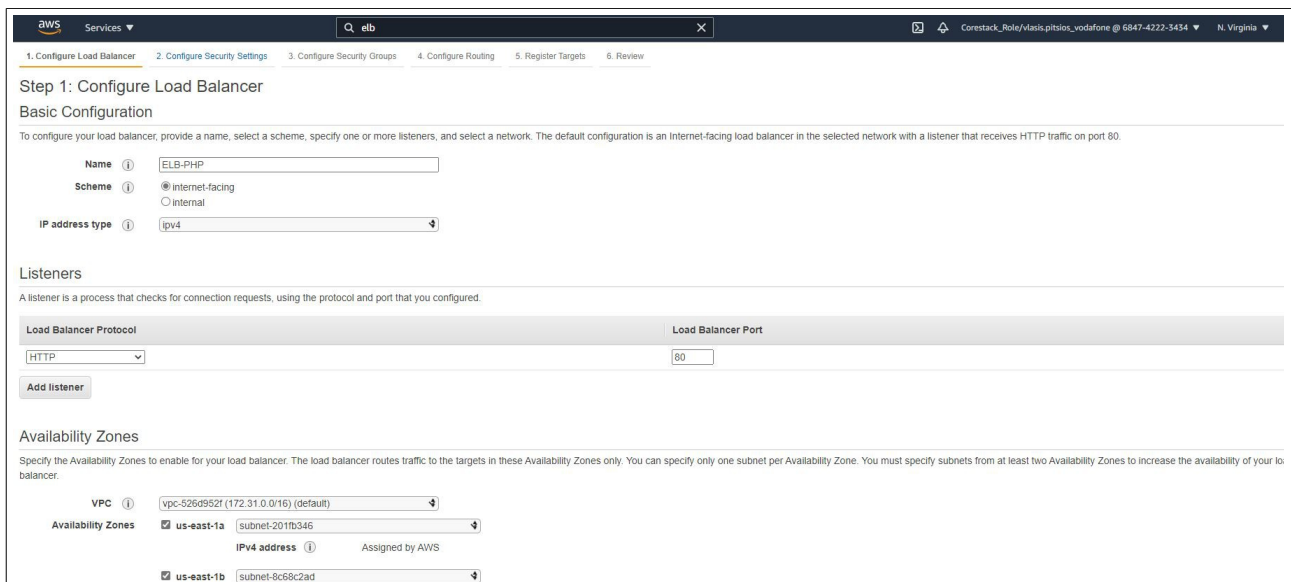
Screenshot 14

A second customer was inserted successfully on MySQL after pushing the "Add Data" button on the second EC2 instance (machine 2).



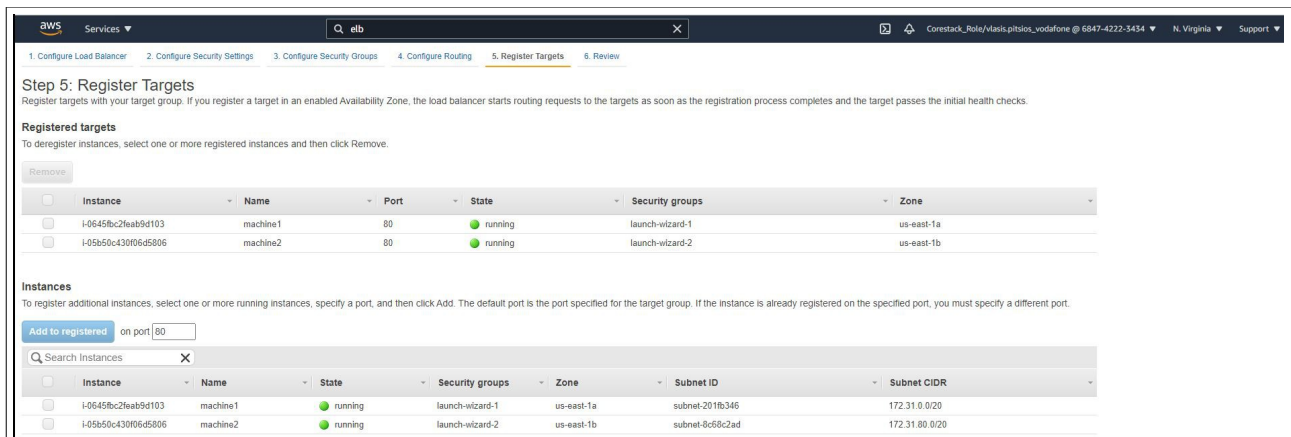
Screenshot 15

The two customers are inserted successfully on MySQL. Checking by running a Select statement on the database directly.



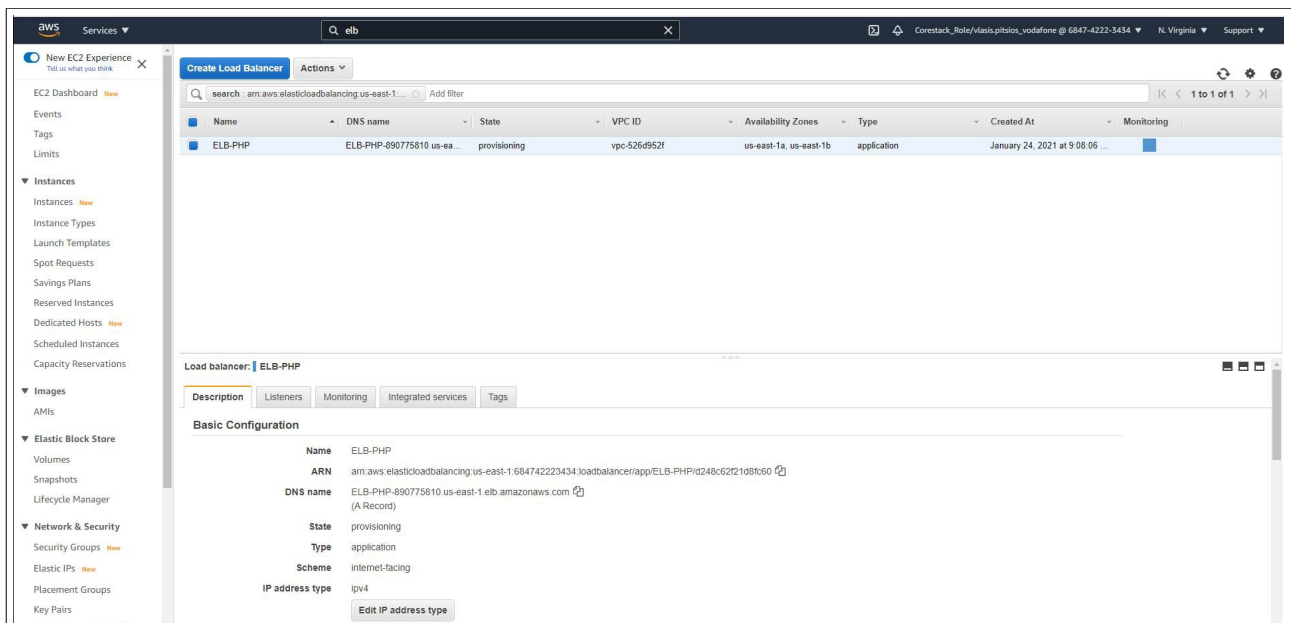
Screenshot 16

Configuration of an Elastic Load Balancer (AWS ELB) in order to split traffic in two EC2 instances on different availability zones (us-east-1 and us-east-2).



Screenshot 17

Register the two EC2 instances as targets on the load balancer.



Screenshot 18

The Load balancer is ready. Also a DNS name is available in order to have access to the load balancer and execute requests.



Screenshot 19

Access the load balancer on the DNS name with "index.php" included and add one more customer. The request is served by the first EC2 instance (machine 2).

3.231.26.56/index.php x 3.85.43.54/index.php x elb-php-890775810.us-east-1.elb.amazonaws.com/index.php +

← → ↻ ⚠ Not secure | elb-php-890775810.us-east-1.elb.amazonaws.com/index.php 🔍 ⚙

Information about customers from i-0645fbc2feab9d103

NAME ADDRESS CITY COUNTRY

Add Data

ID	NAME	ADDRESS	CITY	COUNTRY
6	Vlasis	Street1	Athens	Greece
7	John	Street2	Berlin	Germany
11	Jane	Street 3	London	United Kingdom

Screenshot 20

Run again the same request on load balancer's DNS name. The second request is served by the second EC2 instance (machine 1).