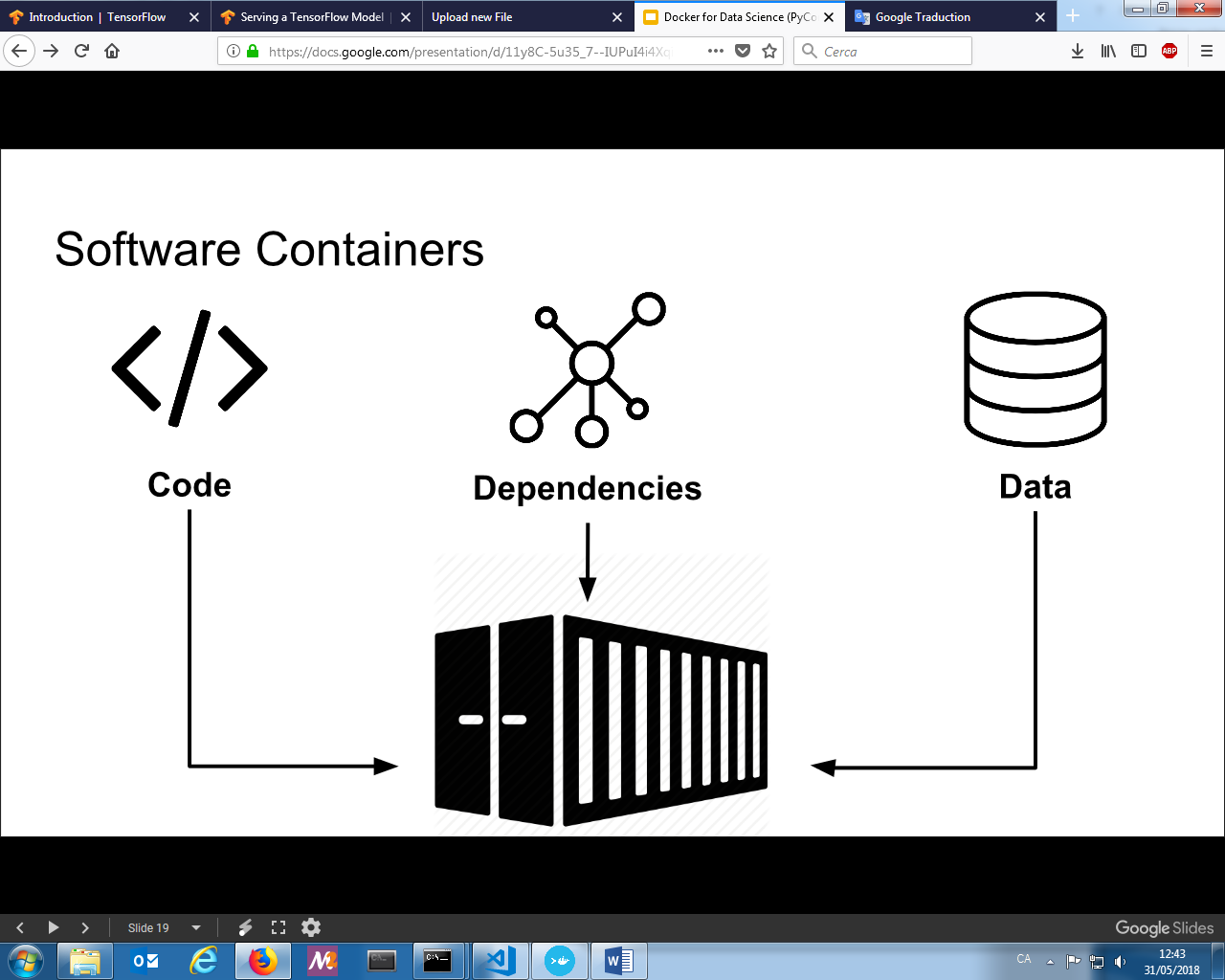
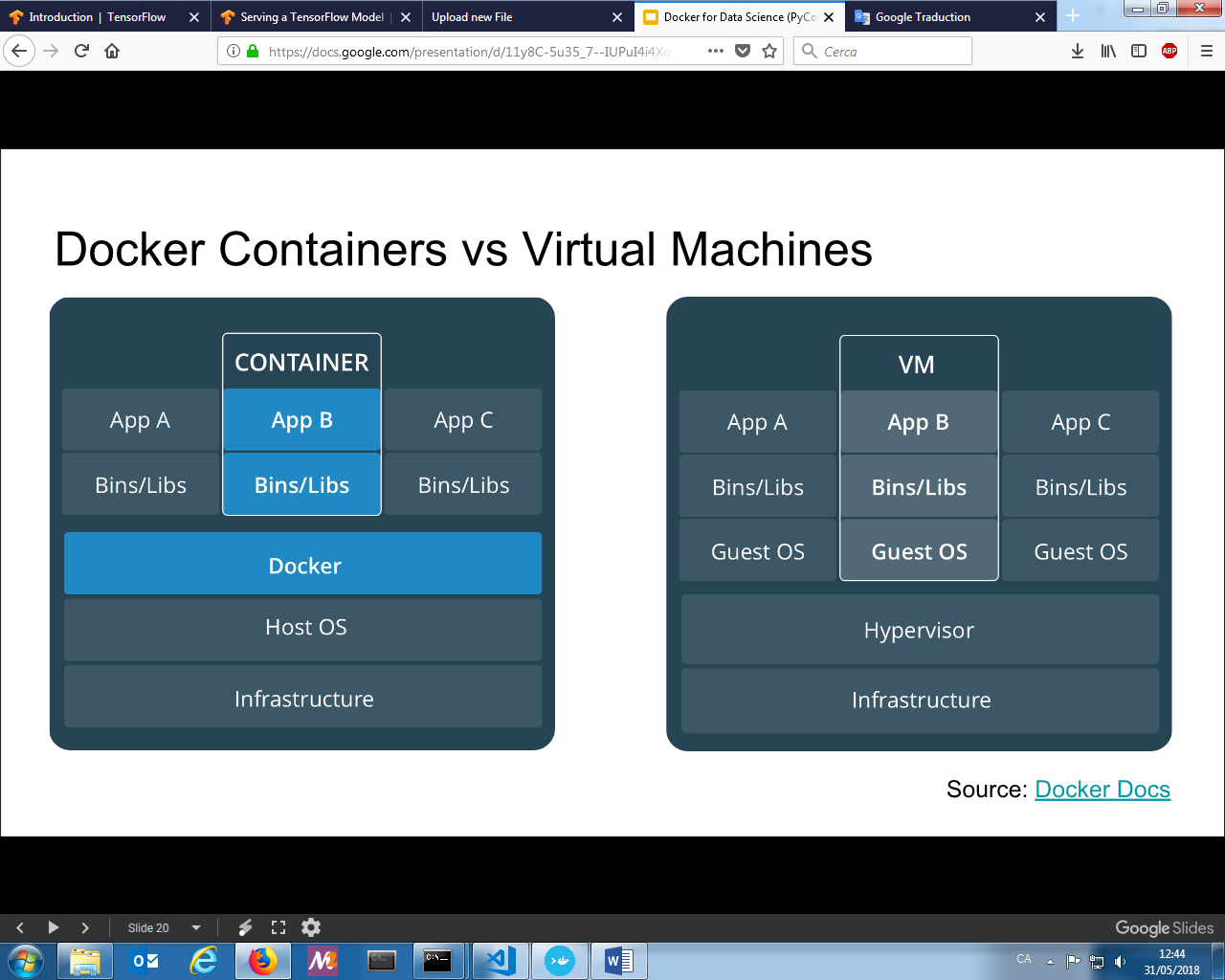
1. What is Docker

Docker is a manager of container.

Container:

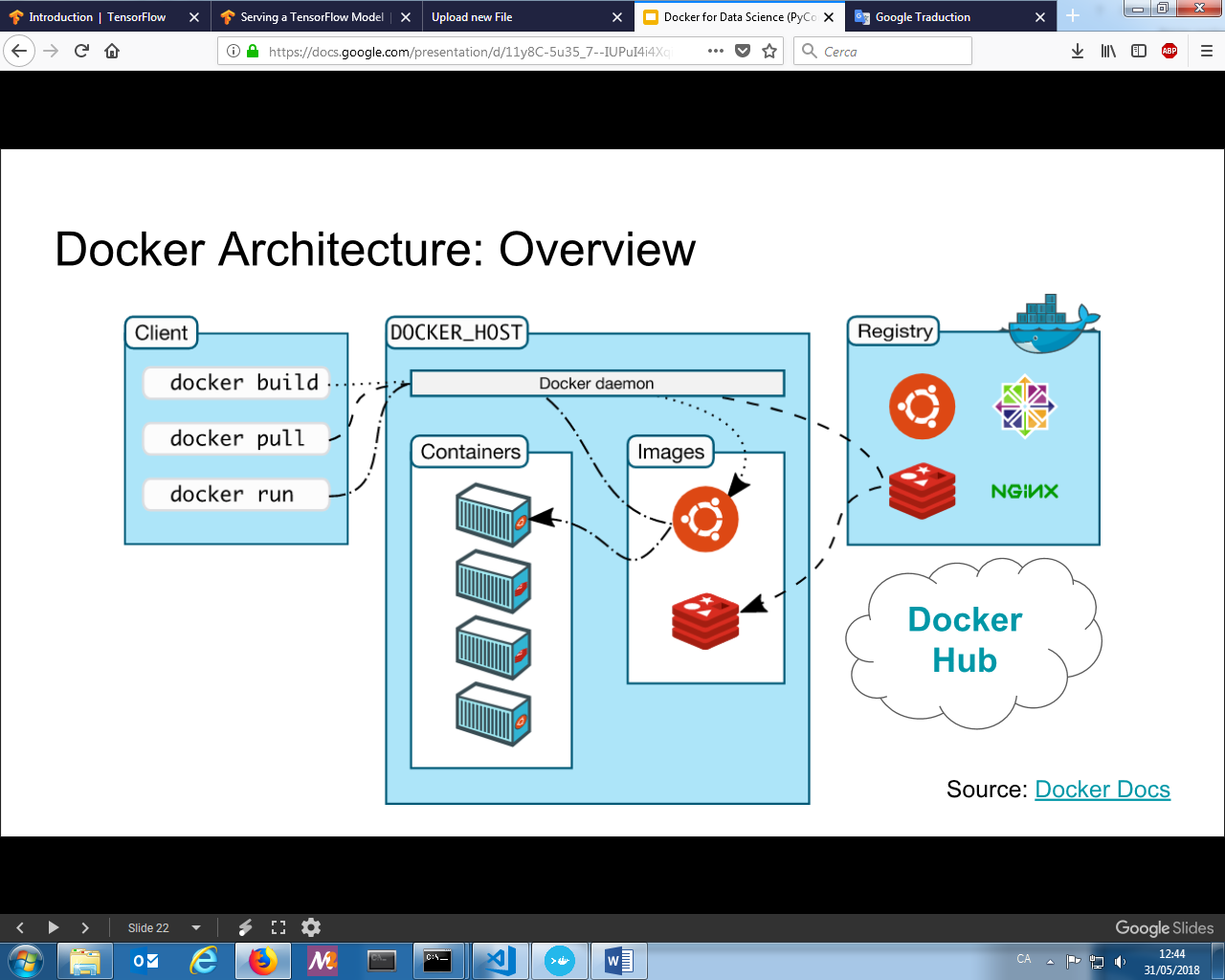


* Isolates app from each other
* Shares the same kernel – kernel shares resources with the host and interacts with the containers



Advantages

* Quicker to launch (no os to boot)
* Portability (less dependencies between processed layers)
* Efficiency

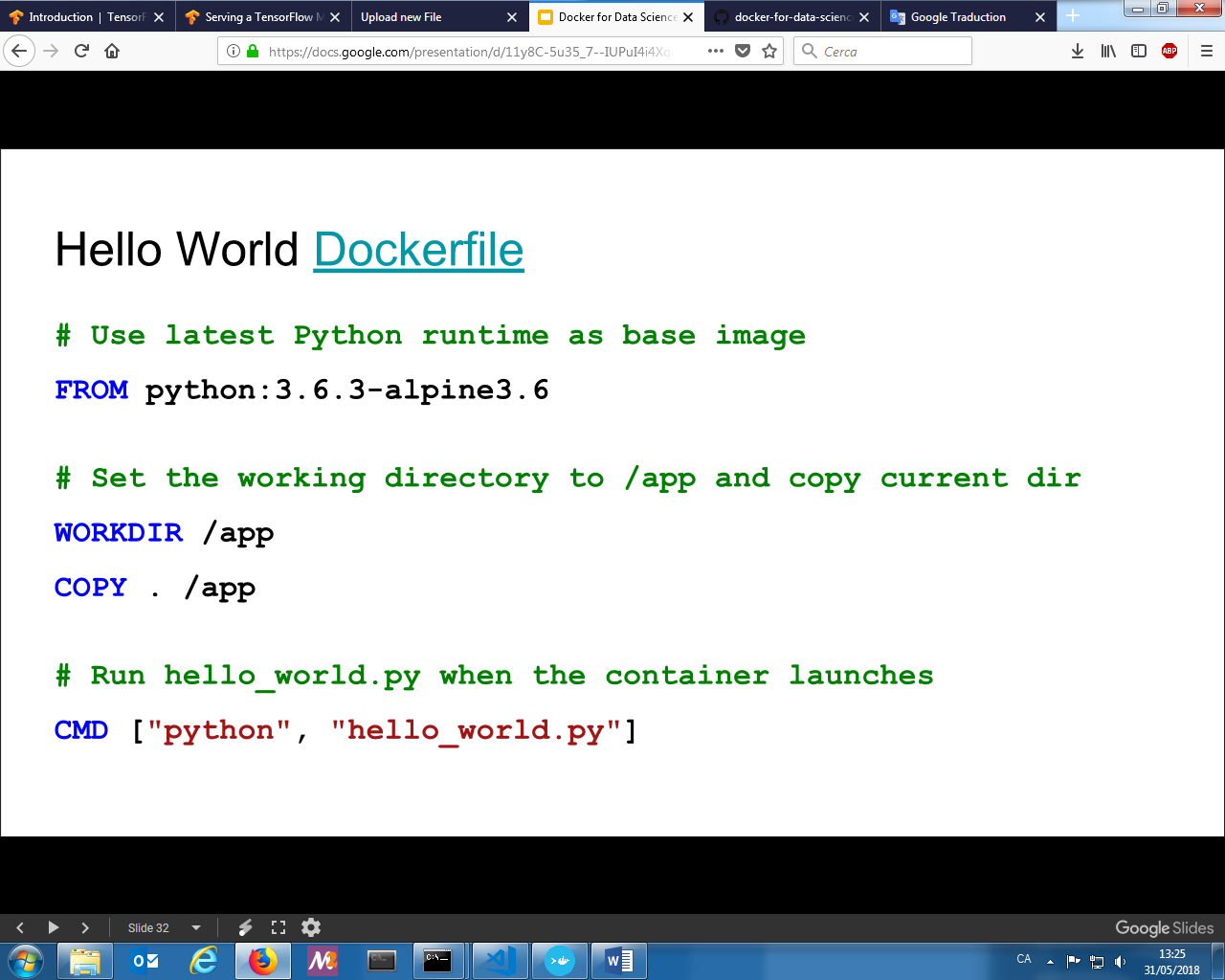


1. Run a container
2. Pull an image from Docker hub: docker login then docker pull <image>
3. Check the image: docker images
4. Create a container: docker run –it <nom>
5. Stop the container: docker stop <id container>
6. Interact with a container with shell: docker run –it <nom container> /bin/bash
7. Stop the container
8. Confirm the container has been stop: docker ps –a
9. Delete the container: docker rm <id container>
10. Build your own container
11. Create a directory

Commands:

|  |  |
| --- | --- |
| FROM | Set base image |
| LABEL | Add metadata |
| COPY | Copy files/directories into the image |
| ENV | Set environment variable |
| WORKDIR | Set working directory |
| RUN | Execute shell commands in a new layer |
| ENTRYPOINT | Configure container to run as executable |
| CMD | Provide default for executing container |

Example



1. Add dependencies by creating reauirements.txt
2. Build your image: docker build –t <nom> .
3. Create the container: docker run <nom>
4. Upload to docker hub: docker login then docker push <image>
5. Exercise
   1. Create a directory
   2. Create your Dockerfile in this new directory.

FROM ubuntu:latest

RUN apt-get update && apt-get install -y python3 \

python3-pip

COPY requirements.txt .

RUN pip3 install jupyter

RUN pip3 install -r requirements.txt

RUN useradd -ms /bin/bash jupyter

USER jupyter

WORKDIR /home/jupyter

COPY success.txt .

ENTRYPOINT ["jupyter", "notebook", "--ip=\*"]

* 1. Create requirements.txt with Tensorflow, Keras.
  2. Create success.txt
  3. Build your image: docker build -t jupyter .
  4. Get your image id: docker images
  5. Create a new container: docker run -it -p 8888:8888 <image\_id> /home