Demo Placeholder

- Running first app on kubernetes

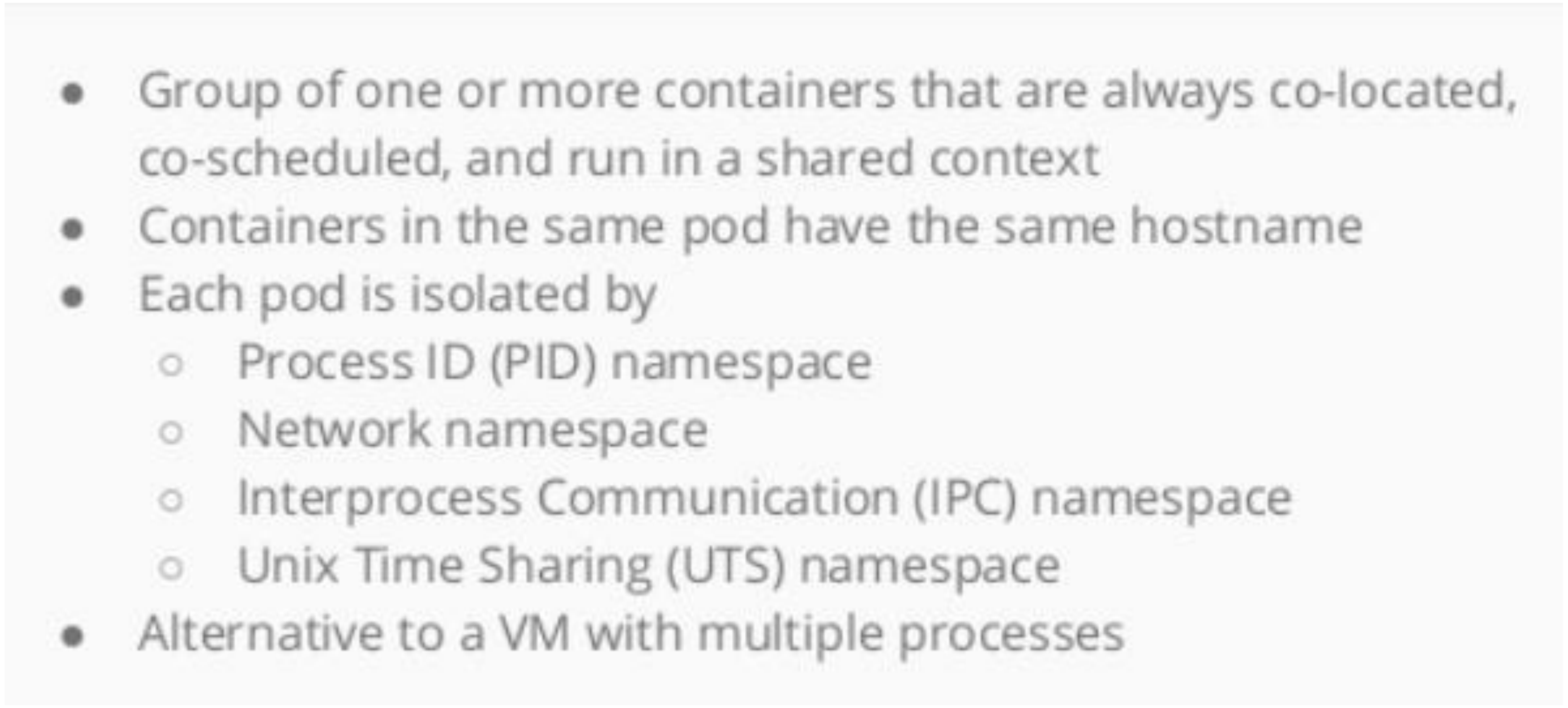
First app

- Let’s run our newly built application on the new Kubernetes cluster
- Before we can launch a container based on the image, we need to create a **pod definition**
- **A pod** describe an application running on kubernetes
- A pod can contain one or more tightly coupled containers, that make up app
  - Those apps can easily communicate with each other using their local **port numbers**
- Our app only has **one** container

Demo

- CREATING POD WITH LABELS USING YAML
- a simple manifest that creates 2 containers.

Pods

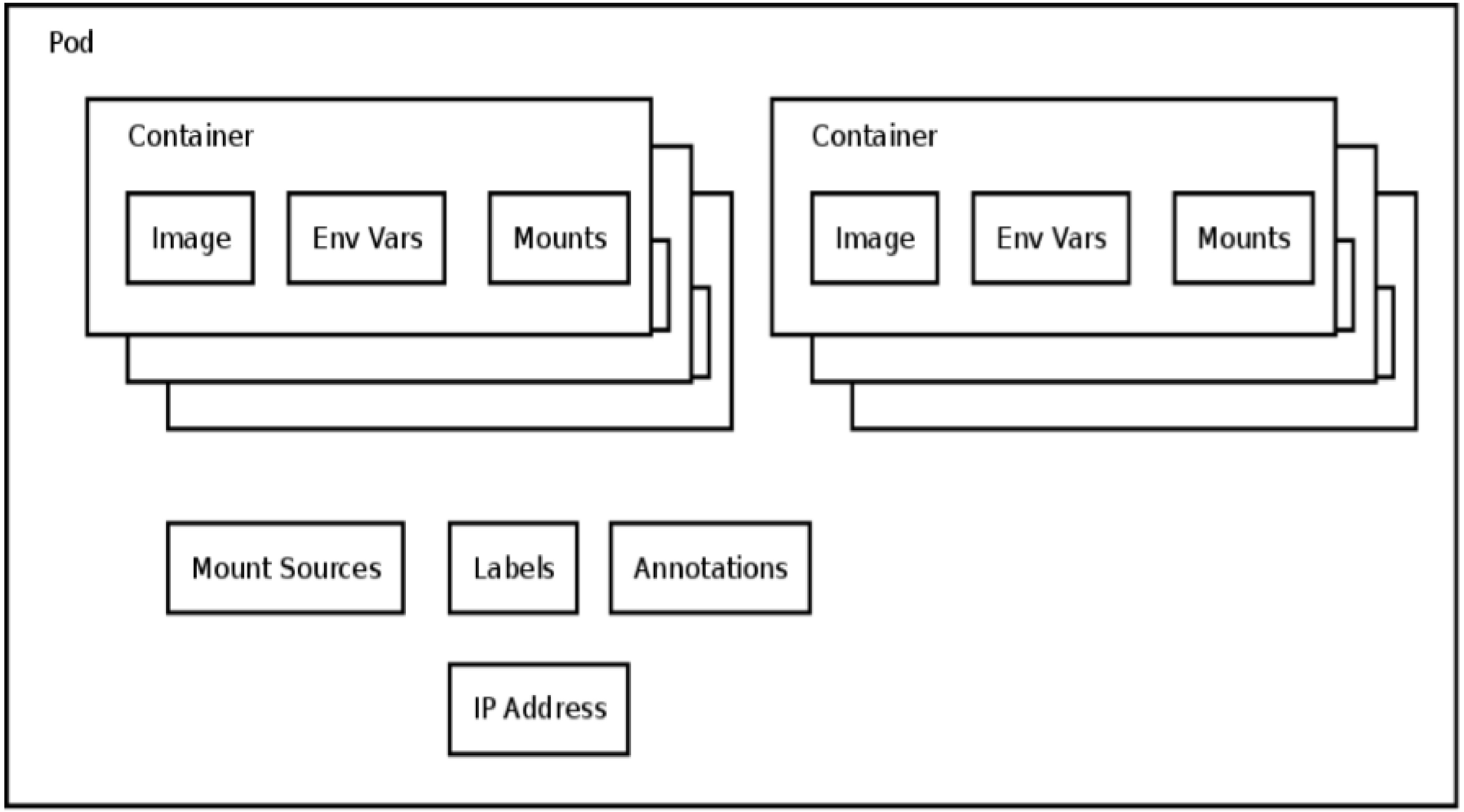


- Group of one or more containers that are always co-located, co-scheduled, and run in a shared context
- Containers in the same pod have the same hostname
- Each pod is isolated by
    - Process ID (PID) namespace
    - Network namespace
    - Interprocess Communication (IPC) namespace
    - Unix Time Sharing (UTS) namespace
- Alternative to a VM with multiple processes

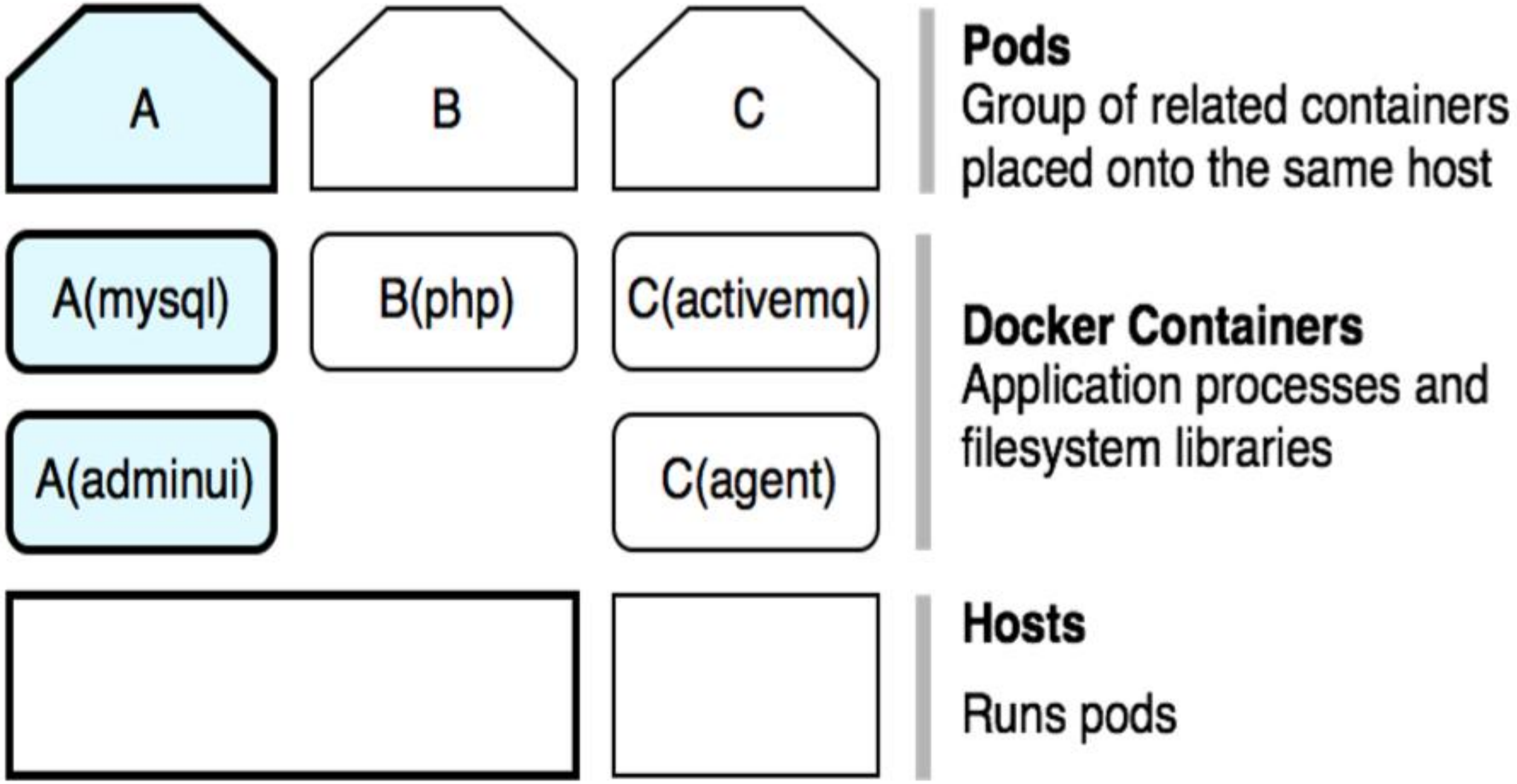
Pod

- Single schedulable unit of work
  - Can not move between machines
  - Can not span machines
- One or more containers
  - Shared network namespace
- Metadata about the container(s)
- Env vars – configuration for the container
- Every pod gets an unique IP
  - Assigned by the container engine, not kube!

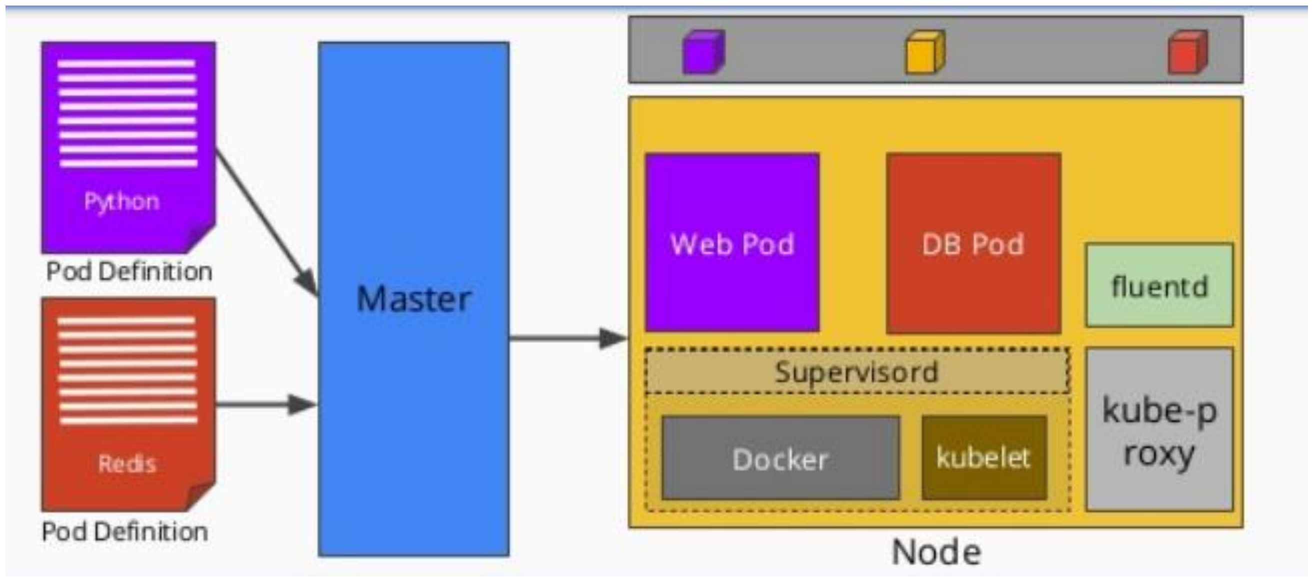
Pod


Pod
Container
Image
Env Vars
Mounts
Container
Image
Env Vars
Mounts
Mount Sources
Labels
Annotations
IP Address

POD (CONTINUED)


A
B
C
A(mysql)
B(php)
C(activemq)
A(adminui)
C(agent)
Pods
Group of related containers placed onto the same host
Docker Containers
Application processes and filesystem libraries
Hosts
Runs pods

Deploying pod



Node Object Definition

The following is an example node object definition in Kubernetes:

```
apiVersion: v1
kind: Node
metadata:
  creationTimestamp: null
  labels:
    kubernetes.io/hostname: node1.example.com
  name: node1.example.com
spec:
  externalID: node1.example.com
status:
  nodeInfo:
    bootID: ""
    containerRuntimeVersion: ""
    kernelVersion: ""
    kubeProxyVersion: ""
    kubeletVersion: ""
    machineID: ""
    osImage: ""
    systemUUID: ""
```

apiVersion defines the API version to use.
kind set to Node identifies this as a definition for a node object.
metadata.labels lists any labels that have been added to the node.
metadata.name is a required value that defines the name of the node object. This value is shown in the NAME column when running the oc get nodes command.
spec.externalID defines the fully-qualified domain name where the node can be reached. Defaults to the metadata.name value when empty.

Demo Pods

- How to create a pods