RUBY IN CONTAINERS

Rubyconf Kenya 2018

About me : Joannah Nanjekye

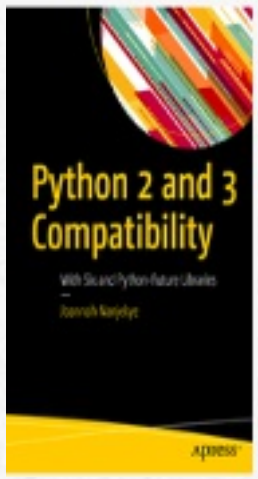
Software Engineer, Aeronautical Engineer to Be !!!

I contribute to :

OUTREACHY

Rails Girls Summer of Code

Author of :



@Captain_Joannah

Deploying Ruby Applications in 2018

- Physical machine
- Virtual Machines
- Containers

@Captain_Joannah

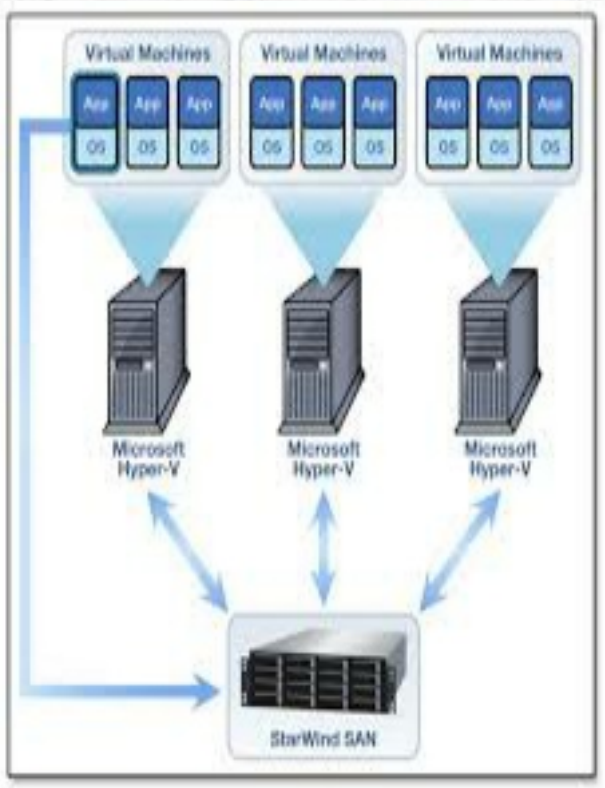
Physical machine

- Typically one application per host or physical machine.

@Captain_Joannah

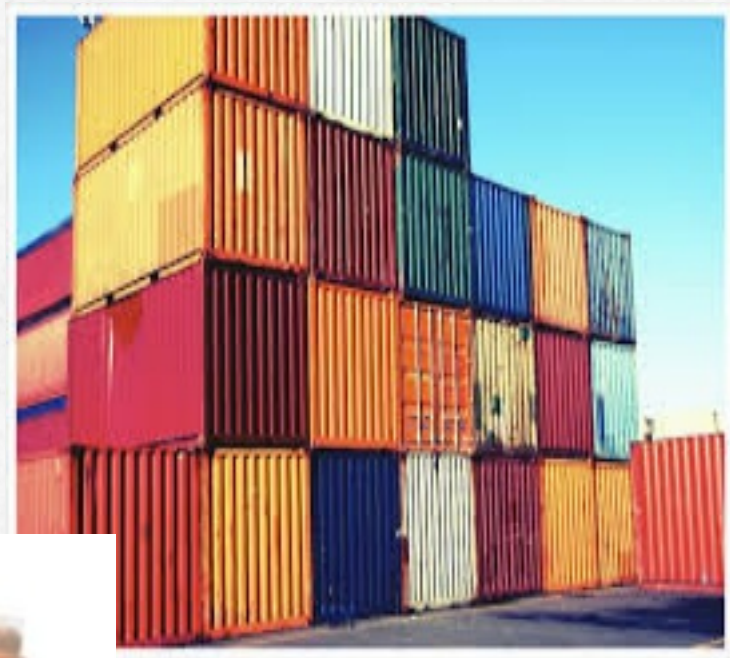
Virtual Machine

- Virtualize the hardware.
- Use same hardware to run multiple applications.



@Captain_Joannah

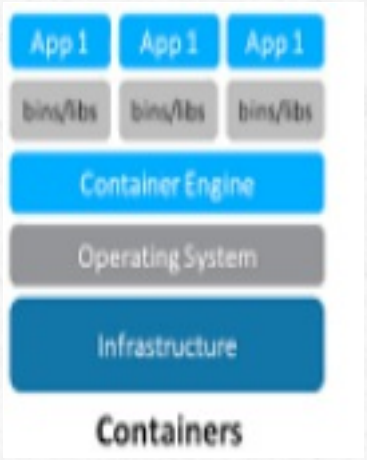
Containers



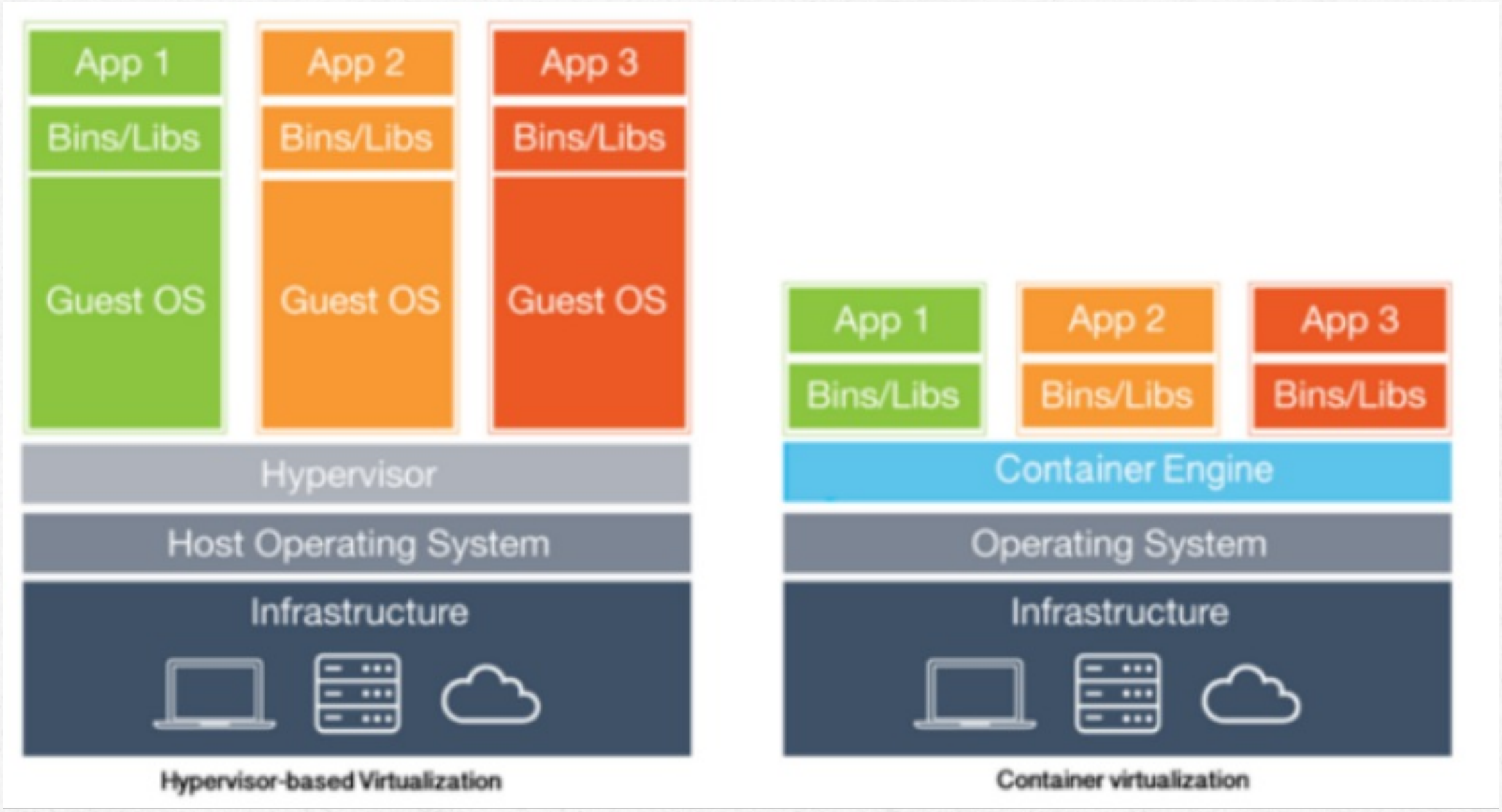
@Captain_Joannah

What are Containers

- Sandbox for application processes.
- logical packaging for applications to run abstracted away from the environment.



Containers Vs Virtual Machines



@Captain_Joannah

Benefits of containers

- Portability
- Improvement in software delivery through the image format
- Scaling
- Clustering.

How do we create containerized apps with Docker

- Use a container technology like Docker.

@Captain_Joannah

Live Demo: Show me the code

@Captain_Joannah

Docker image

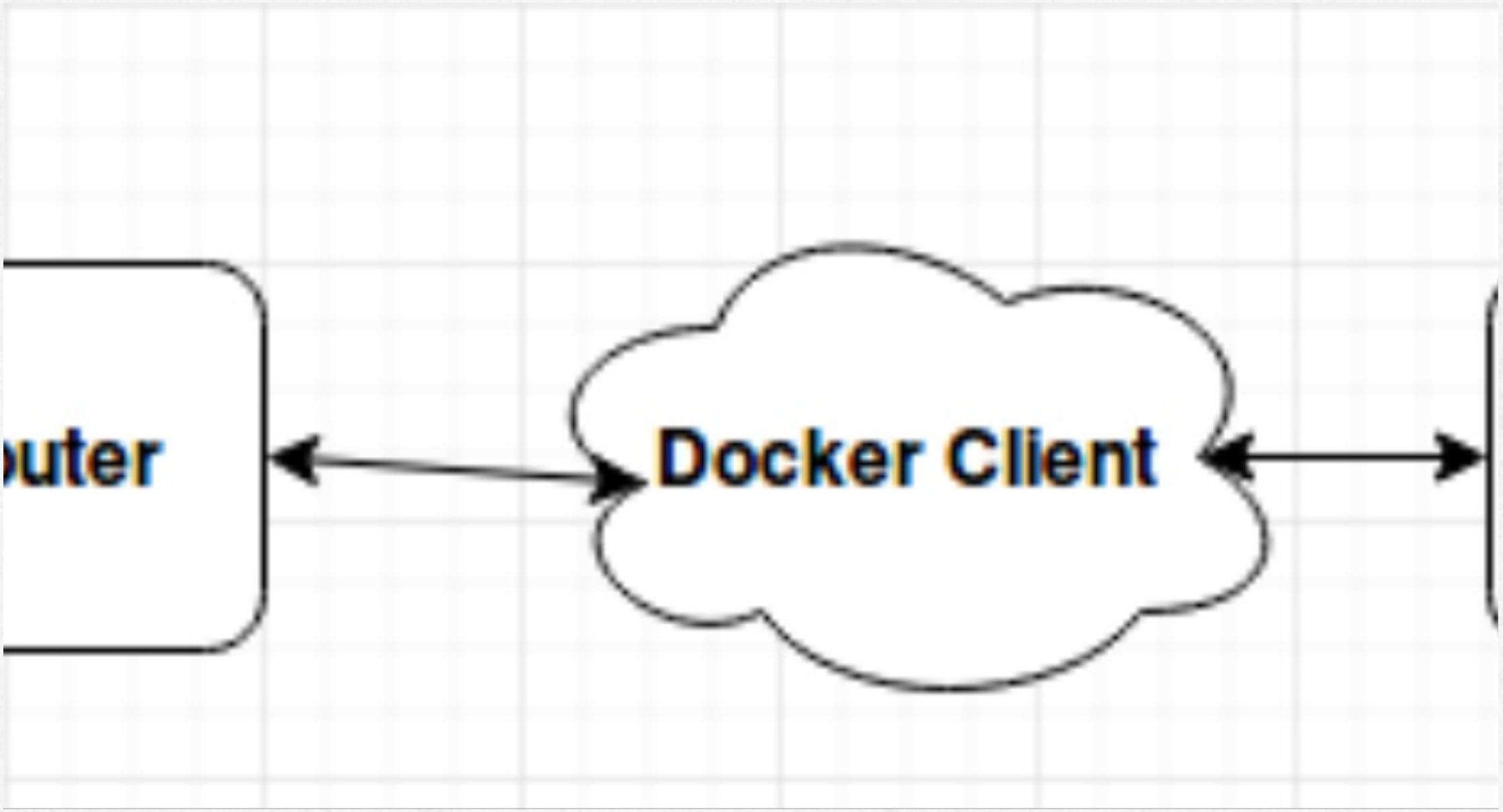
build the image

```
docker build -t app .
```

Run the image

```
docker run app
```

How docker works



When to use containers

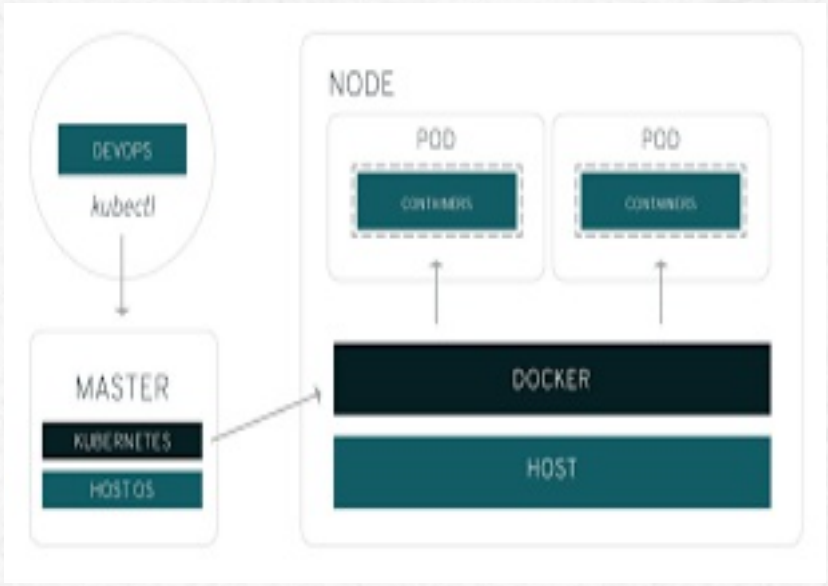
- Microservices
- Portability
- Continuous Integration and Deployment
- Batch Jobs

The last mile : Kubernetes

@Captain_Joannah

What is kubernetes

- Kubernetes is a container orchestrator



@Captain_Joannah

kubernetes helps us do:

- Deployments
- Monitoring
- Scaling

Deployments

- In a deployment we specify;
  - The image from which to create a container
  - The deployment criteria i.e CPU, RAM, storage

Monitoring

- Health check
- Autorecovery

@Captain_Joannah

Scaling

- Automatically scales your containers on available hosts.

Terminologies

- Pod : A pod is a group of containers.
  - Containers share environment and resources
  - Deployed as single application
- Node : This is a worker machine which can be a virtual machine or physical machine.
- A volume : A volume is generally a storage facility which can be in a file or on a local disk.

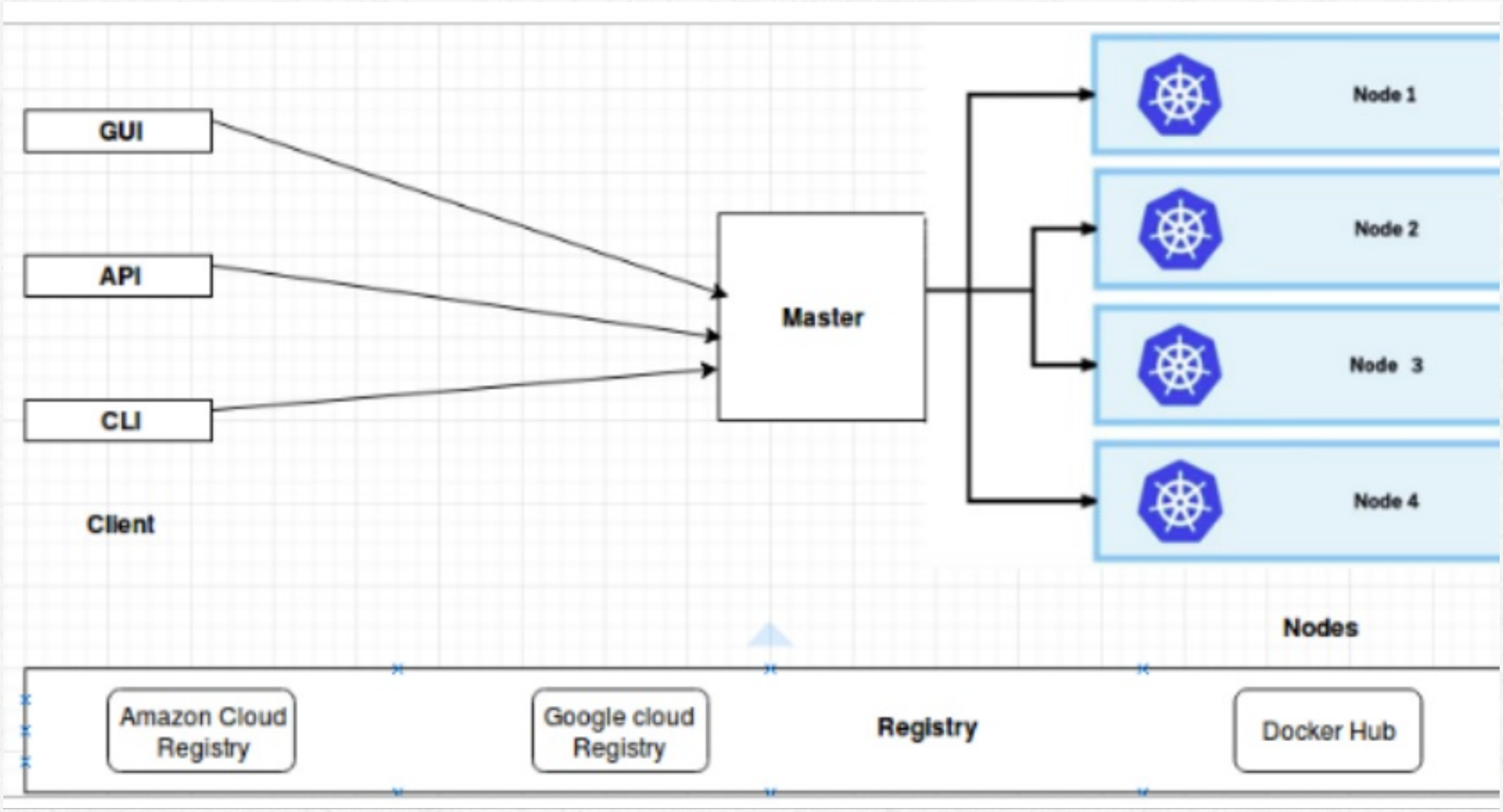
Terminologies

- Service : A service groups related containers into a single entity and provides a single access to them by external application.
- Cluster : A cluster is a set consisting of a master node and other worker nodes.
- Namespaces: These are multiple virtual clusters backed by a single physical or real cluster .

Kubernetes Architecture

- client
- Master
- Nodes

K8s Architecture

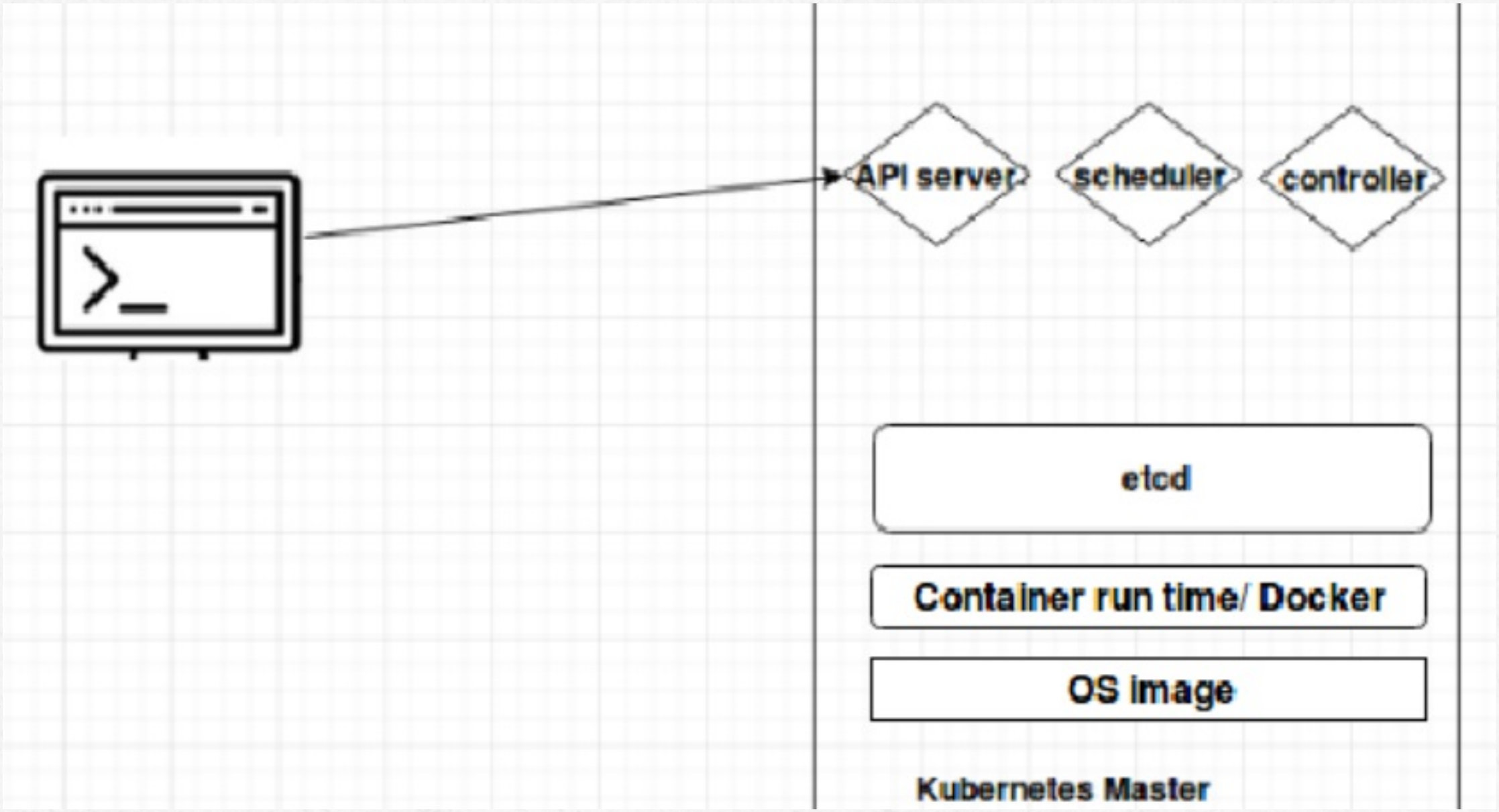


@Captain_Joannah

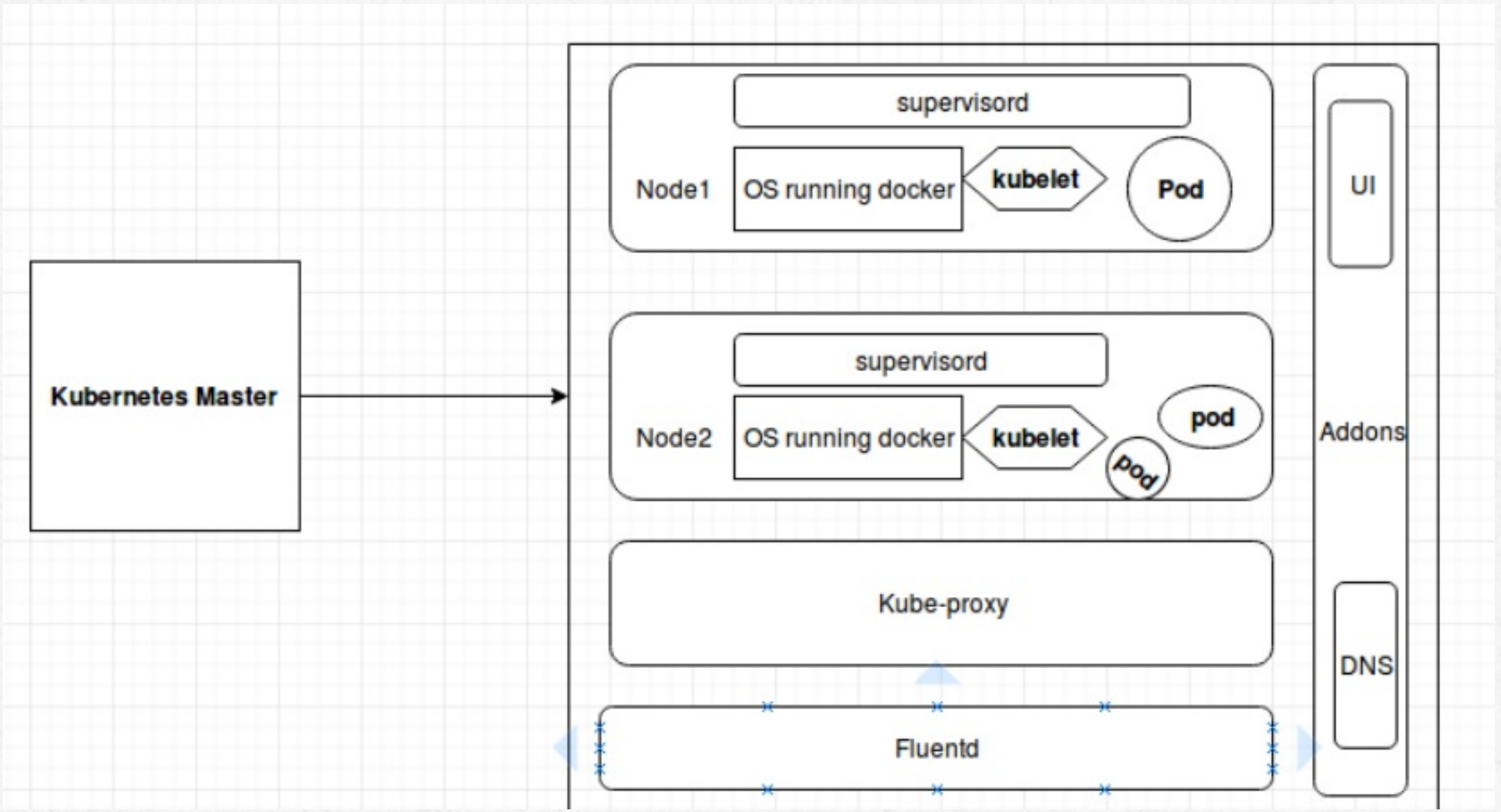
Client

- Makes API calls to the master.

Master


API server
scheduler
controller
etcd
Container run time/ Docker
OS image
Kubernetes Master

Nodes



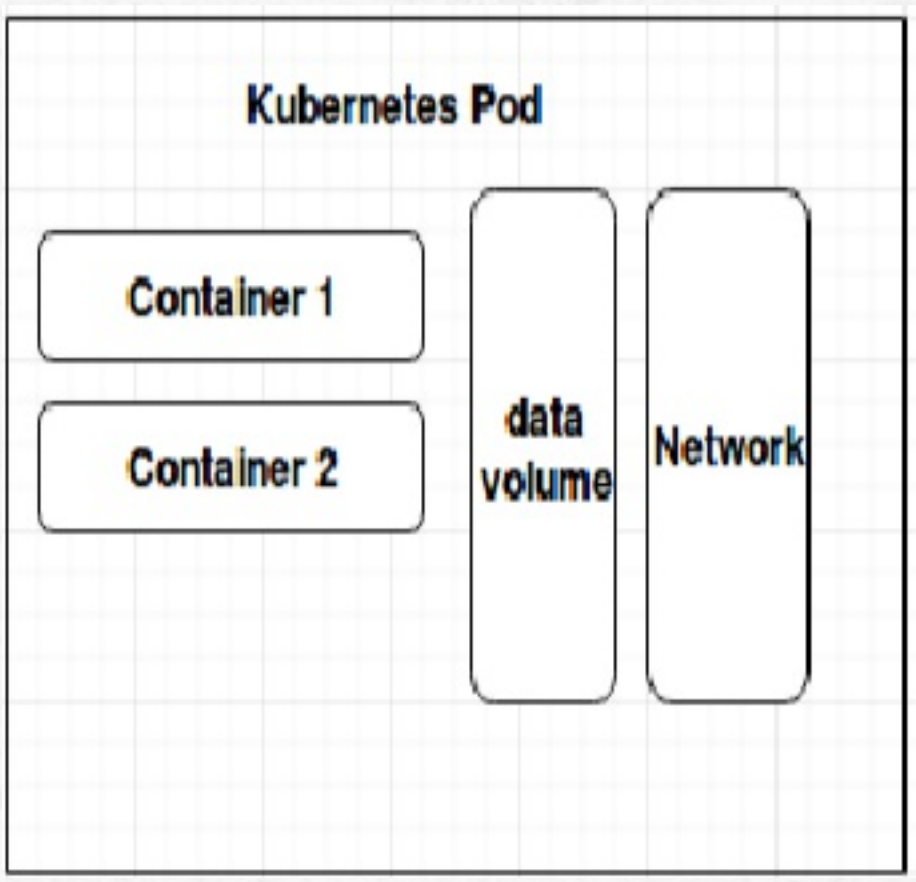
@Captain_Joannah

what Kubernetes can do for your Ruby Apps

@Captain_Joannah

Grouping of containers in pods

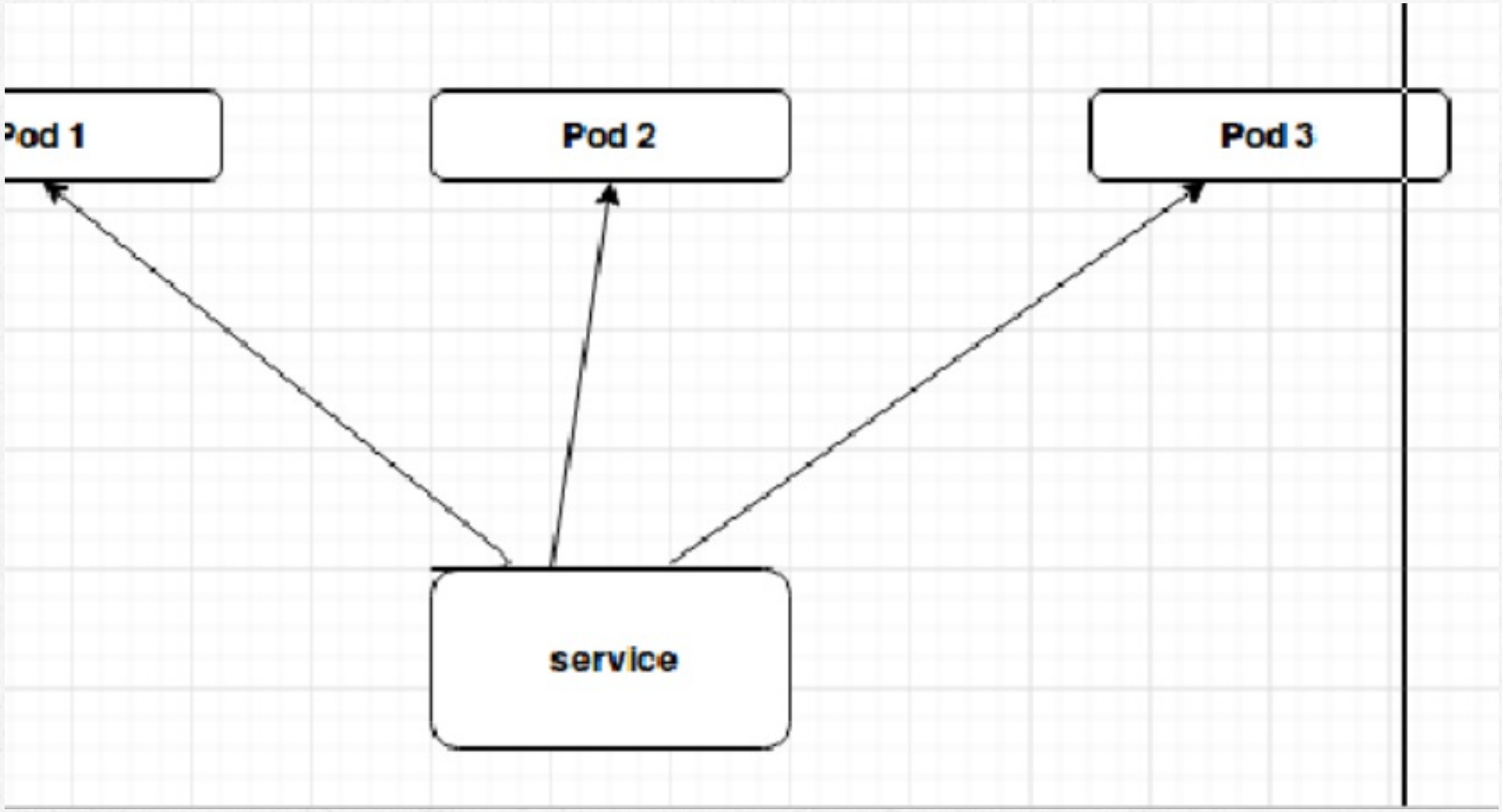
- Group containers in small deployable units.



Networking for your containers

- container- to-container communication through pods localhost communications
- pod-to-pod communication through the the software defined networking and pod and external communication achieved through services.

Services



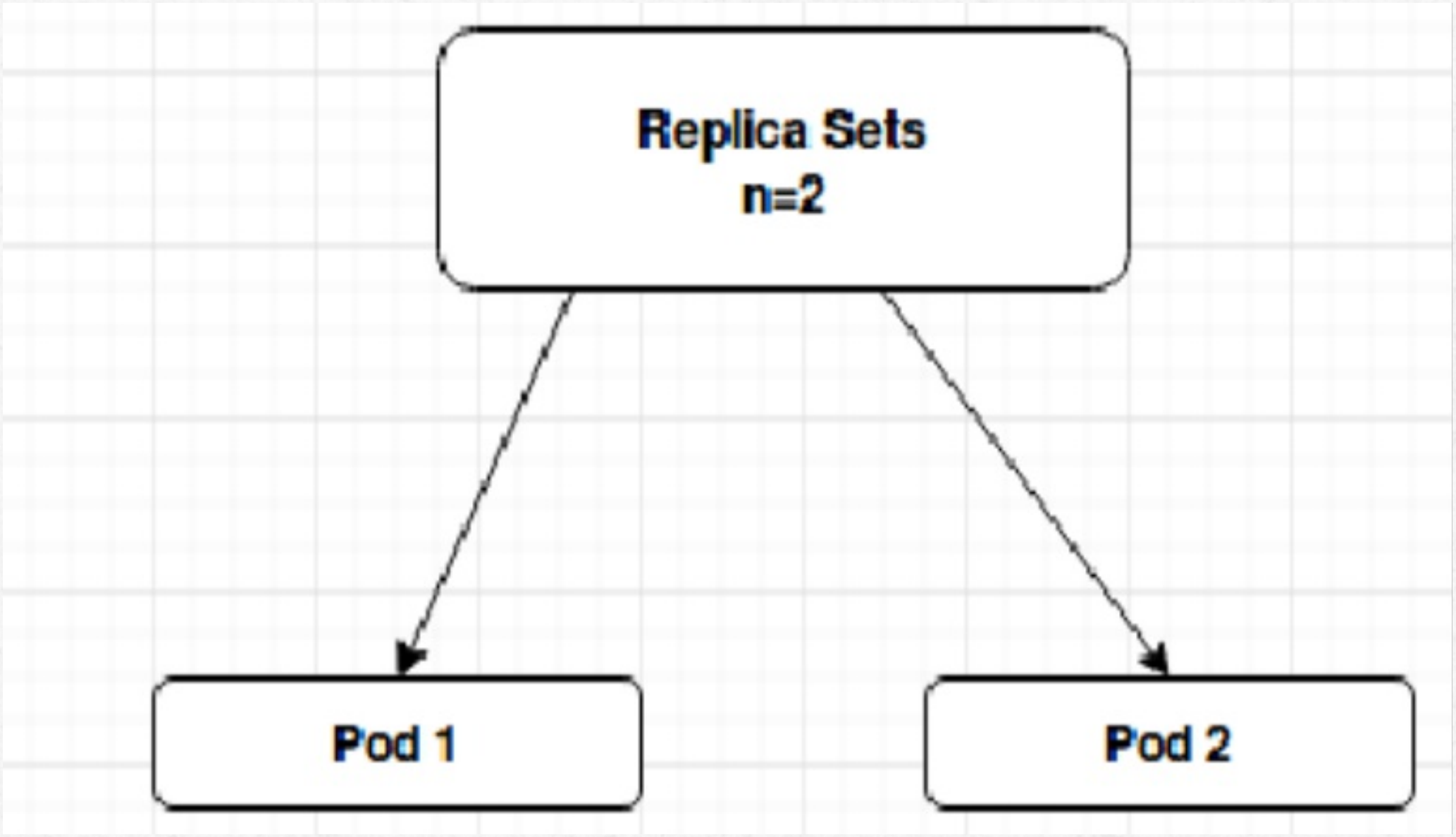
Service Discovery

- Kubernetes provides DNS based service discovery.
- Tracks new services in clusters and creates a set of DNS records for each.

Ingress

- Map URLs to services for use by external world.
- public cloud - ingress provider
- private cloud - nginx controller

Control in replication



Persistent Storage

- When a pod or container is killed or rescheduled, all data is lost.
- kubernetes allows you to make use of persistent systems like Amazon Elastic Block Storage (EBS) or Google Compute Engine Persistent Disks (GCE PD) or a distributed file system such as the Network File System (NFS) or the Gluster File System (GFS)

Monitoring

- heapster - collects metrics related to CPU, memory, I/O, and network stats of all running containers.

Checking health of the nodes

- kubelet is responsible for checking the health of nodes in a cluster.

Automatic Scaling

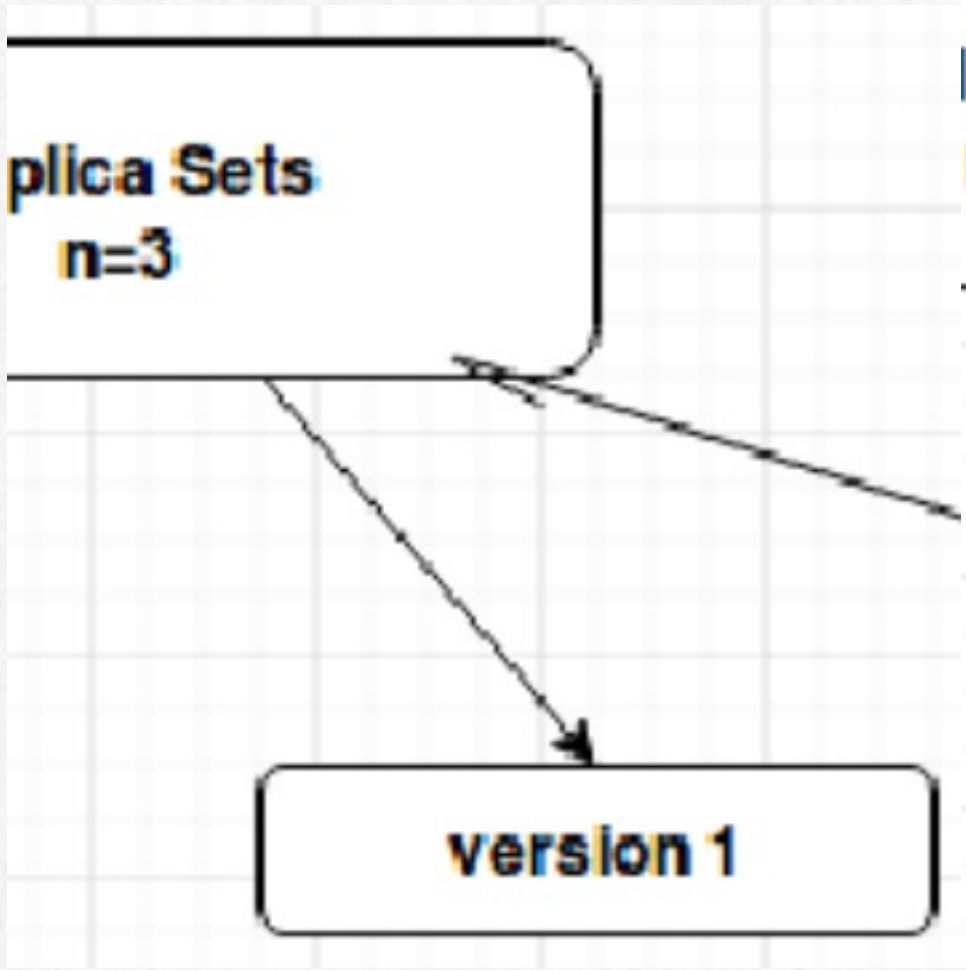
- Horizontal Pod scaling: This is where the replicasets are scaled depending the CPU usage of the pods or specified application metrics.
- Cluster Autoscaling: This is where the number of nodes are scaled in a cluster depending on CPU usage and memory and if it is the cloud then it depends on the cloud provider.

Secrets and configmaps

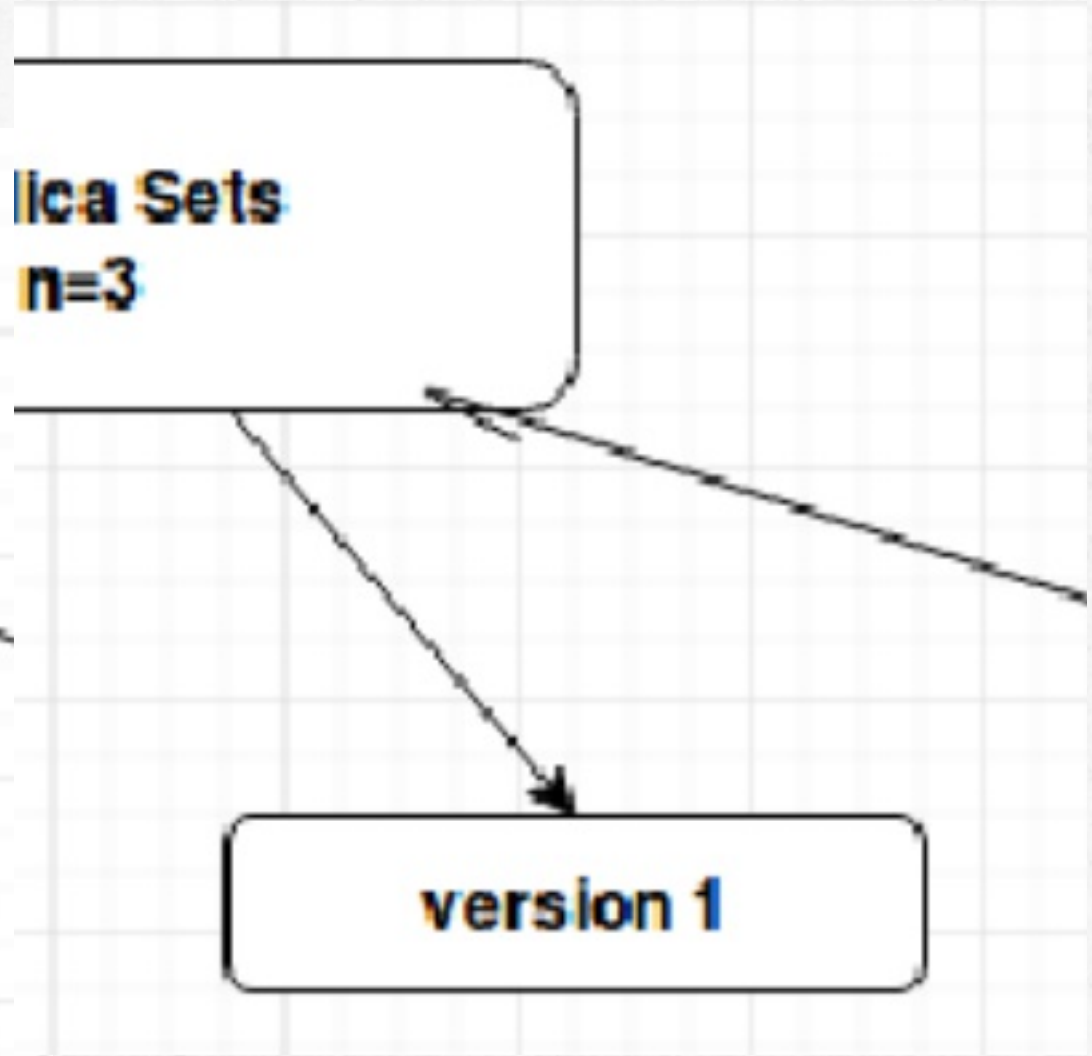
- ConfigMap allow us to inject a configuration depending on the environment keeping the container image identical across multiple environments.
- Secrets allow you to store sensitive data such as passwords, OAuth tokens,used by your applications.

Rolling Updates

Before new deployment



After new deployment



API

- All kubernetes functionality is provided through APIs making it easy to integrate into your existing workflows.

Logging

- kubernetes through daemonsets allows you to do logging and general monitoring.

Deploying to kubernetes

Tag the images we built.

```
docker tag app:latest app:0.1
```

Push the images to the registry

```
docker push app
```

Live Demo: Deploying to k8s

@Captain_Joannah

Summary

- Containers logically package your application abstracting them from your environment.
- Kubernetes helps us automate most of the container related operations like deployment and scaling.

@Captain_Joannah

Inspiration

I have written a Blog :

Running a Python Application on Kubernetes - Published on Apress and opensource.com

Sample Code: nanjekyejoannah/k8s_python_sample_code

@Captain_Joannah