greatlearning

PGPCC | Project

Deploying a data entry application on Containers

Prelude-From Infrastructure to Containers

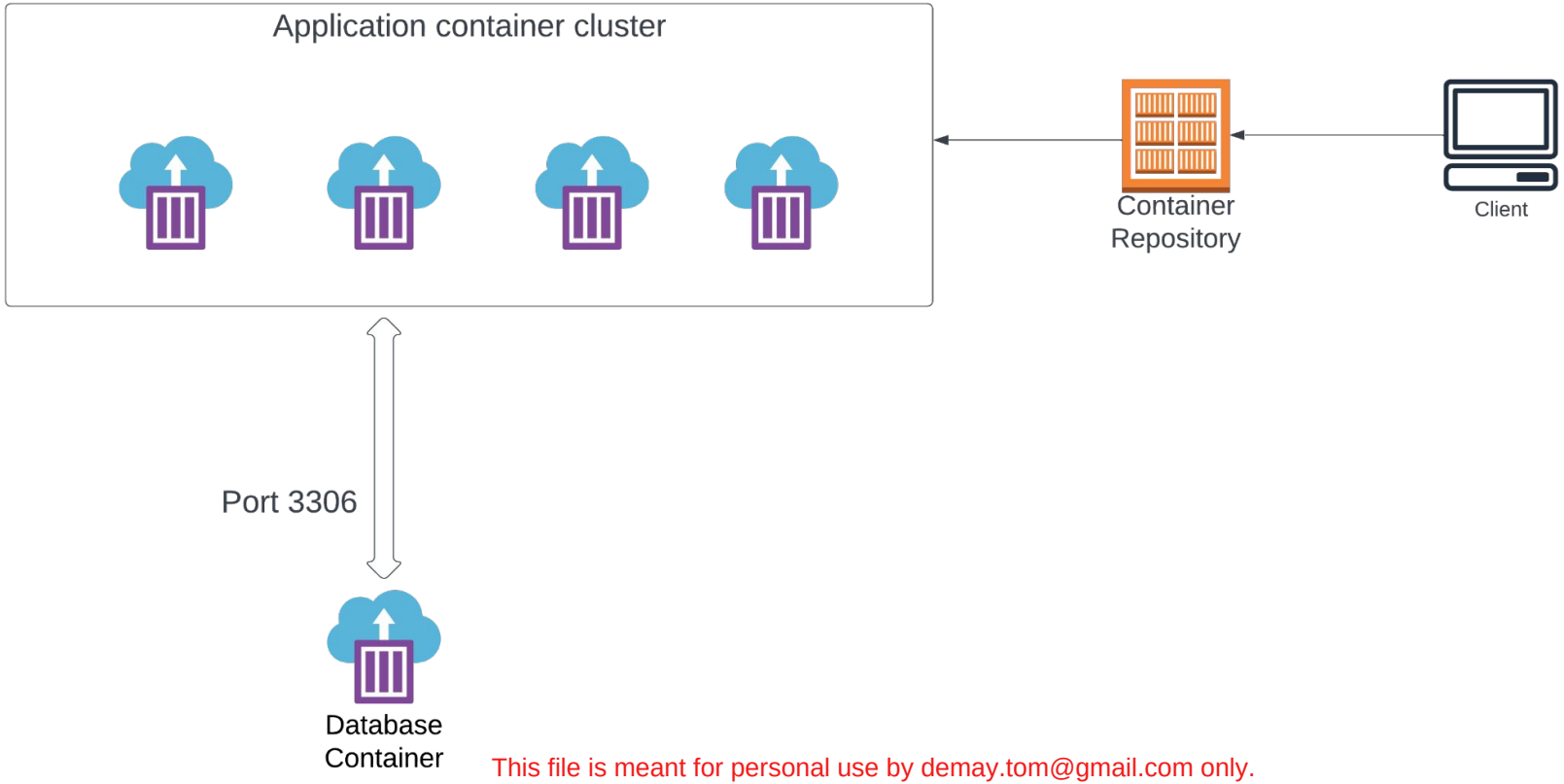
Containerization of an application has a distinct advantage over an equivalent IaaS deployment, in that the deployment becomes essentially hardware architecture-independent and can easily be ported to multiple platforms with minimal effort. This is a stark contrast to infrastructure-based deployments where multiple factors must be taken into account, such as operating system, hardware,underlying libraries and dependencies etc.

According to Gartner, the container management market has seen accelerated growth over the past year. The market is forecast to reach $1.4 billion by 2025, with a 25.1% CAGR. As per their estimate, by the end of 2022 ,more than 75% of global organizations will be running containerized applications in production, up from less than 30% in 2020.As a result of this, enterprise demand for container management is increasing. Container management provides software and/or services that support the management of containers, at scale, in production environments.

Scenario

This deployment would require you to host the provided PHP application on an ECS cluster with the backend database being hosted on an EC2 instance running a MySQL container on Docker. The database container will need to be exposed for remote connections and logins. The PHP application should be configured with the database parameters and credentials and have the required libraries enabled in the container.

Base Architecture



What are you expected to do?

1. **Phase 1 – Architecture**

   Create an architecture diagram for the final implementation (Point 2 in this slide). (hint. will be an update to base architecture). *This is for your implementation reference and you are required to submit this as part of your Solution Document (Project Report).*

2. **Phase 2 – Implementation**

   A. Download the zip file crud.zip provided with this project document on the Olympus portal
   B. Create an EC2 instance using Amazon Linux 2
   C. Deploy a MySQL container on the EC2 instance
   D. Package the provided PHP application into a Docker Image
   E. Push this Docker image into an ECR repository
   F. Deploy this image on ECS Fargate with a Load Balancer attached
   G. Access the PHP application on the web browser using the port configured in ECS

Note : Use the built-in roles LabInstanceProfile and LabRole for EC2 and ECS respectively if you are using your AWS Academy account for this exercise

greatlearning

MySQL setup as a Container

1. Pull the mysql/mysql-server:latest image from Dockerhub
2. Start the container using the docker run command. A sample command has been provided below
   docker run –name=mysql_docker -e MYSQL_ROOT_HOST=% -p 3306:3306 -d mysql/mysql-server
3. Access the temporary password set by MySQL using the below command
   docker logs <container name>
4. Log in to the container bash using the below command
   docker exec -it <container name> bash
5. Login to the MySQL shell
6. Change the root temporary password to one of your choice using the ALTER USER command on 'root'@'localhost' and 'root'@'%'
7. Create a database in the MySQL shell using the CREATE DATABASE command.
8. Exit out of MySQL and the container bash and login to the MySQL container from the EC2 environment.
9. Deploy the provided SQL file crud.sql into the MySQL container. This file is provided in the zip file crud.zip

Setting up the application as a container

1. Create the Dockerfile for the application. This can use the standard PHP image from Dockerhub
2. Modify the file backend/database.php to contain the parameters selected while creating the MySQL container
3. Create an image using the applicationfiles and Dockerfile. Either docker build or docker compose can be used for this
4. Enable mysqli() inside the container using the below commands after logging into the container bash
   docker-php-ext-install mysqli
   apachectl restart
5. Save the above change to the image using the below command
   docker commit -p <php container name> <image name>

Helpful Links

- Installing MySQL on Docker
- Installing PHP on Docker
- Docker documentation
- AWS documentation
- Container and AWS DevOps course – Instructional Modules

greatlearning

Mandatory Step-Resource CleanUp

1. Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
2. **After completing with the lab, make sure to delete each resource created in the reverse chronological order.**
3. Check resources in each cloud region that you have worked on before logging off.
4. Since the dashboard doesn't show cross-region resources, it is upto you to find and delete them.

Note: ***If you fail to clean up your lab account, your submitted project will not be evaluated and treated as incomplete work.***

Submission Guidelines

- The solution document should strictly follow the sequence of steps listed in “The Solution” slide.
- Each screenshot needs to be qualified with a brief description of what is it about.
- Participants should explicitly write comments and remarks if they wish to notify the evaluator of specific points.
- It is mandatory to share “Lessons & Observations” at the end of the solution document.
- RESOURCE CLEANUP & MEANINGFUL USAGE is mandatory for all participants.
    - When you are not working on the project, make sure that you have stopped all resources.
    - When you have finished your project and documented everything, you MUST clean up your AWS account by deleting all the resources created by you. (except for default items)
    - Clean-up screenshots should also be included in the submission document. If you fail to include these screenshots your submission will not be evaluated and treated as incomplete.
- DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.

greatlearning

Academic Honesty & Anti-Plagiarism Policy

Cheating, plagiarism, and all forms of academic dishonesty are expressly forbidden in this program, and by our Policy on Academic Integrity, any form of cheating will immediately earn you a failing grade for the entire course.

Note: Unlike labs where we encourage peer to peer learning and support, participants are strongly advised to not help each other for Projects. This is an individual exercise. Any form of help/support whether offered explicitly, proactively or as a response will be treated as plagiarism.

greatlearning

How to submit your solution?

1. Navigate to the relevant “PROJECTS” course in Olympus.
2. Name your solution document appropriately in the format of: BATCH_FIRSTNAME_LASTNAME_PROJECT1;
    - e.g. PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf
    - e.g. pgpccmay18_vijay_dwivedi_project1.pdf
3. Upload your solution document and hit submit.
4. Upload any associated files, if you wish to substantiate your solution.

***Note**: If you wish to make modifications to your submitted solution, you can resubmit your solution document “within the submission window” and mark your comments accordingly.*