



DevOps series

Kubernetes





Somkiat Puisungnoen

Search

Somkiat | Home

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานาญกิจ พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata



somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button



Software delivery

Design

Develop

Test

Deploy

Docker and Kubernetes

Monitoring and Observability



Section 1 (Container and Docker)

1. Introduction to containers
2. Container design patterns
3. Manage containers with Docker
4. Docker ecosystem
5. Workshop with Docker



Section 2 (Kubernetes)

1. Introduction to Kubernetes
2. Kubernetes architecture
3. Components in Kubernetes
4. Install and config Kubernetes cluster
5. Networking
6. Stateful application
7. Security and Monitoring cluster



**[https://github.com/up1/
course-kubernetes](https://github.com/up1/course-kubernetes)**



Software Delivery

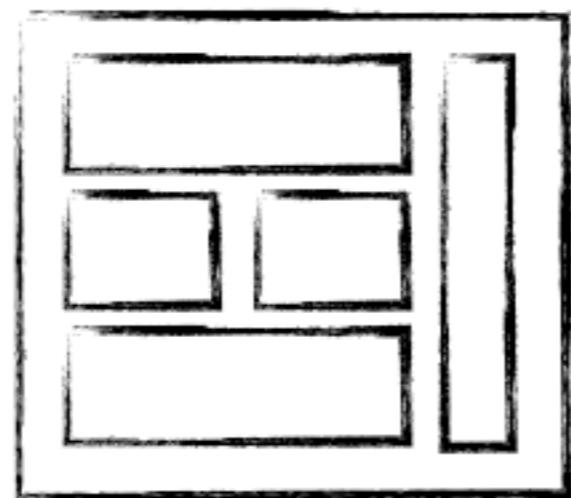


Manage containers



Why docker ?

Software industry has changed !!



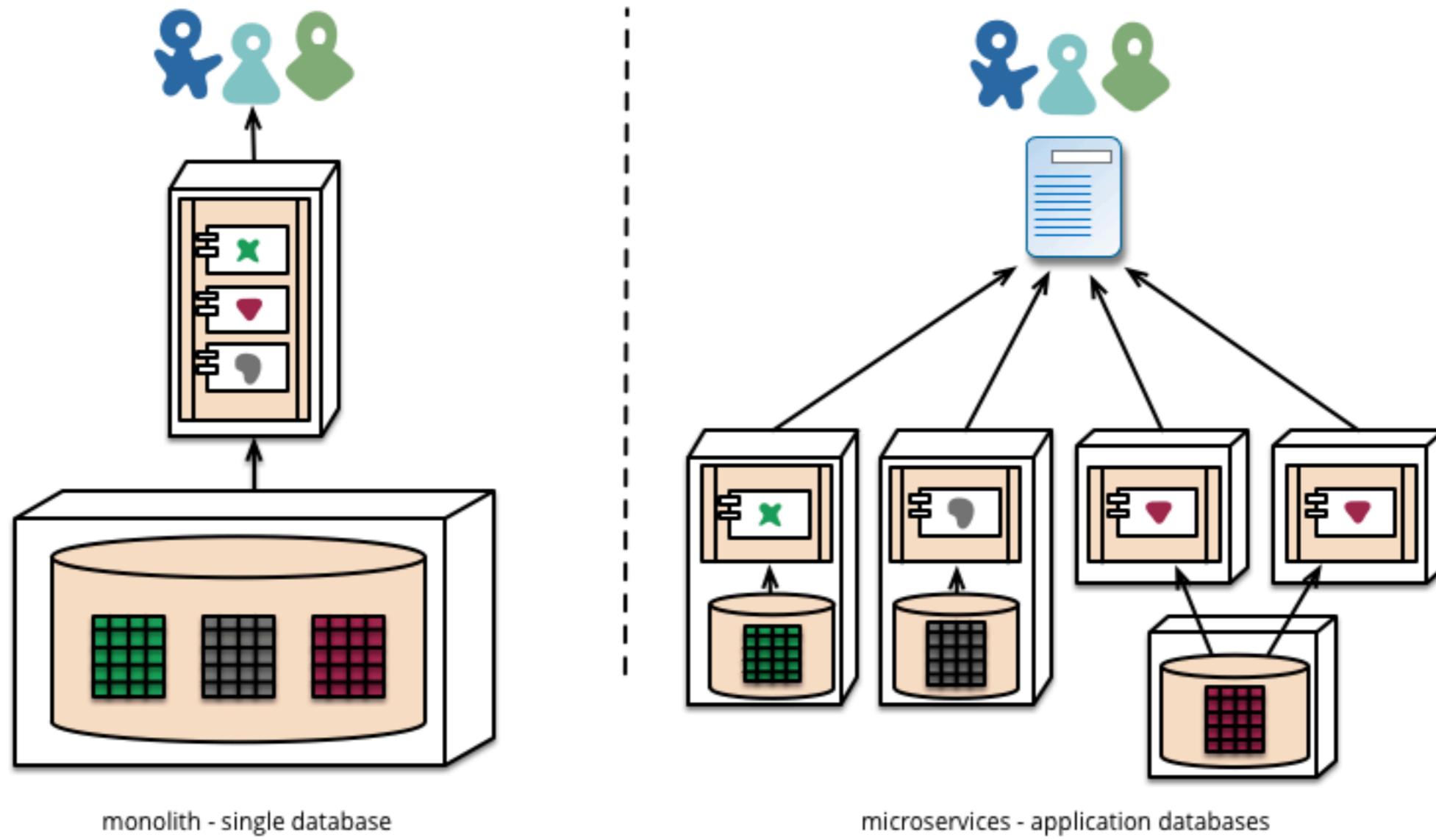
MONOLITHIC/LAYERED



MICRO SERVICES



Microservice architecture



<https://martinfowler.com/articles/microservices.html>

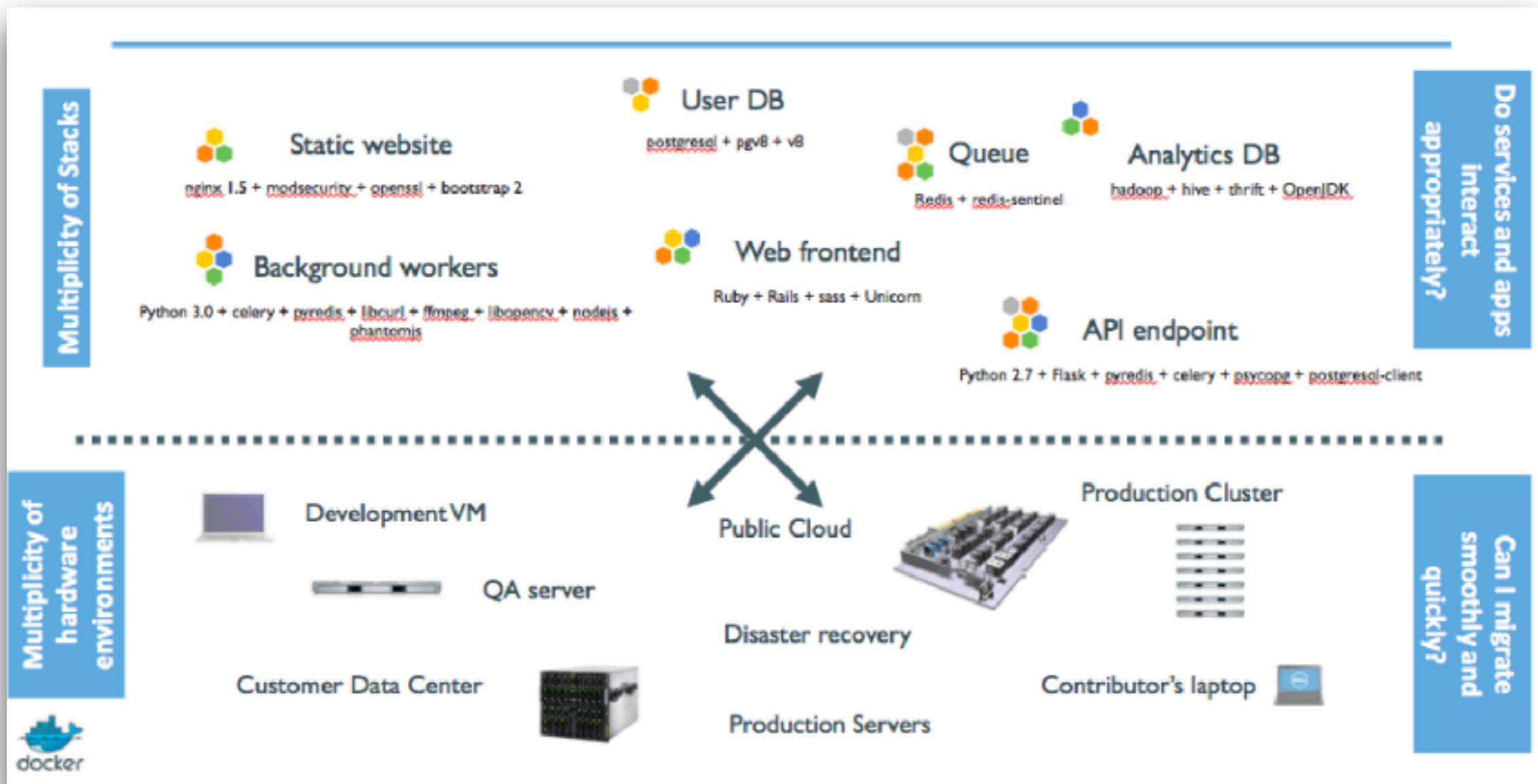


Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Why docker ?

We have problem !!



Kubernetes

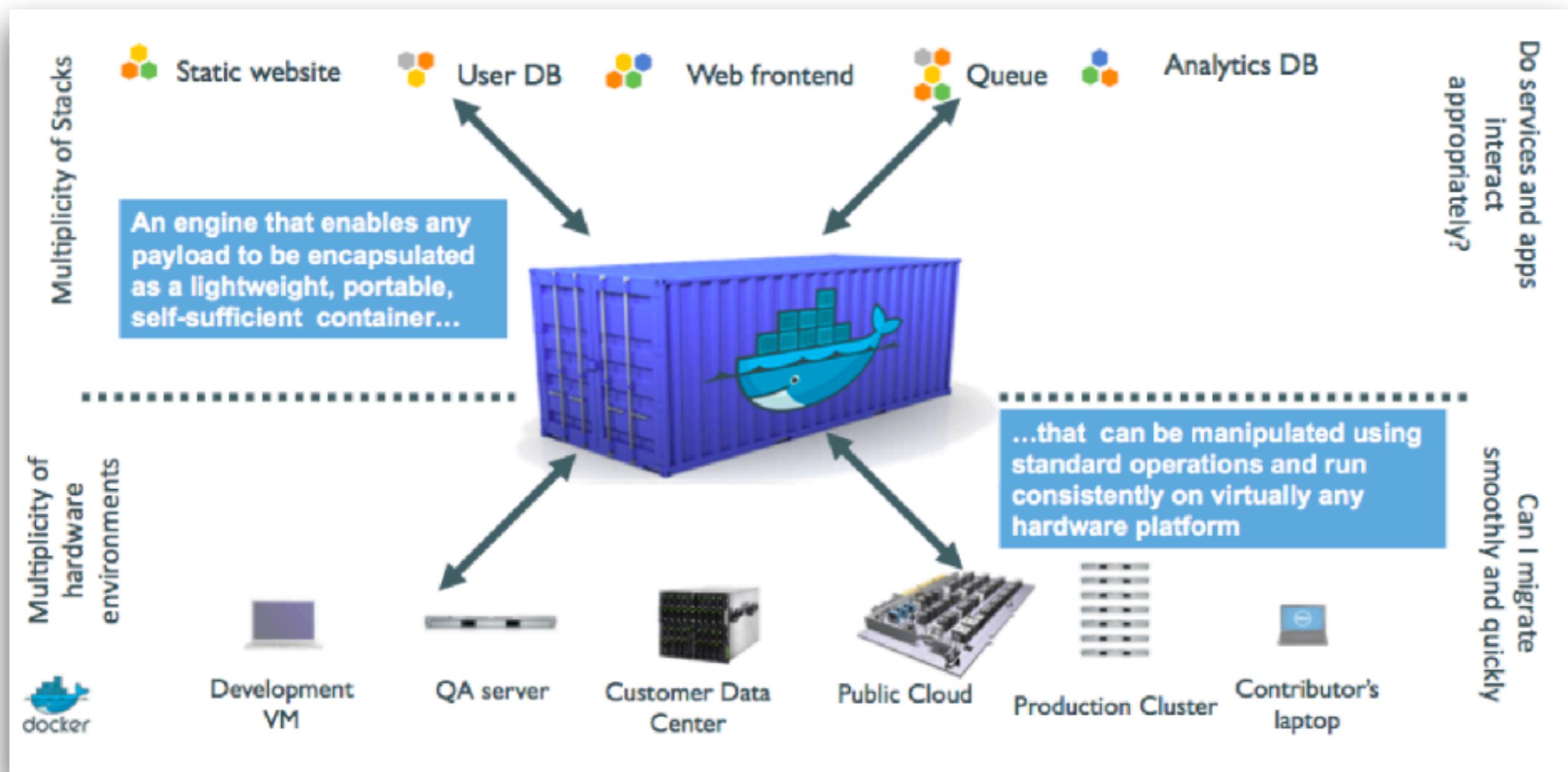
© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Problem ?

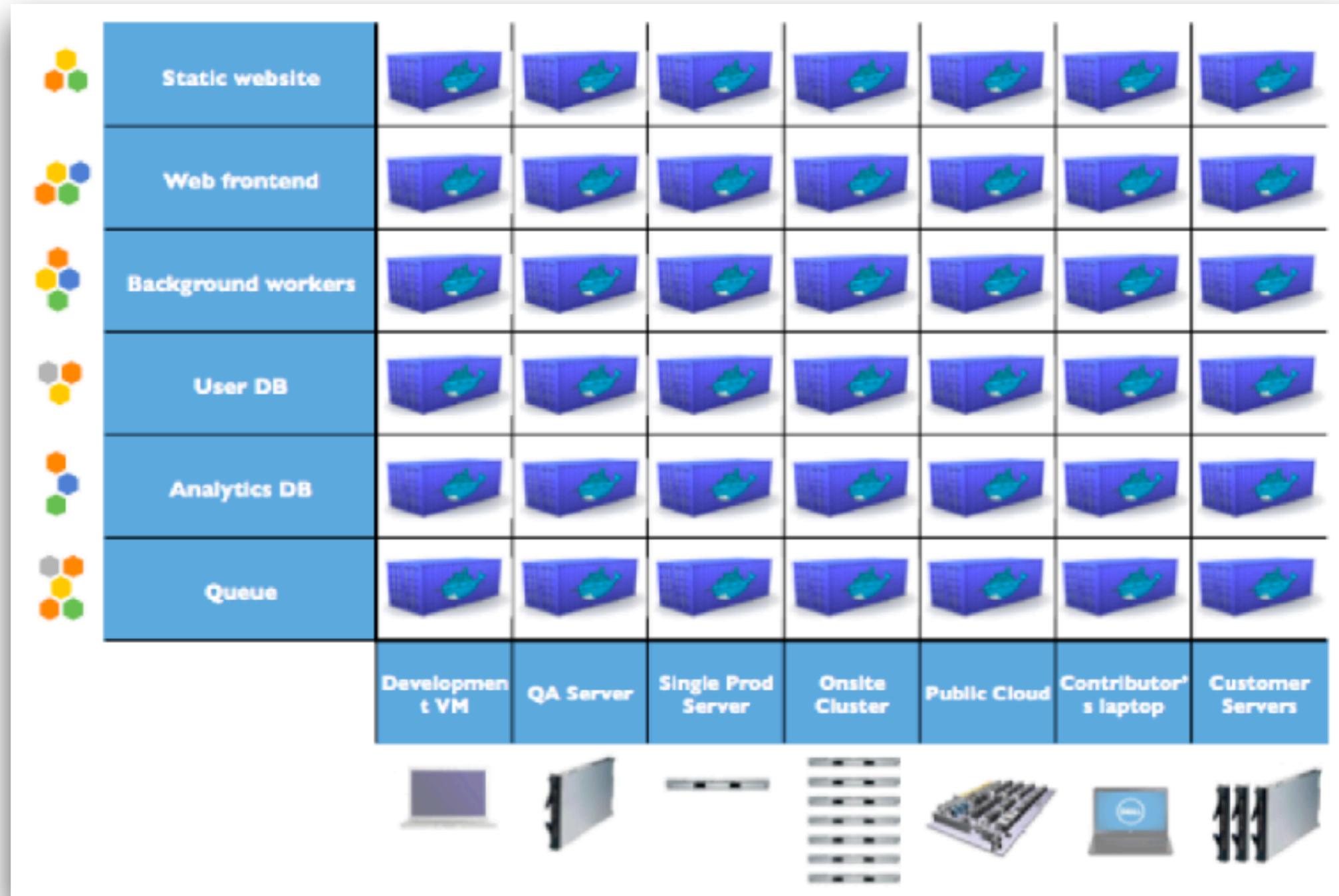
Static website	?	?	?	?	?	?	?
Web frontend	?	?	?	?	?	?	?
Background workers	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
	Development VM	QA Server	Single Prod Server	Onsite Cluster	Public Cloud	Contributor's laptop	Customer Servers



Solution ?

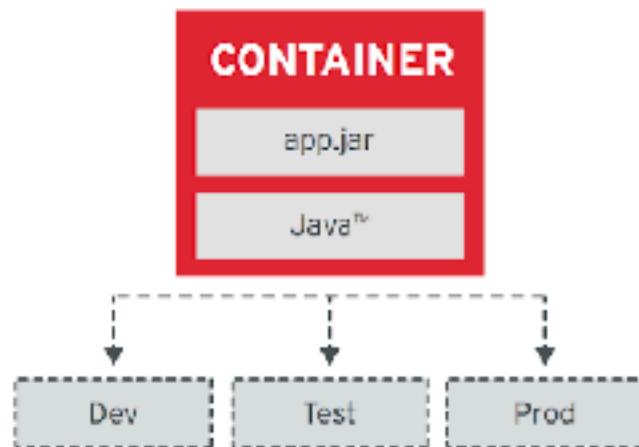


Solution ?



Container Design Principles

Image Immutability Principle



High Observability Principle



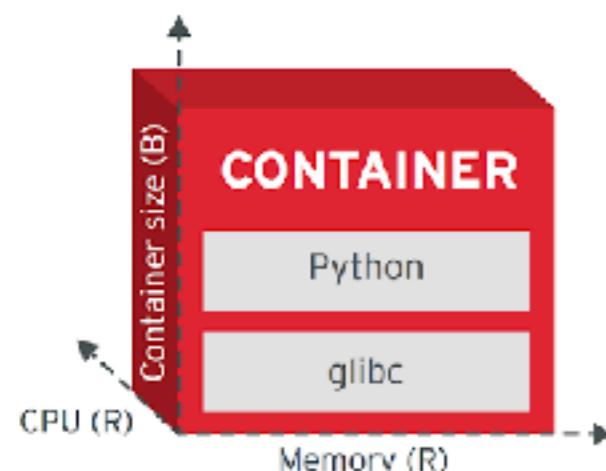
Process Disposability Principle



Lifecycle Conformance Principle



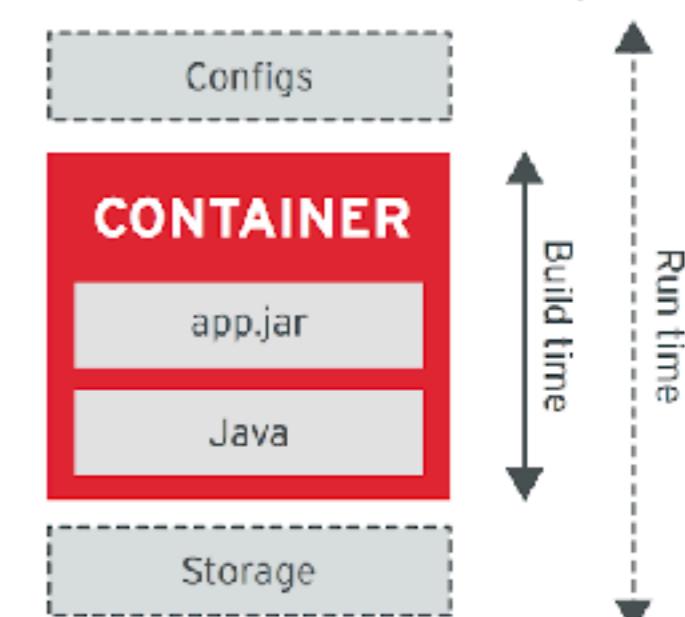
Runtime Confinement Principle



Single Concern Principle



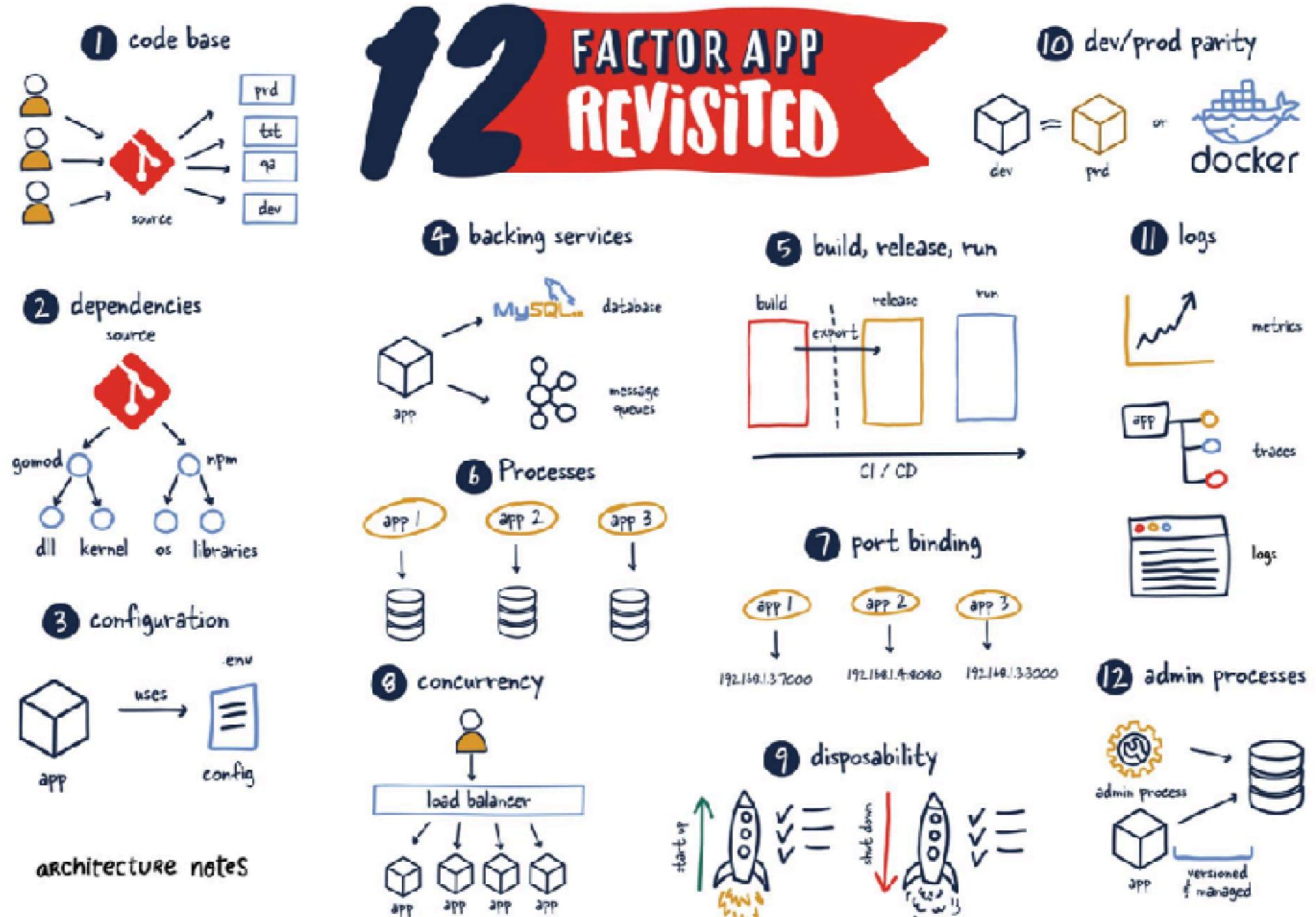
Self-Containment Principle



<https://kubernetes.io/blog/2018/03/principles-of-container-app-design/>



The Twelve-Factor App



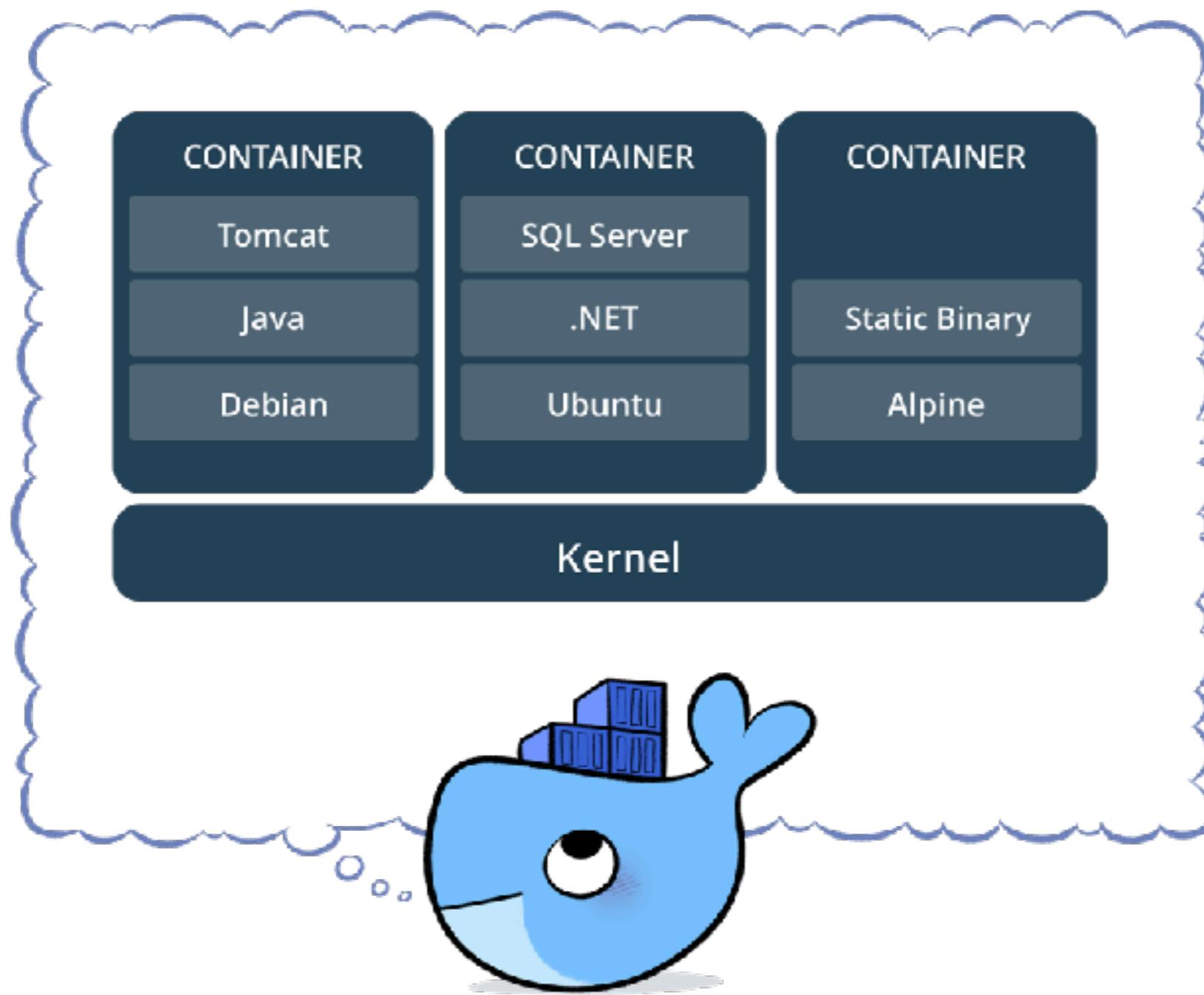
<https://12factor.net/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Container with Docker



Basic of Docker

**Image
Container
Dockerfile
Registry
Volume and network**



Image vs Container

Image like template/blueprint
Containers are created from image



Docker image

Collection of files and some meta data

Made of **layers**

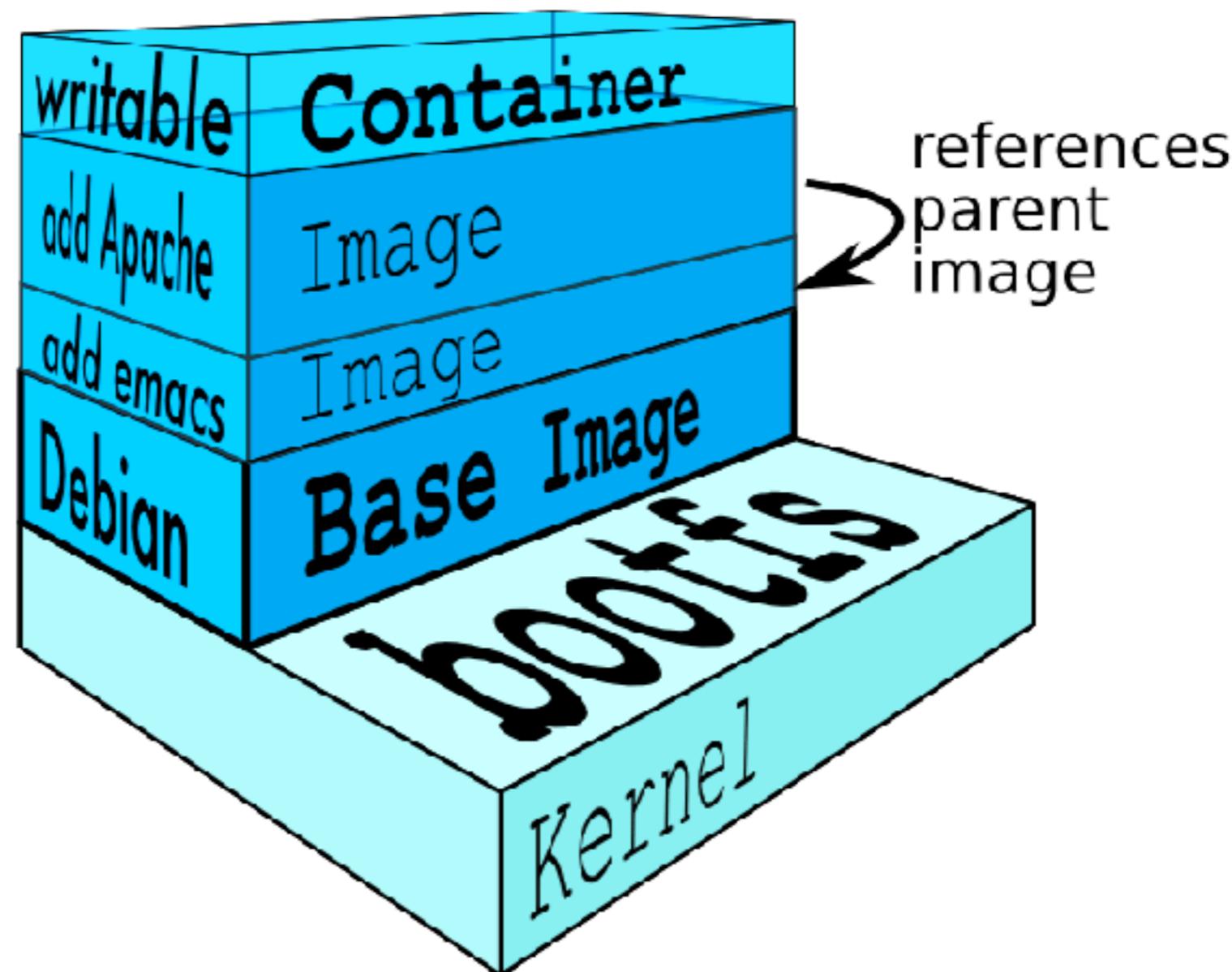
Each layer can add/change/remove files

Image can share layers

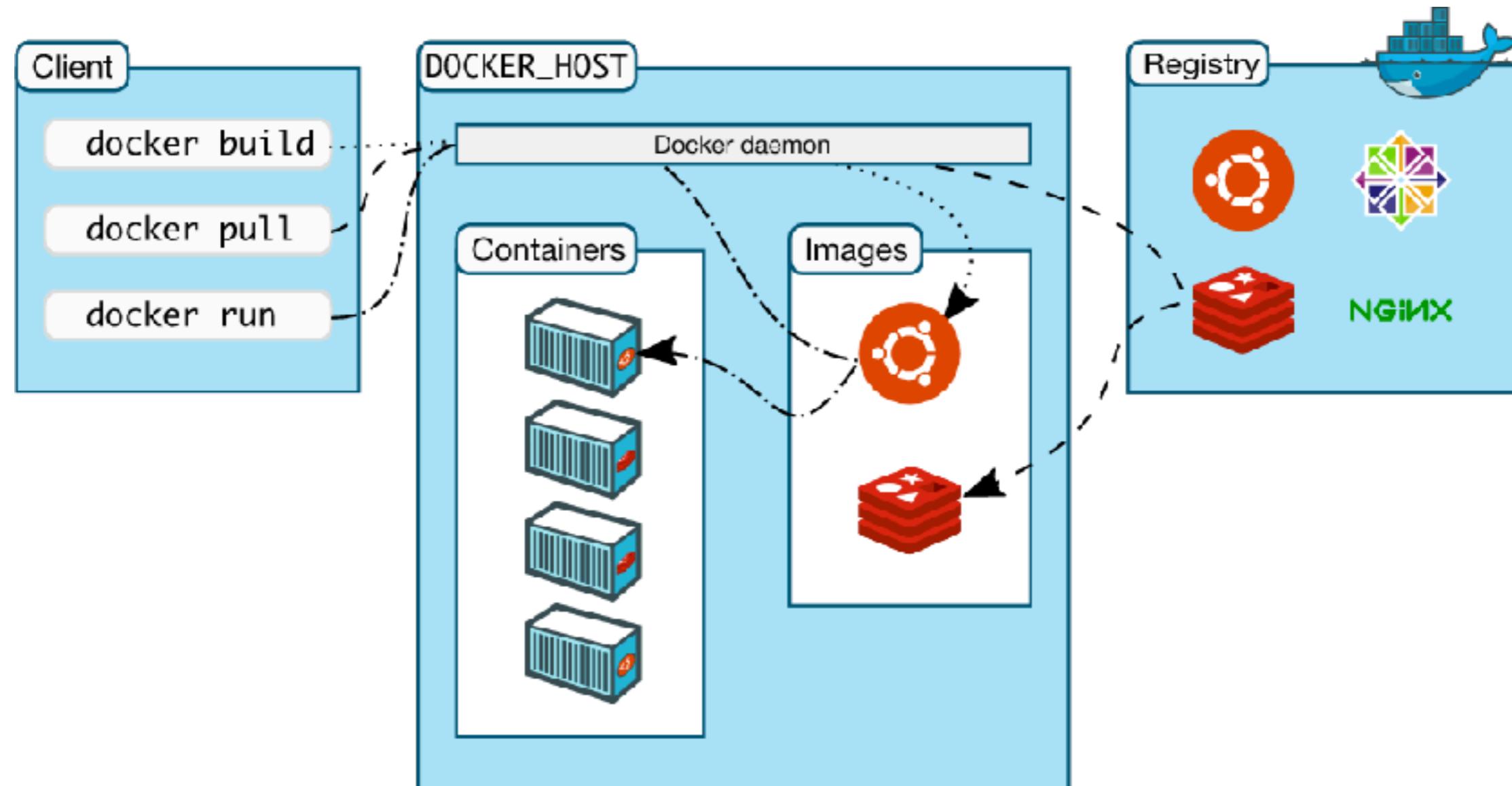
Read-only file system



Docker image layer



How docker works ?



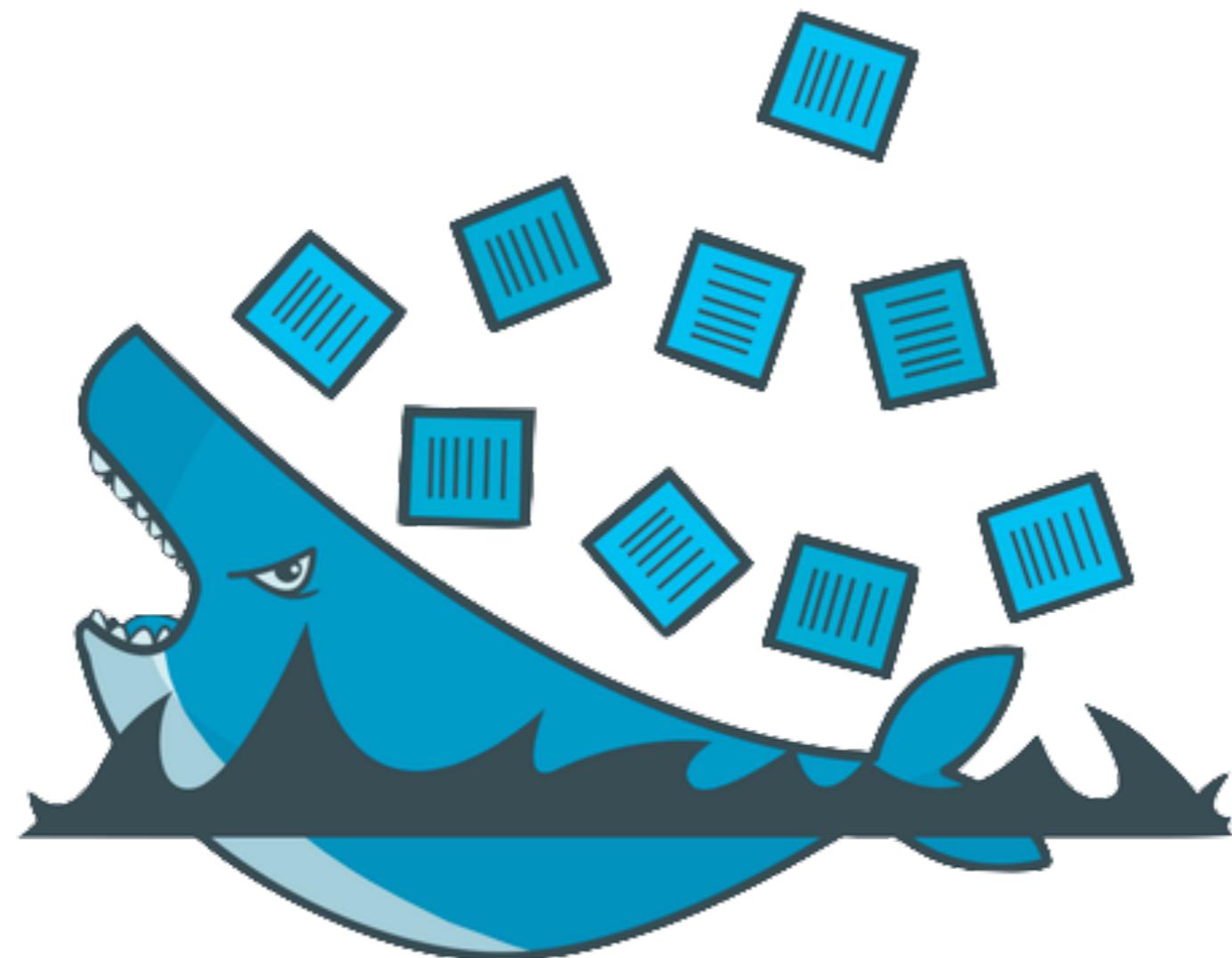
<https://docs.docker.com/get-started/overview/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Workshop



Working with Docker compose

<https://docs.docker.com/compose/>



Multiple containers app!!

Difficult to create and manage !!

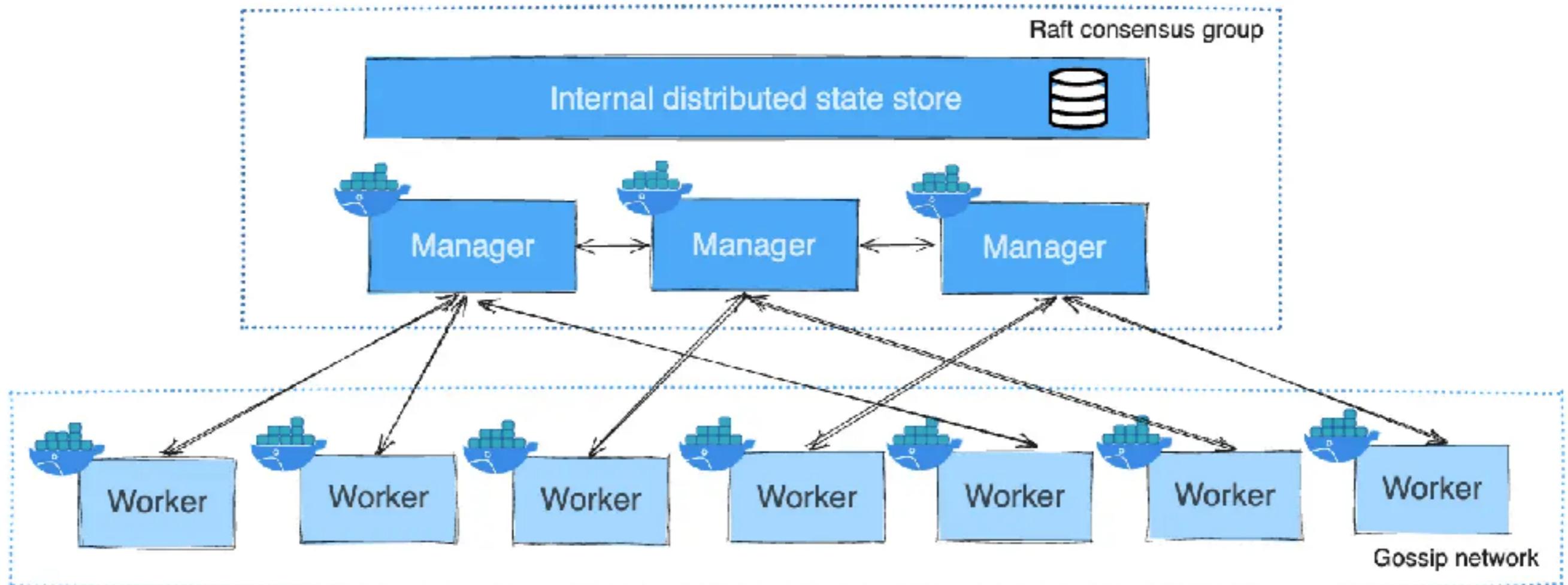


Mutilple containers app!!

Build images from Dockerfile
Pull images from Hub/private/cache
Configuration and create containers
Start and stop containers
Stream their logs



Docker Swarm



<https://docs.docker.com/engine/swarm/how-swarm-mode-works/nodes/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Introduction



kubernetes





Why we need Kubernetes ?

Managing containers for production is challenging
Need something to manage beyond a container
engine !



Key capabilities was missing !!

Using multiple containers with **shared** resources

Monitoring running containers

Handling dead containers

Autoscaling container instances to handle load

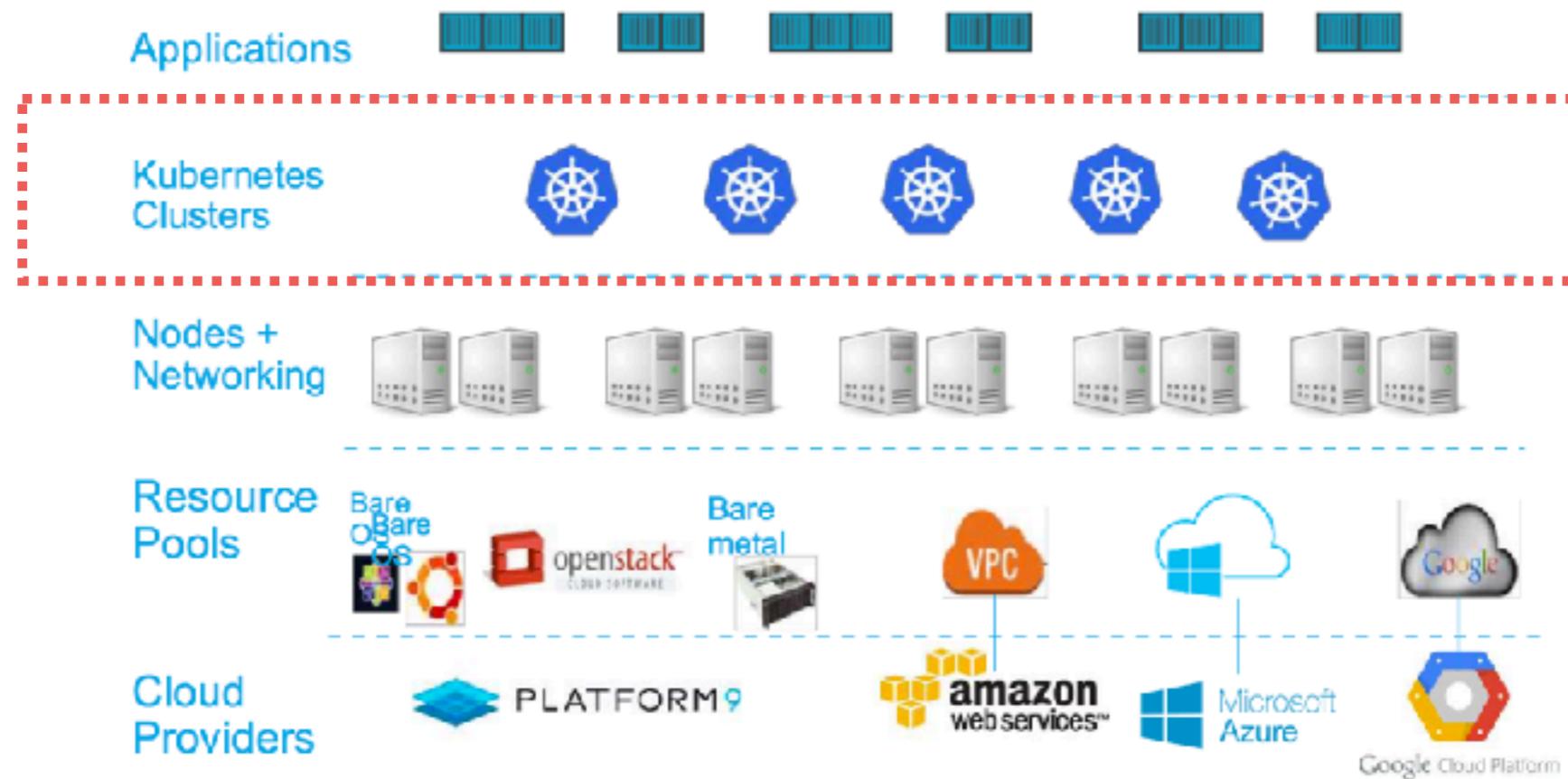
Making the container services **easily** accessible

Handling dead connecting containers to a variety of external data sources



Write once, run anywhere

Eliminate infrastructure lock-in
Use containers
Provides management for containers



Deployment, not infrastructure

Software deployment is hard
Infrastructure provisioning/re-provisioning
Configuration networking and load balance
Redundancy (scale-out)
Lifecycle management (Software update)



K8s support for deployment

Scale-out service

Rolling update for new version

Rollback to a previous version

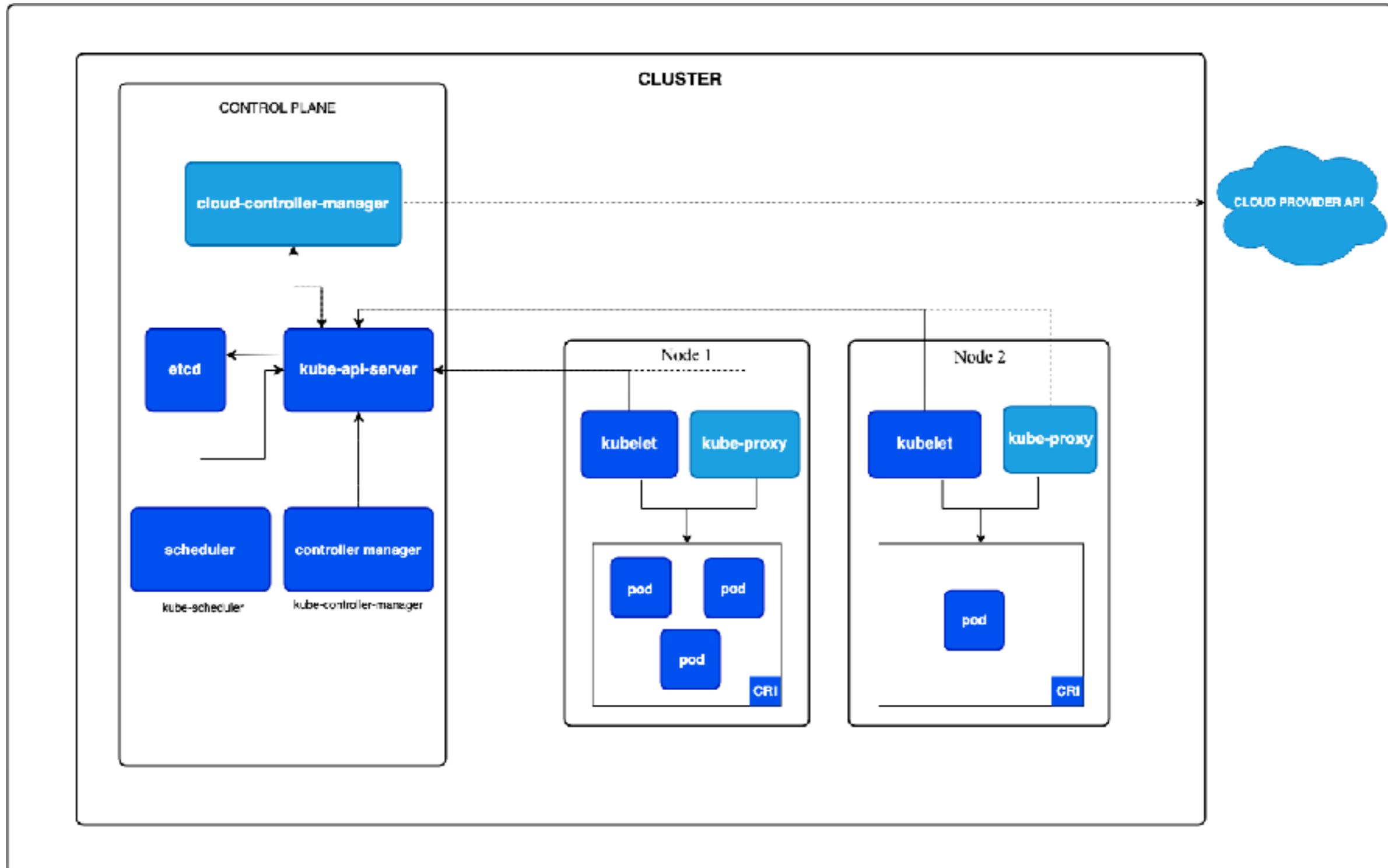
Pause and resume a deployment

Horizontal auto-scaling

Canary deployment



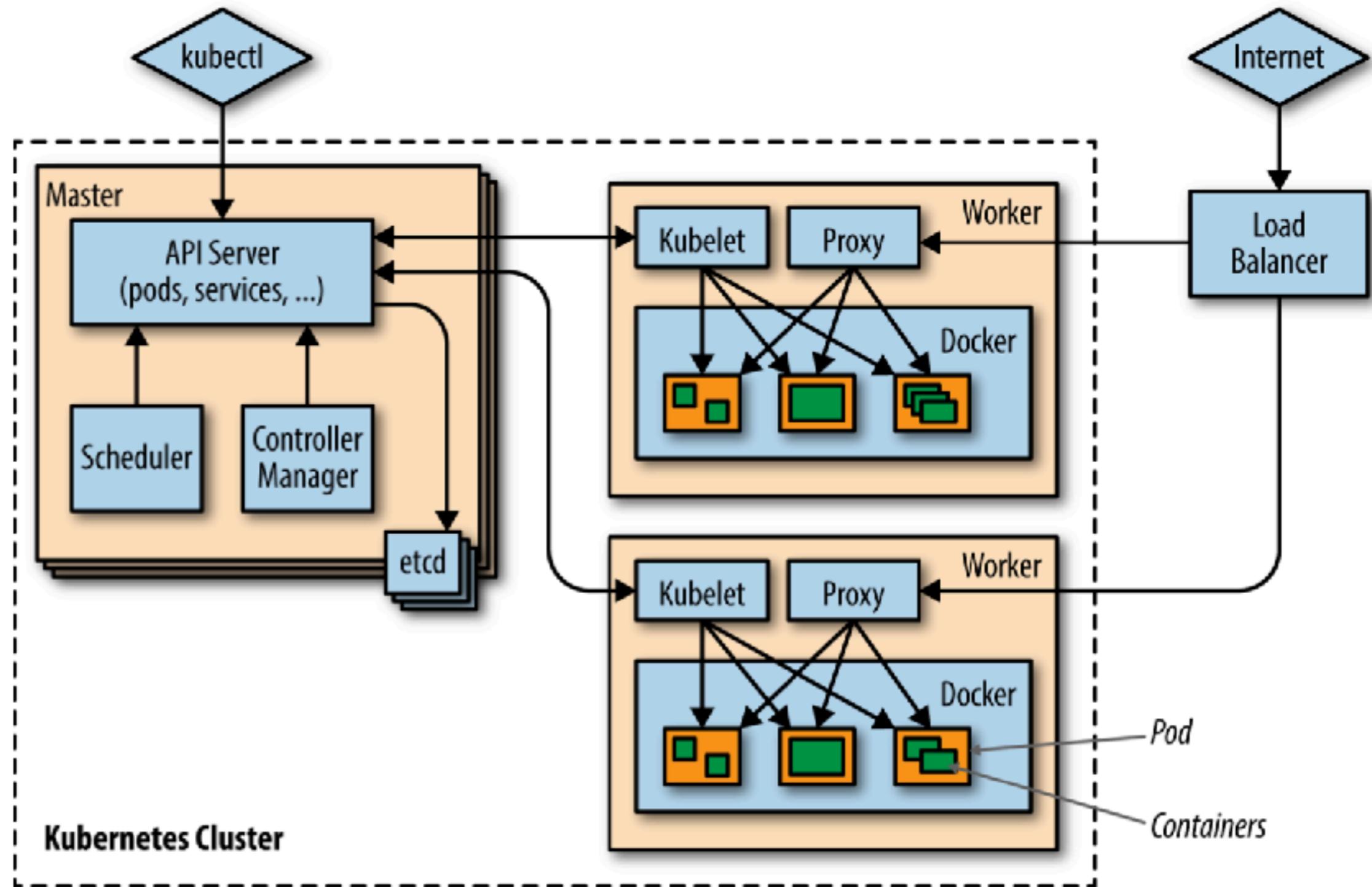
Kubernetes Architecture



<https://kubernetes.io/docs/concepts/architecture/>

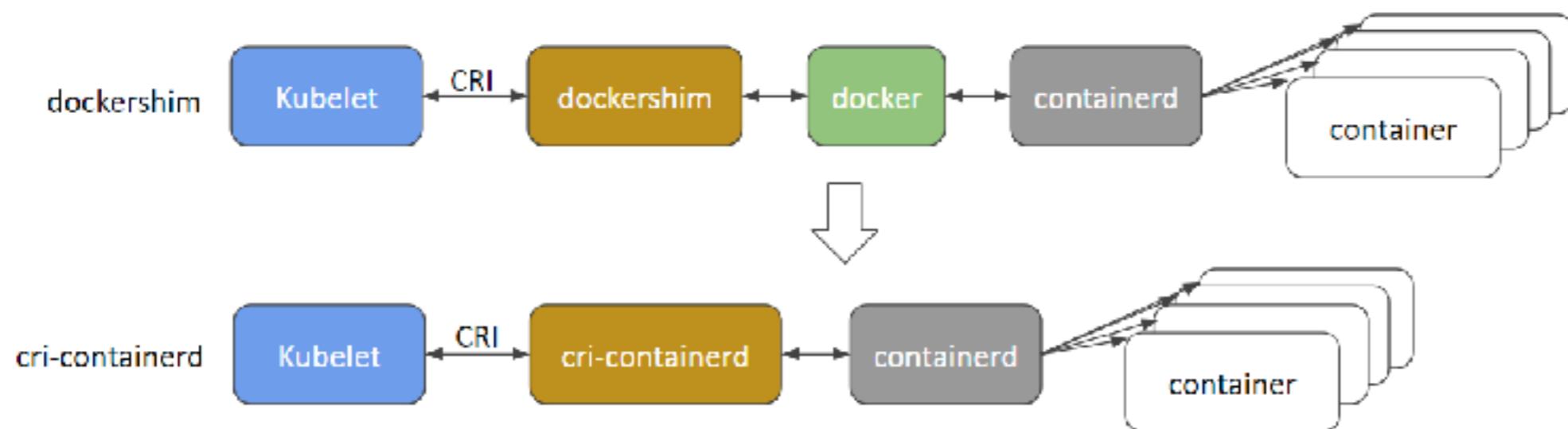


Kubernetes Architecture



Container runtime ?

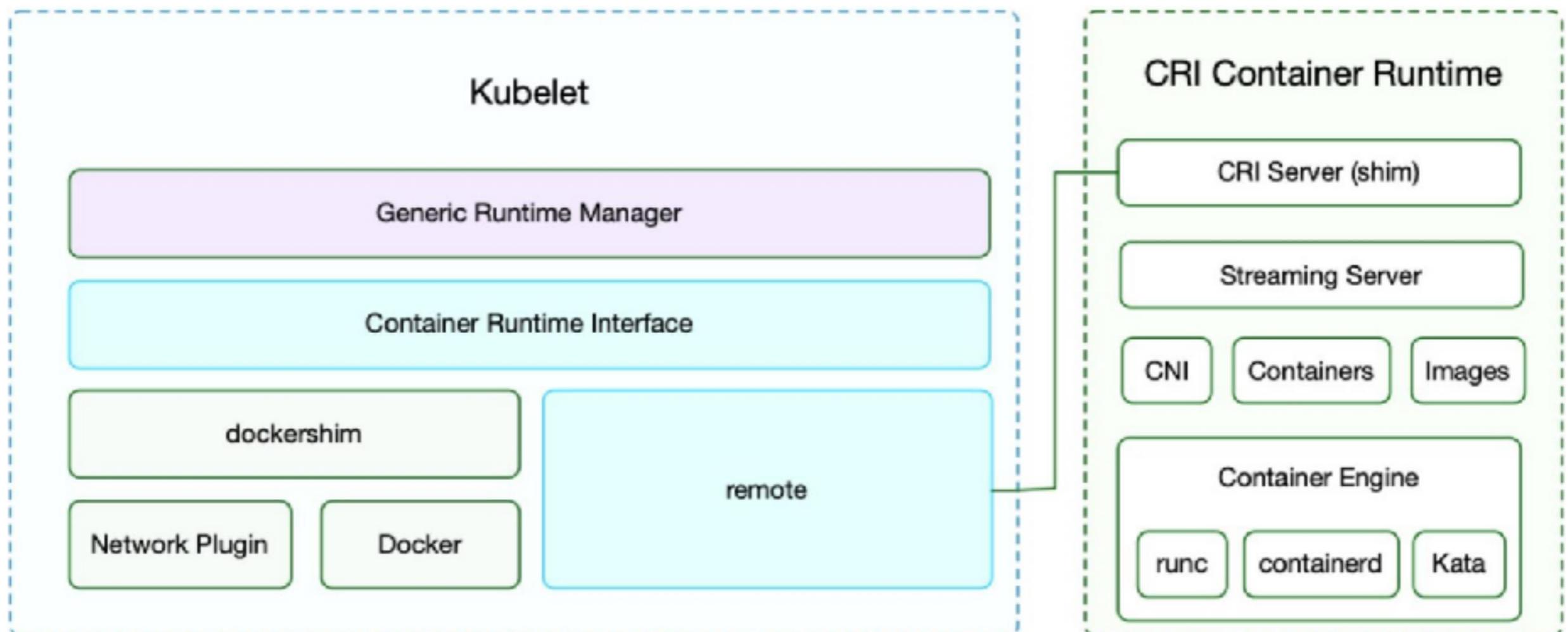
Need to install container runtime into every nodes
Support Container Runtime Interface (CRI)
Kubernetes removed dockershim !!



<https://kubernetes.io/docs/setup/production-environment/container-runtimes/>



Container Runtime Interface



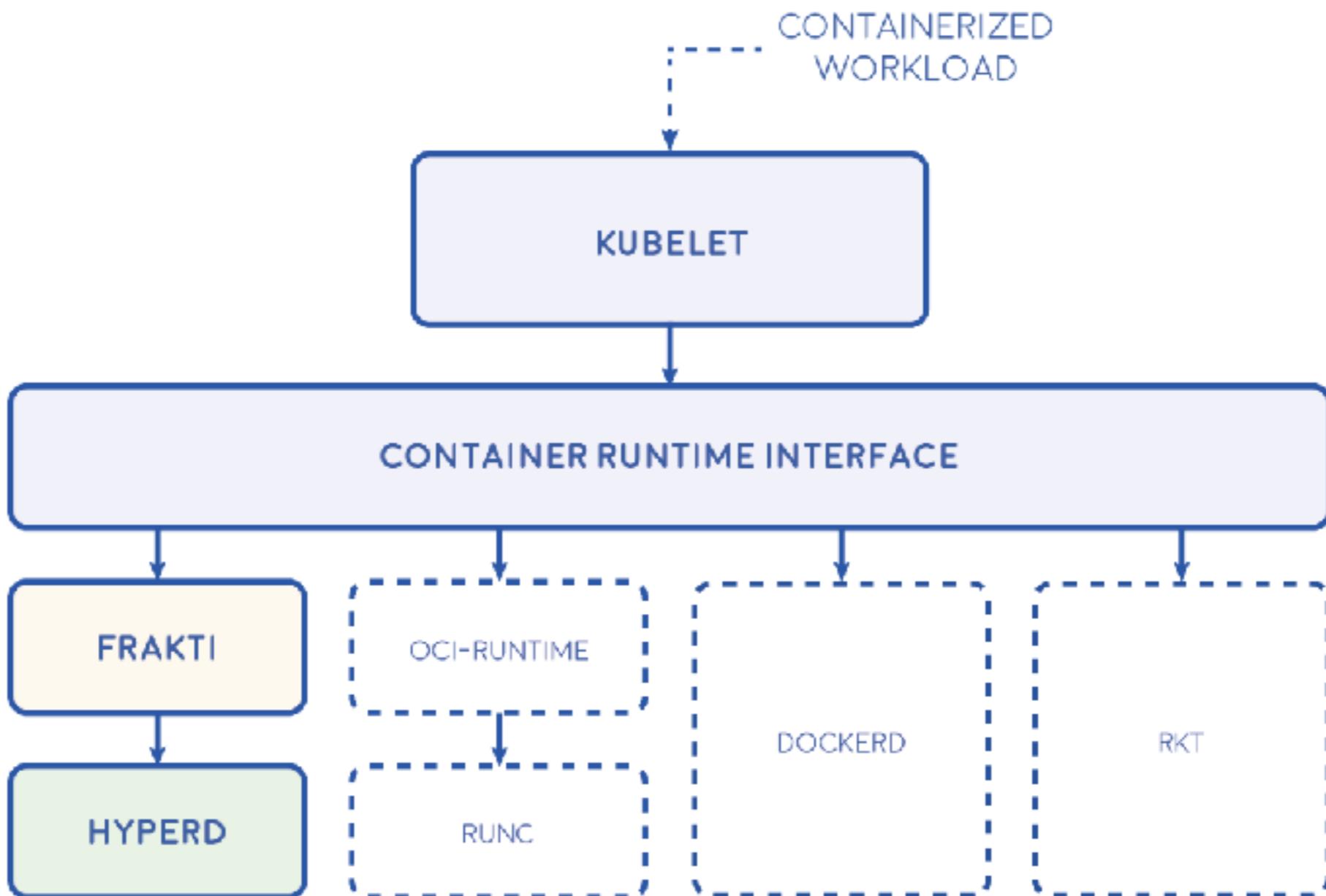
<https://kubernetes.io/docs/setup/production-environment/container-runtimes/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Container Runtime Interface



<https://www.devopsschool.com/blog/list-of-top-container-runtime-interface-projects/>

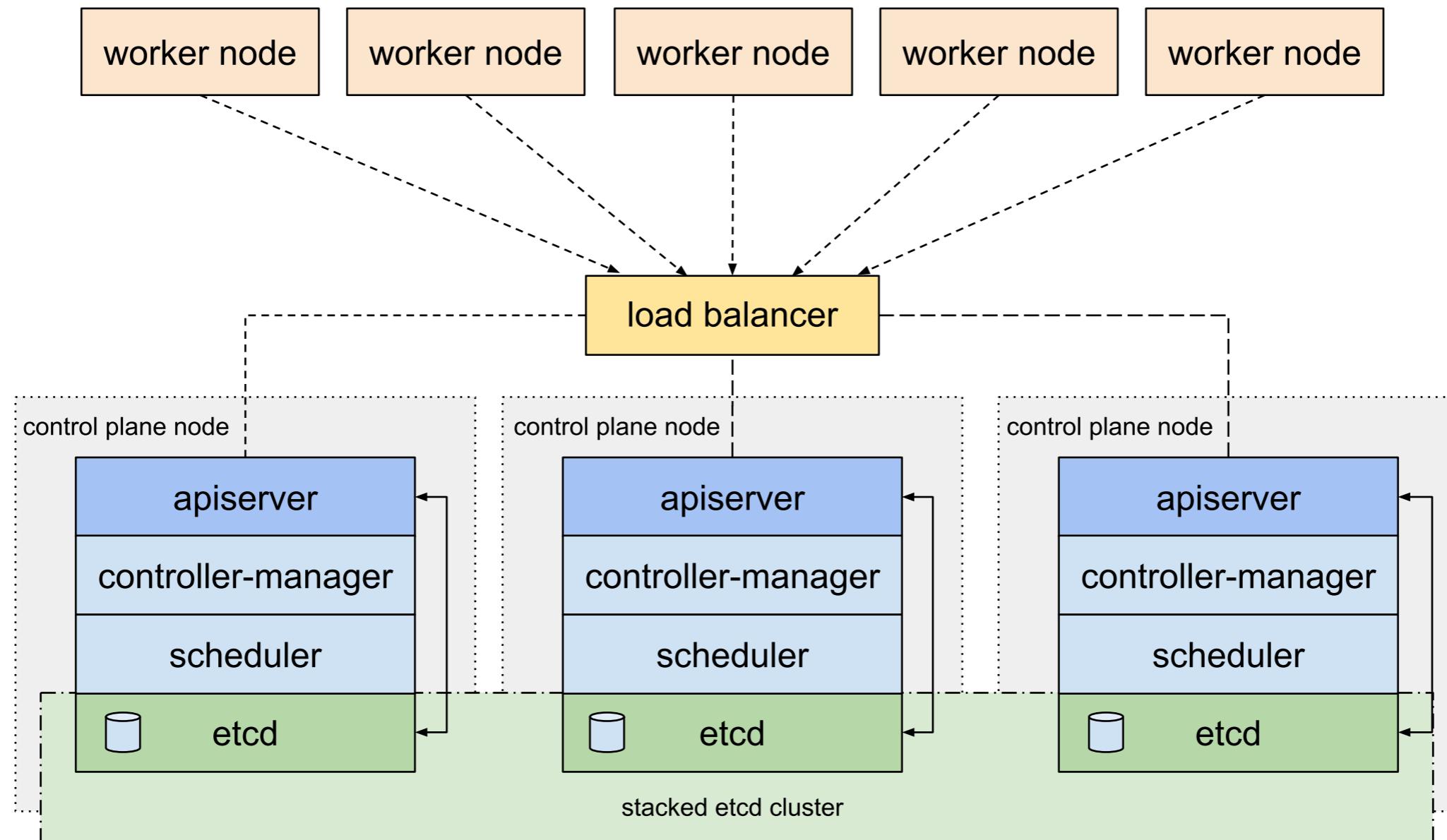


High Availability Cluster



High Availability Cluster (1)

kubeadm HA topology - stacked etcd



<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/ha-topology/>

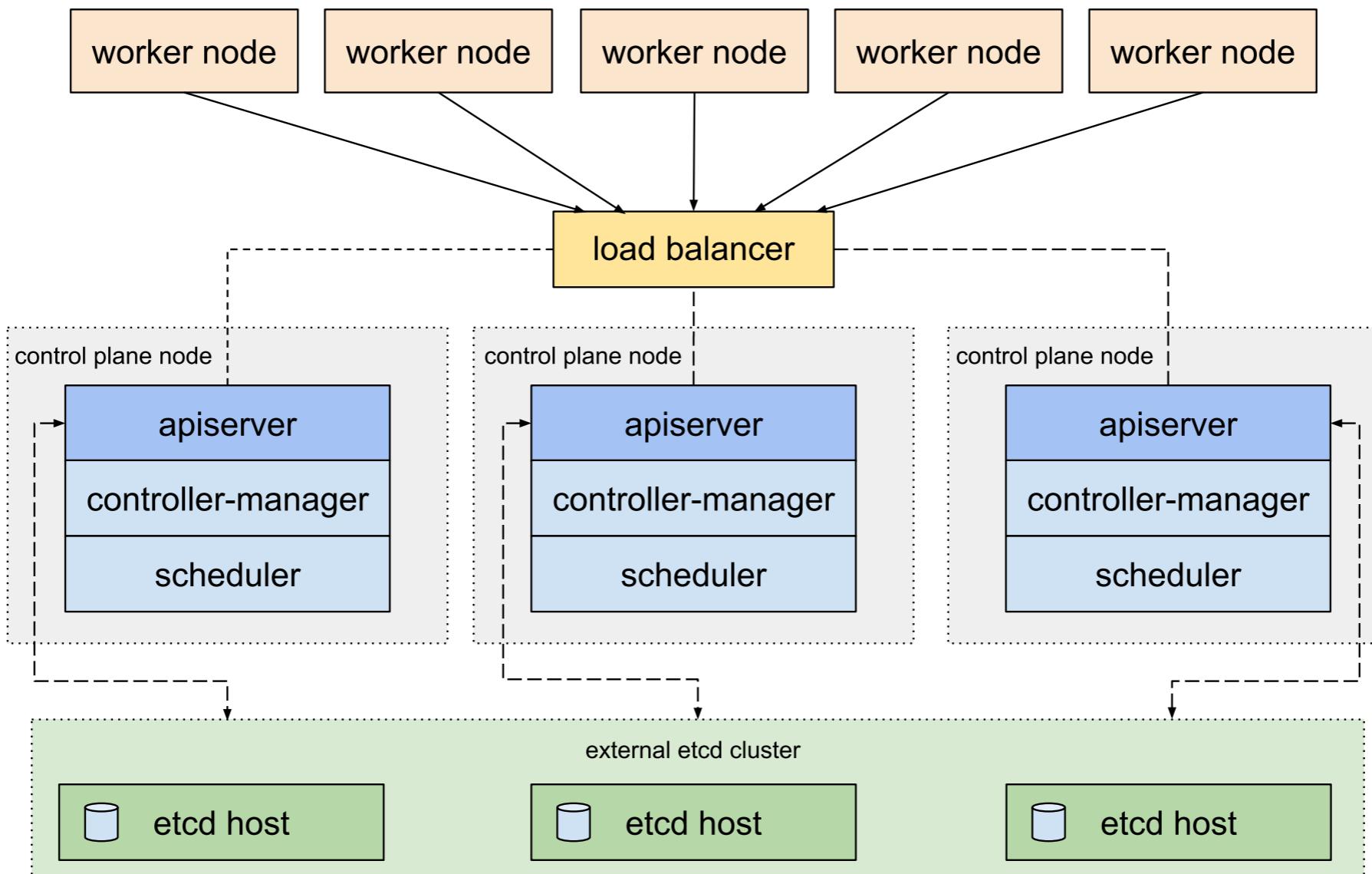


Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

High Availability Cluster (2)

kubeadm HA topology - external etcd



<https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/ha-topology/>

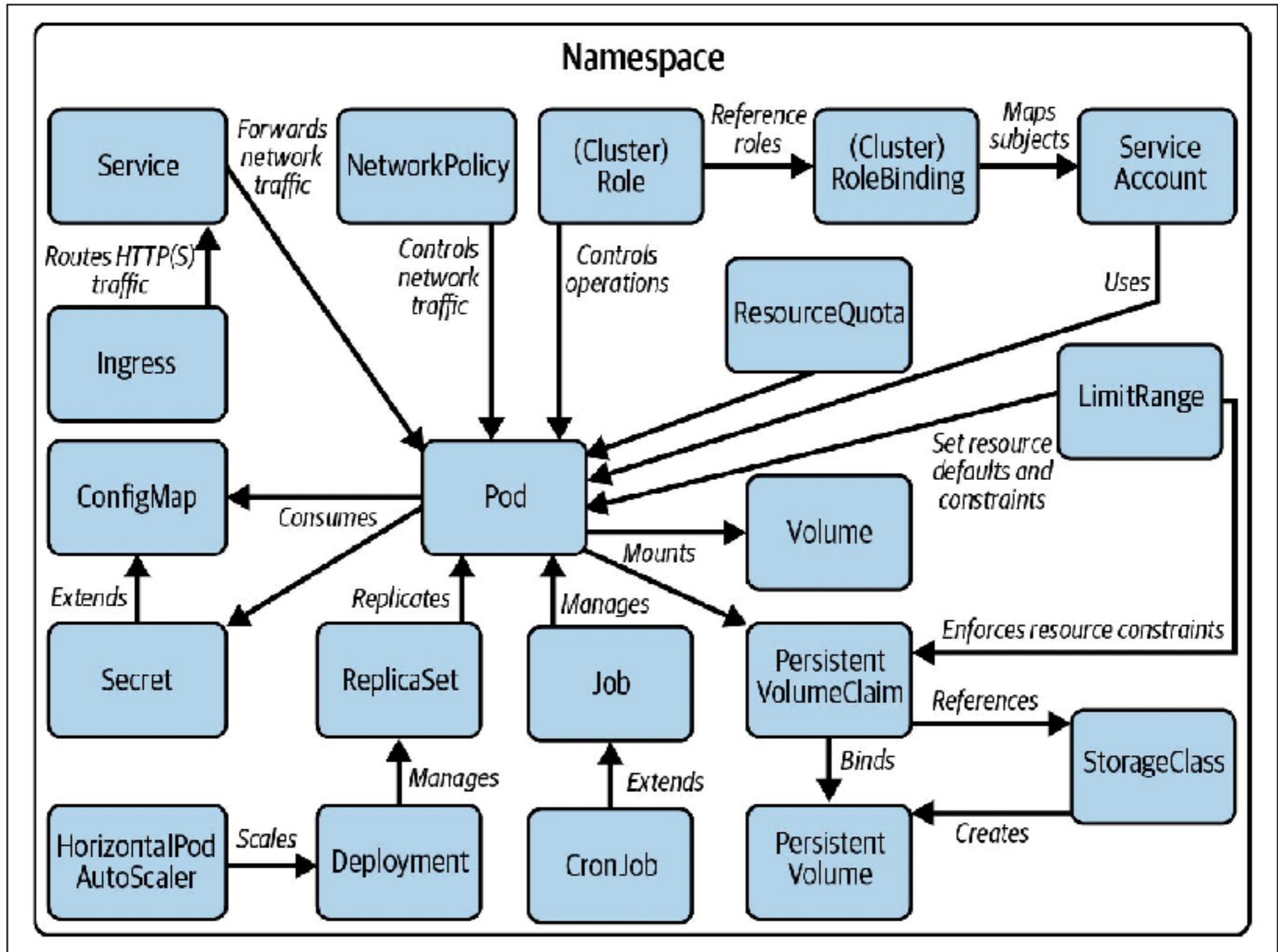


Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

K8s components





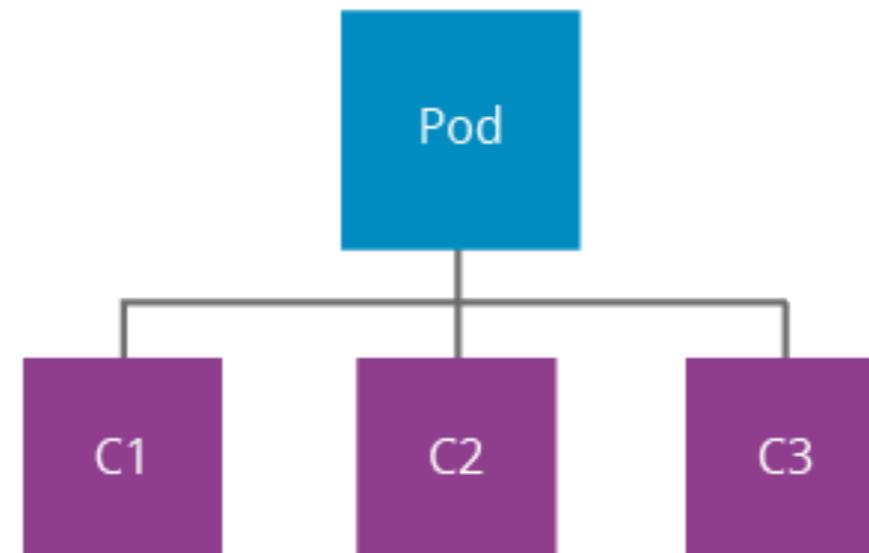
K8s components

Pods
Services
Deployments
ConfigMap and Secret
Volumes
StatefulSets



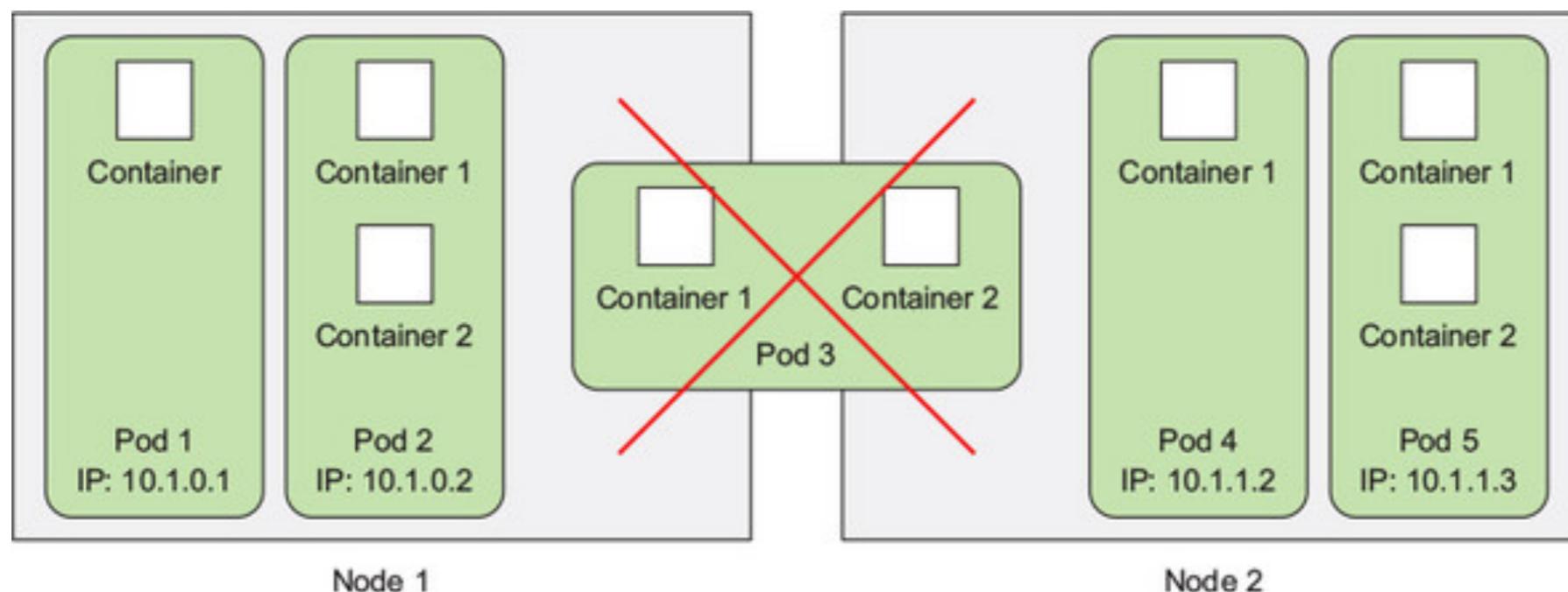
Pods

Small group of co-located containers
Optionally shared volume between containers
Basic deployment unit in Kubernetes



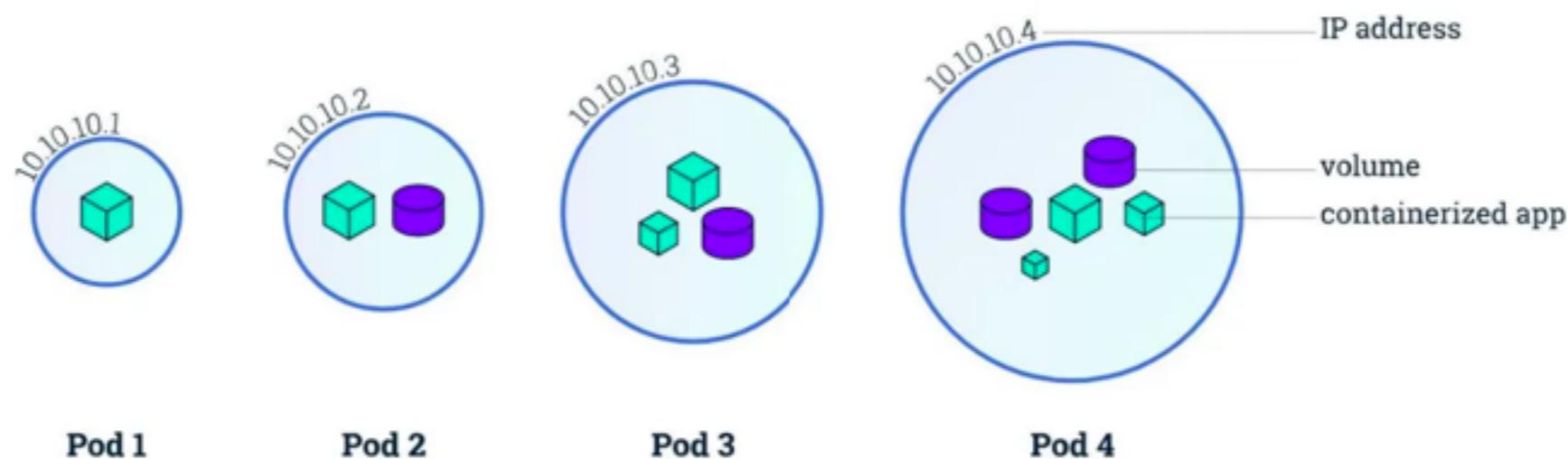
Pods

1 pods = 1 container
1 pods = N containers



All containers in same Pods

Share process ID
Share network interface
Share Hostname/IP/Port
Share Unix Time Sharing (UTS)



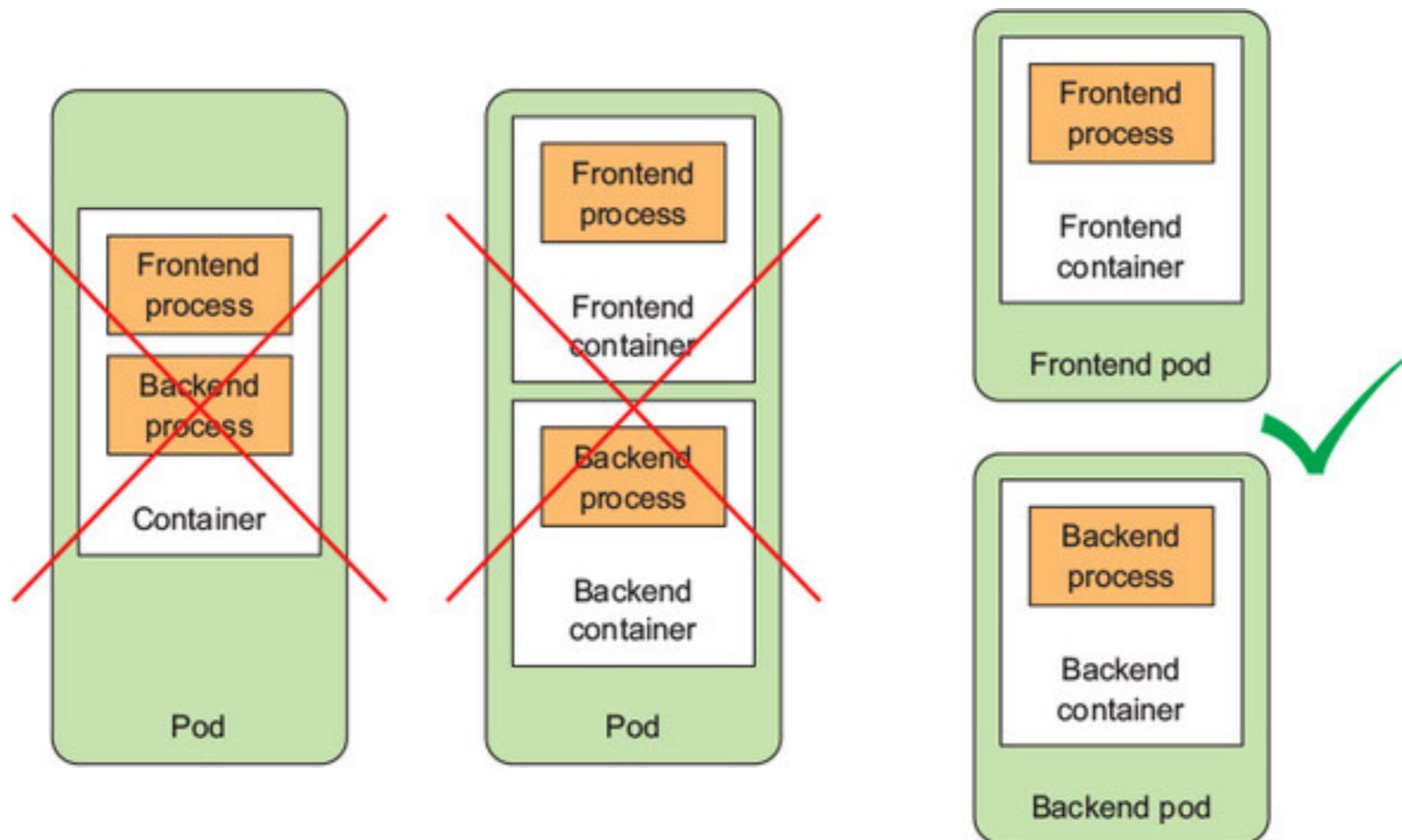
Pods Lifecycle

Phase	Description
Pending	Accepted by kubernetes but container not created yet
Running	Pods bound to the node, all containers created and at least one container is running/start/restarting
Successed	Containers exited with status 0
Failed	All containers exit and at least one exited with non-zero status
Unknow	State of Pods can't be determined due to communication issues with its node



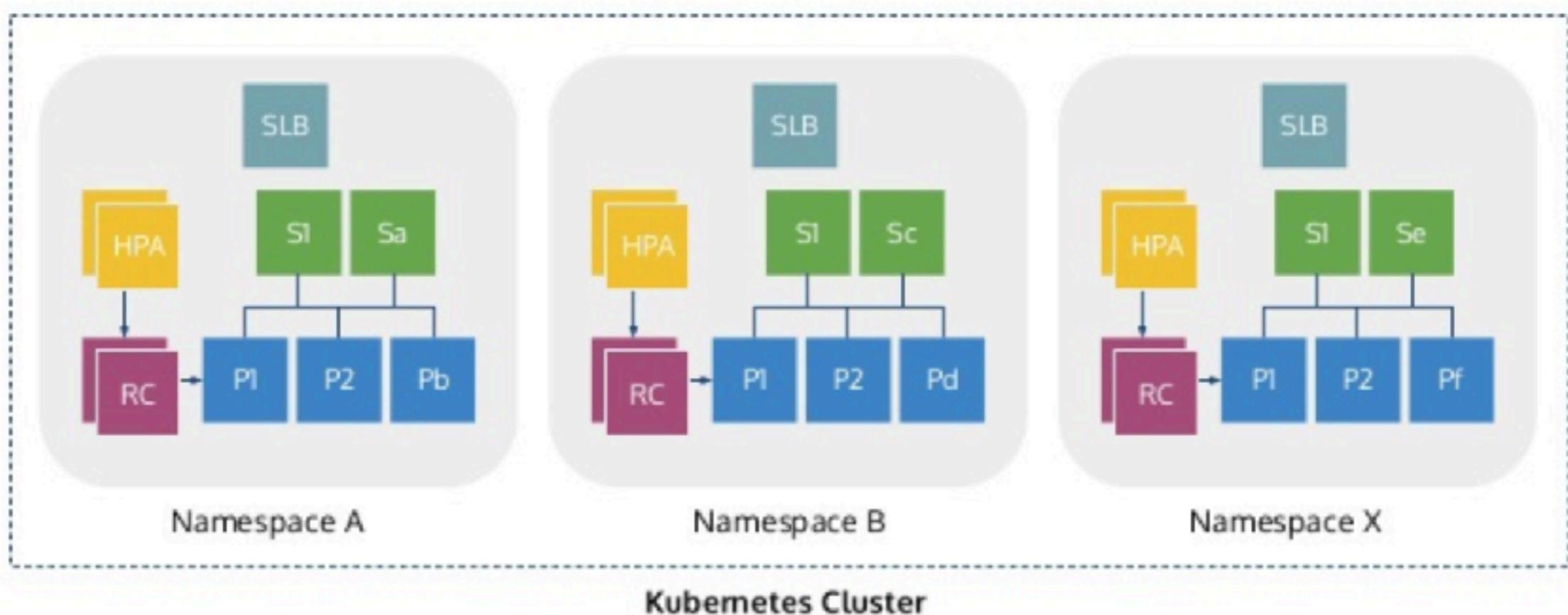
Organize container across Pods

Split multi-tiers app into multiple pods



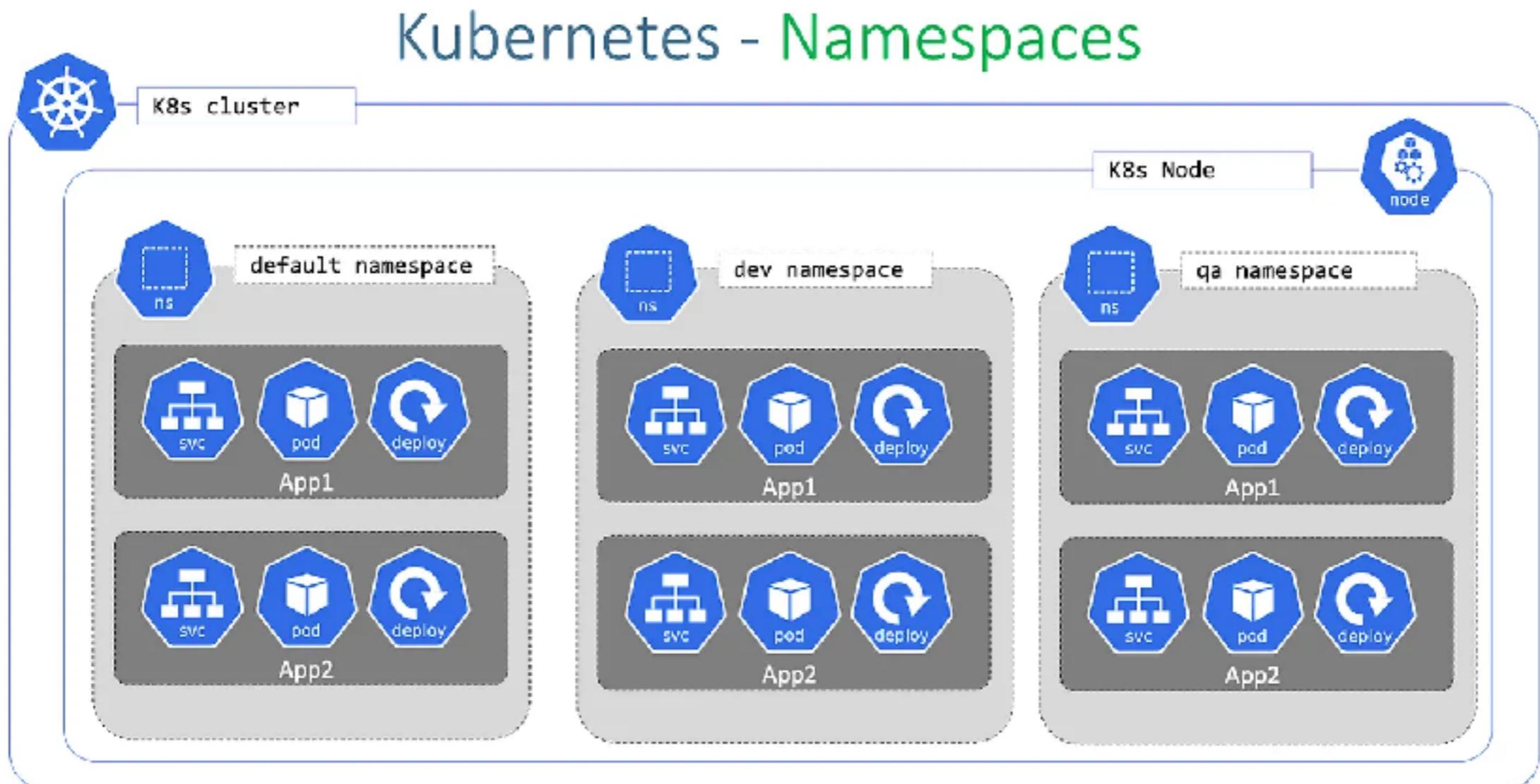
Pods Namespaces

Allow different teams to use the same cluster



Pods Namespaces

Manage by environments



<https://blog.devops.dev/day18-kubernetes-resource-quota-and-namespace-6a21045b0d97>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Deployment ReplicaSet (RS)



Deployment and ReplicaSet (RS)

Next-generation of Replication Controller (RC)
Provide function to maintain versioning of Pods

- Update new version (Rollout)
- Revert to old version (Rollback)
- Scale a deployment
- Pause/Resume process

<https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>



Kubernetes Deployment

Define the application lifecycle
How to update application ?



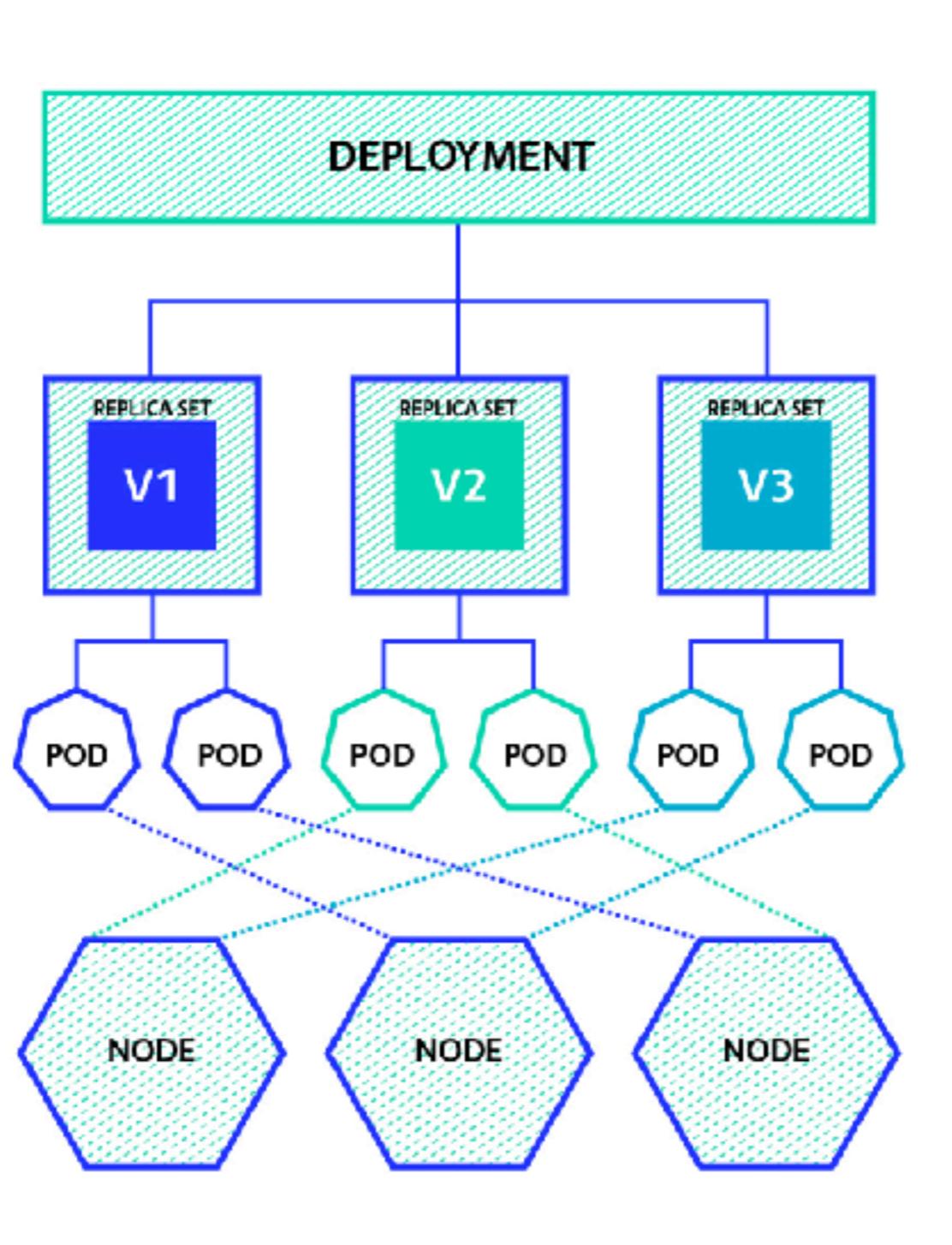
<https://www.plural.sh/blog/kubernetes-deployments-guide/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Deployment and ReplicaSet



Deployment strategies ?

Control the behavior of Pod replacement

Ensure availability

Minimal disruption



Deployment strategies ?

**Rolling update
(Default)**

Recreate

Blue/Green

Canary

A/B testing

Ramped slow
rollout

Shadow
deployment

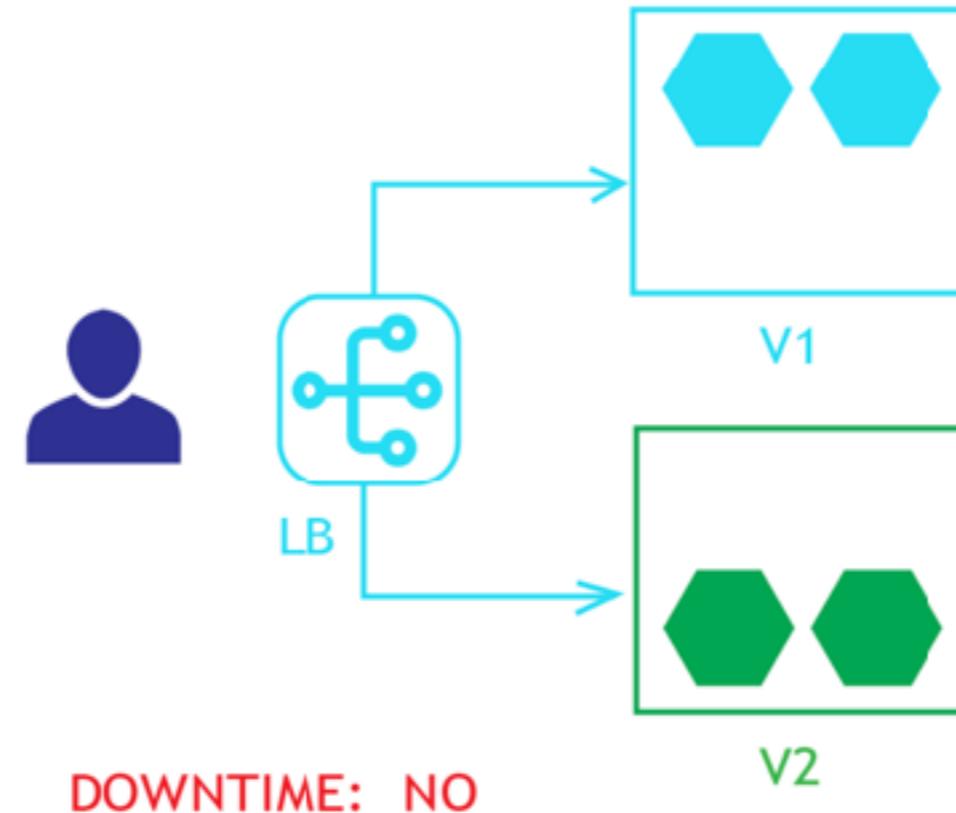
Best-effort
controlled rollout

<https://spacelift.io/blog/kubernetes-deployment-strategies>



Rollout update (default)

Gradually replaces old pods with new ones
Minimize downtime

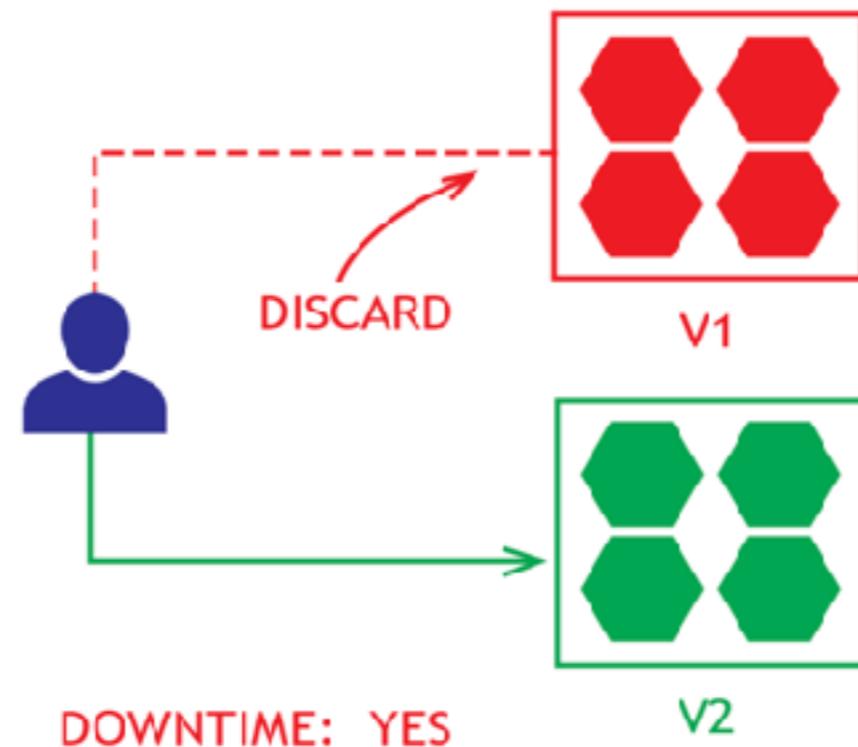


<https://2cloud.io/blog/kubernetes-deployment-strategies>



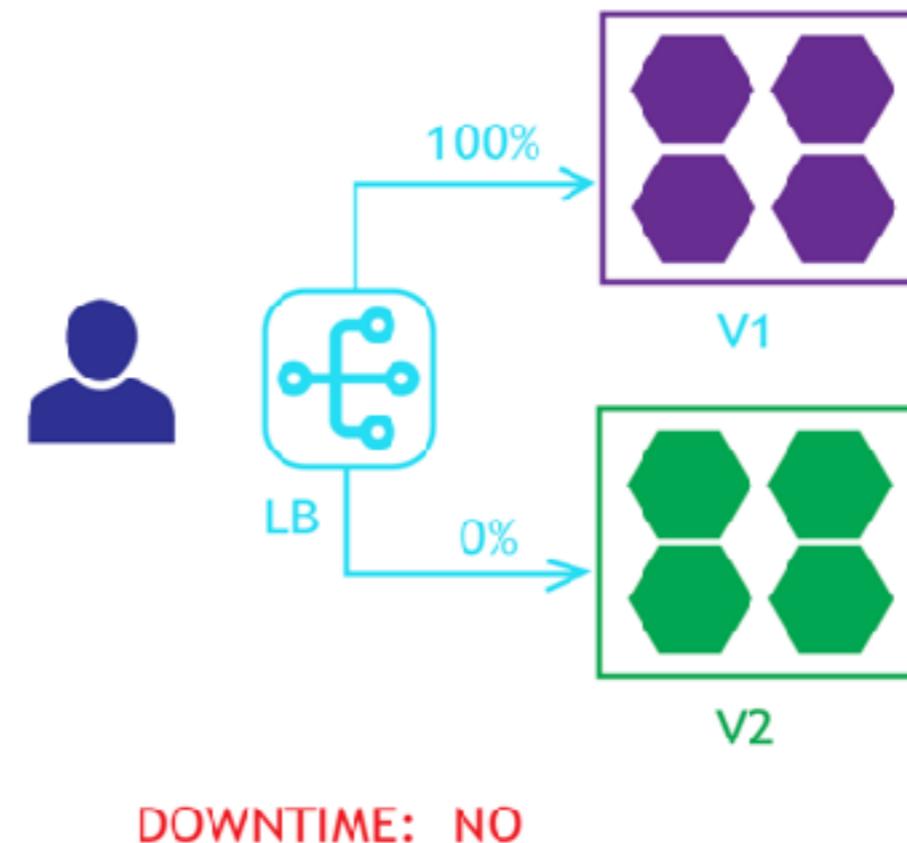
Recreate

Shutdown all existing pods before start new ones
Simple but causes downtime



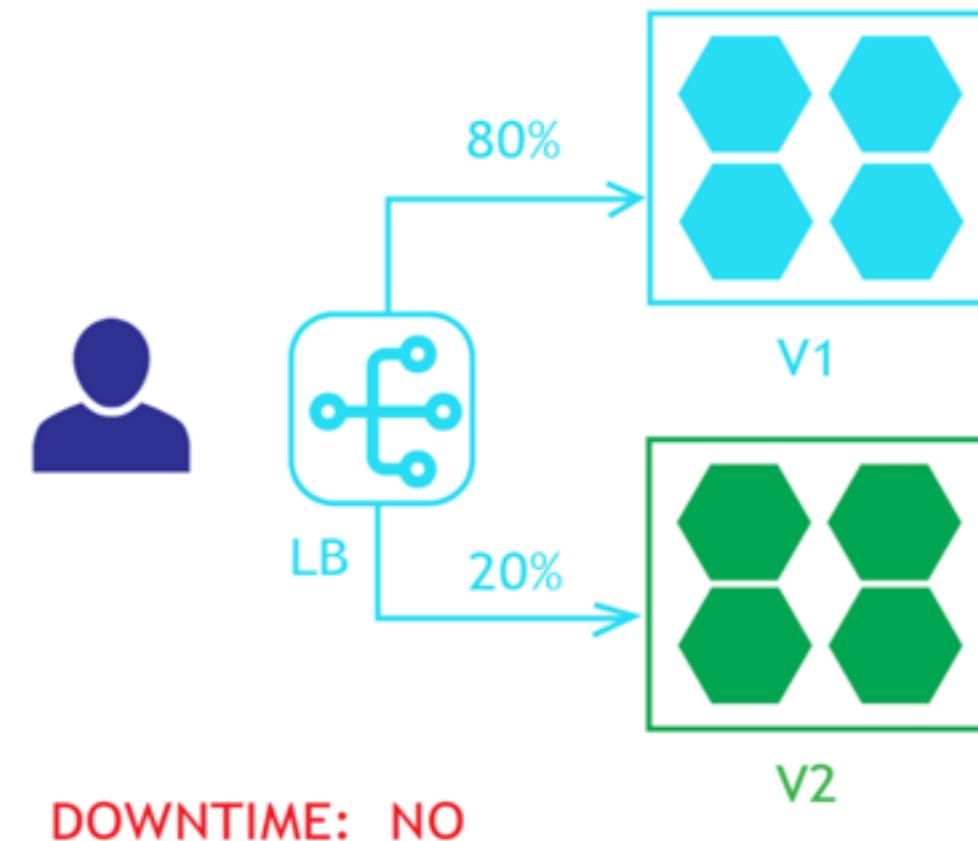
Blue/Green

Deploy a **new version** alongside the **current**
Switch traffic once stable



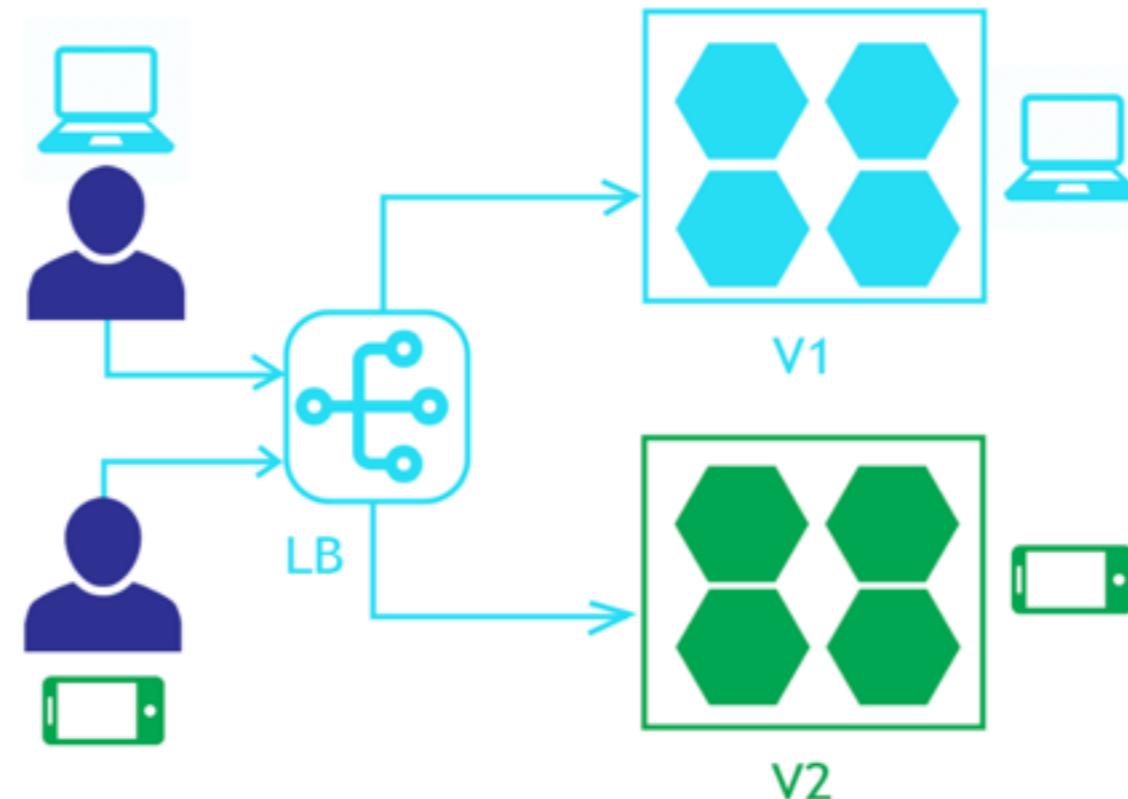
Canary

Send a small percentage of traffic to new version
Gradually increasing after validation



A/B Testing

Route traffic to different versions based on **rules**
User segmentation, geolocation

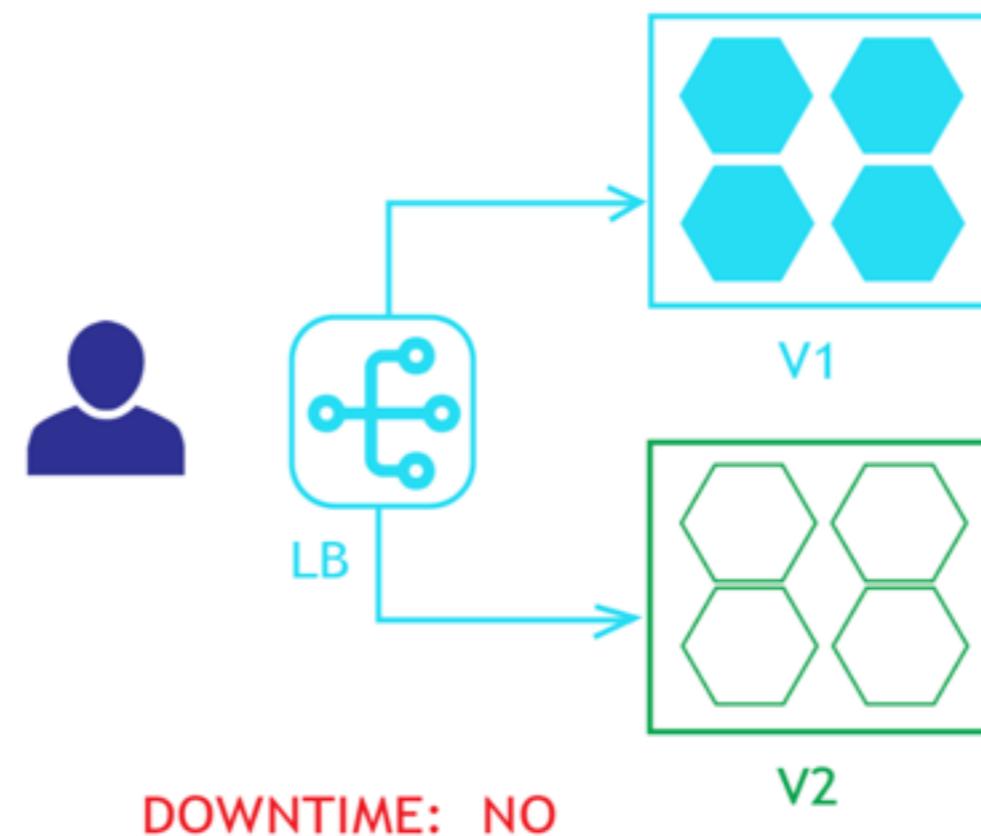


DOWNTIME: NOT APPLICABLE



Shadow deployment

Clone live traffic to a new version for testing
Without effecting users



Services discovery and Load balancing



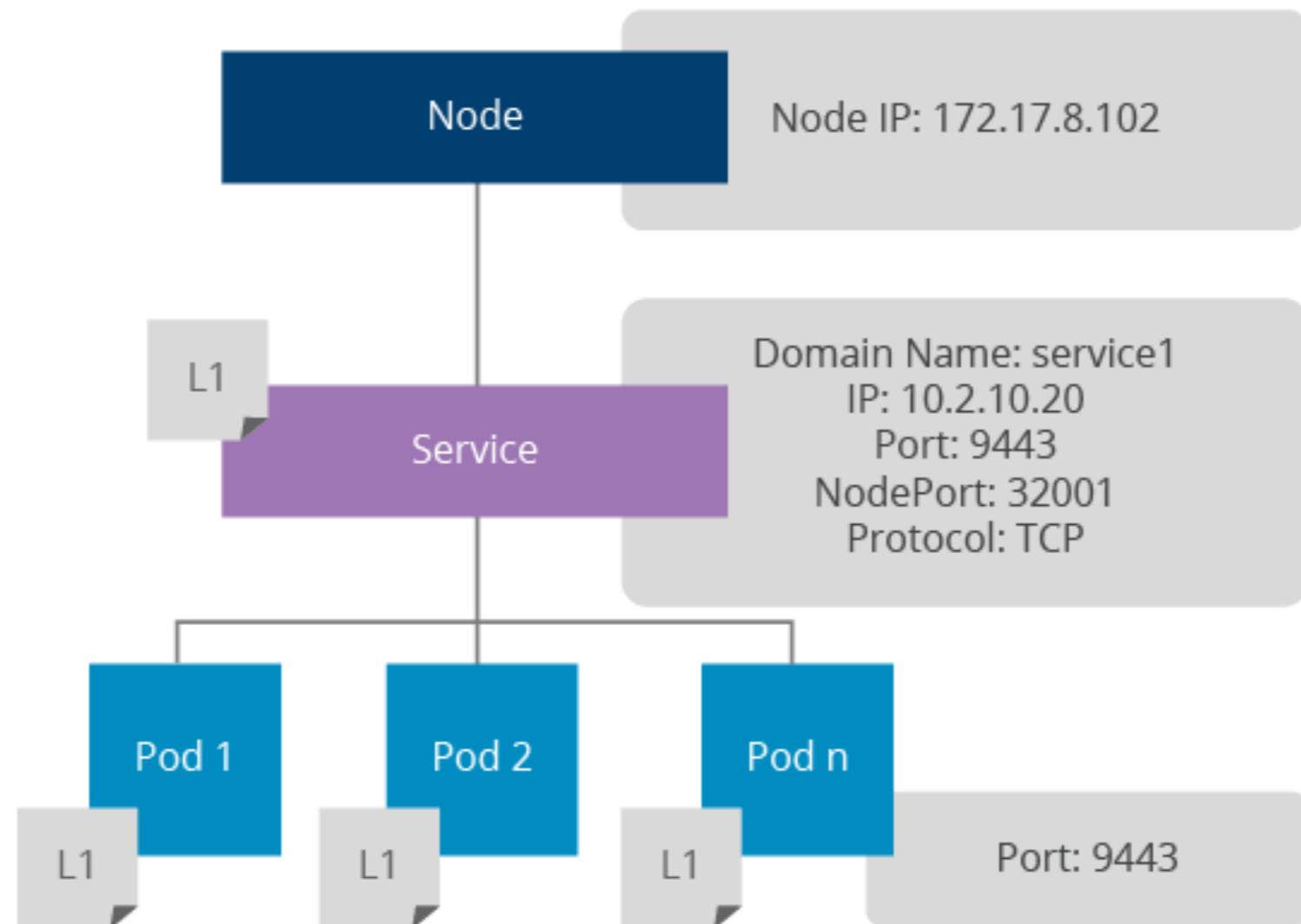
Services

- Independent from Pods
- Abstraction layer of Pods
 - Provide load balance
 - Expose access Pods/Load balance
 - Find Pods by label selector

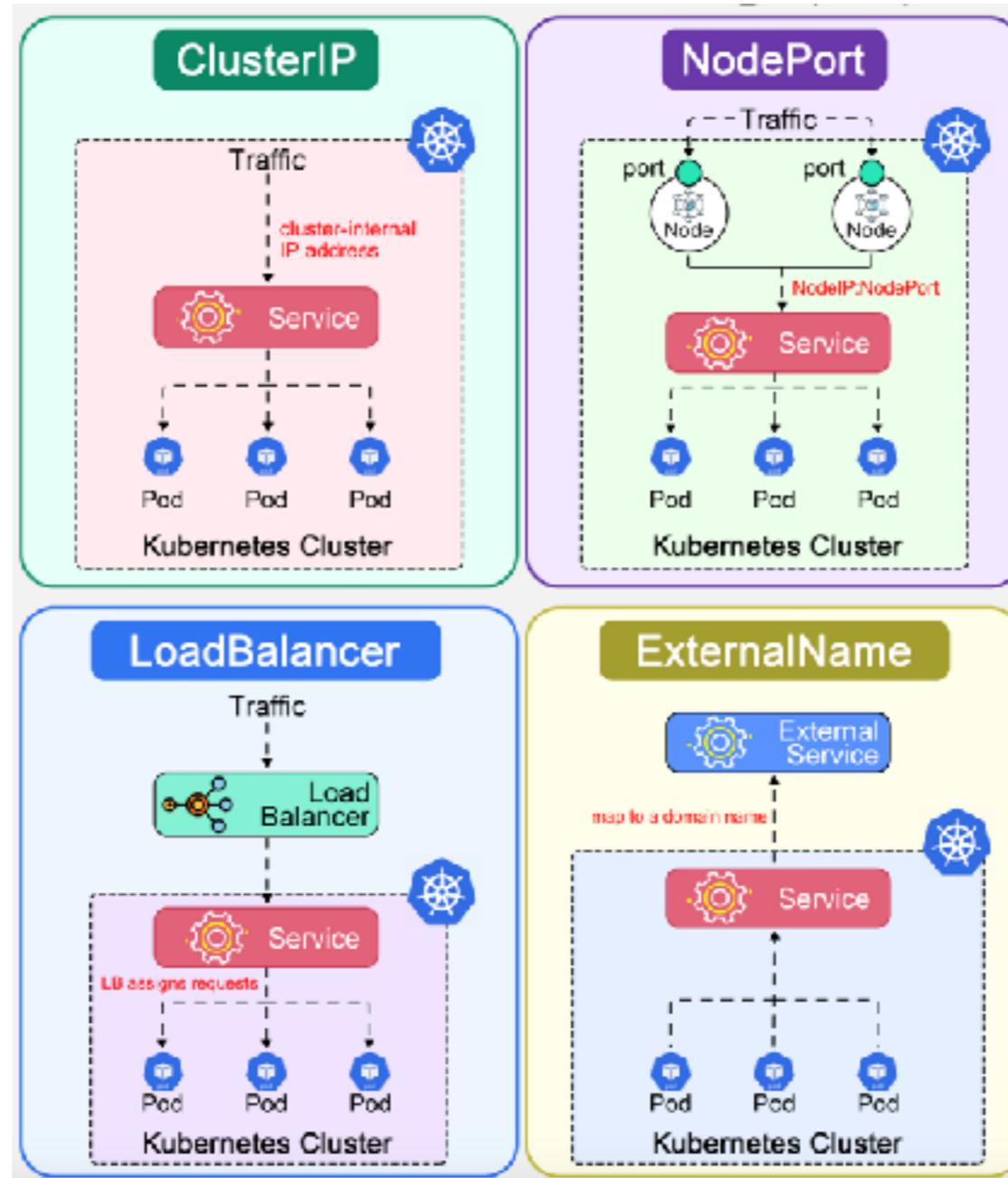
<https://kubernetes.io/docs/concepts/services-networking/service/>



Services



Types of services



<https://bytebytogo.com/guides/top-4-kubernetes-service-types-in-one-diagram/>

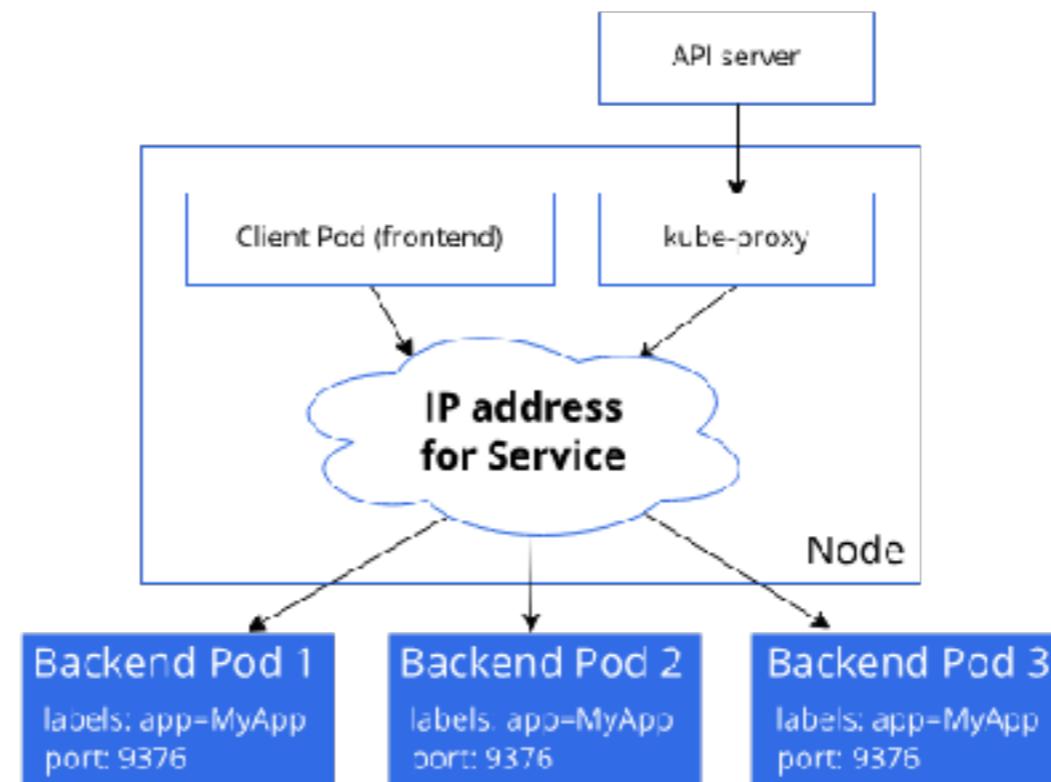


ClusterIP

Default of services

Access only from services in the same cluster

You can expose ClusterIP service with **kube-proxy**



<https://kubernetes.io/docs/concepts/services-networking/cluster-ip-allocation/>

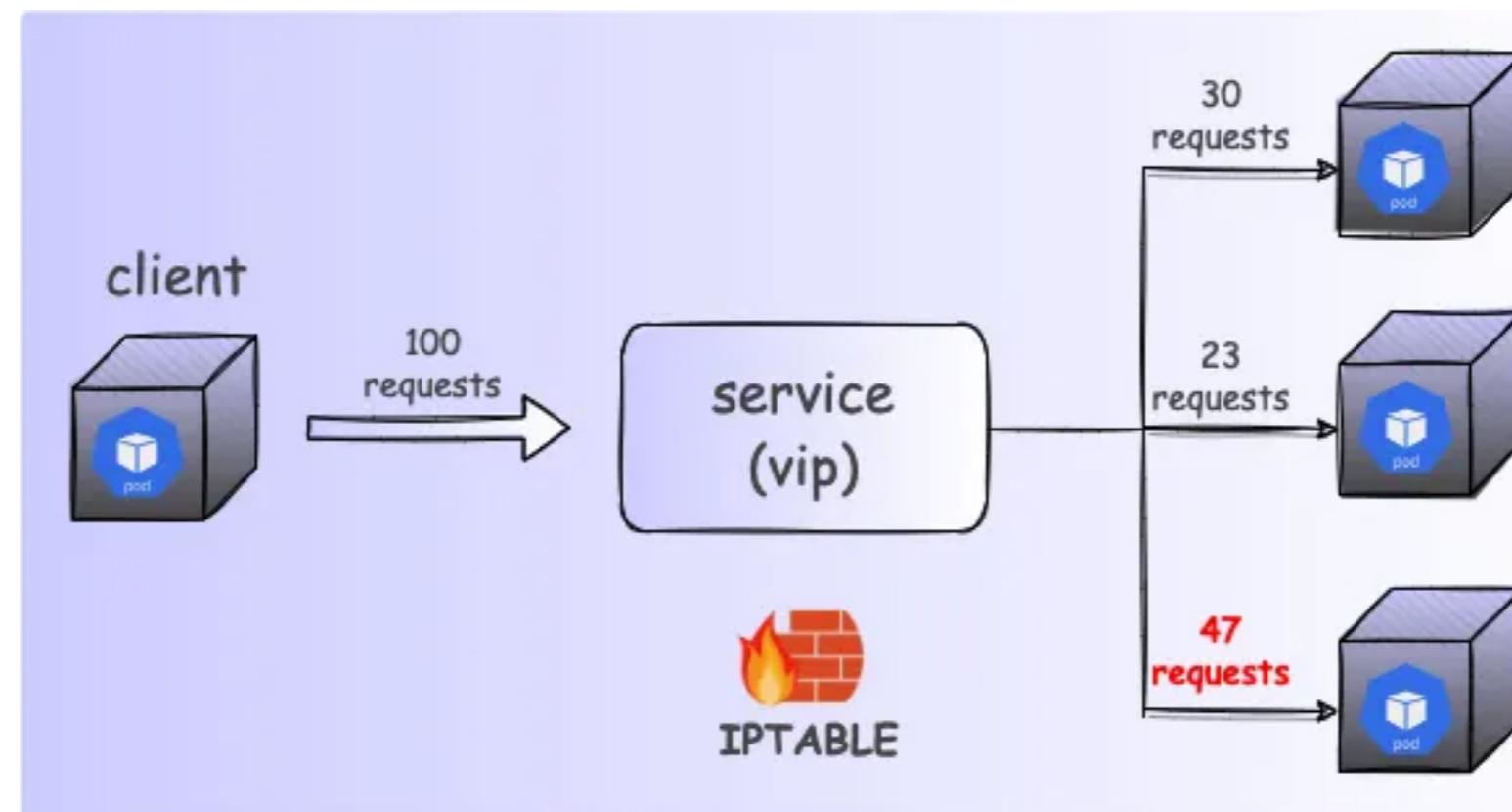


Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Kube-proxy

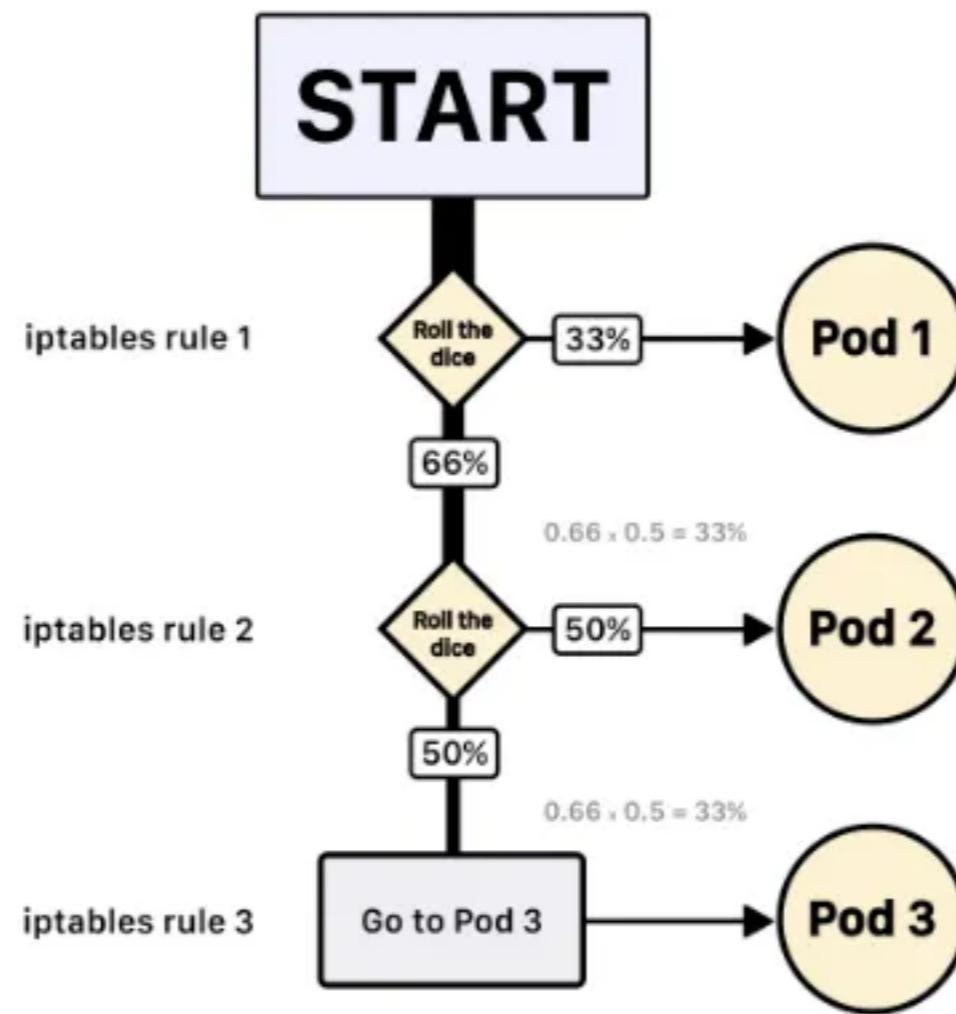
Default = **iptables**
IPVS, userspace



<https://kubernetes.io/docs/reference/networking/virtual-ips/#proxy-modes>



Kube-proxy



<https://learnkube.com/kubernetes-long-lived-connections>



NodePort

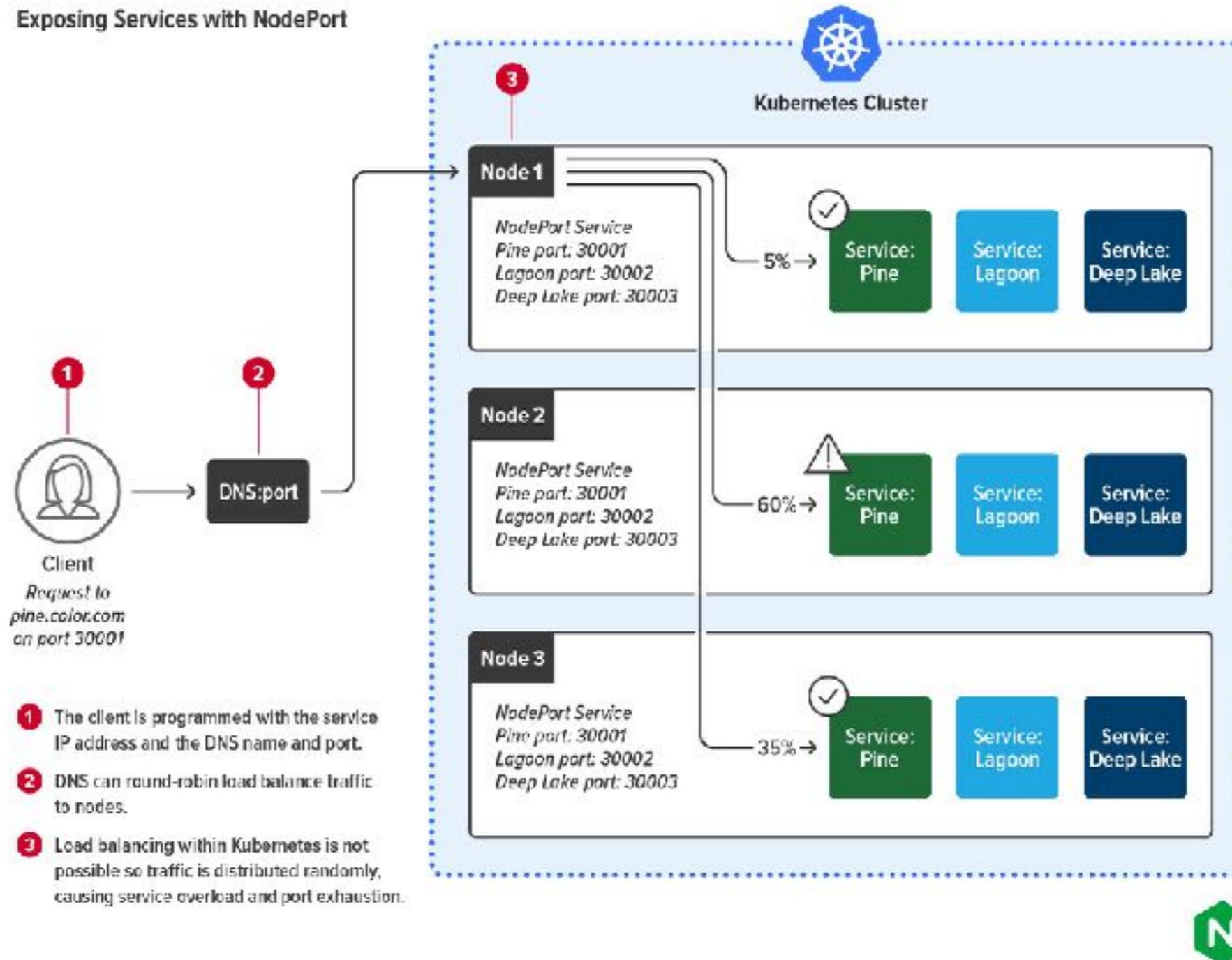
Expose the service outside the cluster with port
Range of port 30000-32767

Use layer 4 routing rules and Linux **iptables** utilities

<https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport>



Expose services with NodePort



<https://www.f5.com/company/blog/nginx/kubernetes-networking-101>



Load Balance (LB)

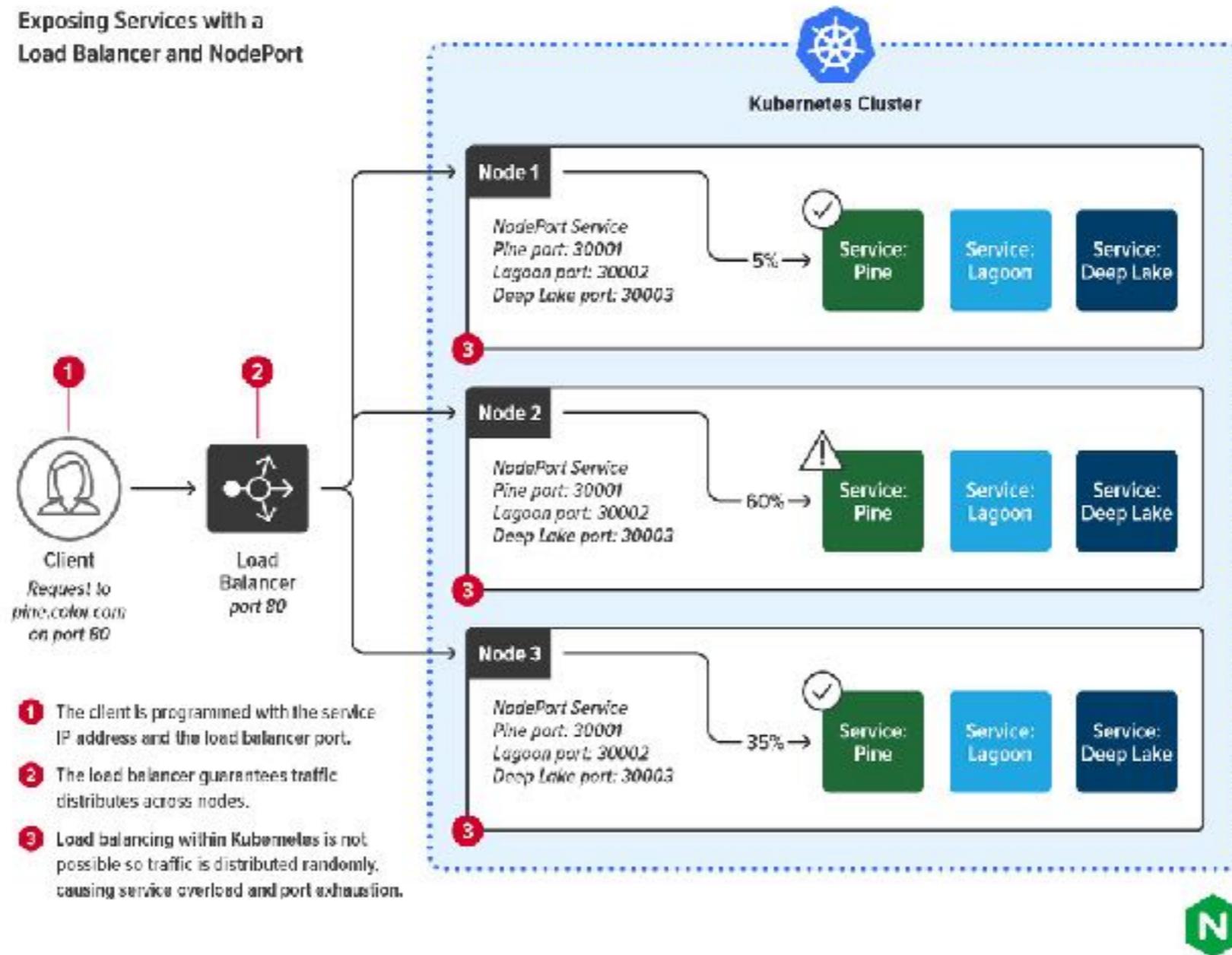
Expose the Service externally using a cloud provider's load balancer

Good choice for small and static deployment

<https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport>



Expose services with LB



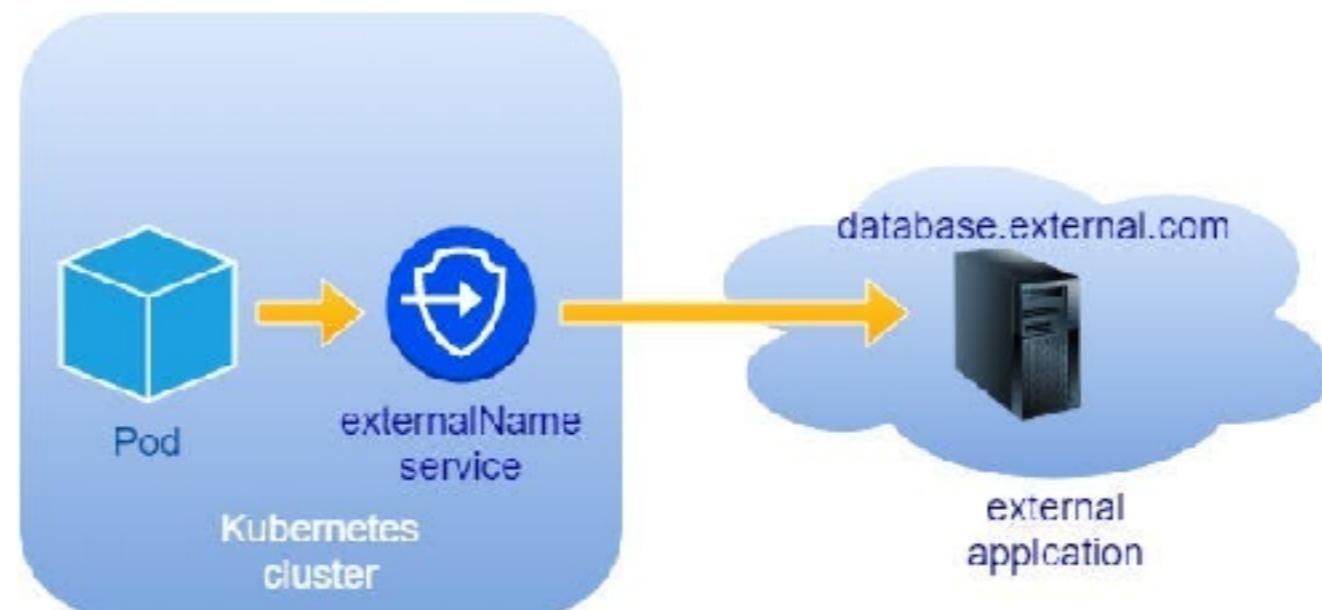
<https://www.f5.com/company/blog/nginx/kubernetes-networking-101>



ExternalName

Mapping a service to a domain name

This is commonly used to create a service within Kubernetes to represent an **external database**



<https://kubernetes.io/docs/concepts/services-networking/service/#type-nodeport>



Ingress !!

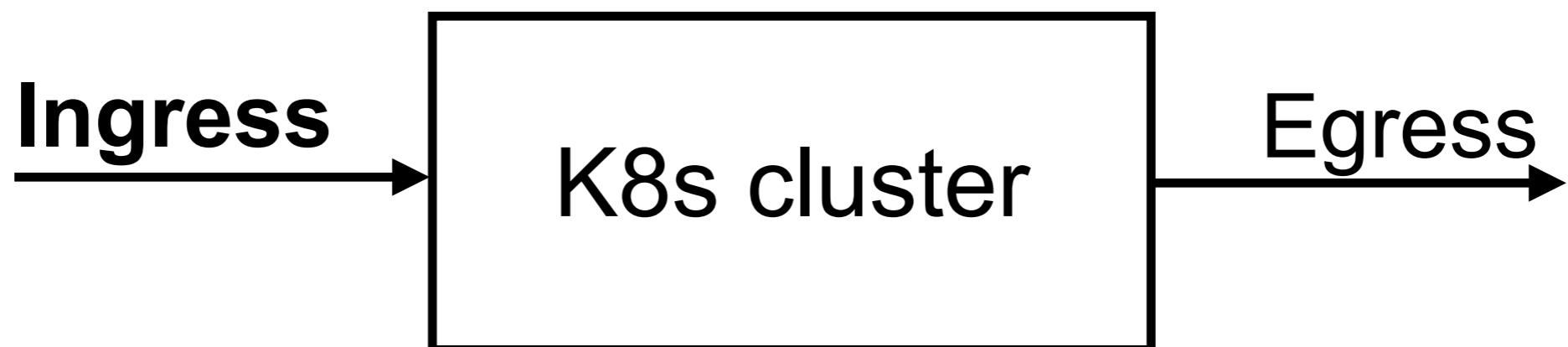
 **FEATURE STATE:** Kubernetes v1.19 [stable]

<https://kubernetes.io/docs/concepts/services-networking/ingress/>



Ingress ?

The traffic that enter to the cluster
A kind of resource in cluster

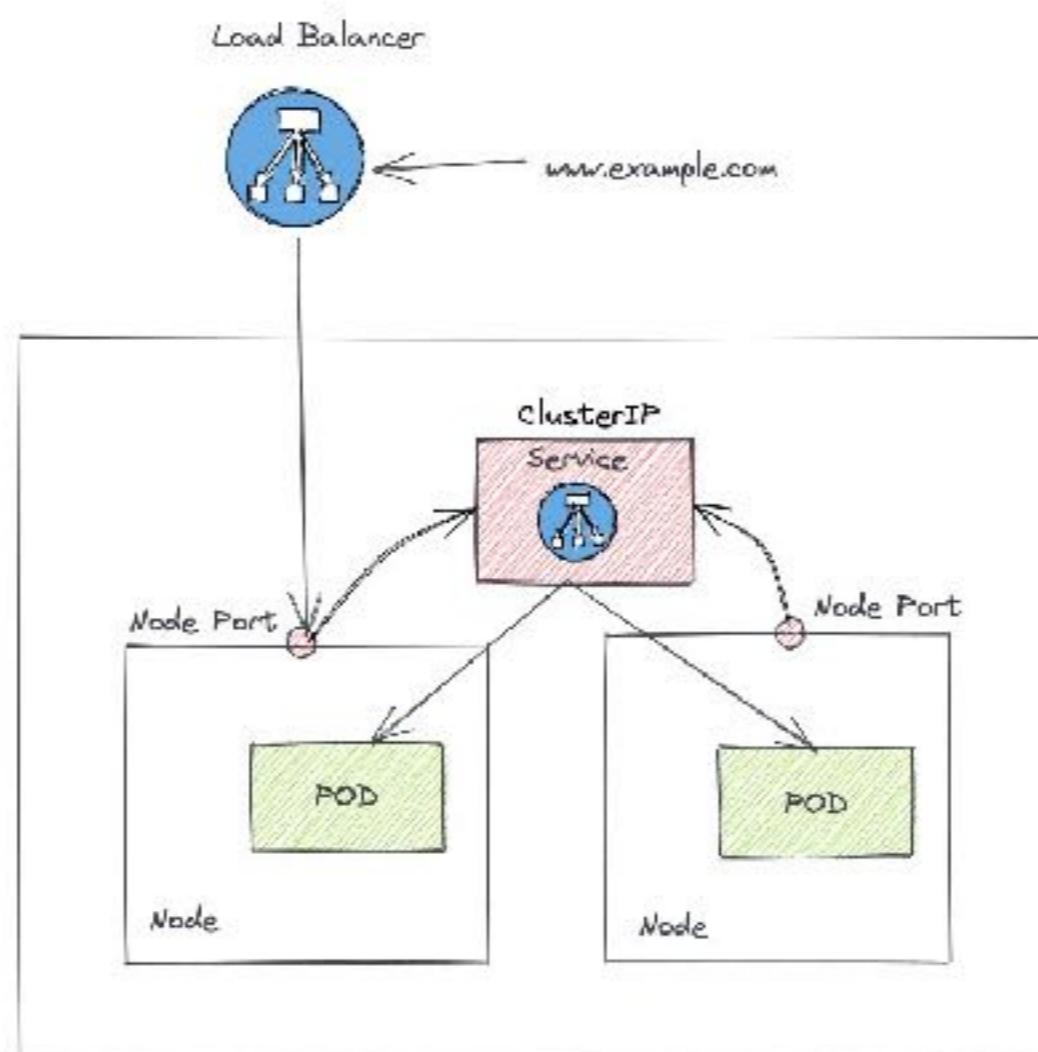


<https://kubernetes.io/docs/concepts/services-networking/ingress/>



Before Ingress ?

Use service with NodePort type



<https://devopscube.com/kubernetes-ingress-tutorial/>

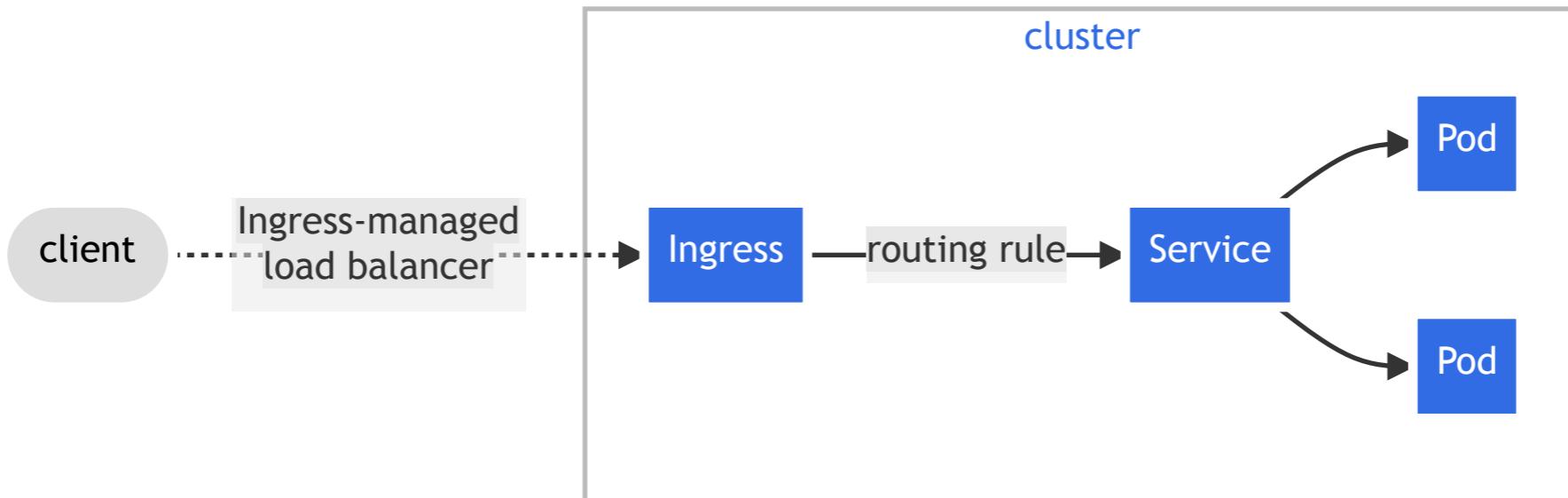


Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Ingress ?

How to handle multiple services in same port ?
How to limit protocol to access ?



<https://kubernetes.io/docs/concepts/services-networking/ingress/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

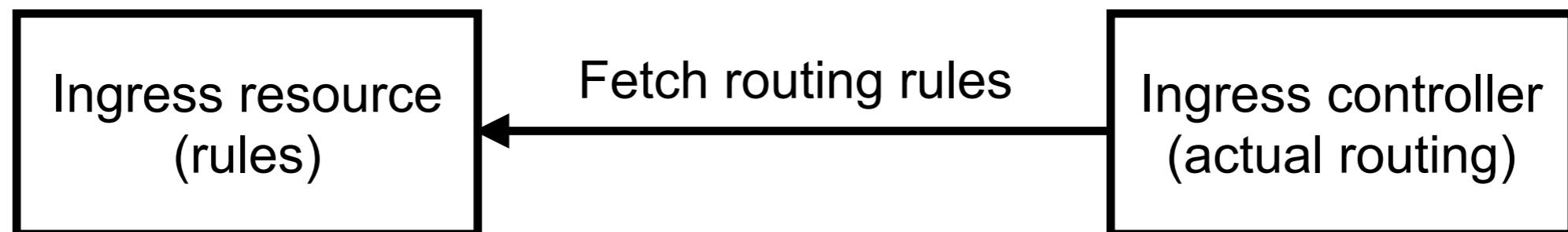
Ingress components ?

Ingress Resource

Store DNS routing rules in the cluster

Ingress Controller

Routing by accessing the DNS rules applied through resource



<https://kubernetes.io/docs/concepts/services-networking/ingress/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Ingress Controller ?

Must be an ingress controller at least one controller

Nginx

AWS

CGE

APISIX

Alibaba

Kong

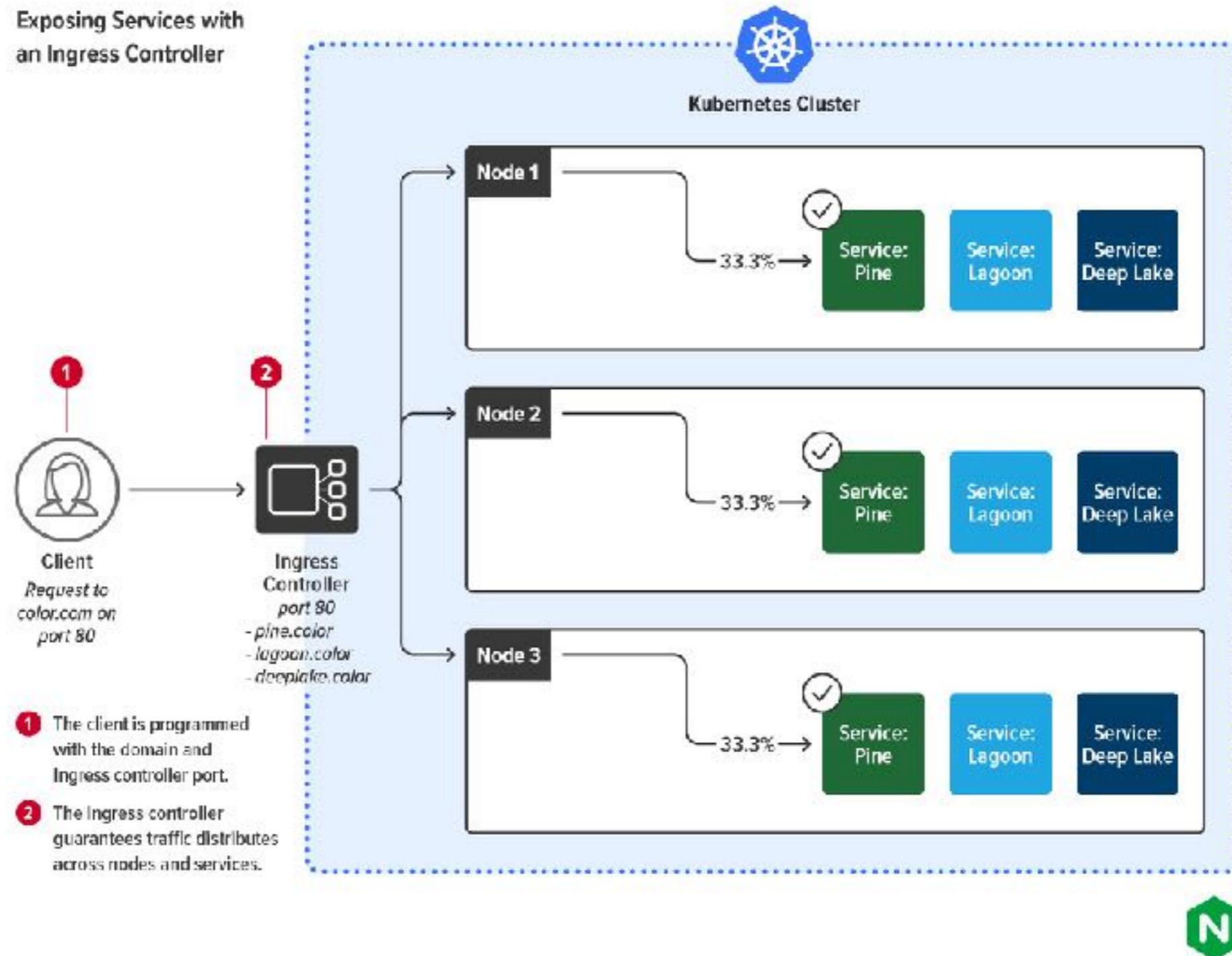
<https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

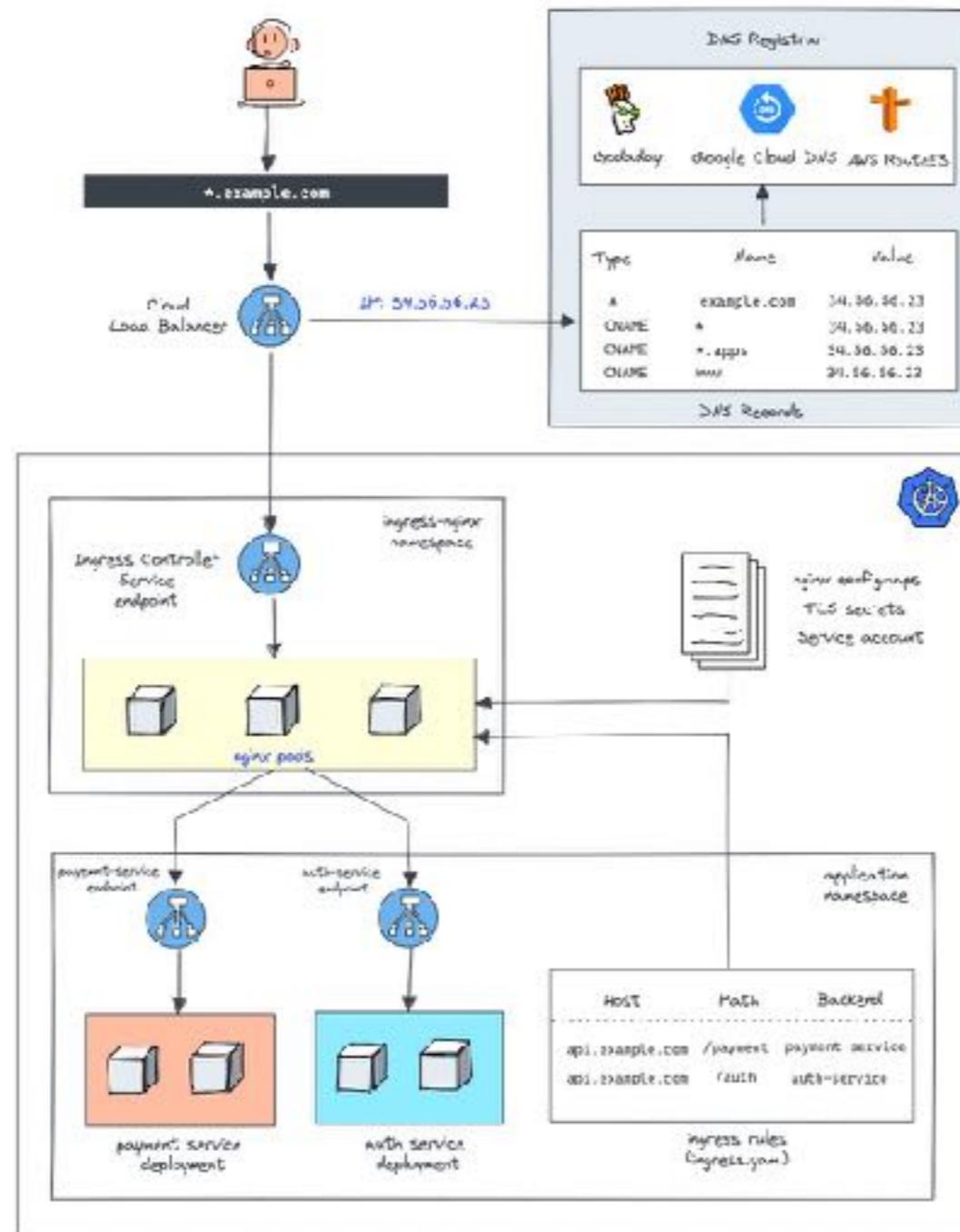
Expose with Ingress Controller



<https://www.f5.com/company/blog/nginx/kubernetes-networking-101>



Ingress and Ingress Controller



<https://devopscube.com/kubernetes-ingress-tutorial/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

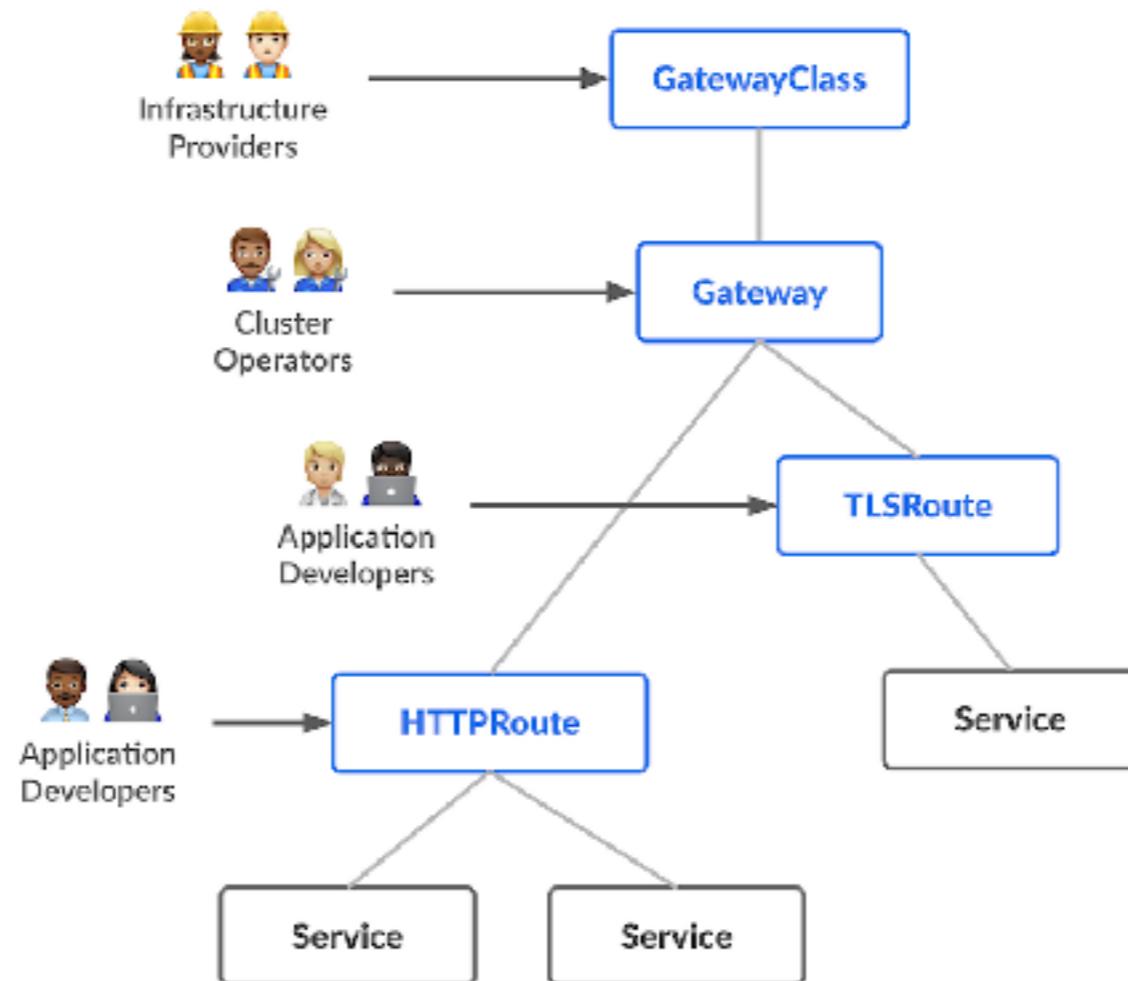
Gateway API

<https://kubernetes.io/docs/concepts/services-networking/gateway/>



Gateway API

Dynamic infrastructure provisioning
Advance traffic routing



<https://kubernetes.io/docs/concepts/services-networking/gateway/>

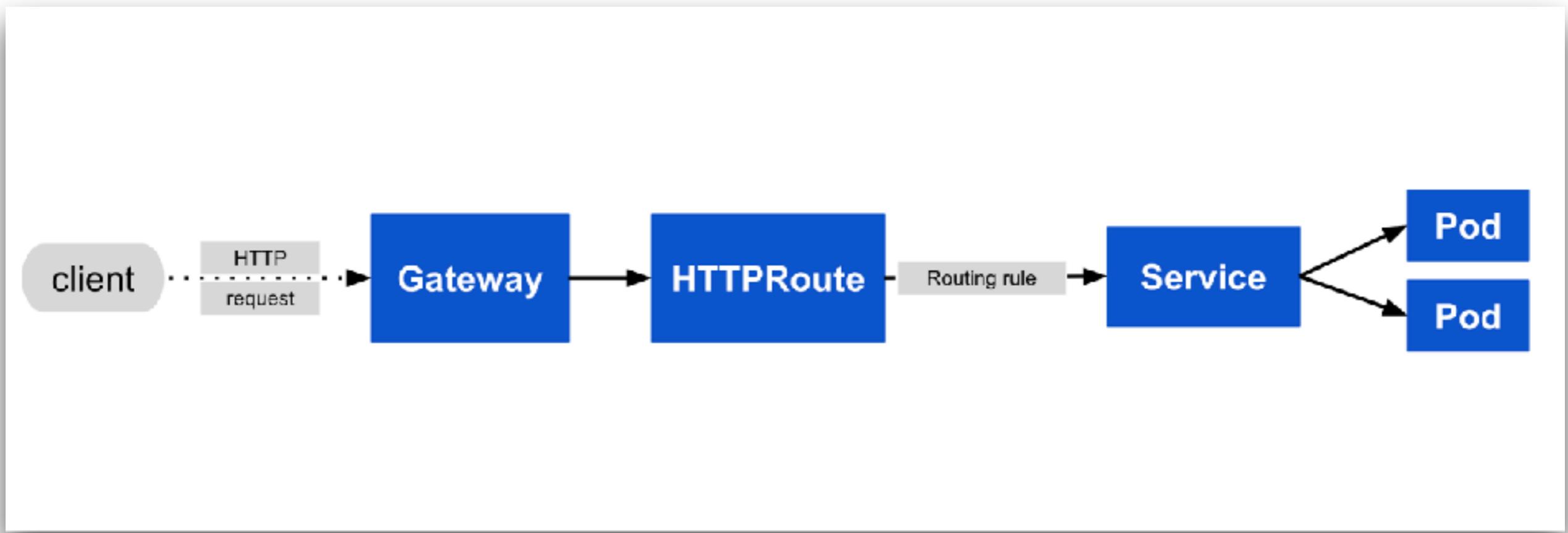


Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Gateway API

Request flow of HTTP traffic



<https://gateway-api.sigs.k8s.io/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Storages



Storage in Kubernetes

Long-term and temporary storage to Pods in cluster
By default, each container has **ephemeral storage**

Volume

PV and PVC

StorageClass

<https://kubernetes.io/docs/concepts/storage/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Storage in Kubernetes

Volume Type	Description	Pod-Level	Persistent
emptyDir	Temporary directory on node	Yes	No
hostPath	Directory on node	Yes	Yes
awsElasticBlockStore / gcePersistentDisk / azureDisk / azureFile	Cloud provider-specific storage	No	Yes
nfs	Network File System share	No	Yes
persistentVolumeClaim	Use a Persistent Volume	No	Yes
configMap / secret / downwardAPI	Expose pod info and cluster info	Yes	No
csi	Use third-party storage providers	No	Depends on the specific CSI driver

<https://kubernetes.io/docs/concepts/storage/>



Kubernetes

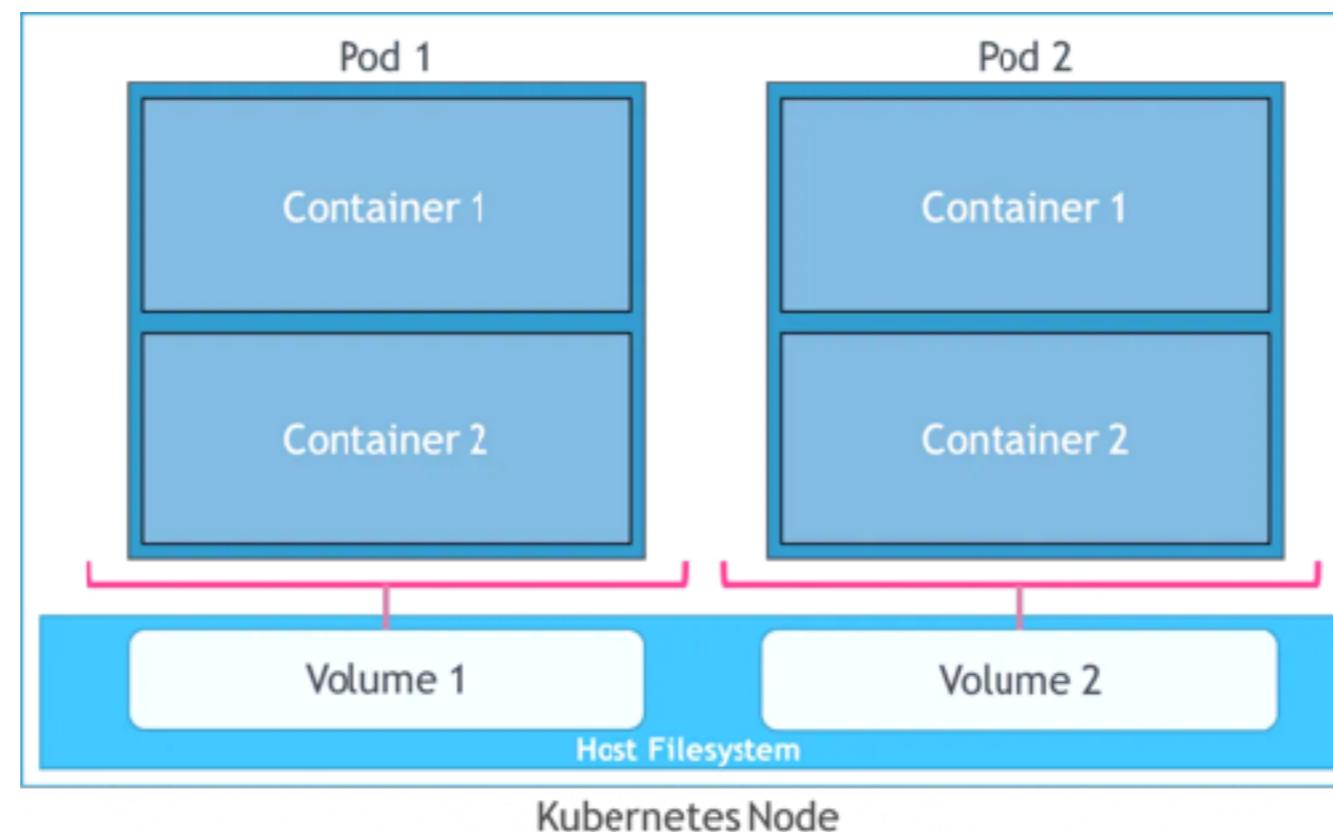
© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Volume

Storage unit associated with Pod

Shared data between containers in the same Pod
eg. EmptyDir, HostPath

Support all types of storages (local, network storage)



PV and PVC

Persistent volume

Storage unit in the cluster, provisioning by admin

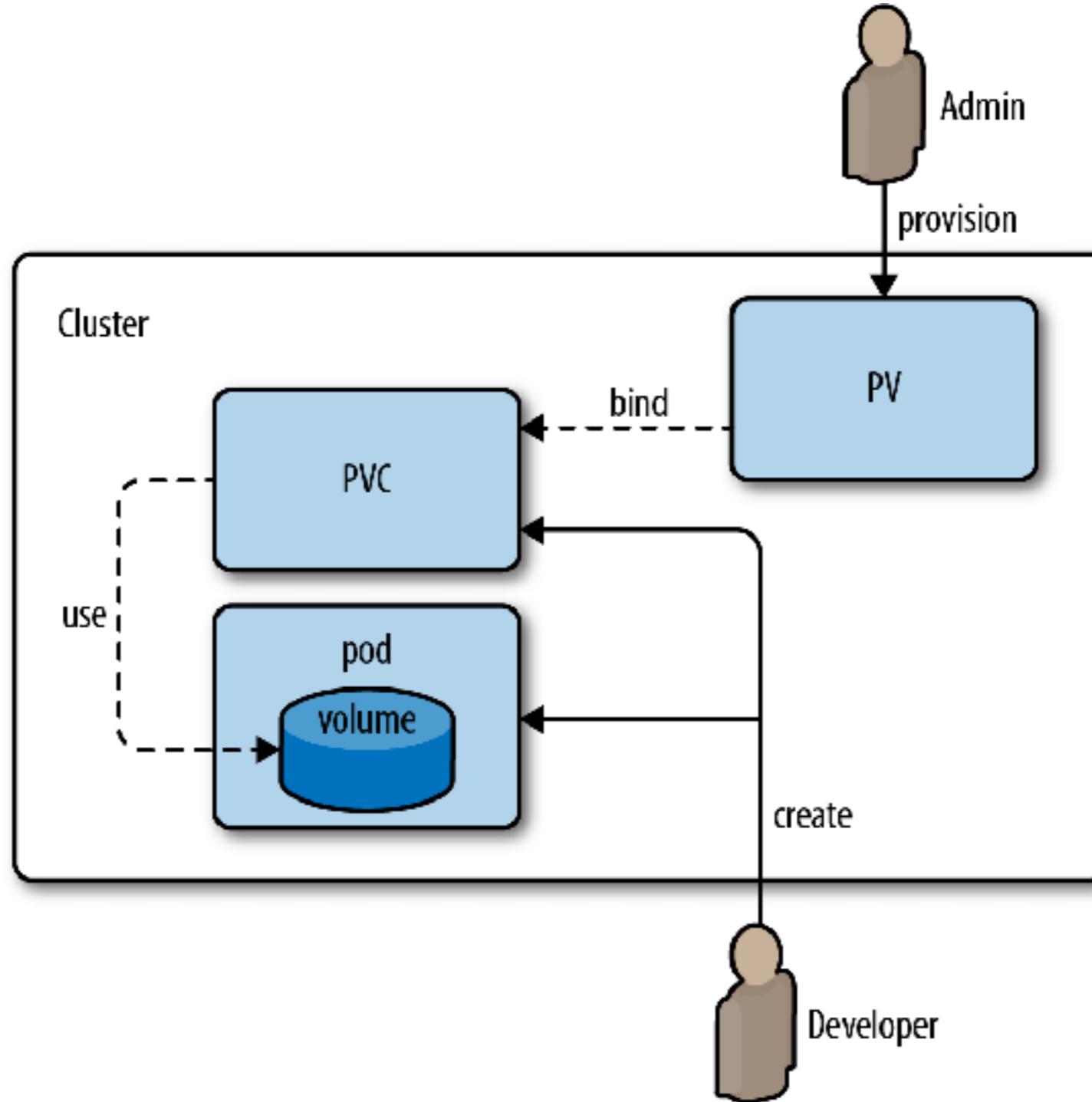
Persistent volume

User's request for storage
PVC utilize PV resources

Provide more robust solution for long-term data storage



PV and PVC



StorageClass in Kubernetes

Dynamic volume provisioning

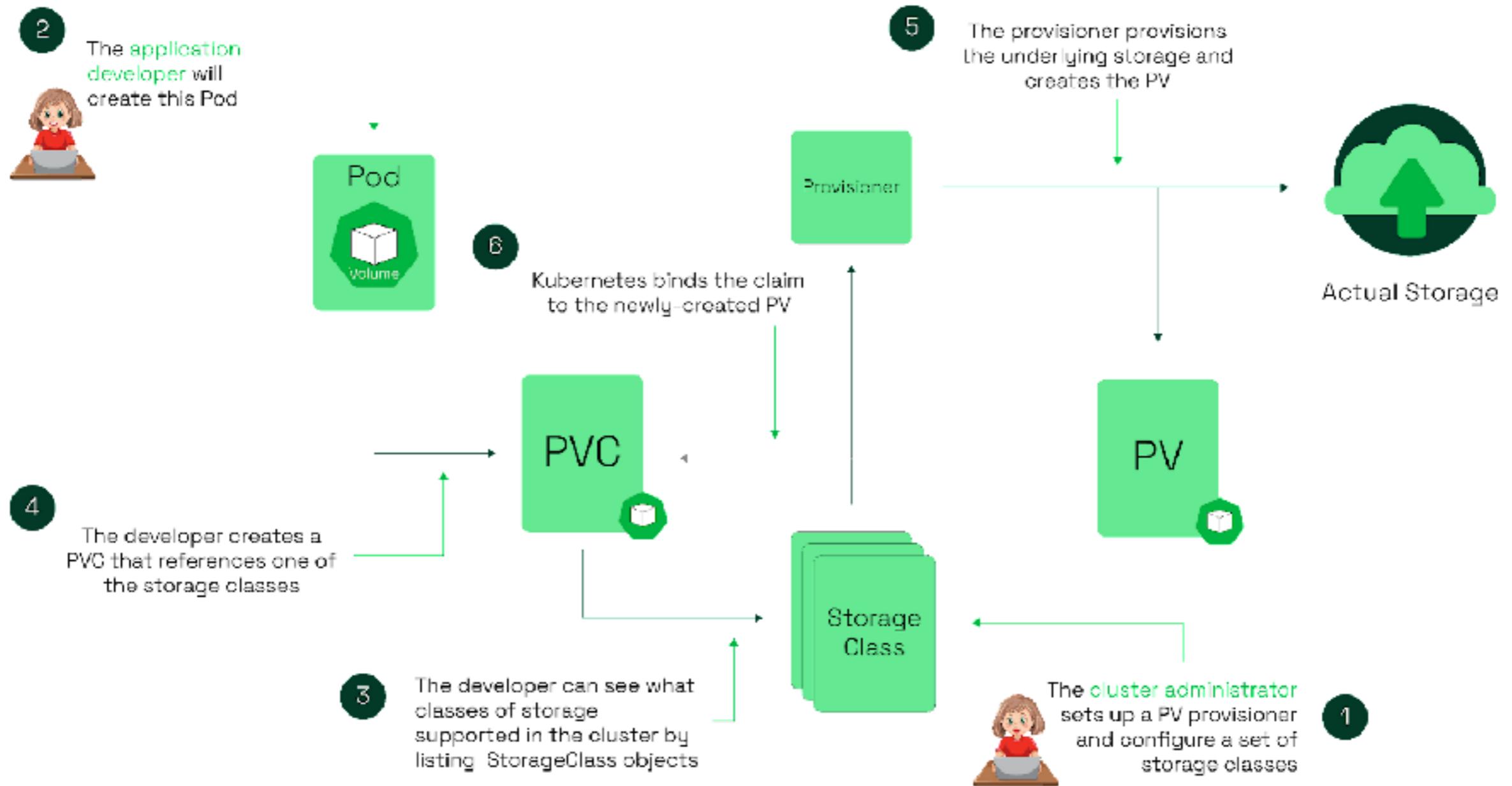
Define **classes or **tiers** of storage**

Volumes are provisioned with classes by user's need

<https://kubernetes.io/docs/concepts/storage/storage-classes/>



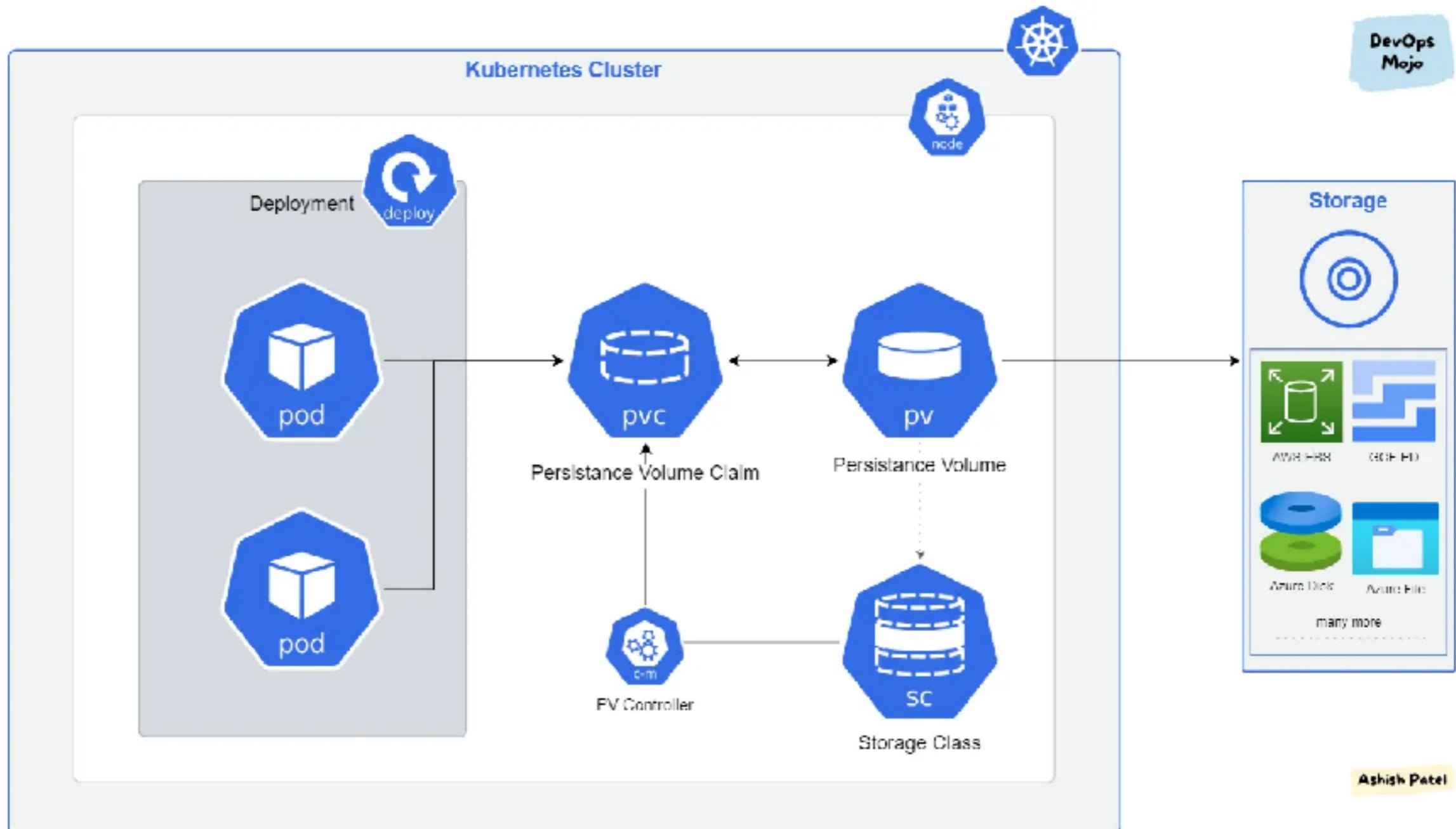
StorageClass in Kubernetes



<https://www.apptio.com/topics/kubernetes/best-practices/storage-class/>



StorageClass in Kubernetes



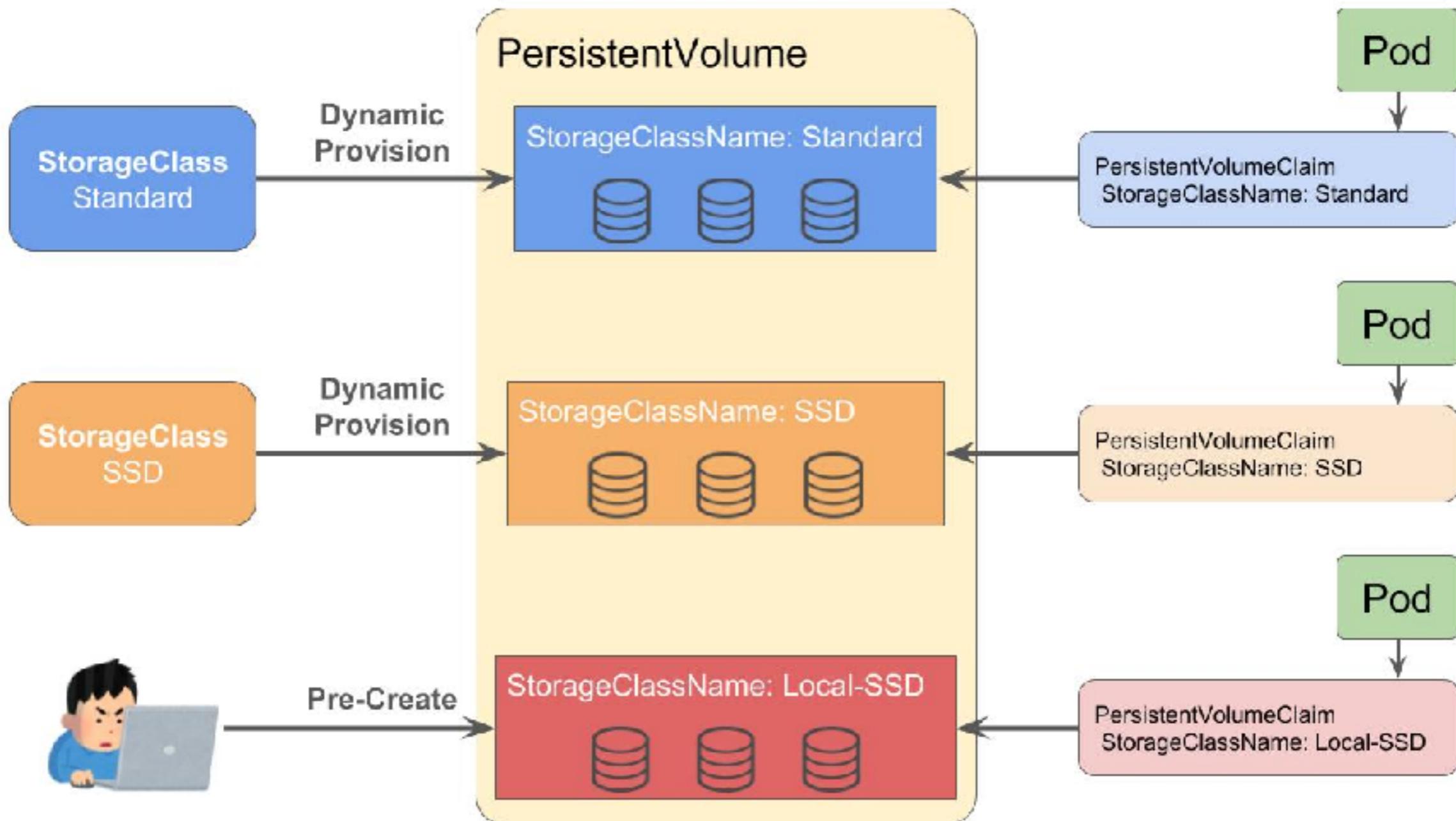
<https://medium.com/devops-mojo>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

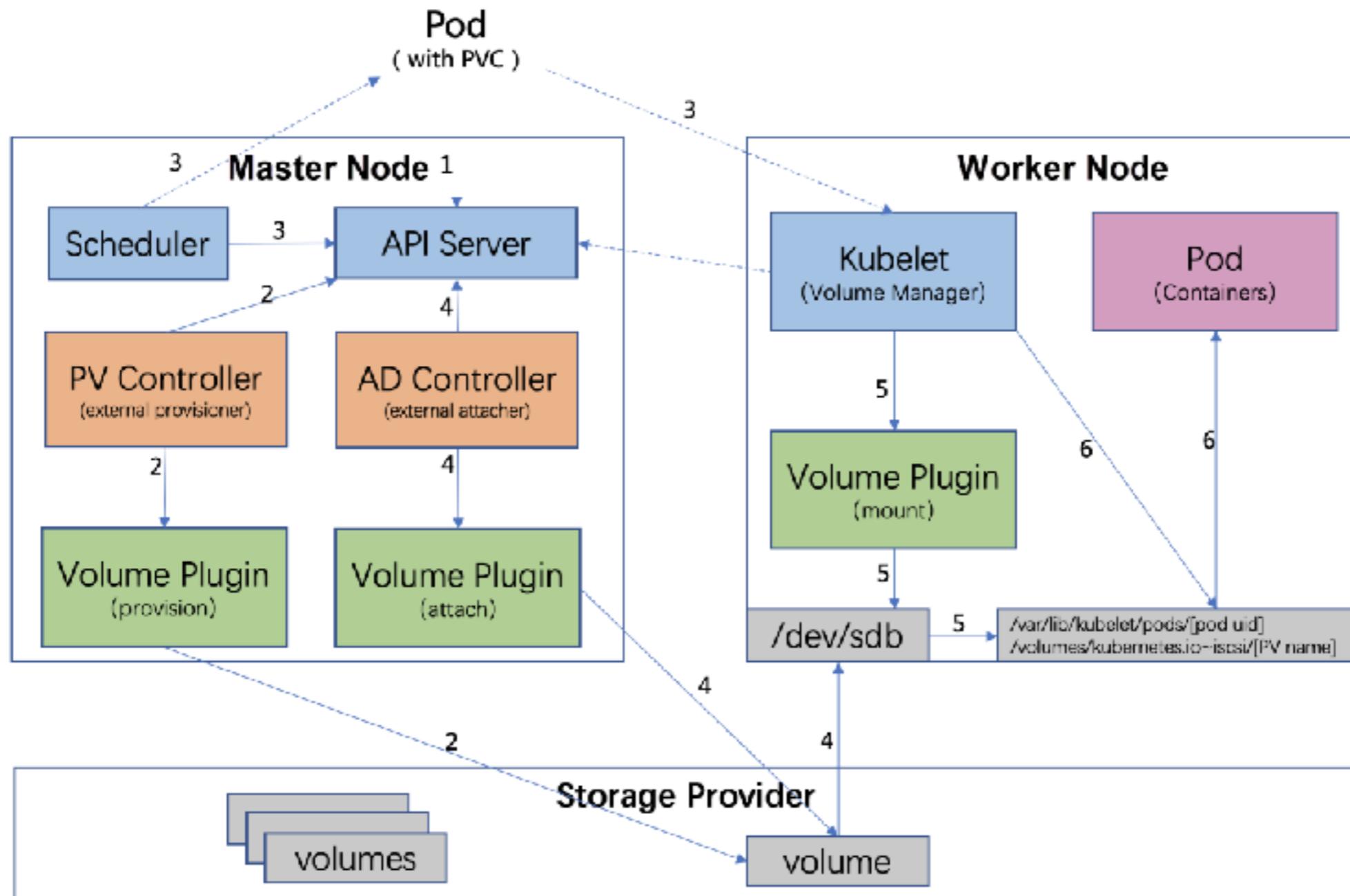
StorageClass in Kubernetes



<https://medium.com/devops-mojo>



Processes of Persistent Storage



https://www.alibabacloud.com/blog/kubernetes-persistent-storage-process_596505



Components ?

PV Controller

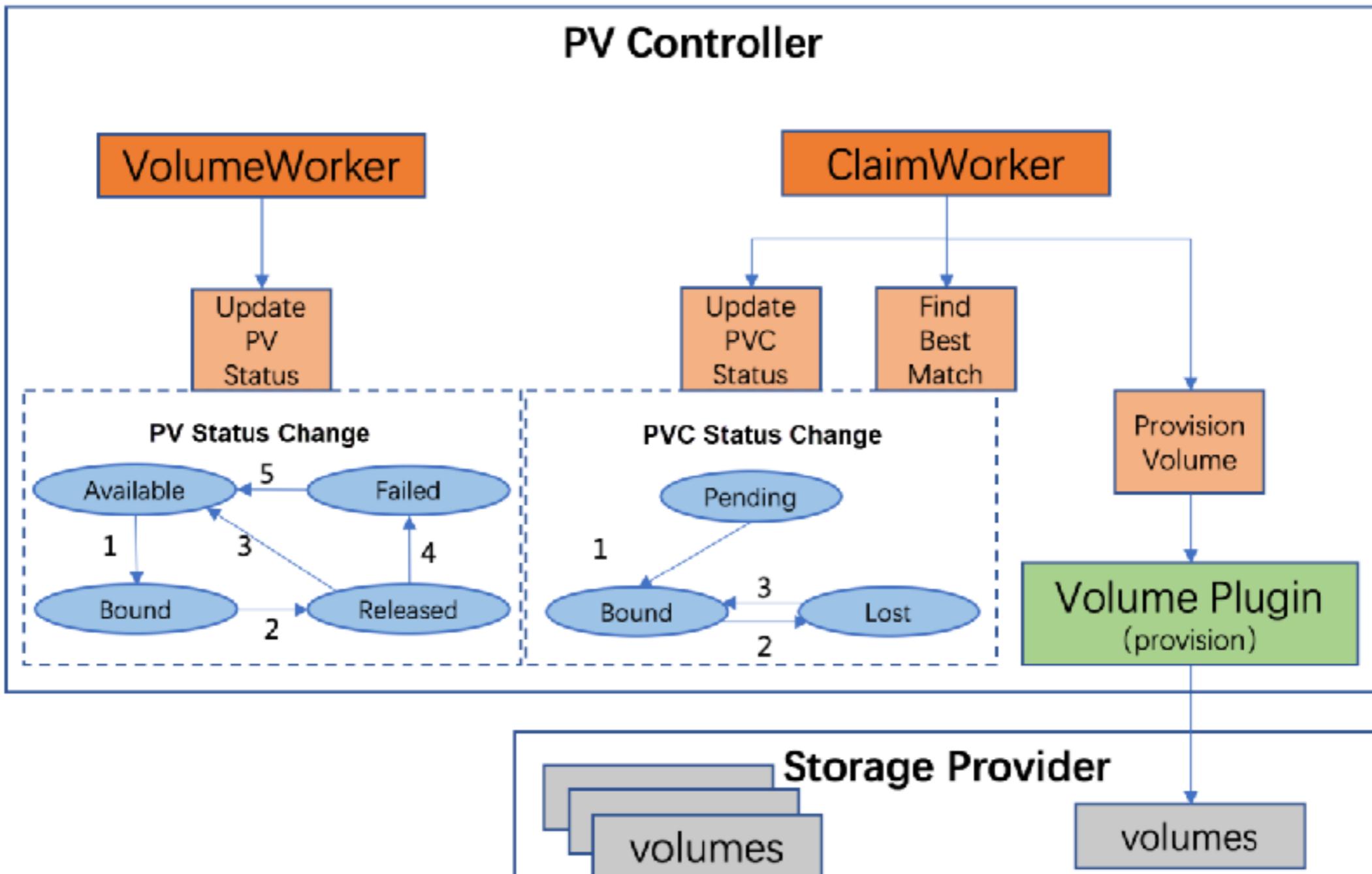
Binding PVs and PVCs and manage life cycle
Provision and delete operations on data volumes

AD Controller

Attach and detach operations on data volumes
Attach devices to target nodes



PV Controller

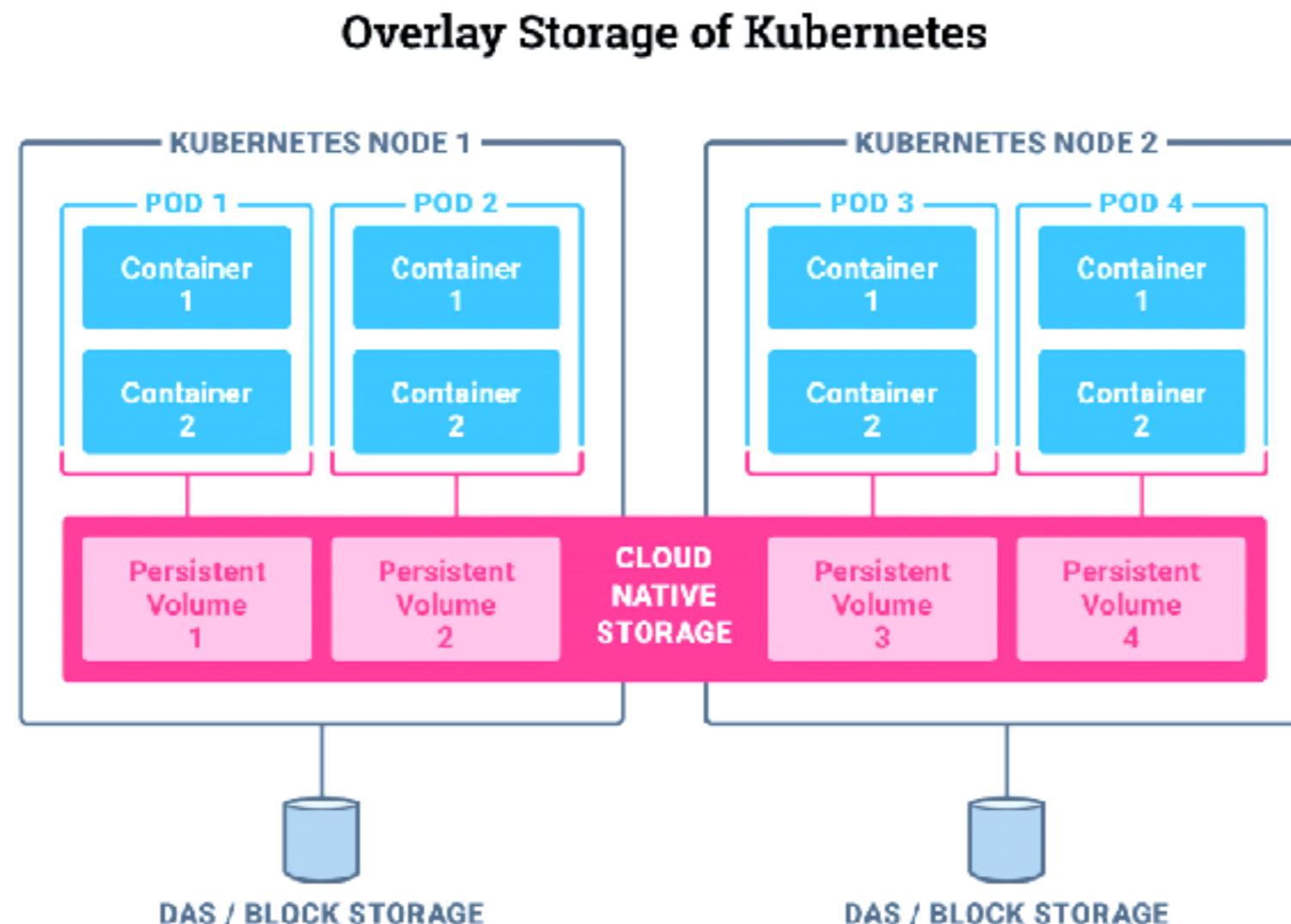


https://www.alibabacloud.com/blog/kubernetes-persistent-storage-process_596505



Overlay storage in Kubernetes

Working with Container Storage Interface (CSI)



Source: Janakiram MSV

© 2020 THE NEW STACK

<https://thenewstack.io/how-kubernetes-provides-networking-and-storage-to-applications/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Opensource Storage solutions ?



OpenEBS

Rook



GlusterFS



Longhorn



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

StatefulSet



StatefulSet

Bringing the concept of ReplicaSets to **stateful** Pods

Enable running Pods in **cluster** mode

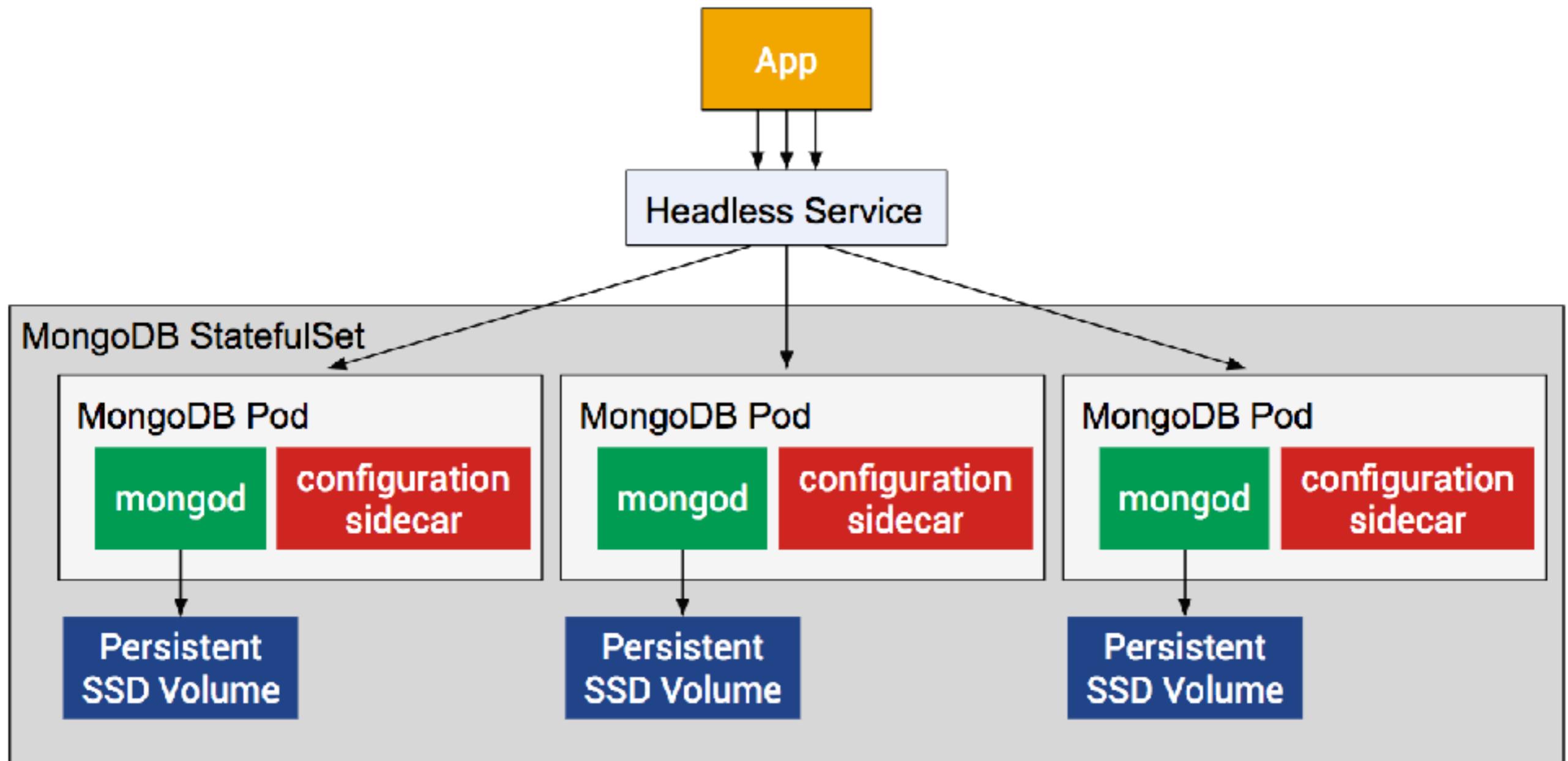
Ideal for deploy **highly** available database **workload**

Pods are created **sequentially**

Pods are terminated in **LiFo** (Last in, First out)



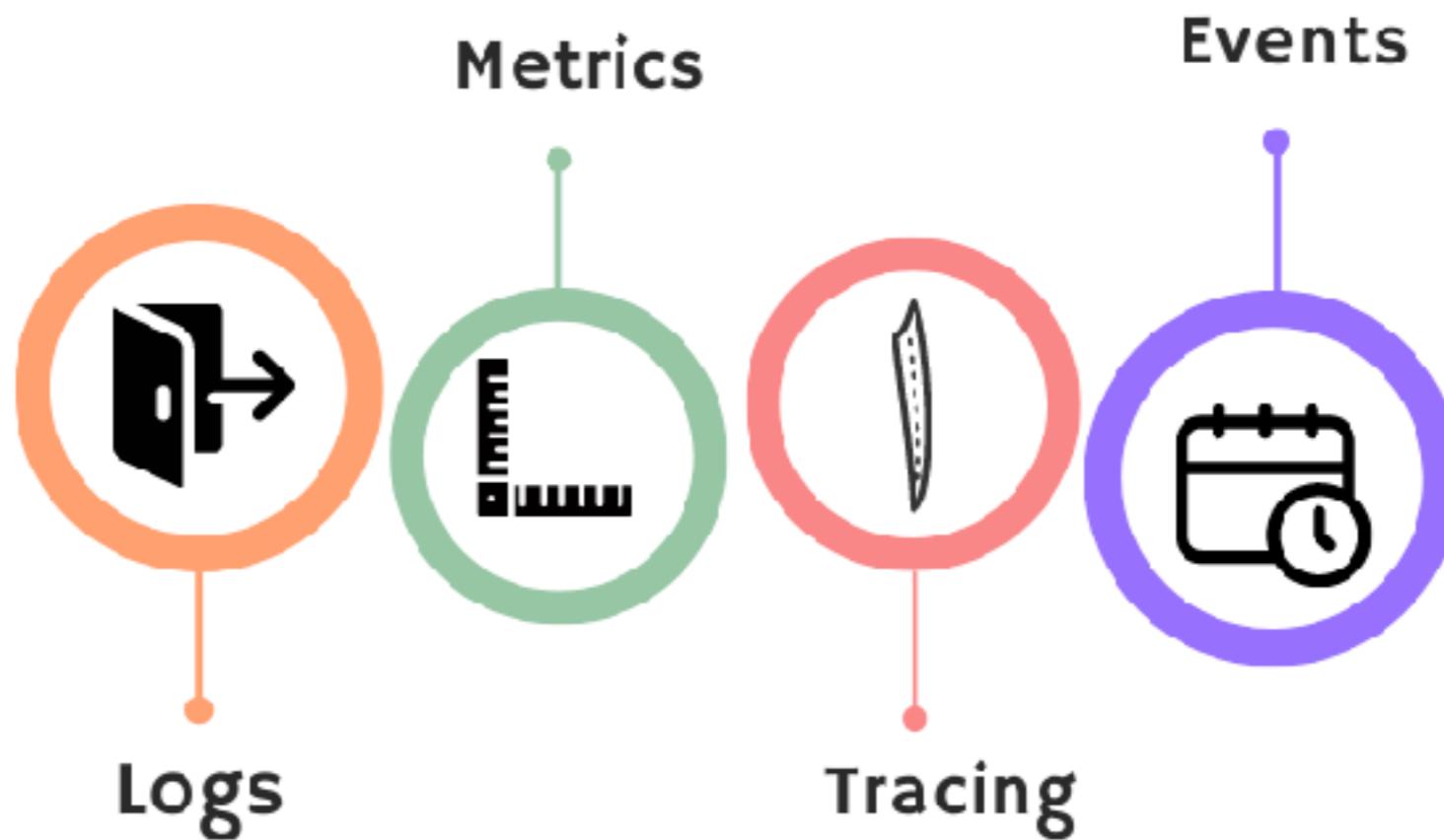
StatefulSet



Observability Monitoring



THE FOUR PILLARS OF API OBSERVABILITY



{APIToolkit.io}

<https://apitoolkit.io/blog/the-four-pillars-of-api-observability/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Reasons to monitoring ?

- Realtime alerts and early error detection
- Better workload management and optimization
- Easily troubleshooting
- Realtime cost visibility
- Powerful insight

<https://grafana.com/solutions/kubernetes/kubernetes-monitoring-introduction/>



Kubernetes monitoring ?

Node
view

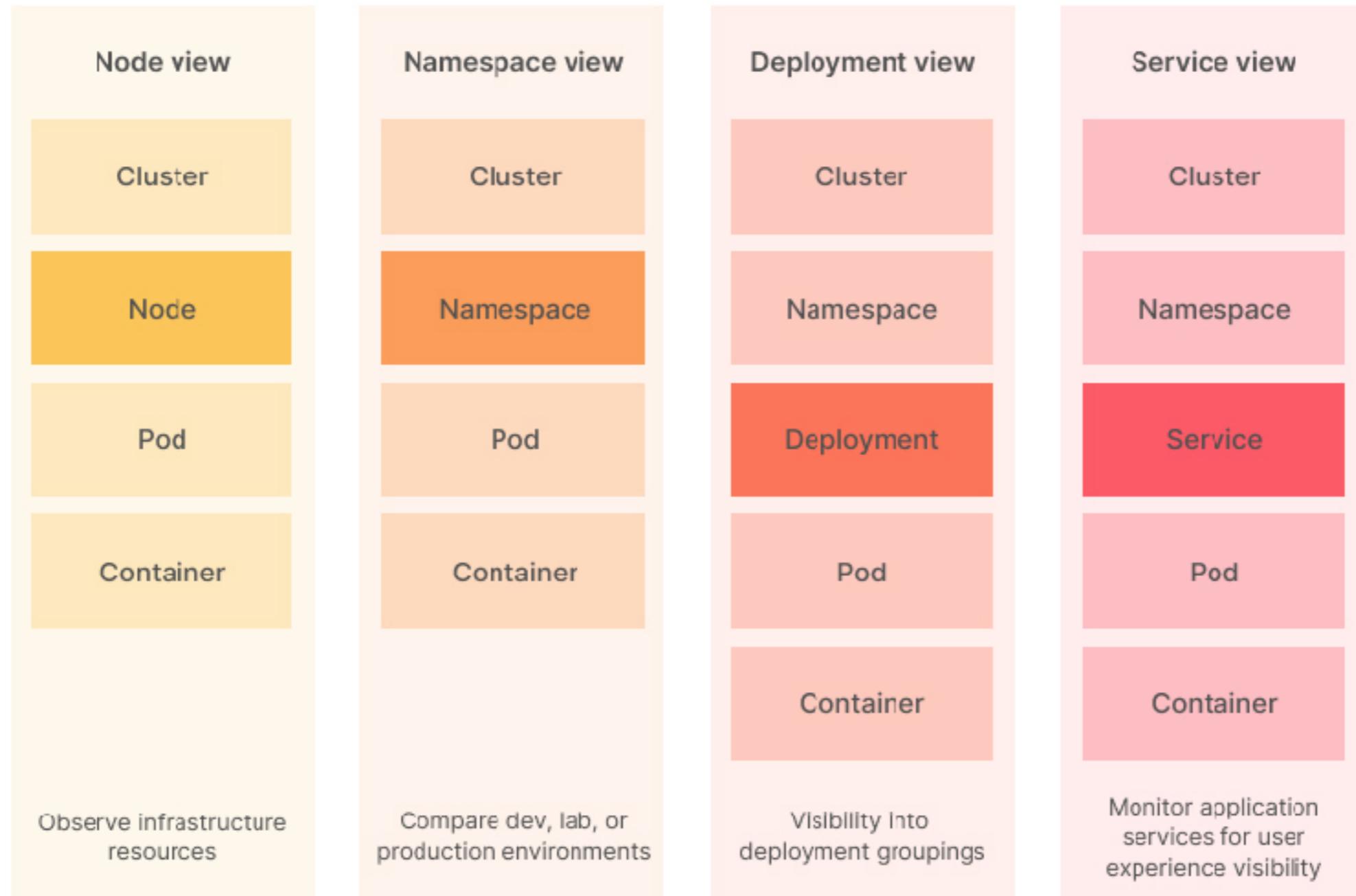
Namespace
view

Deployment
view

Service
view



Kubernetes monitoring ?



<https://grafana.com/solutions/kubernetes/kubernetes-monitoring-introduction/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Performance metrics ?

Node resources utilization

Object capacity

Health metrics

Storage health

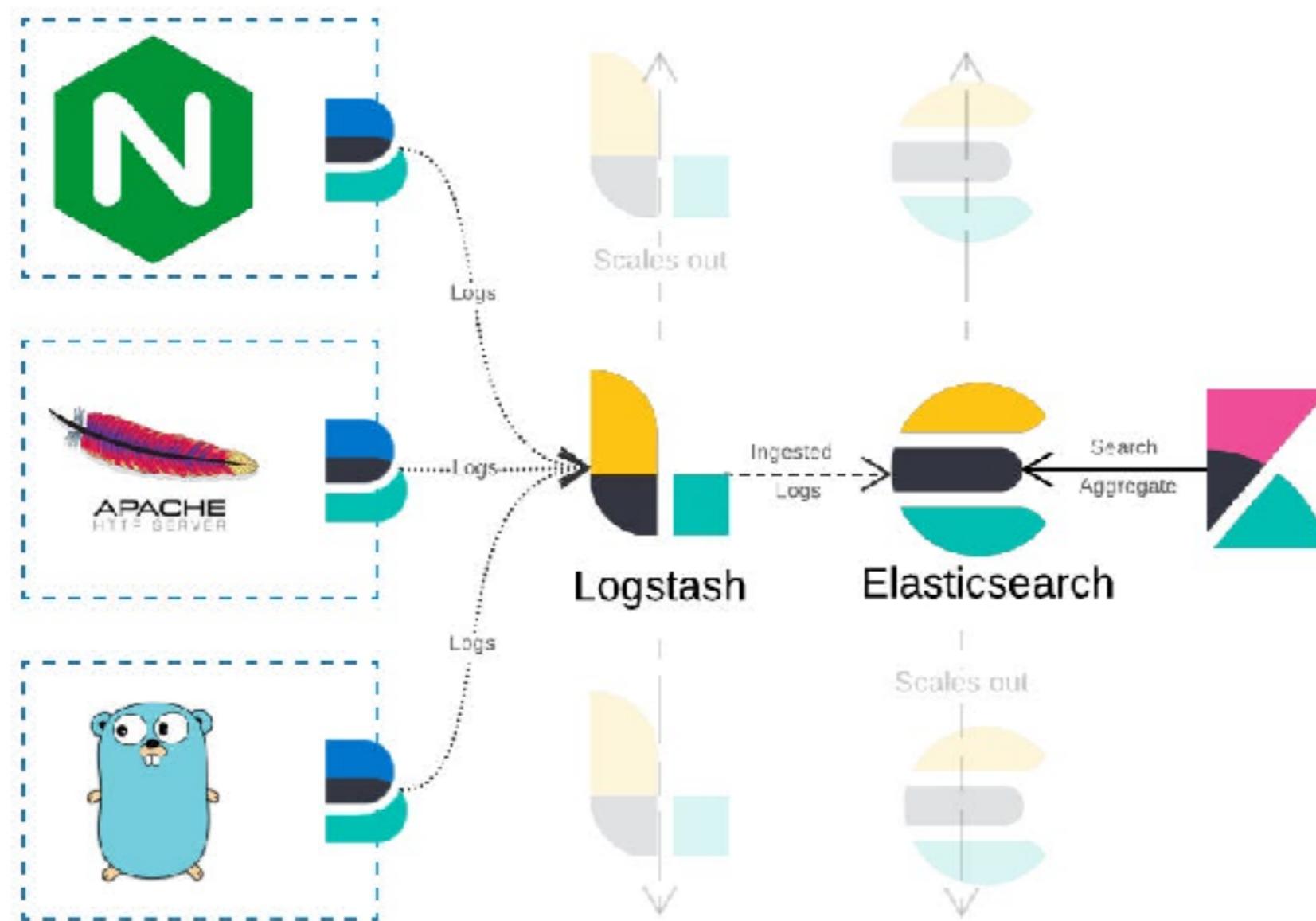
Kubelet
metrics



Technology Stacks ?



ELK Stack



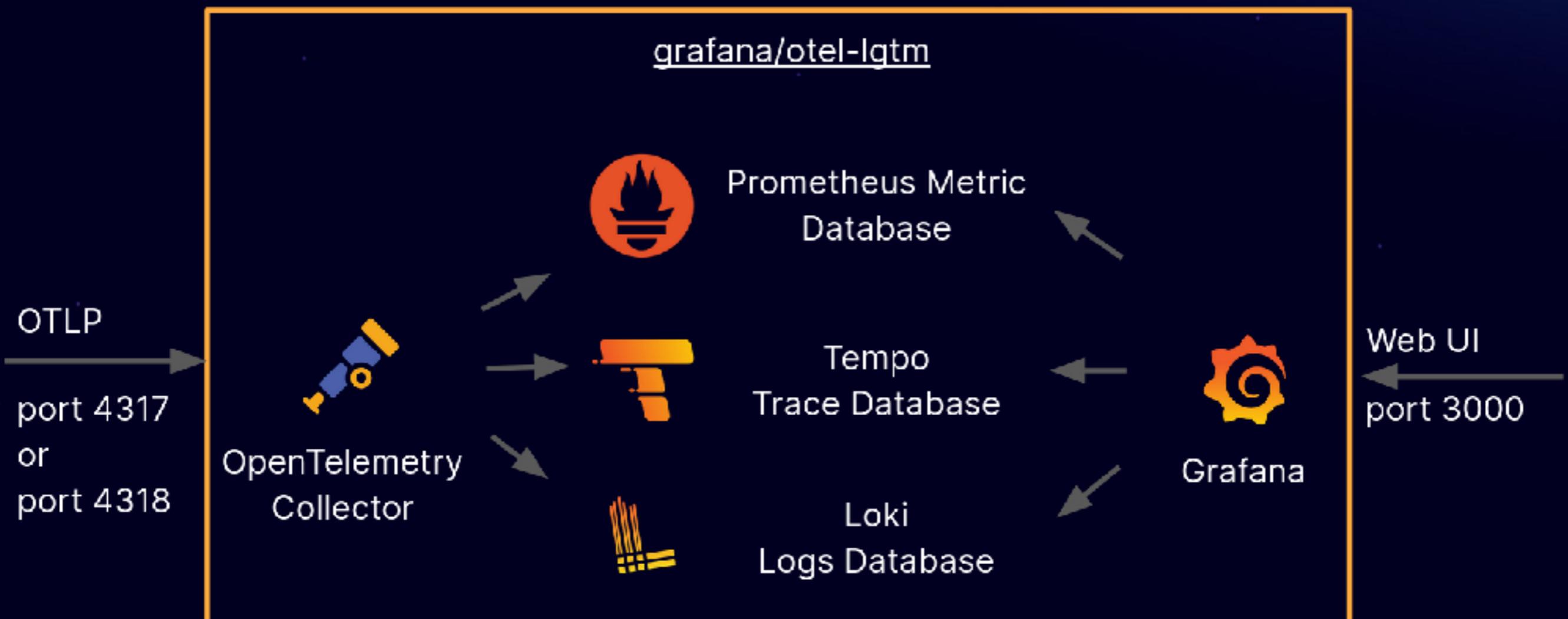
<https://www.elastic.co/elastic-stack>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

LGTM Stack



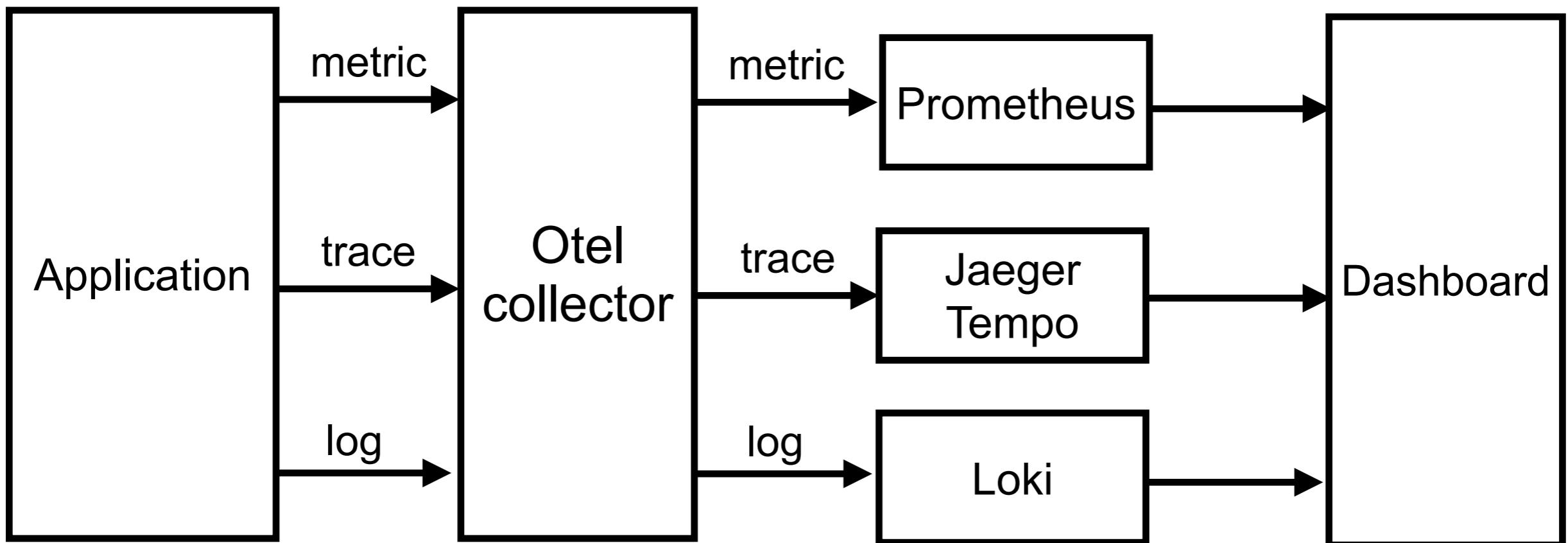
<https://github.com/grafana/docker-otel-lgtm/>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

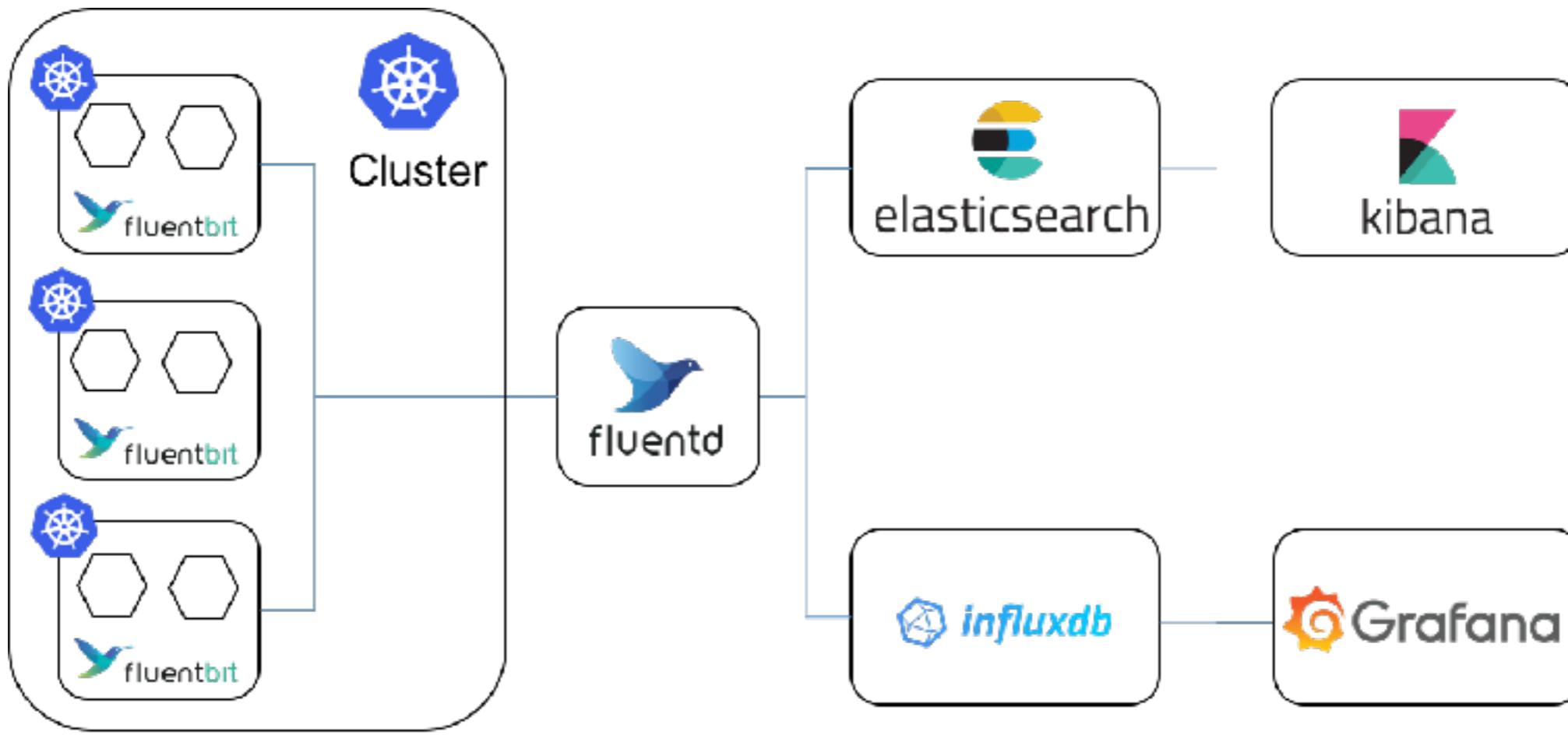
OpenTelemetry



<https://github.com/up1/workshop-lgtm-go>



Fluent and FluentBit



Data
Collection

Data
Aggregation
& Processing

Indexing &
storage

Analysis &
visualization

<https://dzone.com/articles/fluentd-vs-fluent-bit-side-by-side-comparison-logz>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Kubernetes Operations

Cluster
management

Application
management

Security



Steps to upgrade your cluster ?

<https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/>



Check cluster use deprecated API ?



Kubent

<https://github.com/doitintl/kube-no-trouble>



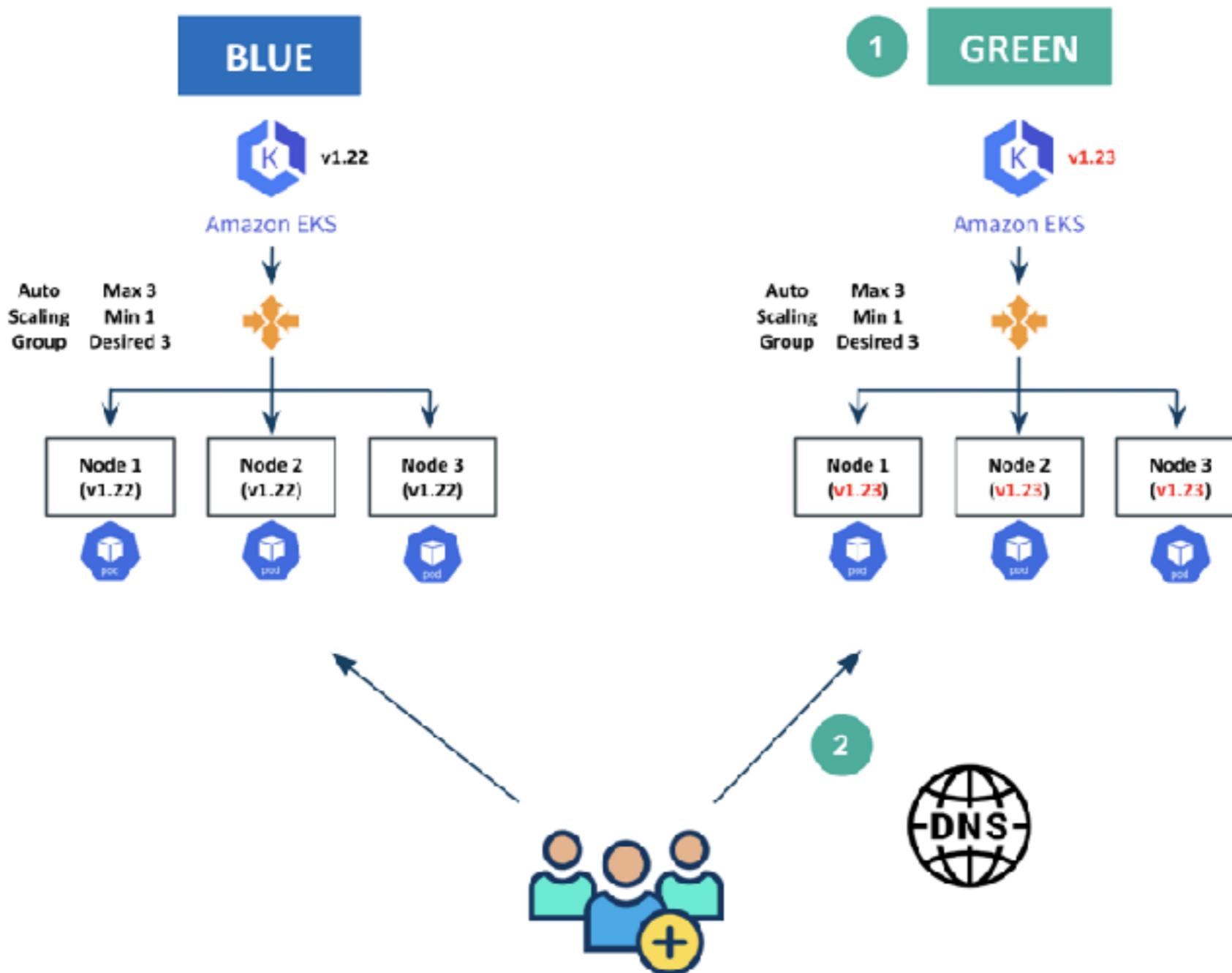
Upgrade cluster strategies ?

Rolling update

Blue/Green



Blue/Green update ?



<https://rafay.co/ai-and-cloud-native-blog/mastering-amazon-eks-upgrades-with-rafays-kubernetes-operations-platform/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Try to upgrade in Testing environment !!



Troubleshooting Cluster !!

Cluster level

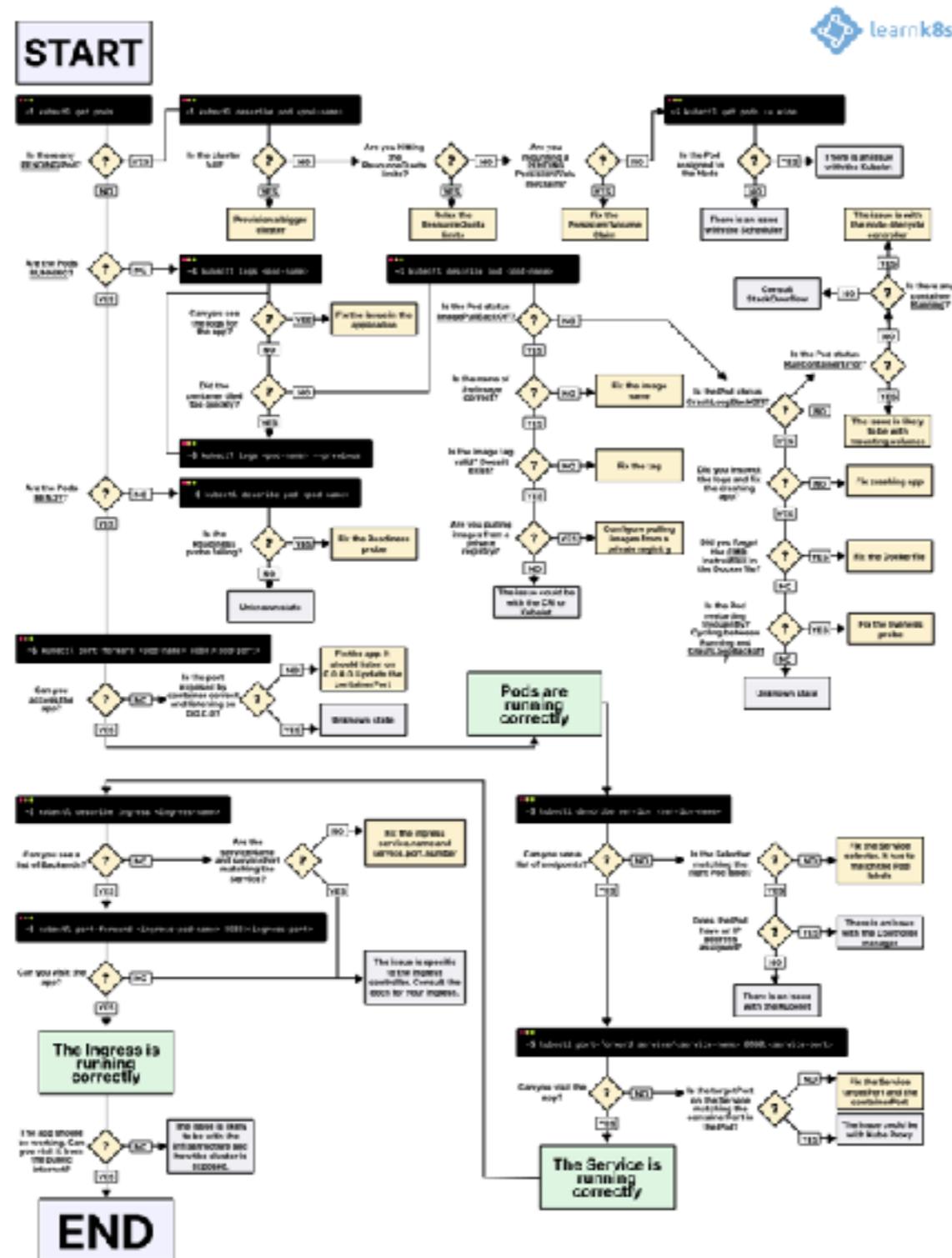
Application level

<https://kubernetes.io/docs/tasks/debug/debug-cluster/>



Kubernetes

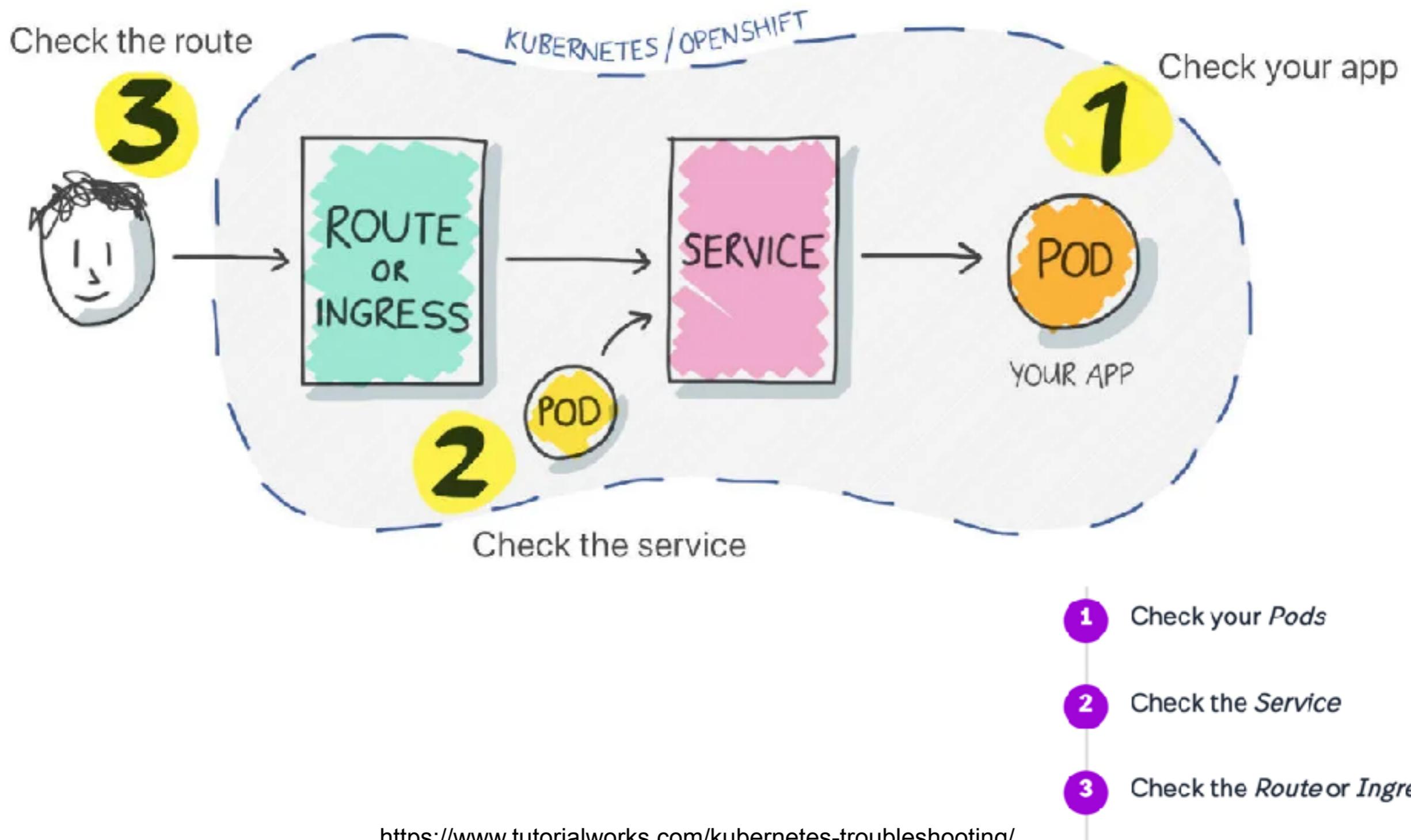
© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.



<https://learnkube.com/troubleshooting-deployments>



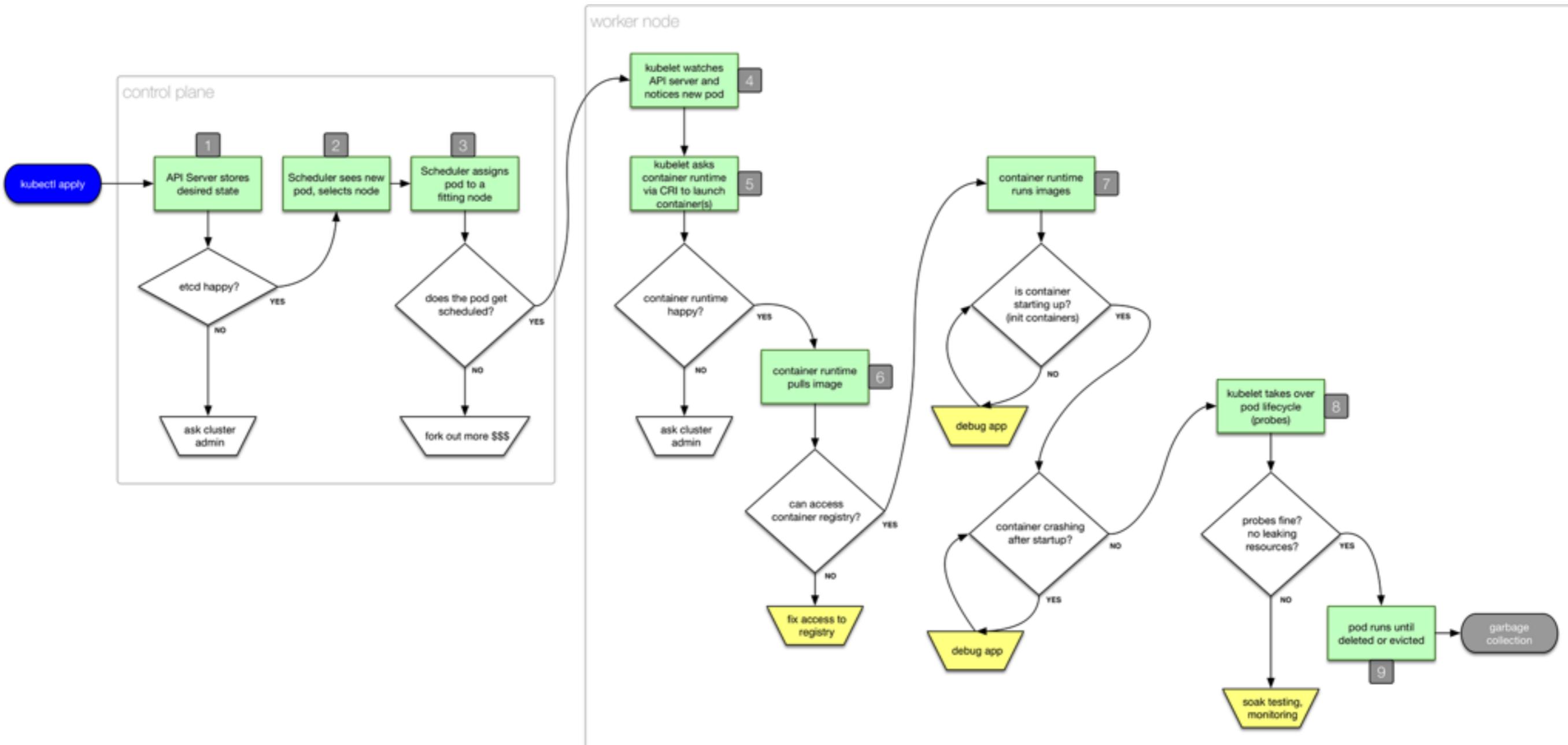
TROUBLESHOOTING APPS IN KUBERNETES



<https://www.tutorialworks.com/kubernetes-troubleshooting/>



Pods's Lifecycle ?



<http://troubleshooting.kubernetes.sh/>



Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Tools



KOps

Installation cluster
Upgrade and manage cluster



<https://github.com/kubernetes/kops>



K8sGPT



Giving Kubernetes Superpowers to Everyone

K8sGPT is an AI-powered tool that helps diagnose and fix Kubernetes issues with intelligent insights and automated troubleshooting.

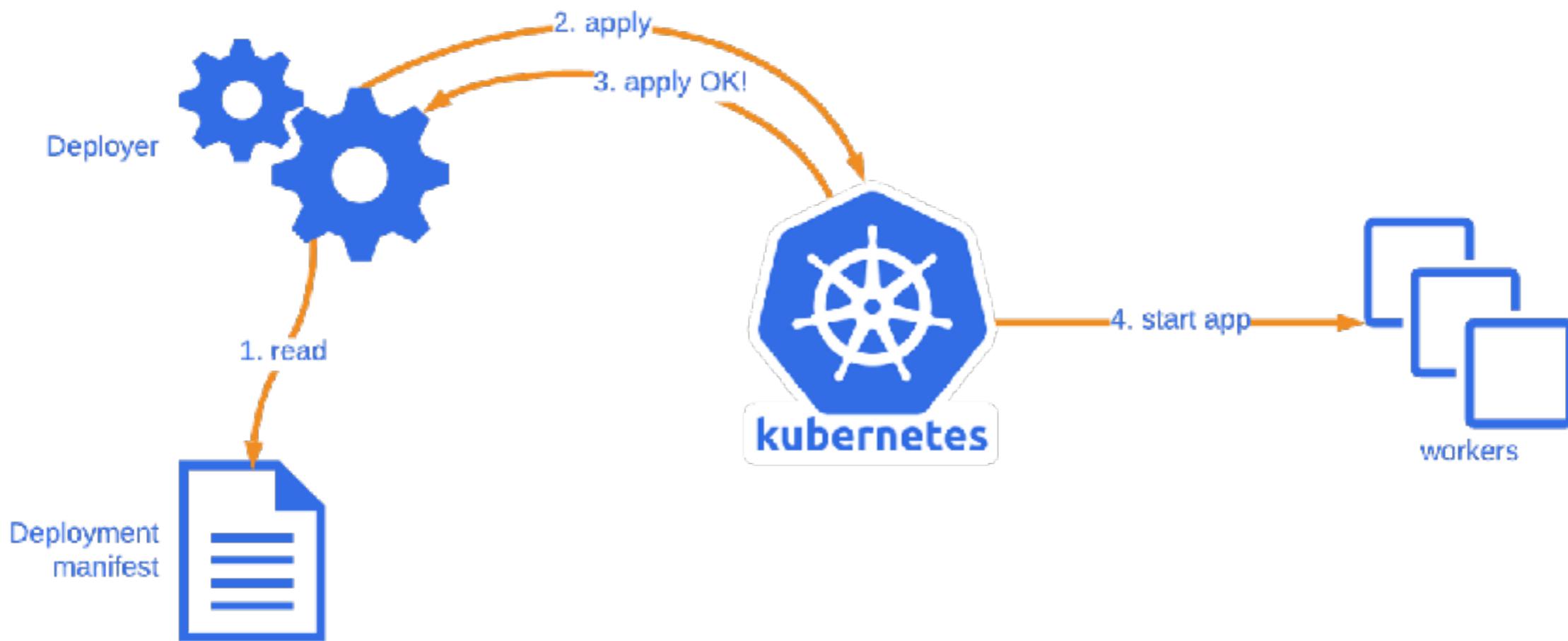
<https://k8sgpt.ai/>



Kubernetes best practices ?

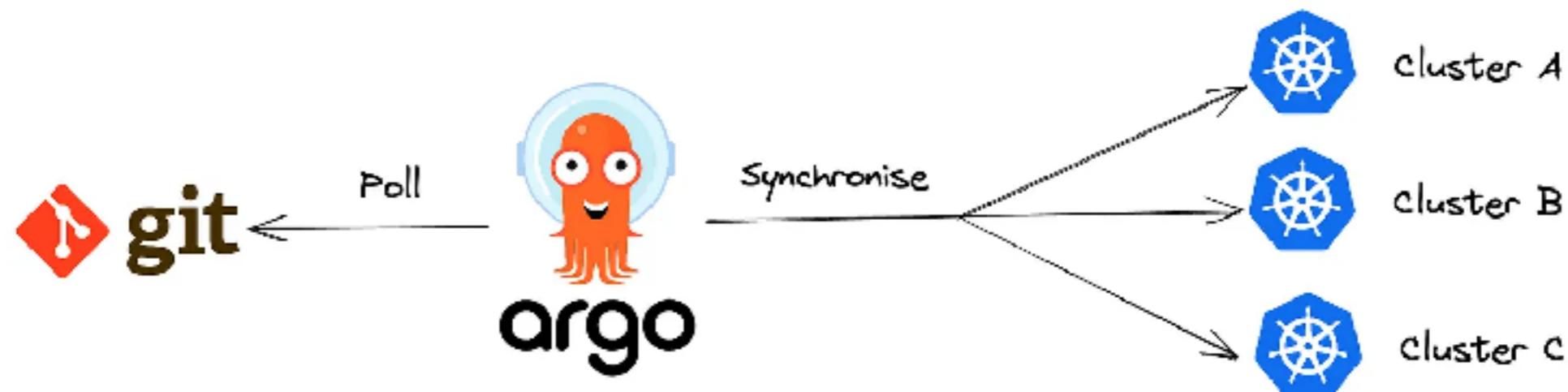


Simple deploy with kubectl !!



Use GitOps approach

Use git repository as the source of truth for defining the desired application state

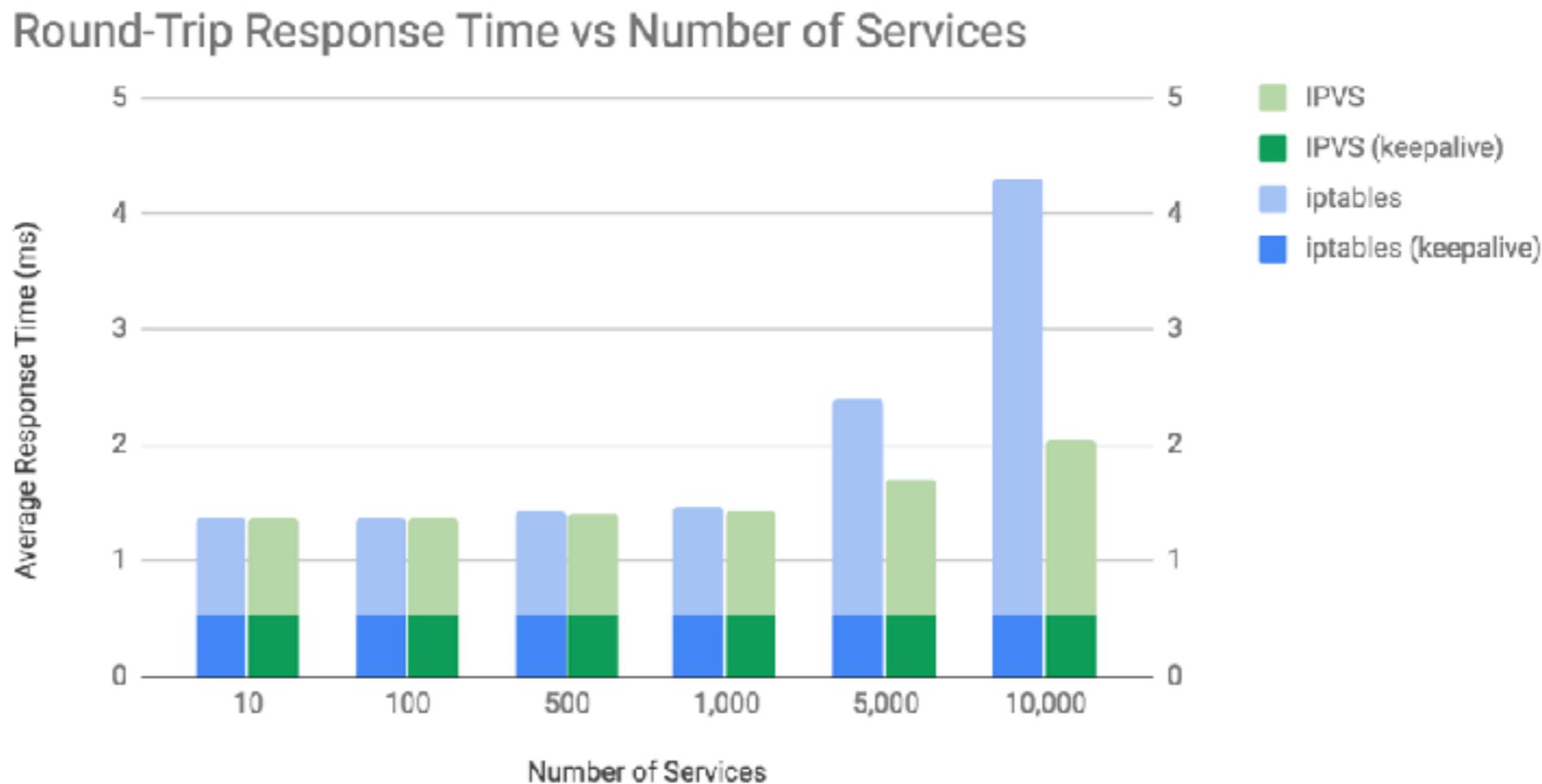


<https://redhat-scholars.github.io/argocd-tutorial/argocd-tutorial/index.html>



Use IPVS in service

IPVS (IP Virtual Server)
Linux kernel load balancer over layer 4



<https://www.tigera.io/blog/comparing-kube-proxy-modes-iptables-or-ipvs/>



Don't keep sensitive data in K8s secret

K8s secret == base64 encoding

AWS
secret manager

Hashicorp vault

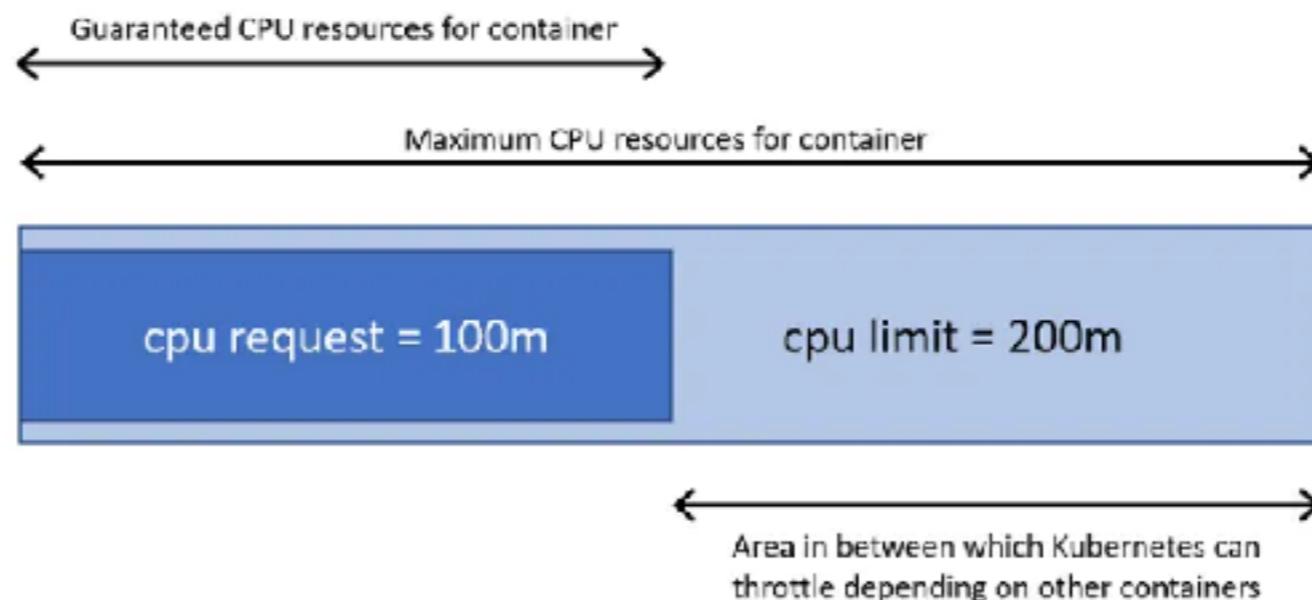
Azure key vault

Google
secret manager



Always set resources limit

```
resources:  
  limits:  
    cpu: 1  
    memory: 1Gi  
  requests:  
    cpu: 500m  
    memory: 512Mi
```



Use liveness and readiness probes

Liveness

Readiness

Startup

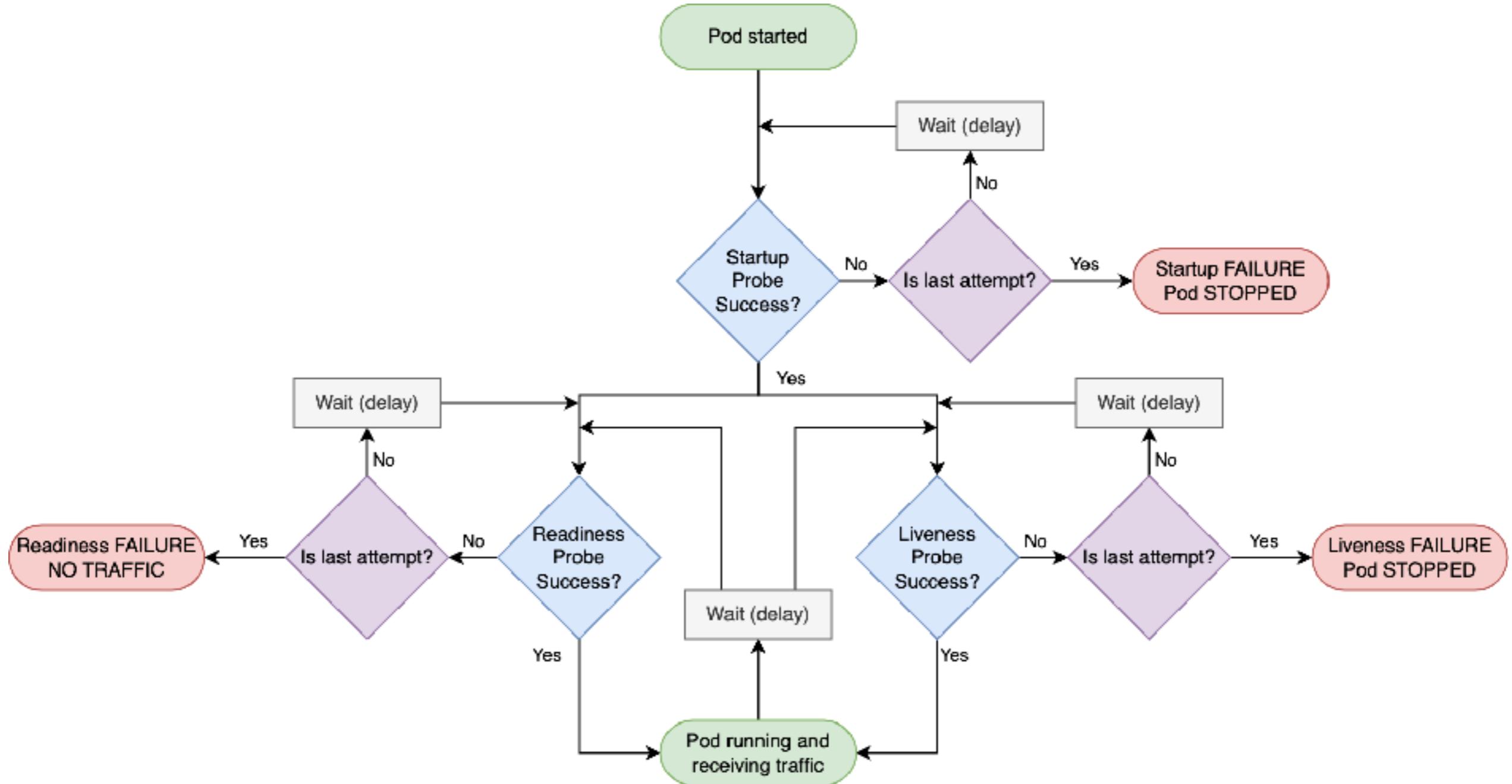
<https://cloud.google.com/blog/products/container-kubernetes/kubernetes-best-practices-setting-up-health-checks-with-readiness-and-liveness-probes>



Kubernetes

© 2017 - 2025 Siam Chamnankit Company Limited. All rights reserved.

Startup probe



<https://andrewlock.net/deploying-asp-net-core-applications-to-kubernetes-part-6-adding-health-checks-with-liveness-readiness-and-startup-probes/>



Kubernetes

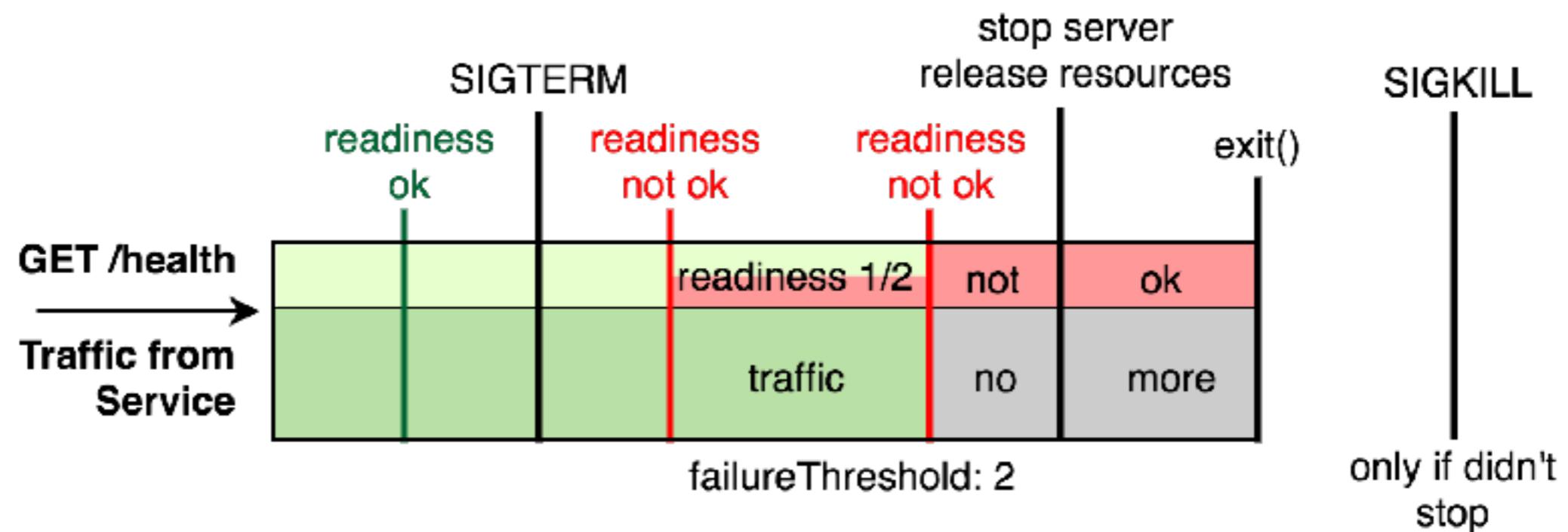
© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

Terminating with grace

<https://learnkube.com/graceful-shutdown>



Graceful Shutdown in Kubernetes



<https://learnkube.com/graceful-shutdown>

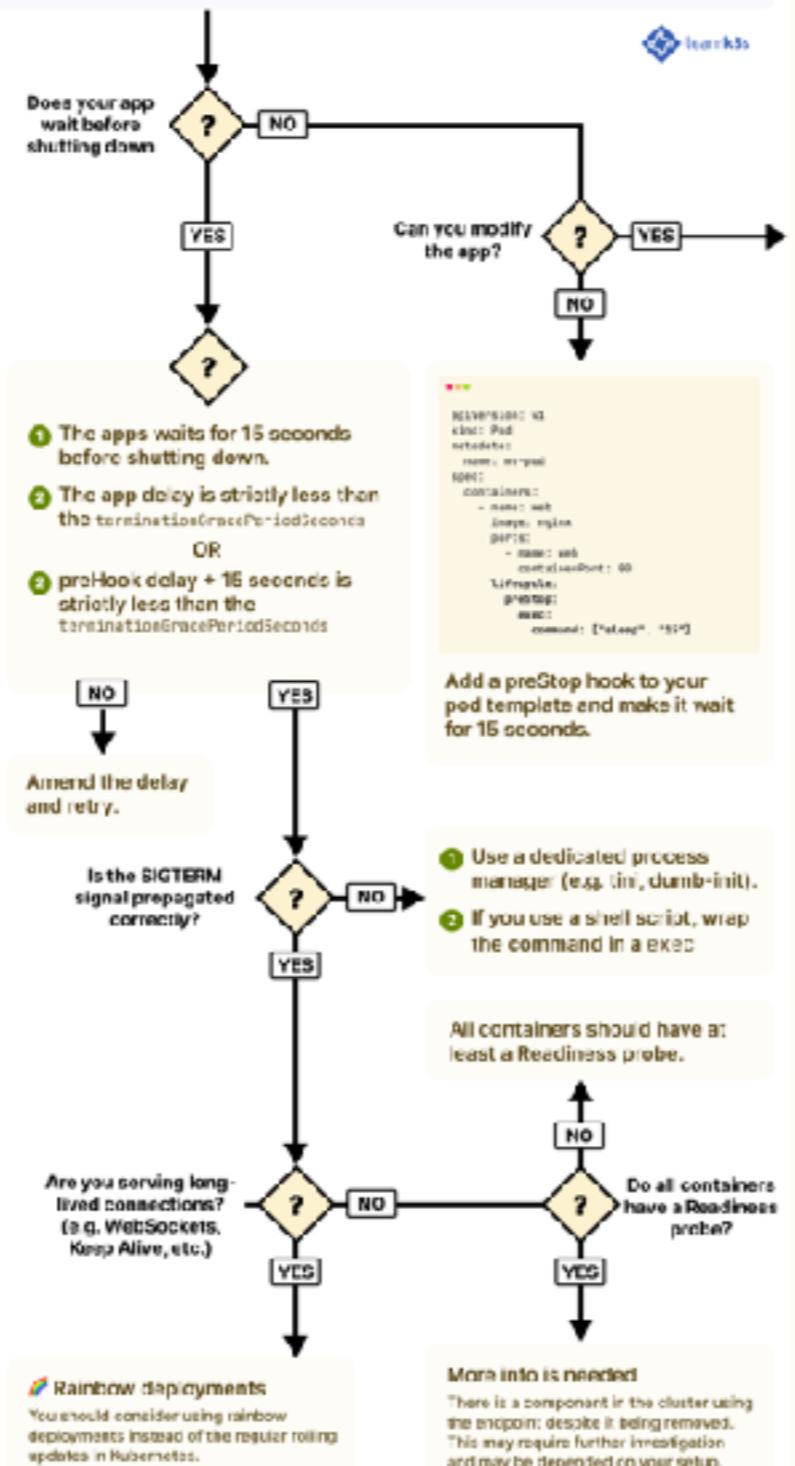


Kubernetes

© 2017 - 2025 Siam Chamnkit Company Limited. All rights reserved.

I SEE 5XX ERRORS IN MY KUBERNETES DEPLOYMENTS

This flowchart helps you debug issues with pod graceful shutdown in Kubernetes.



You should wait for 15 seconds and, in the next 15 seconds, close any connections and shut down the application.

package main
import ("log"
"os"
"os/signal"
"syscall")

func main() {
 sig := os/signal.Signals(SIGHUP, SIGTERM)
 // Registers the channel.
 signal.Notify(sig, syscall.SIGTERM)

 go func() {
 sig := &sig
 for i := range sig {
 log.Println("Caught SIGTERM, shutting down")
 // Finish any outstanding requests, then...
 done = true
 }
 }()

 log.Println("Starting application")
 // Main logic goes here
 // done
 log.Println("Exiting")
}

Python

```
import signal, time, os  
  
def shutdown(signum, frame):  
    print('Caught SIGTERM, shutting down')  
    # Finish any outstanding requests, then...  
    exit(0)  
  
if __name__ == "__main__":  
    # Register handler  
    signal.signal(signal.SIGTERM, shutdown)  
    # THIS logic goes here
```

Node.js

```
process.on('SIGTERM', () => {  
    console.log('The service is about to shut down')  
  
    // Finish any outstanding requests, then...  
    process.exit(0);  
});
```

Java

```
public class App {  
    public static void main(String[] args) {  
        var shutdownHandler = new Thread() {  
            public void run() {  
                // Main logic goes here  
            }  
        };  
        shutdownHandler.start(); additionalShutdown[shutdownHandler] = true;  
    }  
  
    internal static LifecycleEventService lifecycleService {  
        private final ThreadLocal<LifecycleEventService> lifecycle;  
  
        public LifecycleEventService lifecycle() {  
            logger = LifecycleEventService + logger;  
            ThreadLocal<LifecycleEventService> lifecycle = ThreadLocal.withInitial(() -> {  
                logger = null;  
                return new LifecycleEventService();  
            });  
            return lifecycle.get();  
        }  
        public void shutdown() {  
            logger.info("Shutting down application");  
            ApplicationLifecycleEventBus.unregister(lifecycle.get());  
            ApplicationLifecycleEventBus.unregister(lifecycle.get());  
            ApplicationLifecycleEventBus.unregister(lifecycle.get());  
        }  
    }  
}  
  
private void shutdown() {  
    private void unscope() {  
        // Main logic goes here  
    }  
    private void unscope() {  
        // Main logic goes here  
    }  
}
```

.NET Core

<https://learmk8s.io/graceful-shutdown>



Production Checklist ?

Kubernetes production best practices

A curated checklist of best practices
designed to help you release to
production

<https://learnkube.com/production-best-practices>



Q/A

