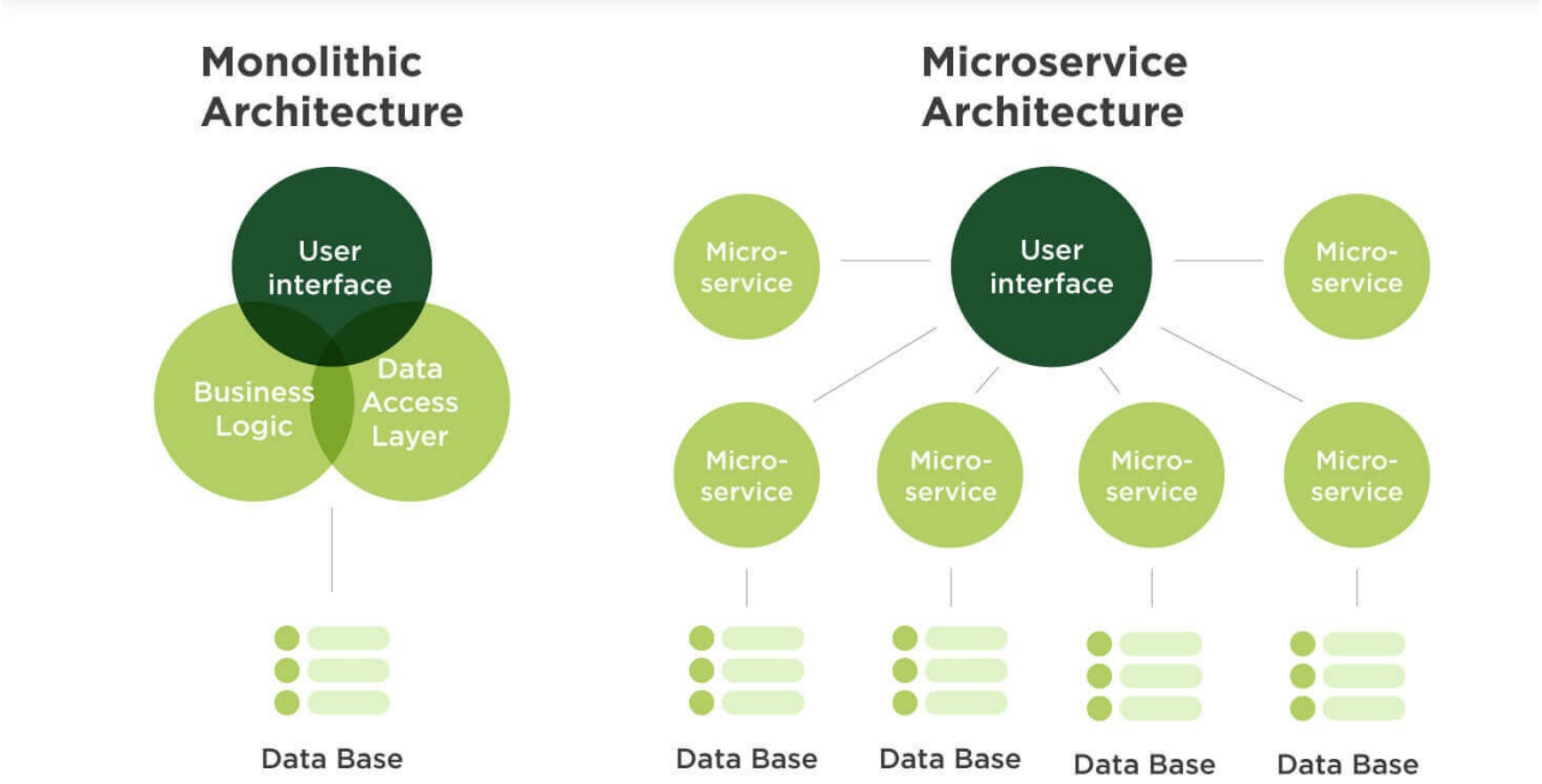
Introduçao ao Kubernetes

Agenda

Tópicos
Motivação
Arquitetura do Kubernetes
Objetos do Kubernetes
Dashboard
Ferramentas de Monitoramento
Azure Kubernertes Services - AKS
Contexto
Kubernetes Cheat Sheets
Kubectl Command Cheat Sheets

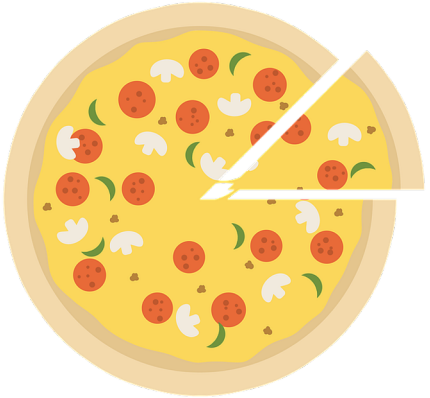
Motivaçao

Microservice Architecture



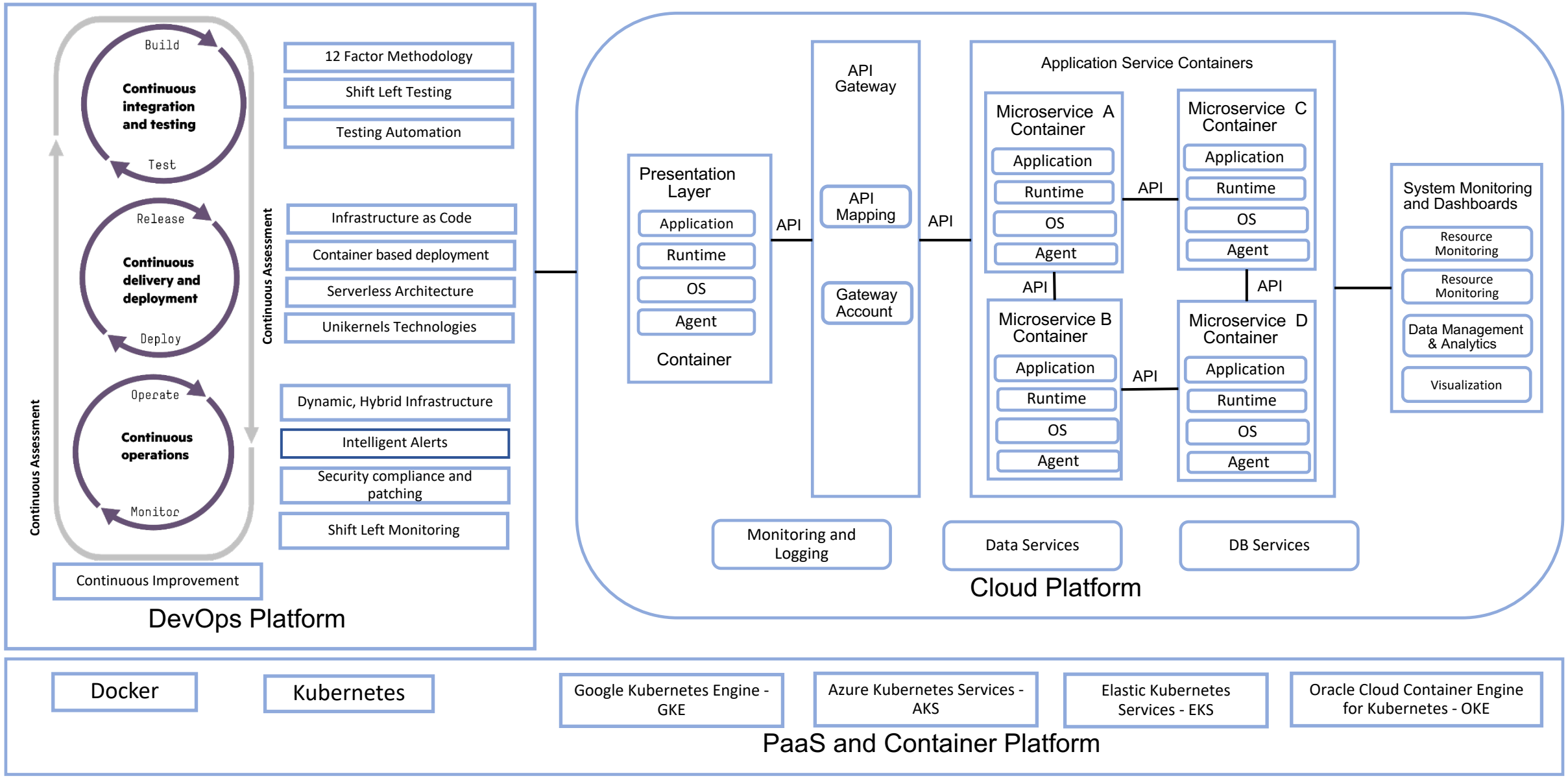
Microservices Principles

1. Deployment Independence - updates to an individual microservice have no negative impact to any other component of the system. Optimized for Replacement
2. Organized around business capabilities
3. Products not Projects
4. API Focused
5. Smart endpoints and dumb pipes
6. Decentralized Governance
7. Decentralized Data Management
8. Infrastructure Automation (infrastructure as code)
9. Design for failure
10. Evolutionary Design



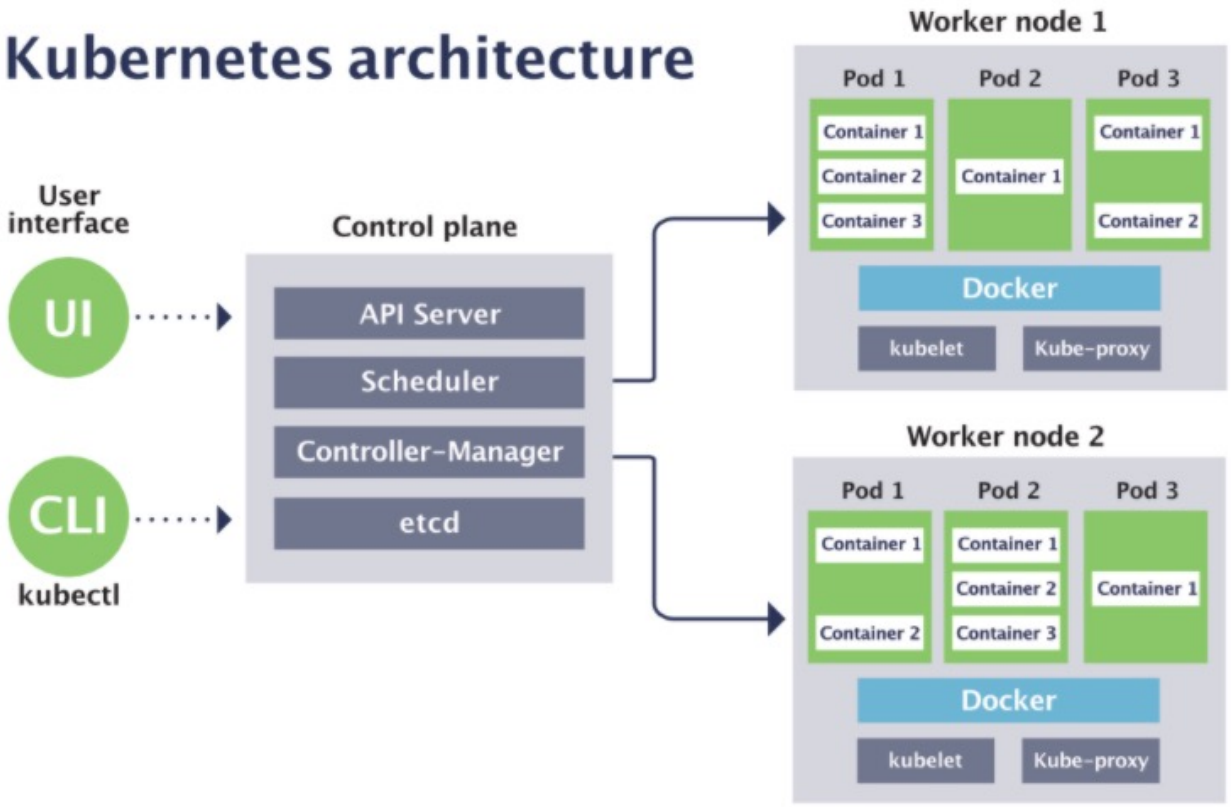
2 Pizza Team

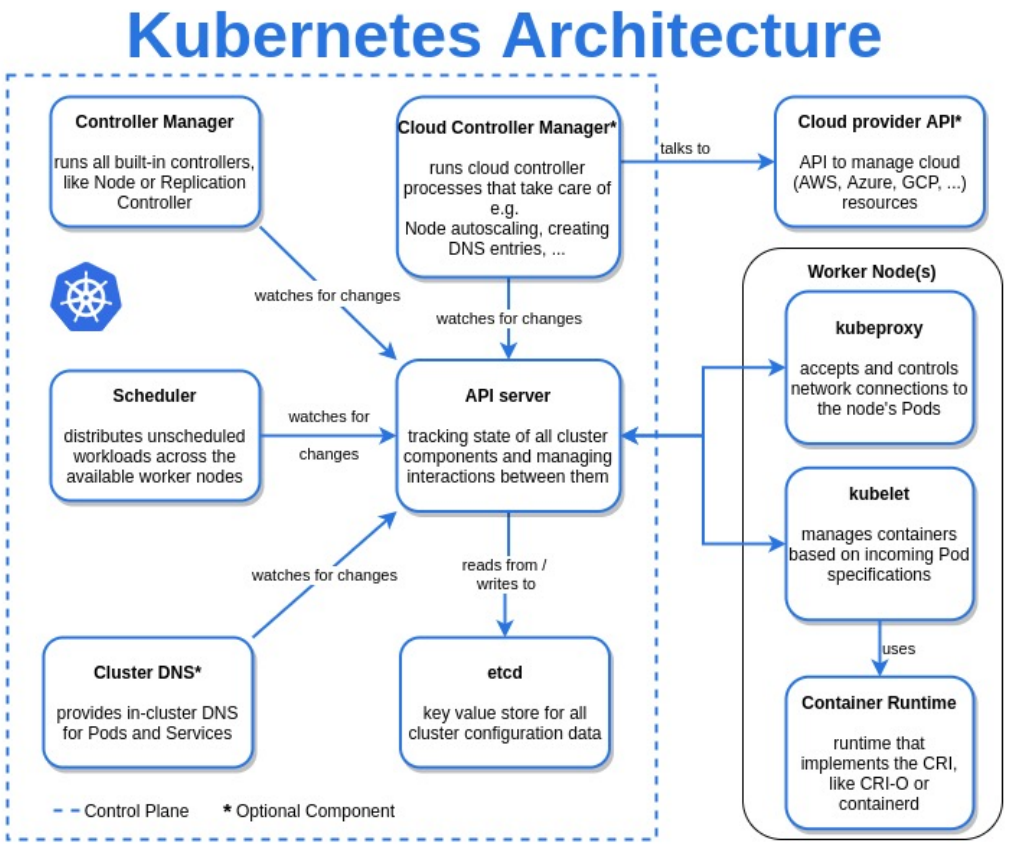
Microservice | API and Container Based Management Strategy



Arquitetura do Kubernetes

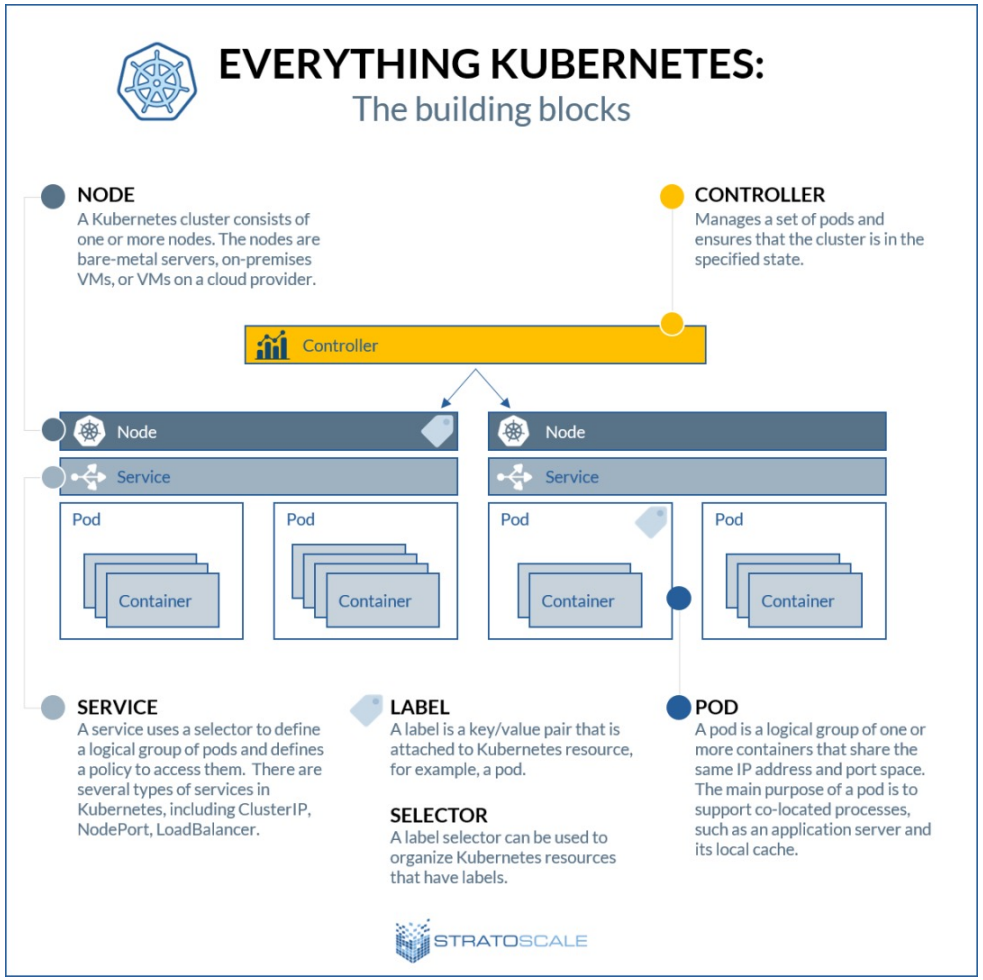
Kubernetes Architecture



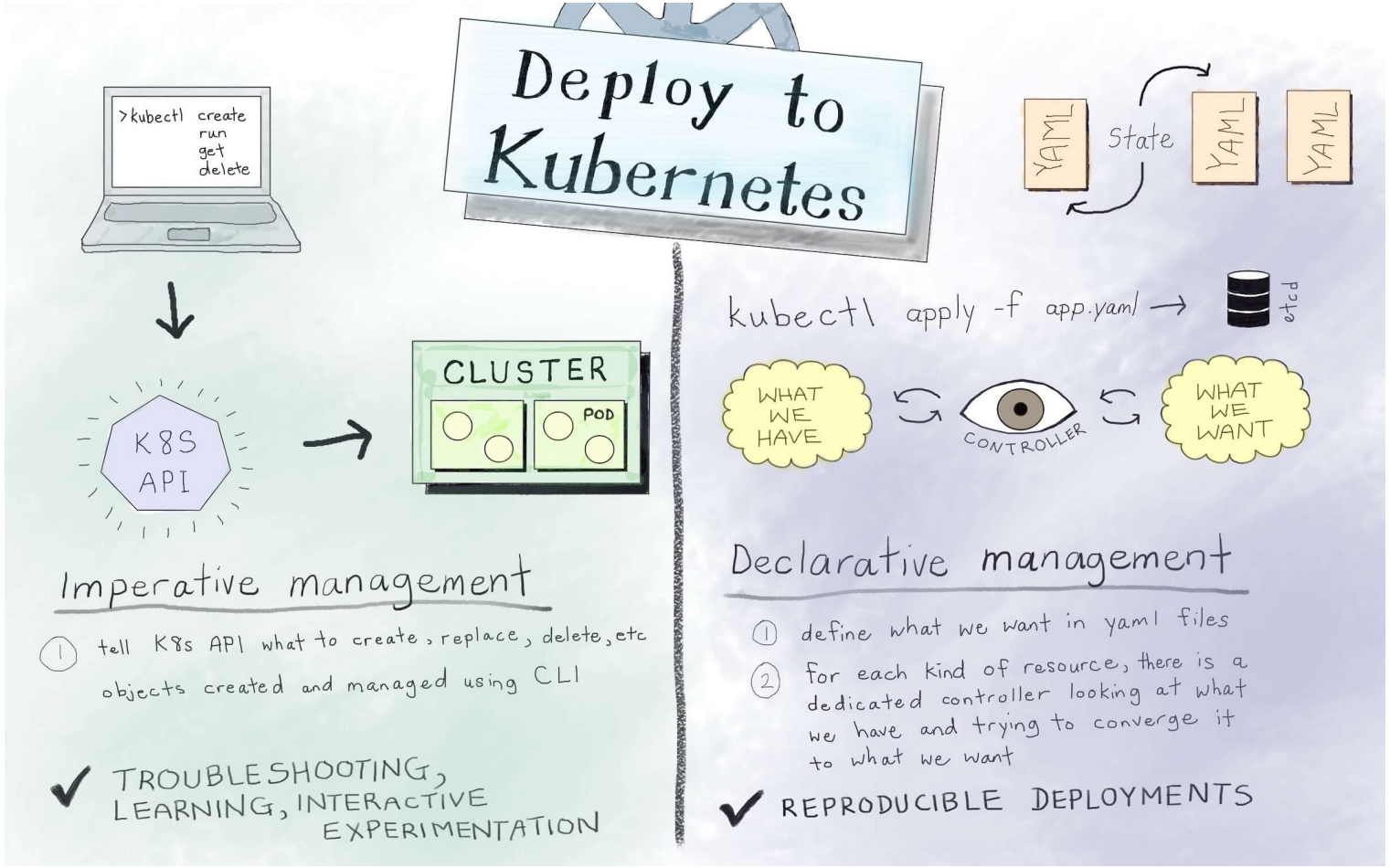


Objetos do Kubernetes

Kubernetes Objects

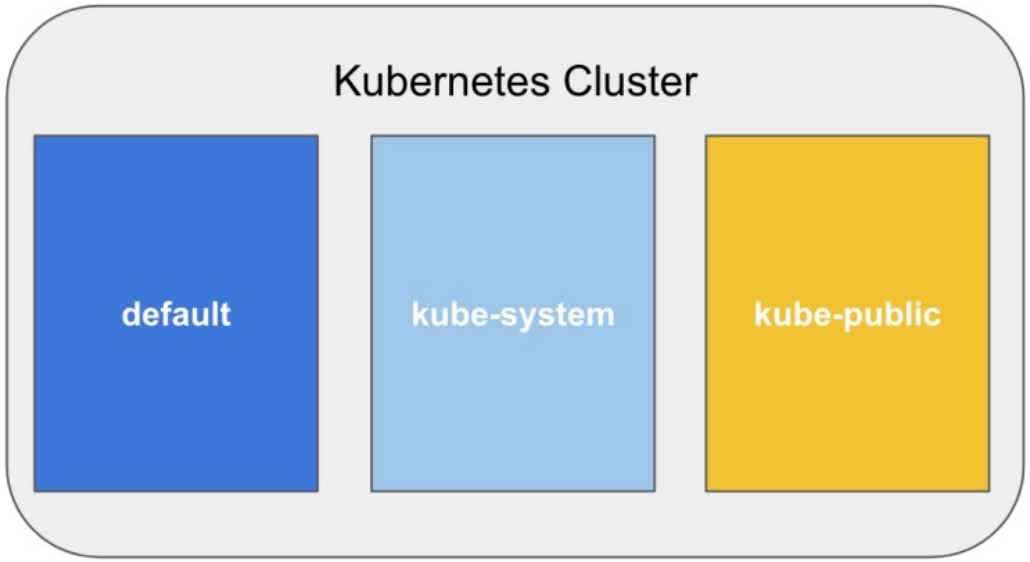


Imperative vs Declarative



Namespace

- Logical Partition within the cluster
- Controlled access to resources such as pods, services and etc
- Limit resource consumption for that namespace
- Use labels for slight difference

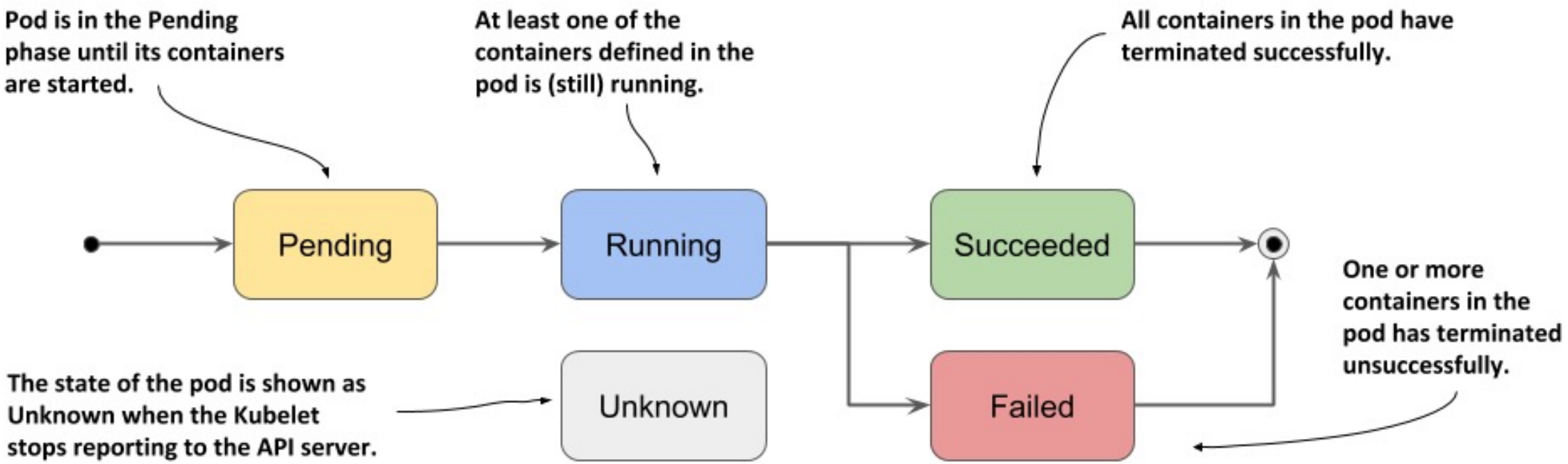


Pod

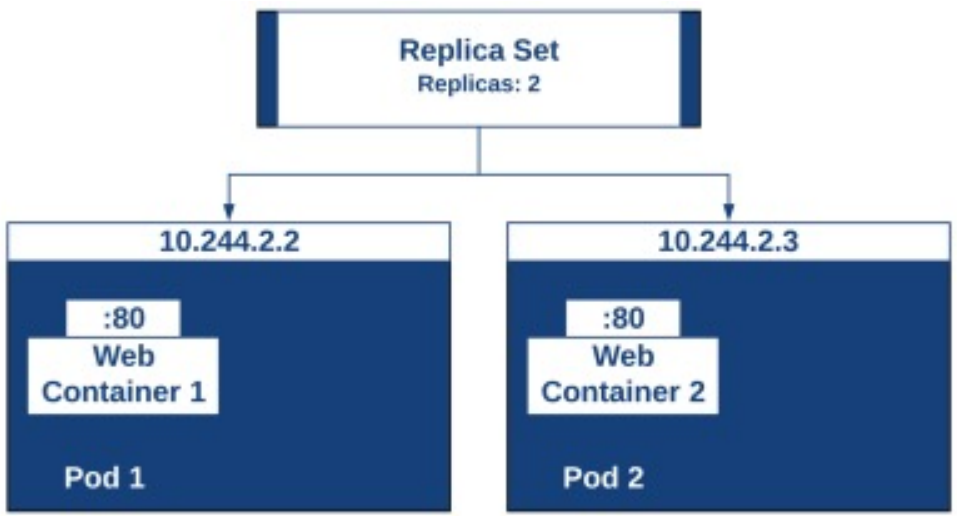
Single Container Pods

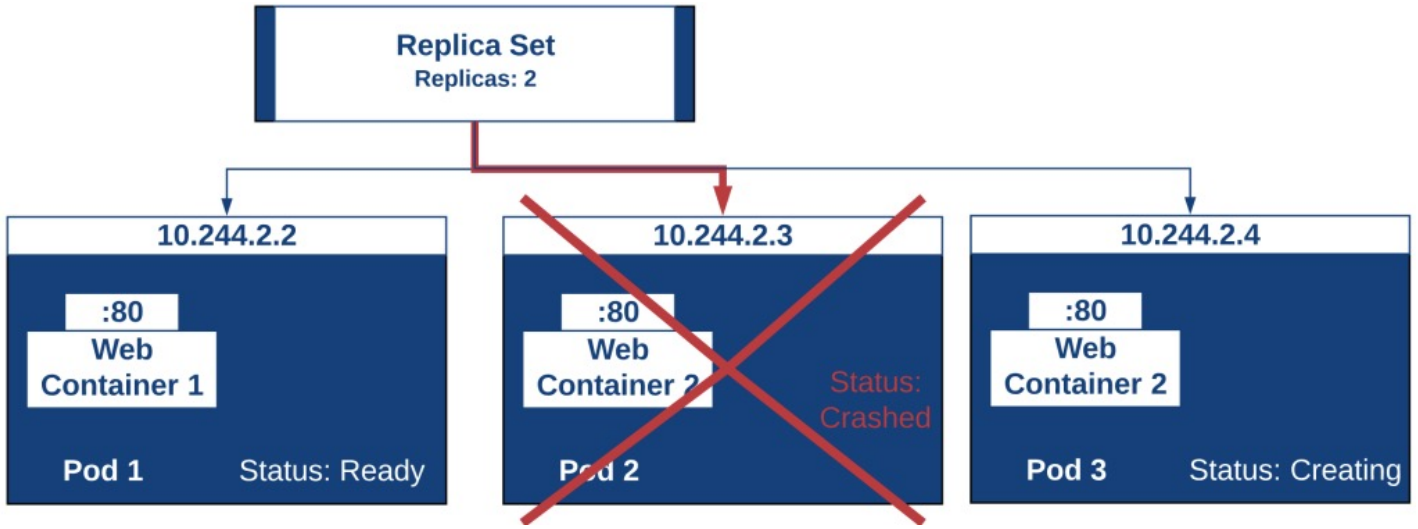
Multi-Container Pods

Pod Status

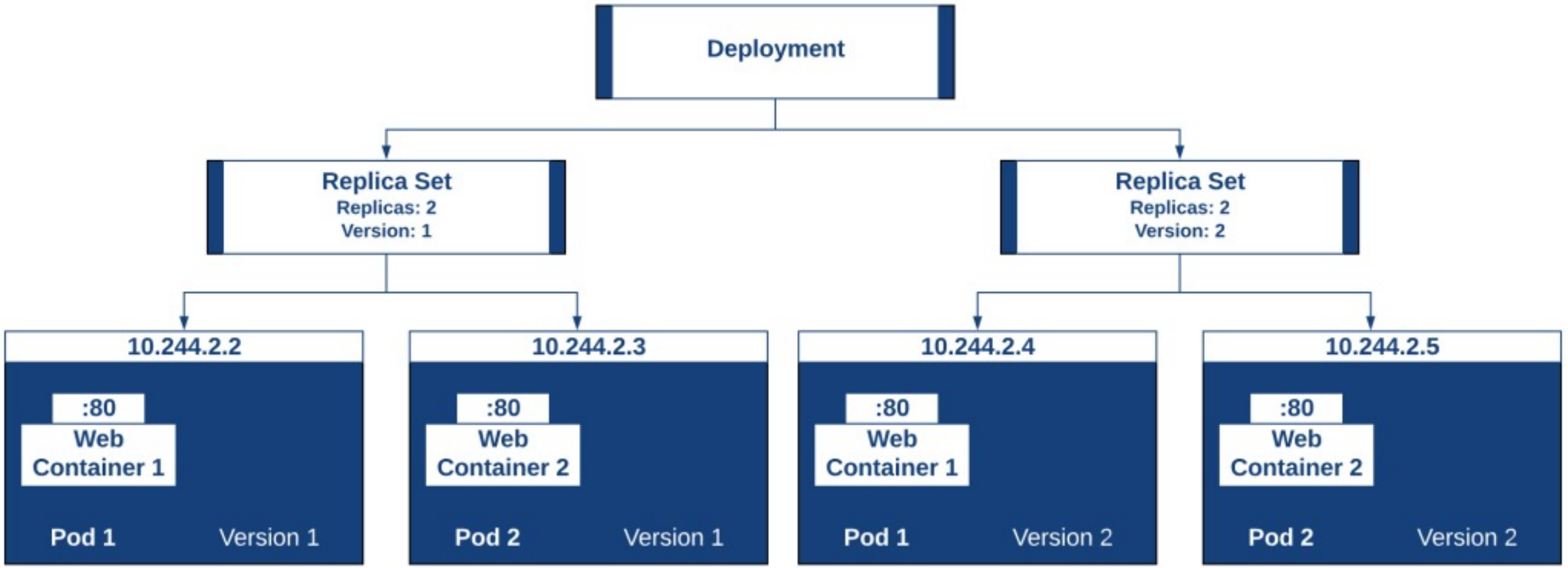

Pod is in the Pending phase until its containers are started.
At least one of the containers defined in the pod is (still) running.
All containers in the pod have terminated successfully.
Pending
Running
Succeeded
One or more containers in the pod has terminated unsuccessfully.
The state of the pod is shown as Unknown when the Kubelet stops reporting to the API server.
Unknown
Failed

Replicaset

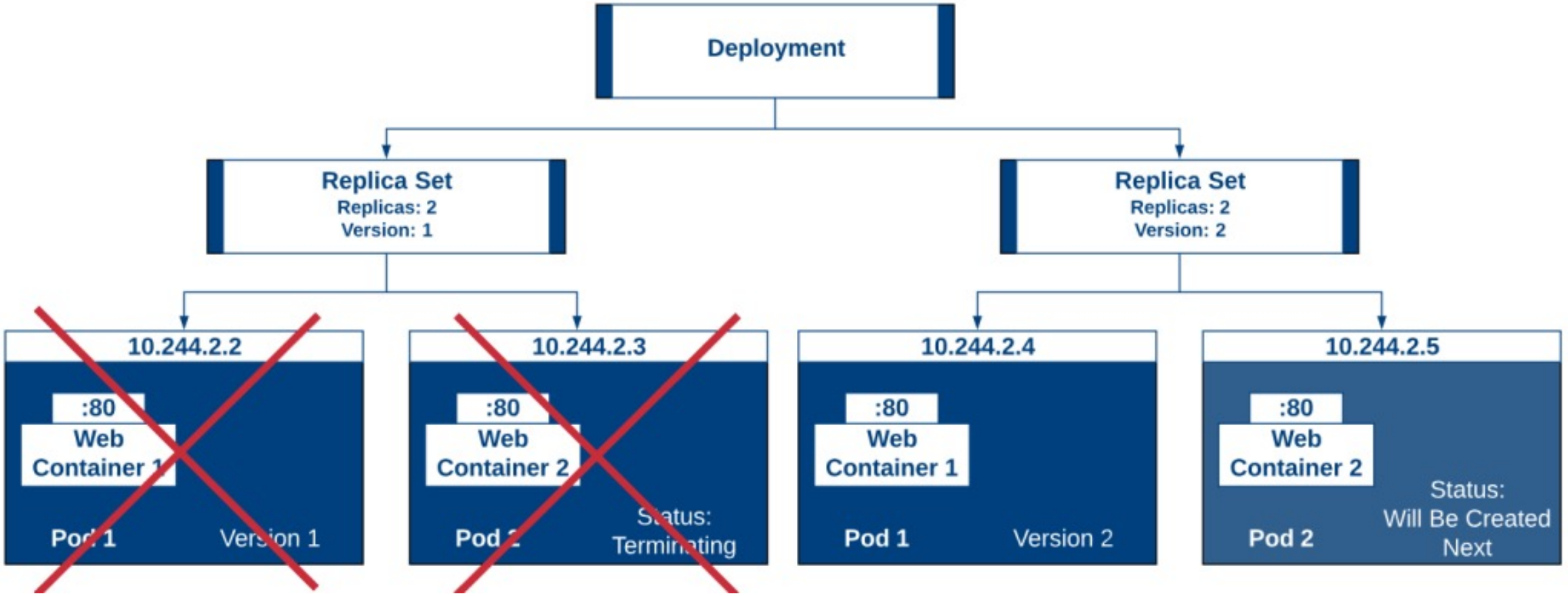

Replica Set
Replicas: 2
10.244.2.2
:80
Web
Container 1
Pod 1
10.244.2.3
:80
Web
Container 2
Pod 2


Replica Set
Replicas: 2
10.244.2.2
:80
Web
Container 1
Pod 1
Status: Ready
10.244.2.3
:80
Web
Container 2
Status:
Crashed
10.244.2.4
:80
Web
Container 2
Pod 3
Status: Creating

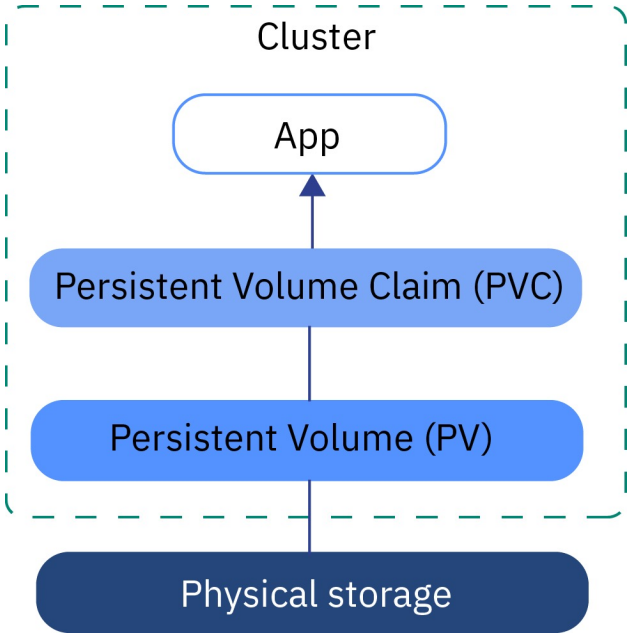
Deployment

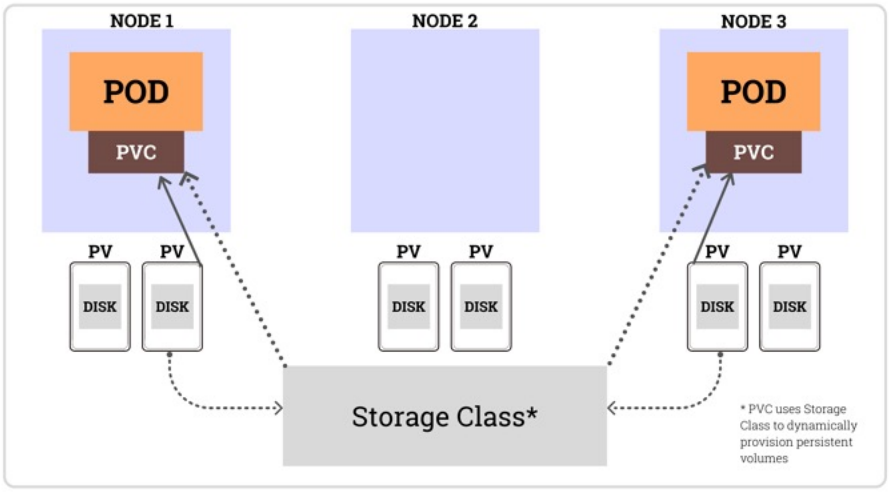

Deployment
Replica Set
Replicas: 2
Version: 1
Replica Set
Replicas: 2
Version: 2
10.244.2.2
:80
Web
Container 1
Pod 1
Version 1
10.244.2.3
:80
Web
Container 2
Pod 2
Version 1
10.244.2.4
:80
Web
Container 1
Pod 1
Version 2
10.244.2.5
:80
Web
Container 2
Pod 2
Version 2

Deployment

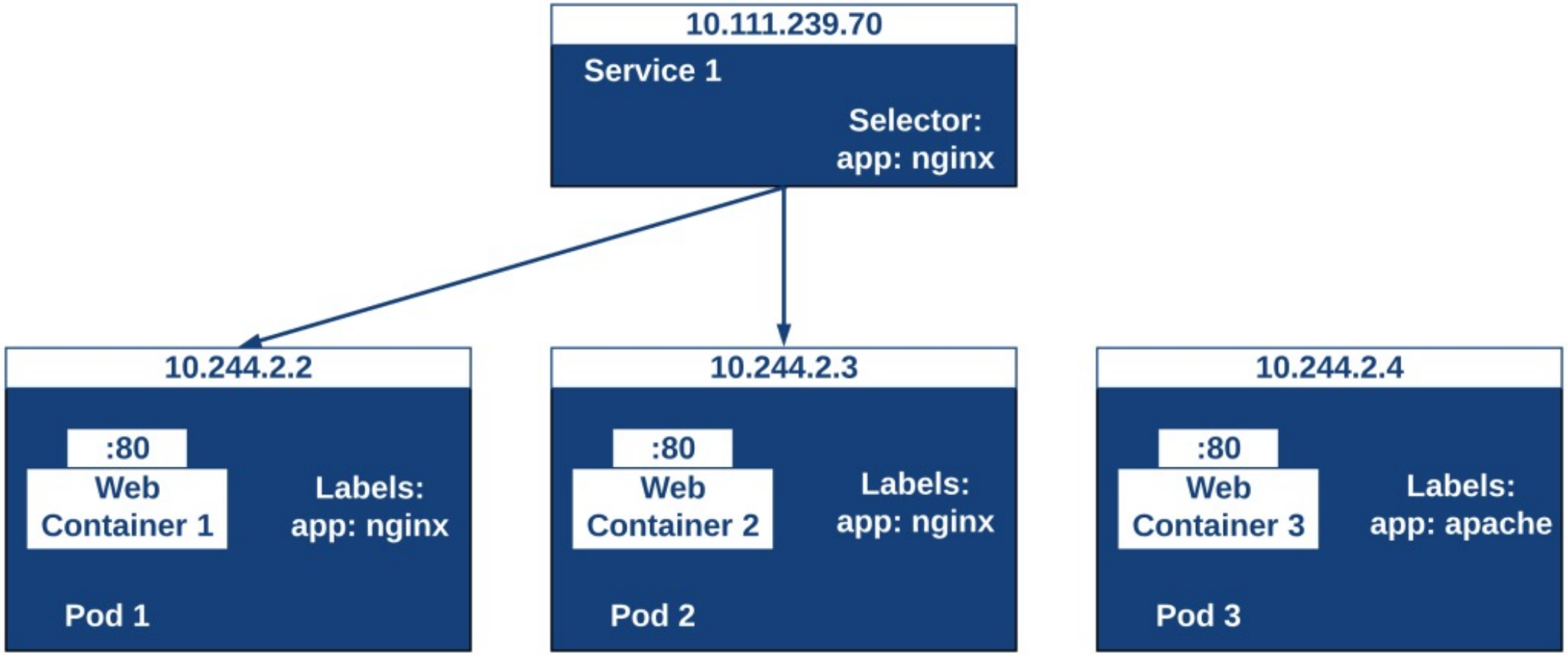

Deployment
Replica Set
Replicas: 2
Version: 1
Replica Set
Replicas: 2
Version: 2
10.244.2.2
:80
Web
Container 1
Pod 1
Version 1
10.244.2.3
:80
Web
Container 2
Pod 2
Status:
Terminating
10.244.2.4
:80
Web
Container 1
Pod 1
Version 2
10.244.2.5
:80
Web
Container 2
Pod 2
Status:
Will Be Created
Next

Persistent Volumes

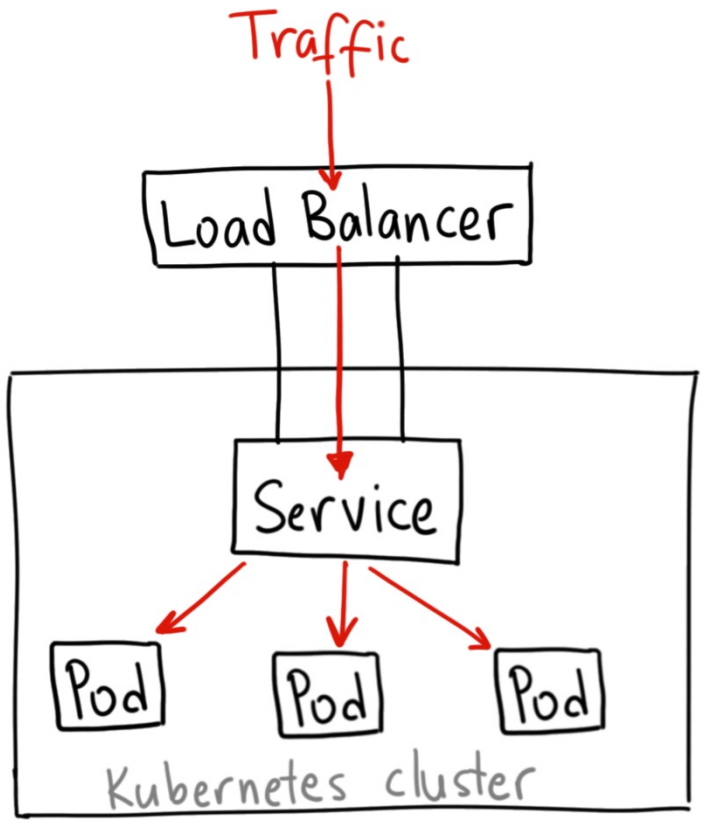




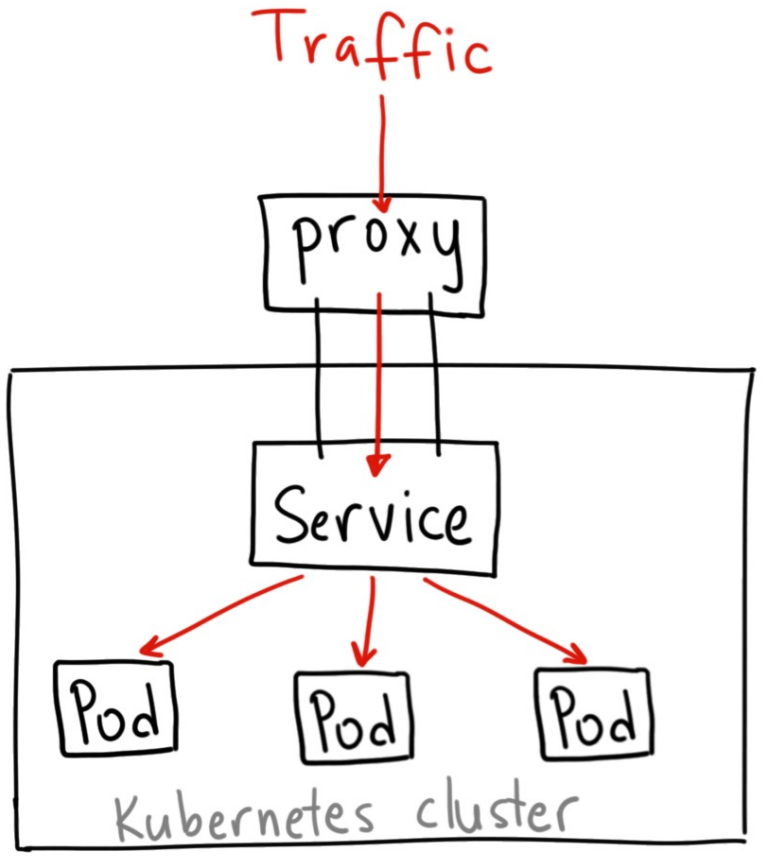
Services and Labels



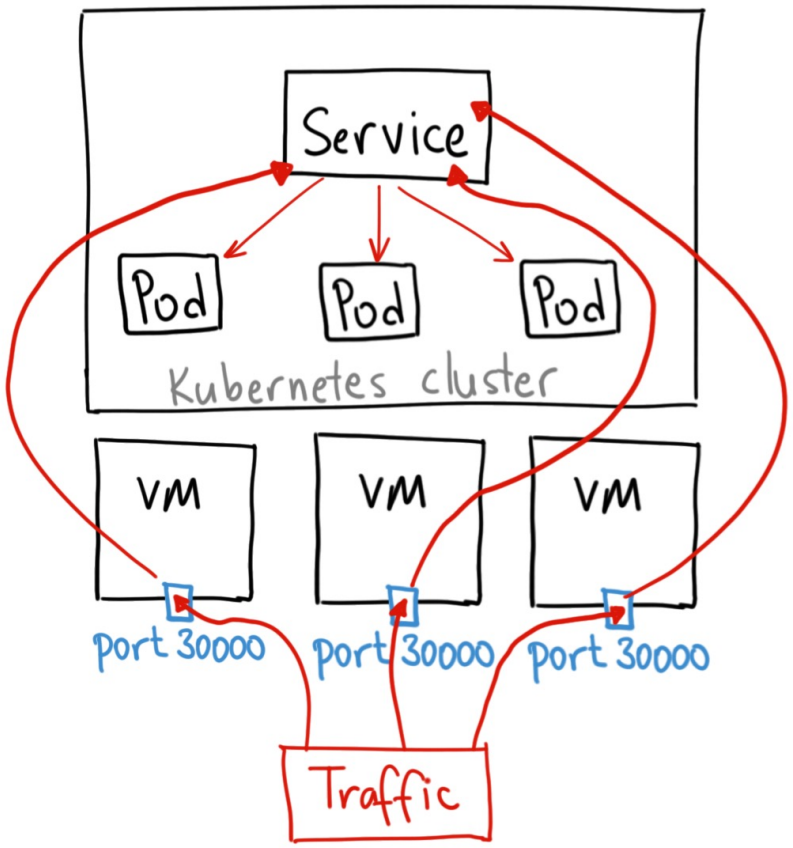
Service - LoadBalancer



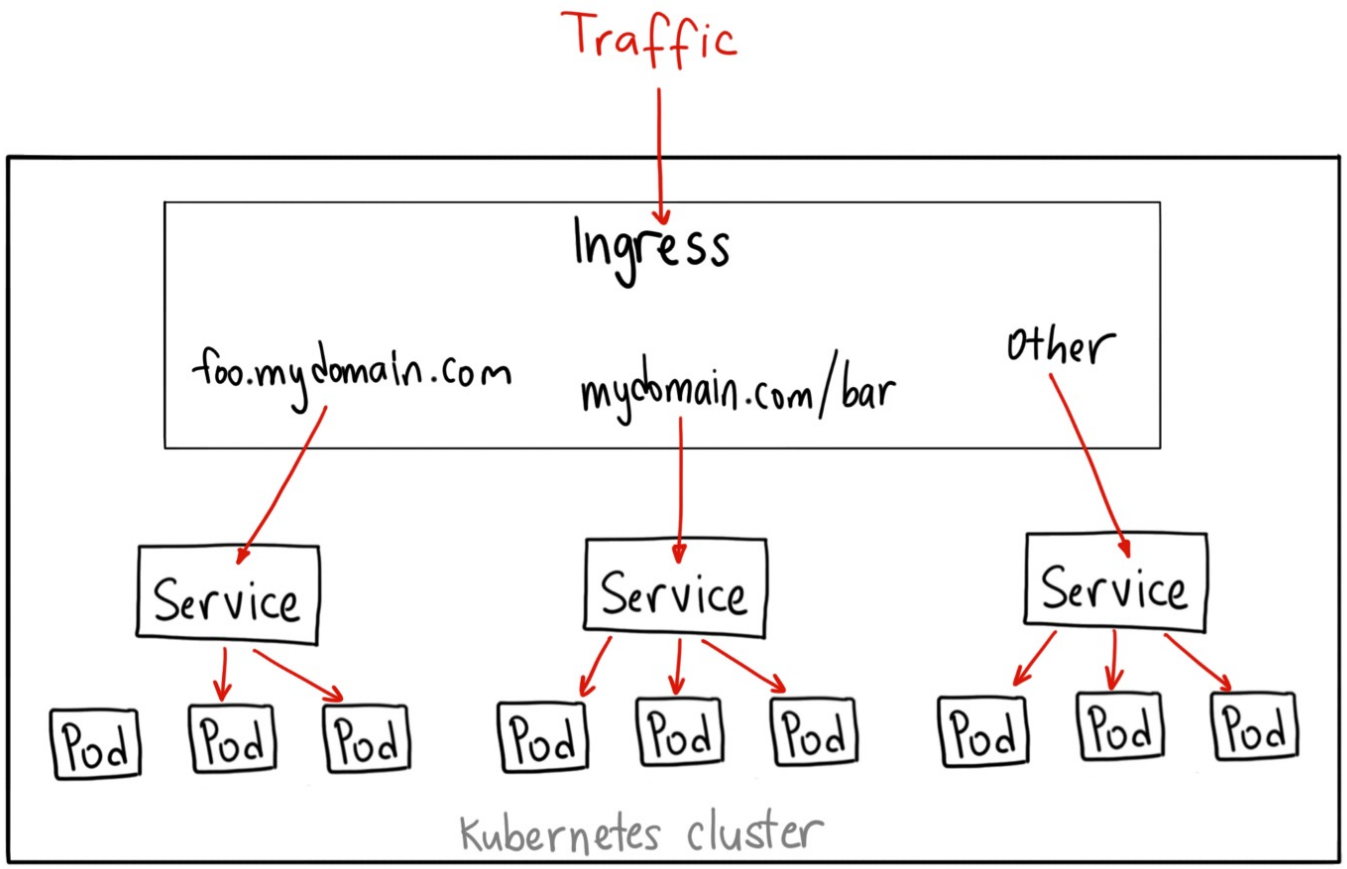
Service - ClusterIP



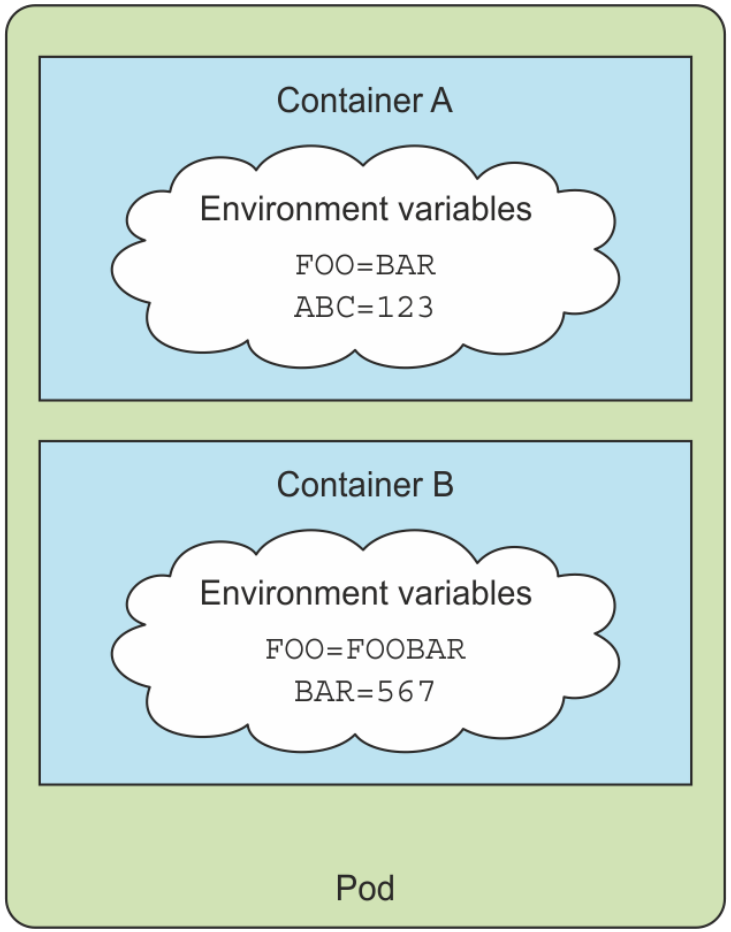
Service - NodePort

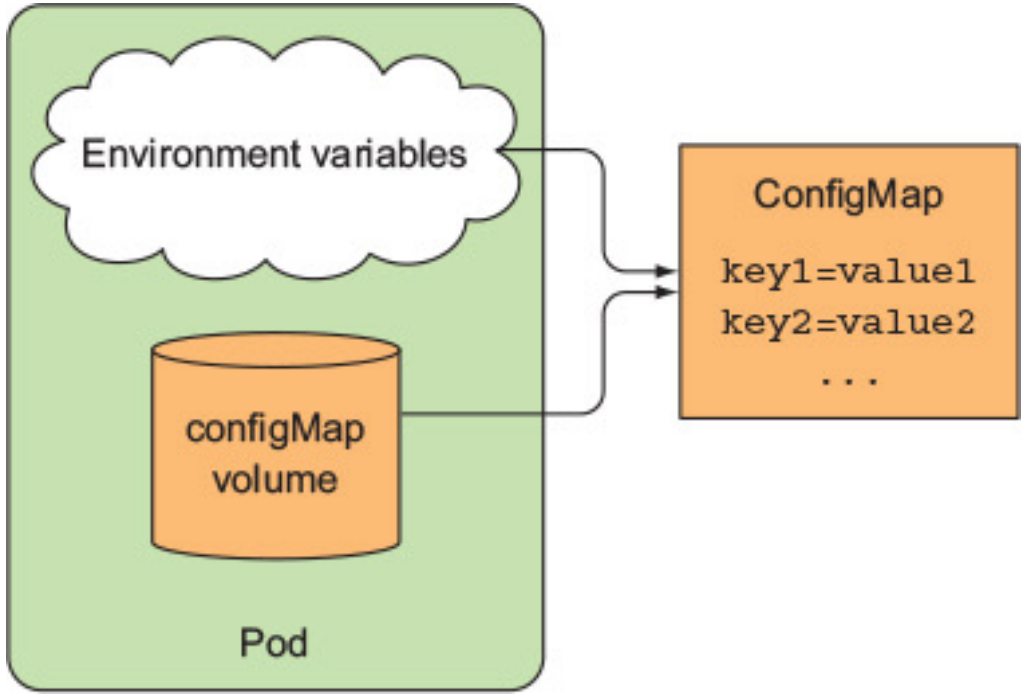


Ingress

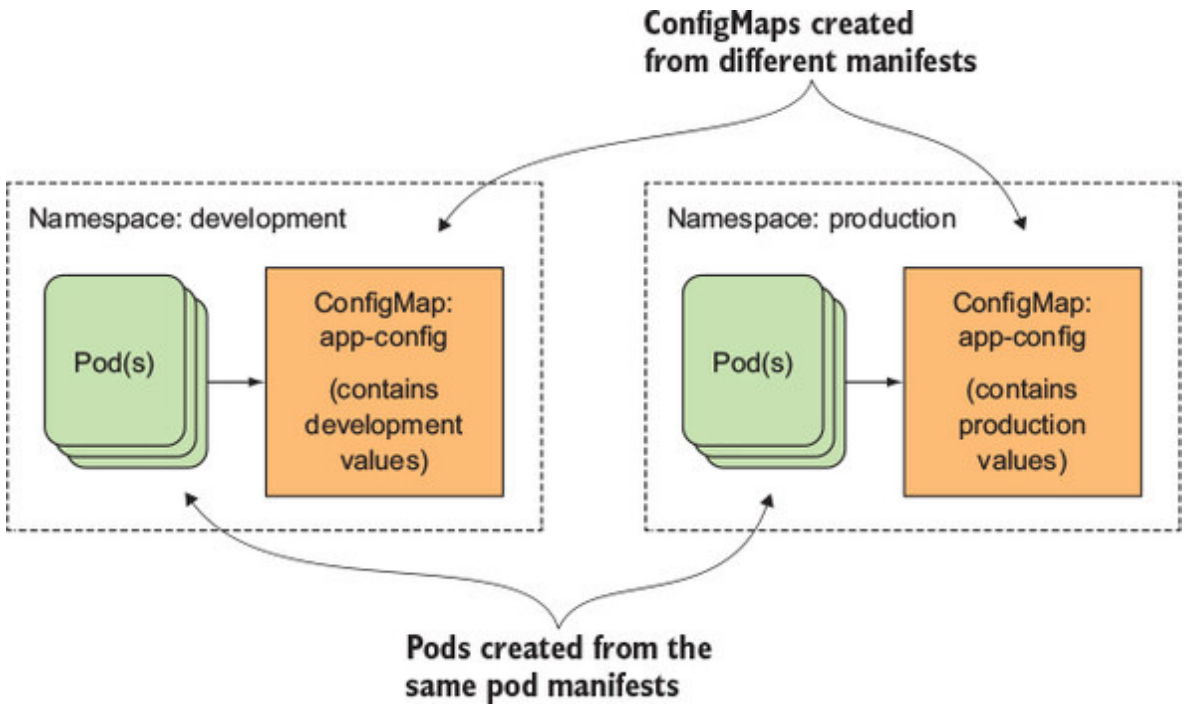

Traffic
Ingress
foo.mydomain.com
mydomain.com/bar
Other
Service
Service
Service
Pod
Pod
Pod
Pod
Pod
Pod
Pod
Pod
Pod
Kubernetes cluster

ConfigMaps | Secret



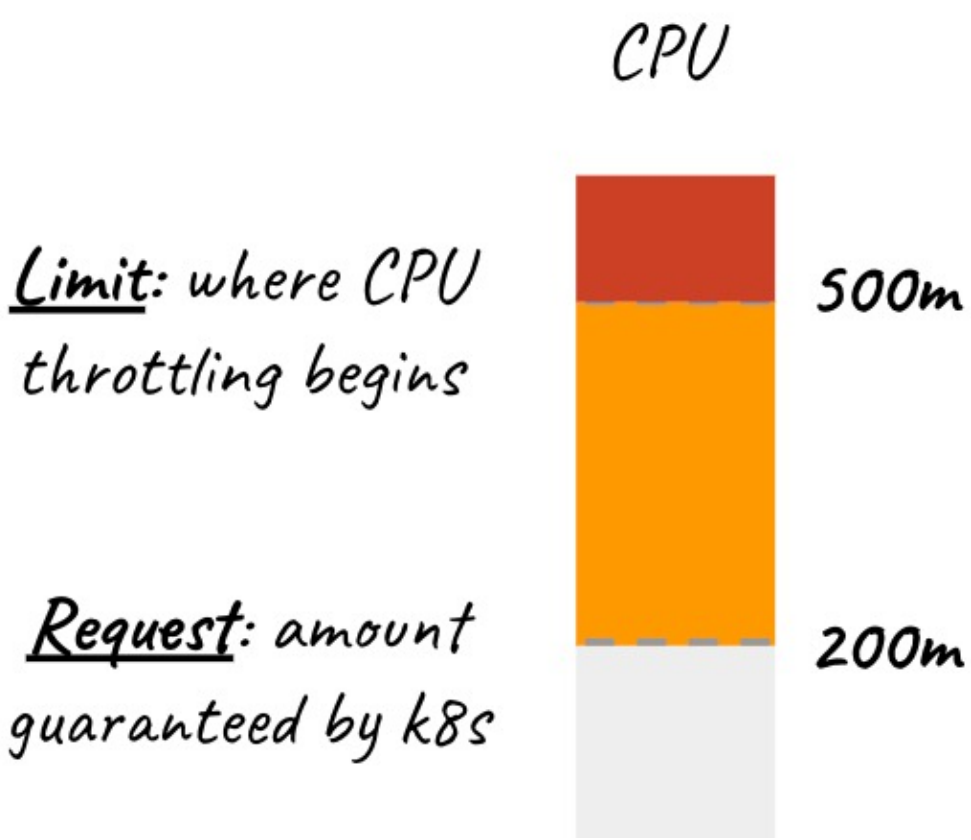


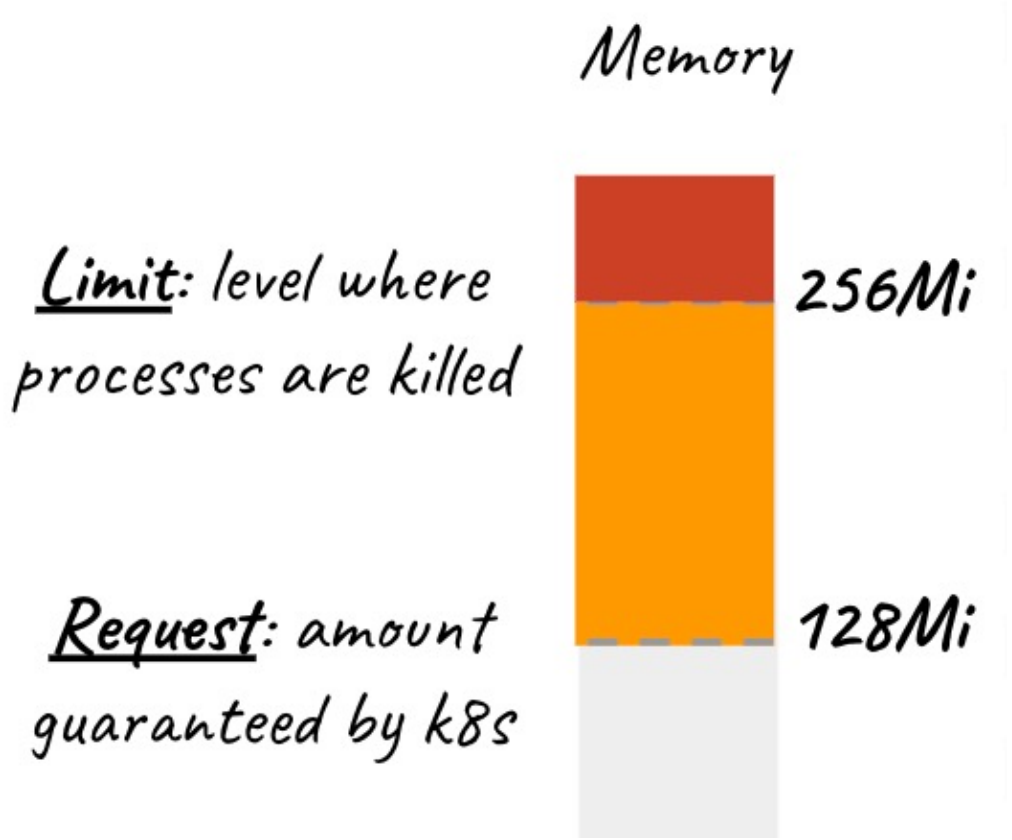
ConfigMaps | Secrets



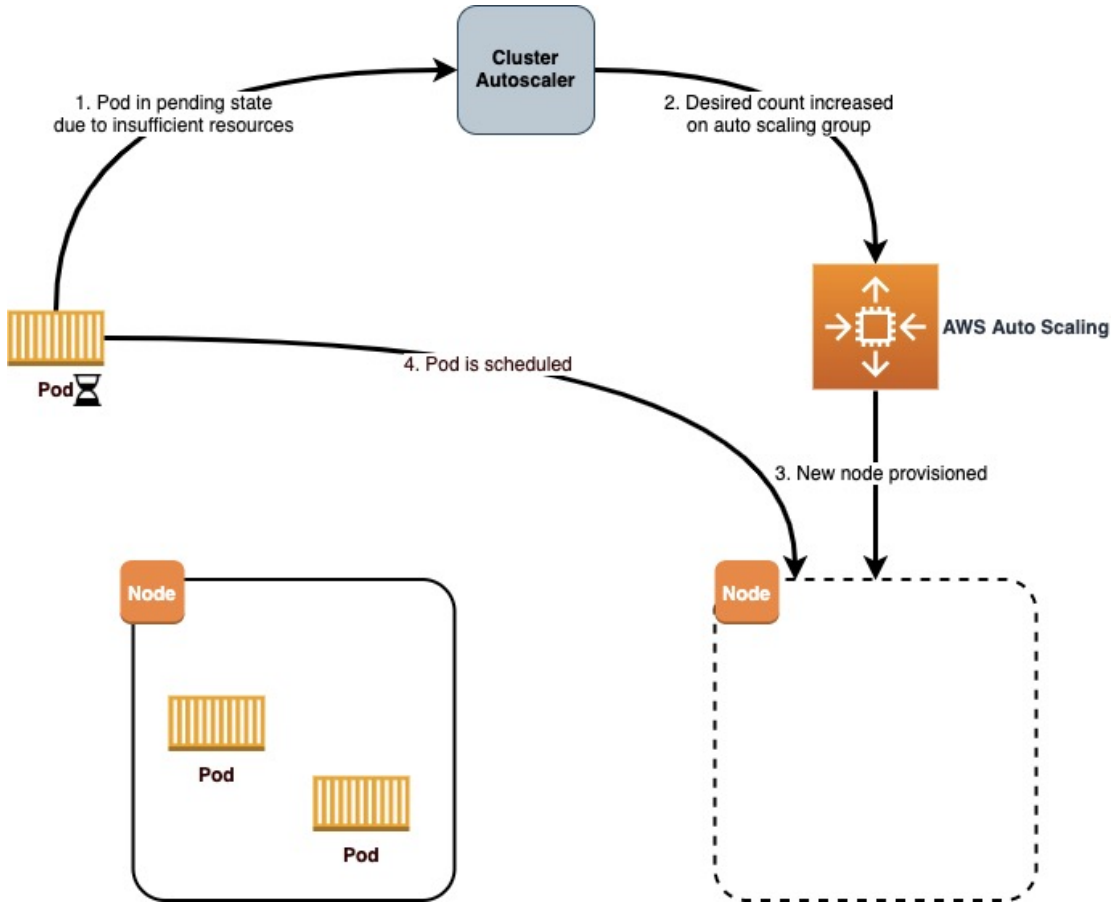
Horizontal Pod Autoscaler - HPA

Requests and Limits

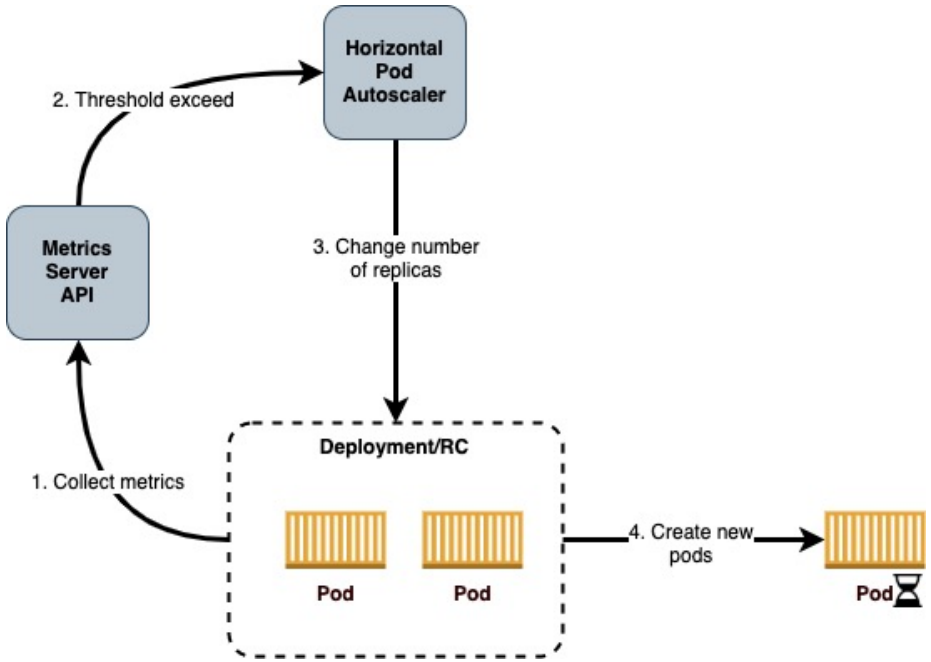




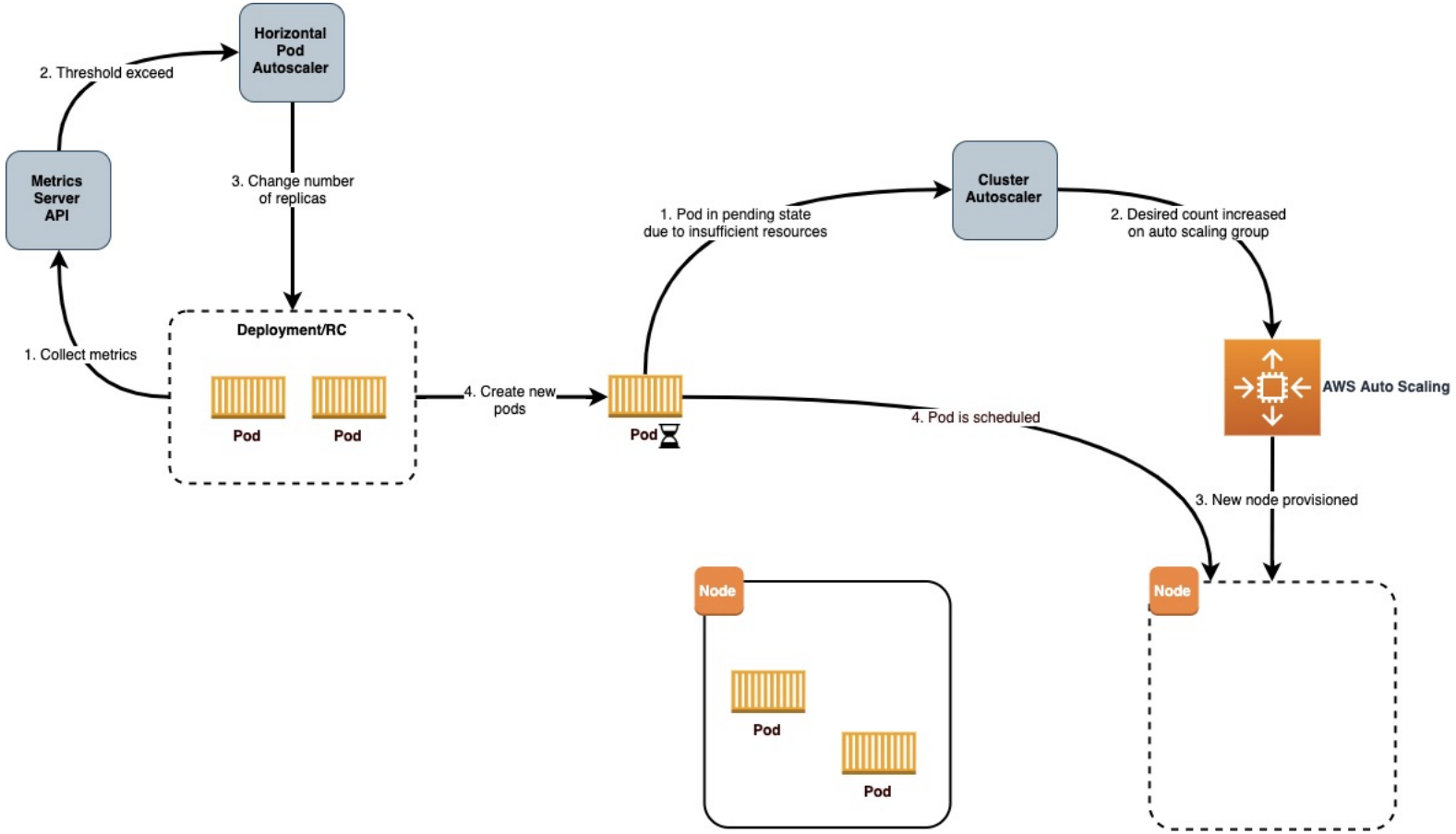
Cluster Autoscaler



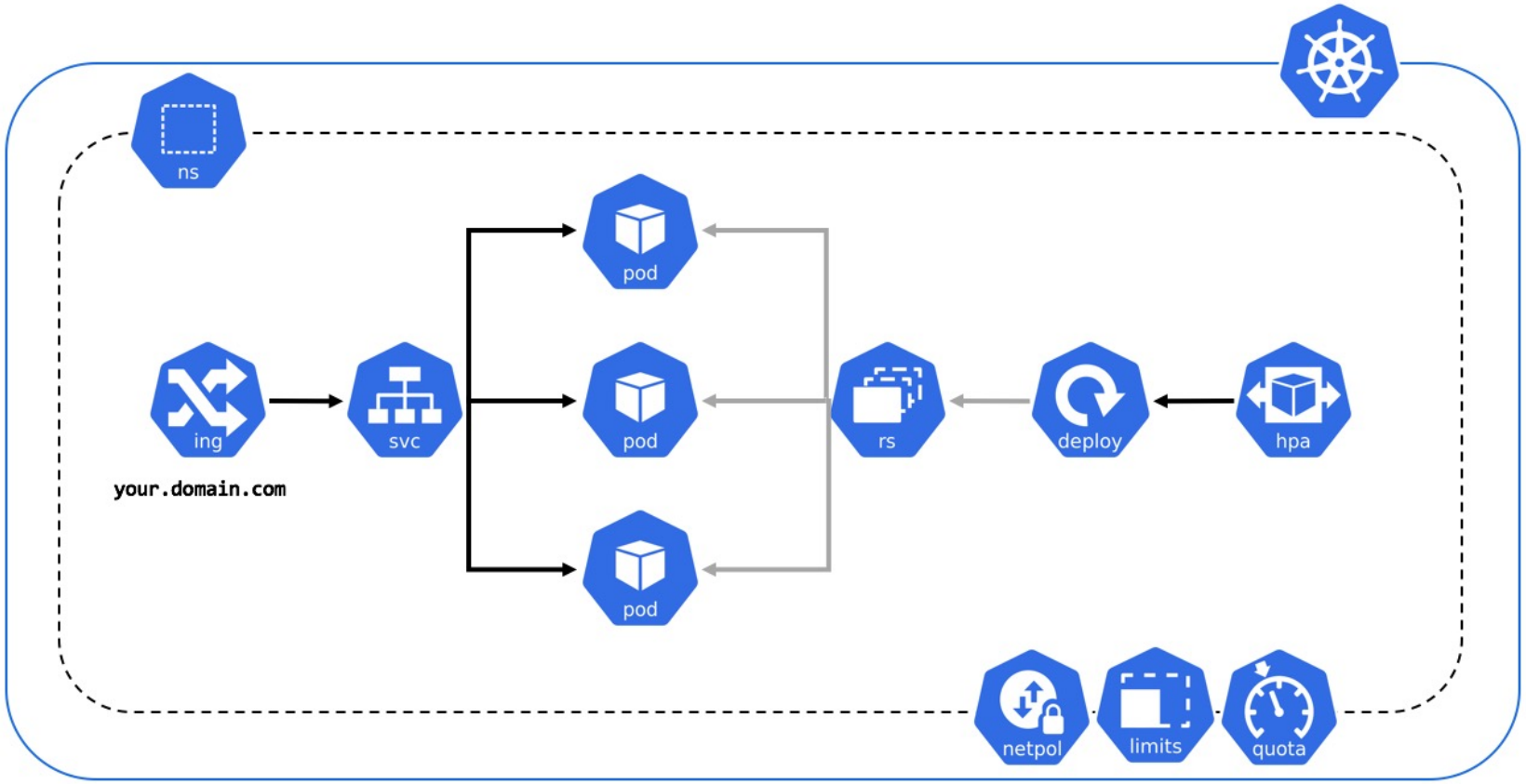
Horizontal Pod Autoscaler - HPA



Horizontal Pod Autoscaler | Cluster Autoscaler

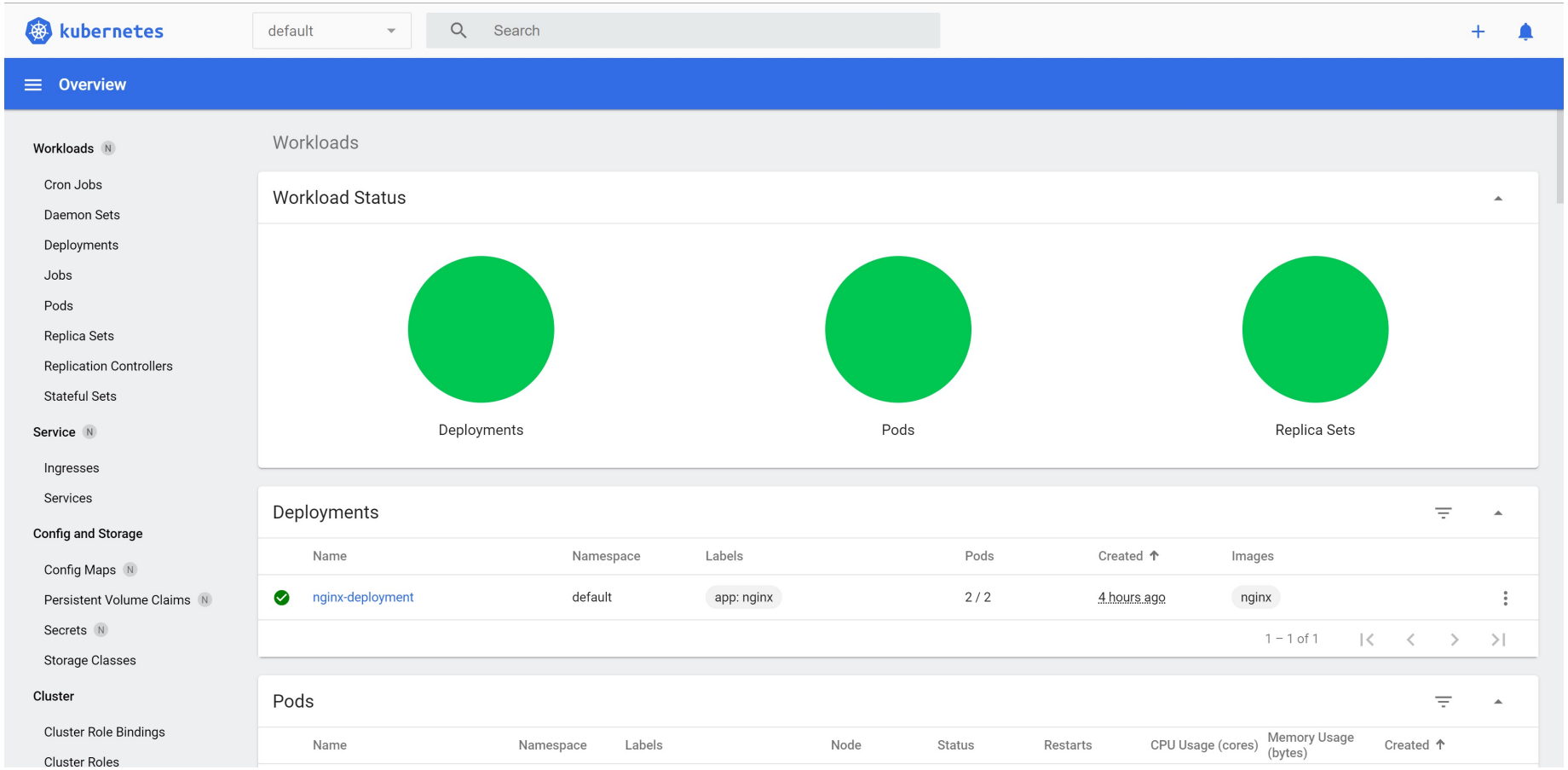


Objetos do Kubernetes



Dashboard

Dashboard


kubernetes
default
Search
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses
Services
Config and Storage
Config Maps
Persistent Volume Claims
Secrets
Storage Classes
Cluster
Cluster Role Bindings
Cluster Roles
Workloads
Workload Status
Deployments
Pods
Replica Sets
Deployments
Name
Namespace
Labels
Pods
Created
Images
nginx-deployment
default
app: nginx
2 / 2
4 hours ago
nginx
1 – 1 of 1
Pods
Name
Namespace
Labels
Node
Status
Restarts
CPU Usage (cores)
Memory Usage (bytes)
Created

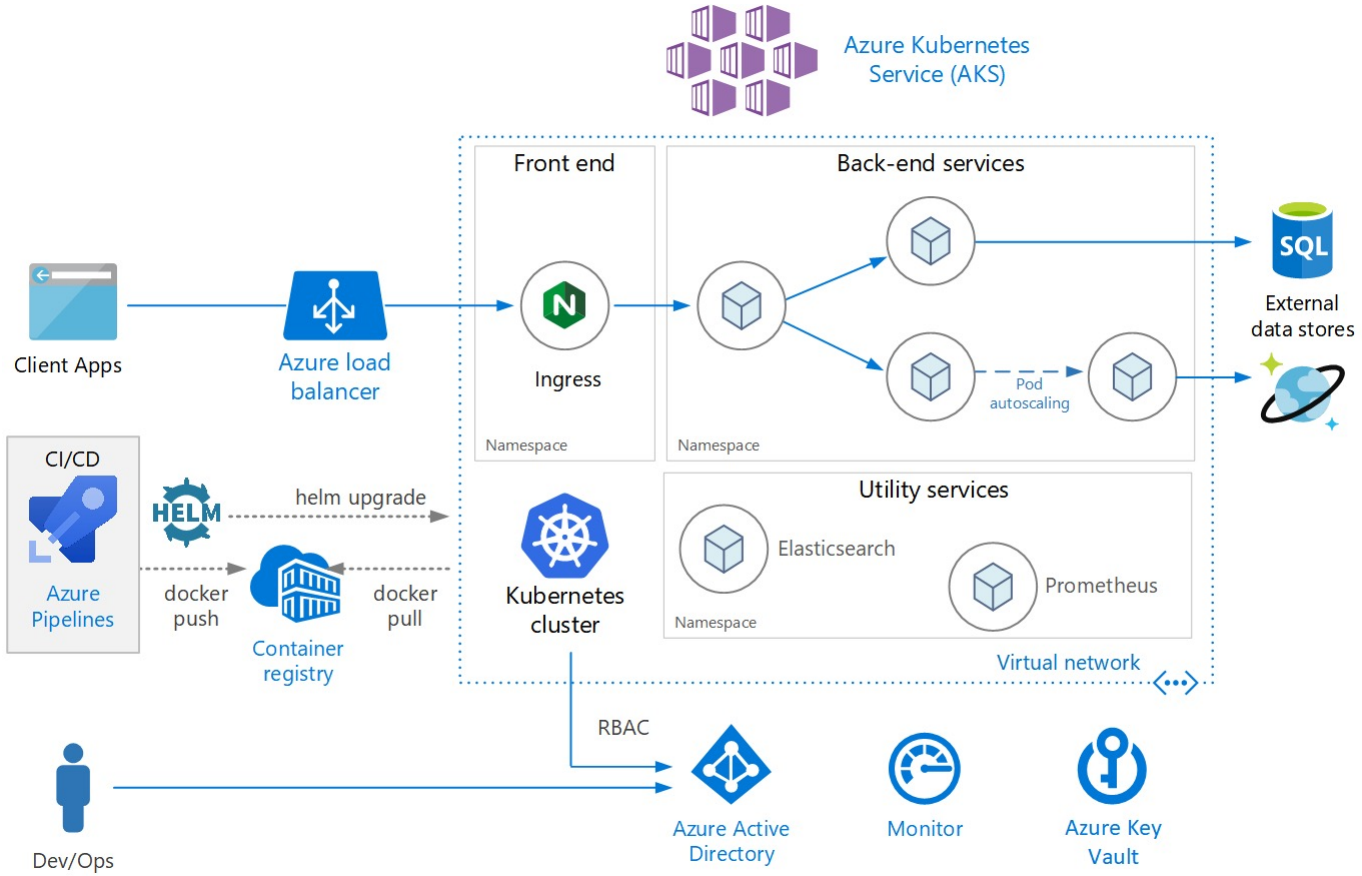
Ferramentas de Monitoramento

Ferramentas de Monitoramento

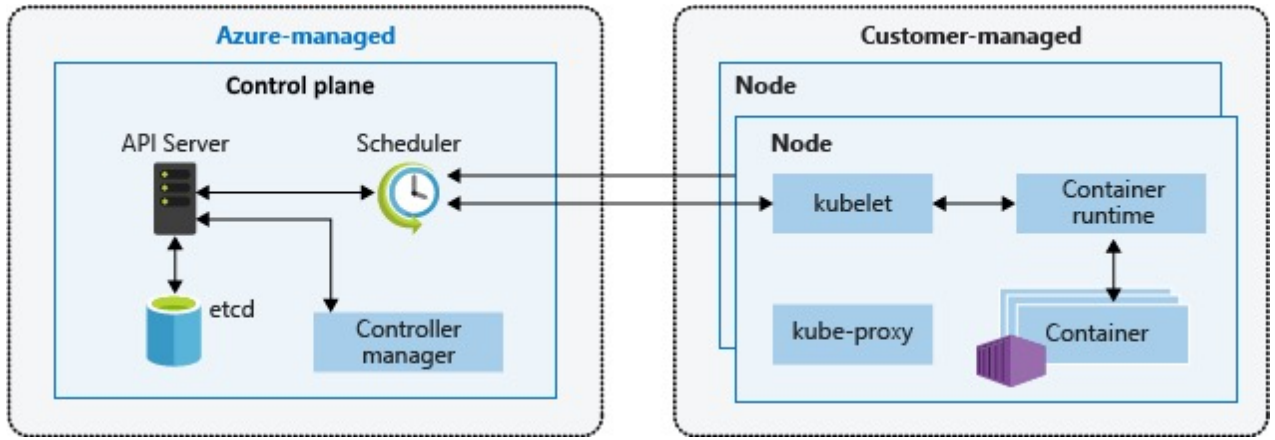
- https://codeberg.org/hjacobs/kube-ops-view
- https://k9scli.io
- https://k8slens.dev/desktop.html

Azure Kubernetes Service - AKS

Azure Kubernetes Service



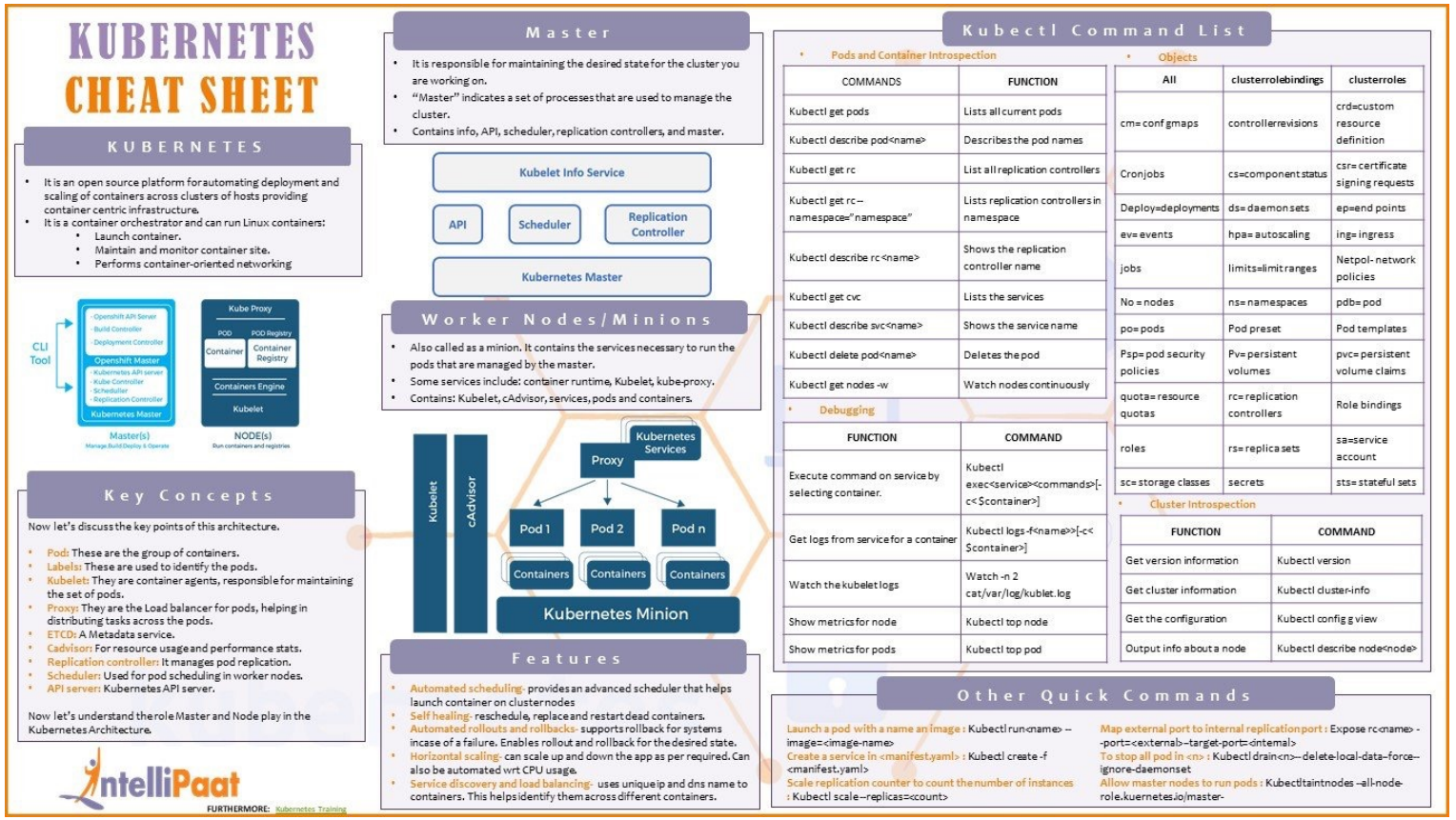
Azure Kubernetes Service



Contexto

Kubernetes Cheat Sheet

Kubernetes Cheat Sheet



KUBERNETES CHEAT SHEET

KUBERNETES

- It is an open source platform for automating deployment and scaling of containers across clusters of hosts providing container centric infrastructure.
- It is a container orchestrator and can run Linux containers:
  - Launch container.
  - Maintain and monitor container site.
  - Performs container-oriented networking

CLI Tool

Openshift Master
- Openshift API Server
- Build Controller
- Deployment Controller
- Kubernetes API server
- Kube Controller
- Scheduller
- Replication Controller

Kubernetes Master

Master(s)
Manage,Build,Deploy & Operate

Kube Proxy
POD
POD Registry
Container
Container Registry
Containers Engine
Kubelet

NODE(s)
Run containers and registries

Key Concepts

Now let's discuss the key points of this architecture.

- **Pod:** These are the group of containers.
- **Labels:** These are used to identify the pods.
- **Kubelet:** They are container agents, responsible for maintaining the set of pods.
- **Proxy:** They are the Load balancer for pods, helping in distributing tasks across the pods.
- **ETCD:** A Metadata service.
- **Cadvisor:** For resource usage and performance stats.
- **Replication controller:** It manages pod replication.
- **Scheduler:** Used for pod scheduling in worker nodes.
- **API server:** Kubernetes API server.

Now let's understand the role Master and Node play in the Kubernetes Architecture.

IntelliPaat

FURTHERMORE: Kubernetes Training

Master

- It is responsible for maintaining the desired state for the cluster you are working on.
- "Master" indicates a set of processes that are used to manage the cluster.
- Contains info, API, scheduler, replication controllers, and master.

Kubelet Info Service

API | Scheduler | Replication Controller

Kubernetes Master

Worker Nodes/Minions

- Also called as a minion. It contains the services necessary to run the pods that are managed by the master.
- Some services include: container runtime, Kubelet, kube-proxy.
- Contains: Kubelet, cAdvisor, services, pods and containers.

Kubelet | cAdvisor

Kubernetes Services
Proxy
Pod 1 | Pod 2 | Pod n
Containers | Containers | Containers

Kubernetes Minion

Features

- **Automated scheduling-** provides an advanced scheduler that helps launch container on cluster nodes
- **Self healing-** reschedule, replace and restart dead containers.
- **Automated rollouts and rollbacks-** supports rollback for systems incase of a failure. Enables rollout and rollback for the desired state.
- **Horizontal scaling-** can scale up and down the app as per required. Can also be automated wrt CPU usage.
- **Service discovery and load balancing-** uses unique ip and dns name to containers. This helps identify them across different containers.

Kubectl Command List

Pods and Container Introspection

COMMANDS	FUNCTION
Kubectl get pods	Lists all current pods
Kubectl describe pod<name>	Describes the pod names
Kubectl get rc	List all replication controllers
Kubectl get rc --namespace="namespace"	Lists replication controllers in namespace
Kubectl describe rc <name>	Shows the replication controller name
Kubectl get cvc	Lists the services
Kubectl describe svc<name>	Shows the service name
Kubectl delete pod<name>	Deletes the pod
Kubectl get nodes -w	Watch nodes continuously

Debugging

FUNCTION	COMMAND
Execute command on service by selecting container.	Kubectl exec<service><commands>[-c< $container>]
Get logs from service for a container	Kubectl logs -f<name>>[-c< $container>]
Watch the kubelet logs	Watch -n 2 cat/var/log/kublet.log
Show metrics for node	Kubectl top node
Show metrics for pods	Kubectl top pod

Objects

All	clusterrolebindings	clusterroles
cm= conf gmaps	controllerrevisions	crd=custom resource definition
Cronjobs	cs=component status	csr= certificate signing requests
Deploy=deployments	ds= daemon sets	ep=end points
ev= events	hpa= autoscaling	ing= ingress
jobs	limits=limit ranges	Netpol- network policies
No = nodes	ns= namespaces	pdb= pod
po= pods	Pod preset	Pod templates
Psp= pod security policies	Pv= persistent volumes	pvc= persistent volume claims
quota= resource quotas	rc= replication controllers	Role bindings
roles	rs= replica sets	sa=service account
sc= storage classes	secrets	sts= stateful sets

Cluster Introspection

FUNCTION	COMMAND
Get version information	Kubectl version
Get cluster information	Kubectl cluster-info
Get the configuration	Kubectl config g view
Output info about a node	Kubectl describe node<node>

Other Quick Commands

Launch a pod with a name an image : Kubectl run<name> --image=<image-name>

Create a service in <manifest.yaml> : Kubectl create -f <manifest.yaml>

Scale replication counter to count the number of instances : Kubectl scale --replicas=<count>

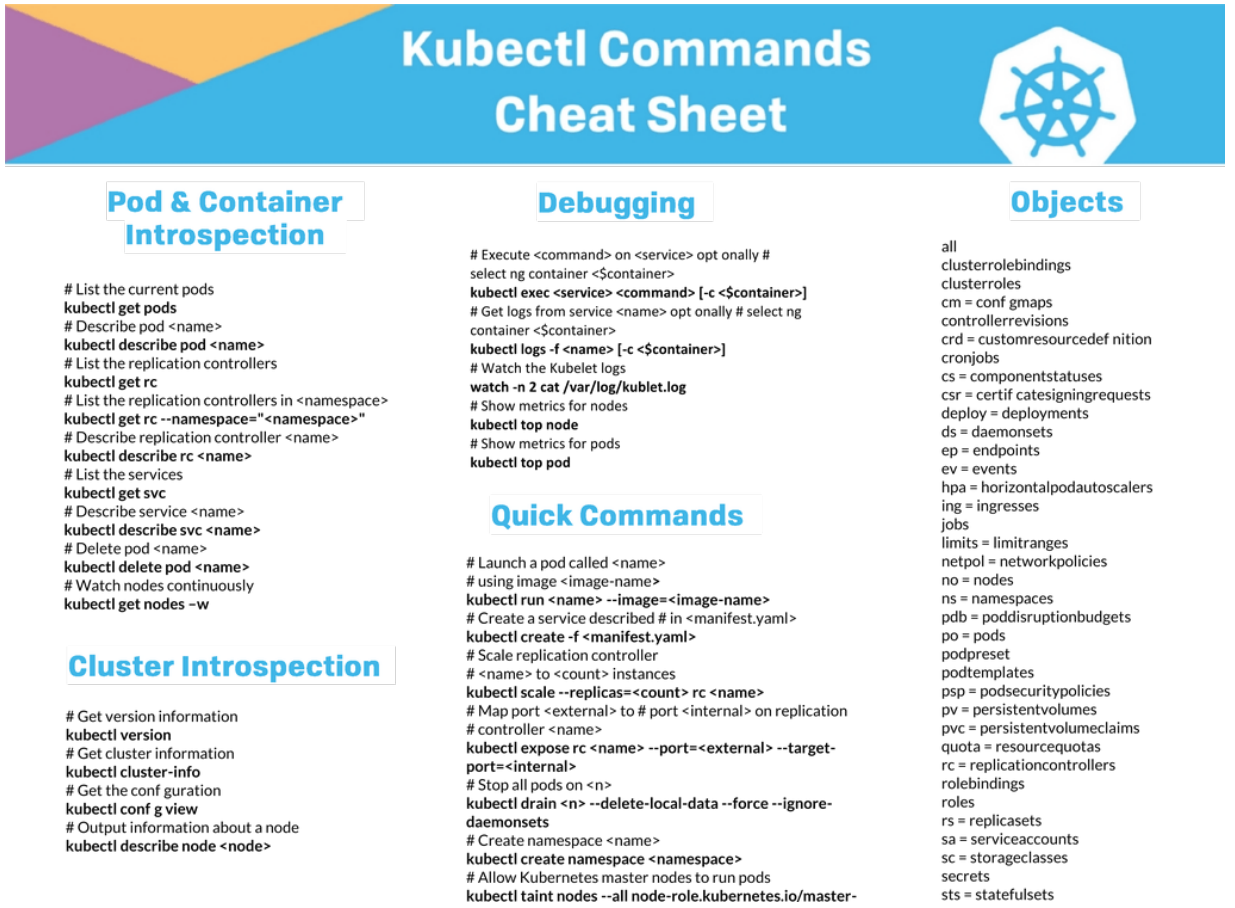
Map external port to internal replication port : Expose rc<name> --port=<external>--target-port=<internal>

To stop all pod in <n> : Kubectl drain<n>--delete-local-data--force--ignore-daemonset

Allow master nodes to run pods : Kubectltaintnodes --all-node-role.kuernetes.io/master-

Kubectl Command Cheat Sheet

Kubectl Command Cheat Sheet



References

1. https://kubernetes.io/
2. https://github.com/norberto-enomoto/workshop-k8s