ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO BÀI TẬP CUỐI KỲ
KHAI PHÁ DỮ LIỆU

ĐỀ TÀI: PHÁT TRIỂN PHẦN MỀM QUẢN LÝ TÁC VỤ CHO CREATIVE AGENCY

Ngày 10 tháng 9 năm 2022

ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO BÀI TẬP CUỐI KỲ
KHAI PHÁ DỮ LIỆU

ĐỀ TÀI: PHÁT TRIỂN PHẦN MỀM QUẢN LÝ TÁC VỤ CHO CREATIVE AGENCY

Giảng viên: Lê Hoàng Quỳnh

Sinh viên: Trần Công Việt An - 19020032

Ngày 10 tháng 9 năm 2022

Lời cam đoan

Em xin cam đoan rằng bản báo cáo này hoàn toàn được hoàn thành dựa trên sự cố gắng, nỗ lực của bản thân.

Báo cáo này hoàn toàn không sao chép từ bất kỳ một nguồn nào khác. Nếu phát hiện có sự gian lận, em xin chịu hoàn toàn trách nhiệm.

Trần Công Việt An

Mục lục

Danh sách hình vẽ

1 Lời nói đầu

Lời đầu tiên, em xin gửi lời tri ân chân thành đến Công ty Cổ phần Công nghệ Young IT - Quý công ty đã tạo cơ hội cho em tham gia thực tập tại một môi trường cởi mở, chuyên nghiệp. Em rất ấn tượng với phong cách làm việc của Quý công ty: Lắng nghe - thấu hiểu - hợp tác - và hành động. Sau 4 tuần kiến tập, được tiếp xúc với văn hoá cũng như cách làm việc của Công ty, em đã trau dồi thêm nhiều kỹ năng, bài học thực tế vô cùng đáng giá. Xin cảm ơn quản lý trực tiếp và người hướng dẫn Trần Long Vũ, bạn đồng hành Đồng Minh Tiến đã hỗ trợ em trong quá trình thực tập ở công ty.

Em cũng xin được cảm ơn Khoa Công nghệ thông tin, Trường Đại Học Công Nghệ, Đại Học Quốc Gia Việt Nam đã tạo điều kiện cho chúng em có 2 tháng thực tập để thu nạp những trải nghiệm quý báu, chân thực nhất. Em tin rằng chúng em đã học được rất nhiều kiến thức, va chạm nhiều bài học xã hội - đây là cơ hội tốt để chúng em phát triển và xác định rõ hơn con đường muốn theo đuổi sau này. Em xin cảm ơn các thầy cô trong khoa đã luôn sát sao, hỗ trợ nhiệt tình để kỳ thực tập của chúng em diễn ra thành công, tốt đẹp nhất. Em xin cảm ơn cô Lê Hoàng Quỳnh đã hướng dẫn và chỉ bảo nhiệt tình giúp em hoàn thành công việc.

Kính chúc Quý Công ty Cổ phần Công nghệ Young IT và các thầy cô Khoa Công nghệ thông tin một sức khoẻ trọn vẹn, hạnh phúc và thành công!

Em xin chân thành cảm ơn!

2 Giới thiệu công ty

2.1 Tổng quan về đơn vị thực tập Young IT

Công ty Cổ phần Công nghệ Young IT là đơn vị tiên phong tại Việt Nam trong lĩnh vực phát triển nhân lực Công nghệ Thông tin trẻ. Với sứ mệnh cách mạng hóa nhân lực IT của Việt Nam, chúng tôi cung cấp các hoạt động cộng đồng, dịch vụ giáo dục, cung ứng nhân sự và dịch vụ phát triển phần mềm cho các doanh nghiệp trong và ngoài nước.

2.2 CitizenDev

CitizenDev, một thương hiệu con của Công ty Cổ phần Công nghệ Young IT, là thương hiệu tiên phong trong làn sóng "Kỷ nguyên toàn dân lập trình" tại Việt Nam, hướng tới việc phổ cập công nghệ No-code/Low-code giúp tất cả mọi người đều có thể bắt đầu phát triển sản phẩm công nghệ mà không cần phải sử dụng đến phương pháp lập trình truyền thống.

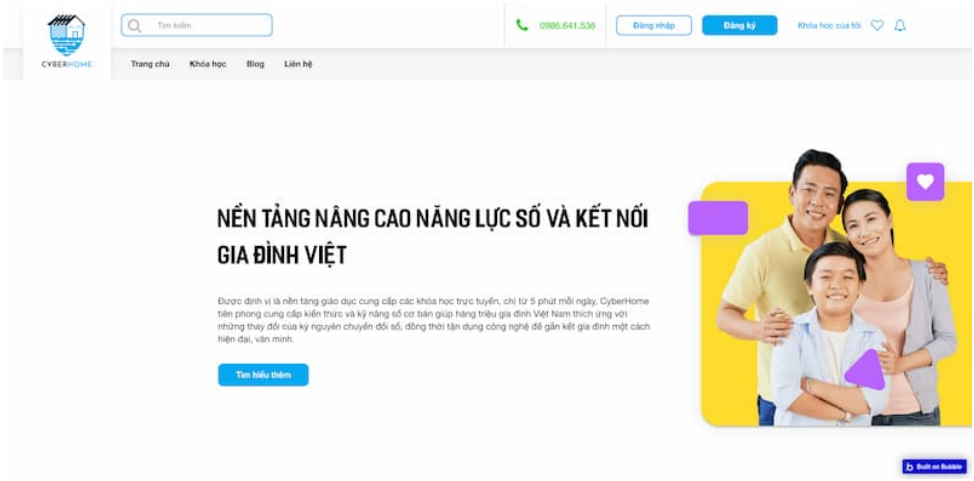
Hiện tại CitizenDev là thương hiệu trung tâm cho các hoạt động của Young IT, giúp Young IT đẩy mạnh định hướng xây dựng nhân lực Công nghệ Thông tin trẻ.

2.3 Đối tác và các dự án của Young IT - CitizenDev

Dưới cả hai thương hiệu CitizenDev và Young IT, công ty đã và đang hợp tác với nhiều doanh nghiệp, tổ chức trong và ngoài nước. Một số doanh nghiệp, tổ chức nổi bật nhất có thể kể đến là: CyberKid Việt Nam, Markdao Agency, MVV Academy.

2.4 Dịch vụ của Young IT - CitizenDev

- Phát triển, gia công phần mềm, thiết kế hệ thống phần mềm dựa trên nhu cầu của khách hàng doanh nghiệp.
- Dịch vụ tư vấn chuyển đổi số cho doanh nghiệp.
- Phát triển Thương hiệu CitizenDev: Trung tâm đào tạo và phát triển công nghệ low-code/no-code cho cá nhân và doanh nghiệp.



Hình 2.1: CyberHome - Dự án giáo dục gia đình về an toàn trên không gian mạng.

3 Cơ sở lý thuyết

3.1 Web App

Web App (ứng dụng mạng) là một ứng dụng chạy trên nền tảng web, sử dụng trình duyệt và công nghệ web để thực hiện các thao tác trực tiếp qua Internet.

- Mã nguồn của Web App được lưu trữ ở máy chủ hoặc mạng lưới cung cấp tài nguyên (CDN) và phân phối đến người dùng qua Internet.
- Dữ liệu cũng như các logic tính năng quan trọng của Web App được lưu trữ ở máy chủ. Mã nguồn của Web App sau khi được phân phối đến trình duyệt của người dùng sẽ có thể truy xuất dữ liệu thông qua API của Web App. Đây được gọi là **Backend** của Web App
- Bạn sẽ thao tác với các chức năng của web app qua giao diện của trình duyệt web. Các tính năng về mặt tương tác người dùng sẽ được thực hiện ở trong môi trường của trình duyệt. Đây được gọi là **Frontend** của Web App.

Đa số các trang web hiện nay đều là web app. Mô hình Web App cho phép xây dựng ứng dụng phân tán, nhiều người dùng với thời gian phát triển ngắn hơn và dễ dàng hơn, dựa vào sự chuẩn hóa của các tiêu chuẩn web cũng như sự phát triển của các công cụ xây dựng Web.

3.2 Backend - Frontend

Cơ chế hoạt động của các web application hiện nay hầu hết được xây dựng trên kiến trúc 3 tầng, với nhiệm vụ được phân hóa tách biệt nhau:

- Tầng dữ liệu: Đây là nơi dữ liệu của ứng dụng được lưu trữ. Các thành phần phần mềm ở đây bao gồm các hệ quản trị cơ sở dữ liệu.
- Tầng logic: Đây là nơi dữ liệu ứng dụng được xử lý và tính toán. Các logic doanh nghiệp được triển khai dưới dạng mã nguồn ở tầng này. Tầng logic giao tiếp với tầng dữ liệu thông qua giao thức của các hệ quản trị cơ sở dữ liệu.
- Tầng trình bày: Đây là nơi người dùng tương tác với ứng dụng. Người dùng thực hiện các thao tác với ứng dụng thông qua giao diện đồ họa được xây dựng trên tầng này. Tầng trình bày tương tác với tầng logic thông qua các giao thức được định nghĩa bởi tầng logic, gọi chung là API.

Kiến trúc 3 tầng có thể được áp dụng cho nhiều kiểu ứng dụng khác nhau, tuy nhiên ứng dụng Web là loại ứng dụng phổ biến nhất sử dụng loại kiến trúc này. Với ứng dụng web, kiến trúc 3 tầng được triển khai trên mô hình tương tác máy chủ - máy khách, trong đó hai tầng dữ liệu và logic được triển khai trên một (cụm) máy chủ và mã nguồn của tầng trình bày được phân phối đến thiết bị của người dùng đầu cuối. Hình 3.1 mô tả kiến trúc 3 tầng

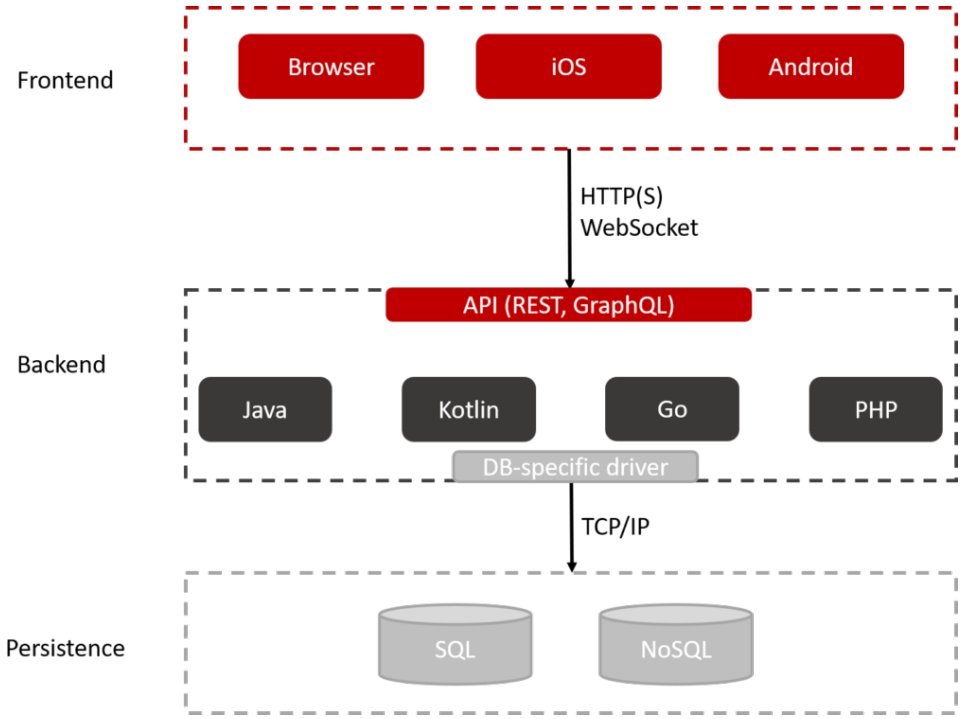
Hai tầng logic và dữ liệu được gọi chung là Backend của một ứng dụng Web (phần người dùng cuối không tương tác trực tiếp). Tầng trình bày của ứng dụng được gọi là Frontend (phần người dùng cuối tương tác trực tiếp).

3.2.1 Backend

Đối với hầu hết các ứng dụng web hiện nay, Backend là một chương trình máy chủ HTTP chạy trên (cụm) máy chủ của ứng dụng. Backend cung cấp các đường dẫn HTTP để thực hiện các thao tác đối với hệ thống, được gọi là HTTP endpoint. Tập hợp các HTTP endpoint tương ứng với một dịch vụ được cung cấp bởi Backend được gọi là một giao diện lập trình ứng dụng (API) Các HTTP endpoint cần tuân theo một quy tắc nhất định về định dạng dữ liệu cần được truyền đến, phương thức HTTP sử dụng, các chỉ dẫn HTTP Header cần được cung cấp đính kèm và định dạng dữ liệu mà HTTP sẽ trả về. Như vậy, HTTP API là một lời gọi hàm từ xa - Remote Procedural Call, với mỗi HTTP endpoint là một hàm được định nghĩa bởi chương trình Backend (xem Hình 3.2)

Các bộ chuẩn quy tắc cho HTTP endpoint phổ biến nhất hiện nay bao gồm:

- REST API: Chuẩn quy tắc này xoay quanh việc thao tác với các dữ liệu dưới góc nhìn các đối tượng. REST API phù hợp với những kiến trúc mang tính chất hướng đối tượng (Object Oriented)



Hình 3.1: Kiến trúc 3 tầng

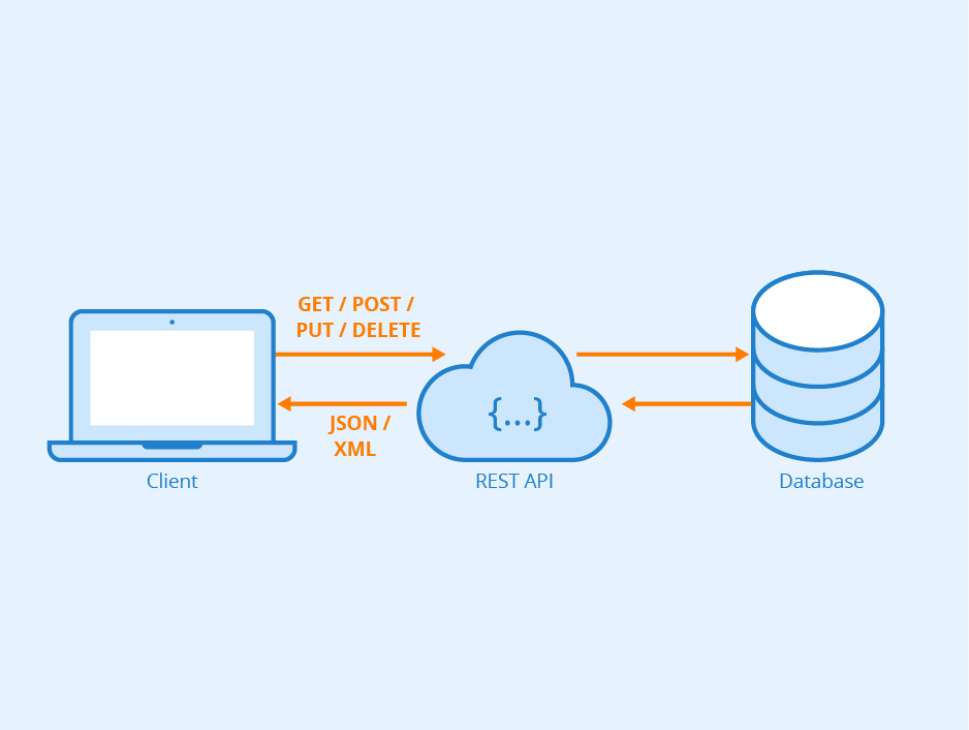
- GraphQL: Chuẩn quy tắc này xoay quanh việc truy xuất dữ liệu dưới dạng đồ thị. GraphQL cho phép tương tác hiệu quả với những ứng dụng có cấu trúc quan hệ phức tạp như mạng xã hội.

Việc xây dựng kiến trúc Backend dưới dạng API cho phép nhà phát triển xây dựng nhiều tầng trình bày khác nhau cho một ứng dụng, ví dụ một giao diện web trên trình duyệt và một ứng dụng di động.

3.2.2 Frontend

Các ứng dụng ở tầng trình bày được gọi chung là Frontend, có nghĩa là phần ứng dụng mà người dùng tương tác trực tiếp. Frontend có thể là một chương trình được viết bằng HTML/CSS/JS để được thực thi trong trình duyệt (web frontend) hoặc là một ứng dụng chạy trực tiếp trên hệ điều hành của thiết bị (mobile frontend). Đối với nội dung của bài báo cáo này, Frontend được ngầm định hiểu là Web Frontend.

Mã nguồn của Frontend có thể được xây dựng trực tiếp bằng HTML/CSS/JS, tuy nhiên phương pháp này có thể trở nên khó tiếp cận đối với những ứng dụng Frontend lớn. Cộng đồng phát triển Frontend đã xây dựng những Framework lớn dựa trên JS như Vue.js, React.js,... để dễ dàng xây dựng những ứng dụng phức tạp này. Tuy nhiên trong khi triển khai sử dụng thực tế, mã nguồn của những Framework này sẽ cần được dịch



Hình 3.2: HTTP API

lại trở thành các ngôn ngữ web cơ bản HTML/CSS/JS để có thể thực thi được trên các trình duyệt Web của người dùng cuối.

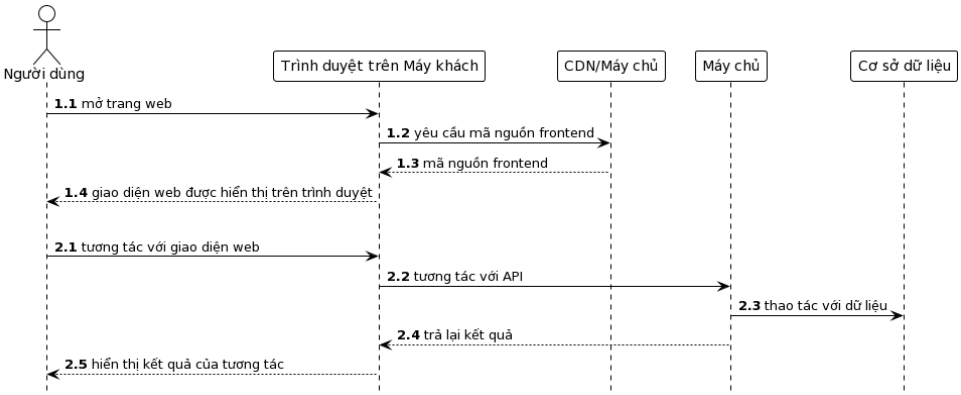
Khối mã này sẽ được cung cấp đến người dùng cuối bởi chính (cụm) máy chủ chứa Backend hoặc bởi một mạng lưới cung cấp nội dung - Content Delivery Network (CDN). Quá trình tương tác này được mô tả chung bằng hình 3.3

Việc xây dựng ứng dụng Frontend tập trung vào các công việc liên quan đến thiết kế đồ họa và tương tác người dùng. Các thao tác đối với dữ liệu và logic thường không được thực thi trên Frontend, việc thực hiện chúng sẽ do Frontend thực thi API tương ứng của Backend.

3.3 Containerized System - Orchestration

3.3.1 Containerization

Việc triển khai ứng dụng trong môi trường ứng dụng thực tế từng là một công việc vô cùng khó khăn. Một trong những vấn đề mà quản trị viên hệ thống từng thường xuyên gặp phải là việc mã nguồn hoạt động đúng trên thiết bị của lập trình viên/nhân viên kiểm soát chất lượng, nhưng gặp lỗi trên hệ thống thực thi thực tế của ứng dụng. Lý do cho điều này là sự khác biệt giữa môi trường thực thi mã nguồn, như hệ điều hành, các thư



Hình 3.3: Sơ đồ quá trình tương tác cơ bản giữa người dùng và ứng dụng web

viện hệ thống mà chương trình phụ thuộc vào, các biến dữ liệu môi trường (environment variables).

Containerization - tạm dịch "Đóng gói chương trình triển khai ứng dụng trong các môi trường gần tương tự với máy ảo gọi là Container. Các chương trình được thực hiện trong container được tách biệt khỏi hệ thống chủ, có thể tương tác với nhau thông qua giao thức mạng TCP/UDP được duy trì bởi hệ thống quản lý Container.

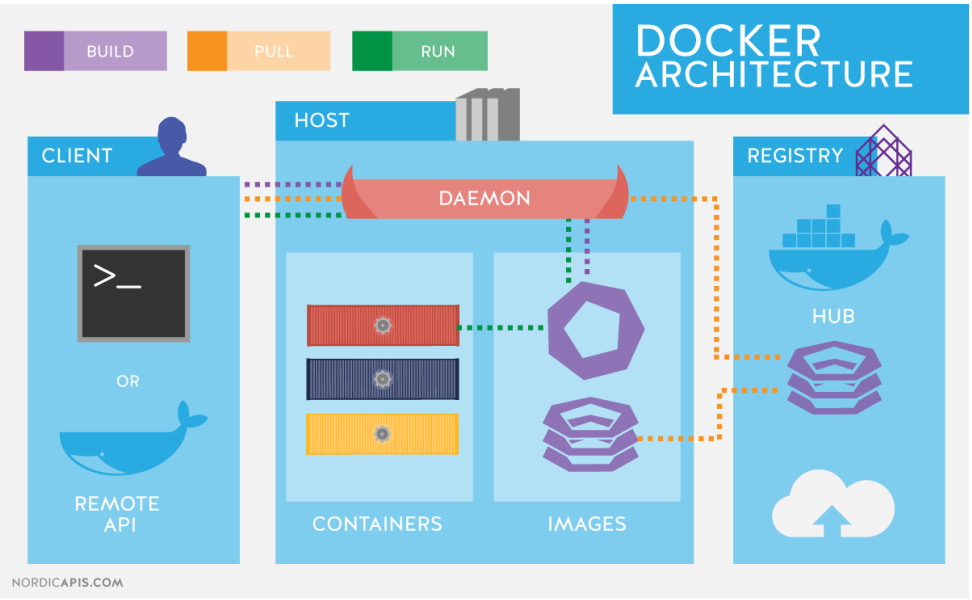
Việc khởi tạo một container được thực hiện bằng cách đọc một bản mô tả của Container, được gọi là Image. Image chứa đầy đủ thông tin của môi trường thực thi và mã nguồn chương trình. Dựa vào đặc điểm này, các Container được chạy từ cùng một Image sẽ được thực thi trong cùng một điều kiện tương đương nhau. Nhờ đó, các nhà phát triển có thể đảm bảo rằng chương trình sau khi được đóng gói thành một Image và được kiểm thử với Image đó sẽ thực thi chính xác ở trên môi trường thực tế.

Các Image cũng có thể được chia sẻ và tiếp tục mở rộng bằng việc ghi thêm các mô tả về mã nguồn và môi trường thực thi. Như vậy, các nhà phát triển có thể sử dụng Image chứa trình thực thi Node.JS và ghi thêm mã nguồn riêng của chương trình của họ, thay vì phải xây dựng một Image có cả hai từ đầu. Các Image sẵn có được phân phối thông qua Registry.

Hệ thống quản lý Container phổ biến nhất hiện nay là Docker. Hình 3.4 mô tả kiến trúc của Docker.

3.3.2 Orchestration

Việc đóng gói mã nguồn thành một định dạng chạy được ngay lập tức của cơ chế Containerization cho phép chúng ta có thể tự động hóa công việc triển khai hệ thống. Đây là



Hình 3.4: Kiến trúc của Docker.

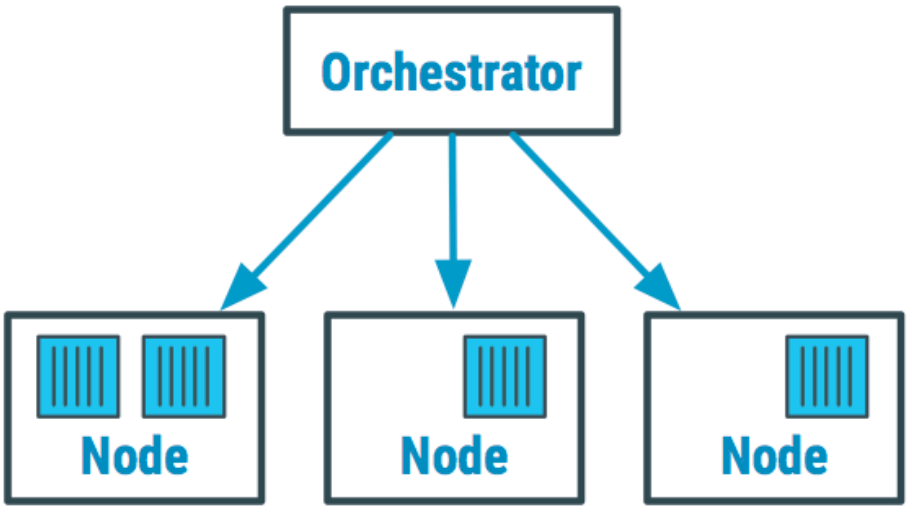
mục đích chính của các phần mềm Container Orchestration. Hệ thống Container Orchestration đảm nhiệm:

- Tự động hóa công việc khởi động các Container cần thiết, dọn dẹp các Container không còn được sử dụng, khởi động lại những Container gặp lỗi.
- Điều khiển thao tác mở rộng ngang cho một chương trình: tự động tạo các Container từ cùng một Image.
- Phân bổ khối lượng tính toán giữa các Container được mở rộng ngang.
- Quản lý giao tiếp giữa các Container.

Đặc biệt, một hệ thống Container Orchestration có thể điều khiển một cụm nhiều máy chủ, tự động phân bố các Container giữa các máy chủ (gọi là các Node trong cụm máy chủ) và đóng vai trò trung gian cho phép các Container trên máy chủ khác nhau giao tiếp được với nhau. Hình 3.5 thể hiện kiến trúc của một hệ thống Container Orchestration. Các hệ thống Container Orchestration cho phép đội ngũ vận hành hệ thống quản lý một cụm máy chủ lớn một cách tương đối đơn giản.

Hệ thống Container Orchestration phổ biến nhất hiện nay là Kubernetes (viết tắt k8s), một phần mềm mã nguồn mở được xây dựng và phát triển bởi Google. Hình 3.6 thể hiện cấu trúc của một hệ thống k8s.

Một hệ thống k8s bao gồm hai thành phần chính: Control plane và Nodes. Control plane là một tập các chương trình con tương tác với nhau nhằm điều khiển các Nodes



Hình 3.5: Hệ thống Container Orchestration

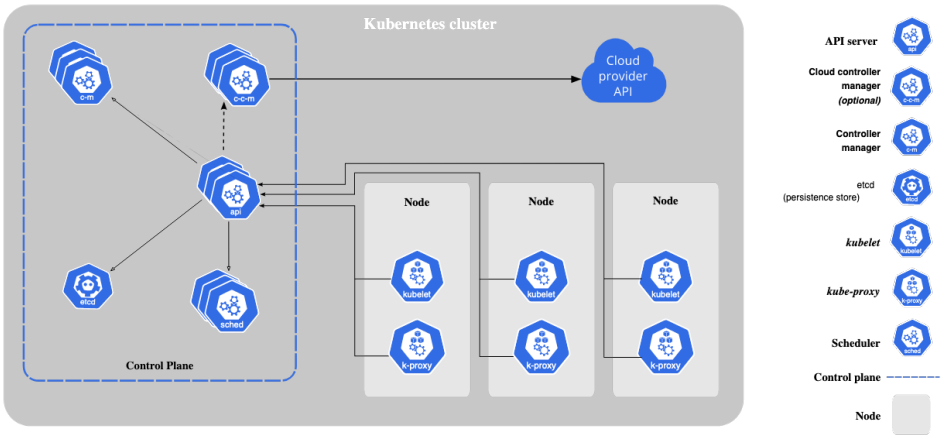
trong mạng. Những chương trình này có thể được cài đặt trên cùng một máy chủ vật lý, hoặc phân tán ra một số máy chủ vật lý đối với những cụm máy chủ lớn. Nodes là những máy chủ kết nối vào hệ thống k8s và thực hiện các công việc tính toán thực tế.

3.4 Cloud Computing

Khái niệm Cloud Computing không còn mới. Điện toán đám mây là từ khóa chung dành cho các dịch vụ cung cấp năng lực tính toán như máy chủ, hệ thống mạng, cơ sở lưu trữ dữ liệu,... Điện toán đám mây cho phép người sử dụng xây dựng các hệ thống phần mềm lớn mà không cần thực hiện công việc quản lý hạ tầng. Các dịch vụ đám mây lớn hiện tại còn cung cấp những giải pháp hệ thống phần mềm nói chung trên hạ tầng của họ, như dịch vụ AI, dịch vụ quản trị cơ sở dữ liệu, web hosting,... Nhà cung cấp dịch vụ điện toán đám mây lớn nhất hiện nay là Amazon Web Service.

3.4.1 EKS - Elastic Kubernetes Service

Amazon Web Service cung cấp dịch vụ vận hành một hệ thống Kubernetes trên hạ tầng của họ. EKS cho phép lập trình viên triển khai chương trình của họ trên một hệ thống k8s mà không cần phải thực hiện công việc cài đặt một hạ tầng cụm máy chủ. Hình 3.7 cho thấy sự đơn giản của việc sử dụng hệ thống k8s được quản lý bởi Amazon

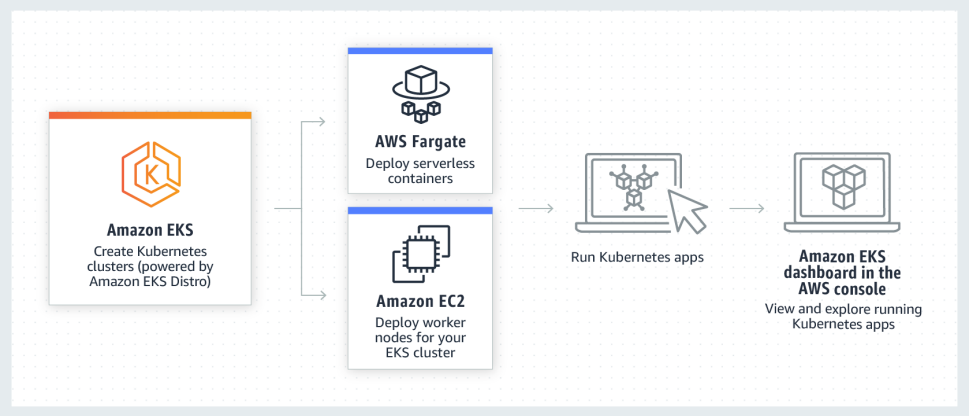


Hình 3.6: Các thành phần của một hệ thống Kubernetes

4 Nội dung dự án

4.1 Tóm tắt dự án - Project Charter

- **Tên dự án:** Dazzy (tên dự án trong báo cáo đã được thay đổi theo yêu cầu của khách hàng)
- **Ngày triển khai:** 02/07/2022
- **Mục đích của dự án:** Xây dựng một hệ thống quản lý tác vụ cho một công ty Creative Agency. Hệ thống cần hỗ trợ các tính năng cơ bản cho công việc này bao gồm:
  - Quản lý dự án, các tác vụ của dự án, quản lý lịch cho các tác vụ này.
  - Giao tiếp, nhắn tin.
  - Quản lý công việc cá nhân.
- **Hạng mục của dự án:**
  - 05/07/2022: Các tài liệu thiết kế của dự án.
  - 12/07/2022: Hệ thống web cơ bản với các tính năng liên quan đến đăng nhập, quản lý tài khoản.
  - 26/07/2022: Tính năng quản lý dự án, quản lý tác vụ, quản lý lịch, thông báo.
  - 09/08/2022: Tính năng chỉnh sửa, bình luận cho dự án, tác vụ.
  - 22/08/2022: Quản lý tài khoản cá nhân.



Hình 3.7: Amazon EKS

  - 30/08/2022: Hoàn thiện dự án.

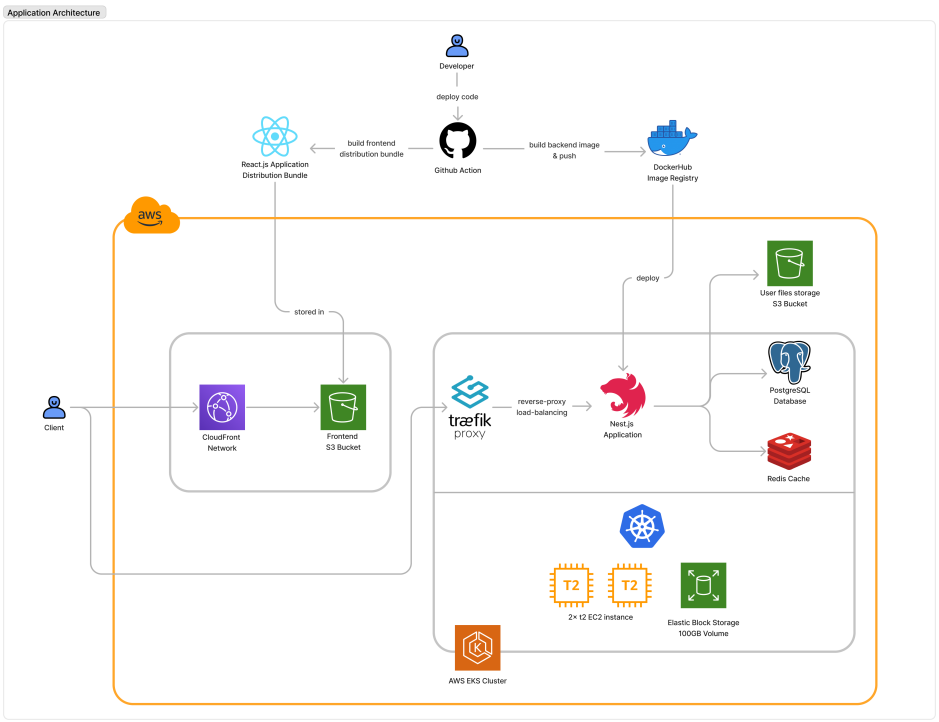
- **Yêu cầu chung cho dự án:**
  - Dễ sử dụng: người dùng không có trình độ công nghệ cao.
  - Chính xác về mặt thời gian: Các hệ thống thông báo cần xảy ra ít lỗi nhất có thể, tránh làm ảnh hưởng đến các dự án.

Hình 4.1 mô tả kiến trúc phần mềm của dự án.

4.2 Nội dung công việc

Đối với dự án trên, em đã thực hiện các công việc:

- Tham gia xây dựng kiến trúc hệ thống.
- Xây dựng API quản lý tác vụ cho các dự án trên hệ thống.
- Xây dựng module thông báo cho dự án, sử dụng RabbitMQ để phân phối tải lượng, tương tác với các dịch vụ bên thứ 3 như SendGrid, Firebase Cloud Messaging để gửi thông báo và email.
- Đóng gói (Containerize) hệ thống phần mềm và triển khai nó trên Kubernetes cluster của dự án.



Hình 4.1: Cấu trúc dự án Dazzy

5 Kết quả thực tập

5.1 Về chuyên môn

Kết thúc quá trình thực tập, em đã:

- Tham gia xây dựng thành công một dự án web application với người dùng thực tế.
- Sử dụng các công nghệ như Kubernetes, Nest.js, Docker và làm quen với các dịch vụ của Amazon Web Services.
- Xây dựng và tối ưu luồng phát triển ứng dụng: các quy trình tự động hóa CI/CD.

5.2 Về kỹ năng làm việc

Về quy trình xây dựng phần mềm, em được học các phương pháp làm việc nhóm trong đội ngũ phát triển phần mềm như Scrum, quy trình tạo merge request cho tính năng, kỹ thuật test driven development.

Ngoài những nội dung kỹ thuật, em cũng được học cách làm việc, giao tiếp hiệu quả, cách trình bày và giải quyết vấn đề, cách giao tiếp và làm việc với khách hàng.

6 Đánh giá

6.1 Ý kiến đánh giá của đại diện công ty

Em Trần Công Việt An là một thực tập sinh có khả năng giải quyết vấn đề: biết suy nghĩ phản biện, độc lập, sáng tạo; biết thu thập thông tin từ nhiều nguồn khác nhau để xử lý vấn đề, tinh thần học hỏi cao. Ngoài ra, bạn còn đáp ứng cả chất lượng và khối lượng công việc; hoàn thành đúng thời hạn; đặt chất lượng công việc lên hàng đầu. Thái độ làm việc chuẩn mực, cầu tiến, luôn chan hòa với cấp trên, đồng nghiệp, đối tác.

Hà Nội, Ngày ... tháng ... năm 2022
Người đại diện

6.2 Ý kiến đánh giá của giảng viên hướng dẫn

Hà Nội, Ngày ... tháng ... năm 2022
Giảng viên hướng dẫn