

INTELLIPAAT

AWS CAPSTONE PROJECT-1

- Iyappan

[Link to source code](https://github.com/Iyappan97/Php_RDS_Autoscale_project.git): https://github.com/Iyappan97/Php_RDS_Autoscale_project.git

Problem Statement:

Company ABC wants to move their product to AWS. They have the following things set up right now:

1. MySQL DB
2. Website (PHP)

The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website.

Steps To Solve:

1. Launch an EC2 Instance
2. Enable Auto Scaling on these instances (minimum 2)
3. Create an RDS Instance
4. Create Database & Table in RDS instance:
 - a. Database name: intel
 - b. Table name: data
 - c. Database password: intel123
5. Change hostname in website
6. Allow traffic from EC2 to RDS instance
7. Allow all-traffic to EC2 instance

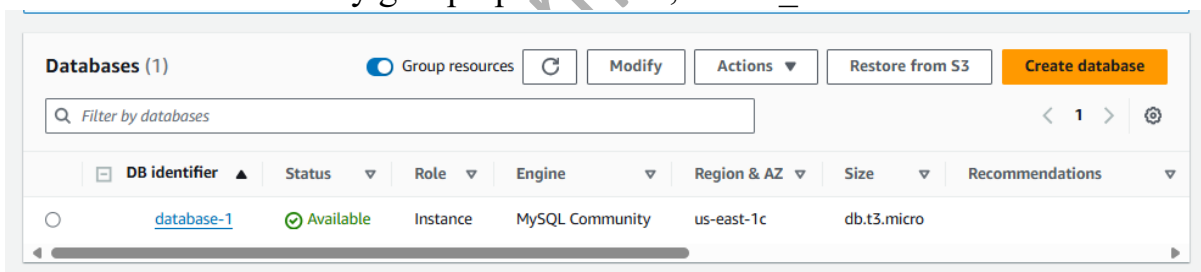
SOLUTION:

PROJECT REQUIREMENT:

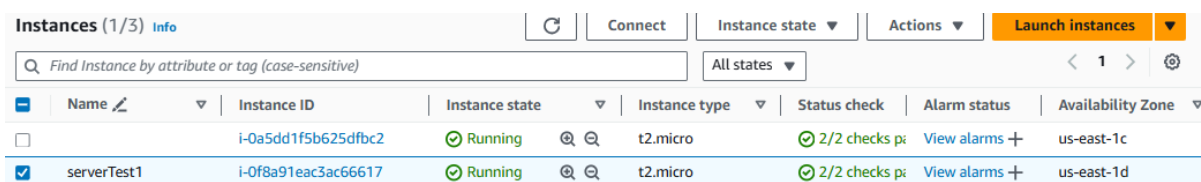
1. Running Ec2 instance with apache2 installed in it.
2. Php application with connection to Database
3. RDS instance to store database
4. Auto-scaling group
5. Load balancer with target group to pointed to Auto-scaling group

STEPS:

1. Create a security group:
 - a. Name: SG-1
 - b. Port: http ; Allow: Anywhere
 - c. Port:SSL : Allow:Anywhere (Restrict this access to allow only from your machine for better security)
2. Create a Target group (TG1) – we will use it when we create Auto-scaling group.
3. Creating RDS instance
 - a. Name : database-1
 - b. Security group : port : 3306 ; Allow_From: SG-1



- c. Create a user and save the pswd
4. Create a Ec2 instance
 - a. Instance_type : (basic is OK) t2.micro(free tier)
 - b. Image : ubuntu
 - c. Secutrity group: Port: 22 (SSL); Allow: anywhere(restrict this)
Port: html; Allow : anywhere

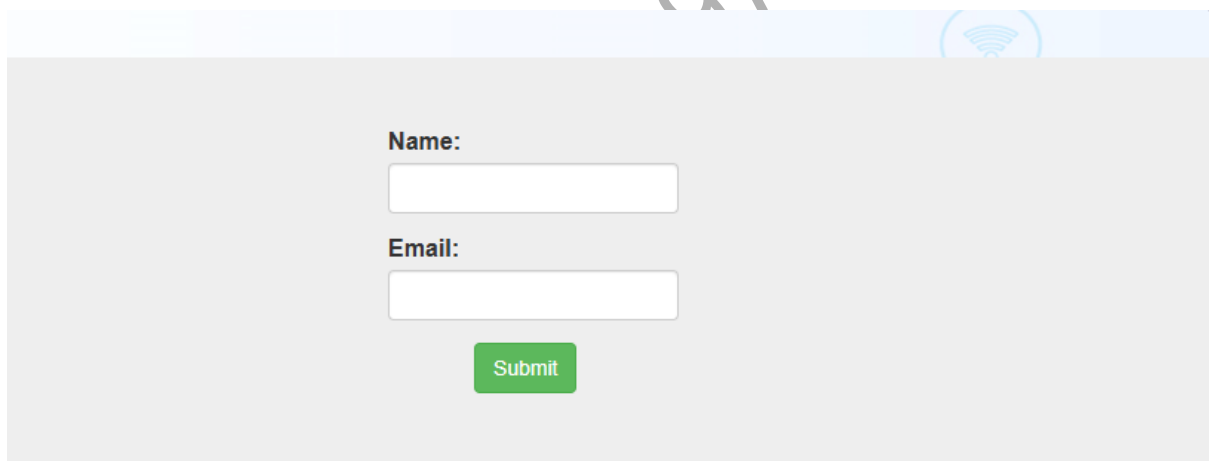


5. Login into Instance

- \$ sudo apt-get update
- \$ sudo apt install apache2
- \$ cd /var/www/html/
- Remove index.html
- Place ur php file with its dependencies

```
ubuntu@ip-172-31-47-170:~$ ls
ubuntu@ip-172-31-47-170:~$ cd /var/www/html/
ubuntu@ip-172-31-47-170:/var/www/html$ ls
images  index.php
```

- Edit the php file to configure it to connect with RDS instance we created.
- Install mysql which supports PHP.
- Login to RDS server create a database & Table as configured in php file.
- Check connection with database server and update and check.
- Open ur php app in web browser



Name:

Email:

Submit

- Add entry into the given field.
- Check the entry from mysql table.

```
mysql> select * from data;
+-----+-----+
| firstname | email |
+-----+-----+
| iyappan   | 000@gmail.com |
| ritheesh  | worstboy@world.com |
| ritheesh  | kuttukunjan@gmail.com |
| ragu      | raguraja@gmail.com |
| amma      | asaithambivalli@gmail |
| jaskbjaskjd | jafjkabskfjbsdkjbfd |
| jahdfkhd  | svadjhvfhdavhjd |
| 111111111 | 1111111111111111111 |
+-----+-----+
8 rows in set (0.00 sec)
```

6. Create a Image out of the running instance that we created.
7. Create a template from the image use the same SG-1.
8. Create a Loadbalancer & attach it to TG1 target group.
9. Create a auto-scaling group with the template that we created mention the followings
 - a. Desired instance
 - b. Maximum instance
 - c. Minimum instance
10. Attach the load balancer to auto-scaling group.

TEST:

1. Ping the laodbalancer endpoint it should reflect the web application.

The image shows two screenshots from the AWS Management Console. The top screenshot displays the 'Load balancers' page with a table listing one load balancer, 'Php-app-LB', which is in an 'Active' state. The bottom screenshot shows the details of this load balancer, including its ARN and DNS name. A notification indicates that the DNS name has been copied. Below this, a web browser window is shown with the URL 'php-app-lb-765025552.us-east-1.elb.amazonaws.com'. The browser displays a simple web form with 'Name:' and 'Email:' input fields and a 'Submit' button. A message at the bottom of the browser window states 'New record created successfully'.

Load balancers (1/1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Filter load balancers

| <input checked="" type="checkbox"/> | Name | DNS name | State | VPC ID | Availability |
|-------------------------------------|------------|---------------------------|--------|-----------------------|--------------|
| <input checked="" type="checkbox"/> | Php-app-LB | Php-app-LB-765025552.u... | Active | vpc-0223d8d1b6d9a0... | 4 Availabili |

Load balancer: Php-app-LB

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:260083387156:loadbalancer/app/Php-app-LB/0f92c0d117512079

subnets: subnet-03356ba85ae443958 (us-east-1a (use1-az1))
subnet-03356ba85ae443958 (us-east-1b (use1-az2))

✔ DNS name copied

Php-app-LB-765025552.us-east-1.elb.amazonaws.com (A Record)

Not secure | php-app-lb-765025552.us-east-1.elb.amazonaws.com

Name:

Email:

Submit

New record created successfully

2. Update entry in the field

Name:

Email:

ew record created successfully

3. check the entry in RDS database.

| firstname | email |
|--------------|-----------------------|
| iyappan | 000@gmail.com |
| ritheesh | worstboy@world.com |
| rithesh | kuttukunjan@gmail.com |
| ragu | raguraja@gmail.com |
| amma | asaithambivalli@gmail |
| jasbkjbaskjd | jafjkabskfjbsdkjbfkj |
| jahdfkhd | svadjhvfhdavhjdf |
| 1111111111 | 11111111111111111111 |
| last_test | it_should_reflect_in_ |

9 rows in set (0.00 sec)

4. delete a instance to check auto scaling is scaling up to desired state.

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availat |
|-------------------------------------|---------------------|----------------|---------------|--------------|---------------|---------|
| <input checked="" type="checkbox"/> | i-0a5dd1f5b625dfbc2 | Shutting-down | t2.micro | 2/2 checks p | View alarms + | us-east |
| <input type="checkbox"/> | serverTest1 | Shutting-down | t2.micro | - | View alarms + | us-east |
| <input checked="" type="checkbox"/> | i-09e606e3467bd53d0 | Shutting-down | t2.micro | 2/2 checks p | View alarms + | us-east |

Auto Scaling group: Php_app_ASG

| Instance ID | Lifecycle | Instance type | Weighted ca... | Launch tem... | Availability ... | Health status | Protected from |
|-------------------------------------|-------------|---------------|----------------|-----------------------------------|------------------|---------------|----------------|
| i-00499948fac0c2392 | Pending | t2.micro | - | Php_app_launchTem | us-east-1d | Healthy | |
| i-09e606e3467bd53d0 | Terminating | t2.micro | - | Php_app_launchTem | us-east-1d | Unhealthy | |
| i-0a5dd1f5b625dfbc2 | InService | t2.micro | - | Php_app_launchTem | us-east-1c | Healthy | |

Hence , we created and tested the project completed