Database and containers

Networking

Configure the vpc

Create two availability zones

Create public and private subnet on each zone, although there was not requirement of other az

Create the rds with mysql engine and place it in zone1 private subnet

Configure the ecs service using ec2 as it is free

Create a task for phpadmin container, set the required parameters for container e.g cpu, memory, ports and docker image.

Run the task on ecs machine

Edit the vpc security group to accept traffic from ecs machine

Edit the ecs machine security group to accept traffic from my machine

I was successfully access the dashboards of my rds through phpadmin container.

**Building a Basic AWS Infrastructure with ECS and RDS**

**Here's a breakdown of my recent AWS project:**

1. **VPC Setup:** I configured a VPC with two Availability Zones to enhance fault tolerance. Within each zone, I created both public and private subnets, even though initially only one zone was required. This decision was made with future scalability in mind.
2. **RDS Deployment:** A MySQL RDS instance was created and placed in the private subnet of Zone 1 to isolate the database from public internet access.
3. **ECS Service Creation:** Given the free tier benefits, I opted for an EC2 launch type for my ECS service.
4. **PHPAdmin Container Task:** I defined a task for a PHPAdmin container, specifying essential parameters such as CPU, memory, port mappings, and the Docker image. This task was then scheduled to run on the ECS machines.
5. **Security Group Configuration:** To enable communication, I adjusted the VPC security group to permit traffic from the ECS machines. Additionally, the ECS machine's security group was configured to accept incoming connections from my local machine.

**Outcome:** Successful implementation of the above steps allowed me to access my RDS instance's dashboard through the PHPAdmin container.

[Relevant hashtags: #AWS #ECS #RDS #PHPAdmin #CloudComputing #Infrastructure #DevOps]