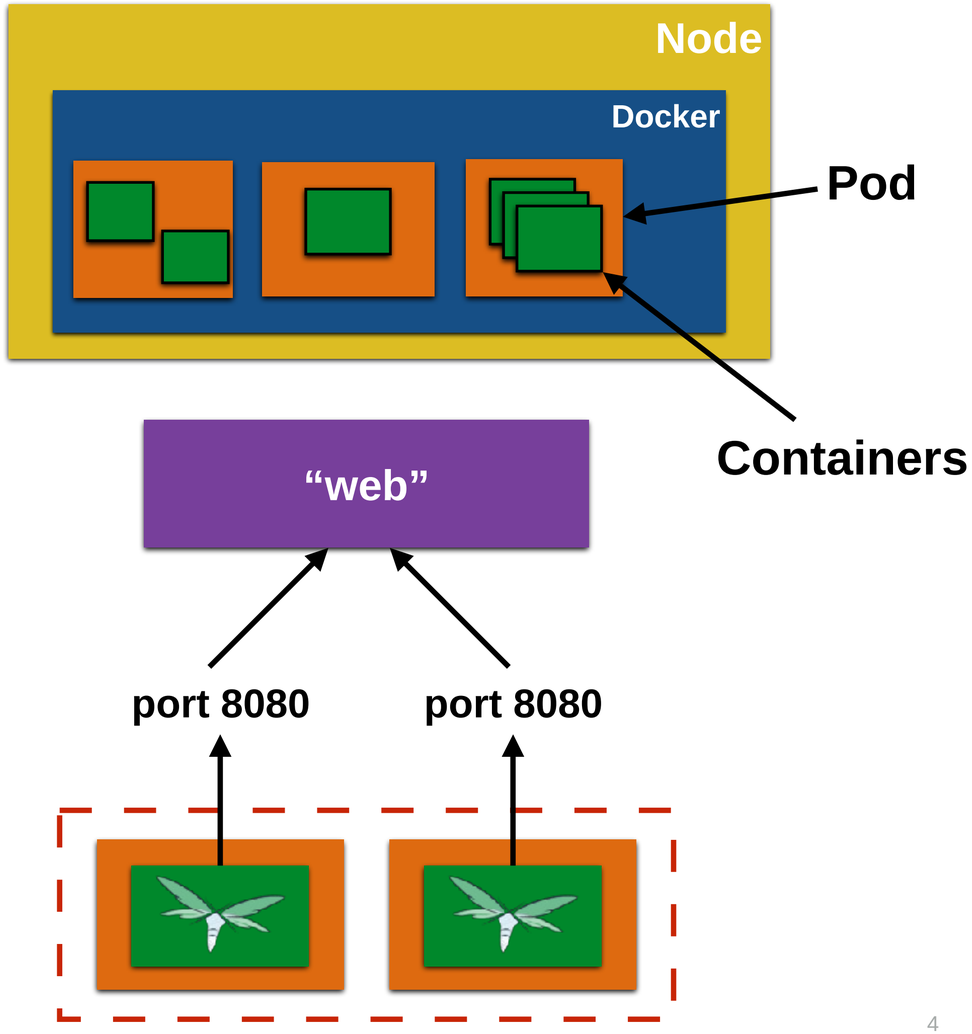
Kubernetes

- Open source orchestration system for containers
  - Docker, rkt, OCI, …
- Started by Google, donated to CNCF
- Provide declarative primitives for the “desired state”
  - Self-healing
  - Horizontal scaling
  - Automatic binpacking
  - Service discovery and load balancing

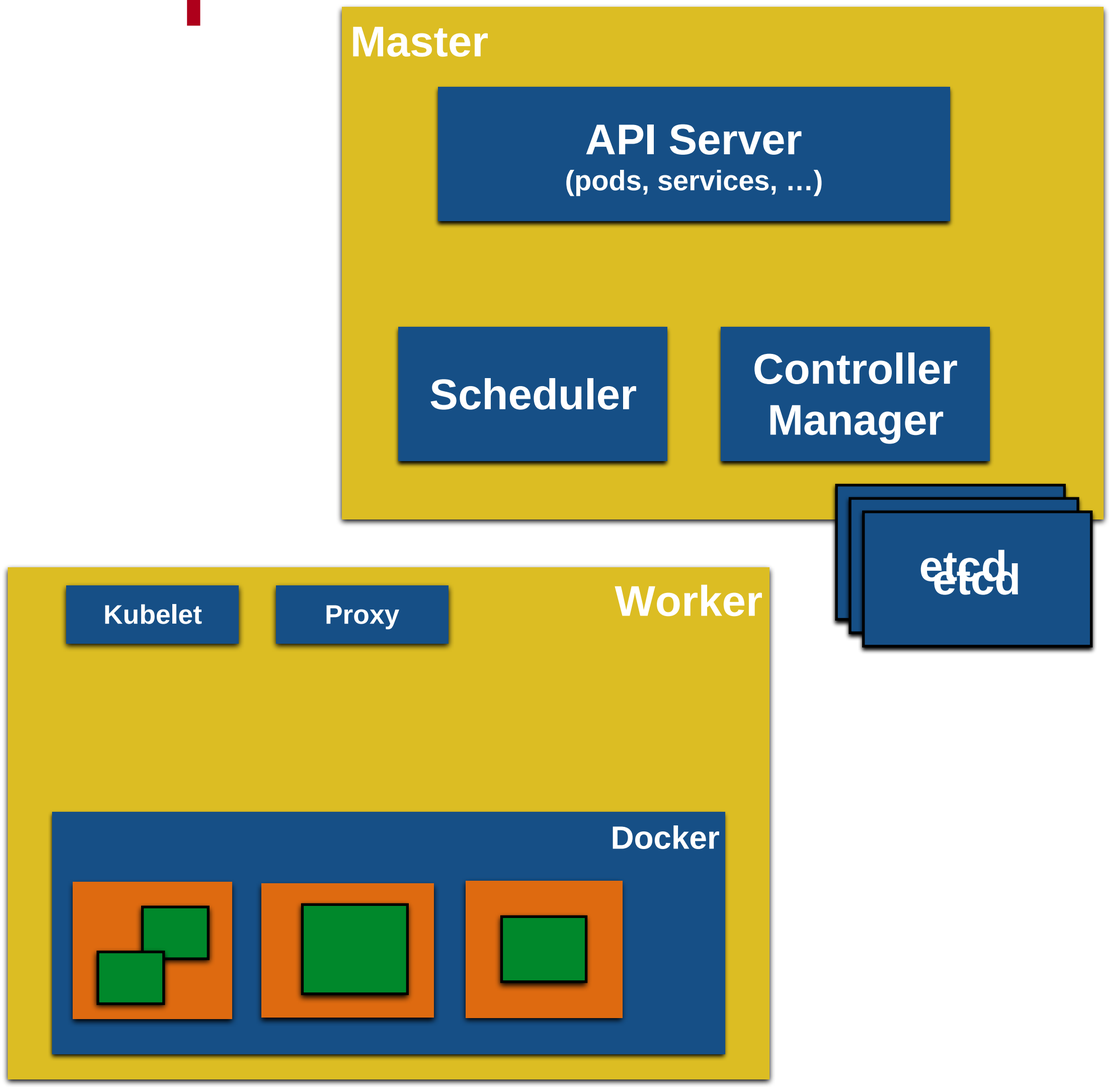
Kubernetes Concepts

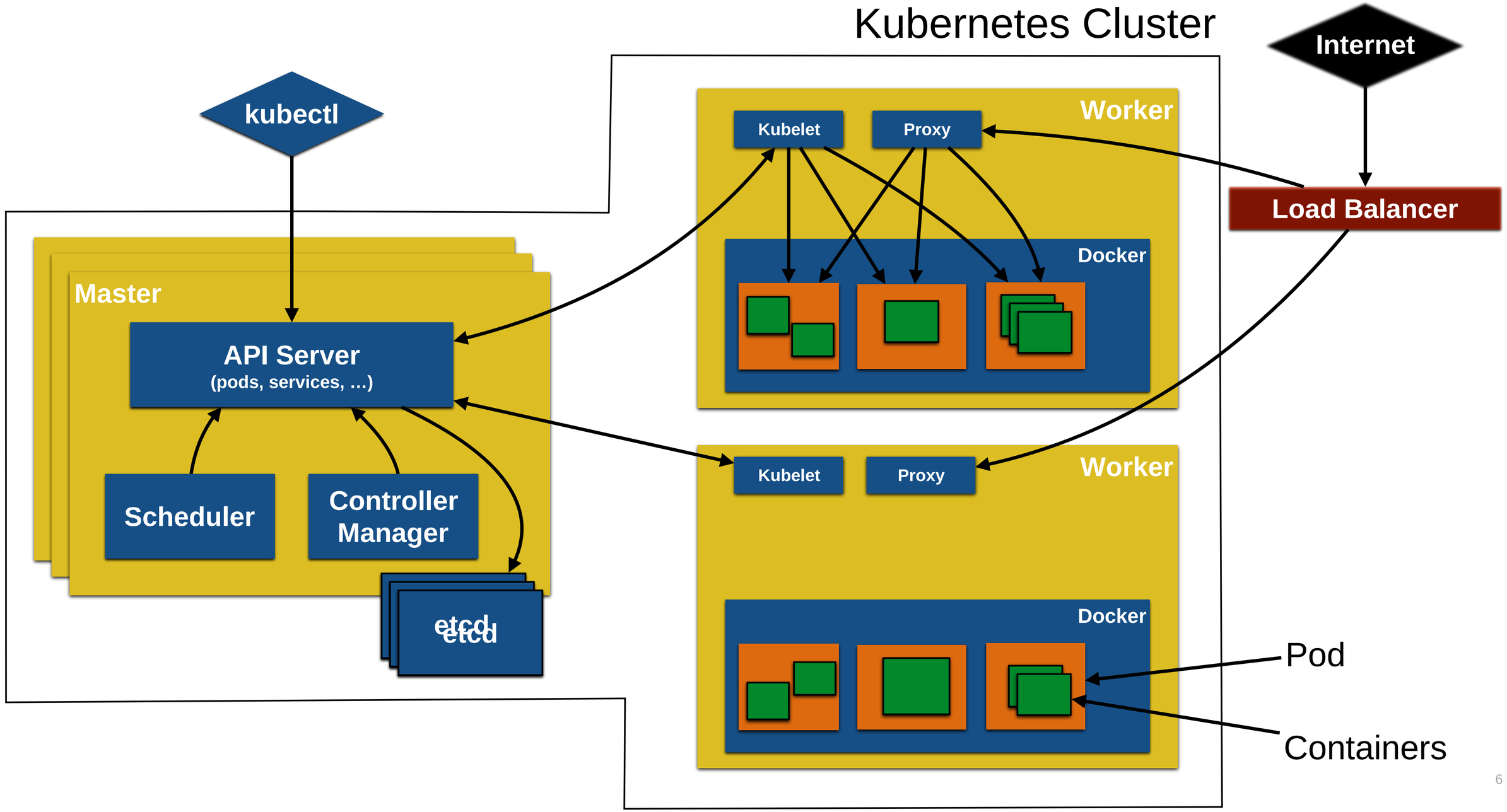
- **Pods**: colocated group of containers that share an IP, namespace, storage volume
- **Replica Set**: manages the lifecycle of pods and ensures specified number are running (next gen Replication Controller)
- **Service**: Single, stable name for a set of pods, also acts as LB
- **Label**: used to organize and select group of objects



Kubernetes Components

- **Node**: Machine or VM in the cluster
- **Master**: Central control plane, provides unified view of the cluster
  - **etcd**: distributed key-value store used to persist Kubernetes system state
- **Worker**: Docker host running kubelet (node agent) and proxy services
  - Runs pods and containers
  - Monitored by systemd (CentOS) or monit (Debian)




Kubernetes Cluster
Internet
kubectl
Worker
Kubelet
Proxy
Load Balancer
Docker
Master
API Server
(pods, services, ...)
Scheduler
Controller Manager
etcd
Worker
Kubelet
Proxy
Docker
Pod
Containers

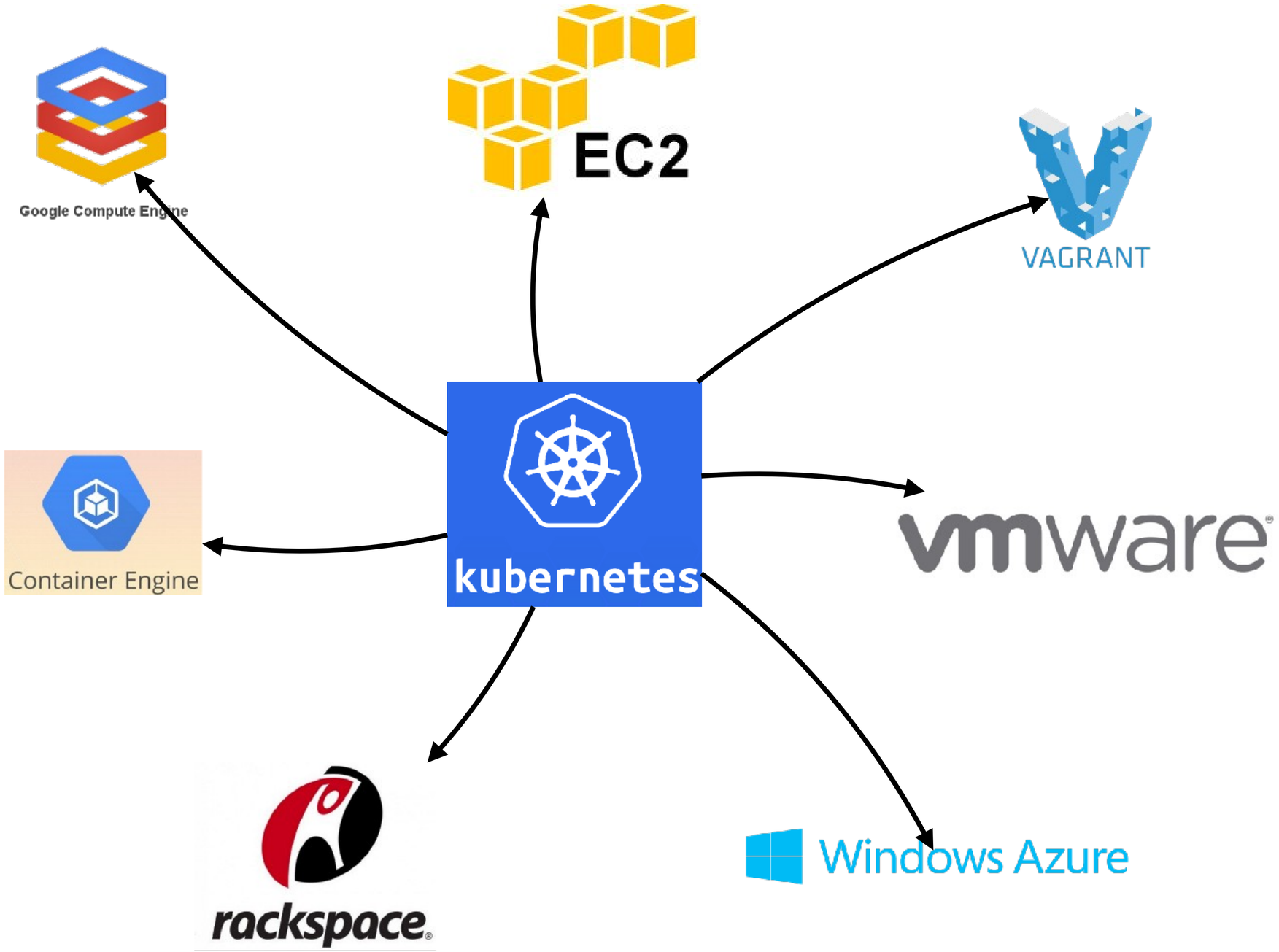
kubectl

- Controls the Kubernetes cluster manager
- CRUD Kubernetes resources
  - `create`, `get`, `describe`, `delete`, …
  - `kubectl create -f <filename>`
- `kubectl get nodes` or `pods`
- `kubectl scale –replicas=3 rc/<name>`

minikube

Minikube

- Runs a single node cluster in a VM
- Targeted for local development
- `minikube start`, `stop`, `docker-env`, ...
- Requires `kubectl` CLI
- github.com/kubernetes/minikube/releases



https://get.k8s.io/

Kubernetes Pod Configuration

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: wildfly-pod
  labels:
    name: wildfly-pod
spec:
  containers:
  - name: wildfly
    image: jboss/wildfly
    ports:
    - containerPort: 8080
```

Kubernetes and Java Developers