Containers, Docker, Kubernetes

Phil Papadopoulos

Shava Smallen

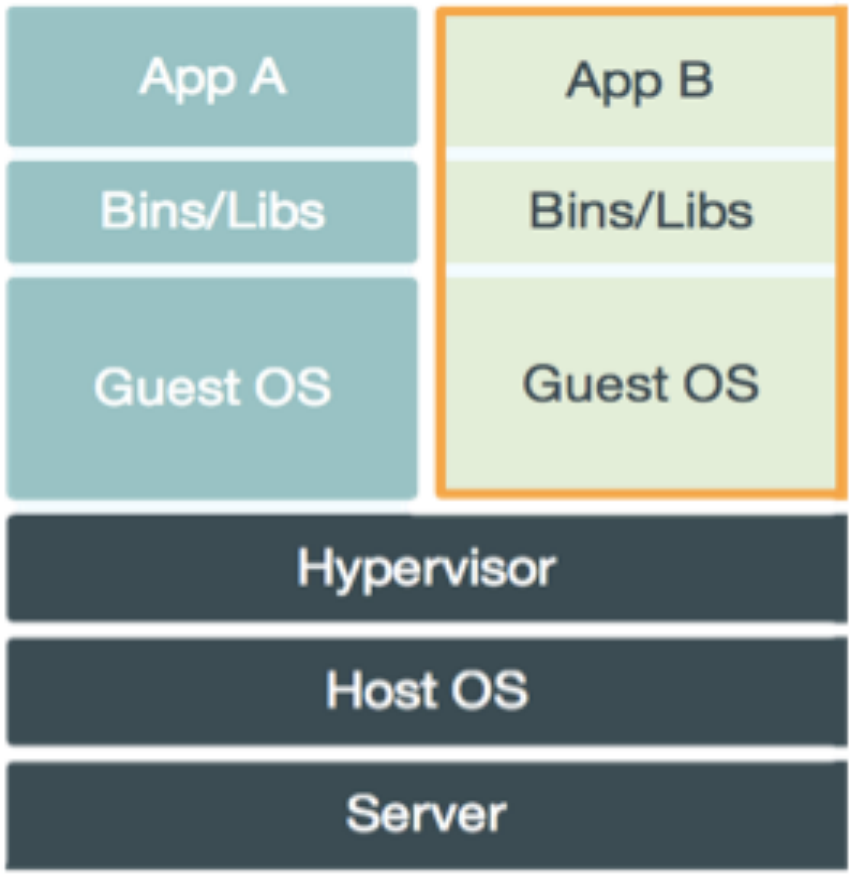
Nadya Williams

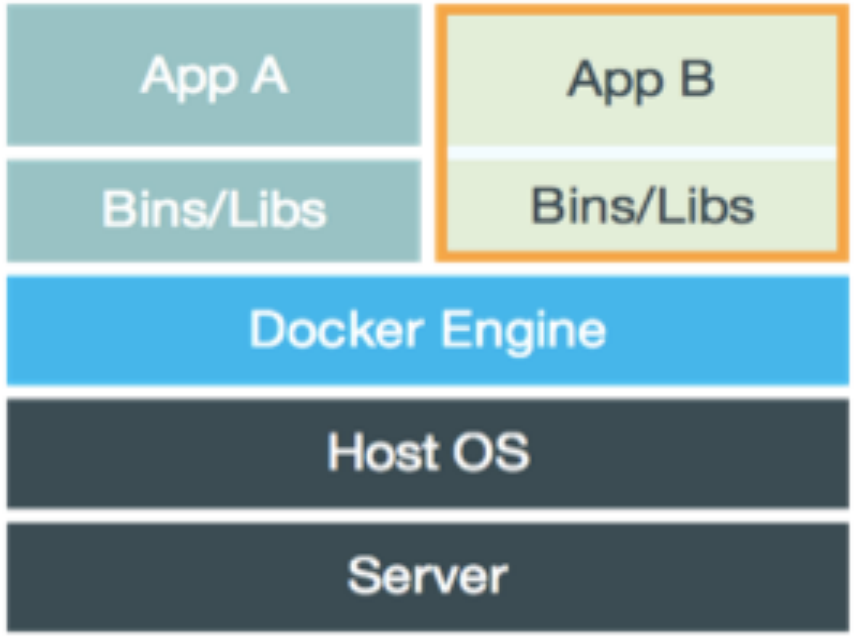
UCSD

Basics

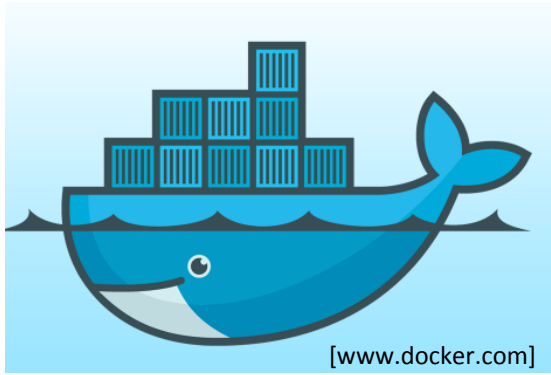
- You have an application (app)
  - And it can depend on
    - Specific library versions
    - Other applications
    - Files/apps in certain locations
- You want to run your app “anywhere”
  - Solution 1: every place you run your app, port it to that environment (Grid Model)
  - Solution 2: build a virtual machine (full operating system) that has your app and dependencies embedded (VMs, Infrastructure as a Service cloud)
  - Solution 3: Package just the essential dependencies that can then be transported as part of your (container)

Virtual Machine vs. Container





What is Docker?


[www.docker.com]

- *Docker is an open-source project that automates the deployment of applications inside software containers*
- Provides a uniformed wrapper around a software package: *«Build, Ship and Run Any App, Anywhere»*

Without anything else, docker runs containerized applications on a single server or laptop

If you have cluster of physical machines and you want to run containers anywhere on your cluster…

This is called ORCHESTRATION

Kubernetes (also written as K8S) is a very popular system for running containers on clusters of hardware

Docker Swarm, Apache Mesos are other orchestration systems that can be deployed to local hardware

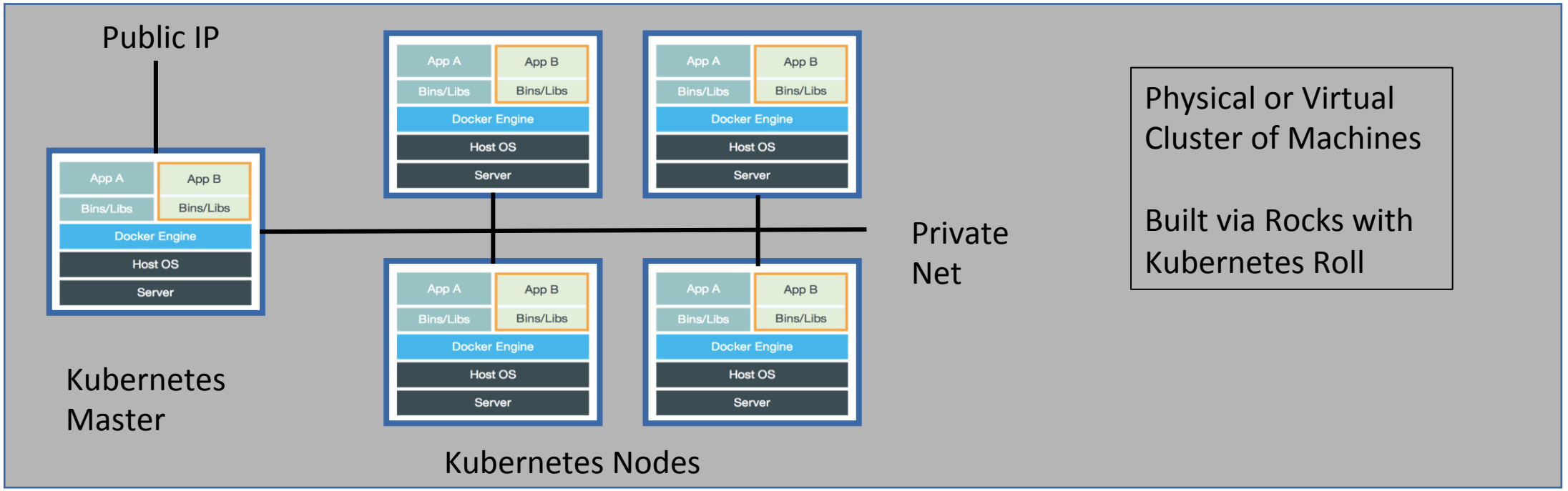
Amazon ECS and Google GCE are commercial (cloud) implementations that allow you to “bring your container” and “leave the orchestration to them

What Does Kubernetes do?

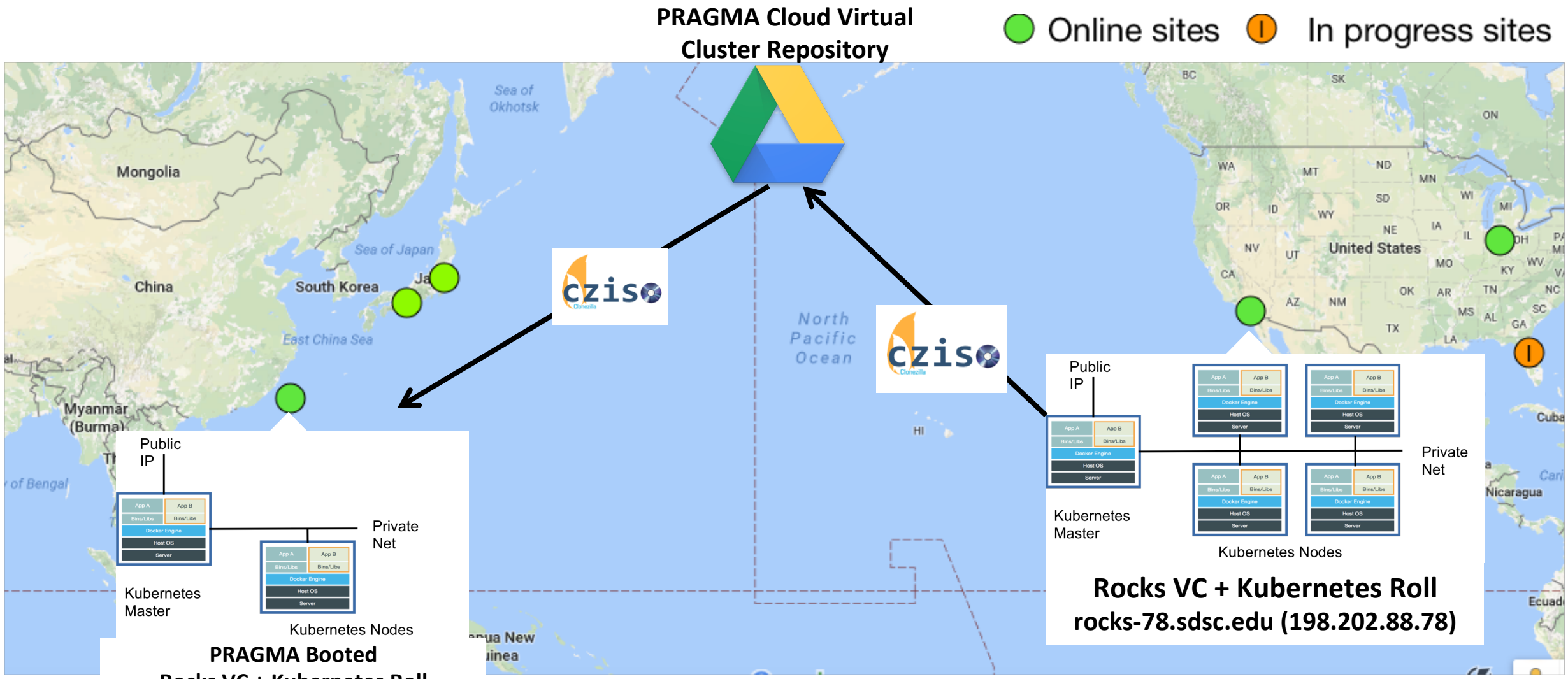
- Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.
- Improves reliability
  - **Continuously monitors** and manages your containers
  - **Can scale your application** to handle changes in load
- **Coordinates what containers run where and when** across your system
- **Creates a networking layer** that enables all the different types of containers in a system talk to each other
- Easily coordinate deployments of your system
  - Which containers need to be deployed
  - Where should the containers be deployed

Are containers a good direction for PRAGMA?

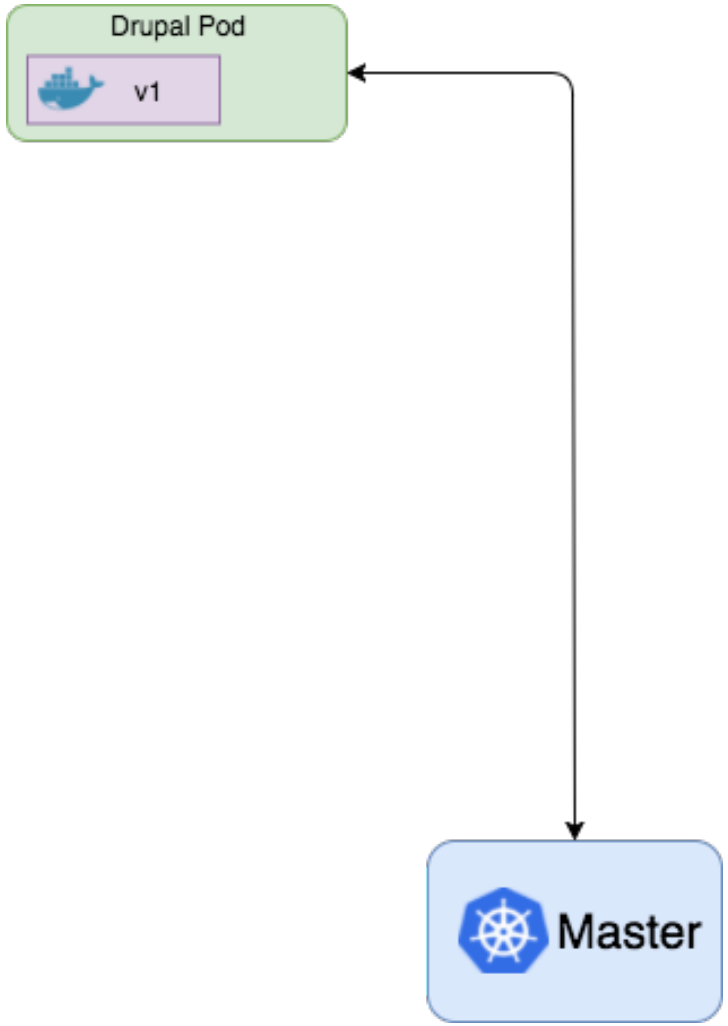
- Need a way to easily build and test containers
- Reduce time for PRAGMA expeditions to experiment with containers
- Build a K8S cluster on a virtual cluster.
- Transport the virtual cluster using PRAGMA Boot to another location



DEMO: PRAGMA Cloud + Kubernetes



The Pod is the core Kubernetes Component



- The Pod is the core component of Kubernetes
- Collection of 1 or more containers
- Each pod should focus on one container, however sidecar containers can be added to enhance features of the core container

```
spec:
  template:
    spec:
      containers:
        - name: drupal
          image: cr.io/repo/mydrupal:v1
```

Docker Container Lifecycle ……

- The Life of a Container
  - Conception
    - **BUILD** an Image from a Dockerfile
  - Birth
    - **RUN** (create+start) a container
  - Reproduction
    - **COMMIT** (persist) a container to a new image
    - **RUN** a new container from an image
  - Sleep
    - **KILL** a running container
  - Wake
    - **START** a stopped container
  - Death
    - **RM** (delete) a stopped container
- Extinction
  - **RMI** a container image (delete image)