

---

# Deploying Apps at Scale

An introduction to Docker and Kubernetes.

Daniel N. Gisolfi

Daniel.Gisolfi1@Marist.edu

Marist Computer Society

# Problem Statement

---

I have a ReactJS app and I want to deploy to a server.

How?

# SystemD - Deploying Apps Prior to a containerized world

---

What do you use to deploy an app?

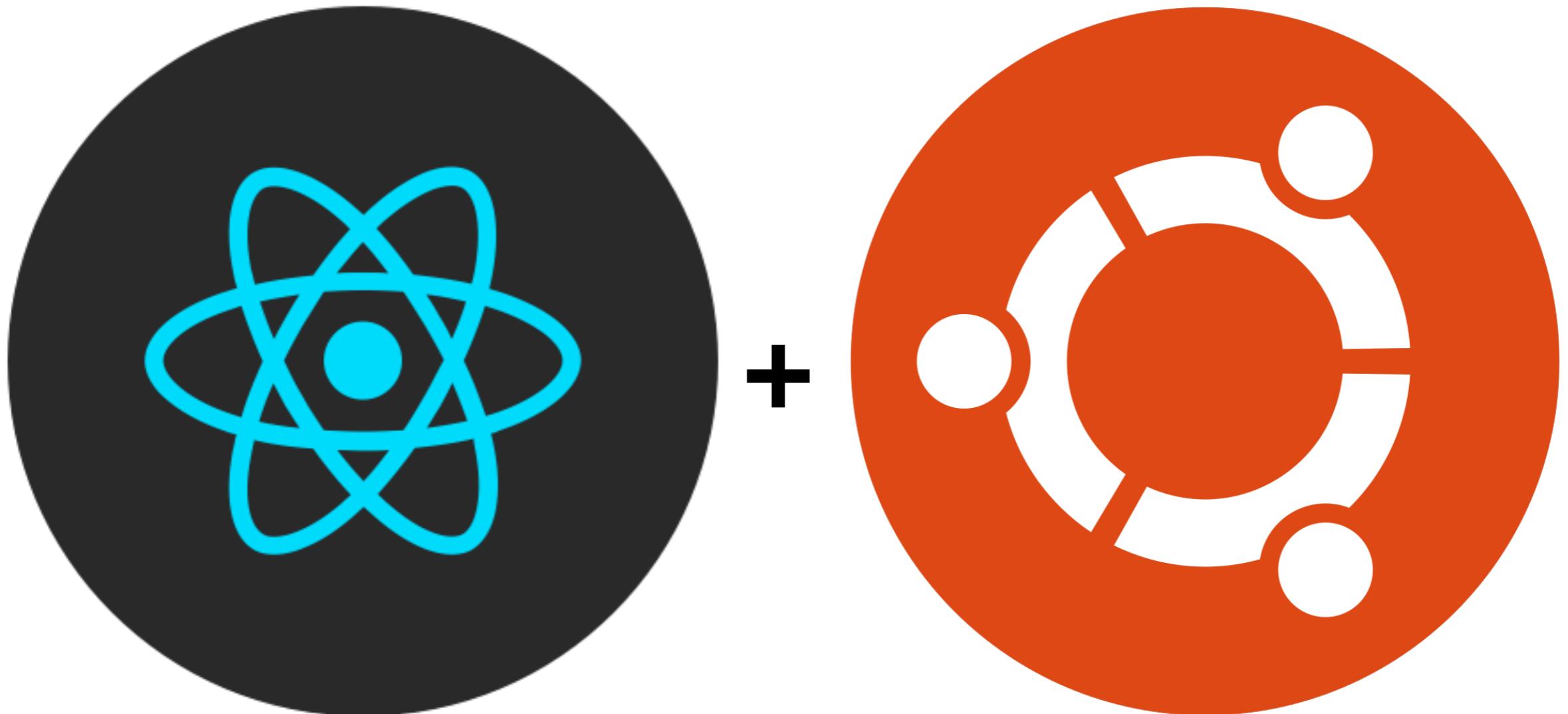
- Depends on the app...
  - Node - 'forever'
- Run as a process?
  - May be linked to the tty
- What about as background process?
  - What if it stops?
  - How can we make sure it stays up?

Possible Solution: SystemD

- SystemD
  - Available on a linux based OS
  - System and service manager for Linux operating systems
  - Manages Daemons
    - Daemon - program running as a background process
- Run an app as a daemon
  - Will be managed/monitored
  - Can be monitored

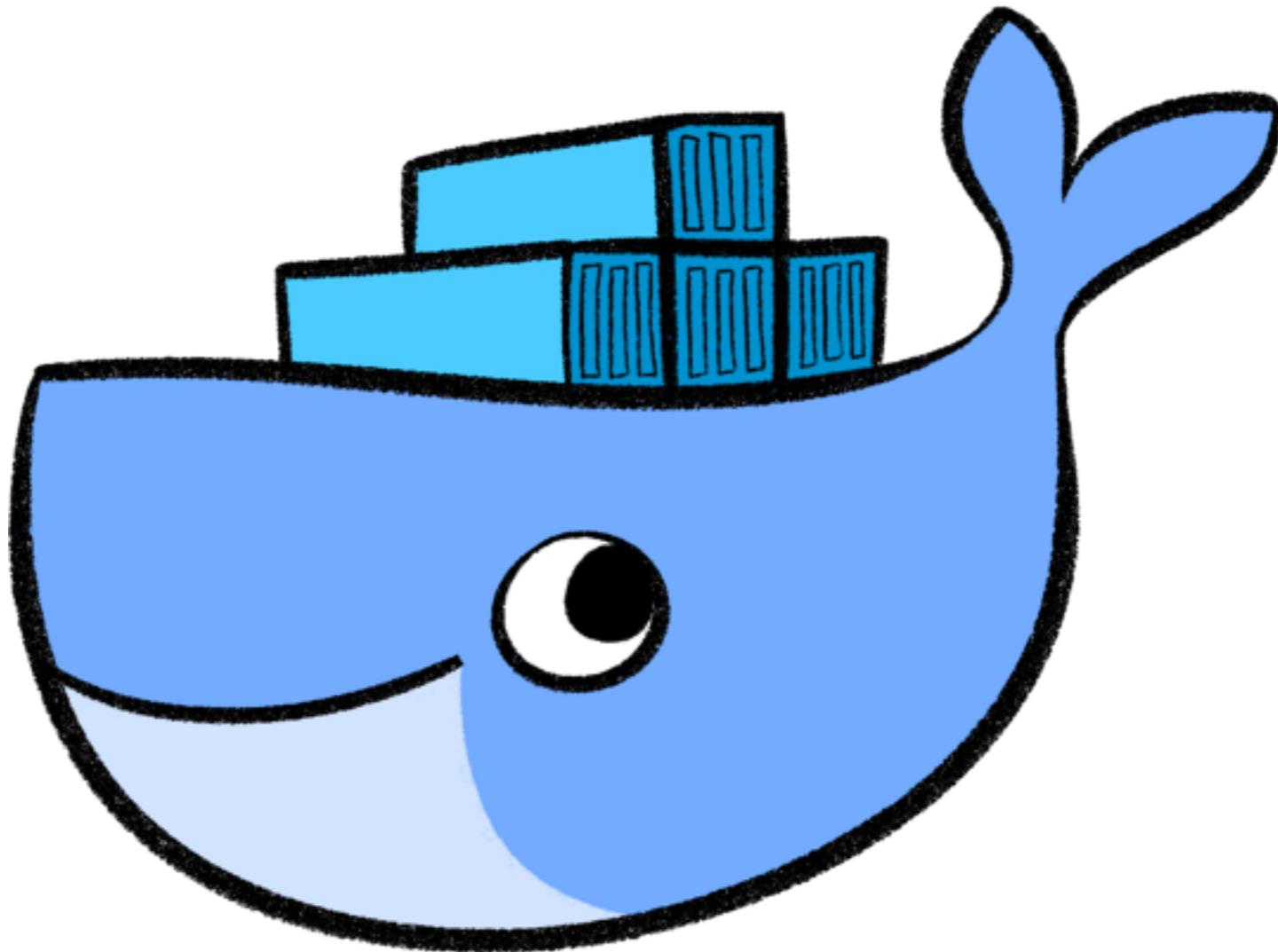
# SystemD - Deploying an App

---



# Docker

---



# Docker - Building the Metaphor

---



# Docker - Why?

---

## Simplicity

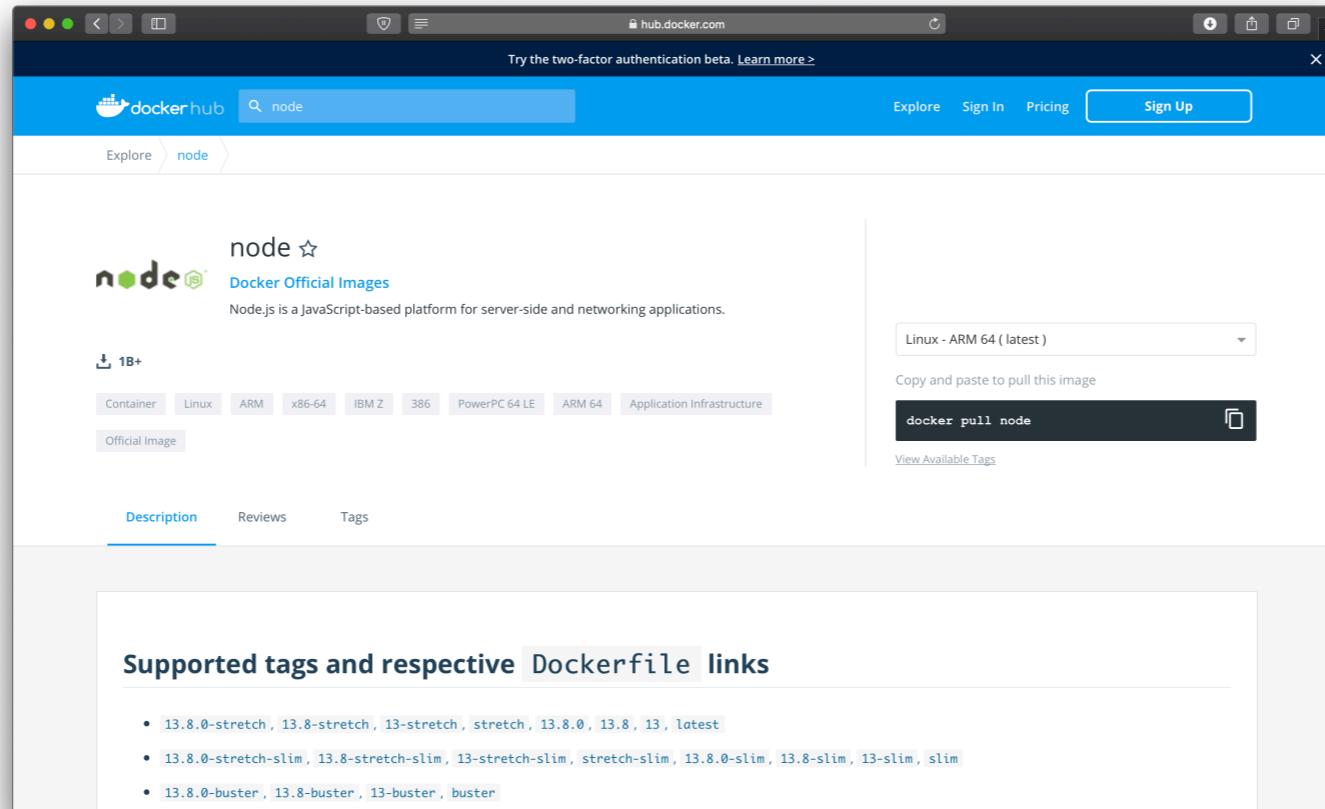
- One file to define everything an app needs to run

## Sharing

- A docker image can be moved between any OS that runs Docker

## Isolation

- Prevents the accumulation of many bins, libs, etc. on your system
- Is isolated from host Network

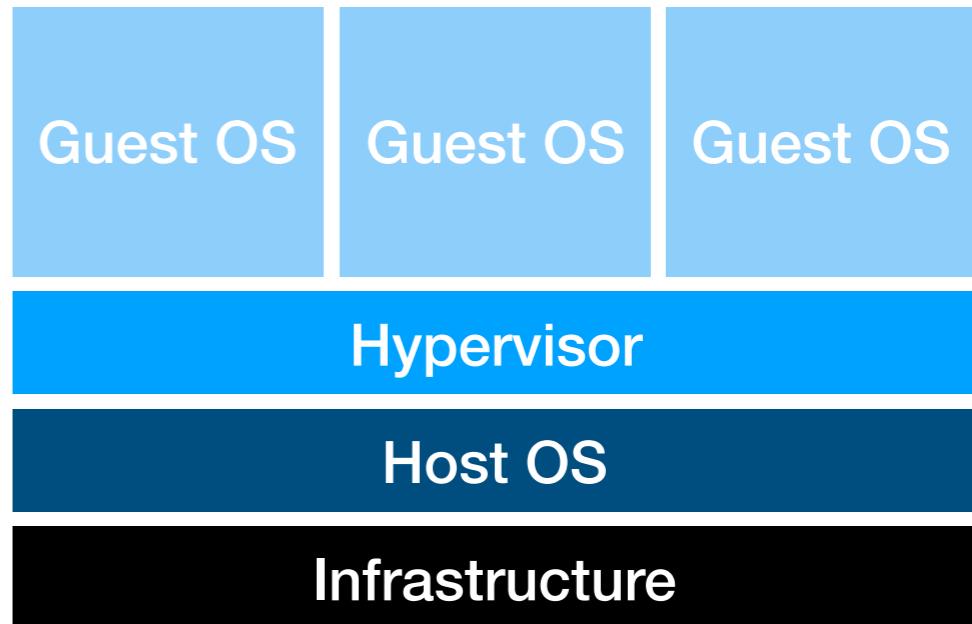


# Docker - Introduction

---

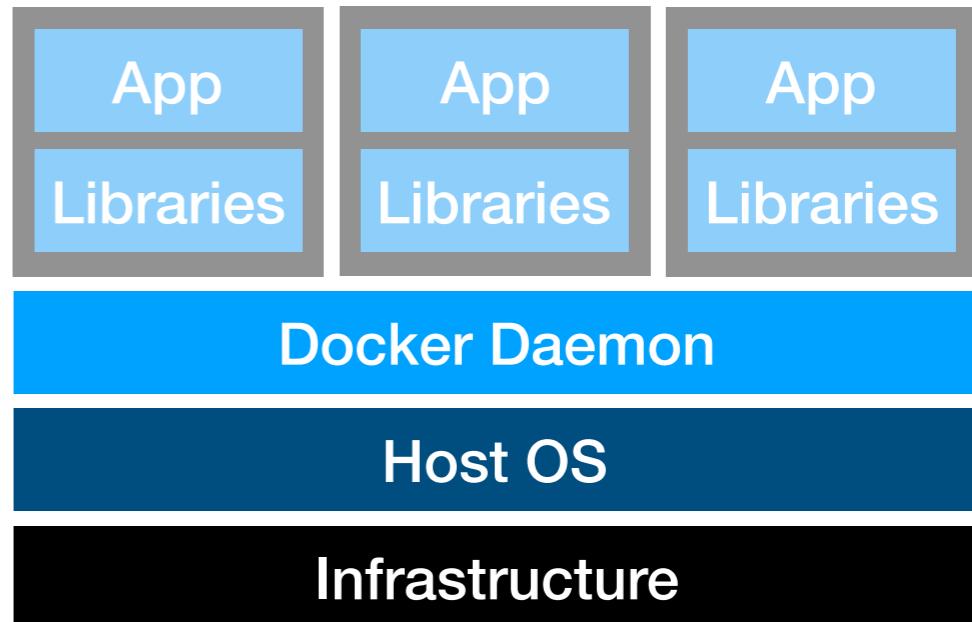
## VM

- Runs on top of the hypervisor
- Each VM has its own Kernel
- Overhead is increased due to a full OS being run



## Docker

- Each container runs a minimal OS and is isolated from the Host
- Small and quick to deploy or shutdown
- Runs anywhere



# Docker - Resources

## Image

- A read-only template for the exact details of how the environment is setup

## Container

- Encapsulated environment that runs applications based on images

## Network

- Runs internal networks isolated from the host network
- Can connect containers together

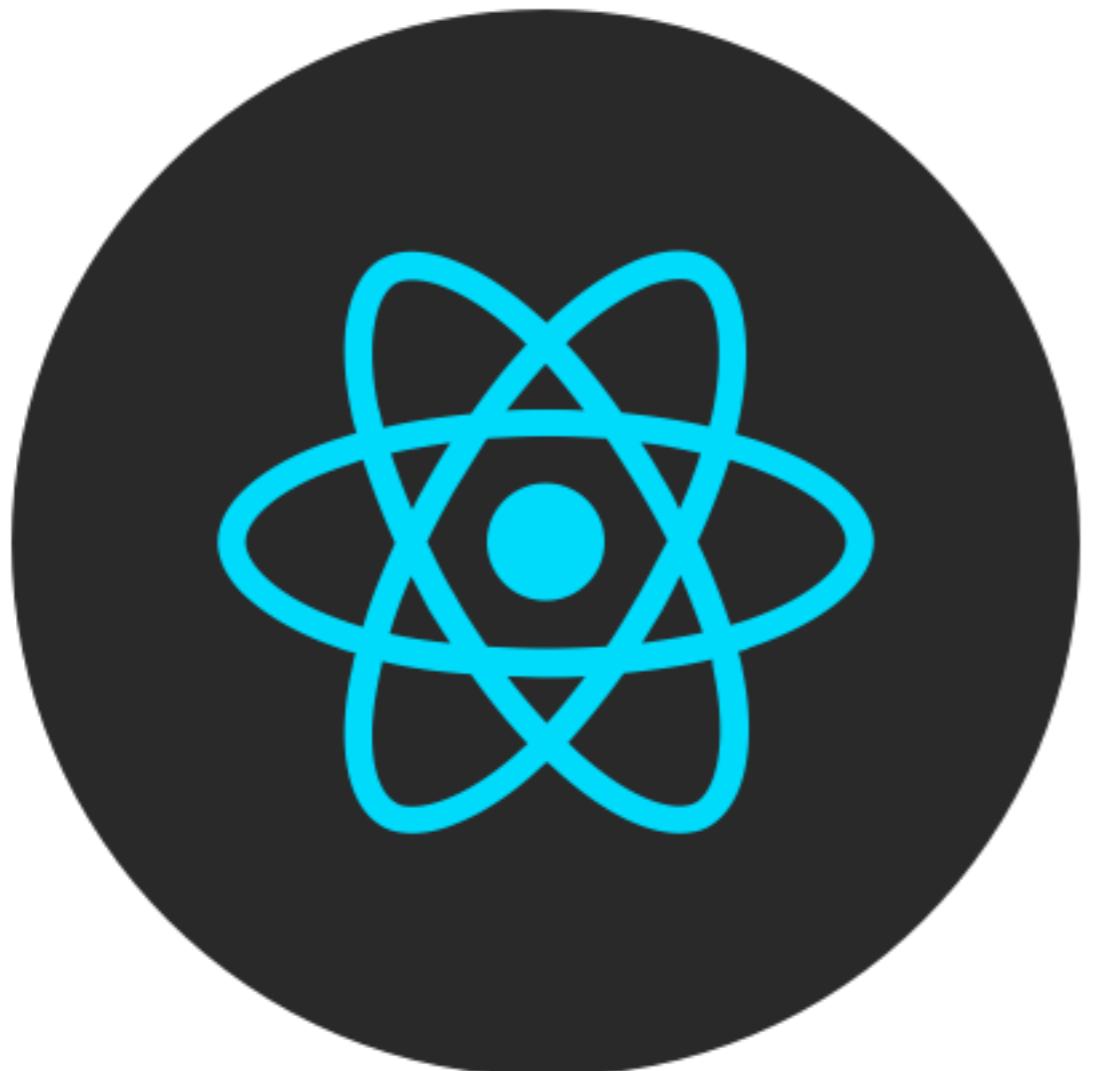
## Even More

- Plugins
- Volumes
- etc.

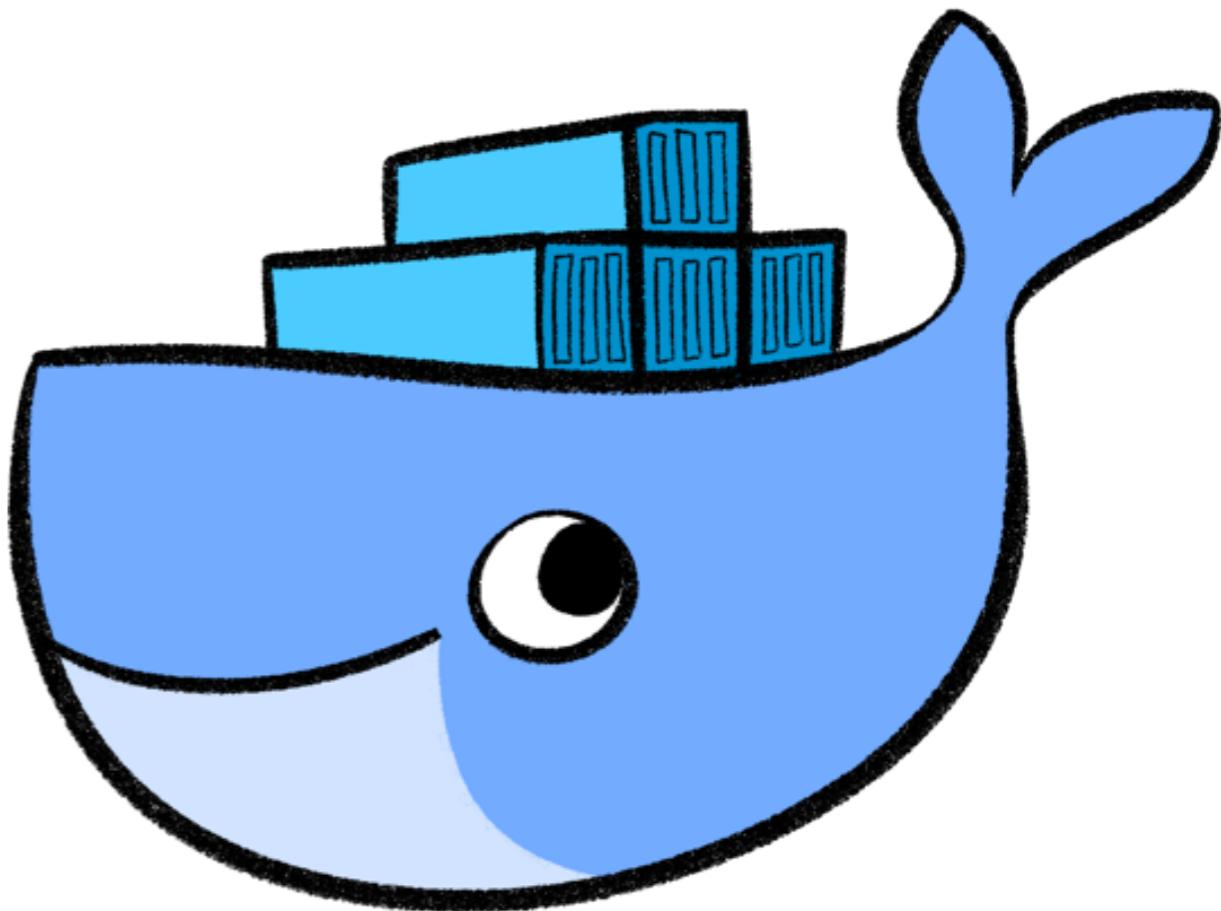
```
lcars@lcars-master:~$ docker network ls
NETWORK ID      NAME    DRIVER      SCOPE
ba83eb0ba69f   bridge  bridge      local
6956077da661   host    host       local
4a9e1daf48d9   nginx_default  bridge      local
3ca4c1d0d6fb   none    null       local
lcars@lcars-master:~$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
dgisolfi/app        1.0.0    19511caaef16  2 hours ago   358MB
app                 latest   19511caaef16  2 hours ago   358MB
node                10-slim   000100717b61  7 days ago    135MB
nginx               latest   2073e0bcb60e  4 weeks ago   127MB
k8s.gcr.io/kube-apiserver  v1.14.10  676810812abc  2 months ago  209MB
k8s.gcr.io/kube-controller-manager  v1.14.10  bcf954f722dc  2 months ago  157MB
k8s.gcr.io/kube-scheduler   v1.14.10  3fa2504a839b  2 months ago  81.6MB
k8s.gcr.io/kube-proxy     v1.14.10  63b4dab6fab0  2 months ago  82.1MB
weaveworks/weave-npc    2.6.0    5105e13e253e  4 months ago  34.9MB
weaveworks/weave-kube   2.6.0    174e0e8ef23d  4 months ago  114MB
k8s.gcr.io/etcdd        3.3.10   2c4adeb21b4f  15 months ago 258MB
k8s.gcr.io/pause        3.1     da86e6ba6ca1  2 years ago   742kB
lcars@lcars-master:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
577cc586c49f        weaveworks/weave-npc   "/usr/bin/launch.sh"   4 days ago        Up 4 days           k8s_weave-npc_weave-net-gkmvc_kube-system_1da1dc78-5b54-11ea-8d28-0050568945
b0_0                weaveworks/weave-kube   "/home/weave/launch._"  4 days ago        Up 4 days           k8s_weave_weave-net-gkmvc_kube-system_1da1dc78-5b54-11ea-8d28-0050568945b0_0
2b512670b1c1        k8s.gcr.io/pause:3.1    "/pause"           4 days ago        Up 4 days           k8s_POD_weave-net-gkmvc_kube-system_1da1dc78-5b54-11ea-8d28-0050568945b0_0
14e62fc49393        63b4dab6fab0      "/usr/local/bin/kube_"  4 days ago        Up 4 days           k8s_kube-proxy_kube-proxy-xhr4l_kube-system_d9d9f984-5b52-11ea-8d28-00505689
45b0_0              97c222d2e772     "/pause"           4 days ago        Up 4 days           k8s_gcr.io/pause:3.1
```

# Docker - Dockerize and App

---



+



# Kubernetes

---



# Kubernetes - Building the Metaphor

---



# Kubernetes - Introduction/Basic Resources

## Cluster

- A collection of 1 or more nodes

## Node

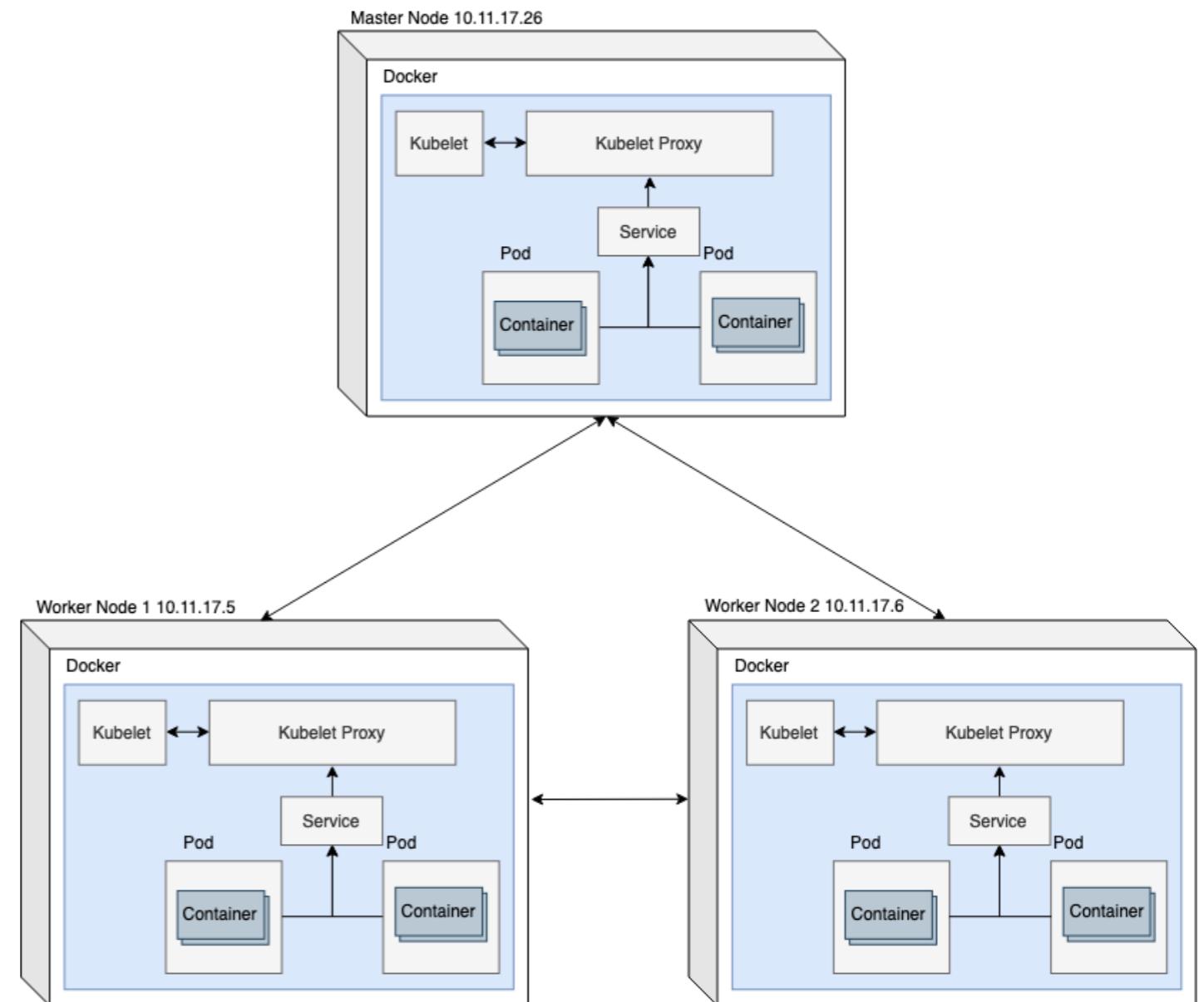
- Each node is a server or VM that has Kubernetes and Docker installed
- The master node controls all pods on all nodes of the cluster

## Pod

- A group of one or more containers with shared storage/network

## Service

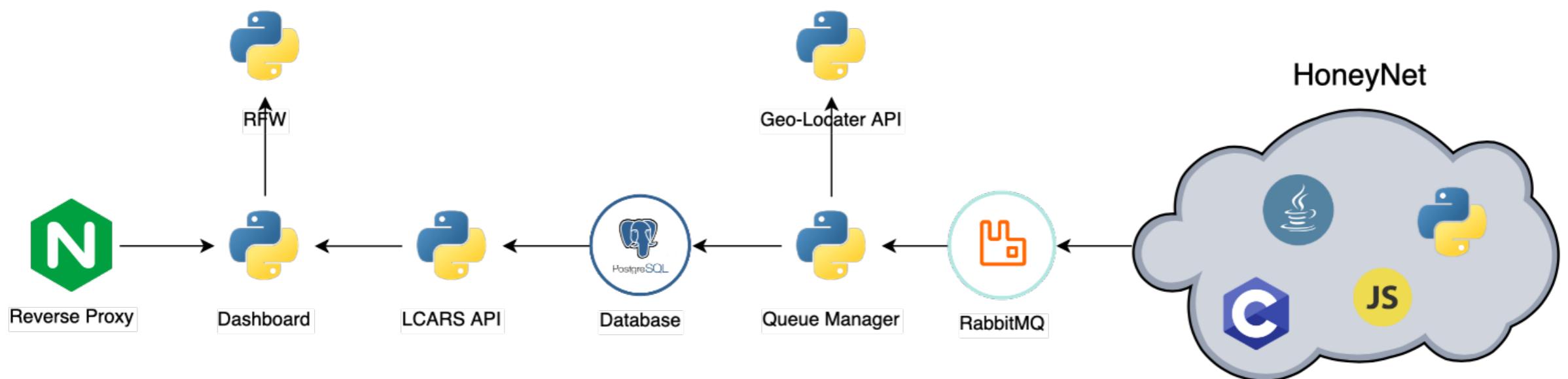
- Exposes an application running on a set of Pods as a network service



# Kubernetes - Use Case

## LCARS

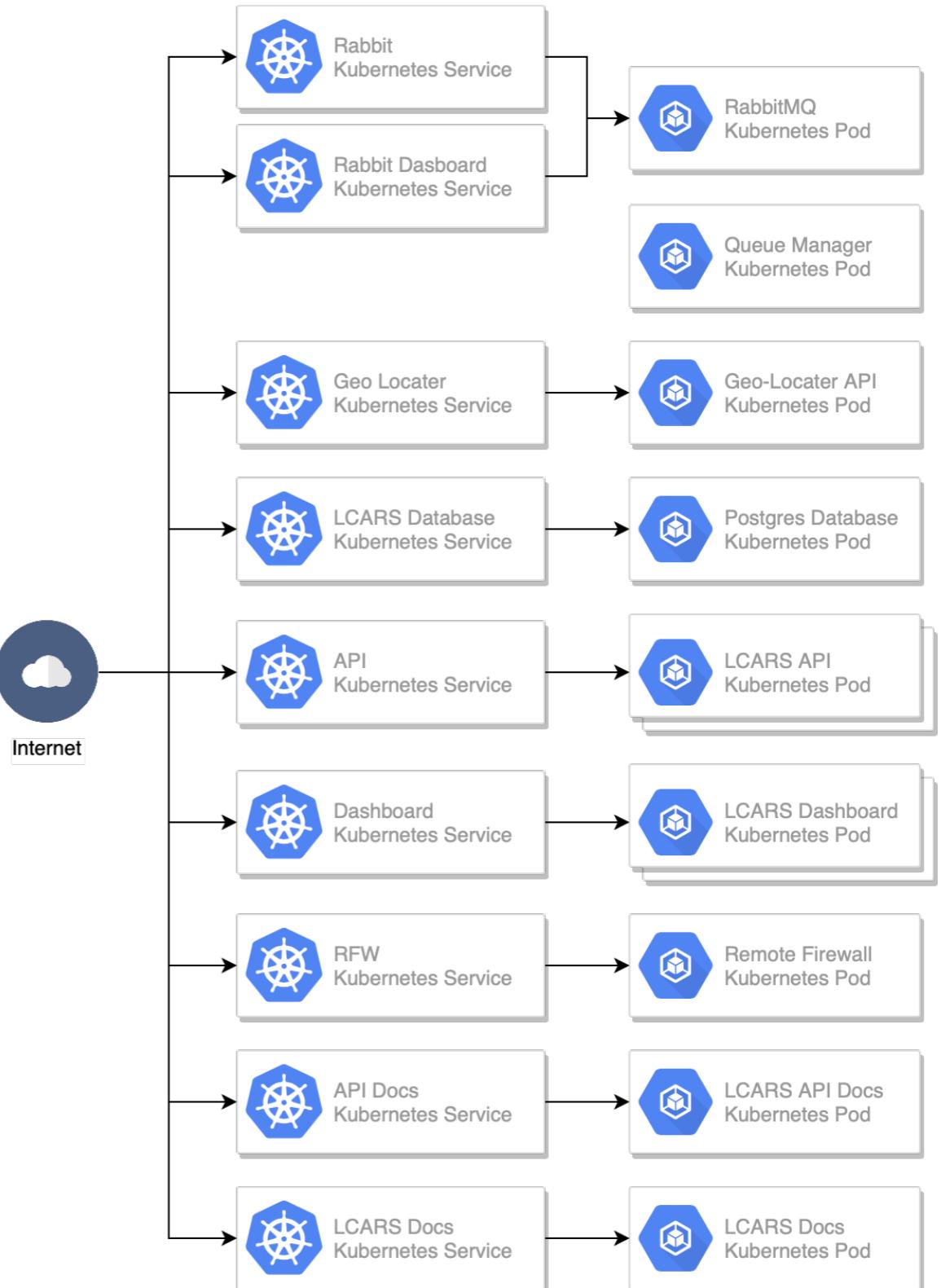
- Comprised of many Micro services
- Needs to be available!
- Adds extra network security
- More than 20 Million attacks



# Kubernetes - Use Case

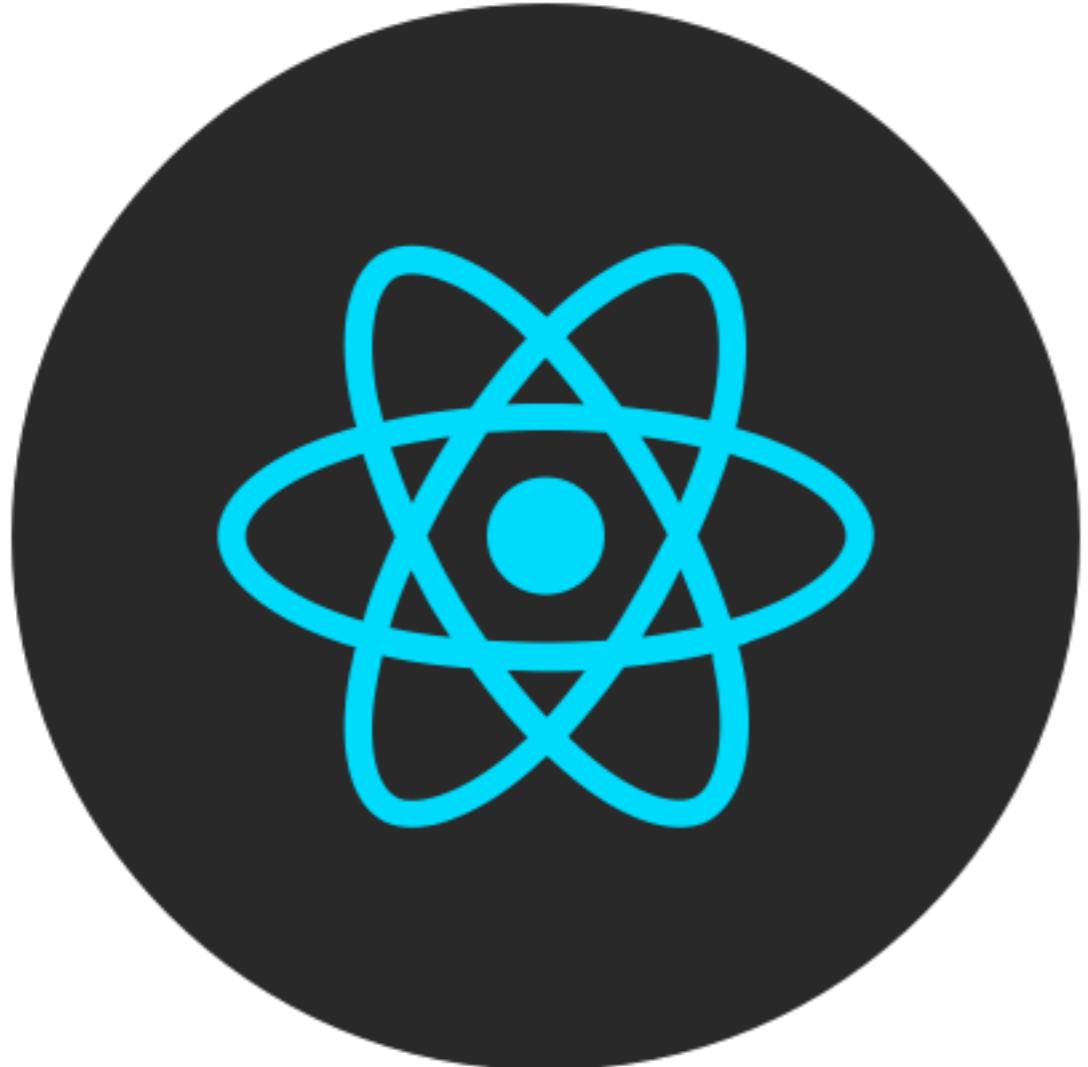
## Why?

- Runs LCARS in the Marist ECRL
- LCARS is made up of 9 **unique** pods
- LCARS can run in docker but becomes difficult/confusing to maintain and update.
- High Availability
- Can run any other needed applications for the IBM/Marist Joint Study



# Kubernetes - Kuberize and App

---



+



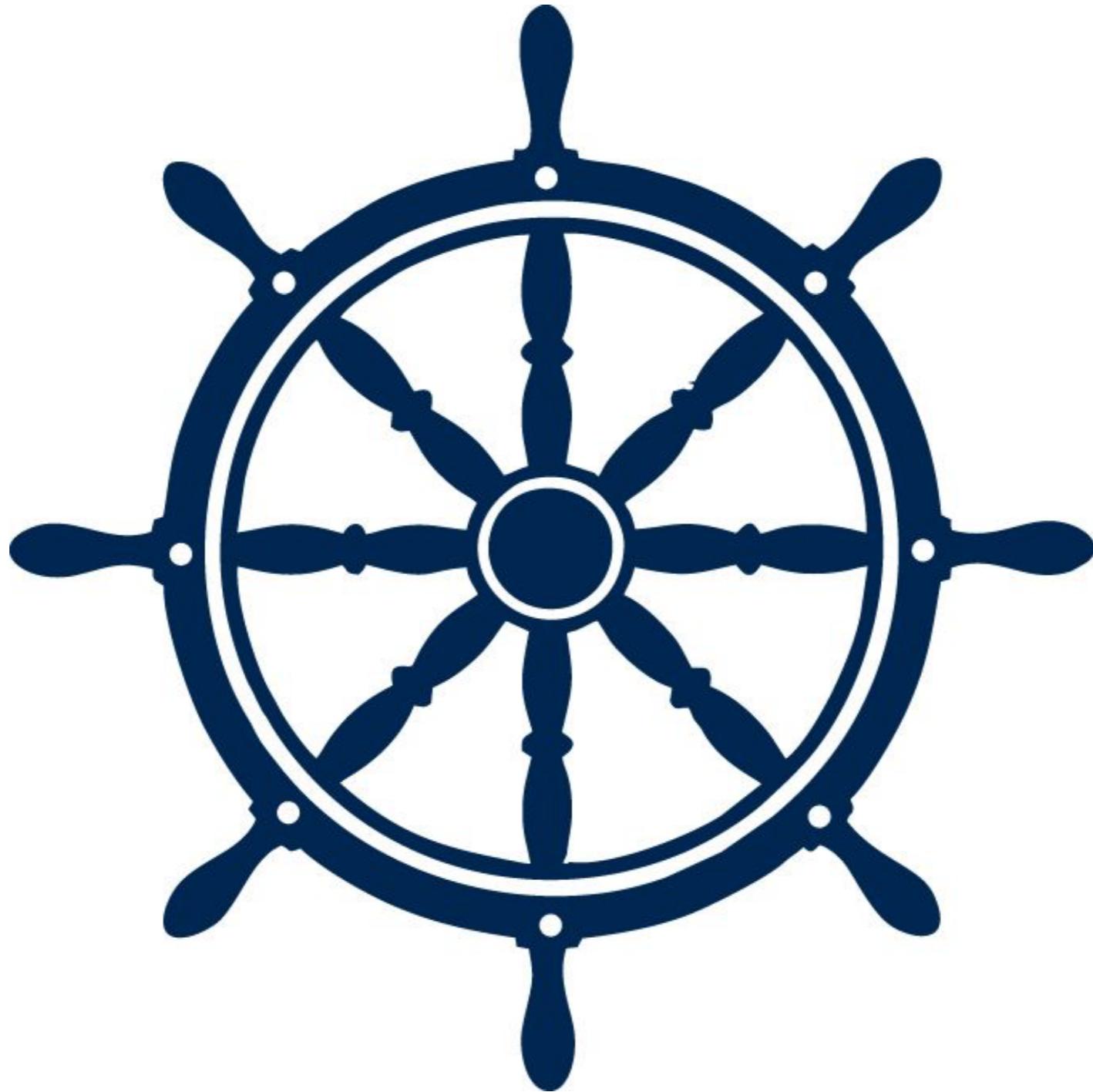
# Helm

---



# Helm - Building the Metaphor

---



# Helm - What is it? & Why bother?

---

## What is it?

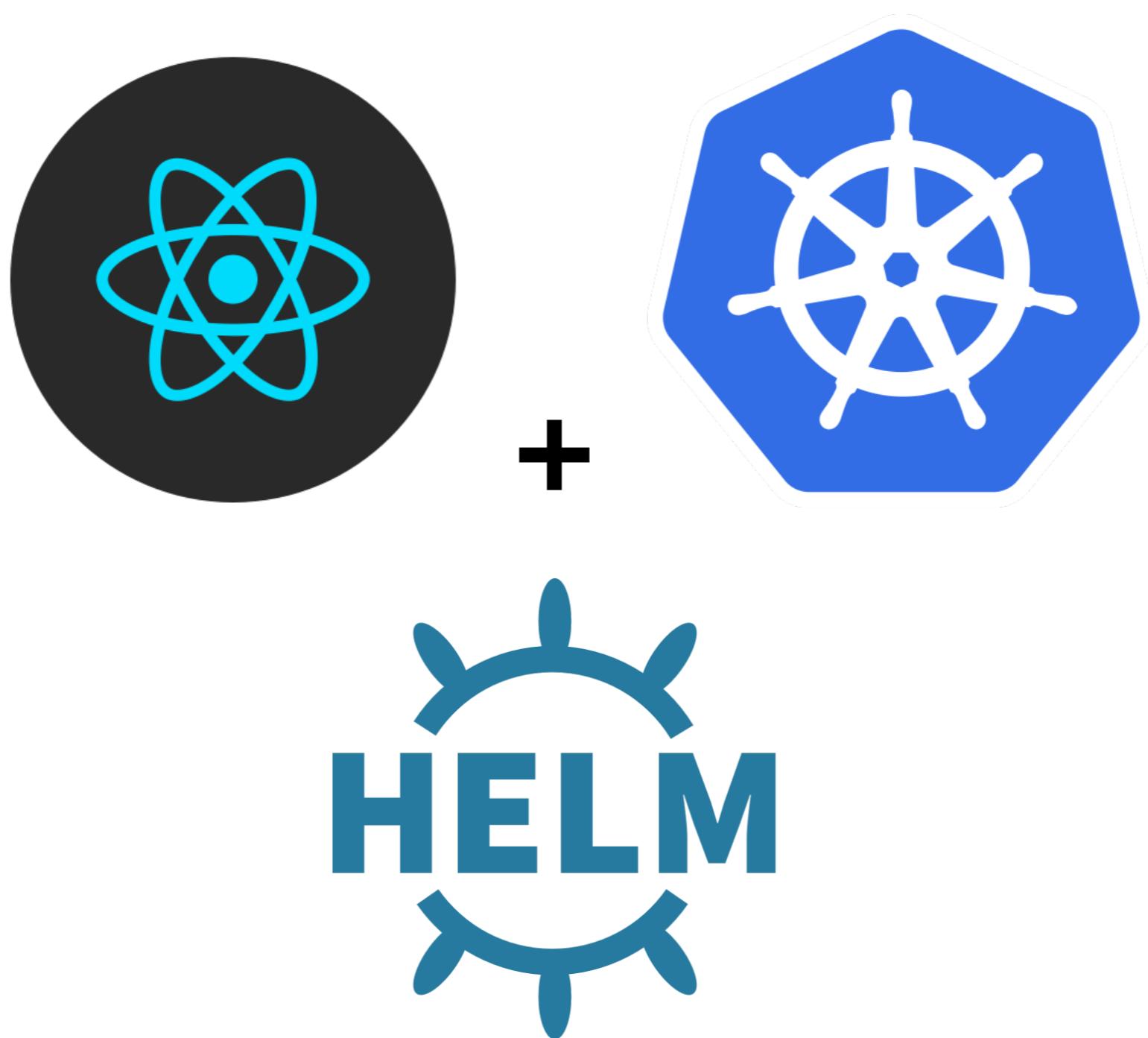
- A package manager for Kubernetes
- Uses charts to deploy apps
- What is a chart?
  - A collection of files that describe a set of Kubernetes resources
- Are they hard to make?
  - Nope - one command

## Why?

- Use Case 1 - I need to share my “App”
  - Will deploy the app and all necessary dependencies
- Use Case 2 - CI/CD
  - Using either Jenkins or Travis helm can manage the application deployment process. Leaving you to developing.

# Helm - An easier way to Kuberize an App

---



# Final Remarks / Advice

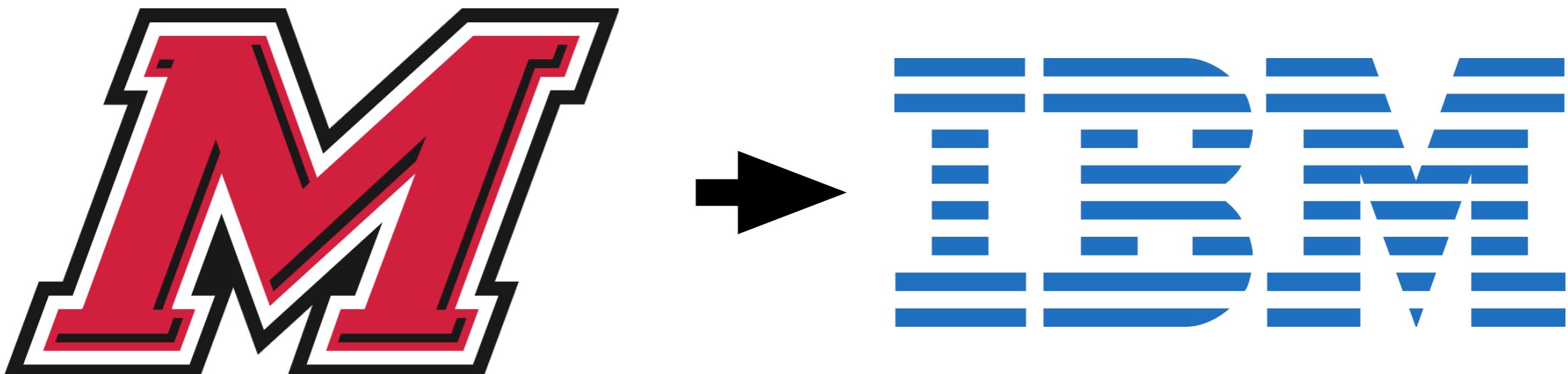
---

## Academic

- Get involved in your classes, take projects and ideas to ridiculous heights
- Lock in the Basics...or it will haunt you
- Use the time at Marist to get an edge
  - Take the time to find your style, interests within Computer Science and run with it
  - Keep an eye on what's up and coming in the industry — Keep up or be left behind

## Practical

- Create a web app and deploy it, start to finish
- Learn a Low level Language as well as a High level one
- **Learn Linux**



Thank you. Questions?

---

# Deploying Apps at Scale

An introduction to Docker and Kubernetes.

Daniel N. Gisolfi

Daniel.Gisolfi1@Marist.edu