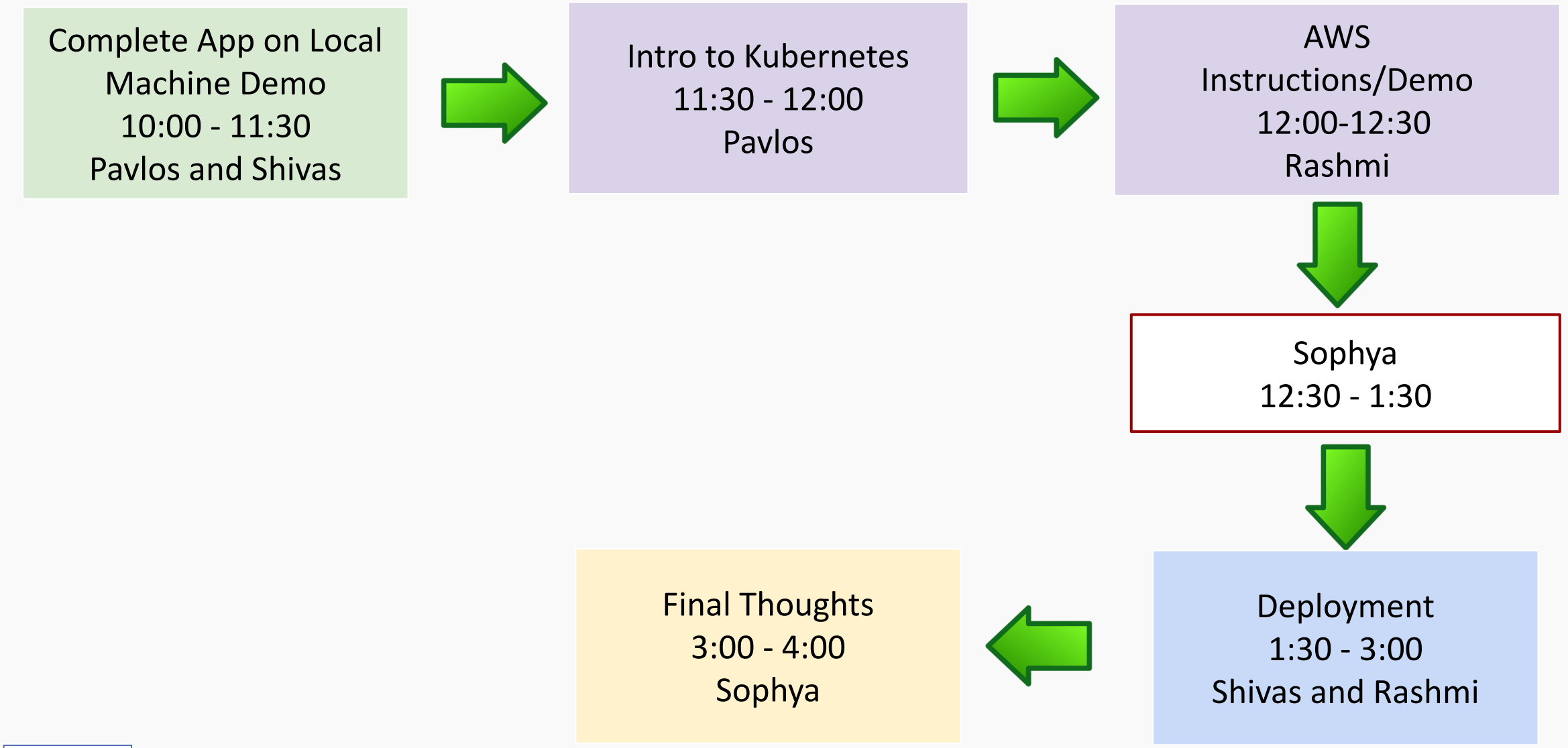
Day 4: Deployment: Front-end, Kubernetes and AWS

Pavlos Protopapas
Institute for Applied Computational Science
Harvard

Workshop Overview for Day 1



Outline

1. Review of Day 1-3
2. Front End
3. Motivation for Kubernetes

Virtual Environment

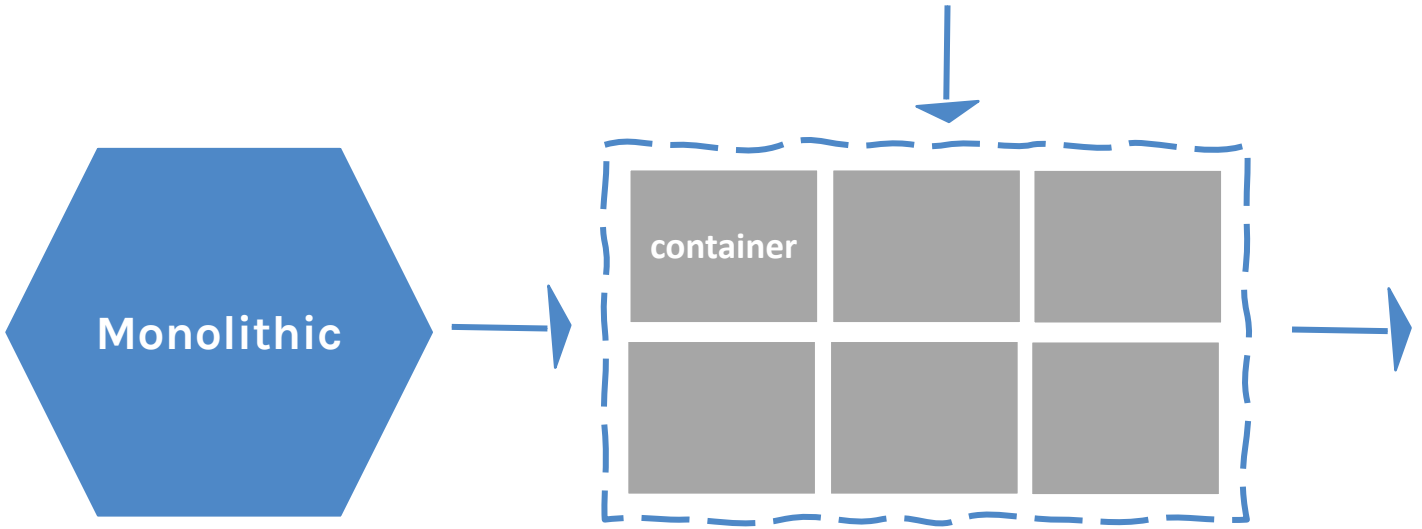
Pros: remove complexity
Cons: does not isolate from OS

Containers

Pros: lightweight
Cons: issues with security, scalability, and control

Virtual Machines

Pros: isolate OS guest from the host
Cons: intensive use of hardware



How to manage microservices?

Recap

We talked about pros/cons of

environments:

remove complexity but does not isolate from OS

virtual machines:

isolate OS guest from host but intensive use of the hardware

containers

lightweight but issue with security, scalability, and control

Recap

Goal:

find effective ways to deploy our apps

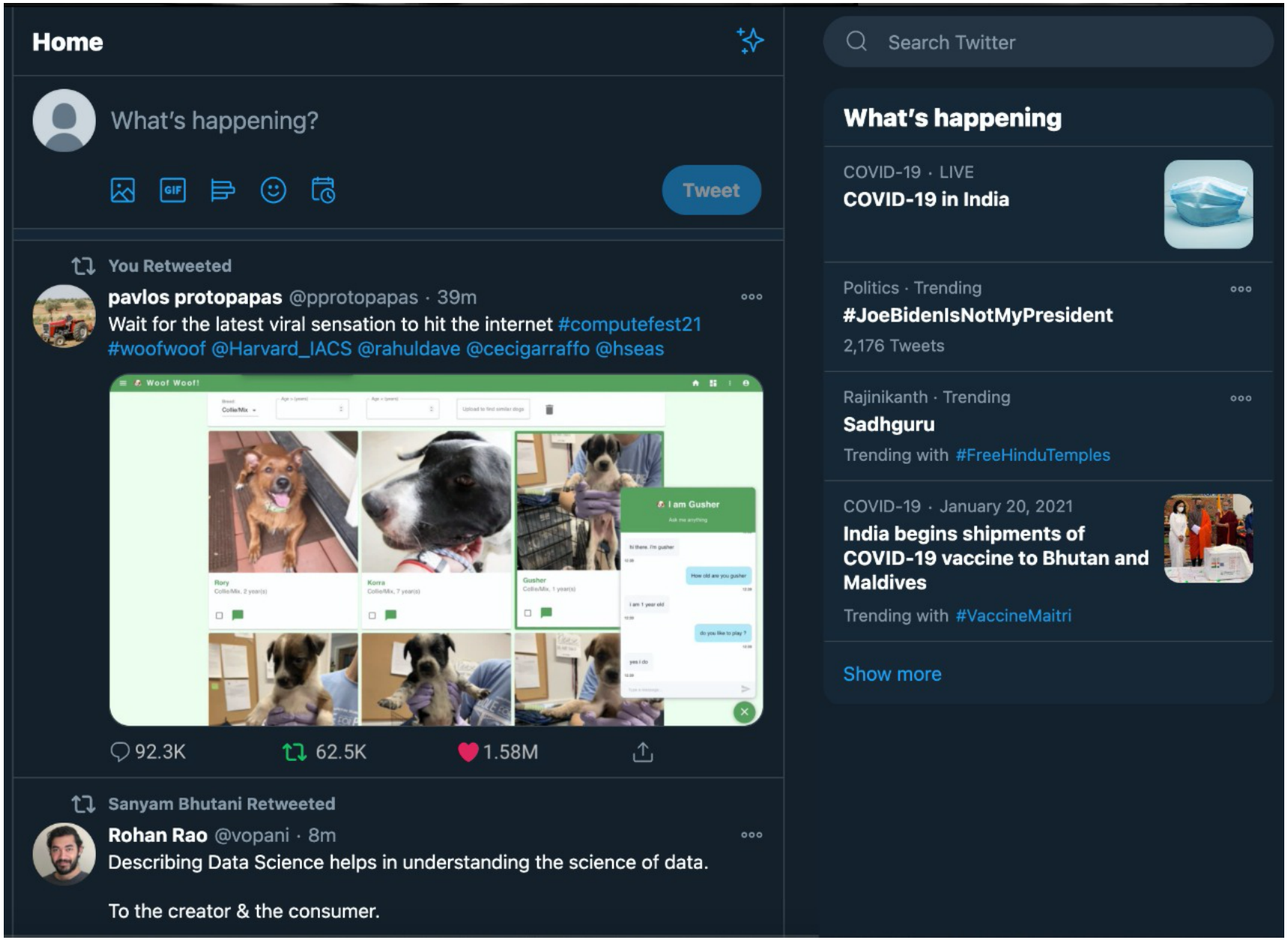
and to **break down a complex application** into smaller ones (*i.e. microservices*)

Issues we have fixed so far:

- Conflicting of different operating system
- different dependencies
- "inexplicable" strange behavior

Outline

1. Review of Day 1-3
2. **Complete the app/Front End**
3. Motivation for Kubernetes


Home
What's happening?
Tweet
You Retweeted
pavlos protopapas @pprotopapas · 39m
Wait for the latest viral sensation to hit the internet #computefest21 #woofwoof @Harvard_IACS @rahuldave @cecigarraffo @hseas
92.3K
62.5K
1.58M
Sanyam Bhutani Retweeted
Rohan Rao @vopani · 8m
Describing Data Science helps in understanding the science of data.
To the creator & the consumer.
Search Twitter
What's happening
COVID-19 · LIVE
COVID-19 in India
Politics · Trending
#JoeBidenIsNotMyPresident
2,176 Tweets
Rajinikanth · Trending
Sadhguru
Trending with #FreeHinduTemples
COVID-19 · January 20, 2021
India begins shipments of COVID-19 vaccine to Bhutan and Maldives
Trending with #VaccineMaitri
Show more

Introduction to Kubernetes <K8s>

K8s manages containers

K8s is an open-source platform for container management developed by Google and introduced in 2014. It has become the standard API for building cloud-native applications, present in nearly every public cloud.

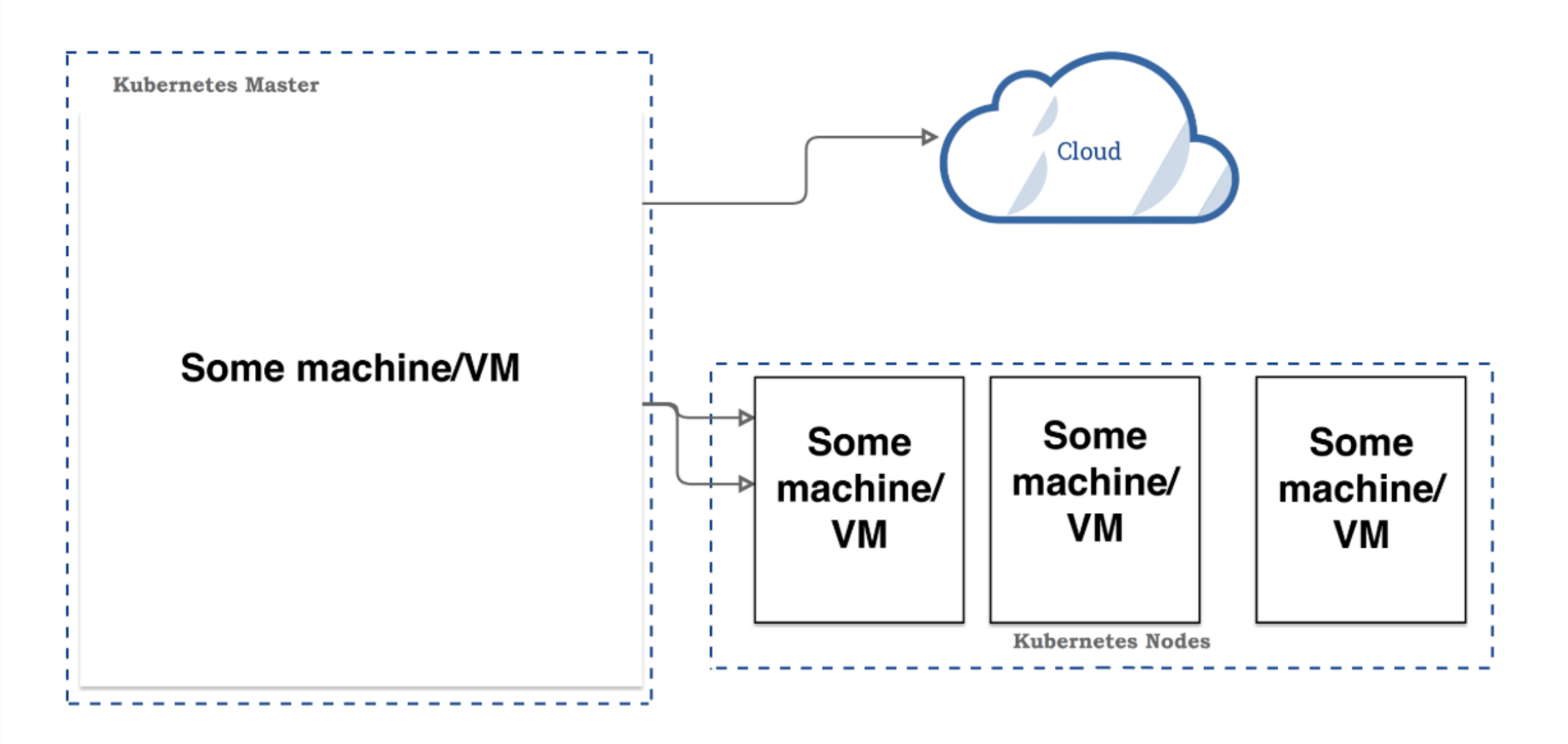
K8s users define rules for how container management should occur, and then K8s handles the rest!

> link to website <

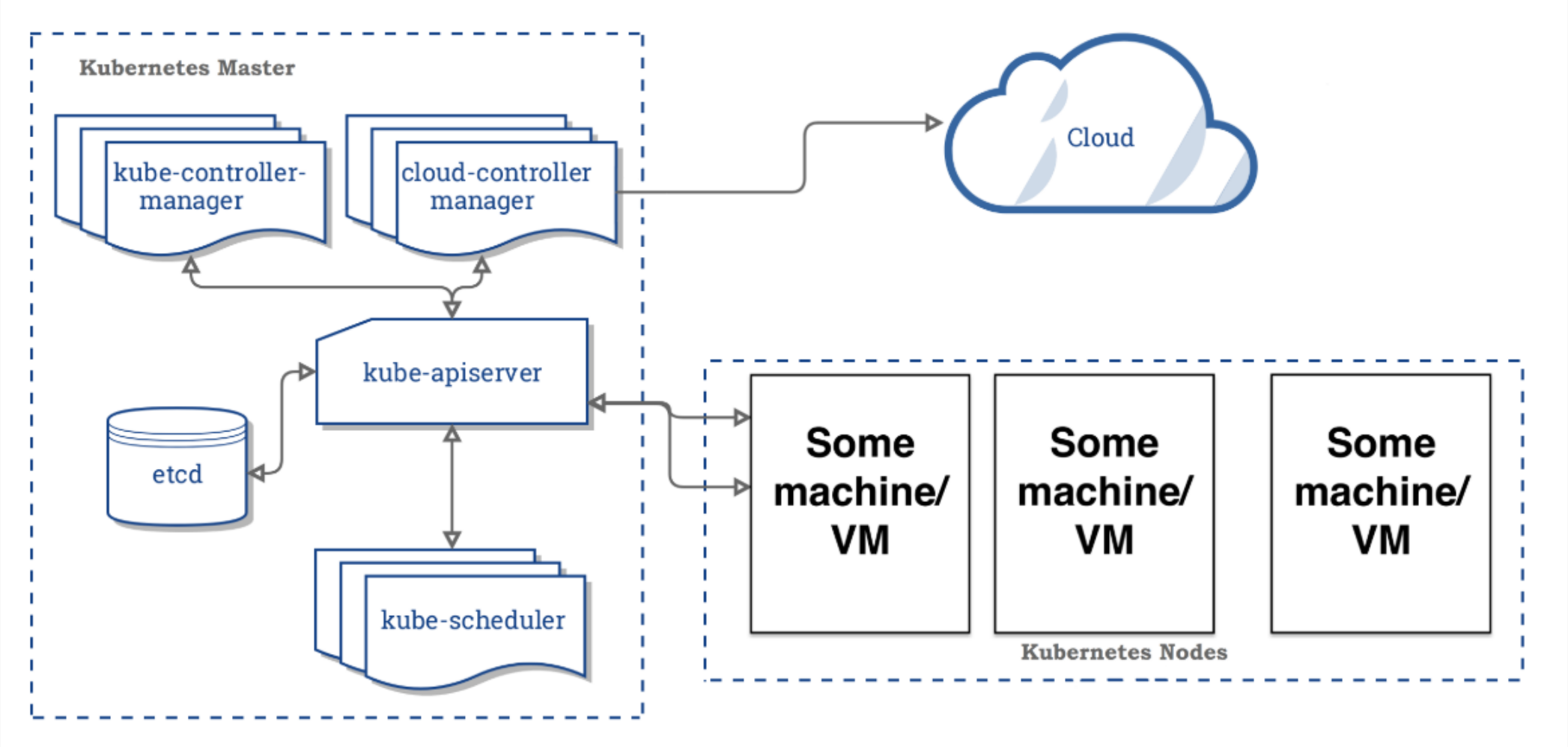
Anatomy of Kubernetes Cluster

- K8s works on a cluster of machines/nodes
- This could be VMs on your local machine or a group of machines through a cloud provider
- The cluster includes one master node and at least one worker node

Anatomy of Kubernetes Cluster <cont>



Anatomy of Kubernetes Cluster | Master Node



> to learn more on etcd <

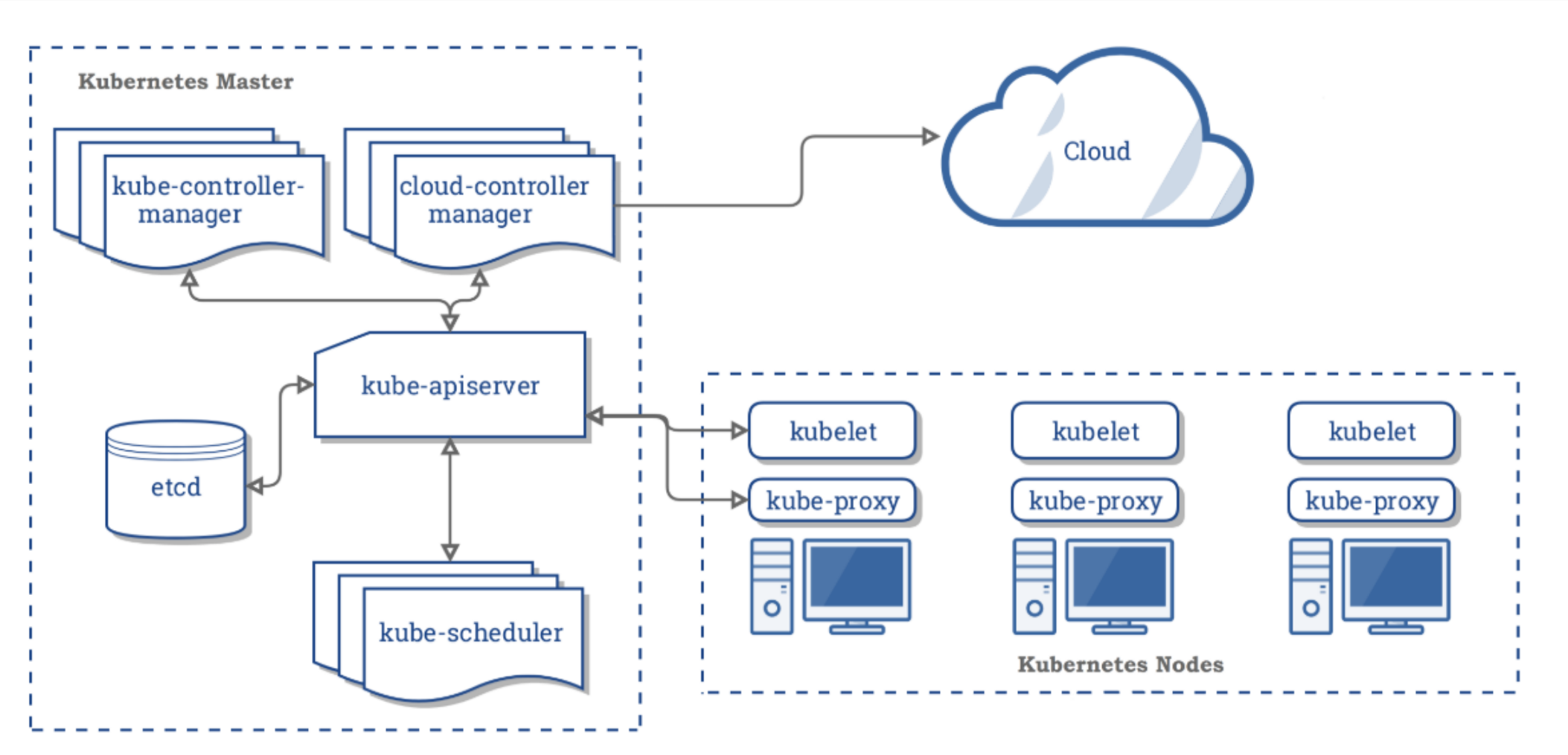
Anatomy of Kubernetes Cluster | Master Node

Master node main task is to manage the worker node(s) to run an application

The master node consists of:

1) **API server** contains various methods to directly access the Kubernetes
2) **Scheduler** assigns to each worker node an application
3) **Controller manager**
   - 3a) Keeps track of worker nodes
   - 3b) Handles node failures and replicates if needed
   - 3c) Provide endpoints to access the application from the outside world
4) **Cloud controller** communicates with cloud provide regarding resources such as nodes and IP addresses
5) **Etcd** works as backend for service discovery that stores the cluster's state and its configuration

Anatomy of Kubernetes Cluster | Worker Nodes



Anatomy of Kubernetes Cluster | Worker Nodes

A worker node consists of:

1) **Container runtime** that pulls a specified Docker image and deploys it on a worker node

2) **Kubelet** talks to the API server and manages containers on its node

3) **Kube-proxy** load-balances network traffic between application components and the outside world

Workshop Overview for Day 1