TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CNTT2**

**XÂY DỰNG WEBSITE ĐĂNG TIN TUYỂN DỤNG CỦA DOANH NGHIỆP CHO SINH VIÊN TDTU SỬ DỤNG CÔNG NGHỆ NODEJS VÀ REACT**

*Người hướng dẫn*: **GV DOÃN XUÂN THANH**

*Người thực hiện*: **NGUYỄN HỮU TẤN ĐẠT - 51702075**

**NGÔ MINH HIẾU - 51702017**

Lớp **: 17050201**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CNTT2**

**XÂY DỰNG WEBSITE ĐĂNG TIN TUYỂN DỤNG CỦA DOANH NGHIỆP CHO SINH VIÊN TDTU SỬ DỤNG CÔNG NGHỆ NODEJS VÀ REACT**

*Người hướng dẫn*: **GV DOÃN XUÂN THANH**

*Người thực hiện*: **NGUYỄN HỮU TẤN ĐẠT - 51702075**

**NGÔ MINH HIẾU - 51702017**

Lớp **: 17050201**

Khoá  **: 21**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành đến thầy **Doãn Xuân Thanh** – Giảng viên hướng dẫn của chúng em trong việc thực hiện dự án công nghệ thông tin 2, và khoa Công Nghệ Thông Tin đã cho em cơ hội thực hiện đề tài này và cũng như là cơ hội em được tiếp xúc với một số công nghệ được xem là trang bị cho em trong môi trường làm việc thực tế, cũng như trong việc hoàn thành đề tài xây dựng trang ứng dụng này.

Vì áp dụng 1 số kiến thức ở trường cũng như tìm hiểu và ứng dụng một số công nghệ mới nên chúng em sẽ không tránh khỏi được những sai sót và sự hạn chế của kiến thức tự tìm hiểu , đánh giá và trình bày về đề tài này. Rất mong nhận được sự quan tâm, góp ý của các thầy/cô để chung em biết mình thiếu sót những gì và giúp cho đề tài của chúng em được đầy đủ và hoành chỉnh hơn. Chúng em xin chân thành cảm ơn.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của GV Doãn Xuân Thanh ;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 03 tháng 12 năm 2022*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Đạt*

*Nguyễn Hữu Tấn Đạt*

*Hiếu*

*Ngô Minh Hiếu*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

* Giới thiệu các công nghệ và một số thuật ngữ được sử dụng trong quá trình xây dựng trang ứng dụng
  + Docker, CI/CD, VPS, NGINX, NODEJS
* Xác định các tính năng cần có , xây dựng các bảng dữ liệu lưu trữ cần có để thực tiến hành viết API và giao diện tương ứng
* Quá trình xây dựng API và cách gọi đến API
* Quá trình xây dựng Giao diện.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc121063626)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc121063627)

[TÓM TẮT iii](#_Toc121063628)

[MỤC LỤC 1](#_Toc121063629)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc121063630)

[CHƯƠNG 1 – GIỚI THIỆU 4](#_Toc121063631)

[1. GIỚI THIỆU VỀ ĐỀ TÀI 4](#_Toc121063632)

[2. GIỚI THIỆU VỀ CÔNG NGHỆ VÀ THUẬT NGỮ 4](#_Toc121063633)

[1) Docker 4](#_Toc121063634)

[2) CI/CD 10](#_Toc121063635)

[3) VPS 17](#_Toc121063636)

[4) NODEJS(EXPRESS) 21](#_Toc121063637)

[5) ORM (PRISMA) 22](#_Toc121063638)

[3. Nội dung của chương này 29](#_Toc121063639)

[CHƯƠNG 2 – QUÁ TRÌNH THỰC HIỆN 30](#_Toc121063640)

[1 Quá trình thiết lập server 30](#_Toc121063641)

[CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM 39](#_Toc121063642)

[3.1 Chèn bảng: 39](#_Toc121063643)

[3.2 Viết tắt 39](#_Toc121063644)

[3.3 Trích dẫn 39](#_Toc121063645)

[3.3.1 Tài liệu tham khảo và cách trích dẫn 39](#_Toc121063646)

[3.3.2 Qui định của Khoa Công nghệ thông tin 40](#_Toc121063647)

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

CI/CD Chỉ quá trình tích hợp liên tục và phân phối liên tục

JS Là một ngôn ngữ lập trình JavaScript

TS Là một ngôn ngữ lập trình được phát triển bởi Microsoft và có cách viết tương tự như JavaScript nhưng chặt chẽ hơn và dễ phát hiện lỗi sai trong khi viết

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Kiến trúc FTP 1](#_Toc387689394)

**DANH MỤC BẢNG**

[Bảng 3.1 Ví dụ cho chèn bảng 1](#_Toc387689363)

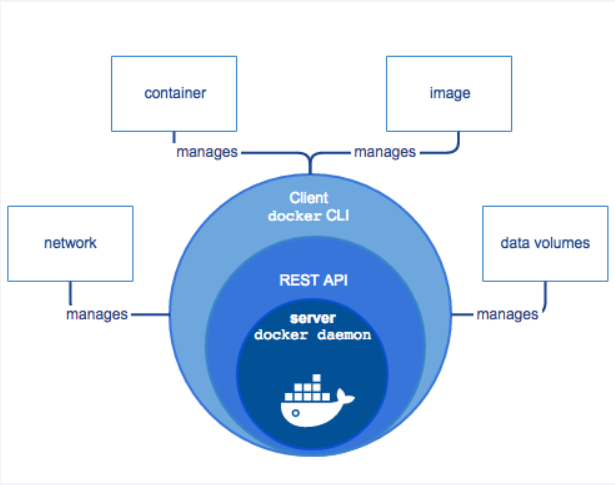
CHƯƠNG 1 – GIỚI THIỆU

1. GIỚI THIỆU VỀ ĐỀ TÀI

* Với nhu cầu tìm việc của nguồn lực trường, Trường đại học Tôn Đức Thắng muốn tăng cường cơ hội tìm việc cũng như có thêm một kênh thông tin về cơ hội việc làm cùng với các kênh khác như topcv, glint, hoặc facebook. Nhầm để đáp ứng với nhu cầu đó thì chung em sẽ xem và xây dựng một kênh tìm việc cũng như đăng các tin tuyển dụng của các nhà tuyển dụng lên để tạo cơ hội việc làm cho sinh viên của trường
* Kênh tìm kiếm việc này sẽ cho phép sinh viên tạo hồ sơ ứng tuyển của riêng mình cũng như dễ dàng tìm được công việc phù hợp dựa vào nguồn việc làm từ các đối tác lớn của nhà trường cũng như các công ty nhỏ và cần nguồn nhân lực chất lượng cao

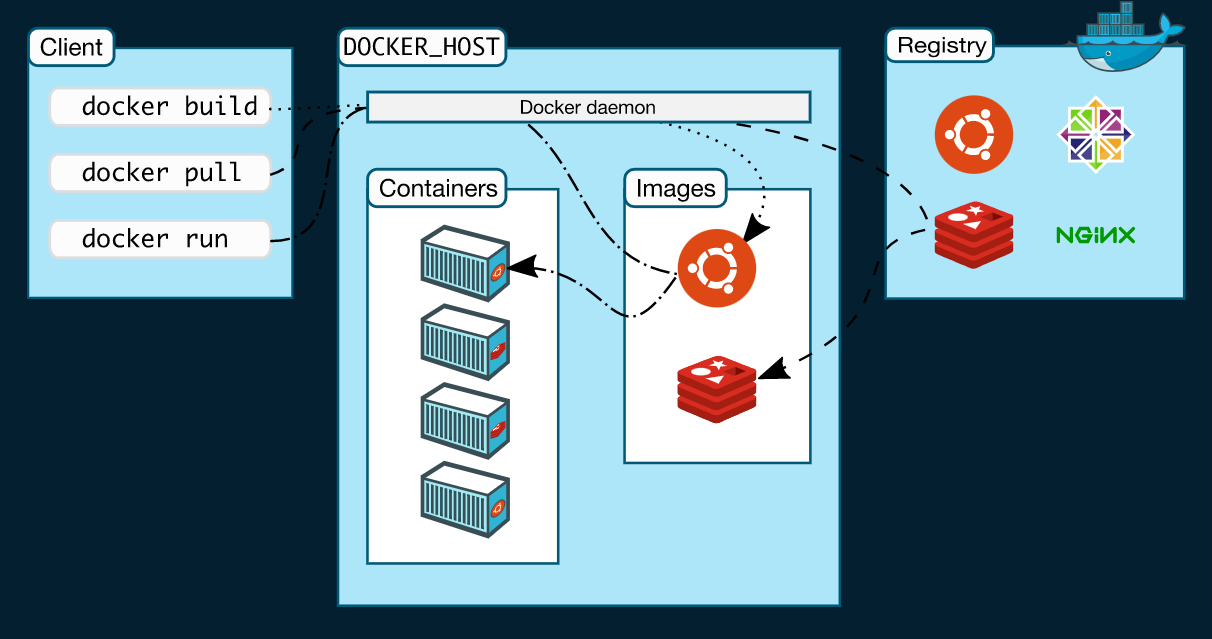
1. GIỚI THIỆU VỀ CÔNG NGHỆ VÀ THUẬT NGỮ
2. Docker

* Để một source code được khởi động và hoạt động trơn tru ở một máy tính đòi hỏi máy tính đó phải có đầy đủ tài nguyên như môi trường , các biến , một số thư viện , cũng như một số công cụ để hỗ trợ cho ngôn ngữ và công nghệ đó hoạt động. Với Docker bạn có thể quản lý các tài nguyên của source code một cách dễ dàng và giảm độ trễ giữ việc viết code với việc chạy chúng trong sản phẩm của bản thân.
  + VD: Nếu một máy tính đang chạy một phiên bản môi trường ví dụ như NodeJs ở phiên bản 16 nhưng ở một máy của người đồng nghiệp lại chạy NodeJs ở phiên bản cao hơn hoặc thấp hơn điều này sẽ dẫn đến việc là khi source code chạy ở máy mình hoạt động bình thường nhưng khi sang máy của bạn hoặc đồng nghiệp lại không được vì không hỗ trợ cách viết JS đó hoặc thiếu gì đó và dẫn đến mấy thời gian rất nhiều
* Chính vì thế thì Docker( Container ) được sinh ra để giải quyết những vấn đề này
  + Docker là một công cụ để cung cấp cho việc Build , Deploy, và Run source code ứng dụng hay là một trang web nào đó một cách dễ dàng. Và docker ban đầu được viết bằng ngôn ngữ Python và sau này là GoLang
  + Để thực hiện các việc như Build , Deploy , Run như thế thì Docker áp dụng 1 thuật ngữ là Containers . Containers cho phép người lập trình viên đóng gói source code với tất cả các điều kiện mà source code cần để chạy được như là thư viện , môi trường , hay là công cụ hỗ trợ.
  + Dựa vào Container sau khi ta khai báo và đóng gói các package cần thiết thì ta có thể đưa nó vào máy chủ chạy bất kì hệ điều hành nào cũng được và phổ biến nhất ở đấy là Linux, Ubuntu , Centos,…. Và có thể là Windows của Microsoft tùy thuộc vào máy chủ mà người tập trình viên đó thao tác với nó
  + Sự độc lập và bảo mật cho phép bạn chạy riêng biệt nhiều Container cùng một máy chủ. Các container hoạt động chiếm một lượng dung lượng của máy chủ tùy theo độ lớn nhỏ của dự án mà ta đưa vào nó.
  + Docker cung cấp các công cụ và nền tảng để quản lý quá trình hoạt động và vòng đời container của bạn.
    - Phát triển ứng dụng của bạn và các thành phần hỗ trợ trong contrainers
    - Container cũng đóng vai trò là đơn vị cung cấp và thử nghiệm ứng dụng hay trang web của bạn
    - Nếu bạn có nhu cầu đưa ứng dụng hay trạng web của mình lên môi trường sản phẩm và đưa ra cho mọi người sử dụng dưới dạng là 1 container hoặc là một dịch vụ nào đó cho dù môi trường đó là nằm ở máy của bạn hay là nằm trên một máy chủ khác ở điện toán đám mây hoặc có thể là sử dụng cả hai.
* Docker Engine là gì ?
  + Một tiến trình Docker daemon có nhiệm vụ các bước ban đầu để tạo và quản lý các container, image , volume , network
  + Docker Client sẽ giao tiếp với Docker daemon thông qua API theo chuẩn REST ( tức có nghĩa sẽ có các phương thức GET , POST , PUT , DELETE ) để yêu cầu Docker daemon thực thi các lệnh mà người dùng gọi đến
  + Docker CLI được xem là Client hay được hiểu là giao diện tương tác giữa người dùng ở đây là sử dụng các công cụ hỗ trợ như là Command Line (CMD), Window Terminal, …. Để gửi các yêu cầu tương ứng đến docker daemon thông qua API



Hình 1: Sơ đồ về kiến trúc của docker

* Kiến trúc của Docker
  + Docker có kiến trúc là Client – Server và giao tiếp qua REST API (Cổng mạng và một số phương thức khác) để có thể gửi các yêu cầu đến Docker daemon. Việc giao tiếp này thực hiện các công việc như Run , Deploy , Build các docker container. Ta có thể cấu hình cho Docker Client – Docker Daemon trên cùng 1 máy , hoặc có thể dùng Docker Client để giao tiếp từ xa với Docker Daemon. Và minh họa cho kiến trúc của docker như sau

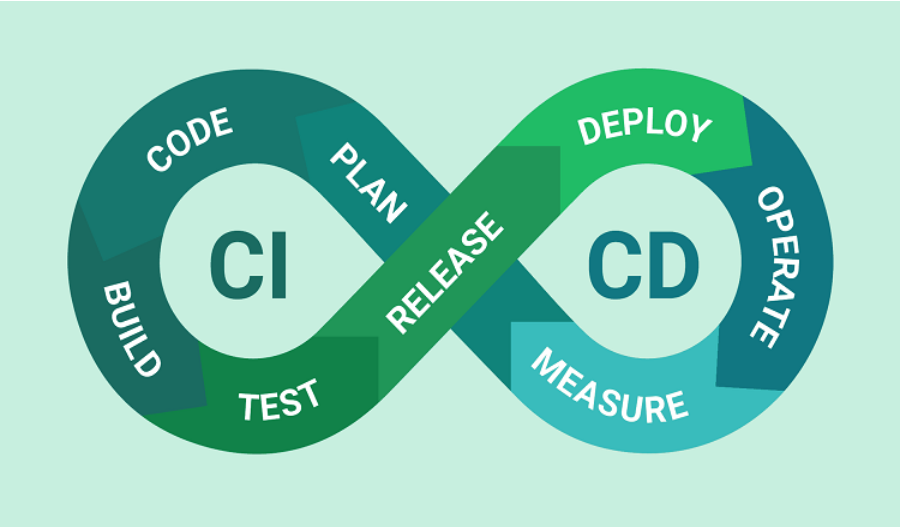


Hình 2 : Chi tiết và cách hoạt động và thao tác với docker

Một số thành phần khác của docker

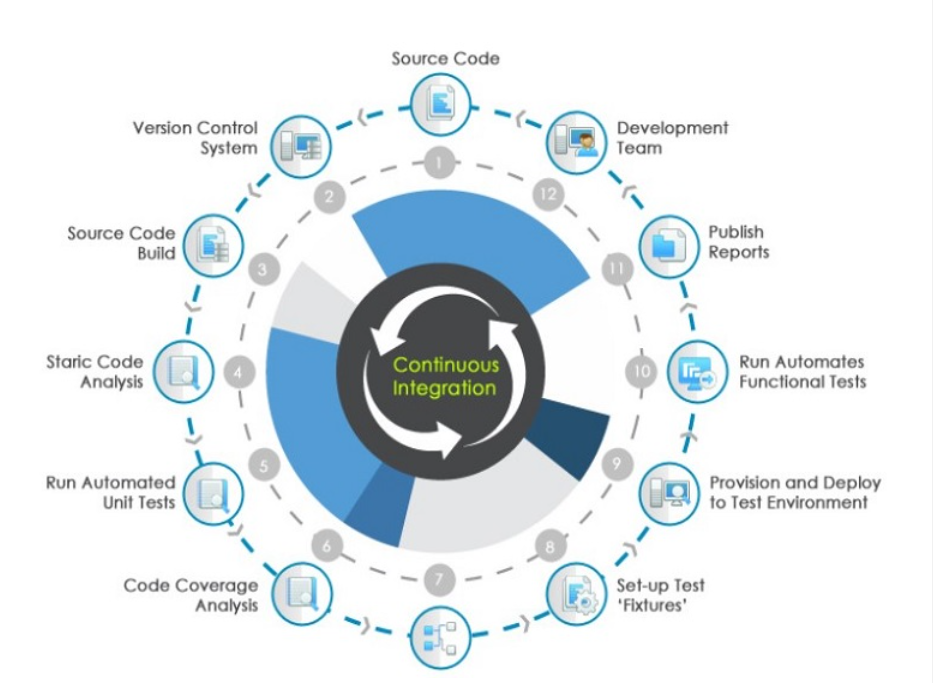
* Docker daemon
  + Daemon Docker (dockerd) lắng nghe các yêu cầu của người dùng thông qua Docker API và quản lý các Docker đối tượng như image, container, network và volume. Một daemon cũng có thể giao tiếp với các daemon khác để quản lý dịch vụ Docker.
* Docker Client
  + Docker client (docker) là cách tốt nhất để người thiết lập docker giao tiếp với no. Khi bạn sử dụng các câu lệnh như docker run, client thông qua Docker API sẽ gửi các câu lệnh này đến dockerd, thực hiện chúng. Docker client có thể giao tiếp với nhiều hơn một daemon Docker.
* Docker Registry
  + Docker registry sẽ là nơi lưu trữ các hình ảnh Docker. Docker Hub là nơi lưu trữ Docker image public (public registry) mà bất kỳ ai cũng có thể sử dụng và Docker được định cấu hình mặc định để tìm kiếm hình ảnh trên Docker Hub. Ngoài ra, người dùng có thể cấu hình các tư vấn đăng ký khác nhau để lưu trữ hình ảnh Docker.
* Khi người dùng sử dụng câu lệnh “docker pull” hoặc “docker run”, các image only sẽ được tải về dựa trên registry đã được cấu hình trước đó. Khi người dùng sử dụng câu lệnh “docker push”, image cũng sẽ được tải lên registry mà người dùng đã cấu hình từ trước đó.
* Docker image
  + Docker Image là template read-only (chỉ cho phép đọc) với các hướng dẫn để tạo Docker container. Image sẽ được sử dụng để đóng gói các ứng dụng và các thành phần đi kèm của ứng dụng, được lưu trữ ở server hoặc trên registry. Ví dụ bạn có thể sử dụng Dockerfile để tạo ra một Docker image sử dụng hệ điều hành Ubuntu và cài đặt Apache server với những cài đặt, cấu hình tùy chỉnh của riêng mình.
* Docker container
  + Container là 1 “runable instance” của image. Bạn có thể khởi tạo, dừng hoặc xóa container bằng cách sử dụng Docker API hoặc CLI. Bạn có thể kết nối container đến 1 hoặc nhiều network, thư mục lưu trữ, hoặc thậm chí tạo ra 1 image mới dựa trên tình trạng hiện tại của container. Mặc định 1 container được “cách ly” với các container và server nếu người dùng không có các cài đặt gì thêm.
* Docker volume
  + Volume được thiết kể để làm nơi lưu trữ các dữ liệu độc lập với vòng đời của container.
* Docker network
  + Cung cấp một private network mà chỉ tồn tại giữa container và server, giúp các container có thể giao tiếp được với nhau một cách dễ dàng.
* Docker service
  + Docker Service cho phép bạn mở rộng các contaner thông qua nhiều Docker daemon, chúng giao tiếp với nhau thông qua swarm cluster bao gồm nhiều manager và worker. Mỗi một node của swarm là 1 Docker daemon giao tiếp với nhau bằng cách sử dụng Docker API. Theo mặc định thì service được cân bằng tải trên các node.

1. CI/CD



Hình 3 : Quá trình hoạt động của CI/CD

* CI ( Continous Integration ) là cách thức phát triển phần mềm bằng cách tích hợp liên tục có nghĩa là nếu trong một nhóm những người cùng một dự án khi thực hiện một tính năng hay tham gia vào việc bảo trì một tính năng nào đó và sinh ra các đoạn mã mới thay cho đoạn mã cũ và đưa đoạn mã trên nhánh của người đó lên nhánh tổng ( master ) hoặc là nhánh develop thì ở đây CI sẽ thực hiện công việc của nó là tự động chạy toàn bộ source code cùng với đoạn mã của người viết đoạn mã vừa mới đưa lên và kiểm tra và check xem độ chính xác về mặt quy tắc viết mã của ngôn ngữ mà source đó sử dụng hay không. Nếu không thì sẽ tiếp tục bước tiếp theo nhưng khi có vẫn đề xảy ra thì sẽ báo lỗi cho người viết đoạn mã đó chưa khớp với đoạn mã trên nơi tổng mà các nhánh của các thành viên đổ về đó, và cũng như đưa ra những gợi ý cho người viết mã là một đề xuất để sửa. CI cũng đảm bảo source code của bạn sau gom mã của thành viên trong nhóm lại sẽ có thể hoạt động ở bất kì điều kiện nào và không xảy ra lỗi. Quá trình Build Run và Deploy sẽ được chạy và cho người quản lý source code tiến trình nó xảy ra như thế nào.
* Quá trình đưa mã code từ nhánh cá nhân lên nhánh chính ( develop hoặc master) thì quá trình CI hoạt động sẽ được hoạt động tự động nếu người quản lý source code cấu hình và cài đặt trước . Nến quá trình gom source code -> kiểm tra -> phản hồi sẽ diễn ra liền mạch và tự động. Người xây dựng mã người có thể tiết kiệm thời gian để xây dựng tính năng khác hoặc sửa lỗi nếu có và sau khi quá trình kết thúc thì người quản lý có thể xem được quá trình build và test.



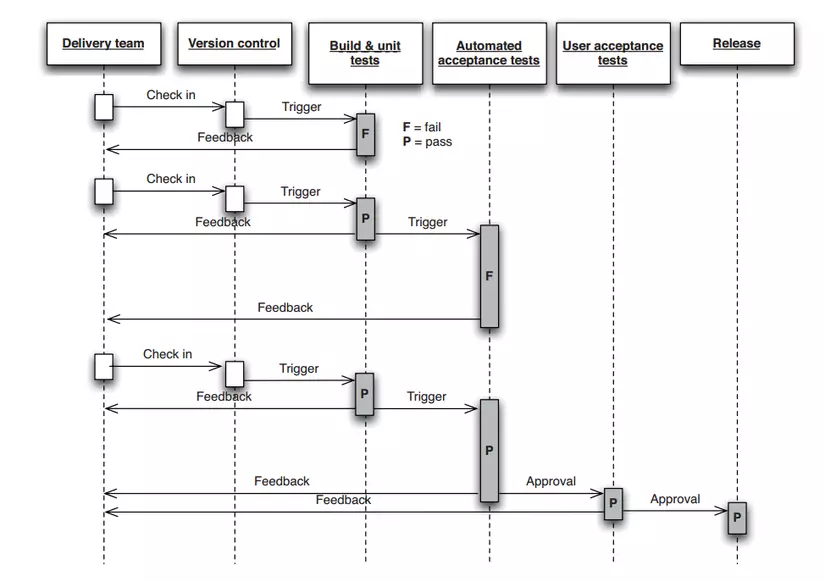
Hình 4 : Quá trình phát triển ứng dụng.

Lợi ích của việc sử dụng CI

* Giảm thiểu rủi ro nhờ việc phát hiện lỗi và fix sớm, tăng chất lượng phần mềm nhờ việc tự động test và inspect (đây cũng là một trong những lợi ích của CI, code được inspect tự động dựa theo config đã cài đặt, đảm bảo coding style, chẳng hạn một function chỉ được dài không quá 10 dòng code ...)
* Giảm thiểu những quy trình thủ công lặp đi lặp lại (build css, js, migrate, test...), thay vì đó là build tự động, chạy test tự động
* Sinh ra phần mềm có thể deploy ở bất kì thời gian, địa điểm

CD là gì : Thì đi kèm với CI là việc tích hợp liên tục thì còn một quá trình nữa gọi là chuyển giao liên tục thì nếu source code qua quá trình CI không có lỗi xảy ra thì sẽ sang một giai đoạn gọi là CD để tiến hành kiểm tra những thay đổi về mã nguồn đã được build lên và mã nguồn trong môi trường kiểm thử trước đó. CD cho phép người viết mã nguồn cài đặt những chu kì tự động ngắn và liên tục . Từ đó nhanh chống phát hiện ra lỗi sai và tiến hành chỉnh sửa ngay trong những phần tương tự.

* Do sự tiện lợi của 2 quá trình này thì người quản lý mã nguồn thường tích hợp cả hai CI/CD với nhau để tiết kiệm thời gian , chi phí và tài nguyên bộ nhớ cho 2 quá trình này, CD sẽ sử dụng giai đoạn cuối cùng của CI để bắt đầu quá trình chuyển giao của chính nó
* Trong quá trình CD thực hiện tiến trình của mình thì có thể mã nguồn sẽ trải qua một số phần testing được người quản lý mã nguồn thiết lập và một số bài test bao gồm UI Testing , API Testing, …. CD sử dụng một thuật ngữ được gọi là Deployment Pipeline giúp chia quy trình chuyển giao thành các giao đoạn. Mỗi giai đoạn sẽ đáp ứng những mục đích riêng bao gồm xác minh chất lượng của các tính năng từ một số góc độ khác nhau để kiểm định chức năng và tránh lỗi ảnh hưởng đến người dùng. Pipeline sẽ thể hiện phản hồi cho nhóm trong việc cung cấp tính năng mới hoặc sửa chữa các tính năng đã có sẵn. Deployment Pipelien là quy trình để chuyển phần mềm từ version control đến tay người dùng.
* Việc kết hợp 2 quá trình này thì ta sẽ được chu trình hoạt động của CICD gói gọn trong 3 bước là Xây dựng -> Kiểm tra -> Triển khai



Hình 5 : Cách hoạt động của CI/CD

Lợi ích của CI/CD

* **Giảm thiểu rủi ro không đáng có**

Đây có thể là một lợi ích vô cùng hữu ích của CI/CD, nó cho phép làm giảm thiểu đi những rủi ro nhờ việc phát hiện và sửa lỗi sớm, giúp tăng chất lượng sản phẩm nhờ khả năng tự động kiểm tra và quan sát. Không những vậy, những quy trình thủ công lặp đi lặp lại hằng ngày cũng được giảm tải, thay vào đó là xây dựng và kiểm thử tự động mà không cần đến sự giúp đỡ của con người. Một đặc điểm nữa của CI CD chính là có thể deploy, triển khai phần mềm ở bất cứ địa điểm và thời gian nào

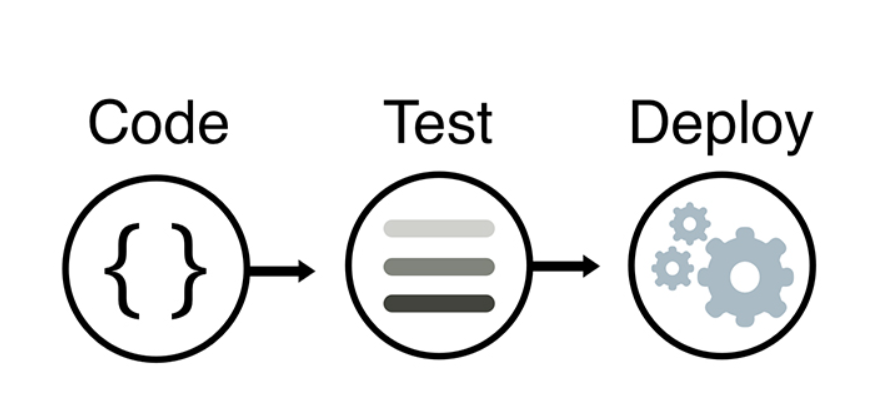
* **Thay đổi code nhỏ**

Một lợi ích vô cùng lớn của CI/CD chính là cho phép chúng ta tích hợp nhiều loại mã nhỏ cùng một lúc. Những thay đổi mã này được thực hiện một cách đơn giản và xử lý nhanh hơn so với những đoạn mã khổng lồ, từ đó làm giảm đi khả năng sinh ra những vấn đề liên quan đến việc thay đổi sau này.

Những sự thay đổi mã nhỏ này có thể được thực kiểm tra ngay sau khi chúng được tích hợp vào kho mã. Các nhà phát triển có thể dễ dàng nhận ra vấn đề trước khi lượng lớn công việc tăng lên một cách chóng mặt. Đây thực sự là một lợi thế đối với những nhóm phát triển lớn hoặc người làm việc từ xa giao tiếp được hiệu quả hơn.

* **Hạn chế những ảnh hưởng của lỗi**

CI/CD được thiết kế với hệ thống sao cho khi có lỗi nào đó xảy ra thì những kết quả tiêu cực sẽ bị giới hạn trong phạm vi ảnh hưởng nhất định nào đó. Việc hạn chế các vấn đề này giúp làm giảm khả năng hư hỏng từ đó làm cho hệ thống được bảo trì và xử lý một cách dễ dàng hơn.Với hệ thống CI CD, có thể đảm bảo cho việc cách ly lỗi sẽ được phát hiện một cách nhanh chóng và dễ dàng thực hiện hơn. Chính vì vậy, hậu quả của các lỗi trong ứng dụng sẽ được giới hạn trong phạm vi ảnh hưởng của nó.



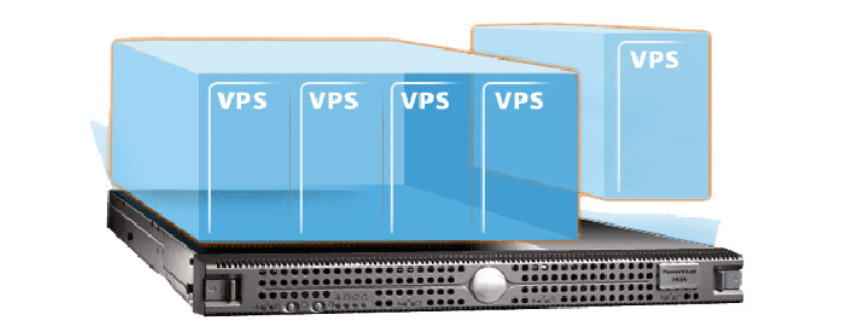
Hình 6 : Tổng quan về quá trình hoạt động với CI/CD

Nhược điểm của CI/CD là gì ?

* Bên cạnh các ưu điểm thì hệ thống CI và CD vẫn tồn tại một số điểm trừ nhất định. Vì CI/CD thường xuyên được nâng cấp để tối ưu hơn nên đòi hỏi các nhà lập trình phải nhanh chóng học hỏi các kiến thức mới.
* Ngoài ra, để quá trình tự động hóa diễn ra trơn tu, người sử dụng phải hiểu rõ cách vận hành của chương trình. Và điều này hoàn toàn không đơn giản bởi CI và CD sở hữu bộ kỹ năng khá phức tạp. Để hoàn toàn sử dụng trơn tru được CI/CD đòi hỏi quá trình rèn luyện lâu dài.
* Để có thể ứng dụng được CI và CD vào công việc đòi hỏi nhà sản xuất phải đầu tư trước một khoản phí khá lớn. Không nhiều công ty dám mạo hiểm bởi như đã nói ở trên, việc sử dụng CI và CD không dễ dàng. Và nhược điểm cuối cùng là một số hệ thống lập trình không cho phép hỗ trợ thêm CI và CD. Nếu không thể thay đổi hệ thống thì khó lòng sử dụng được chương trình này.

1. VPS

* VPS là viết tắt của ( Virtual Private Server) hay còn gọi là máy chủ riêng ảo. Được cung cấp và cho thuê bởi các công ty dịch vụ Internet Hosting được tạo ra bới công nghệ ảo hóa.
* VPS hoạt động như một server để người dùng tạo ra trang web hay ứng dụng trên các tài nguyên ta cài đặt được giành riêng
* Người cài đặt VPS có toàn quyền kiểm soát bằng superuser ( Root , quyền admin ), và thông qua SSH key để điều khiển cài đặt các thư viện, hoặc cấu hình VPS theo ý muốn của người cài đặt.
* Thường thì VPS sẽ được thuê và có những option các hệ điều hành có sẵn khi người muốn thuê lựa chọn các dịch vụ và trả tiền để thành toán phí duy trì của cái máy ảo đó ở máy chủ của công ty hosting đó một số hosting phổ biến như AWS , Digital Ocean , Microsoft Azure , Ở Việt Nam thì có Mắt Bão , ….. trên điện toán đám mây
* VPS an toàn và ổn định tùy theo tình trạng mạng của bạn kết nối và ổn định hơn so với host website vì bạn không chia sẻ không gian lưu trữ với người khác. Và nó cũng có giá thành thấp hơn so với thuê một server riêng



* Thường thì các máy chủ ảo sẽ không có tên miền để thay thế cho việc truy cập đến máy chủ đó mà phải truy cập qua IP của VPS đó được sinh ra khi bạn tiến hành thuê VPS đó
* VPS là một server ảo hoạt động tương tự như một máy chủ vật lý. Máy chủ , hay server, là một máy tinh chứa source code và cơ sở dữ liệu cho ứng dụng của bạn , Môi trường và các thư viện cần thiết. Bất kể khi nào người dùng truy cập vào ứng dụng của bạn trên trình duyệt thì sẽ gửi yêu cầu đến con server của bạn và lúc này dựa vào đường truyền internet các file cần thiết để chạy 1 trang web sẽ được truyền xuống máy tính của người dùng và hiện thị nó trên tình duyệt
* Trên thực tế thì VPS là máy chủ ảo được cấu hình qua mạng như nó sẽ được hình thành trên một server vật lý, và mỗi VPS là sẽ có phần tài nguyên riêng biệt của nó và không bị ảnh hưởng bởi các VPS khác
* Các yếu tố lưu ý đến VPS khi tiến hành thuê
  + **Ram** : Trong máy tính, RAM là loại bộ nhớ chính, nếu VPS của bạn có nhiều RAM thì khả năng truy xuất dữ liệu càng tốt. Bởi vì khi sử dụng VPS bạn sẽ cần RAM để xử lý các vấn đề như xử lý truy vấn nhập xuất của database
  + **SWAP** hay là bộ nhớ ảo có nhiệm vụ lưu lại các hành động xử lý trong quá khứ trong trường hợp bộ nhớ RAM bị quá tải (overload). Hay nói cách dễ hiểu hơn, SWAP là nơi lưu trữ trên ổ cứng thay vì một bộ nhớ độc lập. Tuy nhiên, không phải VPS nào cũng hỗ trợ SWAP
  + **Disk** : Ổ cứng/Ổ đĩa cứng hay còn gọi là Disk là không gian lưu trữ được sử dụng để lưu các file cài đặt của hệ điều hành và các file của mã nguồn website. Có 2 loại ổ đĩa được sử dụng phổ biến, đó là:
    - HDD (Hard Disk Drive): Là loại ổ đĩa thông dụng được sử dụng trên máy tính.
    - SSD (Solid State Drive): SSD hay còn gọi là ổ cứng bán dẫn được sử dụng để lưu trữ dữ liệu nhưng tốc độ truy xuất dữ liệu nhanh hơn so với HDD 300 lần.
    - Do vậy, giá thành của ổ cứng SSD sẽ có đắt hơn so với loại ổ HDD.
  + **CPU Core :** là lõi xử lý của CPU. Một máy chủ riêng sẽ có lượng core nhất định và nó được chia cho các VPS. Thông thường, số core càng cao thì khả năng xử lý dữ liệu càng tốt. Số lượng core trung bình ở các gói VPS là từ 1 core đến 3 core.
  + **Bandwidth và transfer :**  đều có nghĩa là băng thông, chính là lưu lượng được phép truyền tải dữ liệu đi.
  + **IP (Internet Protocol) :** là số lượng địa chỉ IP mà nhà cung cấp dịch vụ VPS sẽ cấp cho bạn.
* Thì các công ty cho thuê dịch vụ máy chủ ảo riêng biệt như thế sẽ cho người thuê toàn quyền quyết định về hệ điều hành, cũng như cho phép chọn cấu hình của phân vùng VPS đó như sử dụng bao nhiều RAM xử lý , Bao nhiêu GB bộ nhớ , Thì Ram và bộ nhớ càng cao ứng dụng của bạn sẽ hoạt động một cách trơn tru hơn nhưng bù lại chi phí của của người thuê bỏ ra để duy trì nó sẽ càng cao hơn.



Ưu điểm của VPS là gì ?

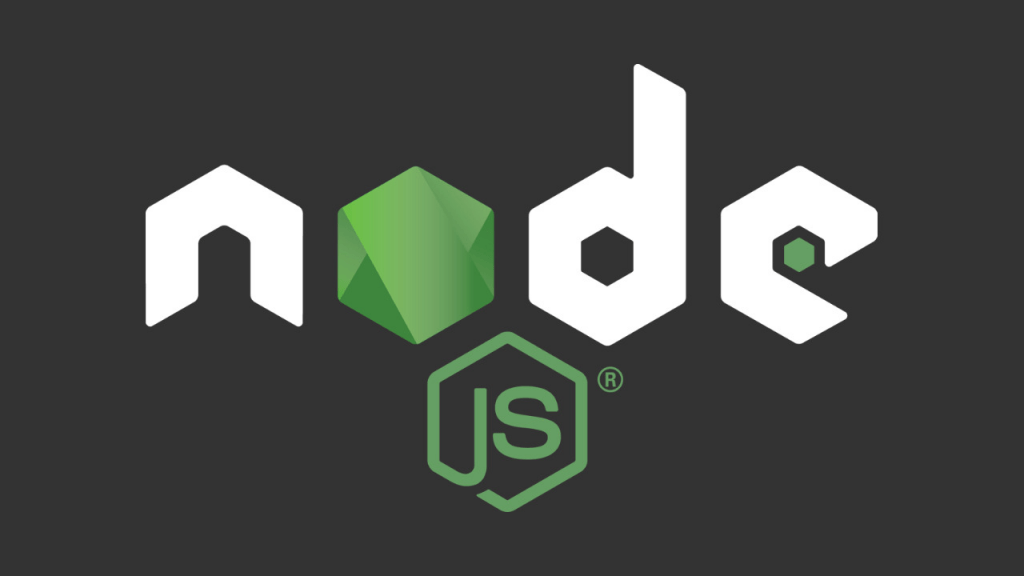
* Nhanh và đáng tin cậy hơn server shared hosting.
* Vì được đảm bảo về thông số server như bộ nhớ và sức mạnh vi xử lý, bạn sẽ không gặp phải vấn đề tài nguyên bị người khác dùng hết.
* Các vấn đề về lượng truy cập đột biến tăng cao không ảnh hưởng đến site của bạn.
* Bạn có quyền superuser (root) trên server.
* Có độ riêng tư cao hơn, vì files và databases bị khóa khỏi hệ thống server của các người dùng khác.
* Dễ dàng nâng cấp. Ngay khi website tăng trưởng, bạn chỉ cần nâng cấp gói hosting để nâng tài nguyên lên mà không phải tốn công chuyển dữ liệu hay chuyển server (RAM, CPU, disk space, bandwidth,…).
* IP riêng. Chính vì điều này khả năng chống DDoS cũng cao hơn.

Nhược điểm

* Giá cao hơn shared hosting.
* Cần ít nhiều kiến thức kỹ thuật để quản lý.
* Cấu hình server không đúng có thể tạo ra lỗ hổng bảo mật.

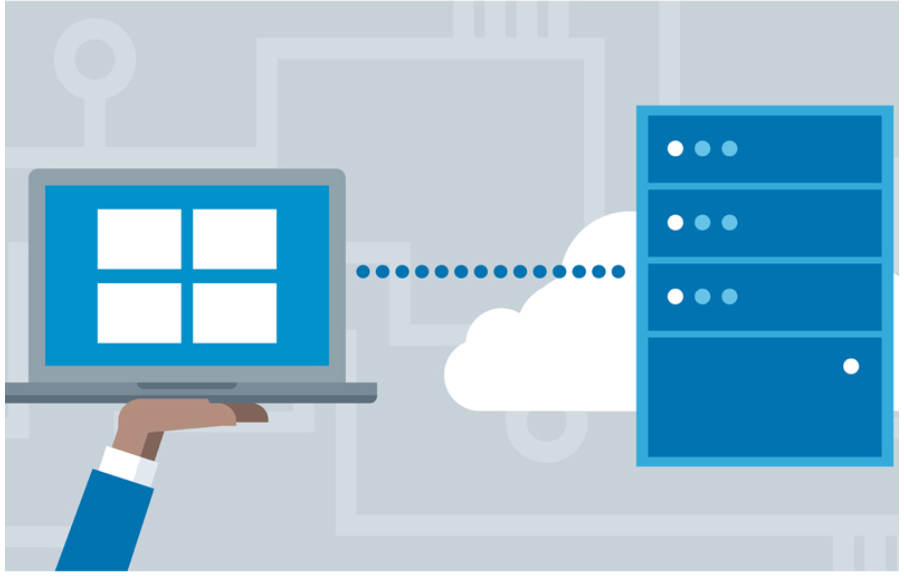
1. NODEJS(EXPRESS)

* NodeJS được xem là nền tảng để viết và chạy các mã nguồn thuộc về ngôn ngữ JavaScript, Nó cho biến ngôn ngữ JavaScript từ một ngôn ngữ chỉ hoạt động trên các client web thành một thứ mà các lập trình viên có thể sử dụng nó để tạo ra các ứng dụng độc lập trong các lĩnh vực như
  + Ứng dụng trò chuyện trong thời gian thực: Nhờ vào cấu trúc không đồng bộ đơn luồng, Node.JS rất thích hợp cho mục đích xử lý giao tiếp trong thời gian thực. Nền tảng này có thể dễ dàng mở rộng quy mô và thường dùng để tạo ra các chatbot.
  + Internet of Things (IoT): Các ứng dụng IoT thường bao gồm nhiều bộ cảm biến phức tạp để gửi những phần dữ liệu nhỏ.
  + Truyền dữ liệu: Netflix là một trong số những công ty lớn trên thế giới chuyên sử dụng Node.JS cho mục đích truyền dữ liệu.
  + Các SPA (Single-page application) phức tạp: Trong SPA, toàn bộ ứng dụng được load vào trong một trang duy nhất, do đó sẽ có một số request được thực hiện trong nền.
  + Các ứng dụng REST dựa trên API: JavaScript được sử dụng trong cả frontend lẫn backend của trang. Do đó một server có thể dễ dàng giao tiếp với frontend qua REST API bằng Node.js
* Express JS: là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile
  + Về các package hỗ trợ: Expressjs có vô số các package hỗ trợ nên các bạn không phải lo lắng khi làm việc với Framework này.
  + Về performance: Express cung cấp thêm về các tính năng (feature) để dev lập trình tốt hơn. Chứ không làm giảm tốc độ của NodeJS.



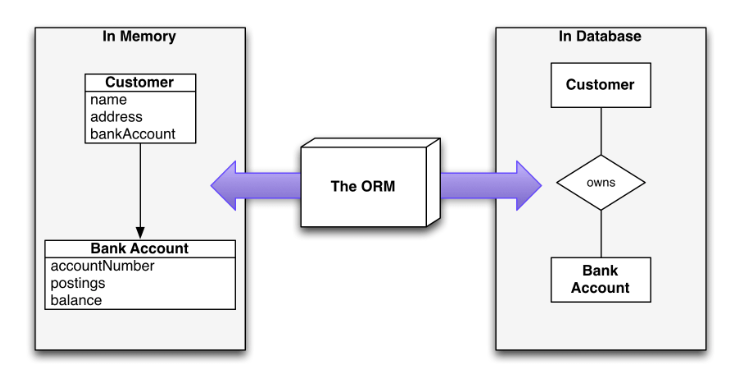
1. ORM (PRISMA)

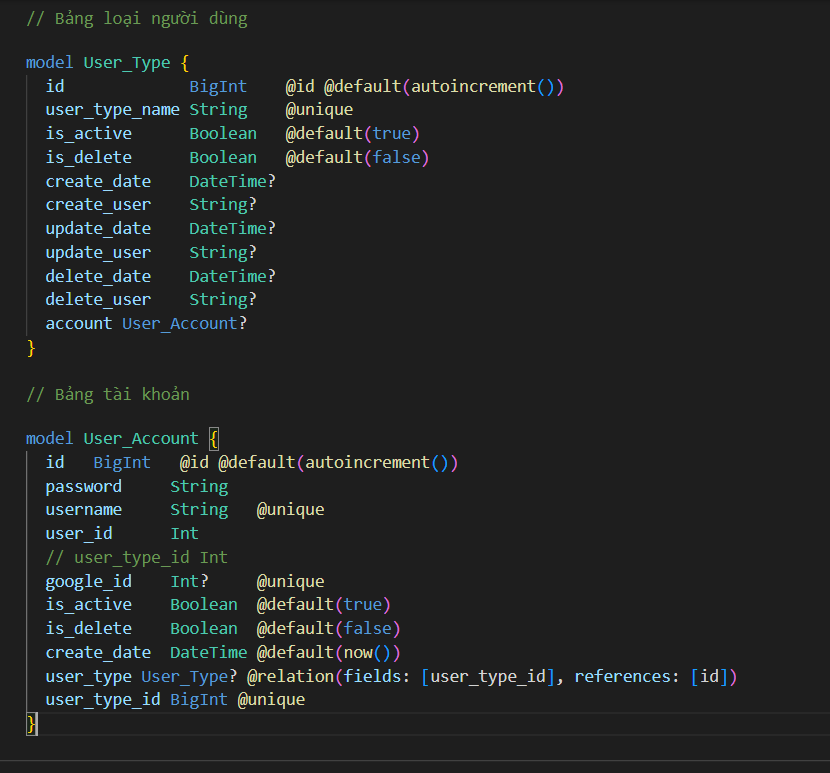
* SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS.
* SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản lý từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có thể kết hợp và hoạt động với các nền tảng khác nhau
* Và để người viết code giao tiếp với SQL server sẽ được giao tiếp với nhau thông qua một ngôn ngữ gọi là SQL , T-SQL
  + T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ,...
  + SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.



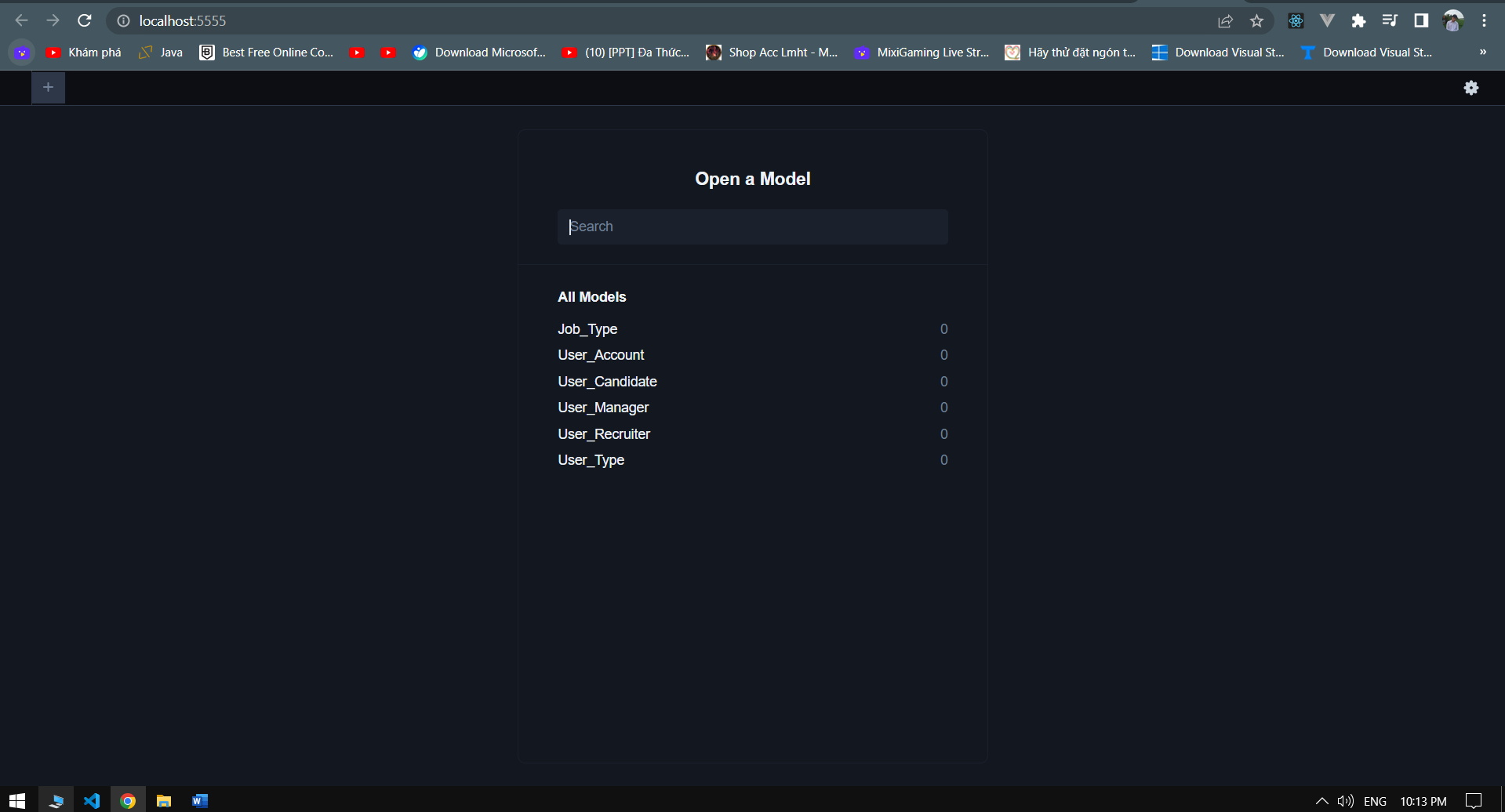
* Thì hiện nay trong thực tế có một số cách thực hiện tạo Database mà không cần thông qua GUI của SQL server mà dùng các đối tượng được định nghĩa và nhờ vào thư viện để ánh xạ các đối tượng đó vào trong cơ sở dữ liệu để hình thành các bảng và mối quan hệ của chúng thì kĩ thuật đó được gọi là ORM
* Vậy ORM là từ viết tắt của (Object Relational Mapping), là một kỹ thuật/cơ chế lập trình thực hiện ánh xạ CSDL sang các đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C# …(các table tương ứng các class, mối ràng buộc giữa các table tương ứng quan hệ giữa các class ‘has a’ , ‘is a’).
* Việc sử dụng ORM cho phép lập trình viên thao tác với database 1 cách hoàn toàn tự nhiên, dễ hiểu thông qua các đối tượng. Lập trình viên không cần tới loại database, kiểu dữ liệu trong database…
* Ưu điểm của ORM:
  + ORM giúp lập trình viên tập trung vào lập trình hướng đối tượng
  + Tính độc lập: Làm việc được với nhiều loại database(hệ quản trị cơ sở dữ liệu), nhiều kiểu dữ liệu khác nhau. Dễ dàng thay đổi loại database hơn. Các câu lệnh SQL không phụ thuộc vào loại database.
  + Đơn giản, dễ sử dụng: Hỗ trợ HSQL, cung cấp nhiều nhiều API truy vấn.
  + Năng suất hơn: viết code ít hơn, dễ hiểu hơn. Phù hợp các case CRUD (Create, Read, Update, Delete)
  + Khả năng sử dụng lại code.
* Nhược điểm của ORM:
  + Khả năng truy vấn bị hạn chế, nhiều trường hợp ta vẫn phải dùng native SQL để truy vấn database.
  + Khó tối ưu câu lệnh SQL (do câu lệnh SQL được ORM tự động sinh ra).
* Một thư viện hỗ trợ việc ánh xạ dữ liệu từ đối tượng trong khi xây dựng back-end bằng NodeJS đó là thư viện Prisma . Thư viện giúp ánh xạ các đối tượng ta khai báo từ dạng model được quy định thành các bảng trong cở sở dữ liệu mà ta muốn thì ở đây chúng em chọn SQL Server
* Prisma là một Object Relational Mapping (ORMs) được dùng để xây dựng các máy chủ như GraphQL Server, SQL Server, RESTful APIs, microservice, ... Prisma đơn giản là 1 layer nằm giữa Webserver và Database. Prisma giúp chúng ta giao tiếp với db một cách dễ dàng hơn.
* Prisma bao gồm ba phần chính:
  + Prisma Client : Trình tạo truy vấn an toàn và được tạo tự động cho Node.js và TypeScript.
  + Prisma Migrate : Một hệ thống di chuyển và mô hình hóa dữ liệu khai báo.
  + Prisma Studio : Một GUI để xem và chỉnh sửa dữ liệu trong cơ sở dữ liệu của bạn.
  + Cách truyền thống mà Webserver giao tiếp với Database là thông qua các câu lệnh SQL Query như SELECT, UPDATE hay DELETE. Giờ đây, nhờ vào các công cụ ORMs nói chung và Prisma nói riêng. Chúng tạo ra một tầng abstraction giữa Webserver và Database. Điều này giúp cho lập trình viên dễ dàng trong việc thao tác với Database. Thay vì viết những câu lệnh SQL khô khan, có thể sai bất cứ lúc nào thì chúng ta có thể viết các hàm tương ứng.







* Thì trên đây là 2 bảng được khởi tạo trong database được khởi tạo thành 2 bảng là User\_type và User\_Account thì các giá trị bắt buộc không được bỏ trống thì sẽ không có dấu ? sau kiểu dữ liệu
* Và cũng như khóa ngoại của bảng User\_account sẽ là id trong bảng User\_type
* Thì với thư viện Prisma ta có thể ánh xạ các model thành các bảng trong cở sở dữ liệu và thư viện này có hỗ trợ một số loại cơ sở dữ liệu phổ biến hiện này như: PostgreSQL , MySQL, SQLite, MongoDB, SQL Server
* Để tạo bảng trong các cơ sở dữ liệu ta chỉ cần thực hiện các thao tác sau
  + npx prisma studio : để mở GUI cho người viết code thấy được giao diện của prisma cùng với các bảng của mình



* + npx prisma migrate : Để khởi tạo các bảng lưu lại quá trình thay đổi cũng như các file có dạng SQL ứng với mỗi migrate sẽ có 1 file SQL sinh ra và Prisma sẽ dựa vào tất cả các file SQL ở từng thư mục trong migrate để khởi tạo cơ sở dữ liệu cũng như là cập nhật các thuộc tính trong quá trình ta thao tác với các model
  + Sau khi chạy câu lệnh này ta trong SQL Server ta khởi tạo và gán địa chỉ vào config để khi tạo từ prisma thì các bảng sẽ vào cơ sở dữ liệu đó sẽ hình thành các bản như các model mà ta đã khai báo trong Prisma ở NodeJS

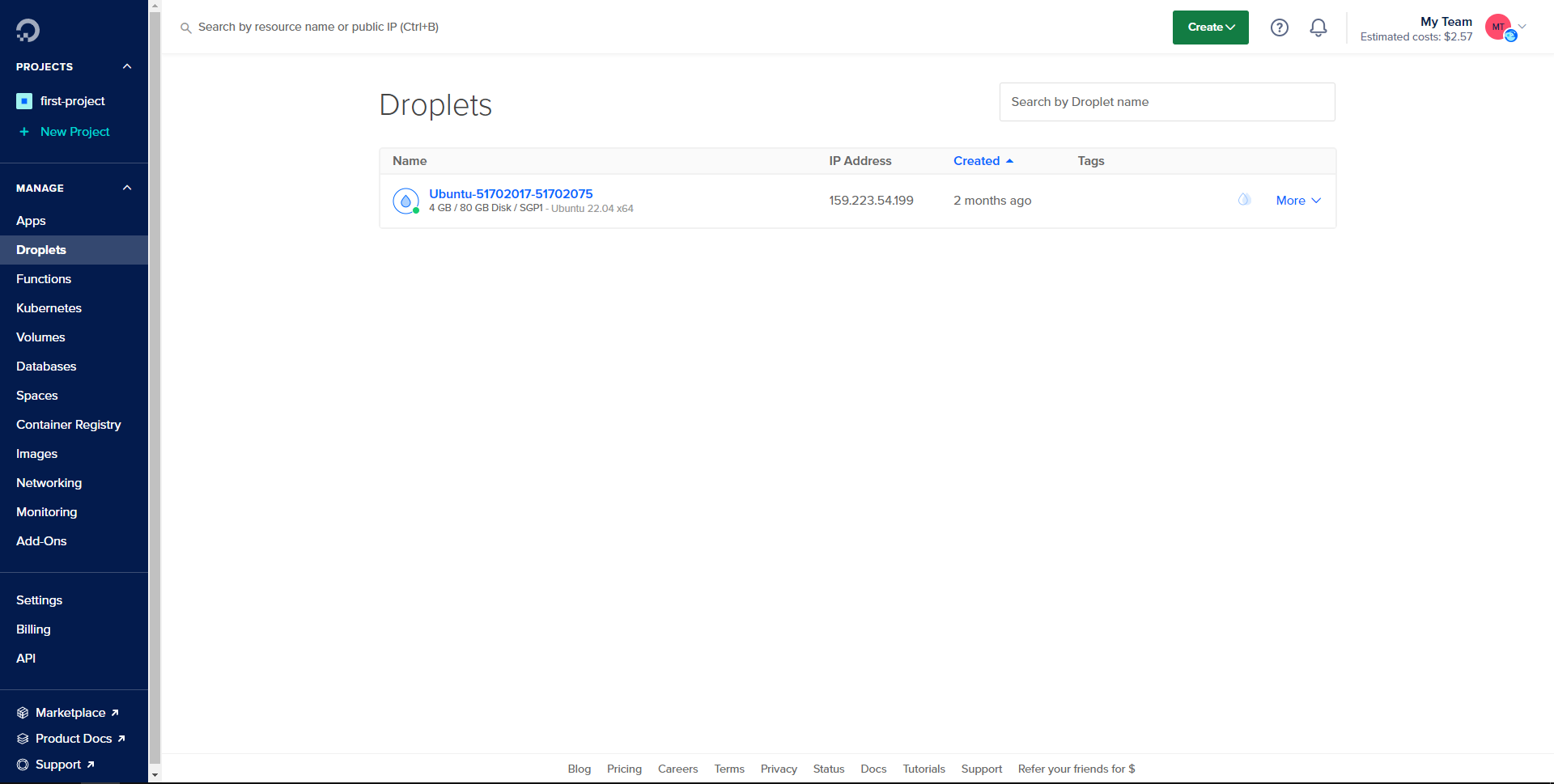
1. Nội dung của chương này

* Giới thiệu về đề tại về xây dựng trang ứng dụng đăng tin tuyển dụng cũng như ứng tuyển dành cho sinh viên trường Đại học Tôn Đức Thắng
* Giới thiệu về các công nghệ sử dụng trong quá trình xây dựng trang ứng dụng này

CHƯƠNG 2 – QUÁ TRÌNH THỰC HIỆN

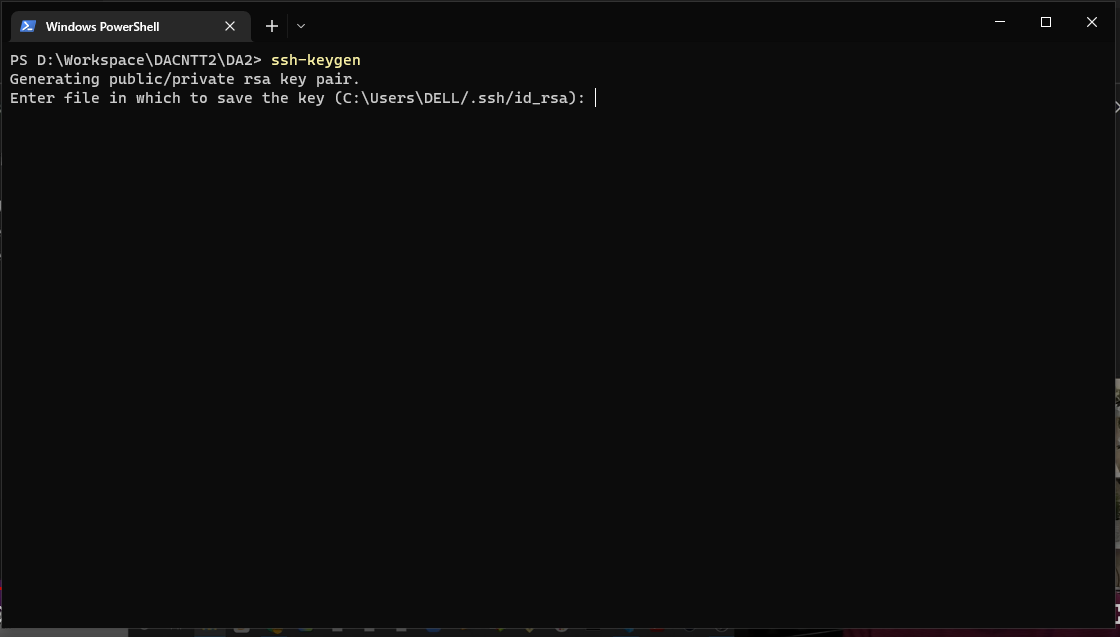
1 Quá trình thiết lập server

* Để đưa source code lên server ta cần một server ở đây chúng em thuê 1 con server chạy hệ điều hành Ubuntu với 4GB ram và 80GB bộ nhớ của Digitalocean với chi phí duy trì hàng tháng là 25 đô (600.000 VND)
* Quá trình thuê thì sẽ đưa thẻ visa của bản thân vào tài khoản để thuê server và mỗi tháng thì digitalocen sẽ trừ khoản tiền đó



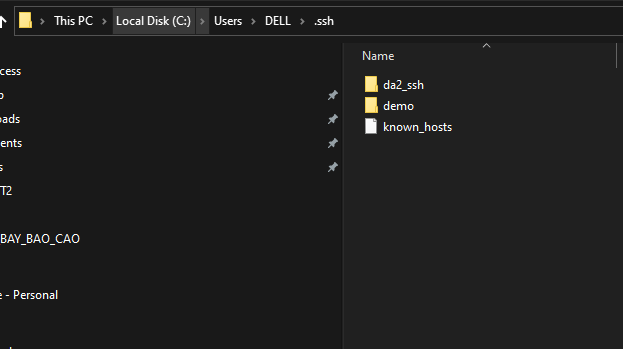
Hình 7 : Hình ảnh thuê server

* Khi đã có server thì bước tiếp theo thì sẽ cấu hình cho việc kết nối với server đó quá một số công cụ như cmd , Window Terminal, …., và các bước như sau:
  + Bước 1 : ta sẽ sử dụng ssh-keygen để tạo ra 1 chuỗi các kí tự ngẫu nhiên được gọi là key để sau khi thiết lập mỗi lần muốn kết nối với server ta chỉ cần gõ lệnh ssh@root + tên server và nhập mật khẩu của server thì lúc này ta sẽ kết nối được với server của mình vừa thuê



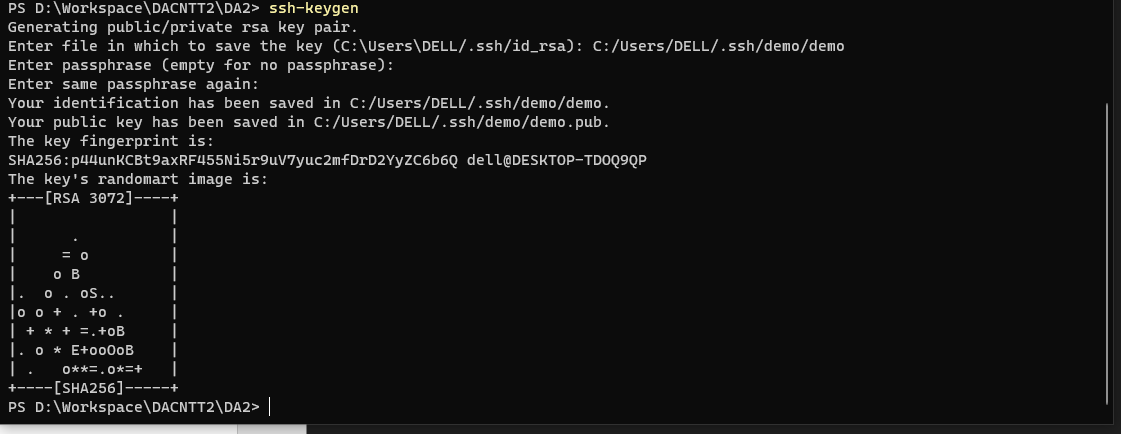
Hình 8 : Gõ ssh-keygen để tạo ra bộ mã cho riêng mình

* + Bước 2 ta vào thư mục C:/Users/(Tên của user)/.ssh và tạo ra 1 thư mục ví dụ ở đây máy cá nhân này có thư mục (demo) để mô phỏng lại quá trình tạo ssh-keygen và da2\_ssh đã tạo trong quá trình thực hiện đề tài này

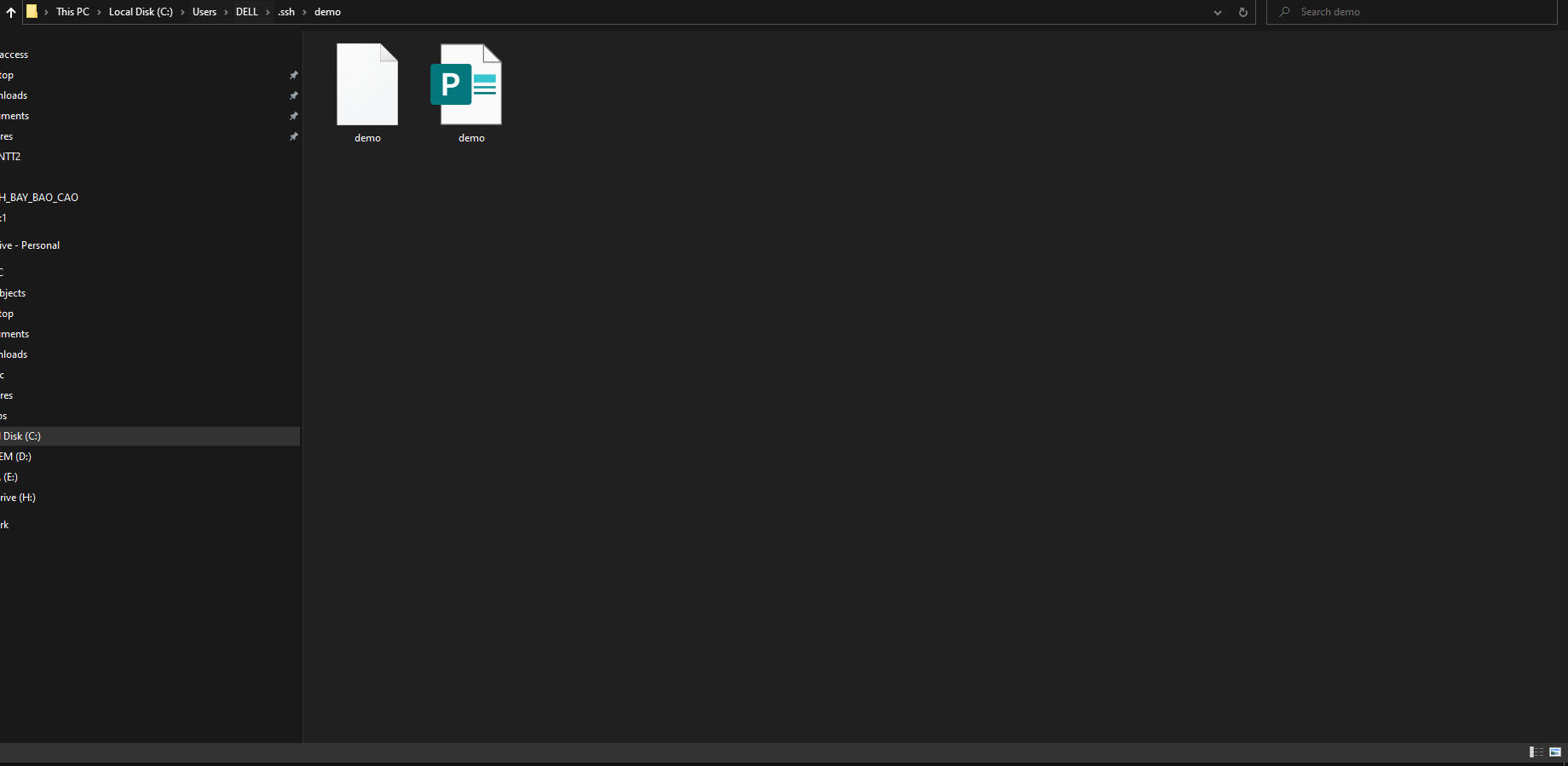


Hình 9 : Thư mục .ssh chứa ssh-keygen

* + Tiếp tục ta ở Window Terminal ta gõ lệnh để với thư mục demo ta gõ lệnh chuyển trỏ nơi tạo file chưa public key và private key

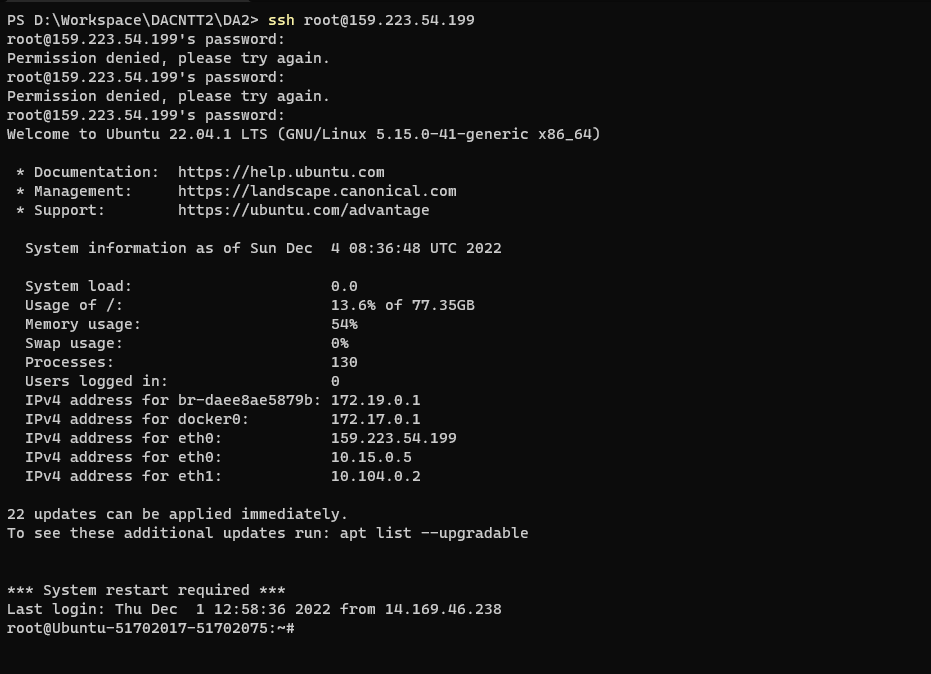


Hình 10 : Hình ảnh tạo ssh key trong thư mục demo



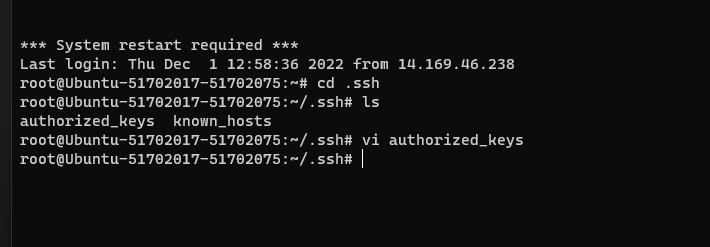
Hình 11 : Hình ảnh thư mục demo chứa ssh key ta vừa tạo với demo

* + Trong thư mục ta vừa tạo thì sẽ có 2 tập tin bào gồm 1 file demo thường và 1 file demo có đuôi là pub tức nghĩa là public key còn file còn lại là secret key
  + Sau khi khởi tạo ssh key ở máy cá nhân ta truy cập với server thông quá ssh bằng dòng lệnh



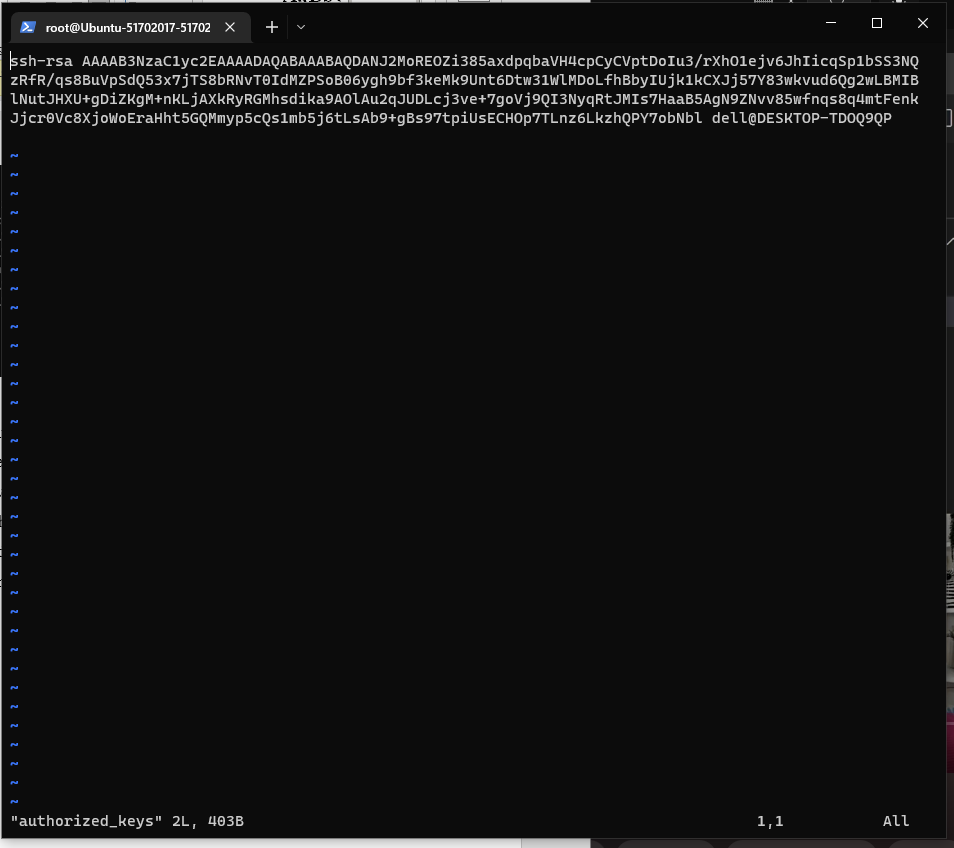
Hình 12 : Kết nối với server thuê được bằng ssh

* + Ta sẽ sử dụng ssh root@(IP) mà nhà cung cấp dịch vụ vps cung cấp cho ta để kết nối và nhập mật khẩu của server để truy cập vào server
  + Khi này ta sẽ có quyền thao tác với server lúc này ta sẽ add ssh được sinh ra ở máy cá nhân



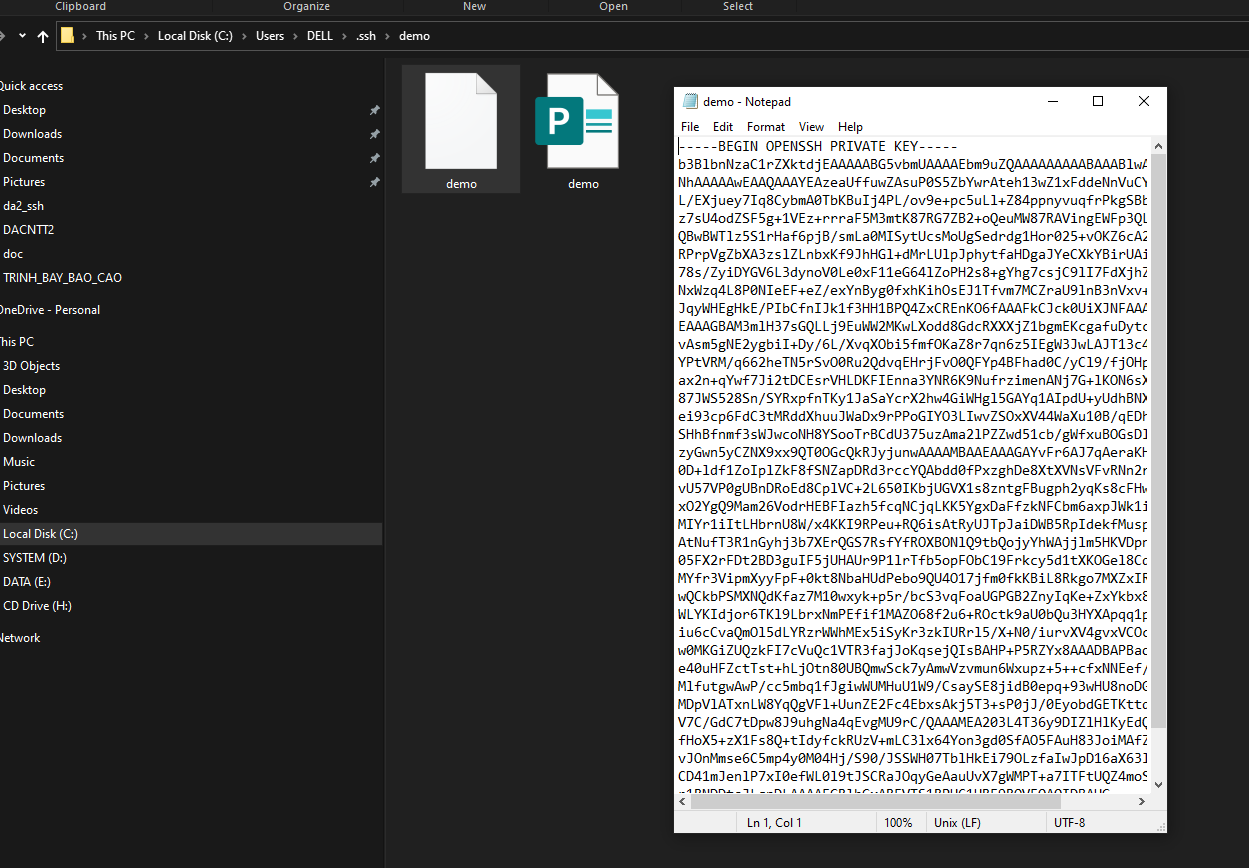
Hình 13 : Thêm ssh key vào máy chủ

* + Khi này ta thực hiện chuyển thư mục bằng cd .ssh và ls để check xem đã có file authorized\_keys hay chưa nên chưa ta sẽ tạo file này bằng sử dụng câu lệnh vừa cd vừa tạo bằng nano “ cd .ssh && sudo nano authorized\_keys “ để chuyển đến thư mục .ssh và có quyền như root sử dụng lệnh viết nano trên tập tin authorized\_keys
  + Ở đây máy chủ đã có tập tin authorized\_keys nên em sẽ dùng vim (vi) để xem nội dung thư mục



Hình 14 : Nội dung tập tin authorized\_keys

* + Nếu là lần đầu thiết lập tập tin này không có gì và ta sẽ thực hiện thao tác nhân nút I trên bàn phím để vào chế độ viết và ta quay lại máy cá nhân thư mục .ssh ta tạo có 2 file public và file thường

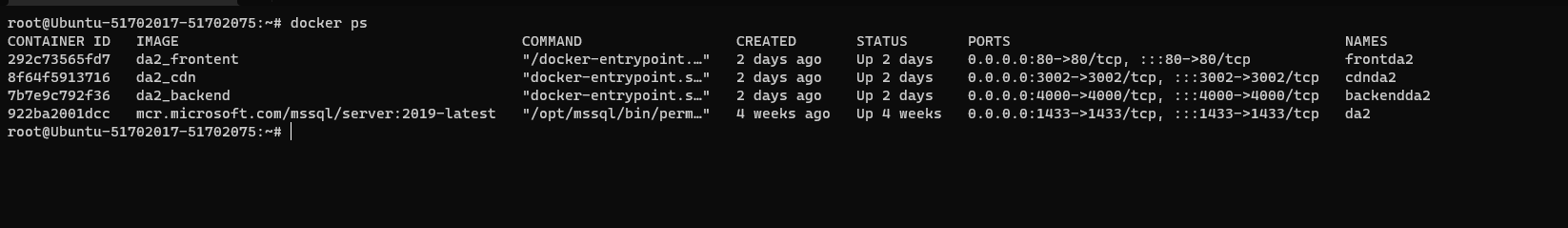


Hình 15 : Ta sẽ copy key trong file private key

* + Và ta sẽ past private key này vào file ở server thì đây là demo sẽ có nội dung khác với key đã được thêm vào server trước đó
  + Sau khi thêm private của máy cá nhân vào server thì ta nhẫn :x để thoát và lưu lại nội dung đã ghi với vim
* Sau khi đã lưu lại private key ta tiến hành update và cài đặt một số thư viện với các dòng lệnh như sau:
  + Sudo apt update : Để Ubuntu sẽ tiến hành việc tải các tài nguyên cần thiết để cấu hình server
  + Sudo apt upgrade : Để nâng cấp phiên bản nếu tài nguyên sau khi update còn quá cũ
  + Sudo apt-get install ca-certificates : Cài đặt tài nguyên này để cho việc chứng thực số cho server này có thể hoạt động trong môi trường mạng
  + Sudo apt-get install curl : Để cài đặt 1 công cụ (tool) có thể cài đặt một số thư viện khác ở đây để tải 1 số thư viện
  + Sudo apt-get install gnupg : Để cài đặt một phần mềm bảo mật cho server
  + Sudo apt-get install lsd-realase : Để cài đặt phiên bản linux tiêu chuẩn nhất giành cho máy chủ nếu phiên bản linux của máy chủ quá cũ
  + Ta sử dụng 1 câu lệnh như sau :

**Curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg –dearmor - /usr/share/keyrings/docker-archive-keyring.gpg** Để thêm khóa mã hóa cho docker (GPG – GNU Private Guard)

* + sudo apt-get install docker-ce docker-ce-cli containerd.io : Để cài đặt các tài nguyên của Docker
* Để sử dụng database mà cụ thể là SQL SERVER (2019) bằng docker thì Microsoft đã có images của sql ở docker hub và ta sẽ thực hiện lệnh docker pull để lấy images sql đó để chạy ở server ở trang chủ của docker : https://hub.docker.com/\_/microsoft-mssql-server
* Ta sẽ thực hiện các lệnh như sau :
* sudo docker pull mcr.microsoft.com/mssql/server:phiên bản-latest thì ở đây máy chủ này đang sử dụng phiên bản 2019 thì sẽ thay phiên bản bằng 2019
* Tiếp theo ta sẽ chạy 1 lệnh nữa để chạy cái images đó thành 1 container trên docker
  + docker run -e “ACCEPT\_EULA =Y” -e “SA\_PASSWORD = MatKhau” – p 1433:1433 –name ten --hostname ten -d https://mcr.microsoft.com/mssql/server:2019-latest
* Để kiểm tra container có hoạt động không ta gõ lệnh docker ps để xem máy chủ có những container nào đang hoạt động



Hình 16 : Hình ảnh thể hiện máy chủ có các container đang hoạt động

CHƯƠNG 3 – CƠ SỞ LÝ THUYẾT / NGHIÊN CỨU THỰC NGHIỆM

*Những nghiên cứu thực nghiệm hoặc lý thuyết*: trình bày các cơ sở lý thuyết, lý luận, giả thuyết khoa học và phương pháp nghiên cứu sẽ được sử dụng trong Luận văn, Luận án;

3.1 Chèn bảng:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tiêu đề A** | **Tiêu đề B** |
| 1 | Nội dung 1 | Nội dung 4 |
| 2 | Nội dung 2 | Nội dung 5 |
| 3 | Nội dung 3 | Nội dung 6 |

Bảng 3.1Ví dụ cho chèn bảng

Khi cần chèn tên bảng thì chọn References \ Caption và chọn “Bảng …”

3.2 Viết tắt

Không lạm dụng việc viết tắt. Chỉ viết tắt những từ, cụm từ hoặc thuật ngữ *được sử dụng nhiều lần trong luận văn*. Không viết tắt những cụm từ dài, những mệnh đề hoặc những cụm từ ít xuất hiện. Nếu cần viết tắt những từ, thuật ngữ, tên các cơ quan, tổ chức... thì được viết tắt sau lần viết thứ nhất có kèm theo chữ viết tắt trong ngoặc đơn. Nếu có quá nhiều chữ viết tắt thì phải có bảng danh mục các chữ viết tắt (xếp theo thứ tự A, B, C) ở phần đầu luận văn.

3.3 Trích dẫn

3.3.1 Tài liệu tham khảo và cách trích dẫn

Mọi ý kiến, khái niệm, phân tích, phát biểu, diễn đạt... có ý nghĩa, mang tính chất gợi ý *không phải của riêng tác giả* và mọi tham khảo khác **phải được trích dẫn và chỉ rõ nguồn trong danh mục Tài liệu tham khảo của luận văn**. Phải nêu rõ cả việc sử dụng những đề xuất hoặc kết quả của đồng tác giả (*đối với công trình đã công bố khác thì phải trích dẫn bình thường như một tài liệu tham khảo*). Nếu sử dụng tài liệu của người khác và của đồng tác giả (bảng biểu, hình vẽ, công thức, đồ thị, phương trình, ý tưởng...) mà không chú dẫn tác giả và nguồn tài liệu thì **luận văn không được duyệt để bảo vệ**.

Không trích dẫn những kiến thức phổ biến, mọi người đều biết tránh làm nặng nề phần tham khảo trích dẫn.

Nếu người dẫn liệu không có điều kiện tiếp cận được một tài liệu gốc mà phải trích dẫn thông qua một tài liệu khác của một tác giả khác, thì phải nêu rõ cách trích dẫn (*lưu ý phải ghi* *đúng nguyên văn từ chính tài liệu tham khảo và hạn chế tối đa hình thức này).* Nếu cần trích dẫn dài hơn thì phải tách phần này thành một đoạn riêng khỏi phần nội dung đang trình bày, in nghiêng, với lề trái lùi vào thêm 2 cm. Khi mở đầu và kết thúc đoạn trích này không phải sử dụng dấu ngoặc kép. Việc trích dẫn là theo thứ tự của tài liệu ở danh mục Tài liệu tham khảo và được đặt trong ngoặc vuông, khi cần có cả số trang, ví dụ [15, tr.314-315]. Đối với phần trích dẫn từ nhiều tài liệu khác nhau, số của từng tài liệu được đặt độc lập trong từng ngoặc vuông, theo thứ tự tăng dần, ví dụ [19], [25], [41], [42].

3.3.2 Qui định của Khoa Công nghệ thông tin

**- Đạo văn** là việc sử dụng từ ngữ hay ý tưởng của người khác như là của mình trong hoạt động học thuật nói riêng và trong hoạt động sáng tạo nói chung. Tại Đại học Tôn Đức Thắng, những hành vi sau đây được xem là đạo văn:

* Sao chép nguyên văn **02** (hai) câu liên tiếp mà không dẫn nguồn đúng quy định;
* Sao chép nguyên văn **03** (ba) câu không liên tiếp mà không dẫn nguồn đúng quy định;
* Diễn đạt lại (rephrase) hoặc dịch (translate) toàn bộ một ý nào đó của người khác mà không dẫn nguồn đúng quy định;
* Sử dụng hơn 30% nội dung của một báo cáo cuối kỳ do chính mình viết để nộp cho 2 lớp khác nhau (cùng học kỳ hoặc khác học kỳ) mà không có sự đồng ý của giảng viên;
* Sao chép một phần hoặc toàn bộ bài làm của người khác.

- Khi luận văn, đồ án, bài tập lớn, được chấm điểm, nếu bị phát hiện đạo văn thì ngay lập tức bị điểm 0. Sinh viên sẽ tiếp tục bị xử lý kỷ luật theo các qui định của Nhà trường.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Quách Ngọc Ân (1992), “Nhìn lại hai năm phát triển lúa lai”, *Di tuyền học ứng dụng*, 98(1), tr. 10-16.
2. Bộ nông nghiệp & PTNT (1996), *Báo cáo tổng kết 5 năm (1992-1996) phát triển lúa lai,* Hà Nội.
3. Nguyễn Hữu Đống, Đào Thanh Bằng, Lâm Quang Dụ, Phan Đức Trực (1997), *Đột biến –* *Cơ sở lý luận và ứng dụng,* Nhà xuất bản nông nghiệp, Viện khoa học kỹ thuật nông nghiệp Việt Nam, Hà Nội.
4. Nguyễn Thị Gấm (1996), *Phát hiện và đánh giá một số dòng bất dục đực cảm ứng nhiệt* *độ,* Luận văn thạc sĩ khoa học nông nghiệp, Viện khoa học kỹ thuật nông nghiệp Việt Nam, Hà Nội.

……….

1. Võ Thị Kim Huệ (2000), *Nghiên cứu chẩn đoán và điều trị bệnh…,* Luận án Tiến sĩ y khoa, Trường đại học y Hà Nội, Hà Nội.

**Tiếng Anh**

1. Anderson J.E. (1985), The Relative Inefficiency of Quota, The Cheese Case, *American* *Economic Review*, 75(1), pp. 178-90.
2. Borkakati R. P.,Virmani S. S. (1997), Genetics of thermosensitive genic male sterility in Rice, *Euphytica* 88, pp. 1-7.
3. Boulding K.E. (1955), *Economics Analysis*, Hamish Hamilton, London.
4. Burton G. W. (1988), “Cytoplasmic male-sterility in pearl millet (penni-setum glaucum L.)”, *Agronomic Journal* 50, pp. 230-231.
5. Central Statistical Oraganisation (1995), *Statistical Year Book*, Beijing.
6. FAO (1971), *Agricultural Commodity Projections (1970-1980)*, Vol. II. Rome.
7. Institute of Economics (1988), *Analysis of Expenditure Pattern of Urban Households in* *Vietnam,* Departement pf Economics, Economic Research Report, Hanoi.

**PHỤ LỤC**

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn