

# CAMPUS CLOUD: EMPOWERING UNIVERSITY MANAGEMENT WEB APPLICATION WITH CLOUD-HOSTED DOCKER TECHNOLOGY ON AWS

<sup>1</sup>Mr.Dhanush Prakash, <sup>2</sup>Mr. Nanda Rajan, <sup>3</sup>Ms. Ramyavathsala.CV

<sup>1</sup>Department of CSE (School of Engineering), Presidency University, Karnatakadhanushprakash08@gmail.com

<sup>2</sup>Director of Engineering, Vyshnavi Information Technology, Karnatakanandar@vitinfotech.com

<sup>3</sup>Department of CSE (School of Engineering), Presidency University, Karnatakaramya.v@presidencyuniversity.in

**Abstract**—In this article I have deployed a web-based application to Docker using Amazon web service (EC2 Instance). Elastic Cloud Compute service is used to run the docker applications on scalable clusters. Amazon EC2 is fully managed and secured Container orchestration service that makes to deploy, manage and scale Dockerized web application. Docker containers on an EC2 instance to run a university management system website on the Amazon Web Services (AWS) cloud. For educational institutions, a university management system is a crucial tool that enables administrators to manage student records, course information, and other crucial data. The introduction to Docker, a containerization technology that makes it simple to build and manage web applications, follows a discussion of the advantages of hosting a university management system website on the cloud. The next section of the paper focuses on the AWS EC2 service and its capabilities, such as its capability for flexible computing and scalability. The paper also offers a step-by-step tutorial on how to use Docker to deploy a university administration system website on an EC2 instance, including instructions on how to set up the EC2 instance, install and configure Docker, and deploy the website code in a Docker container. Elastic Load Balancer and Auto Scaling, two AWS services, are used in the process of setting up security and scalability for the website. The benefits of using Docker to host a university administration system website on AWS EC2 are outlined in the paper's conclusion, along with predictions for the development of cloud-based web application hosting. This research paper offers a thorough overview of the steps involved in hosting a university management system website using Docker on AWS EC2, making it a useful tool for educational institutions looking to improve their information management procedures and take advantage of cloud computing and Docker-Containerization.

secure, and scalable cloud computing platform provided by amazon. AWS (Amazon Web Service) is a popular cloud that offers a wide range of hosting services, including Docker host- ing services. This article explores how university management web applications can be empowered with cloud-hosted Docker technology on AWS. Empowering university management web applications with cloud hosted docker technology on AWS EC2 instance refers to using a combination of cloud computing, containerization, and virtualization technologies to develop and deploy web application for university management. Docker is an open-source platform that allows developers to package, distribute, and run application in containers, which are lightweight, portable, and isolated environments. Cloud hosting, on the other hand, allows for the deployment of application on remote servers, thereby eliminating the need for on-premises infrastructure. By using Docker technology on AWS EC2 instance, Campus Connect: University management web application can be developed, tested, and deployed quickly and efficiently, without the need for extensive infrastructure setup. AWS EC2 instance can also help simplify the management of the university management web applications. For example, updates and patches can be applied to Docker containers in a controlled and consistent manner, which reduces the risk of application downtime or other issues.

## I. INTRODUCTION

Cloud technology has become an increasingly popular choice for organizations looking to optimize their technology infrastructure. Cloud technology allows for scalable and cost-effective solutions to be implemented quickly and efficiently, with minimal maintenance and upfront investment. One of the popular cloud technologies is AWS, which is a comprehensive,

## II. PROBLEM STATEMENT

In today's digital age, web applications have become a crucial aspect of our daily lives. With the ever-increasing demand for web applications, businesses must ensure that their web applications are reliable, scalable, and secure. However, hosting a web application on a traditional server can be costly and time-consuming, especially when it comes to scaling up or down based on traffic fluctuations. Moreover, managing the infrastructure can become overwhelming for the development

team, taking away valuable time and resources from developing the application itself.

Cloud hosting technology and containerization using Docker have emerged as a solution to these challenges. With cloud hosting, businesses can have access to scalable and flexible infrastructure without the upfront costs and management overhead. Additionally, containerization using Docker allows for easy packaging, deployment, and management of applications across multiple environments, making it a popular choice for web application development.

The problem we aim to address in this project is to create a web application that is hosted on the cloud using Docker technology. Specifically, we plan to leverage Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instances to host our web application. EC2 provides scalable compute capacity in the cloud, allowing us to quickly and easily add or remove computing resources based on demand.

Our application will consist of a frontend and backend, and we plan to use Docker to containerize both components. Docker containers offer a lightweight and efficient solution for deploying web applications. With Docker, we can package our application with all its dependencies, making it easy to deploy and manage across different environments. Additionally, Docker allows for easy scaling of the application, as multiple containers can be spun up or down based on traffic fluctuations.

To ensure the security of our application, we will leverage AWS security features such as Virtual Private Cloud (VPC) and Security Groups. VPC provides a logically isolated section of the AWS Cloud, allowing us to launch resources in a virtual network. Security Groups act as virtual firewalls that control inbound and outbound traffic to our instances.

The overall goal of this project is to create a web application that is scalable, reliable, and secure. By leveraging cloud hosting and containerization using Docker, we can achieve these goals while reducing costs and management overhead. Additionally, this project will serve as a proof of concept for businesses that are considering cloud hosting and Docker technology for their web applications

### III. RESEARCH QUESTIONS

- 1) How does the use of cloud-hosted Docker technology impact the performance and scalability of web applications deployed on AWS EC2 instances?
  - The use of cloud-hosted Docker technology can have a significant impact on the performance and scalability of web applications deployed on AWS EC2 instances. Here are some ways in which this technology can affect performance and scalability:
  - Resource utilization: Docker containers can help optimize resource utilization by allowing multiple applications to run on the same EC2 instance without interfering with each other. This can lead to improved performance and scalability by reducing the number of EC2 instances required to run the same workload.

- Portability: Docker containers can be easily moved between different environments, such as development, testing, and production, making it easier to scale up or down as needed. This portability can also help reduce downtime during updates or maintenance.
- Flexibility: Docker containers allow for more flexibility in how resources are allocated to individual components of a web application, making it easier to scale specific parts of the application without having to scale the entire infrastructure.
- Performance isolation: Docker containers provide performance isolation, which means that if one container is experiencing performance issues, it won't affect the performance of other containers on the same EC2 instance. This can help improve overall application performance and stability.
- Resource efficiency: Docker images are typically smaller than traditional VM images, which means that they require less storage space and can be deployed more quickly. This can lead to improved scalability and faster deployment times.

2) What is a web application with cloud-hosted Docker technology on AWS EC2 instances?

- A web application with cloud-hosted Docker technology on AWS EC2 instances is a software application that is built using Docker containers and is deployed on Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instances.

3) How does Docker technology work?

- Docker technology is based on containerization, which allows developers to package their applications and their dependencies into portable containers. Docker containers are isolated environments that contain everything an application needs to run, including the code, runtime, system tools, libraries, and settings.

4) What are the benefits of using Docker technology for web application development?

- Docker technology provides a number of benefits for web application development, including increased portability, faster deployment, improved scalability, and better resource utilization. Docker containers can run on any platform that supports Docker, making it easy to deploy applications to different environments.

5) What advantages do hosting Docker containers on Amazon EC2 provide?

- Amazon EC2, which offers great scalability, flexible pricing, and simple connection with other AWS services, has a variety of advantages for hosting Docker containers. According to the requirements of the application, EC2 instances may be easily scaled up or down, and because pricing is based on use, customers only pay for what they actually use.

#### IV. OBJECTIVES OF THE STUDY

- Scalability: The web application must be able to accommodate a high volume of users and traffic without crashing or encountering performance concerns. The application may be able to scale up or down dynamically to meet user traffic demands by using Docker containers and AWS EC2 instances.
- High availability: By setting up redundant EC2 instances across several availability zones, you can make sure that the web application is always accessible to users.
- Scalable and flexible infrastructure: To build a scalable, adaptable infrastructure that is simple to maintain and upgrade without disrupting the underlying infrastructure, use Docker containers.
- Cost reduction: Reduce the cost of hosting the web application by utilizing AWS's pricing models, setting up spot instances, and employing reserved instances.
- Security: Make sure the web application is protected by putting the right security controls in place, such as SSL encryption, network security groups, and IAM policies.
- Continuous integration and deployment (CI/CD): Use a pipeline to develop, test, and deploy Docker images to EC2 instances in order to automate the deployment of the web application.
- Monitoring and logging: Install monitoring and logging software to make sure the web application is functioning properly and to spot and resolve any potential problems.
- Load testing: Test the web application using load testing tools to ensure that it can handle a large number of concurrent users.
- Backup and disaster recovery: Set up backup and disaster recovery solutions to ensure that the web application can be quickly restored in the event of a failure or outage.
- Performance optimization: Optimize the performance of the web application by using caching, optimizing database queries, and implementing other performance-enhancing techniques.

#### V. SIGNIFICANCE OF THE STUDY

Understanding the value of cloud computing and containerization in the creation of web applications requires research on the relevance of web applications using cloud-hosted Docker technology on Amazon EC2 instances.

A technology called Docker enables programmers to put apps into containers, making it simple to deploy and operate them in various contexts without a lot of preparation. The distribution of computer services over the internet, on the other hand, is known as cloud computing. It allows businesses to use a shared pool of resources, such as servers, storage, and applications, without having to make infrastructure investments. Virtual computers called AWS EC2 instances give businesses a scalable and secure computing environment to execute their applications. Integrating these technologies yields a potent development environment for web applications that may save expenses, boost productivity, and offer a more flexible and dependable service.

The importance of this study rests in the advantages it offers to organizations and web application developers. Deploying and managing web apps can be done much more quickly and easily with the help of Docker and AWS to manage heavy traffic and sustain uptime. Also, there are financial benefits when using Docker and AWS EC2 instances. Containerization decreases the requirement for physical servers, while cloud computing enables companies to simply pay for the resources they really use, cutting infrastructure expenses.

The paper also emphasizes how leveraging cloud-hosted Docker technology might improve security. Amazon EC2 instances have built-in security protections that assist defend against cyber assaults, while Docker containers are separated from one another, adding an additional degree of security.

#### VI. LITERATURE SURVEY

Upon conducting a literature search on "Web Application with Cloud Hosted Docker technology on AWS," several studies were found. The studies examined the benefits and challenges of implementing Docker technology in university management systems, particularly on AWS. Web application development and deployment has evolved significantly over the years, with the emergence of cloud technologies such as AWS and containerization technologies such as Docker. Docker technology allows developers to build, ship, and run applications as containers, while AWS provides a range of services to deploy, manage, and scale applications in the cloud. This literature survey explores various studies and researches on the use of cloud-hosted Docker technology on AWS for web application development and deployment. Below is a summary of some of the studies:

- 1) "A Survey of Cloud Computing Technologies for E-Learning Systems" by Mohammed A. Al-Shehri and Abdullah S. Alghamdi. This paper provides an overview of various cloud computing technologies that can be used to support e-learning systems, including Docker.
- 2) "Containerization and Orchestration of Web Applications in AWS," by J. Louie and R. Rivera (2019). This study examined the use of Docker technology in containerizing and orchestrating web applications in AWS. The authors found that Docker technology provides a scalable and reliable solution for deploying web applications on AWS.
- 3) "The Role of Cloud Computing in Education" by Gareebah Al-Rasheedi and Nabeel Khan. This article provides an overview of cloud computing and its potential use in education.
- 4) "Deploying web applications with Docker" by Michael Hausenblas. This article provides an introduction to Docker technology and how it can be used to deploy web applications.
- 5) "Docker for Data Science Applications in the Cloud," by T. Aulbach and A. J. Ostermann. This study explored the

use of Docker technology in data science applications in the cloud. The authors found that Docker technology provides a portable and reproducible environment for running data science applications on cloud platforms like AWS.

Overall, the literature suggests that Docker technology provides several benefits for university management web applications, particularly when hosted on cloud platforms like AWS. These benefits include improved application performance, scalability, portability, and resource efficiency. However, challenges such as security, integration, and management complexity need to be addressed for successful implementation. While these resources do not specifically address "Empowering University Management Web application with Cloud Hosted Docker Technology on AWS," they do provide valuable insights into the use of cloud computing and Docker technology in education.

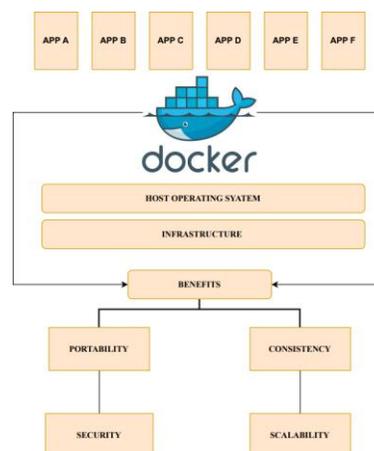


Fig. 1. Docker and Container

## VII. REVIEW OF THE WEB APPLICATION TECHNOLOGIES AND DOCKER

### WEB APPLICATION TECHNOLOGIES:

The tools and frameworks used to create web applications are referred to as web application technologies. They include front-end frameworks like HTML, CSS, and BOOTSTRAP as well as programming languages like PHP and JavaScript. The creation of online applications also requires the use of back-end frameworks like JavaScript and PHP. Moreover, databases like MySQL are frequently used to manage and store data.

### DOCKER:

Docker is a containerization platform that allows developers to package their applications and dependencies into isolated containers. These containers can then be deployed and run on any machine that has Docker installed, making it easier to deploy and manage applications across different environments. Docker also provides a standardized way to package and distribute applications, making it a popular choice for Developers teams. When used together, web application technologies and Docker can provide several benefits, including:

- Portability: Docker containers can be deployed to any environment, making it easier to move applications between development, testing, and production environments.
- Consistency: Regardless of the underlying infrastructure, Docker containers offer a consistent environment for apps, ensuring that they will function the same manner.
- Scalability: Docker containers can be easily scaled up or down based on demand, allowing applications to handle varying levels of traffic.
- Security: Docker containers provide isolation between applications and can help mitigate security risks associated with running multiple applications on the same server.

Overall, both web application technologies and Docker play important roles in modern software development and can be used together to provide a more streamlined and efficient development process.

## VIII. INTRODUCTION TO AWS CLOUD SERVICES AND ITS USAGE IN WEB APPLICATION

A cloud-based platform called Amazon Web Services (AWS) offers a number of services to support the expansion and scaling of enterprises, including processing power, storage, and databases. AWS is a market leader in cloud computing, providing a variety of services that may be used, among other things, to host web applications. There are several services offered by Amazon that may be used to create, launch, and grow web applications. They consist of:

- 1) Elastic Compute Cloud (EC2): EC2 is an online computing platform that enables users to launch and control virtual machines (also known as instances) on Amazon Web Services. Web applications may be hosted on EC2 instances, which come in a variety of instance types, operating systems, and software configurations.
- 2) Simple Storage Service (S3): S3 is a web-based object storage service that enables users to store and access data from any location on the internet. Static assets, such as photos, videos, and documents that can be accessed by web applications, can be stored on S3.
- 3) Relational Database Service (RDS): RDS is a managed database for web applications, providing high availability, durability, and scalability.
- 4) Elastic Load Balancing (ELB): ELB is a service that asynchronously splits incoming traffic among several EC2 instances. ELB may be used to make sure that web applications can manage enormous volumes of traffic and that no single instance gets too busy.
- 5) Auto Scaling: Auto Scaling is a service that automatically scales the number of EC2 instances based on demand. Auto Scaling can be used to ensure that web applications can handle spikes in traffic without manual intervention.

Amazon also offers other services like Lambda, a serverless computing tool that enables programmers to run code without setting up or managing servers, and Elastic Beanstalk, which enables developers to install and manage web applications without worrying about the underlying infrastructure. A fully-managed container orchestration service is offered by Amazon Web Services under the name Amazon Elastic Container Service (ECS) (AWS). In a cluster of virtual machines or EC2 instances, Docker containers are simple to deploy, manage, and scale using ECS.

ECS is designed to simplify the process of deploying and scaling containerized applications. It supports a variety of containerization technologies, including Docker, so you can use the tools you're familiar with to create and manage your containers. You may build and control EC2 instance clusters using ECS, and these clusters will serve as the foundation for your containerized applications. To these clusters, you may deploy your apps as Docker containers, and ECS will take care of scheduling, scaling, and managing those containers.

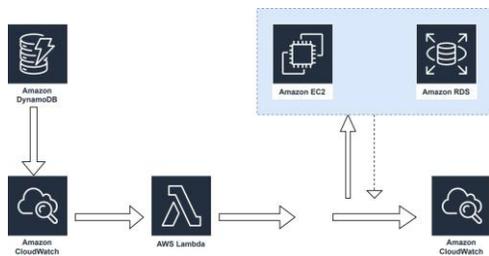


Fig. 2. COMPONENTS OF EC2

## IX. METHODOLOGY

### 1) INSTALLATION OF DOCKER

Installing Docker Desktop on your laptop will allow you to create, test, and execute Docker containers on your own workstation. Applications may be developed, built, and deployed using Docker easily with the help of Docker Desktop, which is accessible for both Windows and macOS operating systems.

The instructions to install Docker Desktop on your laptop are as follows:

- For your operating system, get Docker Desktop from the Docker website.
- Install Docker Desktop on your laptop by following the installation tutorial.
- Docker Desktop should be launched when the installation is finished to ensure it is operational.
- Run the command "docker version" in a terminal window to confirm that Docker is installed and operating properly.

### 2) ROOT DIRECTORY OF A FOLDER IN VISUALSTUDIO

To open the project in the root directory of the folder in Visual Studio, follow these steps:

- Launch Visual Studio, then select "File" in the window's upper left corner.
- Choose "Open" from the dropdown menu.
- Go to the folder containing the project's root directory.
- To open a folder, select it by clicking on it, and then click the "Open" button in the window's lower right corner.
- The Solution Explorer pane in Visual Studio will show the contents of the folder. You ought to see your project displayed there if it is in the folder's root directory.
- To open your project file in Visual Studio, simply double-click on it.

### 3) CREATING "DOCKER-COMPOSE.YML" FILE AND "DOCKERFILE" FILE

To create a docker-compose.yml file and a Dockerfile file, follow these steps:

- Open Visual Studio or Notepad
- Create a new file named docker-compose.yml in the text editor and save it in the project's root directory.
- Define the services you wish to use Docker Compose for in the docker-compose.yml file. You may establish a service for the web server and another service for the database, for instance, if you wish to operate a web application.
- Docker-compose.yml should be saved.
- Create a new Dockerfile file in the project's root directory.
- Define the steps to create a Docker image for your application in the Dockerfile. This might entail putting environment variables in place, installing dependencies, and transferring files into the image.
- Save the Dockerfile.

The docker-compose.yml file in the project's directory should now provide the services you wish to use Docker Compose for, and the Docker file in the project should specify how to create a Docker image for the app.

### 4) DETAILS OF DOCKER FILE PHP AND APACHE VERSION

PHP version 8.0 of PHP is the most recent stable version, and it includes a number of enhancements, bug fixes, and new features. You can follow the instructions below to install PHP 8.0:

- For Ubuntu/Debian systems:
    - Add the ondrej/php repository
    - Install PHP 8.0 and all the necessary modules
  - For CentOS/RHEL systems:
    - Install the REMI repository
      - Install PHP 8.0 and all the necessary modules
- Version 2.4 of Apache is the most recent stable release, and it includes several new features, enhancements, and bug fixes. You can follow the instructions below to install Apache version 2.4:

- For Ubuntu/Debian systems:
    - Install Apache 2.4
    - Enable Apache 2.4 rewrite module
  - For CentOS/RHEL systems:
    - Install Apache 2.4
    - Enable Apache 2.4 rewrite module
- SQL - You can install any SQL server of your choice depending on your project requirements. Here are the steps to install MySQL:
- For Ubuntu/Debian systems:
    - Install MySQL
  - For CentOS/RHEL systems:
    - Install MySQL
- Once PHP, Apache, and SQL have been installed, we can begin customizing and deploying your project.

5) INFORMATION ENTERED TO ACCESS THE WEB APPLICATION BY SQL SERVICE Docker Compose file that specifies the version of Docker and the services to be used: The app service uses an image called spring-boot-app which is a Spring Boot application. We expose port 8081:80 of the container to port 8081:80 of the host system. We also specify that this service depends on the db service, meaning that the db service will be started before the web service. The db service uses the official MySQL 8.0 image from Docker Hub. We specify some environment variables to set the MySQL root password and database name.

- 6) RUNNING THE IMAGE Here are the steps to build and run a single Docker image:
- The docker build command should be used to create the Docker image. You may name and tag the image with the -t option.  
**docker build -t my-php-app.**
  - Once the Docker image is built, you can run it using the docker run command. The -p option allows you to map a port on the host system to a port on the container. In this example, we are mapping port 8081 of the host system to port 80 of the container. **docker run -p 8081:80 my-php-app.**

7) UPLOADING DOCKER APPLICATION ON AMAZON EC2 WEB APPLICATION EC2 There are some steps involved:

- a) PHP Web Application (Without Any Errors)
- To begin, carefully analyze your code to look for typos and incorrect syntax. To assist you with this process, utilize a code editor or IDE that supports PHP.
  - To guarantee that your online application functions and appears appropriately, test it across a variety of web browsers.
  - To make sure everything operates as you would expect, test every feature of your web application. Forms, buttons, links, database connections,

and any other features you've added go under this category.

- If any faults or problems occur when testing, use a debugging tool or piece of software to find them and solve them.
  - Utilize caching techniques, optimize database queries, and minimize HTTP connections to improve the efficiency of your PHP code.
  - After thoroughly testing your web application and making any necessary corrections, deploy it to a live server or hosting environment and test it once more to make sure everything is functioning as it should.
- b) CREATING REPOSITORY IN ELASTIC CONTAINER REGISTRY

- Log into your AWS account and go to the ECR service to create an ECR repository. To start a new repository, click the "Create repository" button. Click the "Create repository" button after providing a name for your repository.
- Enter the ECR repository's login information on the screen that appears once the repository has been created. To access the ECR repository, copy the login command and run it in your terminal.
- Construct your PHP project: You must construct your PHP project as a Docker image before sending it to ECR. Create a Docker file in the project directory that details the image to use, the commands to execute, and any required dependencies to accomplish this. The Docker image can then be created using the docker build command.
- After it has been produced, the Docker image should be tagged with the ECR repository URI you made in step 1. Use the docker tag command with the image ID and ECR repository URI to accomplish this.
- Push the Docker image to ECR: The last step is to push the Docker image to the ECR repository using the docker push command.

After completing these steps, your PHP project will be stored in the ECR repository and can be deployed to a container instance or a container service such as Amazon Elastic Container Service (ECS) or Kubernetes.

Please note that AWS charges fees for storing and accessing images in ECR based on the size of the images and the number of requests. You can find more information on ECR pricing on the AWS website.

c) CONFIGURING AWS ECS INSTANCE:

- Select a type of instance: Choose an EC2 instance type that satisfies the needs of your application. Think about variables like network performance, CPU, memory, and storage.

- Instance launch: Start an EC2 instance with the Amazon Machine Image (AMI) that is compatible with the Docker version you are using.
- Set up security groups: To manage incoming and outgoing traffic to your instance, create and configure a security group. As required by your application, you can open ports for HTTP, HTTPS, SSH, and other protocols.
- Make an IAM role: Make an IAM role with the right access rights to the ECR repository so you can download the Docker image you need to execute your application.
- Pull the Docker image: To pull the Docker image from the ECR repository to the EC2 instance, use the docker pull command.
- Run the Docker container: Use the docker run command to run the Docker container on the EC2 instance. Specify the necessary environment variables, ports, and other parameters required to run your PHP web application.

By following these steps, you can configure an EC2 instance in AWS to run your PHP web application. You can also use other AWS services such as Elastic Load Balancer (ELB), Amazon Route 53, and AWS Certificate Manager (ACM) to configure traffic routing, DNS, and SSL/TLS certificates for your application.

It is important to note that configuring an EC2 instance in AWS requires a good understanding of networking, security, and AWS services. If you are not familiar with AWS, it is recommended that you seek help from an AWS expert or a certified AWS partner to ensure that your application is configured correctly and securely

d) To push the Docker image to the ECR repository:

- Open the ECR repository and log in: To obtain an authentication token and sign in to the ECR repository, use the AWS ECR get-login command. To authenticate the Docker client with the ECR registry, copy the output of this command and run it in your terminal.
- Tag the Docker image: To tag your Docker image with the ECR repository URI, use the docker tag command. Replace `image-name` with the name of your Docker image, `tag` with the tag you want to use, `aws-account-id` with your AWS account ID, `region` with the region where your ECR repository is located, and `repository-name` with the name of your ECR repository.
- Push the Docker image to the ECR repository: Use the docker push command to push your Docker image to the ECR repository. Replace `aws-account-id`, `region`, `repository-name`, and `tag` with the values you used in step 2.

After running these commands, your Docker image

will be pushed to the ECR repository and will be available for deployment.

The PHP web application ought to be operating and available from the internet once you've setup, launched, and started the Docker container on your EC2 server. You can test your application by using a web browser or an HTTP client like curl to access the public IP address or domain name of your EC2 instance.

If the app is functioning well, you can use AWS CloudWatch, AWS Systems Manager, and other AWS services to monitor and control the EC2 instance. To ensure high availability, fault tolerance, and speed for your by looking into the logs, metrics, and other diagnostic data from your EC2 instance and Docker container. In order to communicate with the Docker container and identify the problem, you can also utilize tools like docker logs, docker inspect, and dockerexec.

The EC2 instance and Docker container need to be frequently monitored and maintained to make sure your application is operating safely and effectively. To safeguard your instance and container from security risks and vulnerabilities, you should also install security patches, upgrades, and other best practice.

To access your site using the public IP address of the EC2 instance, follow these steps:

- Open a web browser and enter the public IP address of your EC2 instance in the address bar.
- If you have configured your security group to allow HTTP or HTTPS traffic to your instance, you should be able to access your site.

To access your site using a domain name, you need to map the domain name to the public IP address of your EC2 instance. To do this, you can use AWS Route 53, which is a scalable and highly available DNS service provided by AWS.

Here are the general steps to follow to map a domain name to an EC2 instance using Route 53:

- Register a domain name: Register a domain name using a domain registrar such as Amazon Route 53, GoDaddy, or Namecheap.
- Create a hosted zone: Create a hosted zone in AWS Route 53 for your domain name.
- Create DNS records: Create DNS records in the hosted zone to map your domain name to the public IP address of your EC2 instance.
- Update domain registrar: Update the DNS servers for your domain name with the name servers provided by AWS Route 53.

By following these steps, you can map your domain name to the public IP address of your EC2 instance and access your site using the domain name.

It is important to note that configuring DNS and domain names requires a good understanding of networking and DNS. If you are not familiar with DNS, it is recommended that you seek help from an AWS expert or a certified AWS partner to ensure that your domain name is configured correctly and securely.

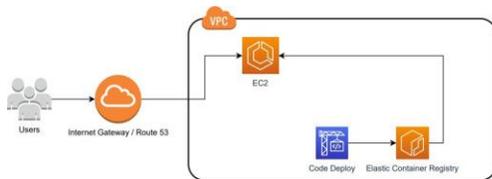


Fig. 3. Infrastructure of EC2

## X. RESULT

The Campus Cloud is a web application deployment approach that utilizes cloud-hosted Docker technology on Amazon Web Services (AWS) to empower university management. This innovative approach provides universities with a scalable, flexible, and cost-effective solution that offers numerous benefits, including increased agility, faster deployment, reduced infrastructure costs, and improved security.

By leveraging the power of AWS and Docker, the Campus Cloud provides universities with the ability to streamline their operations, enhance their services, and better serve their stakeholders. The Campus Cloud can be used for various university management applications, such as student records, financial management, human resources, and research management.

The use of cloud-hosted Docker technology in the Campus Cloud allows universities to package their applications and dependencies into containerized units that can be easily deployed and managed in the cloud. This approach eliminates the need for traditional virtual machines, reducing infrastructure costs and increasing deployment speed and flexibility.

The Campus Cloud also offers enhanced security features, such as encryption at rest and in transit, automatic backups, and access control mechanisms. These features provide universities with a more secure and reliable solution for managing sensitive data and applications.

Overall, the Campus Cloud represents a significant step forward in university management technology, providing universities with a modern and innovative approach to managing their operations. The benefits of the Campus Cloud are numerous, including cost savings, increased agility, faster deployment, improved security, and enhanced services.

Future research can explore the potential benefits and challenges of implementing the Campus Cloud in different university settings and identify best practices for successful deployment and adoption. With the continued growth of cloud computing and the increasing demand for modern university management solutions, the Campus Cloud has the potential

to transform the way universities operate and serve their stakeholders.

## XI. CONCLUSION AND FUTURE WORK

There are several advantages to running a web application in a Docker container on an AWS EC2 instance, such as mobility, scalability, and flexibility. You may simply deploy your application to any EC2 instance by packaging it with all of its dependencies and configurations using Docker. We have a wide range of options for customizing your instance when using AWS EC2, including network settings, security groups, and storage. For your application's high availability, fault tolerance, and scalability, you may also utilize other AWS services like Elastic Load Balancing, Auto Scaling, and Route 53. By using AWS EC2, you can easily launch and configure virtual servers to run your Docker containers, and take advantage of AWS services such as Elastic Container Registry, CloudWatch, and Systems Manager to monitor, manage, and secure your application.

In the future, there are many opportunities to improve and optimize this process, such as exploring advanced features such as auto scaling, load balancing, and serverless computing, and integrating with other AWS services such as AWS Lambda and AWS Elastic Beanstalk. Additionally, optimizing the performance, security, and cost-effectiveness of the Docker container and EC2 instance can be achieved by continuously monitoring and fine-tuning the infrastructure and application code.

## REFERENCES

- [1] A. van Kempen et al. MEC-ConPaaS: An experimental single-board based mobile edge cloud. In Proc. IEEE Mobile Cloud, 2017.
- [2] P. Bellavista et al. Feasibility of fog computing deployment based on Docker containerization over RaspberryPi. In Proc. ICDCN, 2017.
- [3] Docker Inc. Docker: Build, ship, and run any app, anywhere. <https://www.docker.com/>.
- [4] Arif Ahmed and Guillaume Pierre. Docker Container Deployment in Fog Computing Infrastructures. In Proc. IEEE EDGE, July 2018.
- [5] <https://docs.docker.com/get-started/overview/>.
- [6] D. Patel, M. K. Patra and B. Sahoo, "Energy Efficient Genetic Algorithm for Container Consolidation in Cloud System," 2020 7th International Conference on Signal Processing and Integrated Networks (SPIN), 2020, pp. 1066-1071, doi: 10.1109/SPIN48934.2020.9071005.
- [7] Sultan, Sari, Imtiaz Ahmad, and Tassos Dimitriou. "Container security: Issues, challenges, and the road ahead." IEEE Access 7 (2019): 52976-52996.
- [8] Tomar, Aparna, Diksha Jeena, Preeti Mishra, and Rahul Bisht. "Docker security: A threat model, attack taxonomy and real-time attack scenario of dos." In 2020 10th International Conference on Cloud Computing, Data Science and Engineering (Confluence), pp. 150-155. IEEE, 2020.

- [9] A. Krasnov, R. R. Maiti and D. M. Wilborne, "Data Storage Security in Docker," 2020 SoutheastCon, 2020, pp. 1-1, doi:10.1109/SoutheastCon44009.2020.9249757.
- [10] W. S. Shameem Ahamed, P. Zavorsky and B. Swar, "Security Audit of Docker Container Images in Cloud Architecture," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), 2021, pp. 202-207, doi: 10.1109/ICSCCC51823.2021.9478100.
- [11] <https://geekflare.com/docker-architecture/>
- [12] C. Boettiger, "An introduction to Docker Reproducible Research with Examples from R Environment," 2014.
- [13] C. Tozzi, "Securing Docker containers" 216, <https://www.sumologic.com/blog/security/securing-docker-containers/>
- [14] E. Mavungu, "Docker storage: An introduction," 2017, <https://blog.codeship.com/docker-storage-introduction/>.