

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT VINH LONG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**  
**MÔN: ĐIỆN TOÁN Đám MÂY**

**ĐỀ TÀI: TÌM HIỂU DOCKER VÀ DOCKER SWARM**  
**TRONG XÂY DỰNG DỊCH VỤ HƯỚNG**  
**MICRO-SERVICES**

Người thực hiện: Lê Nguyễn Quang Bình - 21022010

Huỳnh Bảo Thắng - 21022019

Giáo viên hướng dẫn: ThS. Trần Phan An Trường

GV. Lê Thị Mỹ Tiên

Vinh Long, 2023

## NHẬN XÉT & ĐÁNH GIÁ ĐIỂM CỦA NGƯỜI HƯỚNG DẪN

Ý thức thực hiện: .....

.....

.....

.....

Nội dung thực hiện: .....

.....

.....

.....

Hình thức trình bày: .....

.....

.....

.....

Tổng hợp kết quả: .....

.....

.....

.....

☐ Tổ chức báo cáo trước hội đồng

☐ Tổ chức chấm thuyết minh

Vĩnh Long, ....ngày....tháng...năm....

Người hướng dẫn

(Ký và ghi rõ họ tên)

## **LỜI CAM ĐOAN**

Nhóm em cam đoan rằng bài báo cáo này được lập dựa trên các nguồn thông tin đáng tin cậy và đã được kiểm tra tính chính xác của số liệu, thông tin và tài liệu được sử dụng. Bài báo cáo này là kết quả của công việc nghiêm túc và trung thực của nhóm em, dựa trên hiểu biết và nghiên cứu của nhóm vào thời điểm biên soạn. Chúng em chịu trách nhiệm về tính chính xác và nội dung của báo cáo này và sẽ hoan nghênh thông tin bổ sung hoặc sửa đổi để cải thiện nó trong tương lai. Nhóm em xin cảm ơn sự quan tâm và hỗ trợ của quý thầy cô đối với bài báo cáo này.

## LỜI CẢM ƠN

Nhóm em muốn bày tỏ lòng biết ơn đến tất cả những người đã hỗ trợ và giúp đỡ trong quá trình thực hiện đề tài này. Nhóm em đặc biệt muốn gửi lời cảm ơn sâu sắc đến ThS. Trần Phan An Trường, người đã hướng dẫn môn học "Điện toán đám mây" và cung cấp sự hỗ trợ nhiệt tình và tận tâm. Thầy đã chia sẻ kiến thức chuyên môn, giới thiệu các khái niệm quan trọng và hướng dẫn trong quá trình nghiên cứu và thực hiện đề tài. Đóng góp của thầy đã giúp nhóm em nắm rõ hơn về lĩnh vực điện toán đám mây và phát triển kỹ năng trong lĩnh vực này.

Nhóm em cũng đánh giá cao mọi ý kiến đóng góp và phản hồi mà tôi nhận được. Tôi sẽ sử dụng chúng để hoàn thiện đề tài và học hỏi từ kinh nghiệm đó để cải thiện trong tương lai. Cảm ơn mọi người rất nhiều!

## MỤC LỤC

LỜI CAM ĐOAN .....	ii
LỜI CẢM ƠN .....	iii
MỤC LỤC.....	iv
DANH MỤC HÌNH ẢNH .....	vi
DANH MỤC CÁC BẢNG.....	vii
DANH MỤC CÁC CHỮ VIẾT TẮT .....	viii
MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI.....	2
1.1 Tổng quan điện toán đám mây .....	2
1.2 Lịch sử .....	2
1.3 Ưu điểm .....	3
1.4 Nhược điểm .....	3
1.5 Lĩnh vực của điện toán đám mây.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	5
2.1 Cơ sở lý thuyết.....	5
2.1.1 Khái niệm điện toán đám mây .....	5
2.1.2 Giới thiệu về Docker.....	7
2.1.3 Giới thiệu về Docker Swarm .....	8
2.1.1 Giới thiệu về Micro-Services.....	10
CHƯƠNG 3: NỘI DUNG NGHIÊN CỨU.....	12
3.1 Cài đặt.....	12
3.2 Demo .....	13
CHƯƠNG 4: KẾT LUẬN .....	17

4.1	Ưu điểm .....	17
4.2	Nhược điểm .....	17
TÀI LIỆU THAM KHẢO.....		19

## DANH MỤC HÌNH ẢNH

Hình 2.1 Mô hình điện toán đám mây. ....	6
Hình 2.2 Dịch vụ điện toán đám mây. ....	6
Hình 2.3 Docker .....	8
Hình 2.4 Kiến trúc docker swarm. ....	9
Hình 2.5 Monolithic sang Microservices. ....	11
Hình 3.1 Tải Docker. ....	12
Hình 3.2 Cài đặt Docker. ....	12
Hình 3.3 Giao diện Docker Desktop .....	13
Hình 3.4 Docker CLI. ....	13
Hình 3.5 Docker images. ....	14
Hình 3.6 Chạy các services trên flie docker compose. ....	14
Hình 3.7 Kiểm tra services đang hoạt động. ....	15
Hình 3.8 Triển khai Docker swarm. ....	15
Hình 3.9 Khởi tạo mạng riêng. ....	16
Hình 3.10 Khởi tạo Docker swarm. ....	16

## **DANH MỤC CÁC BẢNG**

Bảng 2.1 Ưu điểm và nhược điểm của điện toán đám mây. ....	7
--	---



**DANH MỤC CÁC CHỮ VIẾT TẮT**

<b>STT</b>	<b>Kí hiệu</b>	<b>Nội dung đầy đủ</b>
1	EC2	Elastic Compute Cloud
2	VR	Virtual Reality
3	IaaS	Infrastructure as a Service
4	PaaS	Platform as a Service
5	SaaS	Software as a Service
6	IoT	Internet of Things
7	ML	Machine Learning
8	AI	Artificial Intelligence
9	API	Application Programming Interface
10	CI	Continuous Integration
11	CD	Continuous Delivery

## MỞ ĐẦU

Đám mây tính hình thành một mô hình cung cấp tài nguyên máy tính, kho lưu trữ, ứng dụng và dịch vụ thông qua Internet. Thay vì phải mua và quản lý riêng từng bộ phần cứng và phần mềm, người dùng có khả năng truy cập các tài nguyên này theo nhu cầu và chỉ thanh toán cho những gì họ sử dụng.

Gần đây, đám mây tính đã trở thành một công nghệ quan trọng trong nhiều lĩnh vực, bao gồm doanh nghiệp, chính phủ và người tiêu dùng. Theo một báo cáo từ Gartner, dự kiến chi tiêu toàn cầu cho đám mây tính sẽ tăng lên 482,8 tỷ USD vào năm 2023, tăng 19,4% so với năm 2022.

Có một số xu hướng quan trọng trong lĩnh vực đám mây tính:

- Đám mây đa đám mây: Doanh nghiệp sẽ tận dụng nhiều nhà cung cấp đám mây để tối ưu hóa chi phí và hiệu suất.
- Đám mây theo chiều dọc: Các nhà cung cấp đám mây sẽ tạo ra các dịch vụ và ứng dụng được tùy chỉnh cho từng ngành cụ thể.
- Đám mây tiên tiến: Các công nghệ mới như trí tuệ nhân tạo (AI), học máy (ML) và thực tế ảo (VR) sẽ được tích hợp vào đám mây tính.

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

## 1.1 Tổng quan điện toán đám mây

Điện toán đám mây (cloud computing) là một mô hình điện toán cho phép người dùng truy cập các tài nguyên máy tính, lưu trữ và ứng dụng qua Internet. Các tài nguyên này có thể bao gồm máy chủ, lưu trữ, phần mềm, mạng, cơ sở dữ liệu, trí tuệ nhân tạo (AI) và phân tích.

## 1.2 Lịch sử

Điện toán đám mây bắt đầu xuất hiện từ những năm 1950, khi các tổ chức sử dụng máy tính mainframe để lưu trữ và xử lý dữ liệu. Tuy nhiên, nó trở nên phổ biến hơn trong những năm gần đây do sự phát triển của Internet và sự xuất hiện của các nhà cung cấp đám mây lớn như Amazon Web Services, Microsoft Azure và Google Cloud Platform.

Các cột mốc lịch sử:

- 1950: Sử dụng máy tính mainframe bắt đầu.
- 1960: Khái niệm "máy tính như một dịch vụ" được đề xuất.
- 1960: Phòng thí nghiệm Bell Labs phát triển ARPANET, mạng máy tính tiên phong.
- 1970: Công nghệ phân tán và chia sẻ tài nguyên phát triển.
- 1980: Điện toán lưới (grid computing) xuất hiện, cho phép chia sẻ tài nguyên giữa máy tính.
- 1990: Internet tốc độ cao và công nghệ web phát triển.
- 1999: Salesforce.com ra mắt ứng dụng kinh doanh dựa trên web.
- 2000: Amazon ra mắt dịch vụ Elastic Compute Cloud (EC2).
- 2006: Microsoft giới thiệu nền tảng đám mây Azure.
- 2008: Google ra mắt Google Cloud Platform.
- 2010: Điện toán đám mây trở nên phổ biến hơn.

Hiện tại, điện toán đám mây đang phát triển nhanh chóng và trở thành một phần quan trọng của cuộc sống, được sử dụng rộng rãi trong nhiều lĩnh vực như doanh nghiệp, giáo dục, y tế và giải trí. Dự kiến rằng trong tương lai, điện toán đám mây sẽ

tiếp tục phát triển và cung cấp nhiều dịch vụ và công nghệ mới để đáp ứng nhu cầu ngày càng tăng của người dùng.

### 1.3 Ưu điểm

- Điện toán đám mây mang tính linh hoạt, cho phép người dùng truy cập tài nguyên từ bất kỳ thiết bị nào có kết nối Internet, giúp họ làm việc từ mọi nơi và mọi lúc.
- Khả năng mở rộng dễ dàng, người dùng có thể thay đổi quy mô tài nguyên đám mây để phù hợp với nhu cầu biến đổi, giúp tiết kiệm chi phí và nâng cao hiệu quả.
- Đám mây có tính sẵn sàng cao nhờ đầu tư vào hạ tầng, đảm bảo hoạt động liên tục và bảo vệ dữ liệu của doanh nghiệp.
- Giảm thiểu chi phí, doanh nghiệp có thể tiết kiệm chi phí đầu tư và vận hành cơ sở hạ tầng CNTT.
- Tích hợp công nghệ mới dễ dàng và đẩy nhanh quá trình đổi mới thương mại.
- Tạo điều kiện tốt cho việc hợp tác đa điểm, giúp người dùng từ xa cùng làm việc cùng nhau.
- Điện toán đám mây cung cấp tính năng bảo mật tiên tiến để đảm bảo an toàn cho dữ liệu của người dùng.

### 1.4 Nhược điểm

- Bảo mật dữ liệu là một ưu tiên quan trọng, đặc biệt đối với các doanh nghiệp, bởi họ thường lưu trữ thông tin nhạy cảm như thông tin khách hàng, tài chính và dữ liệu bí mật trên đám mây. Bất kỳ vi phạm bảo mật nào có thể gây hậu quả nghiêm trọng cho doanh nghiệp.
- Tuân thủ các quy định về bảo mật và dữ liệu có thể trở nên phức tạp hơn khi sử dụng điện toán đám mây, do các quy định này khác nhau ở từng quốc gia và khu vực. Điều này có thể gây khó khăn cho việc tuân thủ quy định.
- Quản lý tài nguyên đám mây có thể đòi hỏi kiến thức và kinh nghiệm, và người dùng cần hiểu cách sử dụng các dịch vụ và tính năng đám mây để sử dụng chúng một cách hiệu quả.

- Người dùng thường không có kiểm soát tuyệt đối đối với dữ liệu và ứng dụng của họ khi sử dụng điện toán đám mây, và họ phải phụ thuộc vào nhà cung cấp đám mây để cung cấp và duy trì các dịch vụ.
- Khả năng tùy chỉnh có thể bị hạn chế khi sử dụng các dịch vụ và tính năng tiêu chuẩn từ các nhà cung cấp đám mây.
- Đám mây có thể gặp sự cố hoặc gián đoạn dịch vụ do lỗi hệ thống, sự cố mạng hoặc vấn đề khác, gây ảnh hưởng đến hoạt động kinh doanh và dữ liệu.

### 1.5 Lĩnh vực của điện toán đám mây

- Doanh nghiệp có thể hưởng lợi từ điện toán đám mây bằng cách tiết kiệm chi phí, nâng cao hiệu quả và thúc đẩy sự đổi mới.
- Giáo dục có thể sử dụng điện toán đám mây để mở rộng phạm vi tiếp cận cho sinh viên và cung cấp giáo dục chất lượng cao hơn.
- Trong lĩnh vực y tế, điện toán đám mây có thể giúp cải thiện hiệu suất và dịch vụ chăm sóc bệnh nhân.
- Người dùng có thể dễ dàng truy cập nội dung giải trí thông qua điện toán đám mây, mang lại sự thuận tiện.
- Chính phủ cũng có thể tận dụng điện toán đám mây để cung cấp dịch vụ công hiệu quả và tiết kiệm chi phí.
- Các lĩnh vực mới của điện toán đám mây bao gồm:
  - Điện toán biên, sử dụng máy chủ gần người dùng cuối để hỗ trợ các dịch vụ như phát trực tuyến video và trò chơi.
  - Điện toán lượng tử, sử dụng máy tính lượng tử để thực hiện các phép tính phức tạp mà máy tính thông thường không thể thực hiện.
  - Điện toán đám mây cho Internet vạn vật (IoT), sử dụng để kết nối và quản lý các thiết bị IoT.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1 Cơ sở lý thuyết

#### 2.1.1 Khái niệm điện toán đám mây

Điện toán đám mây (cloud computing) là mô hình điện toán cho phép chia sẻ tài nguyên máy tính, lưu trữ, ứng dụng, và dịch vụ thông qua mạng Internet. Thay vì đầu tư và quản lý các trung tâm dữ liệu và máy chủ riêng, người dùng có thể thuê các dịch vụ điện toán đám mây từ các nhà cung cấp dịch vụ đám mây [1].

#### Lịch sử

Thuật ngữ "Điện toán đám mây" xuất hiện lần đầu tiên vào năm 2007, nhưng ý tưởng về điện toán đám mây đã có từ lâu. Các khái niệm như điện toán lưới (grid computing), điện toán theo nhu cầu (utility computing), và phần mềm dịch vụ (SaaS) đều là những tiền thân của điện toán đám mây.

#### Các đặc tính của điện toán đám mây:

- Tự phục vụ theo yêu cầu (on-demand self-service)
- Sự truy cập mạng rộng rãi (broad network access)
- Tập trung tài nguyên
- Tính mềm dẻo
- Khả năng đo lường

#### Các mô hình điện toán đám mây

- Đám mây riêng (Private Cloud): là một mô hình điện toán đám mây trong đó tài nguyên được sở hữu và vận hành bởi một tổ chức duy nhất.
- Đám mây công cộng (Public Cloud): Đám mây công cộng là một mô hình điện toán đám mây trong đó tài nguyên được sở hữu và vận hành bởi một nhà cung cấp đám mây.
- Đám mây cộng đồng (Community Cloud): Đám mây cộng đồng là một mô hình điện toán đám mây trong đó tài nguyên được sở hữu và vận hành bởi một nhóm tổ chức có chung mục đích.

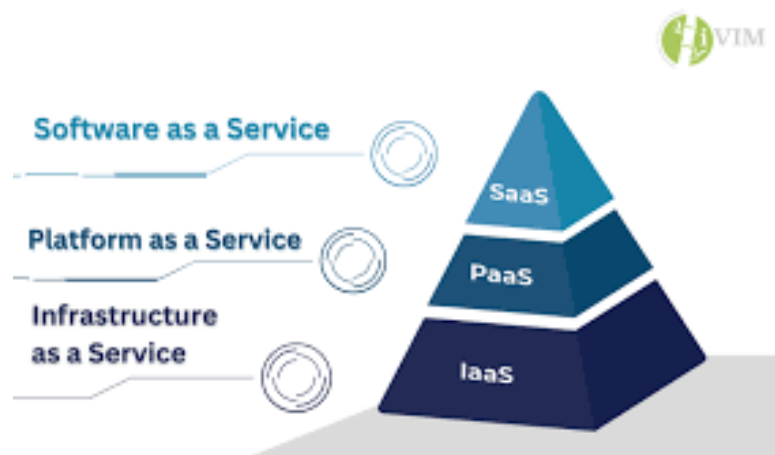
- Đám mây lai (Hybrid Cloud): là một mô hình điện toán đám mây trong đó các tổ chức sử dụng kết hợp các đám mây riêng, đám mây công cộng và đám mây cộng đồng.



Hình 2.1 Mô hình điện toán đám mây.

#### Các mô hình dịch vụ điện toán đám mây chính:

- IaaS (Infrastructure as a Service): Nhà cung cấp đám mây cung cấp cho người dùng quyền truy cập vào cơ sở hạ tầng máy tính, bao gồm máy chủ, lưu trữ và mạng.
- PaaS (Platform as a Service): Nhà cung cấp đám mây cung cấp cho người dùng một nền tảng để phát triển và chạy ứng dụng.
- SaaS (Software as a Service): Nhà cung cấp đám mây cung cấp cho người dùng các ứng dụng phần mềm được triển khai trên đám mây.



Hình 2.2 Dịch vụ điện toán đám mây.

#### Các ưu điểm và nhược điểm của điện toán đám mây

<b>Ưu điểm</b>	<b>Nhược điểm</b>
Triển khai nhanh chóng	Chi phí
Giảm chi phí	Các công cụ giám sát và quản lý
Đa phương tiện truy cập	Chuẩn hóa đám mây
Chia sẻ	Tính sẵn sàng
Khả năng chịu tải nâng cao	Vấn đề tuân thủ hợp đồng
Độ tin cậy	Tính riêng tư
Tính co giãn linh động	Cấp độ dịch vụ
Bảo mật	Khả năng tích hợp với hạ tầng thông tin sẵn có của tổ chức

Bảng 2.1 Ưu điểm và nhược điểm của điện toán đám mây.

### Các ứng dụng của điện toán đám mây

- Trong lĩnh vực tiếp thị, điện toán đám mây có thể lưu trữ dữ liệu khách hàng, thực hiện các chiến dịch tiếp thị trực tuyến và phân tích dữ liệu.
- Trong lĩnh vực công nghệ, điện toán đám mây được sử dụng để phát triển và triển khai ứng dụng, thực hiện mô hình học máy và lưu trữ dữ liệu.
- Doanh nghiệp có thể tận dụng điện toán đám mây để quản lý tài chính, nhân sự và các hoạt động kinh doanh khác.

#### 2.1.2 Giới thiệu về Docker

Docker là công nghệ mã nguồn mở cho phép đóng gói và chạy ứng dụng trong những gói gọi là "containers". Containers là gói độc lập chứa tất cả mọi thứ cần thiết cho ứng dụng hoạt động. Chúng giúp ứng dụng chạy đồng nhất trên mọi môi trường, đơn giản hóa việc triển khai và quản lý ứng dụng.

Tính năng quan trọng của Docker bao gồm khả năng quản lý, triển khai và chạy các containers một cách hiệu quả, khả năng tích hợp với các công cụ phát triển và quy trình triển khai liên tục (CI/CD), cùng với việc sử dụng hệ thống tệp và ảnh Docker.



Lợi ích của việc sử dụng Docker bao gồm sự đảm bảo tính đồng nhất giữa môi trường phát triển và triển khai, khả năng cơ động và mở rộng dễ dàng, giảm bớt xung đột và xung đột hệ thống, và giúp tăng tốc quá trình phát triển và triển khai ứng dụng.



Hình 2.3 Docker

### 2.1.3 Giới thiệu về Docker Swarm

Docker Swarm là một công cụ quản lý cluster cho Docker, tích hợp ngay trong Docker Engine. Nó giúp bạn dễ dàng tạo và quản lý một nhóm máy tính gọi là "swarm," trong đó mỗi máy tính đóng vai trò là một "node" Docker. Docker Swarm giúp quản lý các node này một cách hiệu quả.

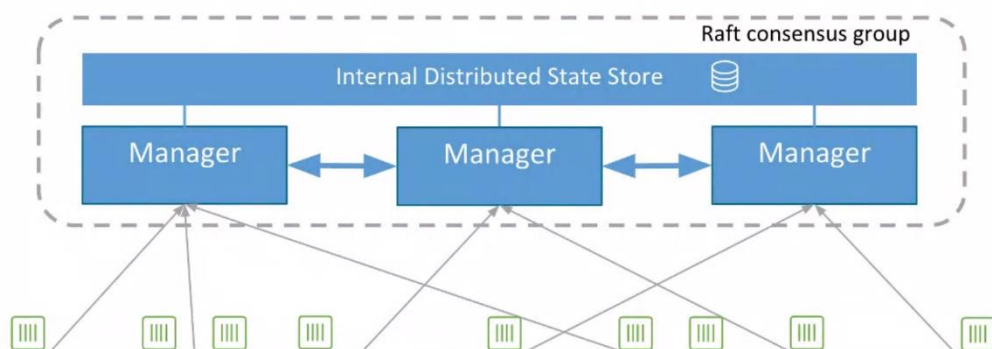
Tính năng của Docker Swarm:

- Quản lý cluster tích hợp với Docker Engine: Docker Swarm cho phép bạn quản lý cluster bằng cách sử dụng Docker CLI để tạo một mạng Swarm.
- Thiết kế phi tập trung: Docker Swarm có thiết kế phi tập trung, cho phép bạn triển khai cả các node quản lý (managers) và node công việc (workers) bằng Docker Engine, giúp tối ưu hóa việc xử lý khác biệt giữa các vai trò của node tại thời điểm chạy.
- Mô hình dịch vụ theo cách khai báo: Docker Swarm sử dụng mô hình khai báo để định nghĩa trạng thái mong muốn của các dịch vụ trong ứng dụng. Ví dụ, bạn có thể mô tả một ứng dụng với phần front-end web, dịch vụ hàng đợi tin nhắn và phần cơ sở dữ liệu.
- Tăng cường khả năng mở rộng: Bạn có thể chỉ định số lượng task (nhiệm vụ) bạn muốn chạy cho mỗi dịch vụ và khi bạn tăng hoặc giảm quy mô,

swarm manager tự động thêm hoặc xóa task để duy trì trạng thái mong muốn.

- Đảm bảo trạng thái mong muốn: Swarm manager giúp đảm bảo trạng thái mong muốn của dịch vụ bằng cách tự động tạo hoặc thay thế các task bị lỗi, đảm bảo tính ổn định và khả năng phục hồi.
- Mạng đa máy chủ: Docker Swarm cho phép bạn tạo các mạng overlay cho các dịch vụ của mình, tự động cung cấp địa chỉ IP cho các container khi chúng khởi tạo và cập nhật ứng dụng.
- Dịch vụ khám phá: Swarm manager gán một DNS duy nhất cho mỗi dịch vụ trong cluster, cho phép bạn truy vấn dịch vụ thông qua DNS.
- Cân bằng tải: Bạn có thể mở cổng của dịch vụ để giao tiếp với bên ngoài và Swarm tự động quản lý việc cân bằng tải cho các dịch vụ này.
- Bảo mật mặc định: Docker Swarm sử dụng giao thức bảo mật TLS để bảo vệ việc giao tiếp giữa các dịch vụ. Bạn có thể tùy chỉnh việc sử dụng chứng chỉ từ gốc hoặc một gốc CA tùy chỉnh.
- Cập nhật từng bước: Swarm cho phép bạn cập nhật hình ảnh của dịch vụ một cách tự động và kiểm soát quy trình triển khai giữa các node khác nhau, cũng như khả năng quay lại phiên bản trước nếu cần.

Kiến trúc của Swarm: Bao gồm các Manager và các Worker. Người dùng có thể khai báo trạng thái mong muốn của nhiều service để chạy trong Swarm sử dụng YAML files.



Hình 2.4 Kiến trúc docker swarm.

- Swarm: là một cluster của một hoặc nhiều Docker Engine đang run (cụ thể ở đây là các node) trong chế độ Swarm, thay vì phải chạy các container bằng câu lệnh thì ta sẽ thiết lập các services để phân bổ các bản replicas tới các node.
- Node: Một node là một máy tính hoặc máy ảo chạy Docker Engine trong chế độ Swarm. Có hai loại node: Manager Node và Worker Node.
- Manager Node: Node quản lý, nhận lệnh từ người dùng để định nghĩa dịch vụ. Nó cũng phân phối các nhiệm vụ đến các Worker Node. Theo mặc định, Manager Node cũng có thể làm việc như Worker Node.
- Worker Node: Node thực thi nhiệm vụ được giao từ Manager Node.
- Service: Một dịch vụ định nghĩa hình ảnh của container và số lượng bản sao mà bạn muốn chạy trong Swarm.
- Task: Mỗi nhiệm vụ là một công việc mà Worker Node phải thực hiện. Nhiệm vụ này chứa một container Docker và các lệnh cần thiết để chạy container đó. Manager Node phân phối các nhiệm vụ này đến các Worker Node.

### 2.1.1 Giới thiệu về Micro-Services

Microservices là một kiến trúc phần mềm mà ứng dụng lớn được chia thành các dịch vụ nhỏ, độc lập và có khả năng hoạt động độc lập với nhau. Mỗi dịch vụ tập trung vào một chức năng cụ thể của ứng dụng và có thể phát triển, triển khai và mở rộng một cách độc lập. Mô hình microservices giúp tăng tính linh hoạt, mở rộng và quản lý ứng dụng, đồng thời giảm thiểu sự phụ thuộc giữa các thành phần của ứng dụng. Điều này làm cho việc phát triển và duy trì ứng dụng trở nên dễ dàng hơn và giúp ứng dụng hoạt động ổn định và hiệu quả hơn.

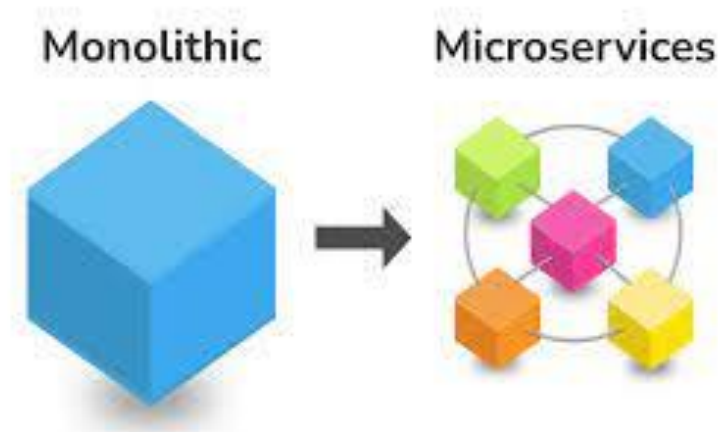
Đặc điểm chính của microservices bao gồm:

- Tách biệt và độc lập: Mỗi microservice là một phần độc lập của ứng dụng, có thể được phát triển, triển khai và quản lý riêng biệt.
- Tích hợp linh hoạt: Microservices có khả năng tích hợp với các dịch vụ khác thông qua giao diện (API) chung.

- Tính nhẹ và linh hoạt: Microservices thường nhẹ và có khả năng mở rộng dễ dàng, giúp tối ưu hóa sử dụng tài nguyên.
- Tự động hóa và tự phục hồi: Microservices có khả năng tự động phục hồi sau lỗi và có thể quản lý mình mà không cần can thiệp thủ công.

Lợi ích của việc sử dụng kiến trúc microservices:

- Tính linh hoạt và mở rộng: Microservices cho phép bạn thay đổi và mở rộng từng phần của ứng dụng mà không ảnh hưởng đến phần còn lại.
- Dễ quản lý và bảo trì: Việc quản lý và bảo trì từng microservice riêng lẻ giúp giảm bớt sự phức tạp.
- Phát triển độc lập và đồng thời: Đội ngũ phát triển có thể làm việc độc lập trên từng microservice, giúp tăng tốc quá trình phát triển.
- Tích hợp linh hoạt: Microservices cho phép tích hợp các công nghệ và ngôn ngữ lập trình khác nhau.
- Tăng khả năng chịu lỗi: Với kiến trúc phân tán, việc lỗi xảy ra trong một microservice không ảnh hưởng đến toàn bộ ứng dụng.
- Mở cửa cho việc sử dụng công nghệ mới: Microservices cho phép bạn sử dụng các công nghệ mới một cách dễ dàng trong từng dịch vụ.



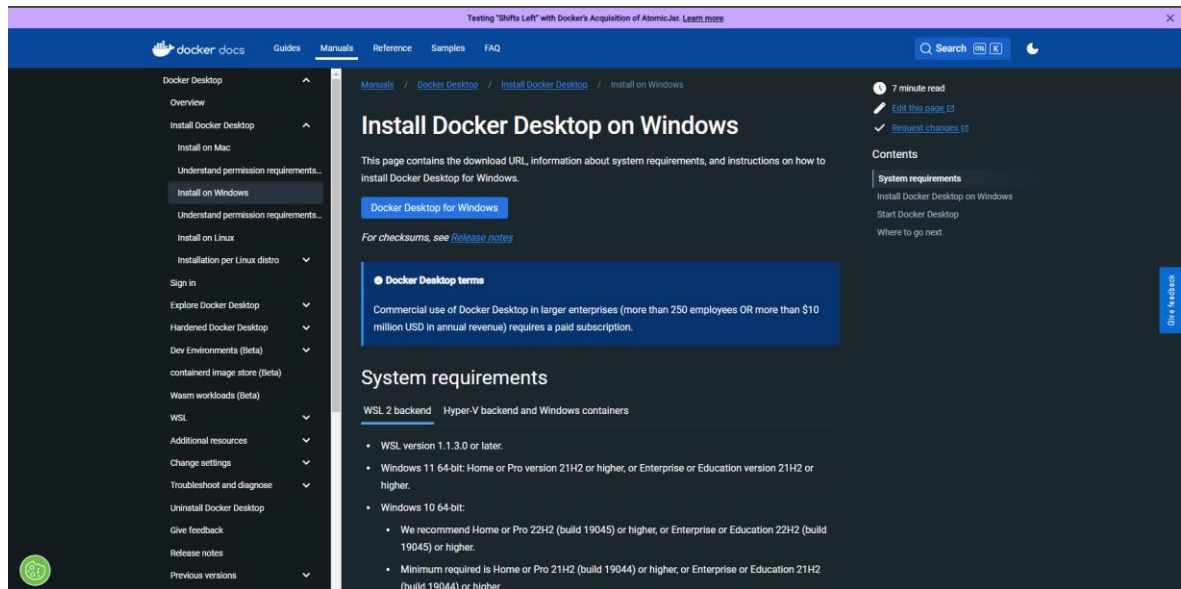
Hình 2.5 Monolithic sang Microservices.

## CHƯƠNG 3: NỘI DUNG NGHIÊN CỨU

### 3.1 Cài đặt

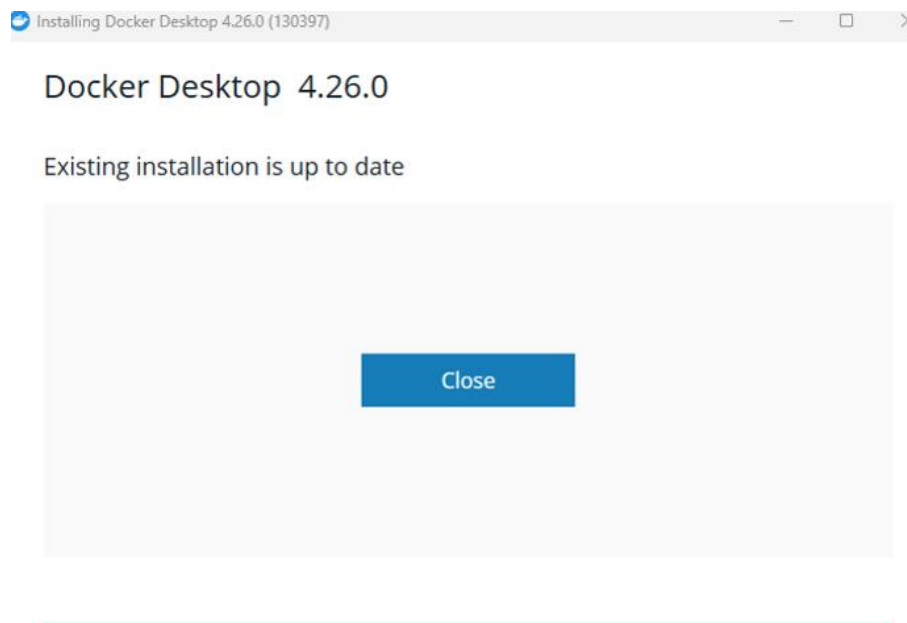
Trên Windows, cài đặt Docker Desktop theo link:

<https://docs.docker.com/desktop/install/windows-install/>



Hình 3.1 Tải Docker.

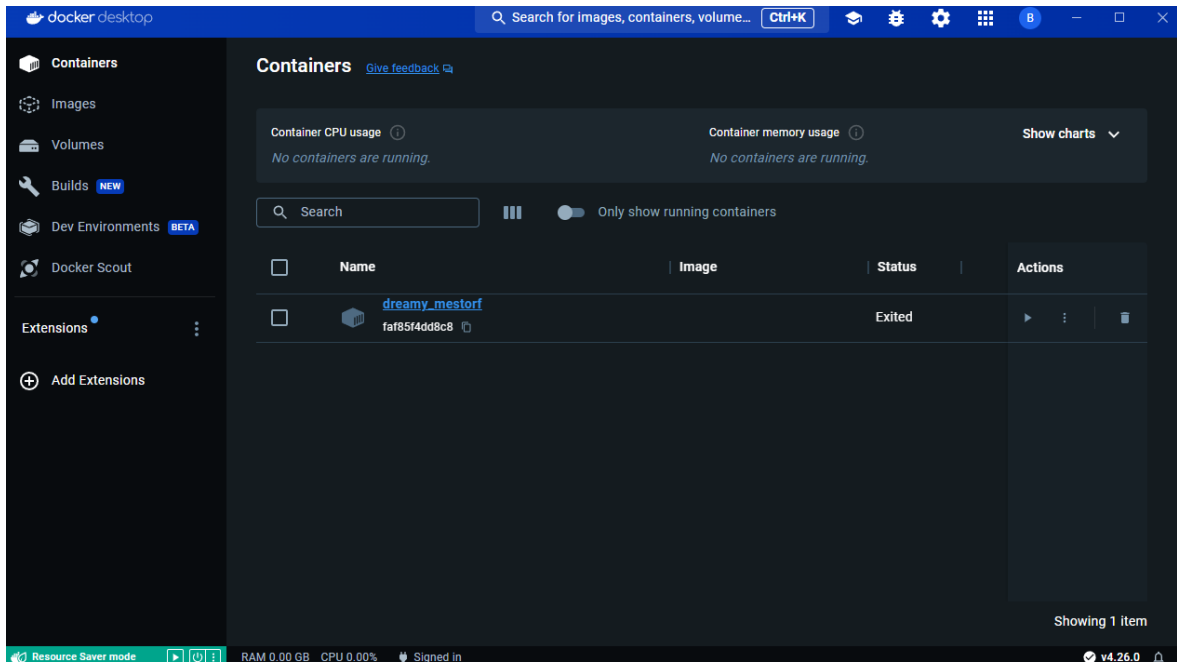
Và thực hiện cài đặt như những phần mềm ứng dụng bình thường khác. Kết quả sau khi cài đặt như hình bên dưới:



Hình 3.2 Cài đặt Docker

### 3.2 Demo

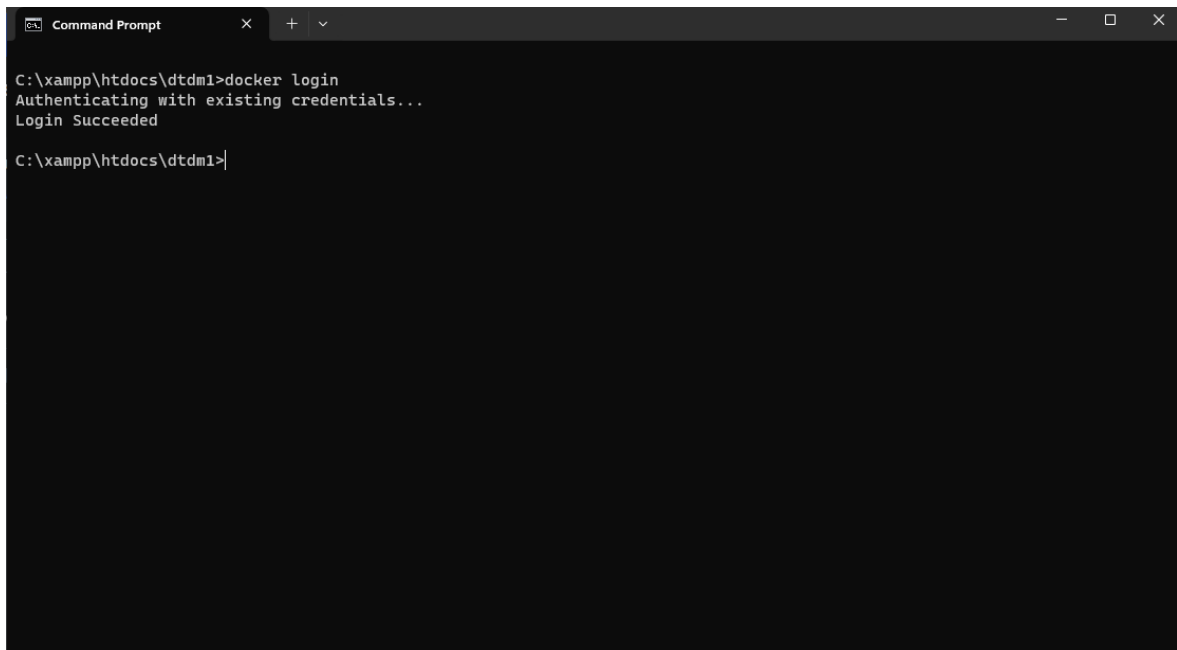
Mở Docker Desktop:



Hình 3.3 Giao diện Docker Desktop

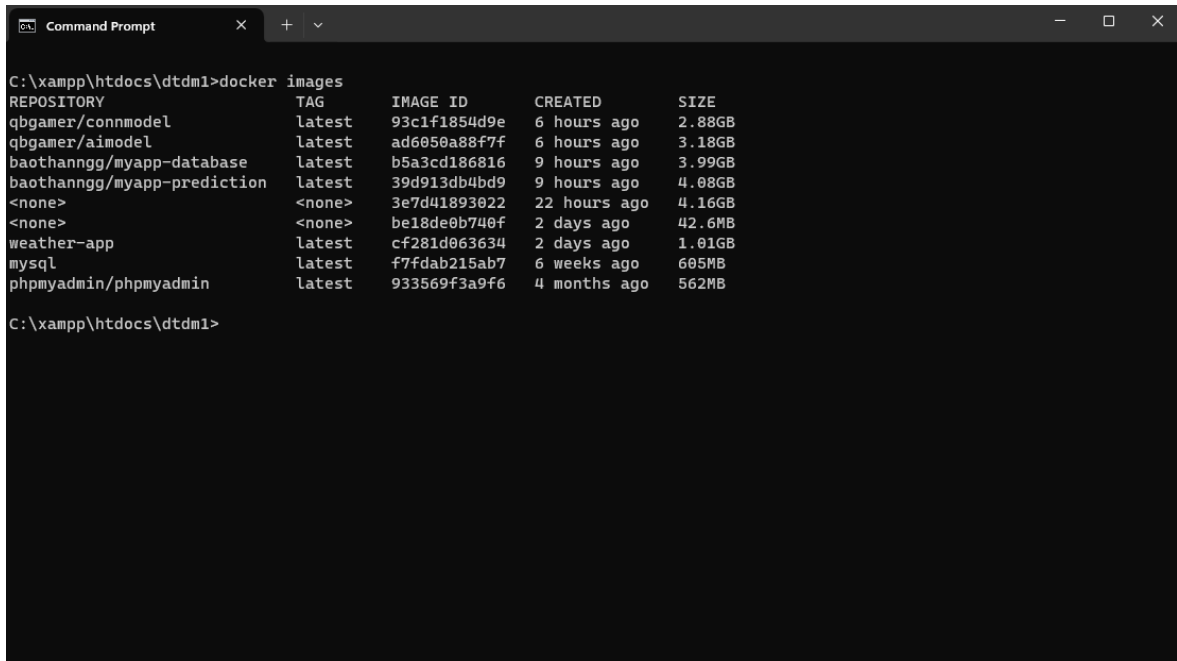
Triển khai Docker bằng Docker CLI:

Đăng nhập:



Hình 3.4 Docker CLI.

Kiểm tra các image:



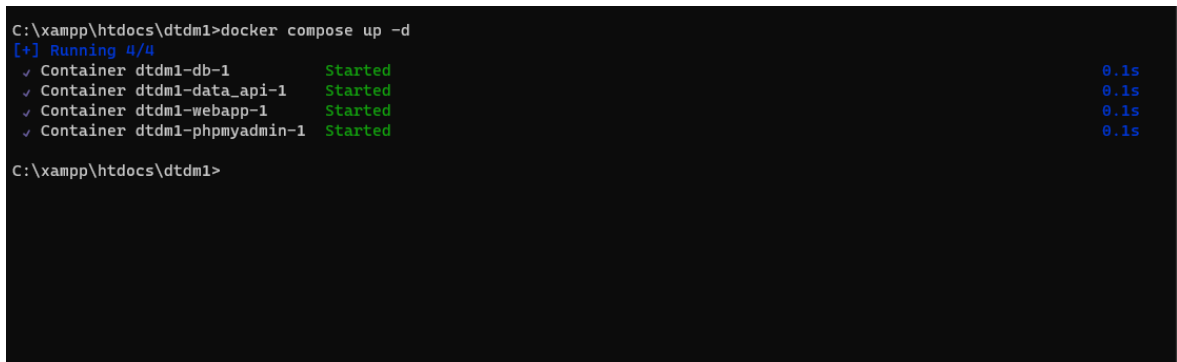
```

C:\xampp\htdocs\dtadm1>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
qbgamer/connmodel   latest          93c1f1854d9e   6 hours ago    2.88GB
qbgamer/aimodel     latest          ad6050a88f7f   6 hours ago    3.18GB
baothanngg/myapp-database latest          b5a3cd186816   9 hours ago    3.99GB
baothanngg/myapp-prediction latest          39d913db4bd9   9 hours ago    4.08GB
<none>              <none>          3e7d41893022   22 hours ago   4.16GB
<none>              <none>          be18de0b740f   2 days ago     42.6MB
weather-app         latest          cf281d063634   2 days ago     1.01GB
mysql               latest          f7fdab215ab7   6 weeks ago    605MB
phpmyadmin/phpmyadmin latest          933569f3a9f6   4 months ago   562MB
C:\xampp\htdocs\dtadm1>

```

Hình 3.5 Docker images.

Chạy file docker-compose.yml đã xây dựng từ trước:



```

C:\xampp\htdocs\dtadm1>docker compose up -d
[+] Running 4/4
 ✓ Container dtadm1-db-1           Started 0.1s
 ✓ Container dtadm1-data_api-1     Started 0.1s
 ✓ Container dtadm1-webapp-1       Started 0.1s
 ✓ Container dtadm1-phpmyadmin-1   Started 0.1s
C:\xampp\htdocs\dtadm1>

```

Hình 3.6 Chạy các services trên file docker compose.

Kiểm tra các container nào đang hoạt động:

```

C:\xampp\htdocs\dtdm1>docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
NAMES
aa482b16a39b   qbgamer/aimodel                    "python predict.py"     42 seconds ago Up 41 seconds 0.0.0.0:5000->5000/tcp
dtdm1-webapp-1
90d128d6ac39   phpmyadmin/phpmyadmin              "/docker-entrypoint.s..." 42 seconds ago Up 41 seconds 0.0.0.0:8080->80/tcp
dtdm1-phpmyadmin-1
31874891014c   qbgamer/connmodel                  "python db.py"          42 seconds ago Up 41 seconds 0.0.0.0:5001->5001/tcp
dtdm1-data_api-1
dd1146388775   mysql                               "docker-entrypoint.s..." 42 seconds ago Up 42 seconds 3306/tcp, 33060/tcp
dtdm1-db-1

C:\xampp\htdocs\dtdm1>

```

Hình 3.7 Kiểm tra services đang hoạt động.

Triển khai Docker Swarm:

```

C:\xampp\htdocs\dtdm1>docker swarm init
Swarm initialized: current node (7veosmz7t1qnaf8uyssal00fe) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-4644sin3ipx733z0u230r0kpggeep3inx218b3y2v0tn6xjmbk-3p78hc0y556upbl8zg4jyuvse 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

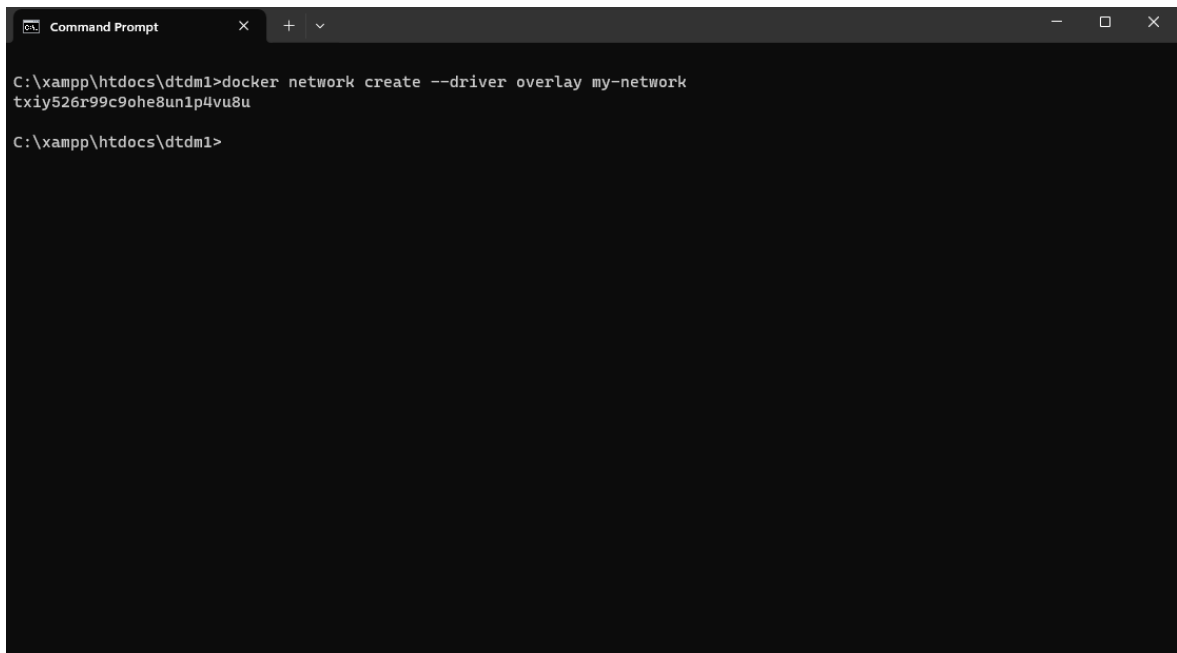
C:\xampp\htdocs\dtdm1>

```

Hình 3.8 Triển khai Docker swarm

Tạo mạng overlay cho Docker Swarm:

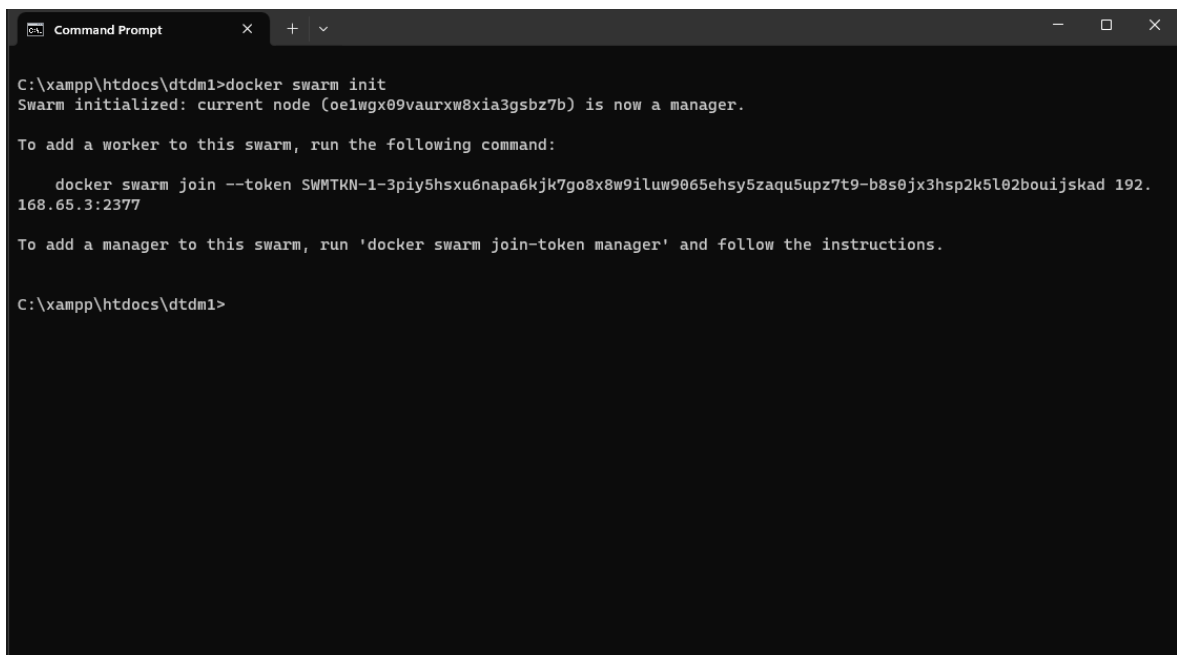




```
Command Prompt
C:\xampp\htdocs\dtdm1>docker network create --driver overlay my-network
txiy526r99c9ohe8un1p4vu8u
C:\xampp\htdocs\dtdm1>
```

Hình 3.9 Khởi tạo mạng riêng.

Khởi tạo Docker Swarm:



```
Command Prompt
C:\xampp\htdocs\dtdm1>docker swarm init
Swarm initialized: current node (oe1wgx09vaurxw8xia3gsbz7b) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3piy5hsxu6napa6kjk7go8x8w9iluw9065ehsy5zaqu5upz7t9-b8s0jx3hsp2k5l02bouijskad 192.168.65.3:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

C:\xampp\htdocs\dtdm1>
```

Hình 3.10 Khởi tạo Docker swarm.

## CHƯƠNG 4: KẾT LUẬN

### 4.1 Ưu điểm

Docker đóng gói ứng dụng và phụ thuộc hệ thống trong containers, giúp đảm bảo tính độc lập và cách ly. Việc triển khai trở nên dễ dàng trên mọi môi trường, từ máy tính phát triển đến môi trường sản xuất. Docker cũng tích hợp tốt với quy trình CI/CD và có một cộng đồng đông đảo hỗ trợ và tài liệu phong phú.

Docker Swarm, một phần của Docker Engine, cho phép mở rộng cụm máy chủ (swarm) một cách đơn giản bằng cách thêm máy tính vào cụm. Nó tự động phân phối nhiệm vụ trên các nodes, đảm bảo hiệu suất và sẵn sàng cao cho ứng dụng. Docker Swarm tích hợp với Docker Compose, giúp đơn giản hóa việc định nghĩa và triển khai ứng dụng phân tán. Nó cũng cung cấp các tùy chọn bảo mật để bảo vệ cụm máy chủ. Đồng thời, Docker Swarm có một cộng đồng mạnh mẽ, giúp người dùng tìm hiểu và giải quyết vấn đề dễ dàng.

### 4.2 Nhược điểm

➤ Nhược điểm của Docker trong xây dựng dịch vụ hướng microservices:

1. Tính nhẹ không hoàn toàn: Mặc dù Docker là một công nghệ container tương đối nhẹ, nhưng việc triển khai nhiều containers có thể tiêu tốn tài nguyên hệ thống đáng kể, đặc biệt trên các máy tính có tài nguyên hạn chế.

2. Quản lý mạng phức tạp: Khi triển khai nhiều containers, việc quản lý và cấu hình mạng có thể trở nên phức tạp, đặc biệt khi ứng dụng yêu cầu một mô hình mạng phức tạp.

➤ Nhược điểm của Docker Swarm trong xây dựng dịch vụ hướng microservices:

1. Khả năng mở rộng hạn chế: Docker Swarm có giới hạn về quy mô, không phù hợp cho các ứng dụng có nhu cầu mở rộng lớn hoặc trong các môi trường phức tạp.

2. Tính năng tự động hóa không hoàn chỉnh: Mặc dù Docker Swarm cung cấp tự động phân phối nhiệm vụ, nhưng tính năng tự động hóa hoàn chỉnh có thể không đáp ứng được một số yêu cầu đặc biệt của ứng dụng.

3. Yêu cầu cấu hình khó khăn: Cấu hình và triển khai Docker Swarm có thể đòi hỏi kiến thức chuyên sâu về Docker và các tùy chọn cấu hình phức tạp.

4. Hiệu suất không cao hơn các giải pháp khác: Một số giải pháp orchestration khác có thể cung cấp hiệu suất cao hơn và tích hợp tính năng nâng cao.

Tuy nhược điểm này tồn tại, nhưng Docker và Docker Swarm vẫn có thể hoạt động tốt trong nhiều trường hợp và với việc lựa chọn phù hợp, các vấn đề có thể được giải quyết hoặc tối ưu hóa.

## **TÀI LIỆU THAM KHẢO**

- [1] TS. Phan Anh Cang (chủ biên), ThS. Trần Phan An Trường, ThS. Nguyễn Thanh Hoàng, “Giáo trình Điện toán đám mây.”, NXB Khoa học và Kỹ thuật.
- [2] Huỳnh Quyết Thắng (chủ biên), Nguyễn Hữu Đức, Doãn Trung Tùng, Nguyễn Bình Minh, Trần Việt Trung, “Điện toán đám mây.”, NXB Bách khoa Hà Nội.
- [3] Docker, Inc. "Docker documentation." Electronic resource]. Docker Documentation:[website]. URL: <https://docs.docker.com/>(accessed: 02.05.2022) (2017).
- [4] Nickoloff, Jeffrey, and Stephen Kuenzli. Docker in action. Simon and Schuster, 2019.