Introduction to Docker and Containers

CS 4230
Jay Urbain, Ph.D.

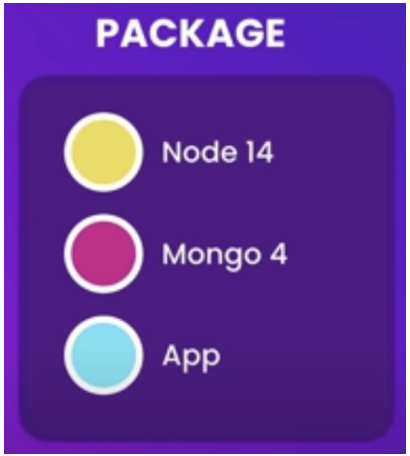

YOU GET A CLUSTER AND YOU GET A CLUSTER
EVERYBODY GETS A CLUSTER
imgflip.com

Problems with application deployment

- One or more files missing
- Software version mismatch on target machine.
- Different configuration settings like environment variables.

Docker

- A platform for building, running and deploying applications in a consistent manner.
- Create a package (container) of your application environment.
- If the package runs on your development machine, it should work on any target machine.



Docker - additional benefits

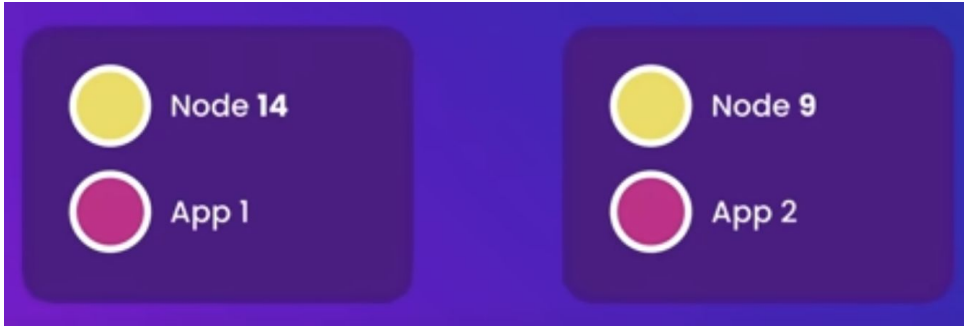
- New developer don’t have to spend a lot of time configuring a development environment.

```
$ docker -compose up
```

Docker will download and run an application environment in an isolated environment called a **container**.

Docker - additional benefits

- Isolated environment allows multiple applications to user different version of the same software side by side.
- Also, when you’re done with a specific version you can easily remove it in one shot.
  - You don’t accumulate multiple versions of libraries, tools and environments on your development system.
  - You don’t have to worry if it’s safe to remove different development artifacts.
  - `$ docker -compose down -rmi all`



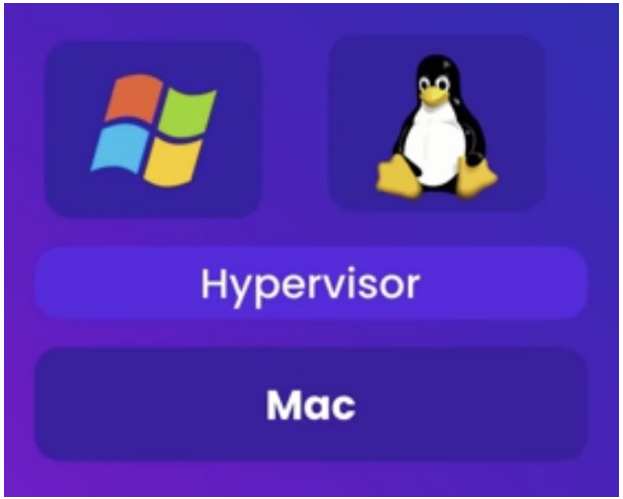
Containers versus Virtual Machines

Advantages/disadvantages of each approach?

CONTAINER	VIRTUAL MACHINE
An isolated environment for running an application	An abstraction of a machine (physical hardware)

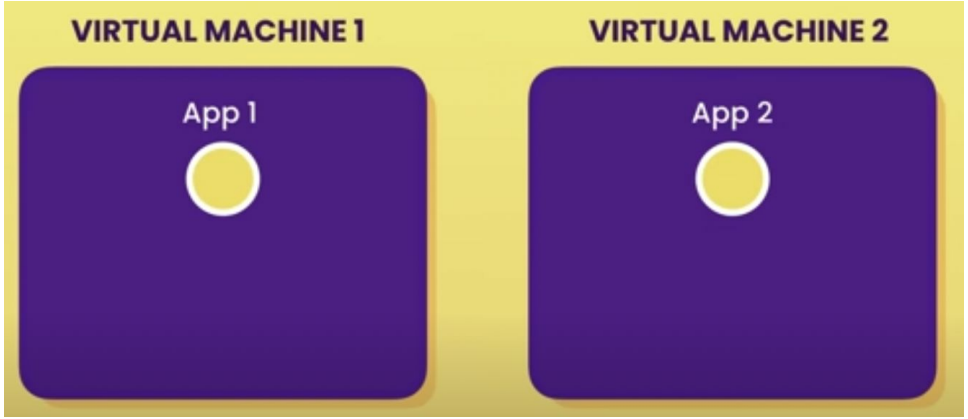
Virtual Machines

- Hypervisor - software to create and manage virtual machines.
- VirtualBox, VMware, Hyper-V (Windows)



Virtual Machines

- Can run an application in near complete isolation inside a virtual machine on the same physical hardware.
- Needs a full copy of OS that may need to be licensed and patched.
- More resource intensive because each virtual machine takes a slice of physical hardware and must load the entire OS.



Virtual Machines

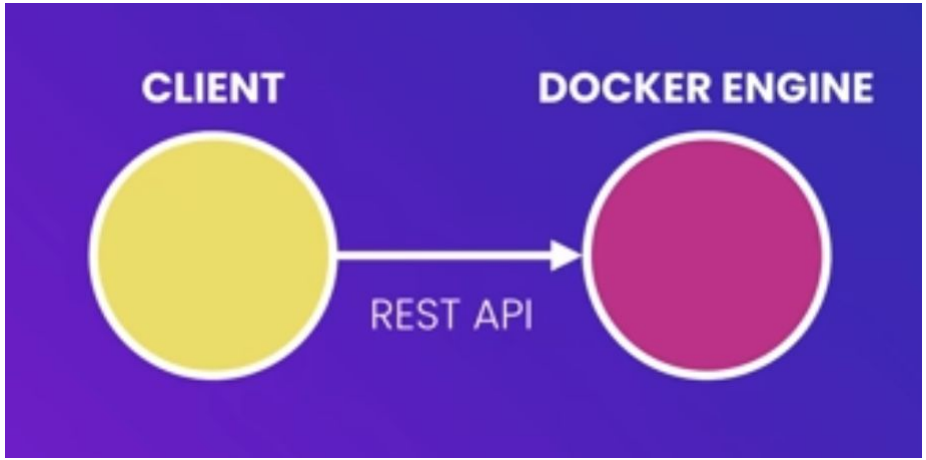
- Can run an application in isolation (exact dependencies) inside a virtual machine on the same physical hardware.
- Needs a full copy of OS that may need to be licensed and patched.
- Slow to start because entire OS had to be loaded.
- More resource intensive because each virtual machine takes a slice of physical hardware.
- Memory has to be divided among virtual machines.

Containers

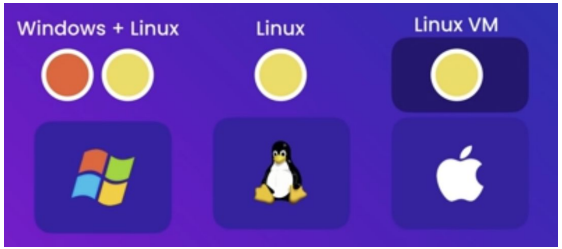
- Allow running multiple apps in isolation.
- More lightweight.
- Uses the OS of the host.
- Starts quickly - doesn’t have to load OS.
- Need less hardware resources - don’t need to allocate specific number of CPU cores or a slice of memory like VM.

Docker

- Client-server architecture
- Docker engine runs in the background and takes care of building and running docker containers.
- Technically, a docker container is just a process like other processes running on your computer.



Docker

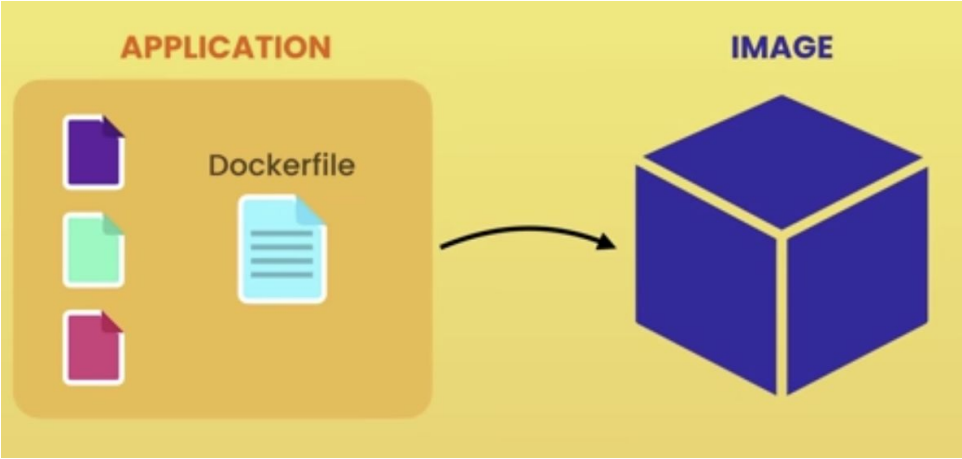


- Containers share the same operating system.
- Specifically the same OS kernel that manages hardware resources like memory, cpu, and scheduling.
- Different kernels have different APIs, that’s why you can’t run a Windows application on Linux.
- So on a Linux host, you can only run Linux containers.
- Can run both windows and Linux containers on Windows 10 since Windows now has a custom built linux kernel

Docker Workflow

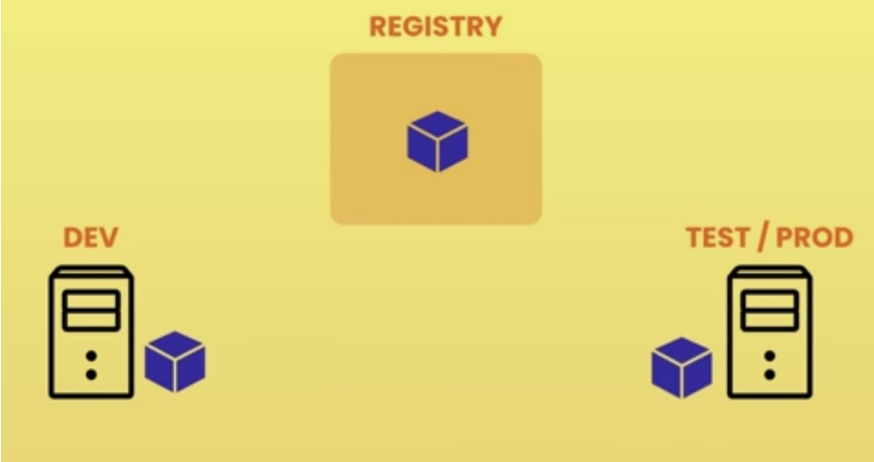
Dockerize your application.

- Dockerfile contains instructions for installing and running your application.
- Minified OS
- Runtime environment, e.g., Node or Python
- Application files
- Third party libraries/packages
- Environment variables



Docker Workflow

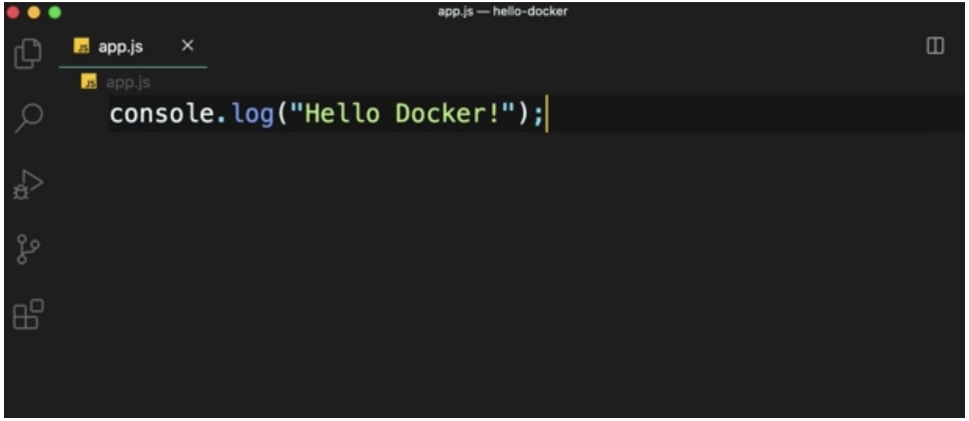
- Special process since it contains its own file system which is provided by the image.
- Test/Prod has the same environment as our development machine.



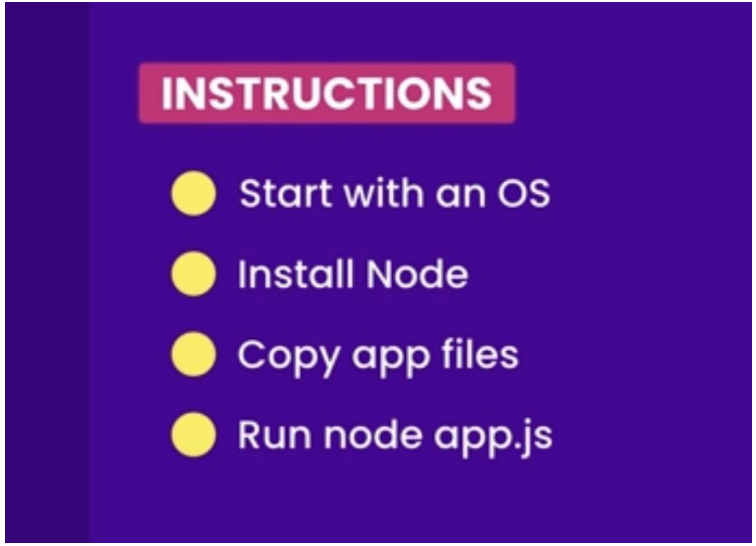
Docker Workflow - create an application directory

```
~/Desktop
✔ mkdir hello-docker

~/Desktop
✔ cd hello-docker

~/Desktop/hello-docker
✔ code .

~/Desktop/hello-docker
✔
```

Docker Workflow - write an application



Docker Workflow - create a Docker image



Docker Workflow

Write instructions for creating and running image within within Dockerfile

Dockerfile ●

Dockerfile > WORKDIR

```
FROM node:alpine
COPY . /app
WORKDIR /app
CMD node app.js
```

Docker Workflow

Build the image

```
~/Desktop/hello-docker
✔  docker build -t hello-docker .
+] Building 1.2s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 99B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:alpine
=> [internal] load build context
=> => transferring context: 158B
=> CACHED [1/3] FROM docker.io/library/node:alpine@sha256:c01b5723
=> [2/3] COPY . /app
=> [3/3] WORKDIR /app
=> exporting to image
=> => exporting layers
```

Docker Workflow

List and run image

```
~/Desktop/hello-docker
> ✔ docker image ls
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
hello-docker    latest    8872d3105639   2 minutes ago    112MB

~/Desktop/hello-docker
> ✔ docker run hello-docker
Hello Docker!
```

Docker Workflow

```
$docker pull jayurbain/deepnlpintro
```

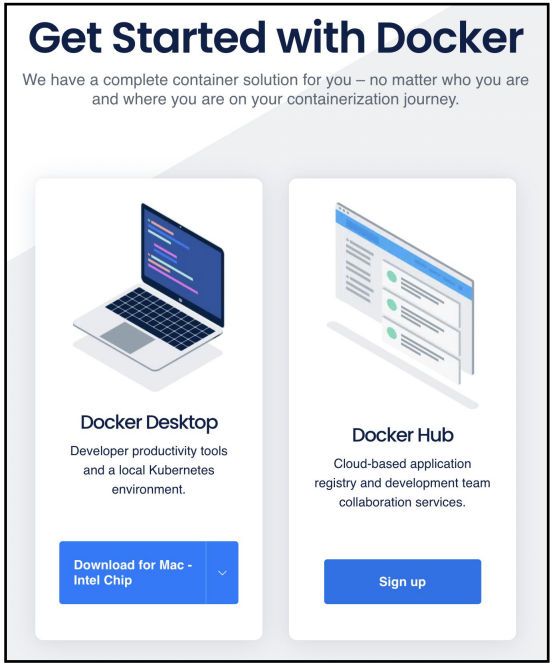
Docker Installation

Check your version of Docker. You want the latest version.

% docker --version
Docker version 20.10.23, build 7155243

Download

https://www.docker.com/get-started/



Docker Installation

Docker Desktop

Docker Desktop is an application for MacOS and Windows machines for the building and sharing of containerized applications.

Docker Hub

Docker Hub is like github for containers.

Docker Installation

Windows users: Make sure you enable Hyper-V on windows machine:

System Requirements

- Windows 10 64-bit: Pro, Enterprise, or Education (Build 17134 or later).

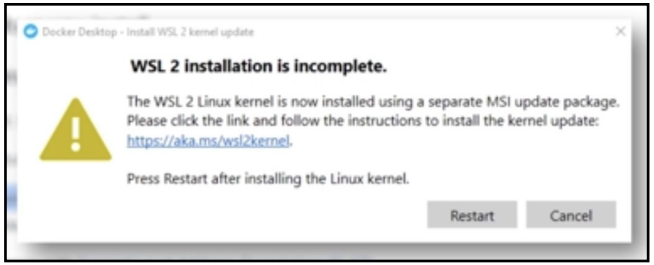
  For Windows 10 Home, see Install Docker Desktop on Windows Home.

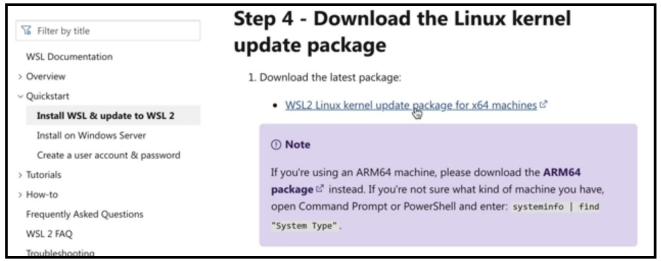
- Hyper-V and Containers Windows features must be enabled.
- The following hardware prerequisites are required to successfully run Client Hyper-V on Windows 10:
  - 64 bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - BIOS-level hardware virtualization support must be enabled in the BIOS settings. For more information, see Virtualization.

Docker Installation

Windows users:

If the following error occurs, you have to update your Windows Linux kernel that is shipped with your version of Windows.



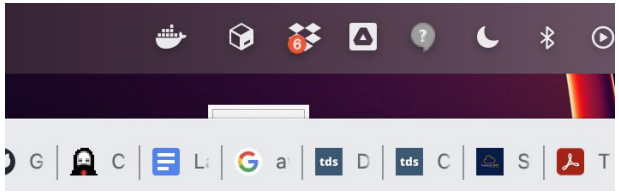


Run Docker

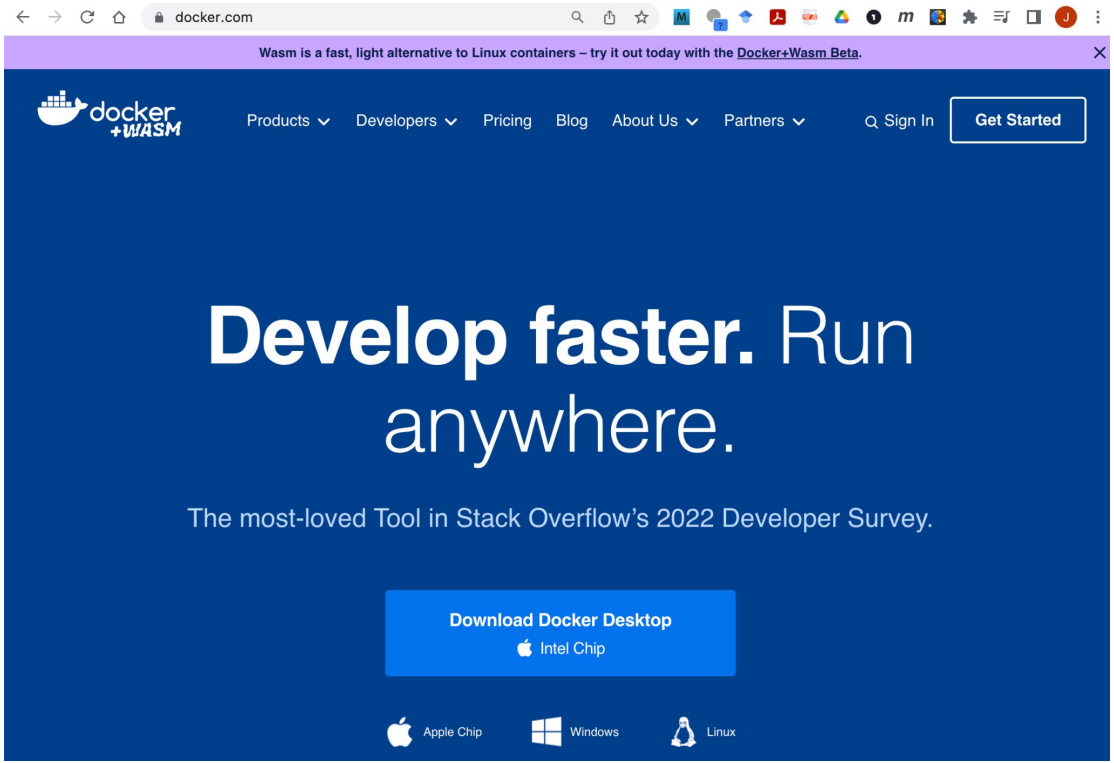
Make sure Docker is running. You should see the following:

```
% docker version
Client:
 Cloud integration: v1.0.31
 Version:           20.10.23
 API version:       1.41
 Go version:        go1.18.10
 Git commit:        7155243
 Built:             Thu Jan 19 17:35:19 2023
 OS/Arch:           darwin/arm64
 …
```

You should also see the whale holding containers in your status bar. This shows the Docker Engine (server is running).



Complete the Get Started Tutorial



Docker pull

Bonus: Download and log into a Docker container from Docker Hub.