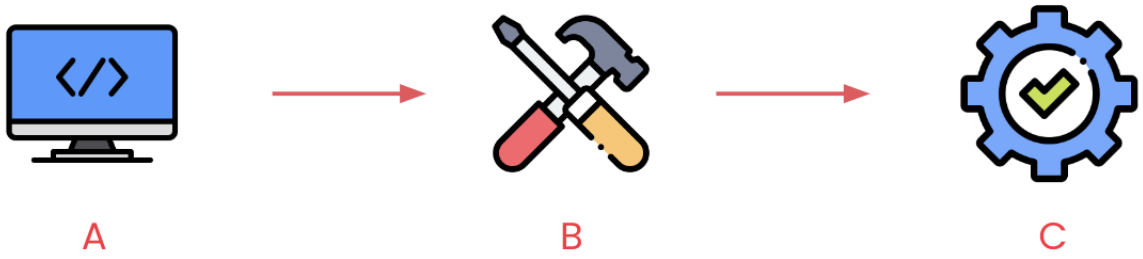
1. จงอธิบาย Process ของการทำ Application จากรูปที่ให้มาด้านล่าง (1 คะแนน)



The transformation of source code into an executable program that can run on an operating system.

A = (Source Code): The human-readable code that developers write to instruct the computer on what the program should do.

B = (Compile/Interpret): Convert the source code into machine code that the computer can understand and execute.

C = (Executable File): The output of the compilation or interpretation process. It's a binary file containing the machine code instructions for the program to run.

- **OS Executable:** (e.g., .exe)
- **Application Runtime:** (e.g., Java Virtual Machine - JVM)

2. หากเราจะสร้าง Web Application, Mobile Application, Application อย่างง่ายจะต้องมีส่วนประกอบอะไรบ้าง ให้บอกส่วนประกอบเหล่านั้นพร้อมคำอธิบาย (1 คะแนน)

Building a simple web application in Python

- **Application Runtime:** (python.exe)
    - Install Python on a system.
- **Dependencies Management:** (requirements.txt)
    - Create a file named `requirements.txt` in a project directory. This file lists all the Python libraries and packages that application depends on.
- **Source Code:** (app.py)
    - Create a Python file named `app.py` in the project directory. This file will contain a web application's code.

3. System(OS) Application ต่างจาก Application Package อย่างไร จงอธิบาย (1 คะแนน)

A **system application** is an integral part of the operating system and is pre-installed, while an **application package** refers to the files and resources that make up a software application that can be installed separately.

System applications are often essential for the core functionality of the device, while **application packages** are additional software that extends the device's capabilities.

4. คำถามเกี่ยวกับ VPC-VPC Communication จงตอบคำถามต่อไปนี้
4.1 ถ้าเราต้องการให้ VPC 2 อัน สามารถติดต่อสื่อสารกันได้ควรใช้เทคนิคอะไรเข้ามาช่วย(1 คะแนน)

VPC peering

4.2 จากข้อ 4.1 จงวาดรูปประกอบพร้อมเขียนวิธีเชื่อมต่อในรูปแบบตาราง (1 คะแนน)

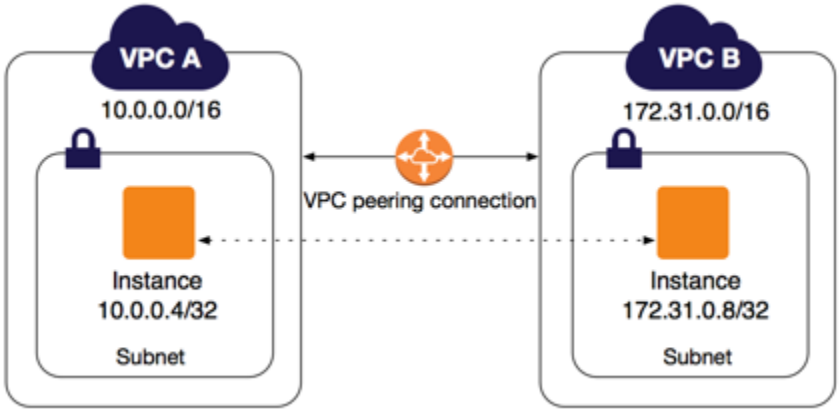
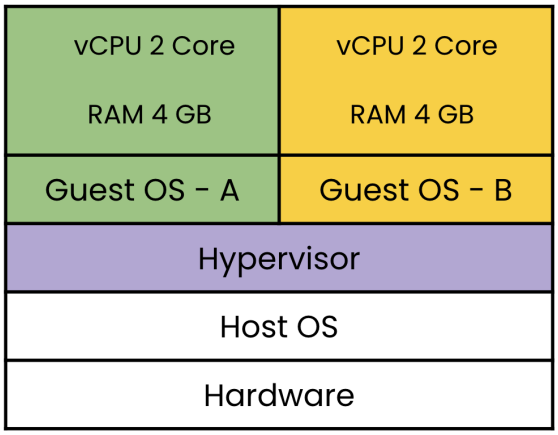
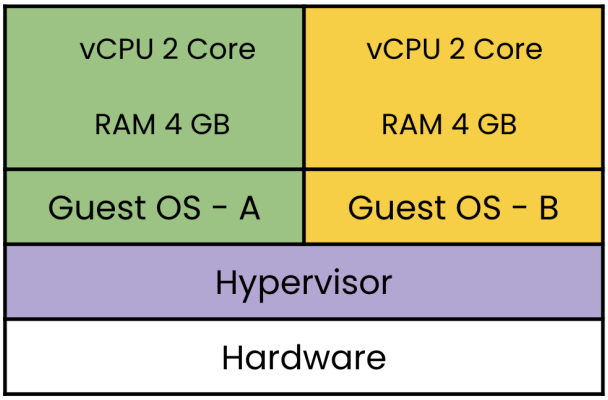


Image source: *(https://disaster-recovery.workshop.aws/images/vpc-peering-diagram.png)*

5. จงตอบคำถามต่อไปนี้

5.1 จากรูปที่กำหนดด้านล่าง จงระบุว่าภาพใดเป็น Hypervisor Type I และภาพใดเป็น Hypervisor Type II (1 คะแนน)





A. .....Type II..... B. .....Type I.....

5.2 Hypervisor แต่ละ type ต่างกันอย่างไร (1 คะแนน)

Type I hypervisors run directly on the hardware, providing high performance and are suitable for production environments where virtualization is a core requirement. **Type II hypervisors** run within a host operating system, are more user-friendly for desktop and development use, but may have slightly lower performance due to the additional layer of the host OS.

5.3 หากเราต้องการเป็นผู้ให้บริการ Cloud หรือที่เรียกว่า Cloud Service Provider ควรเลือกใช้ Hypervisor Type ใด เพราะเหตุใด (1 คะแนน)

Type I, because it runs directly on the physical hardware of the cloud provider's data centers. This architecture allows for better performance, security, and isolation between virtual machines (VMs) compared to Type II hypervisors.

6. WWW คืออะไร ย่อมาจากอะไร และใครเป็นผู้คิดค้น (1 คะแนน)

"**www**" stands for **"World Wide Web**." It is a system of interconnected documents and resources linked together by hyperlinks and URLs (Uniform Resource Locators). The World Wide Web is a key component of the internet, allowing users to access and navigate web pages and content hosted on servers around the world.

It was created by **Sir Tim Berners-Lee**, a British computer scientist.

7. จงตอบคำถามต่อไปนี้เกี่ยวกับ API
7.1 API คืออะไร ย่อมาจากอะไร (1 คะแนน)

API stands for "**Application Programming Interface**." It is a set of rules, protocols, and tools that allows different software applications to communicate with each other. APIs define the methods and data formats that applications can use to request and exchange information, making it easier for developers to integrate services and functionalities from one software component into another.

7.2 จงอธิบายความหมายของ Client กับ Server (1 คะแนน)

Client is a device or software application that requests and consumes services, resources, or data provided by another device or application called a server.

Server is a device or software application that listens for incoming requests from clients and responds by providing the requested services, resources, or data.

7.3 หากเราต้องการทำ API Server ควรมีความรู้ด้านใดบ้าง (1 คะแนน)

Programming Languages: Proficiency in a backend programming language is essential. Common choices include Python, JavaScript (Node.js), Ruby, Java, or C#.

Web Framework: Familiarity with web frameworks such as Express.js, Flask, Django, Ruby on Rails, or Spring Boot can help you streamline API development.

API Design: Understand API design principles, including RESTful or GraphQL design patterns. Knowing how to define endpoints, request/response formats, and authentication mechanisms is crucial.

Database Management: Know how to work with databases (e.g., SQL or NoSQL databases) and design database schemas that suit your API's requirements.

HTTP and Networking: Understand the HTTP protocol, status codes, and headers. Knowledge of network concepts like routing and load balancing can be beneficial.

Authentication and Authorization: Be able to implement secure authentication and authorization mechanisms to protect your API and its data.

Version Control: Proficiency with version control systems like Git to track and manage changes in your codebase.

Testing: Know how to write unit tests and perform integration testing to ensure the reliability of your API.

Documentation: Create clear and comprehensive API documentation for developers who will use your API.

Deployment and DevOps: Understand how to deploy your API to a production environment. Knowledge of containerization (e.g., Docker) and orchestration tools (e.g., Kubernetes) can be valuable.

8. จงเติมคำลงในช่องว่างในเรื่องของ HTTP Method (2 คะแนน)

HTTP Method	CRUD Operation	คำอธิบาย
GET	READ	Retrieve information or data from a resource.
POST	CREATE	Create a new resource or entity by submitting data to the server.
PUT	UPDATE	Replace or update the data of an existing resource with new data.
DELETE	DELETE	Remove a resource or entity from the server.

9. จงเติมคำลงในช่องว่างในเรื่องของ HTTP Status (2 คะแนน)

HTTP Status	คำอธิบาย
100-199	**1xx (Informational):** These status codes indicate that the server has received the request and is continuing to process it. They provide informational responses. For example, `**100 Continue**`
200-299	**2xx (Success):** These status codes indicate that the request was successfully received, understood, and accepted. The client's request was processed successfully. For example, `**200 OK**`
300-399	**3xx (Redirection):** These status codes indicate that further action is needed by the client to complete the request. They are often used for redirection or navigation purposes. For example, `**301 Moved Permanently**`
400-499	**4xx (Client Error):** These status codes indicate that the client seems to have made an error or the request cannot be fulfilled due to client-side issues. For example, `**404 Not Found**`
500-599	**5xx (Server Error):** These status codes indicate that the server encountered an error or was unable to fulfill a valid request from the client. For example, `**500 Internal Server Error**`

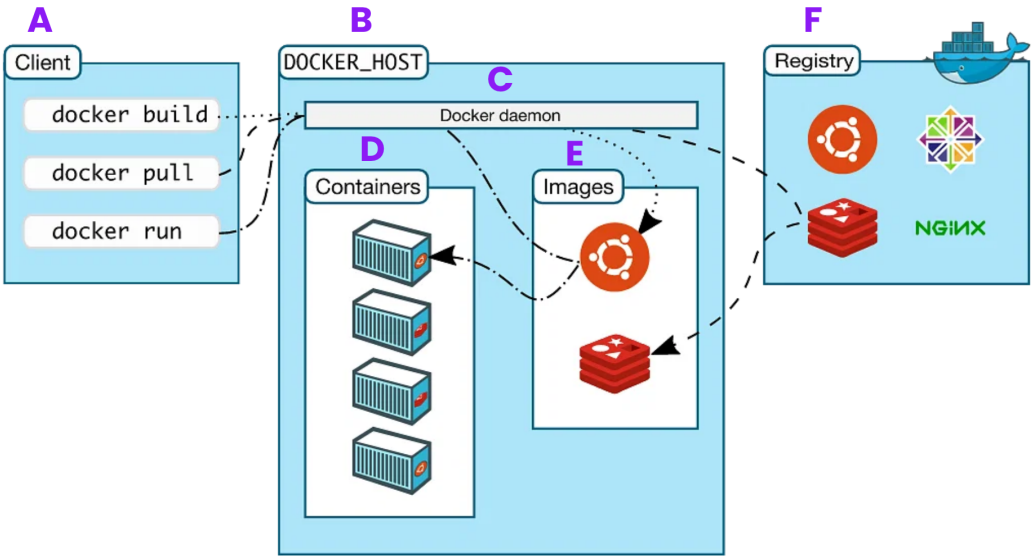
10. จงตอบคำถามต่อไปนี้เกี่ยวกับ Container Technology
10.1 Container เข้ามาแก้ปัญหาด้านใดในอุตสาหกรรมการจัดการ Application (1 คะแนน)

Containers simplify software development, improve application deployment and scalability, and enhance the consistency and reliability of applications across diverse environments. They have become a cornerstone technology for modern application development and deployment practices.

10.2 ตัวที่ทำให้ Container สามารถไปรันที่ไหนก็ได้ชื่อว่าอะไร (1 คะแนน)

Container runtime, such as Docker

11. จงอธิบายหน้าที่ตามตัวอักษรที่กำหนดให้จากรูปภาพด้านล่าง (3 คะแนน)



ตัวอักษร	เรื่อง	คำอธิบาย
A	**Client**	- The Docker client is the command-line interface (CLI) or graphical user interface (GUI) that allows users to interact with Docker. It sends commands to the Docker daemon, which then performs the requested actions. - Users use the Docker client to build, manage, and control containers and images, as well as interact with the Docker host.

B	**Docker Host**	- The Docker host is the physical or virtual machine (VM) where the Docker daemon runs. It's responsible for managing containers and executing Docker commands.
C	**Docker Daemon**	- The Docker daemon is a background service or process that runs on the Docker host. It is responsible for building, running, and managing containers. - The Docker daemon listens for Docker client requests and carries out the requested container operations, such as creating, starting, stopping, or deleting containers.
D	**Containers**	- Containers are lightweight, standalone, and executable packages that include everything needed to run a piece of software, including the code, runtime, system tools, and libraries. - Containers are isolated from one another and share the host OS kernel, making them efficient and portable. They can be easily deployed across different environments.
E	**Images**	- Docker images are read-only templates or blueprints for creating containers. They are the basis for running containers. - Images include application code, libraries, and dependencies, and they are versioned. - Users can create, modify, and share images to facilitate the creation of consistent and reproducible containers.
F	**Registry**	- A Docker registry is a repository for storing and sharing Docker images. The most well-known Docker registry is *Docker Hub*, which is a public registry that hosts a vast collection of Docker images. - Users can push (upload) their custom images to a Docker registry and pull (download) images from it to use in their Docker environments.

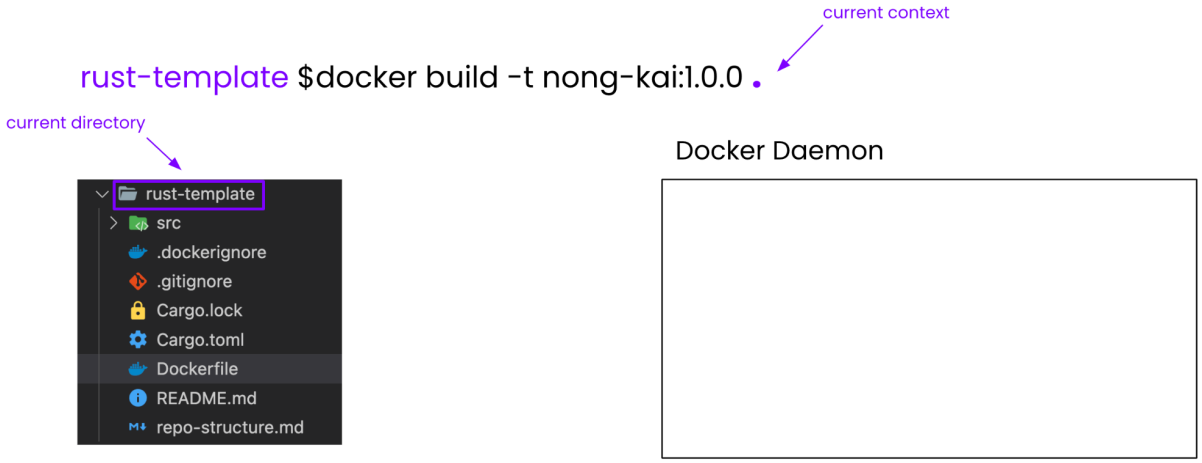
12. จงตอบคำถามต่อไปนี้เกี่ยวกับ Container

12.1 จากคำสั่ง docker build -t <name>:<tag> **<context>** ซึ่งเป็นคำสั่งที่ใช้สำหรับสร้าง Container image จาก Dockerfile และ Context จงเติมคำอธิบายในส่วนที่เกี่ยวข้องลงในช่องว่างให้ถูกต้อง (1 คะแนน)

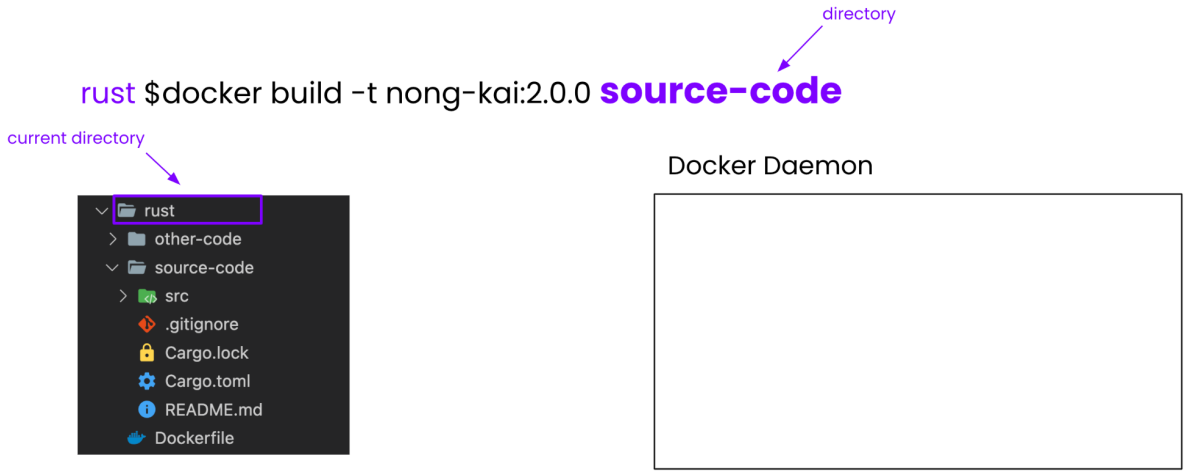
ส่วนที่เกี่ยวข้อง	อธิบายว่าคืออะไร ไว้ใช้ทำอะไร
Dockerfile	A Dockerfile is a text file used to define a set of instructions for building a Docker image. It specifies the base image, sets environment variables, installs software packages, copies files into the image, and configures various aspects of the containerized application. In essence, a Dockerfile serves as a blueprint for creating Docker images.
Context	The <context> refers to the build context. The build context is the directory or path on your local file system where Docker looks for the files and directories that are used as input for building a Docker image. This context is typically specified as the last argument in the docker build command.

12.2 จากข้อที่ 12.1 จงวาดรูปในส่วนที่หายไปจากภาพด้านล่าง (2 คะแนน)

Scenario A: Context is current directory (.)



Scenario B: Context is a directory (rust)



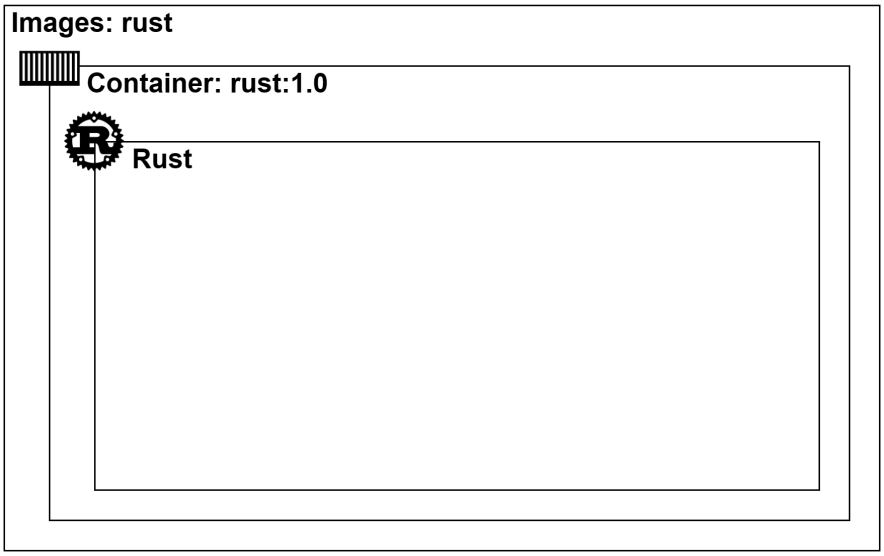
12.3 จงนำ instruction (FROM, WORKDIR, COPY, ...) เติมลงในตารางให้ถูกต้อง (1 คะแนน)

```
FROM rust

WORKDIR /app

COPY . .

RUN cargo build --release

EXPOSE 8080

ENTRYPOINT [ "./target/release/web-server" ]
```

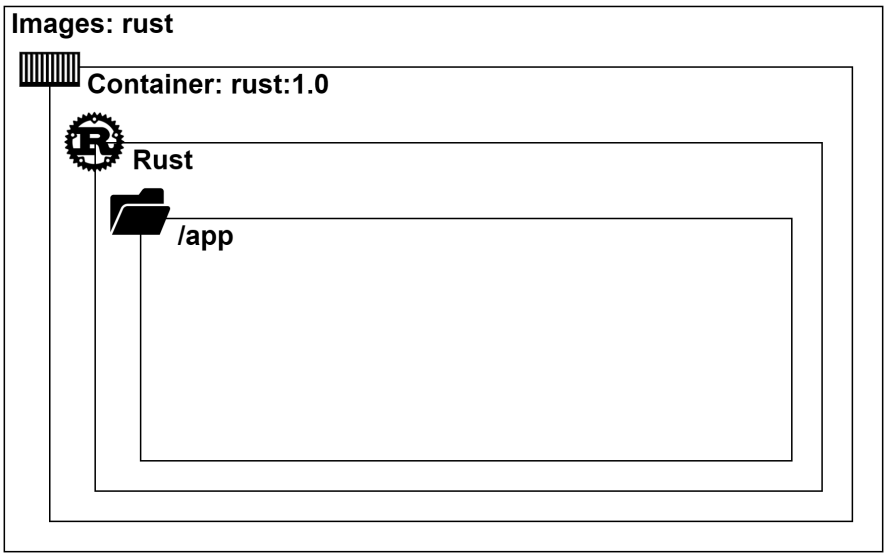
Stage	Instruction
Build time (Building Image)	FROM rust WORKDIR /app COPY . . RUN cargo build --release EXPOSE 8080
Runtime (Container Running)	ENTRYPOINT ["./target/release/web-server"]

12.4 จากรูปภาพ 12.3 จงวาดรูปวิธีการทำงานในแต่ละ Instruction ให้ถูกต้อง (Extra 5 คะแนน)

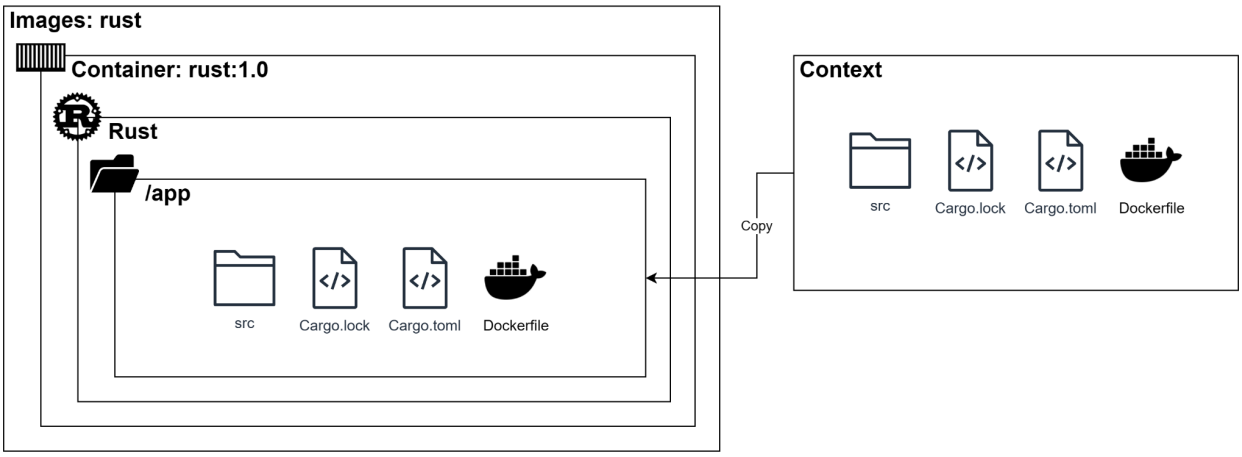
FROM rust



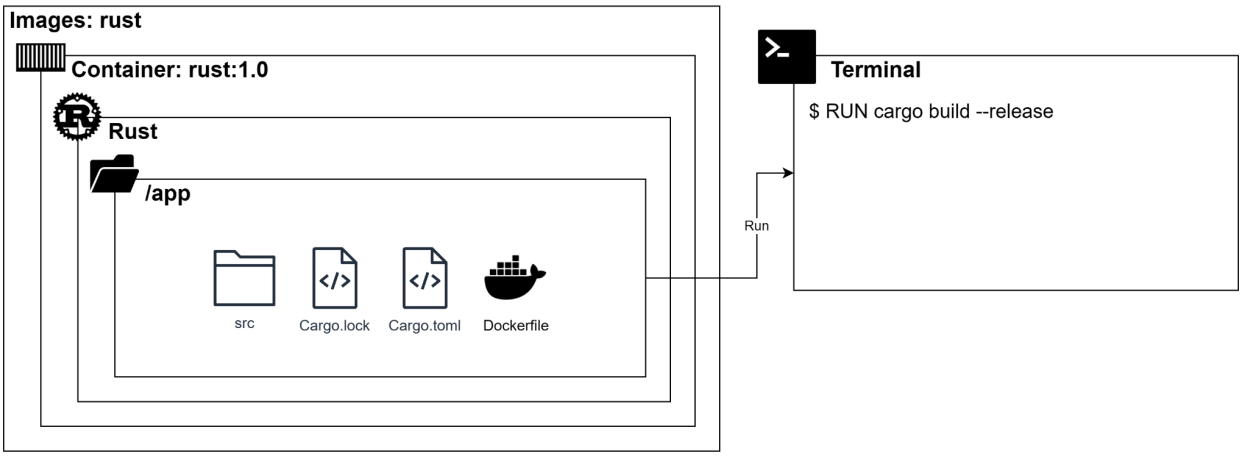
WORKDIR /app



COPY . .



RUN cargo build --release



EXPOSE 8080

ENTRYPOINT ["./target/release/web-server"]