Container 101

How we use kubernetes at RunEffective

About RunEffective

RunEffective helps companies reinvent their workplace.

Company Background

- HQ in Mountain View CA
- Founded - 1/2016

RunEffective

RunEffective Solution

A workplace applications platform:

1. Suite of workplace products with modern consumer experience
2. Powerful central workplace database (system of record)

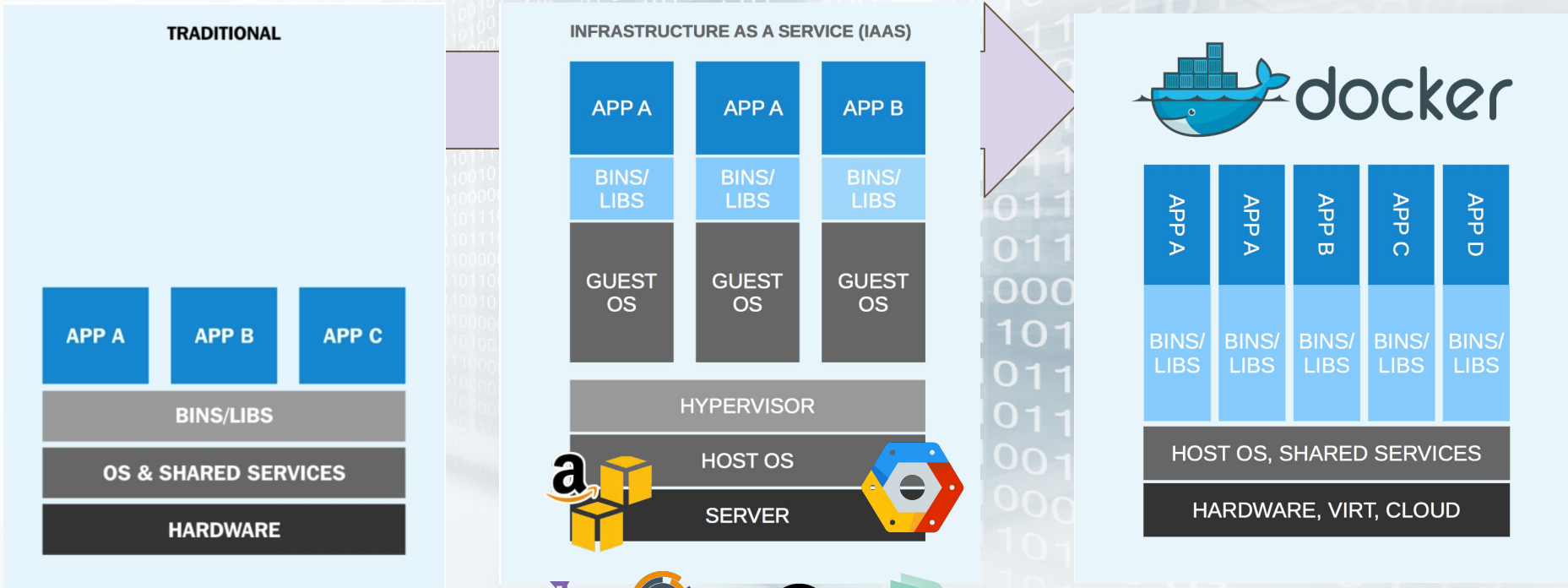
Customer Value

Better productivity and savings for existing workplace activities

Centralized control to drive long-term strategic decisions

RunEffective architecture - MIcroservice

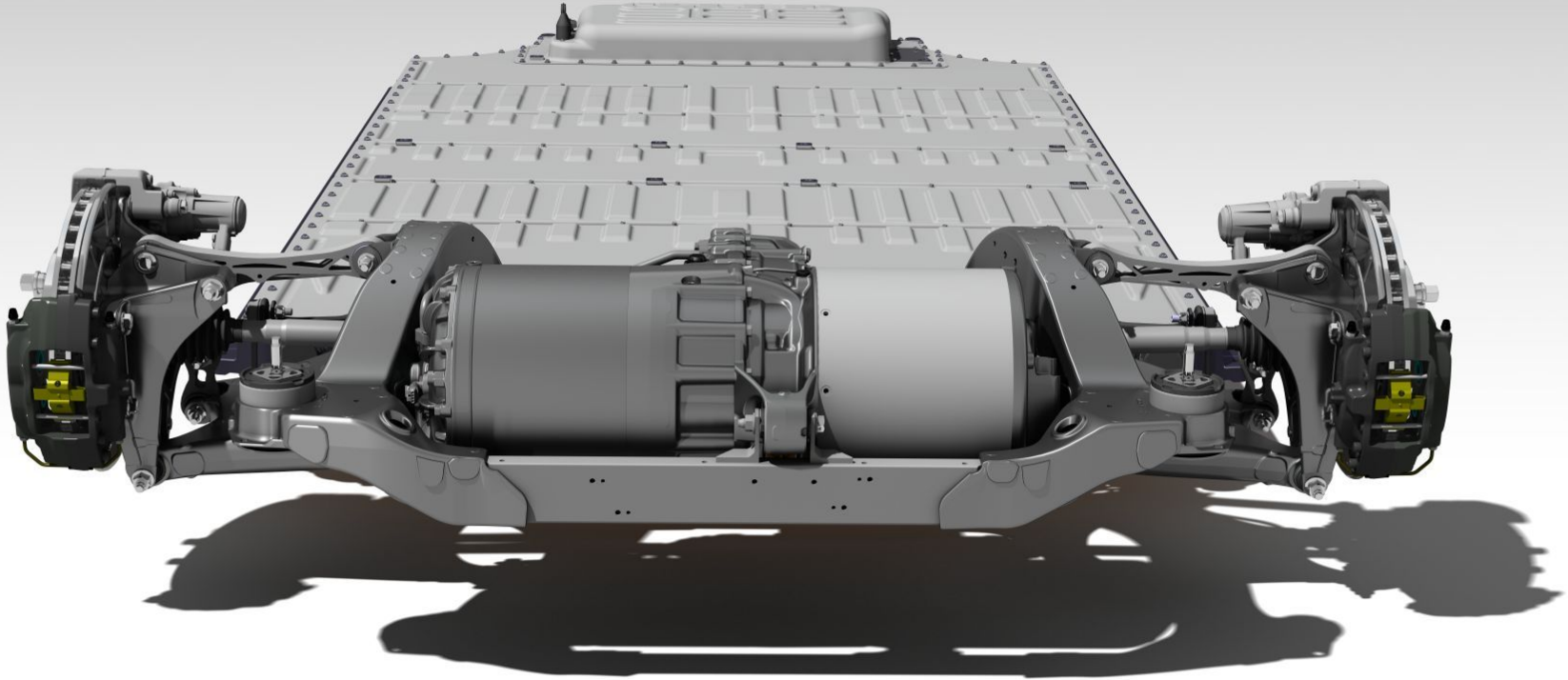
Infrastructure -the path to docker



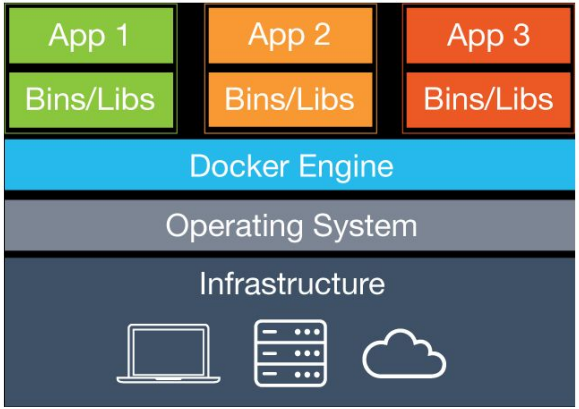
puppet labs CHEF ANSIBLE

Modern Car

TESLA

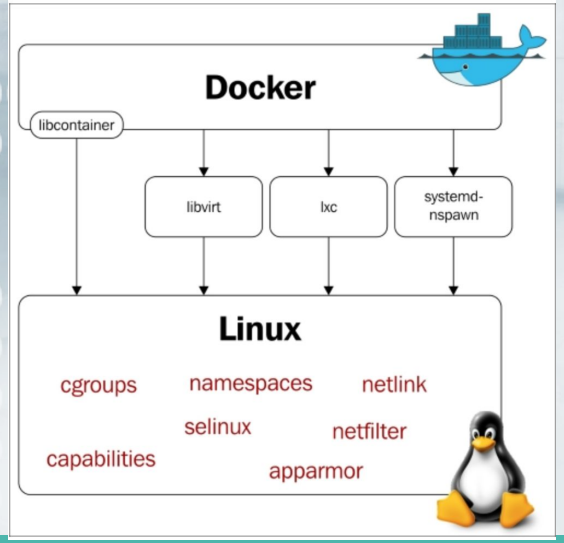


Docker?



Docker: Application centric.

A clean, safe, portable runtime environment for your app.
No more worries about missing dependencies, packages and other pain points during deployments.
Run each app in its own isolated container (fs , cgroup , pid etc ….)
Easy to pack into a box and super portable.



Build once… run anywhere


RENA
REUTERS

Challenges with Docker

- Only a containers engine
  - Single host
  - Poor Networking
- Deployment

Clustering solutions

- Swarm, Compose
- ECS
- Mesos

kubernetes

Google's Open Docker Orchestration Engine

Kubernetes ?

Kubernetes is a container cluster manager. It aims to provide a "platform for automating deployment, scaling, and operations of application containers across clusters of machines.

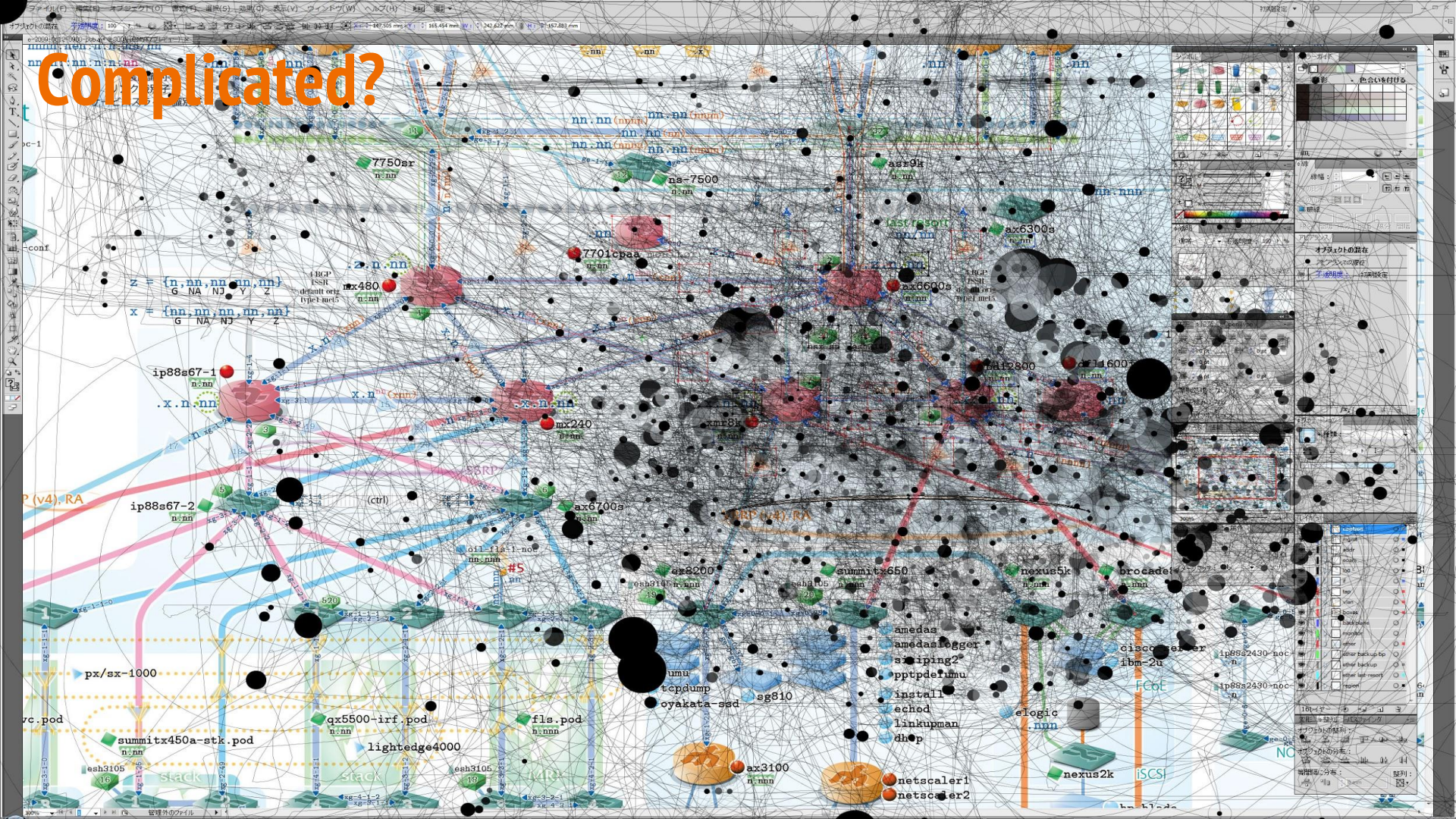
More facts:

- **Originated at Google (Borg).**
- **Supports multiple cloud and bare-metal environments**
- **Supports multiple container runtimes (Docker , rkt)**
- **100% Open source, written in Go**
- **k8s is an abbreviation derived by replacing the 8 letters “ubernete” with 8.**

Manage containerized applications , not machines.

Why k8s

- IaaS as a service
- Host Independent
- Multi environment (Configmap,Secrets)
- Multi Tenancy (Namespaces)
- Built-in “no downtime” deployment
- Active community (ecosystem)
- Simple implementation https://github.com/coreos/kube-aws

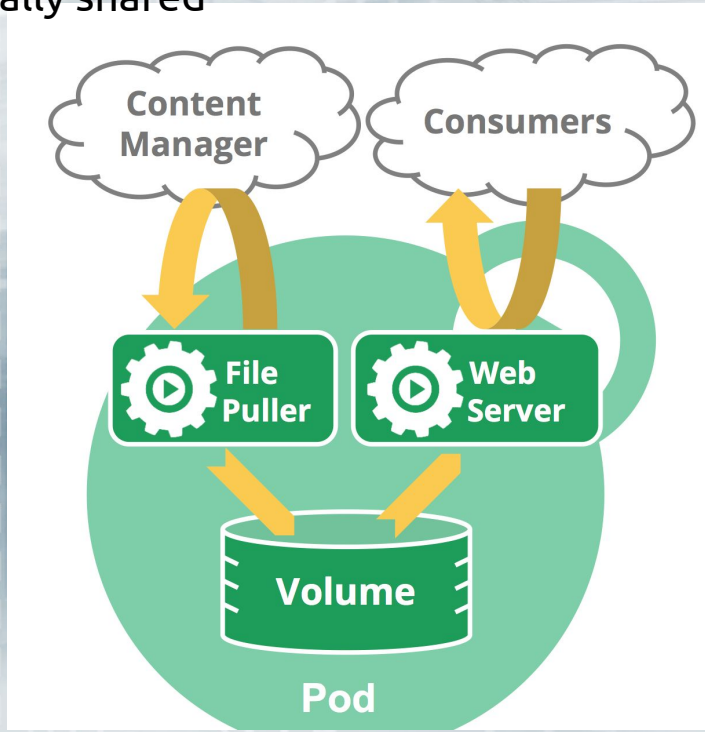

Complicated?
7750sr
ns-7500
asr9k
ax6300s
7701cpaa
mx480
ax6600s
ip88s67-1
mx240
ip88s67-2
ax6700s
ax8200
summitx650
nexus5k
brocade
px/sx-1000
qx5500-irf.pod
fls.pod
summitx450a-stk.pod
lightedge4000
ax3100
netscaler1
netscaler2
nexus2k
amedas
amedaslogger
sniping2
pptpdetumu
install
echod
linkupman
dhcp
tcpdump
oyakata-ssd
sg810
elogic
ibm-2u
FCoE
iSCSI
stack
VRRP (v4), RA

Pod(po)

Pod is a Small group of **co-located** containers with optionally shared volume between the containers.

Pods are the basic deployment unit in Kubernetes.

- Shared namespace
  - Share IP address , localhost
  - Every pod gets a unique IP
- Managed Lifecycle
  - Bound to a node , in place restart
  - Cannot move between nodes

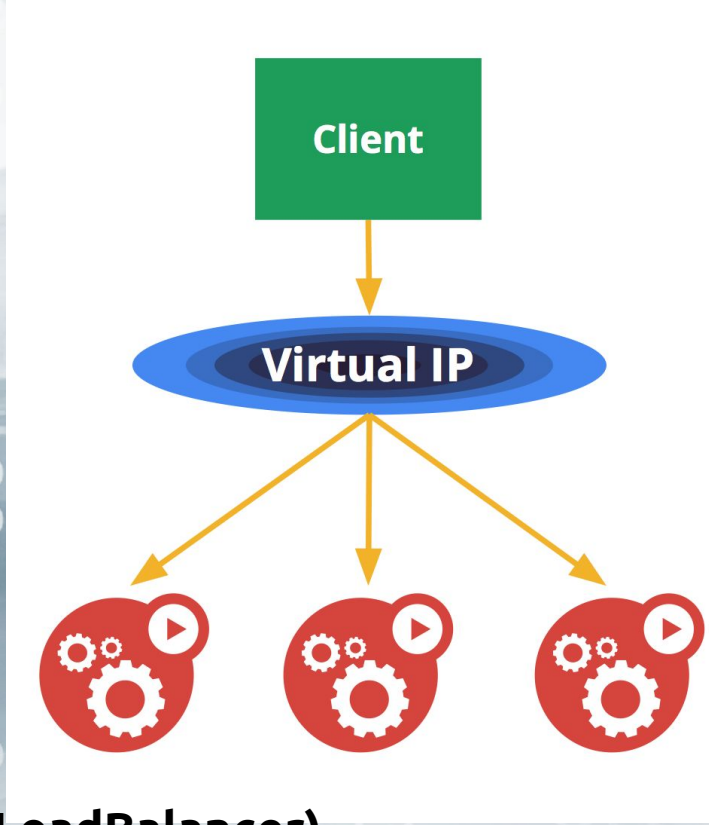


services (svc)

direct traffic to pods

Defines a logical set of pods and a policy by which to access them.

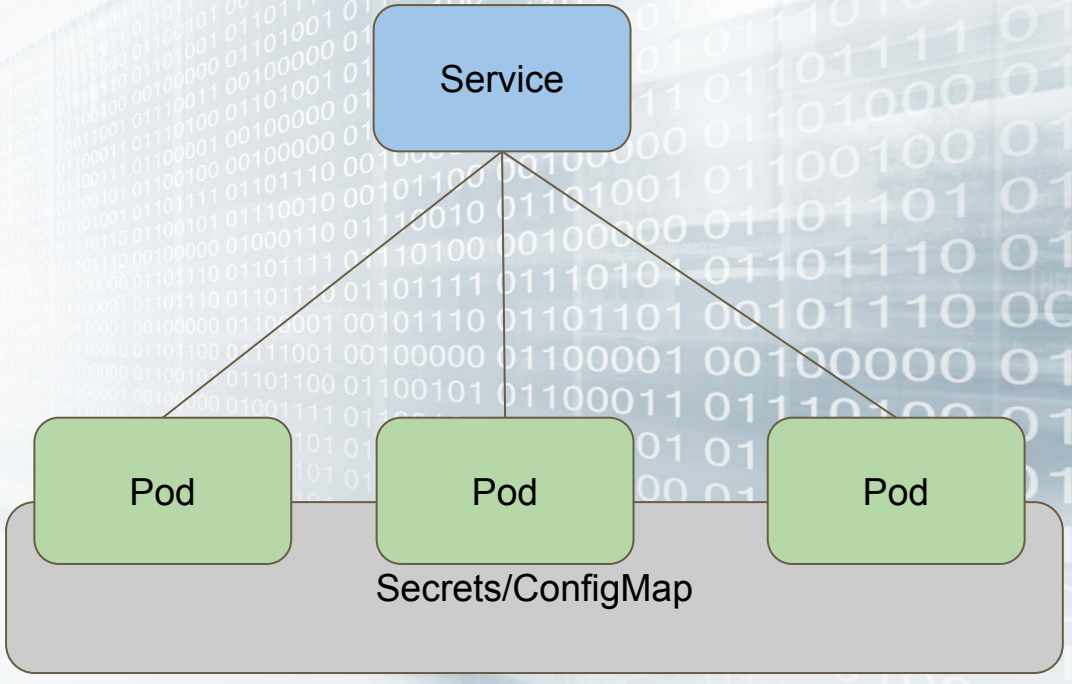
- are abstraction on top of the pods (LB)
- use selector to create the logical set of pods.
- Gets a stable virtual IP and Port.
- Cluster IP are only available **inside** k8s



Can define:

- What the 'internal' IP should be.(**ClusterIP)**
- What the 'external' IP should be. **(NodePort , LoadBalancer)**
- What port the service should listen on.

Kube 101



replication controller (rc) \ ReplicaSet (rs)

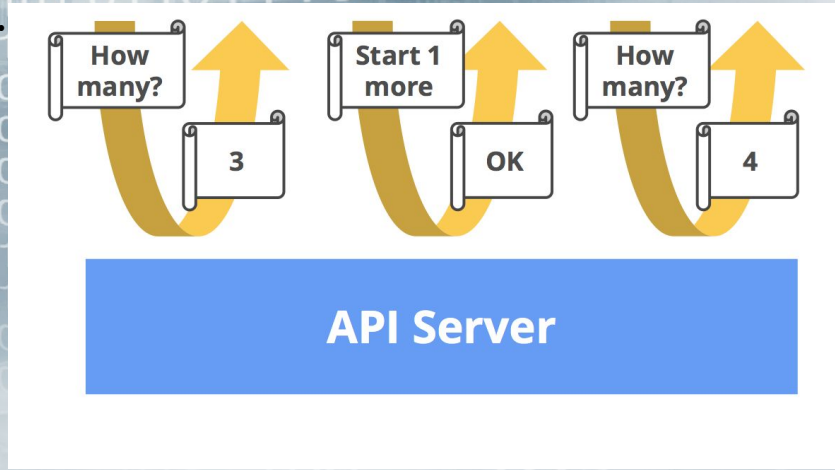
Replication controller == pods supervisor

Ensures that a specified number of pod "replicas" are running at any given time:

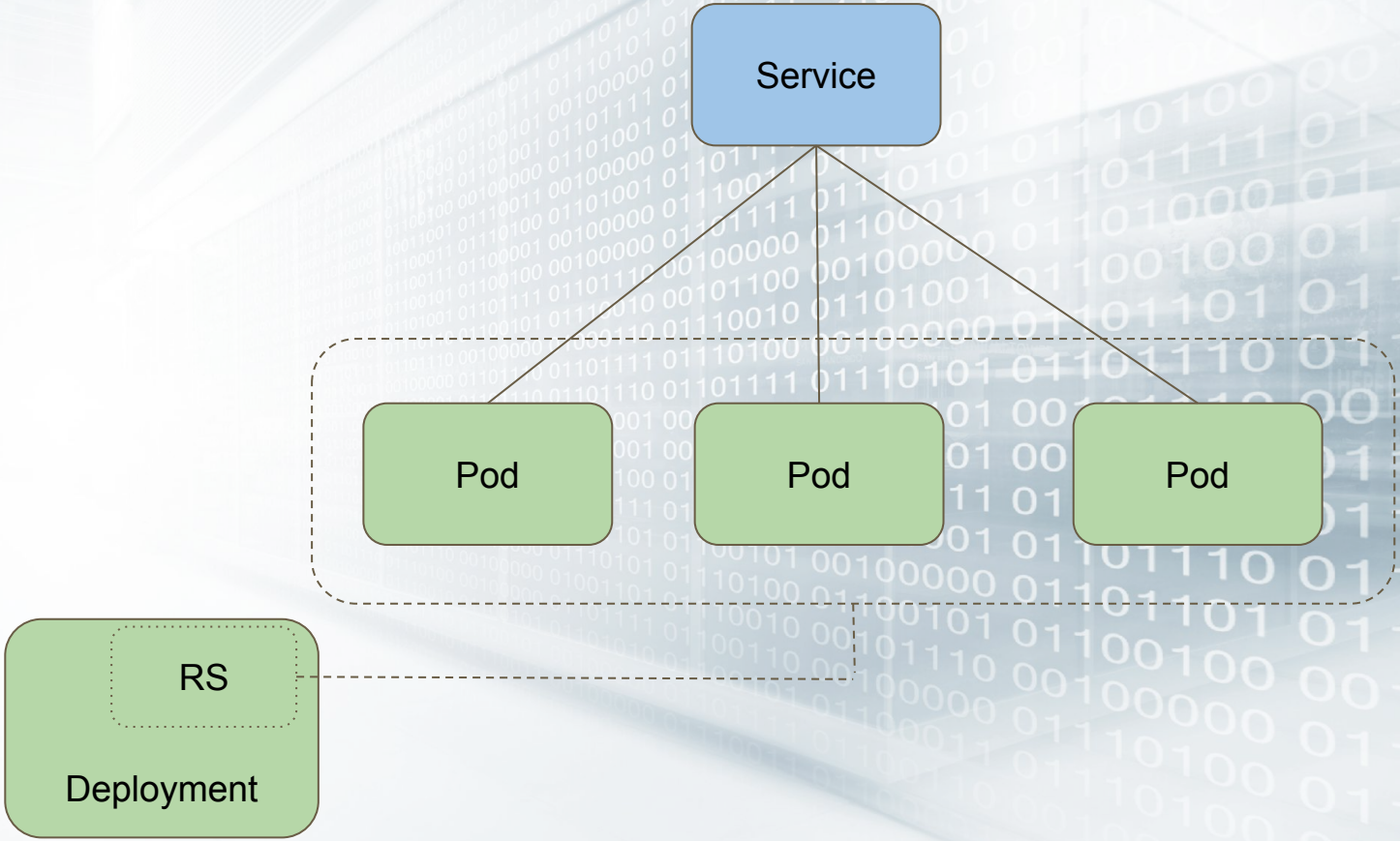
- Too many pods will trigger pods termination.
- Too few pods will trigger new pods creation.
- Main Goal = Replicas: x current / x desired.

replication controller will monitor all the pods defined in the label selector.

```
ReplicationController
replica: 4
name:  rcweb
selector:
  app: rcweb
  phase: production
```



Kube 102



Namespaces

Multi tenancy in kubernetes.

- A single cluster should be able to satisfy the needs of multiple users or groups of users.

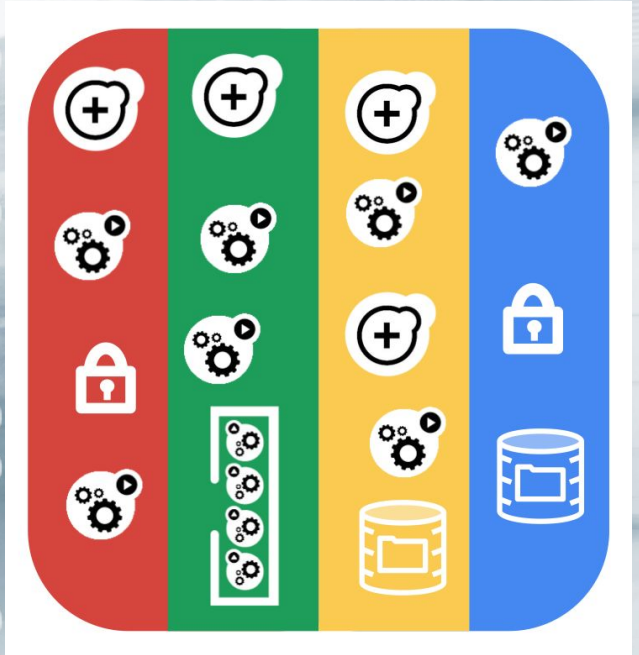
Each user community has its own:

1. resources (pods, services, replication controllers, etc.)
2. policies (who can or cannot perform actions in their community)
3. constraints (this community is allowed this much quota, etc.)

Kubernetes starts with two initial namespaces:

default - The default namespace for objects with no other namespace.

kube-system - The namespace for objects created by the Kubernetes system



K8s at Runeffective

- 2 clusters (Prod/None prod)
- Namespaces
- Secrets for multi env
- home grown tool “Kubit” to deploy our microservices

Kubit

- Deployment
- Templating
- Deployment Verification
- Managing DNS
- Promote and clone environments
- Rollback
- Pre build tasks
- Slack/Datadog integration

Demo