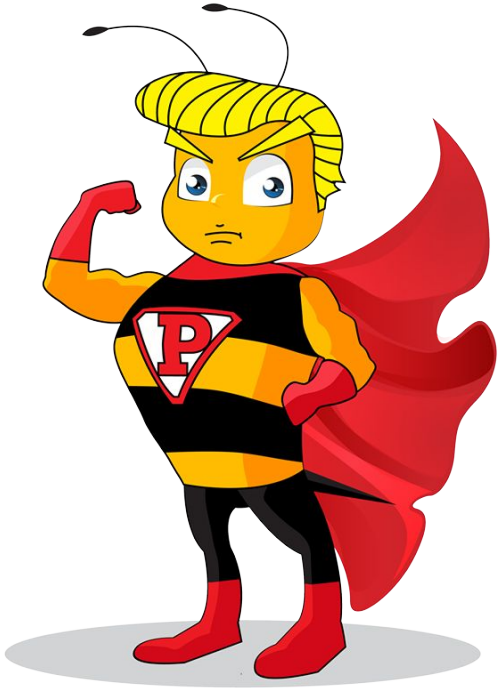
FPT Education

FPT POLYTECHNIC

SOA204

PHẦN MỀM MIỄN PHÍ VÀ MÃ NGUỒN MỞ

OPEN SOURCE & FREE SOFTWARE



BÀI 4:
Ảo hóa trong phát triển phần mềm với Docker

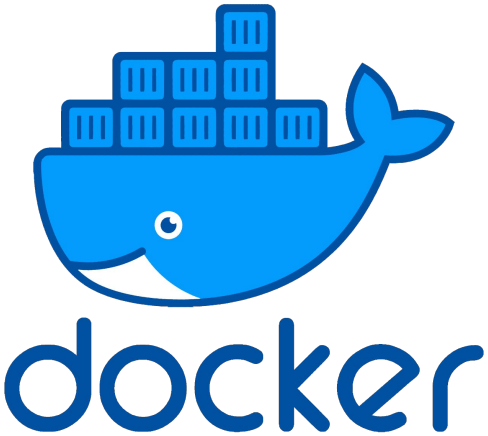
https://caodang.fpt.edu.vn

NỘI DUNG BÀI TRƯỚC

- Giới thiệu hệ điều hành Ubuntu
- Hướng dẫn cài đặt Ubuntu
- Hướng dẫn cài đặt phần mềm trên Ubuntu
- Giới thiệu về filesystem, bash, shell

NỘI DUNG BÀI HỌC

- Ảo hóa là gì?
- Giới thiệu về Docker
- Hướng dẫn cài đặt Docker
- Hướng dẫn sử dụng Docker cơ bản



ẢO HÓA TRONG PHÁT TRIỂN PHẦN MỀM

Ảo hóa *(virtualization hoặc containerization)* là công nghệ mà bạn có thể sử dụng để *tạo các dạng trình bày ảo của máy chủ, kho lưu trữ, mạng và nhiều máy vật lý khác*. Phần mềm ảo mô phỏng các chức năng của phần cứng vật lý để chạy đồng thời nhiều máy ảo trên một máy vật lý duy nhất.

virtualization

Các doanh nghiệp ứng dụng công nghệ ảo hóa để sử dụng hiệu quả tài nguyên phần cứng của họ và thu về lợi nhuận trên vốn đầu tư lớn hơn. Công nghệ này cũng hỗ trợ nhiều dịch vụ điện toán đám mây giúp các tổ chức quản lý cơ sở hạ tầng hiệu quả hơn.

VÌ SAO ẢO HÓA LẠI QUAN TRỌNG?

- Bằng cách sử dụng ảo hóa, bạn có thể tương tác với bất kỳ tài nguyên phần cứng nào với độ linh hoạt cao hơn.
- Ảo hóa sẽ loại bỏ tất cả những giới hạn vật lý giữa bạn và máy chủ bằng cách trừu tượng hóa chức năng của phần cứng vật lý thành phần mềm.
- Bạn có thể quản lý, bảo trì và sử dụng cơ sở hạ tầng phần cứng của mình như một ứng dụng trên web.

VIRTUALIZATION:

- Virtualization là mức ảo hóa phần cứng trong đó một phần cứng vật lý được phân chia thành nhiều phần cứng ảo để có thể chạy đa hệ điều hành và các ứng dụng khác nhau đồng thời.
- Nó bao gồm ảo hóa CPU, RAM, ổ đĩa, card mạng… Bằng cách ảo hóa phần cứng này cho phép nhiều máy ảo chạy cùng trên một máy chủ vật lý mà không ảnh hưởng đến nhau.
- Virtualization cần sử dụng tài nguyên phần cứng khá lớn và có thể gây ra sự trễ trong quá trình hoạt động.

Ví dụ về các công cụ sử dụng Virtualization:

➢ VMWare, VirtualBox, KVM, Hyper-V…

CONTAINERIZATION:

- Containerization là mức ảo hóa hệ điều hành, trong đó mỗi container chứa một ứng dụng hoặc một nhóm ứng dụng đóng gói cùng với các thư viện và tài nguyên cần thiết để chạy ứng dụng đó.
- Các container chia sẻ nhân (kernel) của hệ điều hành vật lý và các thư viện chung với hệ điều hành, do đó Containerization tiết kiệm tài nguyên hơn so với Virtualization.

podman

Ví dụ về các công cụ sử dụng Containerization:

➢ Docker, Podman, Kubernetes…

SO SÁNH VIRTUALIZATION và CONTAINERIZATION:

Mục	Virtualization	Containerization
Isolation *(tính cô lập)*	Cung cấp sự cách ly hoàn toàn với hệ điều hành máy chủ và các máy ảo khác	Thường cung cấp khả năng cách ly nhẹ với máy chủ và các vùng chứa khác, nhưng không cung cấp ranh giới bảo mật mạnh mẽ như VM
Operating System *(hệ điều hành)*	Chạy một hệ điều hành hoàn chỉnh bao gồm kernel, do đó yêu cầu nhiều tài nguyên hệ thống hơn như CPU, bộ nhớ và bộ lưu trữ	Chạy phần chế độ người dùng của hệ điều hành và có thể được điều chỉnh để chỉ chứa các dịch vụ cần thiết cho ứng dụng của bạn bằng cách sử dụng ít tài nguyên hệ thống hơn
Guest compatibility (khả năng tương thích)	Chạy gần như mọi hệ điều hành bên trong máy ảo	Chạy trên cùng phiên bản hệ điều hành với máy chủ

SO SÁNH VIRTUALIZATION và CONTAINERIZATION:

Mục	Virtualization	Containerization
Deployment *(triển khai)*	Triển khai các máy ảo riêng lẻ bằng phần mềm Hypervisor	Triển khai các vùng chứa riêng lẻ bằng cách sử dụng Docker hoặc triển khai nhiều vùng chứa bằng cách sử dụng bộ điều phối như Kubernetes
Persistent storage *(lưu trữ liên tục)*	Sử dụng Đĩa cứng ảo (VHD) để lưu trữ cục bộ cho một VM hoặc chia sẻ tệp Khối tin nhắn máy chủ (SMB) để lưu trữ được chia sẻ bởi nhiều máy chủ	Sử dụng đĩa cục bộ để lưu trữ cục bộ cho một nút hoặc SMB để lưu trữ được chia sẻ bởi nhiều nút hoặc máy chủ
Load balancing (cân bằng tải)	Cân bằng tải máy ảo được thực hiện bằng cách chạy VM trên các máy chủ khác trong cụm chuyển đổi dự phòng	Người điều phối có thể tự động khởi động hoặc dừng các vùng chứa trên các nút cụm để quản lý các thay đổi về tải và tính khả dụng.

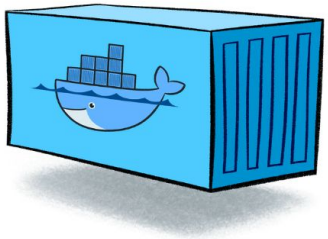
SO SÁNH VIRTUALIZATION và CONTAINERIZATION:

Mục	Virtualization	Containerization
Networking ***(kết nối mạng)***	Sử dụng bộ điều hợp mạng ảo	Sử dụng chế độ xem biệt lập của bộ điều hợp mạng ảo. Do đó, cung cấp ít ảo hóa hơn một chút

TRƯỚC KHI CÓ DOCKER:

- Trước khi Docker xuất hiện hãy tưởng tượng vấn đề của một nhóm lập trình viên có thể sẽ gặp phải như sau:
  - Một nhóm gồm nhiều thành viên, trong đó có người sử dụng MacOS, có người sử dụng Windows và những người khác sử dụng những bản Distro khác nhau của Linux như Ubuntu, Fedora.
  - Sẽ xảy ra trường hợp code chạy ổn trên máy người này nhưng gặp vấn đề trên máy của người khác.
  - Mỗi người sử dụng một phiên bản các thư viện khác nhau…
  - Khi triển khai dự án lên server thì môi trường của máy chủ khác với môi trường ở local.
  - …

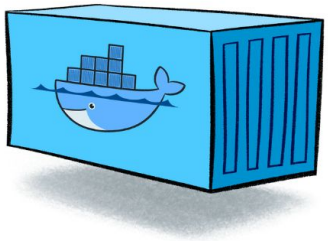
Docker là một công cụ đóng gói phần mềm và triển khai ứng dụng trong các **container**. Các ứng dụng chạy trong các **container** được gọi là các **container Docker**. **Docker** giúp các lập trình viên chỉ cần chạy các **container** lên mà không cần phải lo về việc cài các thư viện hoặc môi trường, điều này giúp các lập trình viên phát triển ứng dụng 1 cách nhanh chóng, đảm bảo tính nhất quán giữa môi trường phát triển và môi trường triển khai.



Docker là một dự án mã nguồn mở được phát hành lần đầu 2013. **Docker** được viết bằng Go. **Docker** có thể chạy trên nhiều hệ điều hành khác nhau như Windows, MacOS, Linux

THÀNH PHẦN:

- **Docker deamon:** *Là một tiến trình nền chạy trên một máy tính Docker host (máy cài Docker), và quản lý các hoạt động Docker như tạo và quản lý các container, images, networks và volumes.*
- **Docker client:** *Là một ứng dụng dòng lệnh hoặc giao diện người dùng đồ họa (GUI) để tương tác với Docker daemon và thực hiện các hoạt động Docker. Docker client sử dụng Docker API để giao tiếp với Docker daemon.*
- **Docker images:** *Là một gói đóng gói của một ứng dụng và các tài nguyên cần thiết để chạy ứng dụng đó trong một container. Một image có thể được tạo từ một Dockerfile hoặc tải từ một kho chứa image trên internet như Docker Hub.*

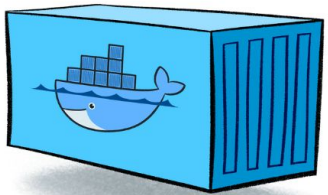


CÁC THÀNH PHẦN CƠ BẢN CỦA DOCKER

FPT Education
FPT POLYTECHNIC

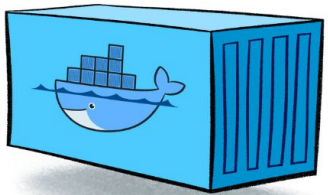
THÀNH PHẦN:

- **Docker container:** *Là một môi trường đóng gói độc lập, chứa tất cả các thành phần cần thiết để chạy một ứng dụng trong một môi trường cô lập, **container** được tạo ra từ **image** sau khi đã đóng gói.*
- **Docker network:** *cho phép các **container** tương tác với nhau và với các dịch vụ khác. Một **Docker network** được tạo ra để tạo một mạng ảo cho các **container** chạy trên cùng một máy **Docker host**.*
- **Docker volume:** *Cho phép các **container** lưu trữ và truy cập dữ liệu được sử dụng bởi các ứng dụng. **Volume** giúp dữ liệu được bảo vệ và giữ cho đồng bộ giữa các **container**.*



THÀNH PHẦN:

- **Docker registry:** *là một kho chứa để lưu trữ các image được tạo bởi người dùng Docker hoặc bởi các nhà cung cấp phần mềm. Docker Hub là một ví dụ về Docker registry công cộng, trong khi các tổ chức cũng có thể triển khai một Docker registry riêng tư. Đây là nơi mà chúng ta tải lên các image sau khi đã build (khá giống việc lưu trữ code trên Github).*
- **Dockerfile:** *Là một tệp cấu hình định nghĩa các bước để xây dựng một image Docker. Dockerfile chứa các chỉ thị để tải các phần mềm, công cụ cần thiết, cài đặt các gói phụ thuộc, cấu hình ứng dụng...*
- **Docker compose:** *Là công cụ được sử dụng để quản lý, triển khai và tự động hóa việc chạy nhiều container cùng một lúc.*



CÁC LỆNH CƠ BẢN CỦA DOCKER

FPT Education
FPT POLYTECHNIC

COMMAND LINE:

- **docker run** : *Lệnh để chạy một container từ một image*
- **docker pull** : *Lệnh để tải một image từ Docker Registry về máy tính.*
- **docker push** : *Lệnh để đẩy một image lên Docker Registry.*
- **docker build** : *Lệnh để tạo một image từ Dockerfile.*
- **docker images** : *Lệnh để hiển thị danh sách các image có sẵn trong Docker.*
- **docker ps** : *Lệnh để hiển thị danh sách các container đang chạy.*
- **docker stop** : *Lệnh để dừng một container đang chạy.*
- **docker rm** : *Lệnh để xóa một container đã dừng hoặc bị lỗi.*
- **docker rmi** : *Lệnh để xóa một image.*
- **docker run** : *Lệnh để thực thi một lệnh trong một container đang chạy.*

FPT Education
FPT POLYTECHNIC

CÀI ĐẶT DOCKER TRÊN UBUNTU

YÊU CẦU CẤU HÌNH:

- kernel và CPU 64-bit có hỗ trợ ảo hóa.
- Hỗ trợ ảo hóa KVM.
- QEMU phải là phiên bản 5.2 trở lên.
- Môi trường máy tính để bàn Gnome, KDE hoặc MATE.
- Ít nhất 4 GB RAM.

B1.1:

- **Thiết lập Docker’s apt repository**

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl gnupg
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

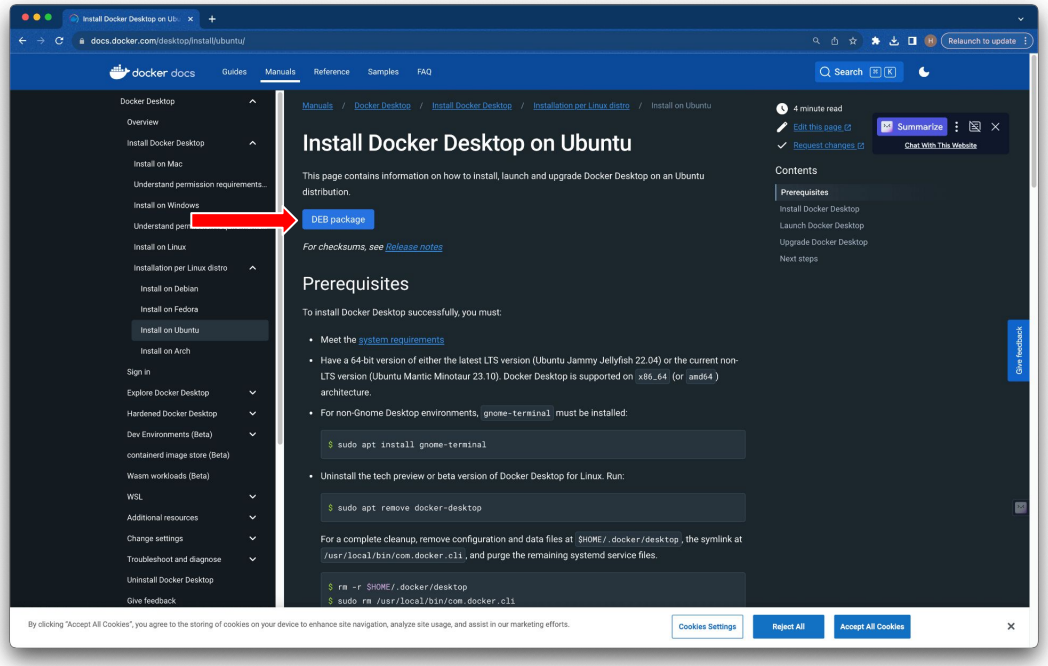
B1.2:

- **Thiết lập Docker’s apt repository**

```
# Add the repository to Apt sources:
echo \
  "deb [arch="$(dpkg --print-architecture)"
signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu
\
  "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

B2:

- Tải bản cài đặt .DEB: https://docs.docker.com/desktop/install/ubuntu/



B3:

- Cài đặt gói vừa tải với apt như sau:

```
$ sudo apt-get update
$ sudo apt-get install ./docker-desktop-<version>-<arch>.deb
```

KHỞI CHẠY DOCKER DESKTOP:

- Tìm icon của Docker Desktop trong Applications và mở lên.
- hoặc Khởi động Docker Desktop từ Terminal:

```
$ systemctl --user start docker-desktop
```

KIỂM TRA VERSION:

- Sau khi cài đặt thành công có thể dễ dàng kiểm tra phiên bản Docker bằng cách:

```
$ docker compose version
Docker Compose version v2.17.3

$ docker --version
Docker version 23.0.5, build bc4487a

$ docker version
Client: Docker Engine - Community
 Cloud integration: v1.0.31
 Version:           23.0.5
 API version:       1.42
<...>
```

FPT Education
FPT POLYTECHNIC

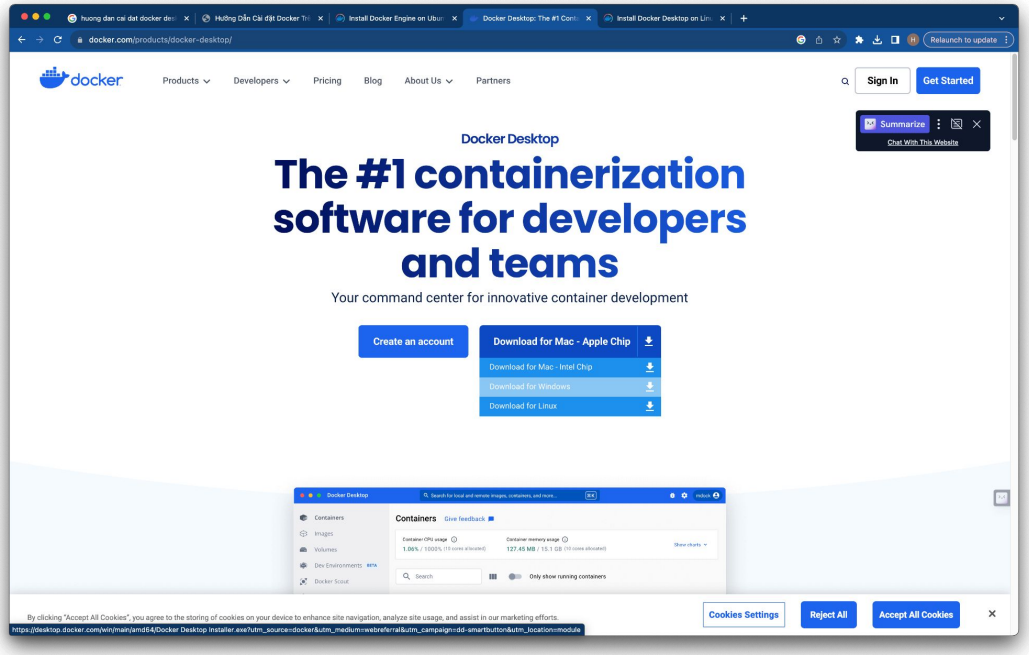
CÀI ĐẶT DOCKER TRÊN WINDOWS

CÀI ĐẶT DOCKER TRÊN WINDOWS

FPT Education
FPT POLYTECHNIC

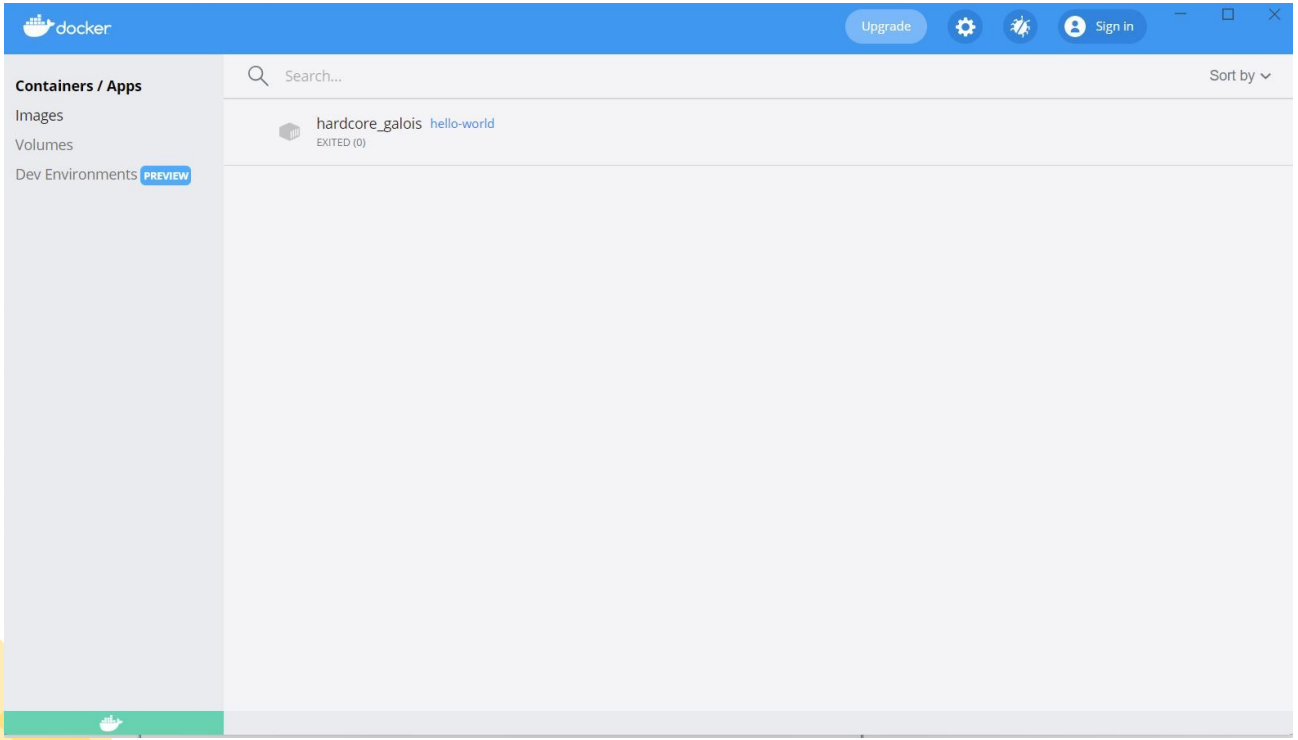
TẢI và CÀI ĐẶT:

- Truy cập: https://www.docker.com/products/docker-desktop/ và tải bản cài đặt dành cho hệ điều hành Windows.



CÀI ĐẶT:

- Chạy file cài đặt bằng quyền Administrator và làm theo hướng dẫn.



DEMO

- Giảng viên demo cài đặt Docker trên Ubuntu hoặc trên Windows.

TỔNG KẾT

- **Ảo hóa** (virtualization hoặc containerization) là công nghệ mà bạn có thể sử dụng để tạo các dạng trình bày ảo của máy chủ, kho lưu trữ, mạng và nhiều máy vật lý khác.
- **Virtualization** là mức ảo hóa phần cứng trong đó một phần cứng vật lý được phân chia thành nhiều phần cứng ảo để có thể chạy đa hệ điều hành và các ứng dụng khác nhau đồng thời.
- **Containerization** là mức ảo hóa hệ điều hành, trong đó mỗi container chứa một ứng dụng hoặc một nhóm ứng dụng đóng gói cùng với các thư viện và tài nguyên cần thiết để chạy ứng dụng đó.
- **Docker** là công cụ ảo hóa mức hệ điều hành.
- **Docker** rất phổ biến hiện nay trong phát triển phần mềm.

- Cài đặt và cấu hình Webserver trên hệ điều hành Linux

FPT Education
FPT POLYTECHNIC

THANK YOU!