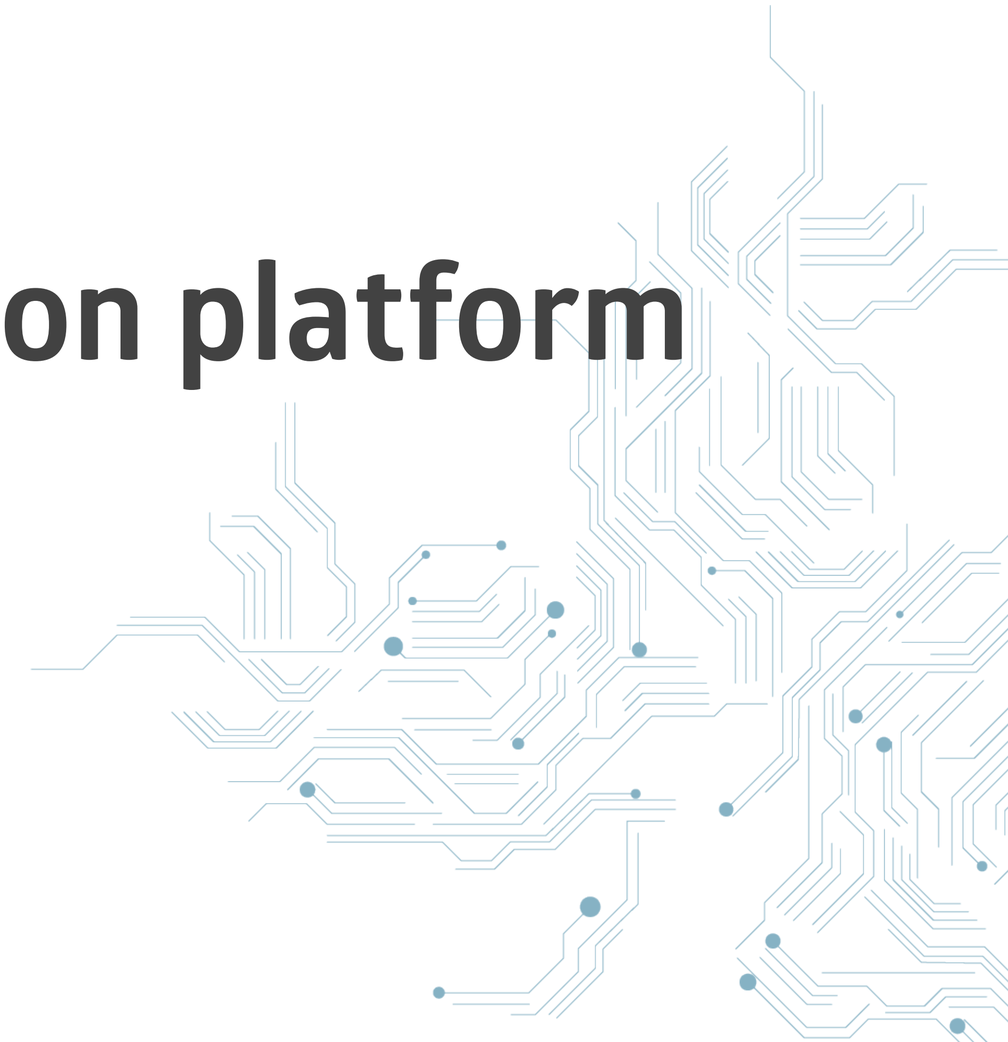
SysEleven
engage · build · run

Kubernetes Workshop

kubernetes

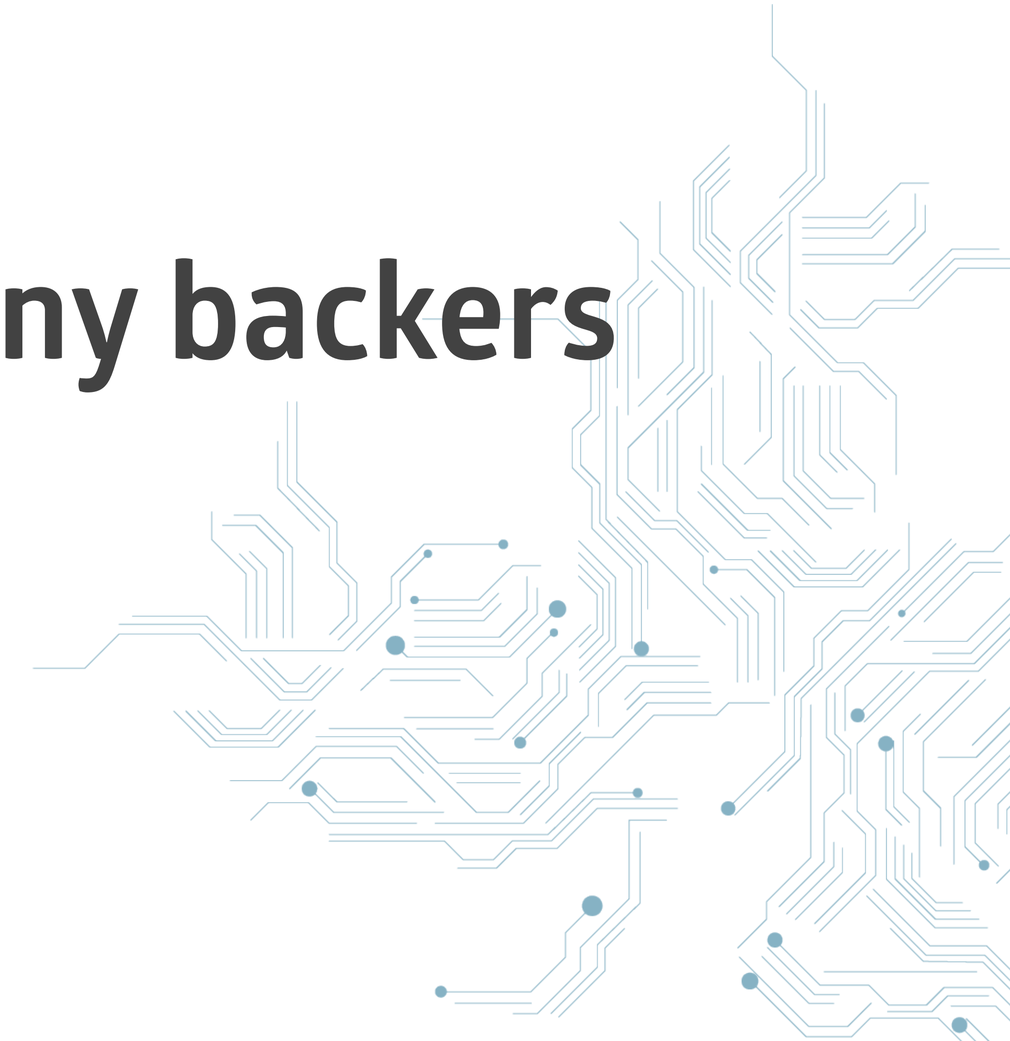
Container orchestration platform



Deploy, run and scale your services in isolated containers

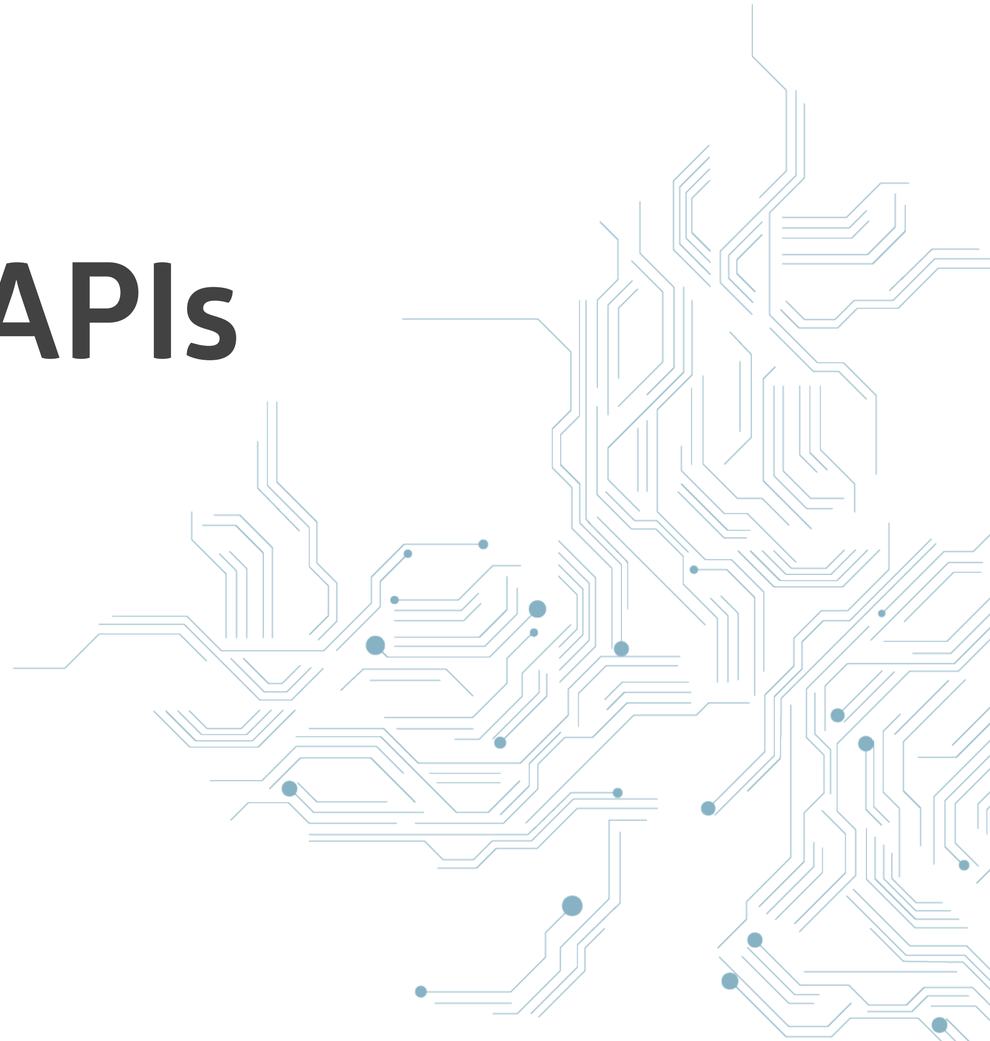
Very Powerful

Lot’s of large company backers



No vendor lock in

Standardized APIs



Runs on

Your laptop

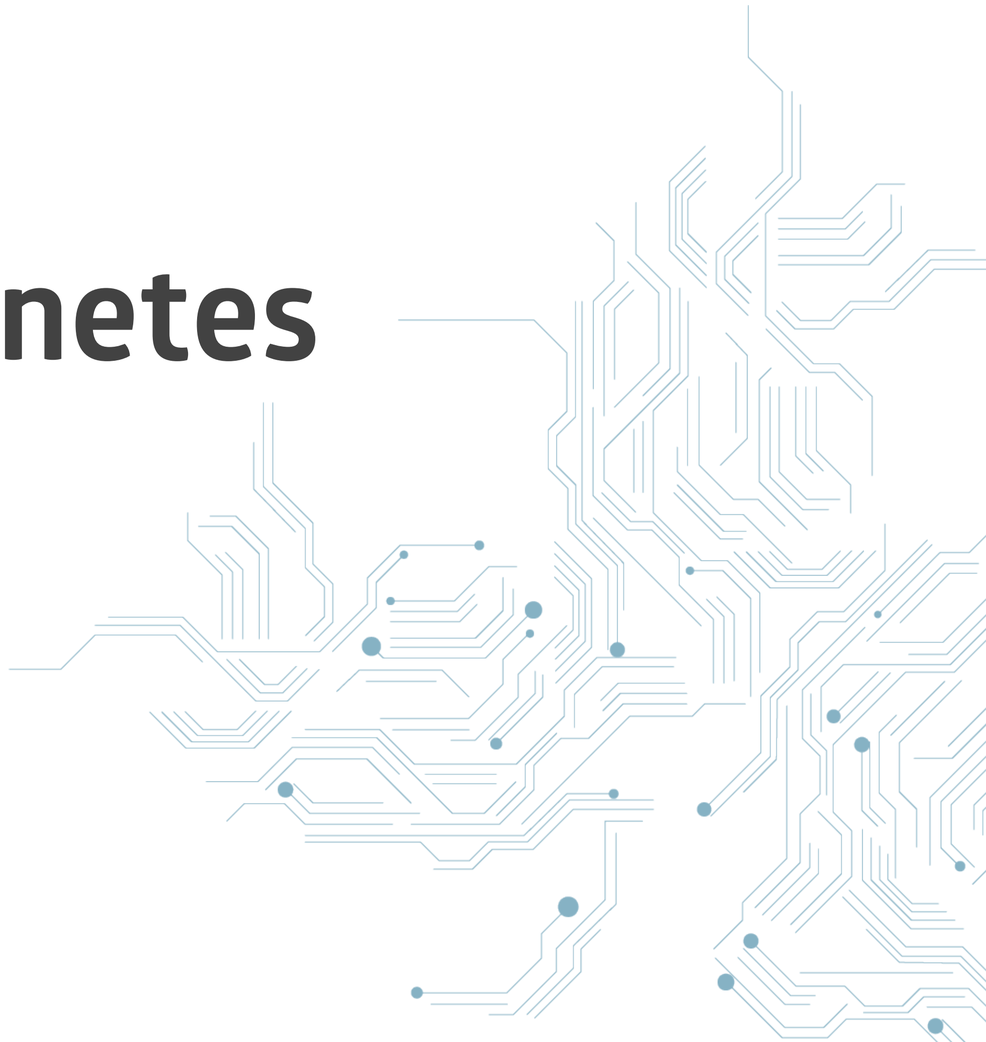

Kubernetes
General
File Sharing
Disk
Advanced
Proxies
Daemon
Kubernetes
Reset
Enable Kubernetes
Start a Kubernetes single-node cluster when starting Docker Desktop.
Deploy Docker Stacks to Kubernetes by default
Make Kubernetes the default orchestrator for "docker stack" commands
(changes "~/.docker/config.json")
Show system containers (advanced)
Show Kubernetes internal containers when using Docker commands.
Apply
Docker Desktop is running

Bare metal

Cloud Providers

And if you don't want to install and maintain Kubernetes yourself

Managed Kubernetes



CNCF Cloud Native Interactive Landscape

CLOUD NATIVE COMPUTING FOUNDATION

The Cloud Native Trail Map (png, pdf) is CNCF's recommended path through the cloud native landscape. The cloud native landscape (png, pdf), serverless landscape (png, pdf), and member landscape (png, pdf) are dynamically generated below. Please open a pull request to correct any issues. Greyed logos are not open source. Last Updated: 2020-11-09 21:11:28Z

You are viewing 46 cards with a total of 788 stars, market cap of $6.68T and funding of $1.55B.

Landscape | Card Mode | Serverless | Members

Tweet 1405

Platform - Certified Kubernetes - Hosted (46)

AKS Engine for Azure Stack ★2 MCap: $1.64T
Microsoft

Alibaba Cloud Container Service for Kubernetes MCap: $832.24B
Alibaba Cloud

Amazon Elastic Container Service for Kubernetes (EKS) MCap: $1.61T
Amazon Web Services

Azure (AKS) Engine ★763 MCap: $1.64T
Microsoft

Azure Kubernetes Service (AKS) MCap: $1.64T
Microsoft

Bizmicro
KBSYS

BoCloud BeyondContainer Funding: $29.77M
Bocloud

Catalyst Kubernetes Service ★2
Catalyst Cloud

Cloud 66 Maestro Funding: $2.24M
Cloud 66

DigitalOcean Kubernetes Funding: $455.61M
DigitalOcean

eBaoCloud Funding: $36M
eBaoTech

ELASTX Private Kubernetes
ELASTX

Google Kubernetes Engine (GKE) MCap: $1.11T
Google

Huawei Cloud Container Engine (CCE)
Huawei Technologies

IBM Cloud Kubernetes Service MCap: $103.31B
IBM

INNOVO Managed Kubernetes Engine
INNOVO Cloud

Inspur Cloud Container Engine and Inspur Open Platform MCap: $6.81B
Inspur Group

Inspur ICKS MCap: $6.81B
Inspur Group

Inspur Open Platform for ARM MCap: $6.81B
Inspur Group

InsureMO Funding: $36M
eBaoTech

JD Cloud JCS for Kubernetes MCap: $127.49B
JD.com

Kingsoft Container Engine MCap: $6.76B
Kingsoft Japan

Launcher Tech LStack Container Service for Kubernetes
Hangzhou Launcher Technology

Linaro Developer Cloud Kubernetes Service ★1
Linaro

Linode Kubernetes Engine
Linode

Mail.Ru Cloud Containers MCap: $14.24B
Mail.Ru Group

NIFCLOUD Hatoba MCap: $26.12B
Fujitsu

Nirmata Managed Kubernetes
Nirmata

Nutanix Karbon MCap: $5.11B
Nutanix

Oracle Container Engine MCap: $180.35B
Oracle

OVH Managed Kubernetes Service Funding: $736.63M
OVHcloud

QingCloud KubeSphere Engine (QKE) ★16 Funding: $280.83M
QingCloud

Rafay Funding: $8M
Rafay Systems

Red Hat OpenShift Dedicated MCap: $103.31B
Red Hat

Red Hat OpenShift on IBM Cloud MCap: $103.31B
IBM

Samsung SDS Kubernetes Service
Samsung SDS

SAP Certified Gardener MCap: $178.56B
SAP

Scaleway Kubernetes Kapsule
Scaleway

StarlingX ★2
OpenStack

SysEleven MetaKube
SysEleven

Tencent Kubernetes Engine (TKE) MCap: $694.19B
Tencent Holdings

UCloud Kubernetes Service（UK8S) MCap: $3.99B
UCloud Information Technology

Ventus Cloud Kubernetes Service
Ventus Cloud

VEXXHOST Kubernetes Container Service ★2
VEXXHOST

WoCloud Container Platform MCap: $22.83B
China Unicom

ZTE TECS OpenPalette MCap: $21.13B
ZTE

Easy setup

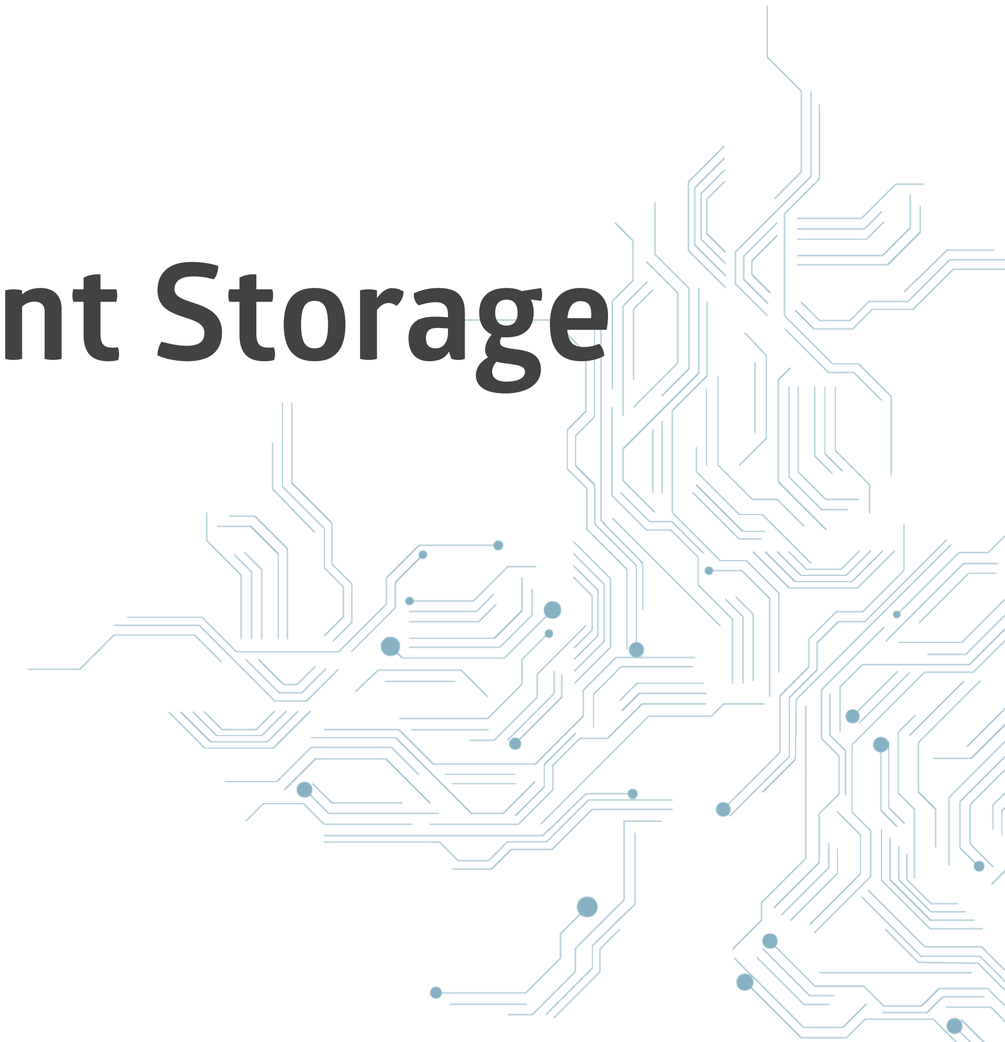
Easy upgrades

Easy scaling

Features

Load Balancing

Distributed Persistent Storage



Backups

But this workshop is about how to use Kubernetes

Learning curve

Agenda


METAKUBE

- Deployments
- CronJobs
- Readiness and Liveness-Probes, NodeSelectors & PodAffinities
- ConfigMaps & Secrets
- External DNS, Let'sEncrypt with cert-manager, nginx-ingress-controller
- Running a MySQL DB
- Helm
- Service Discovery

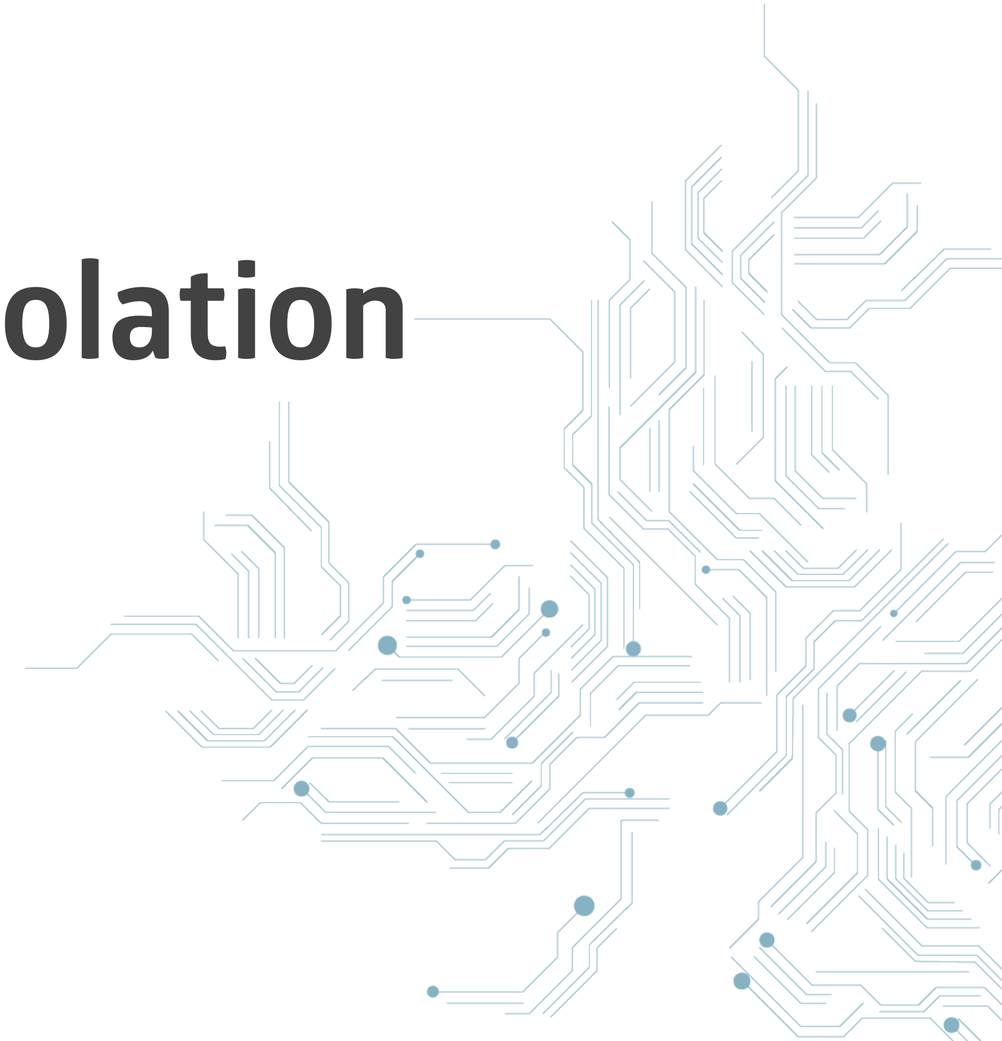
Optionally

- Service Meshes with LinkerD
- Monitoring with Prometheus, Grafana and Alertmanager
- Logging with ElasticSearch, FluentD and Kibana
- GitOps with Flux
- Development with Tilt and Telepresence

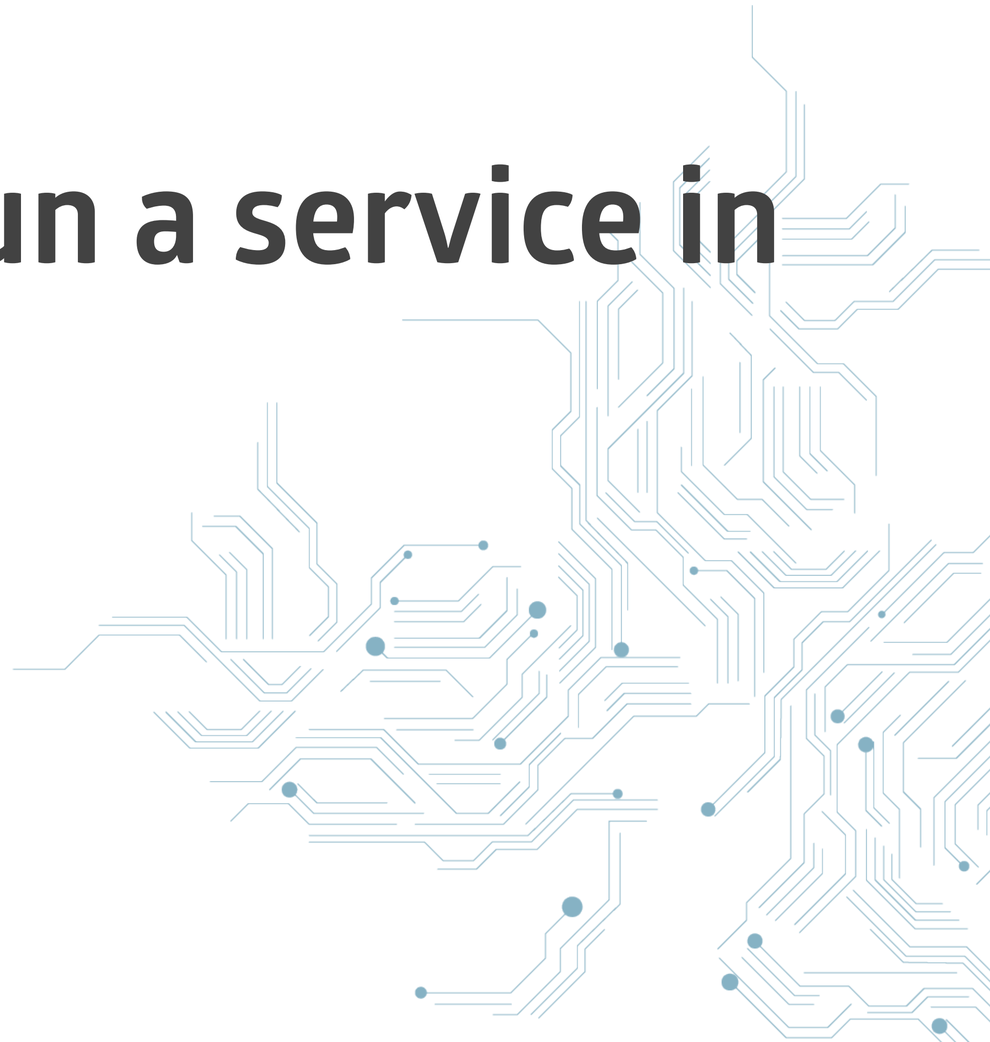
But first

Why containers?

Services run in isolation



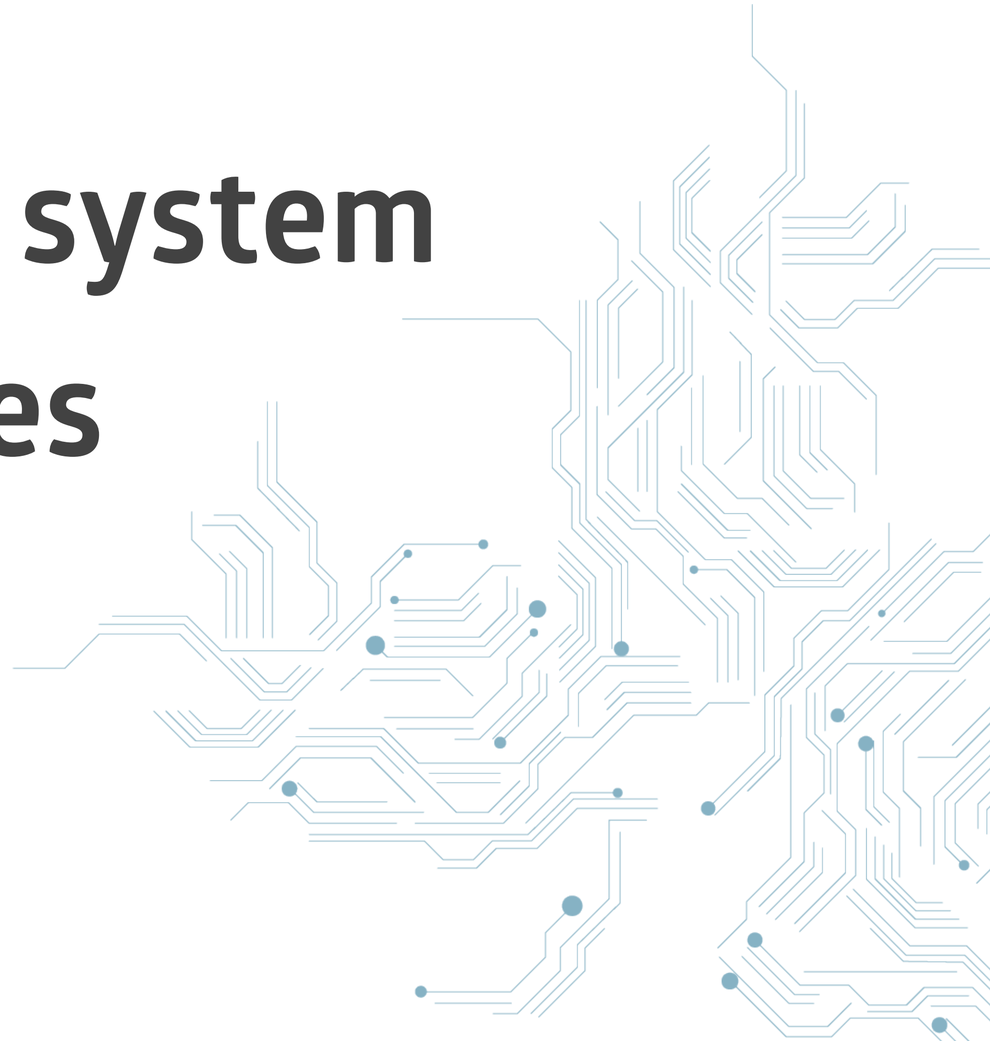
Everything needed to run a service in one image



Make things …

Easier to deploy

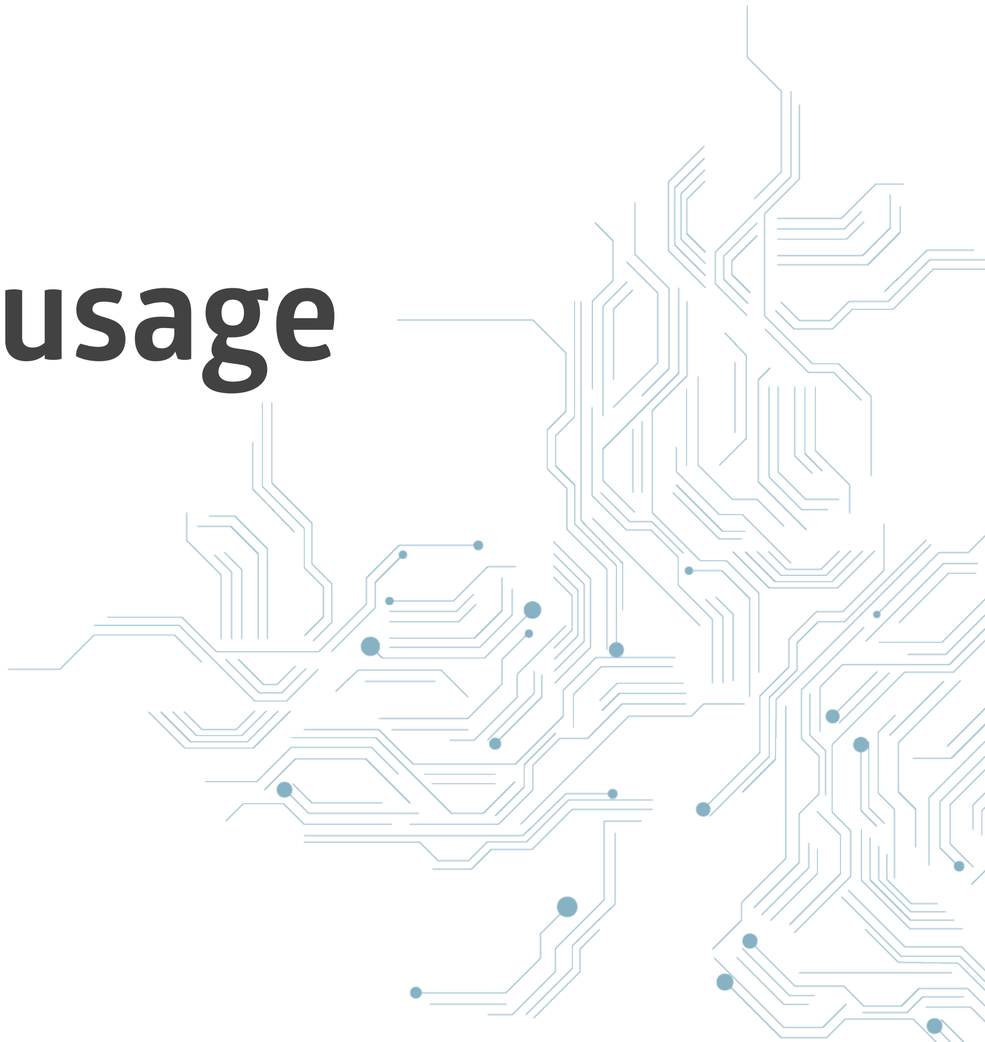
Easier to upgrade system dependencies

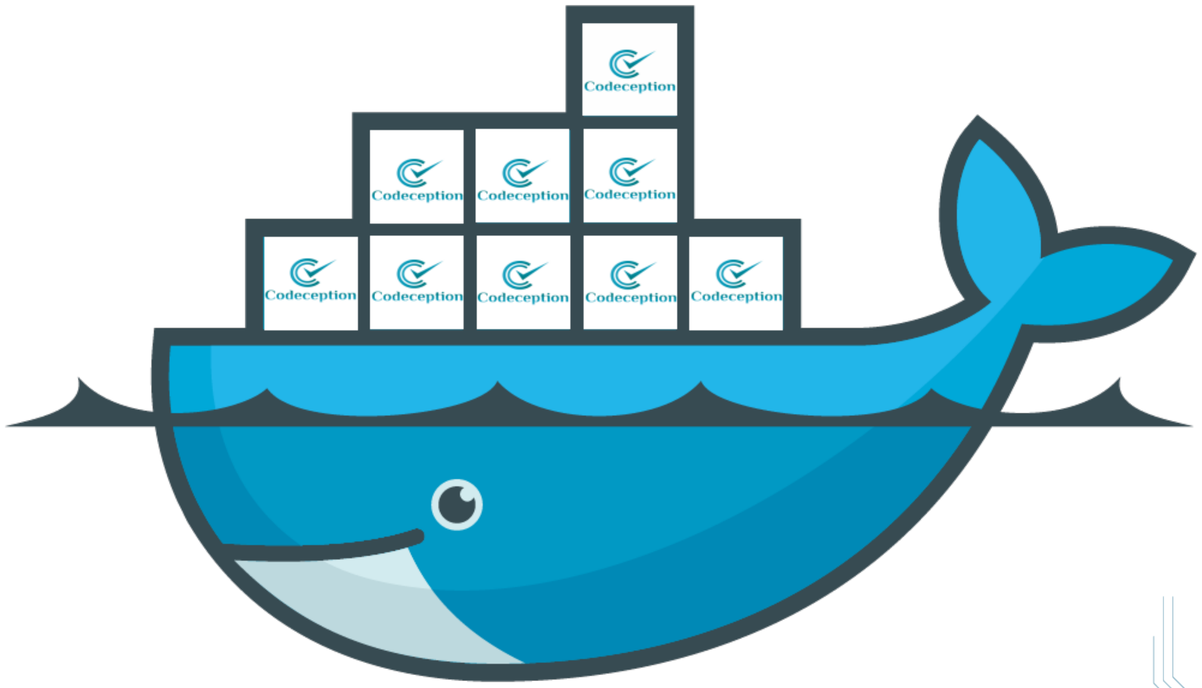


Easier to develop

Easier to scale

Better resource usage




Codeception
Codeception
Codeception
Codeception
Codeception
Codeception
Codeception
Codeception
Codeception
Codeception

docker

Dockerfile

```
FROM php:7.2-apache
WORKDIR /var/www/html

RUN apt-get update -y && \
  apt-get install -y --no-install-recommends curl \
  rm -rf /var/lib/apt/lists/*

ENV TMP_DIR /tmp

COPY . /var/www/html/
```

bash

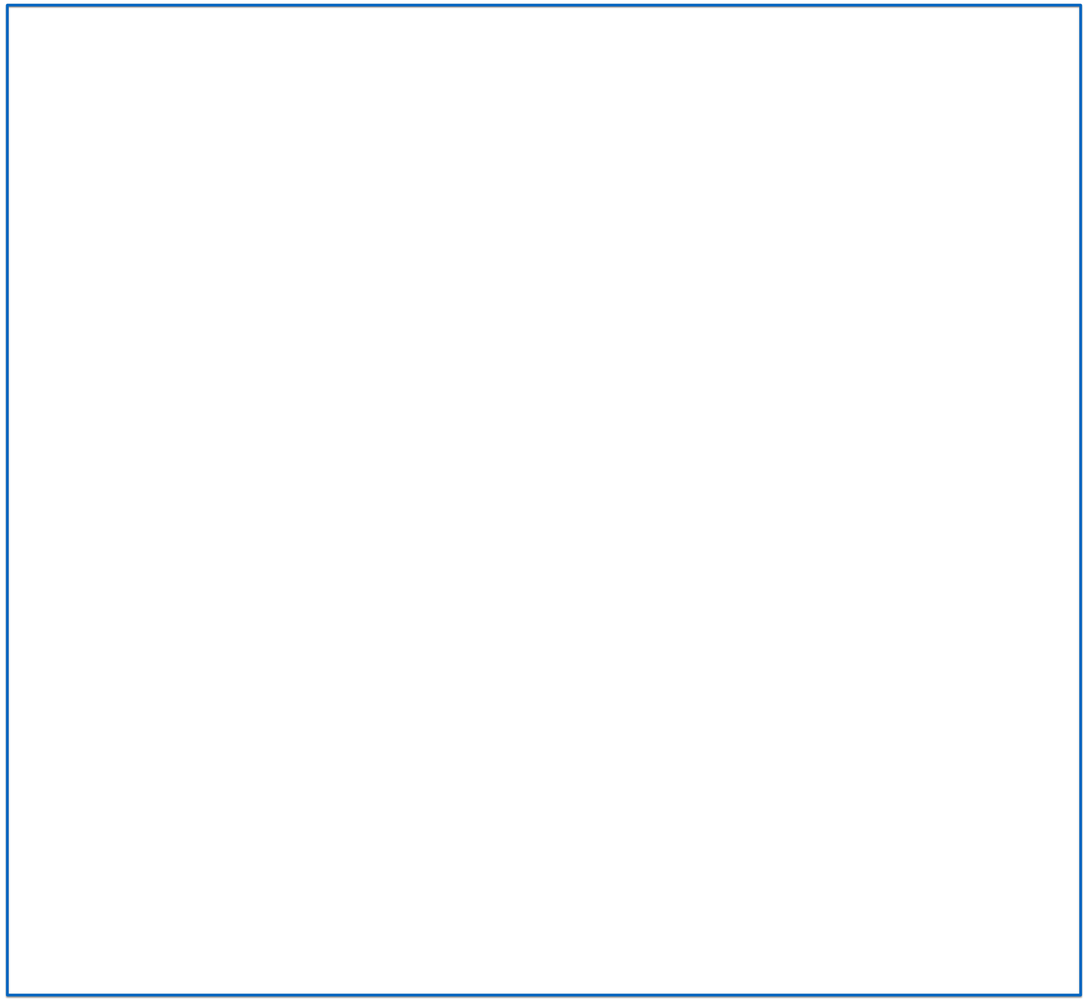
```
$ docker build -t gitlab.syseleven.de/syseleven/symfony-demo:2.0.0 .
```

bash

```
$ docker run -p 8080:80 syseleven/symfony-demo:2.0.0
$ docker push syseleven/symfony-demo:2.0.0
```

Kubernetes helps you to run and deploy containers

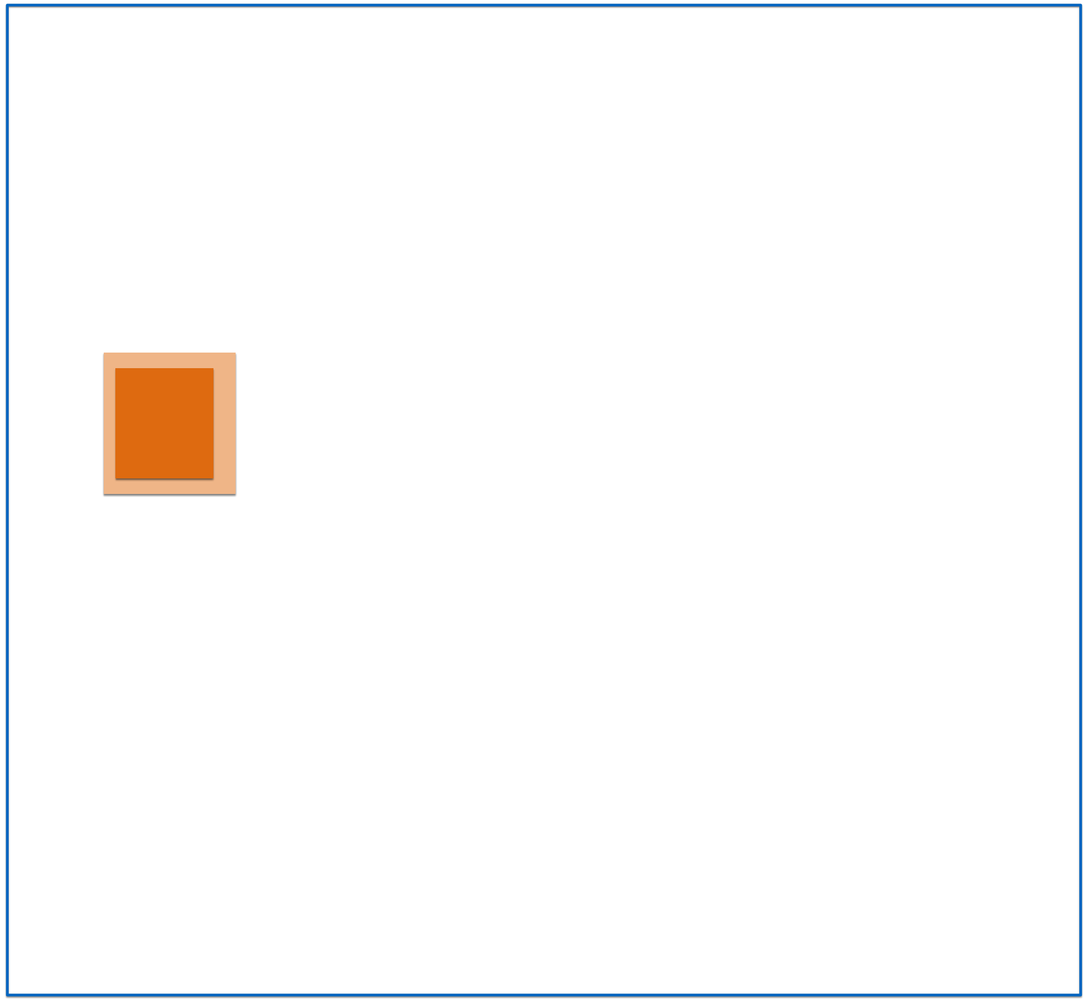
Let’s define some core concepts and terminology first



Kubernetes Cluster

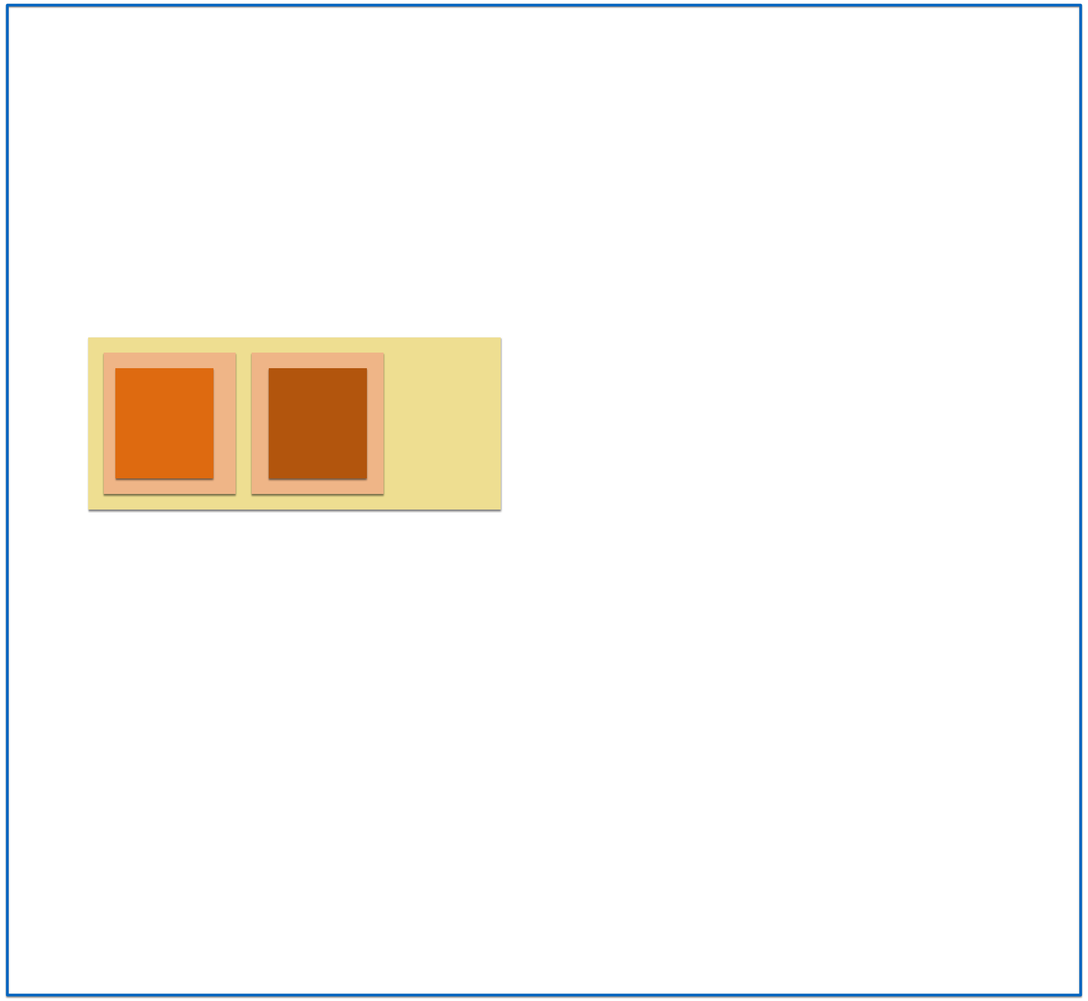
Image

- A docker image built from a Dockerfile that contains everything a service needs to run



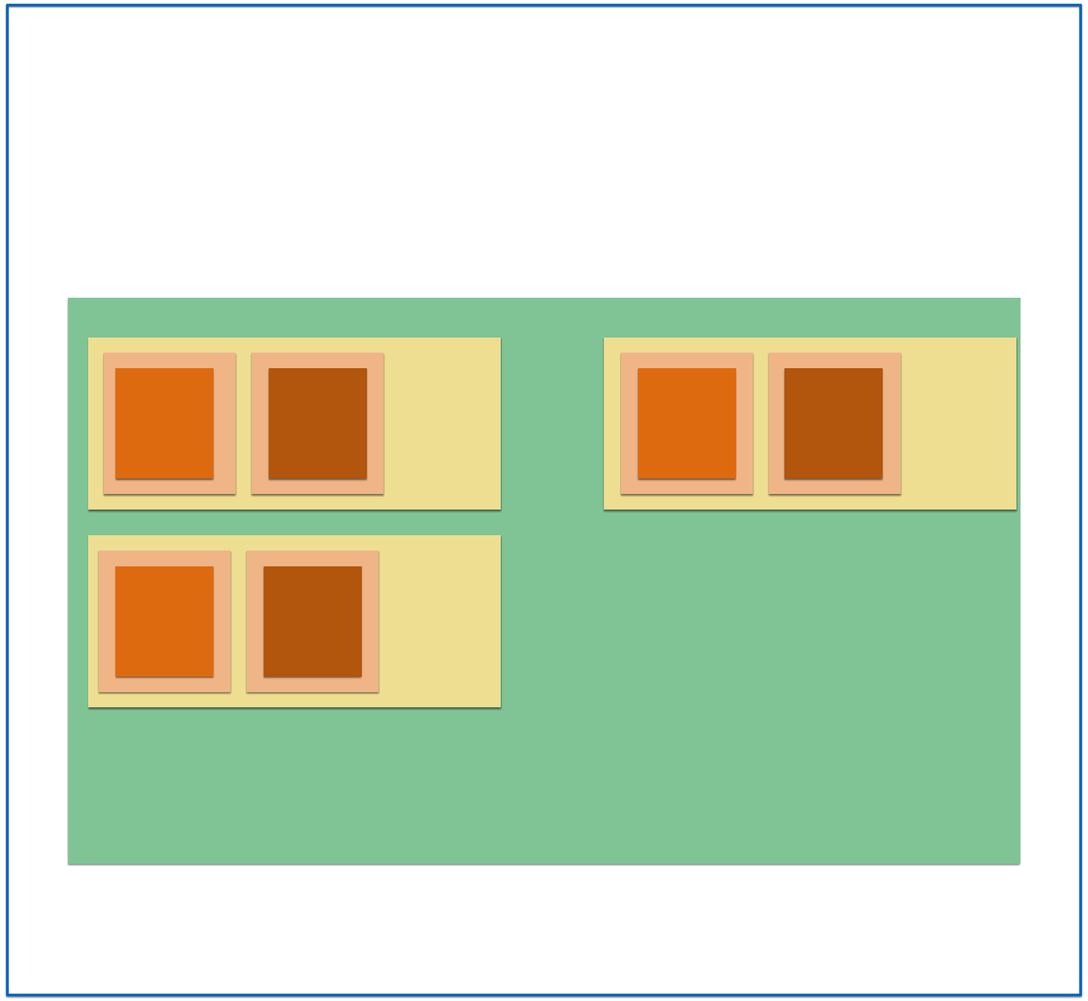
Container

- A container runs a docker image.
- Only 1 process can run inside of a container



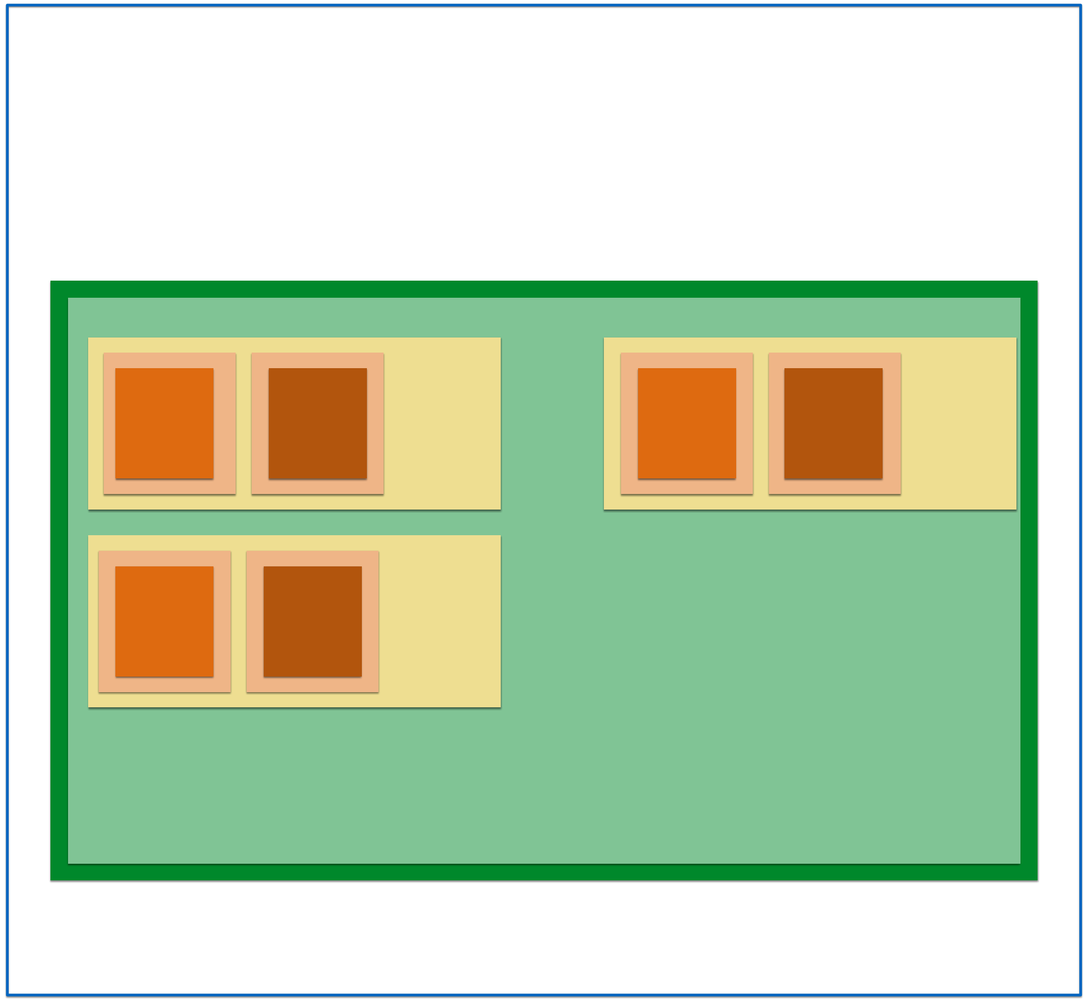
Pod

- A group of 1 or more containers
- Same port space
- Within a Pod: communication over localhost
- Every Pod has it's own IP
- All Pods can talk with each other
- IPs change all the time



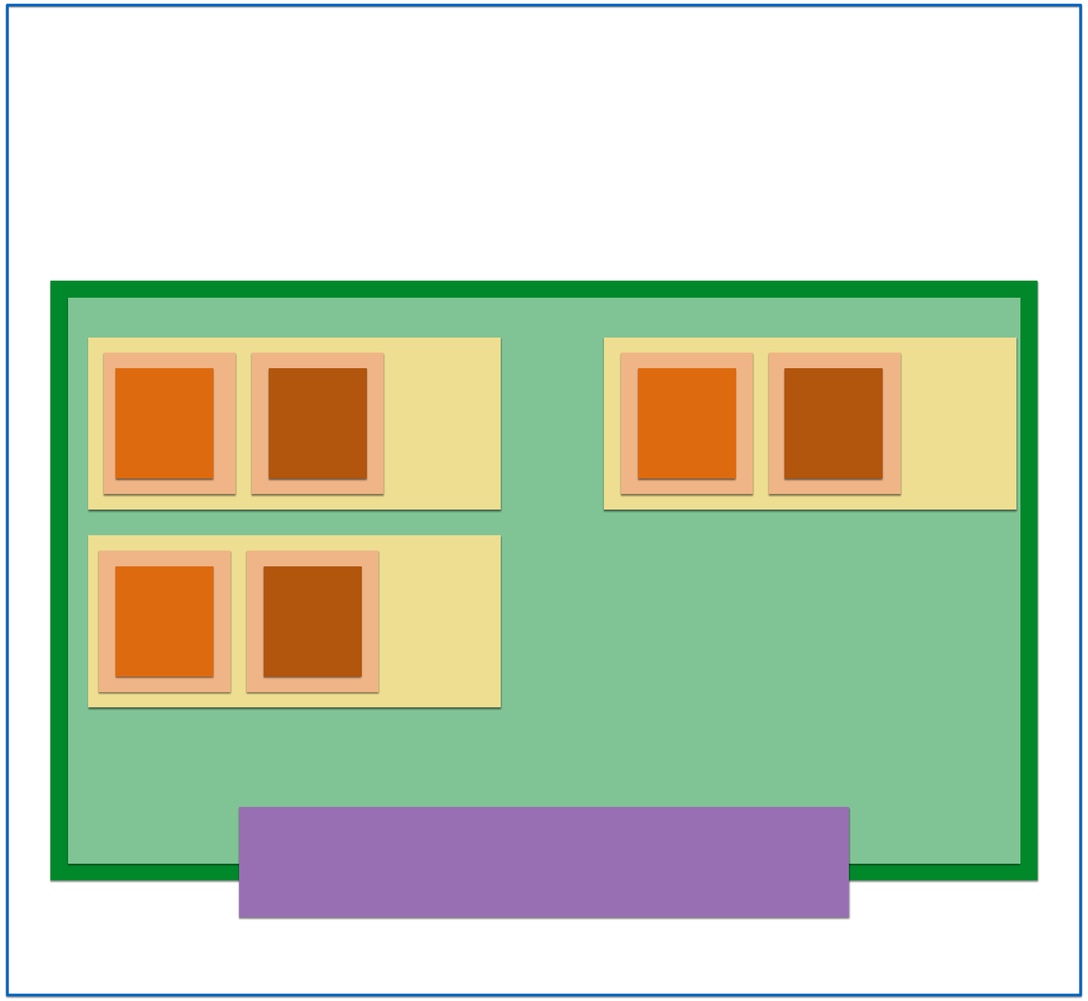
Replica Set

- Defines and manages how many instances of a pod should run
- ReplicaSet is tied to a specific definition of a Pod which is tied to specific image versions of the container
- Image versions in ReplicaSets can't be updated



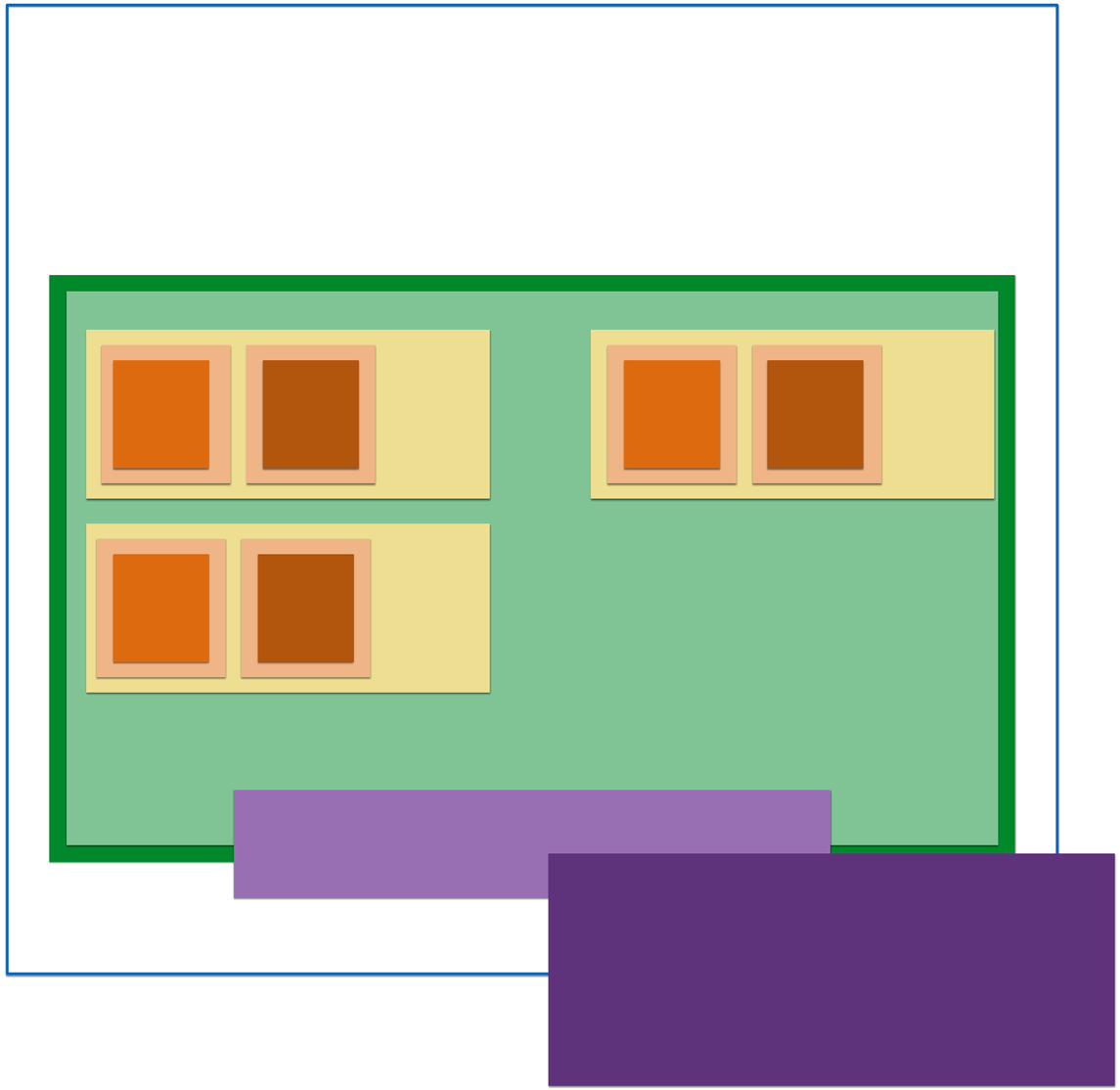
Deployment

- Manages updates and rollbacks of replica sets



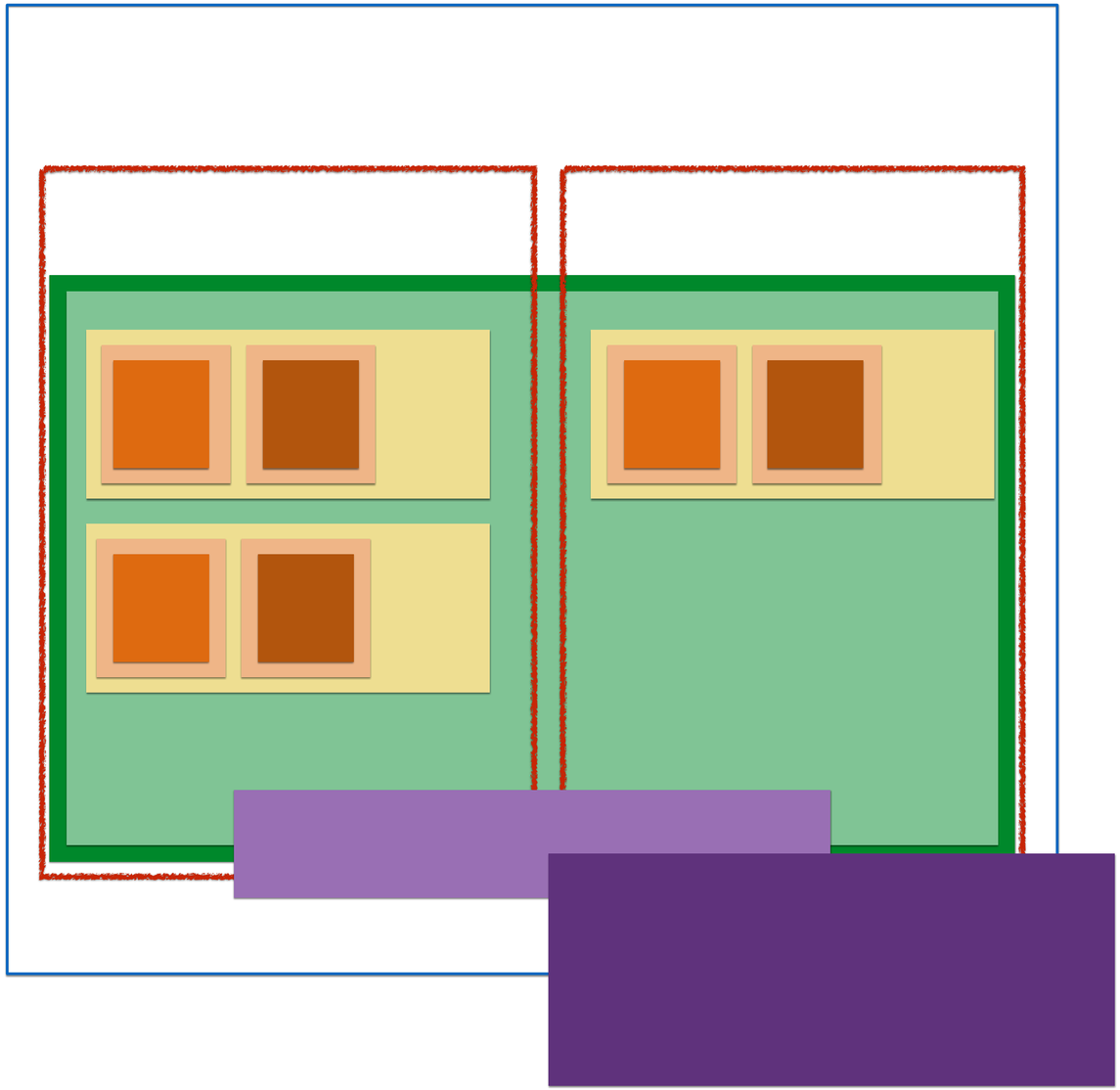
Service

- Internal LoadBalancer
- Makes all pods matching a set of labels accessible through a stable, internal IP address
- You can attach external IP address through an cloud LoadBalancer



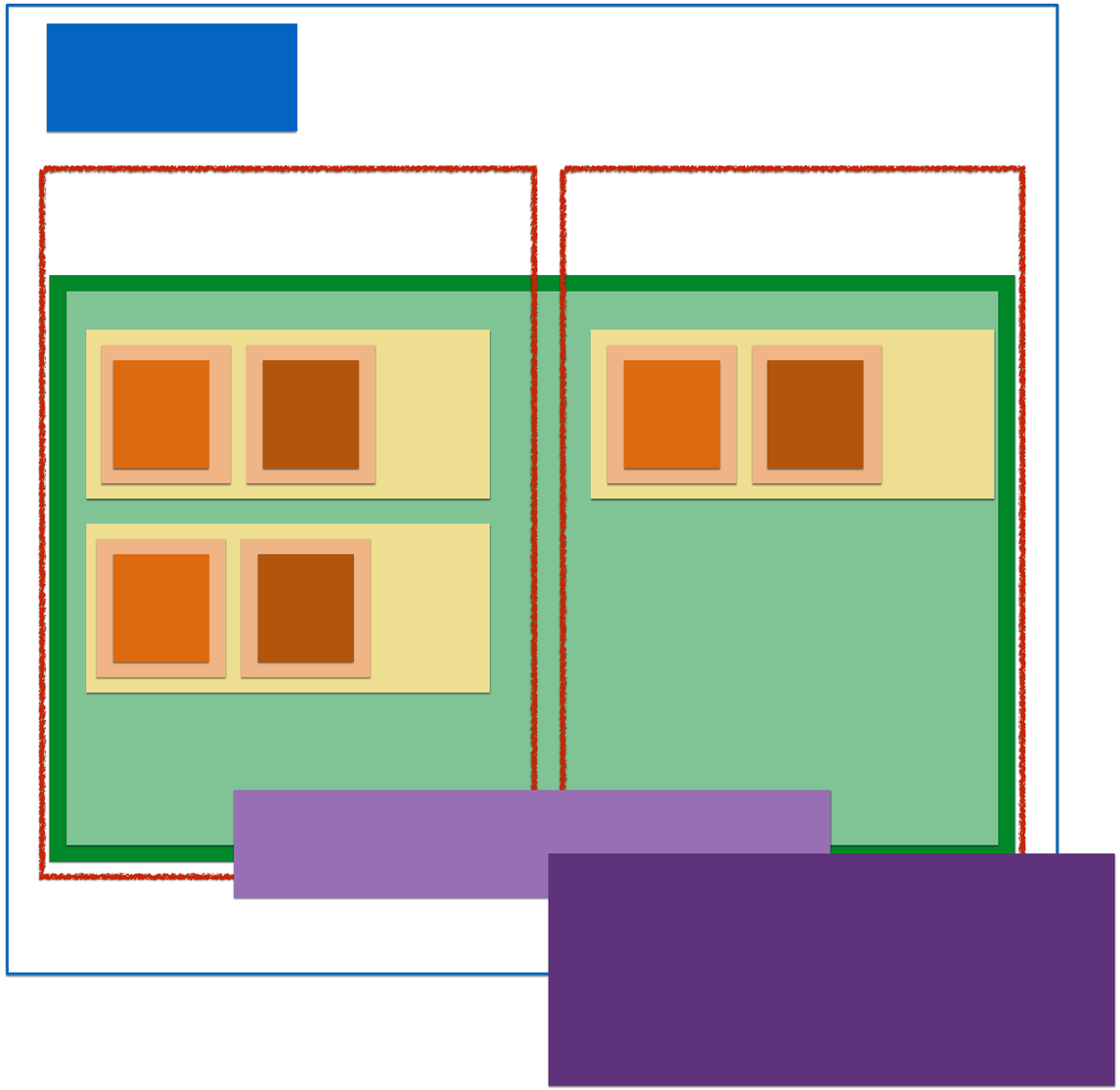
Ingress

- Makes a service accessible to the outside of Kubernetes through an ingress controller (e.g. nginx)
- Traffic is routed by routing rules, usually Host header



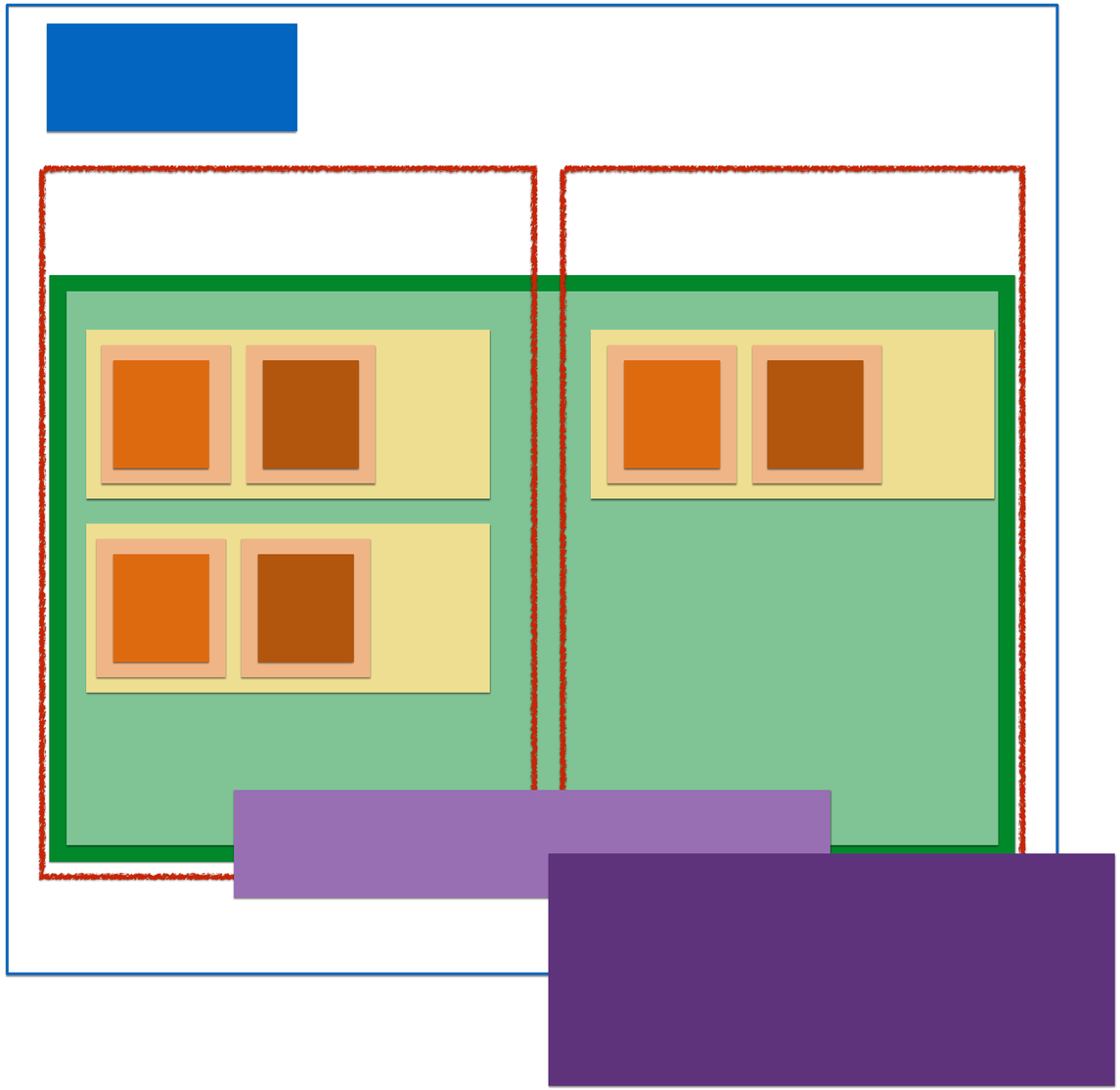
Node

- A physical server
- Containers get distributed automatically



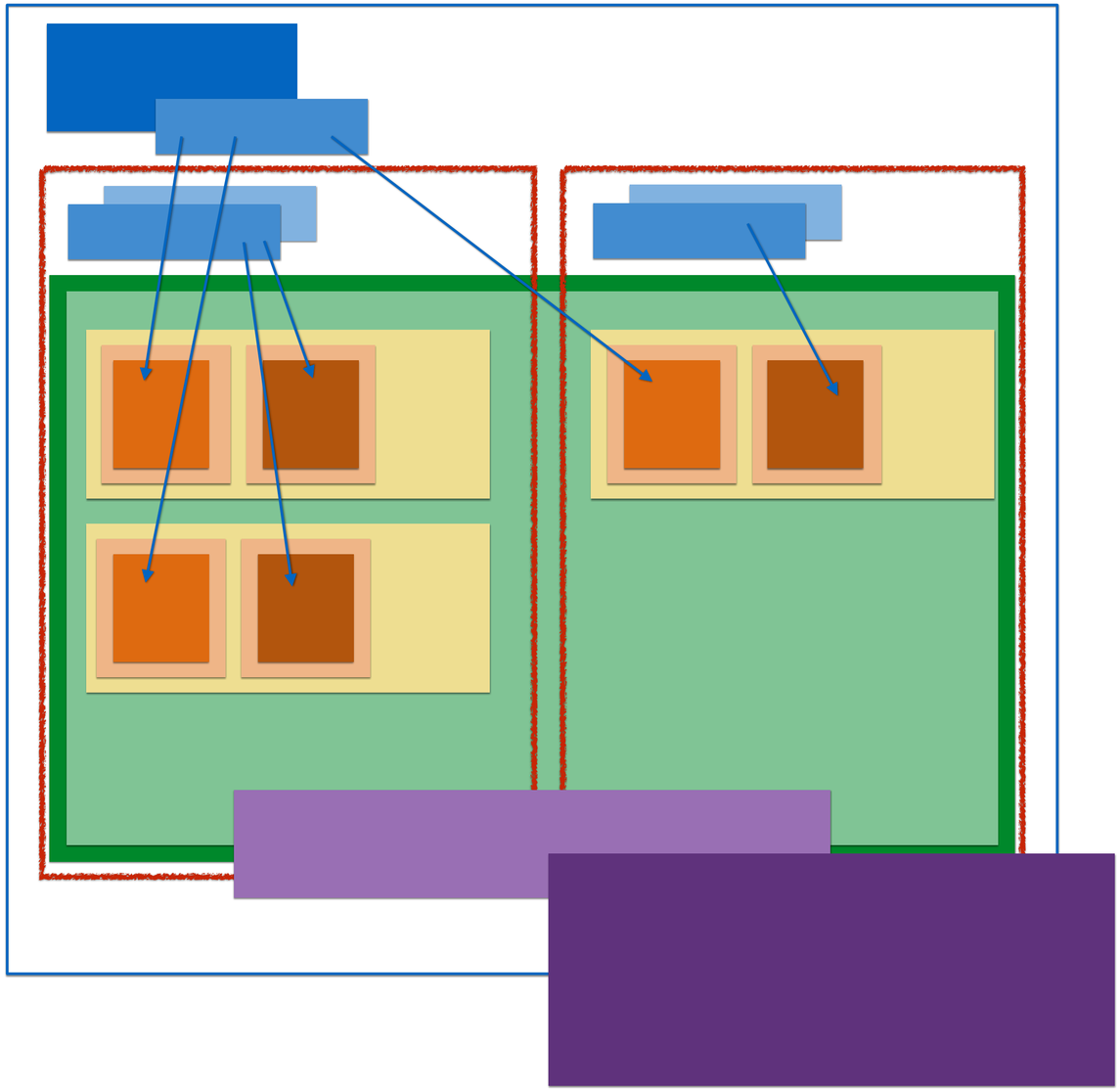
ConfigMap

- Key/Value storage for configuration



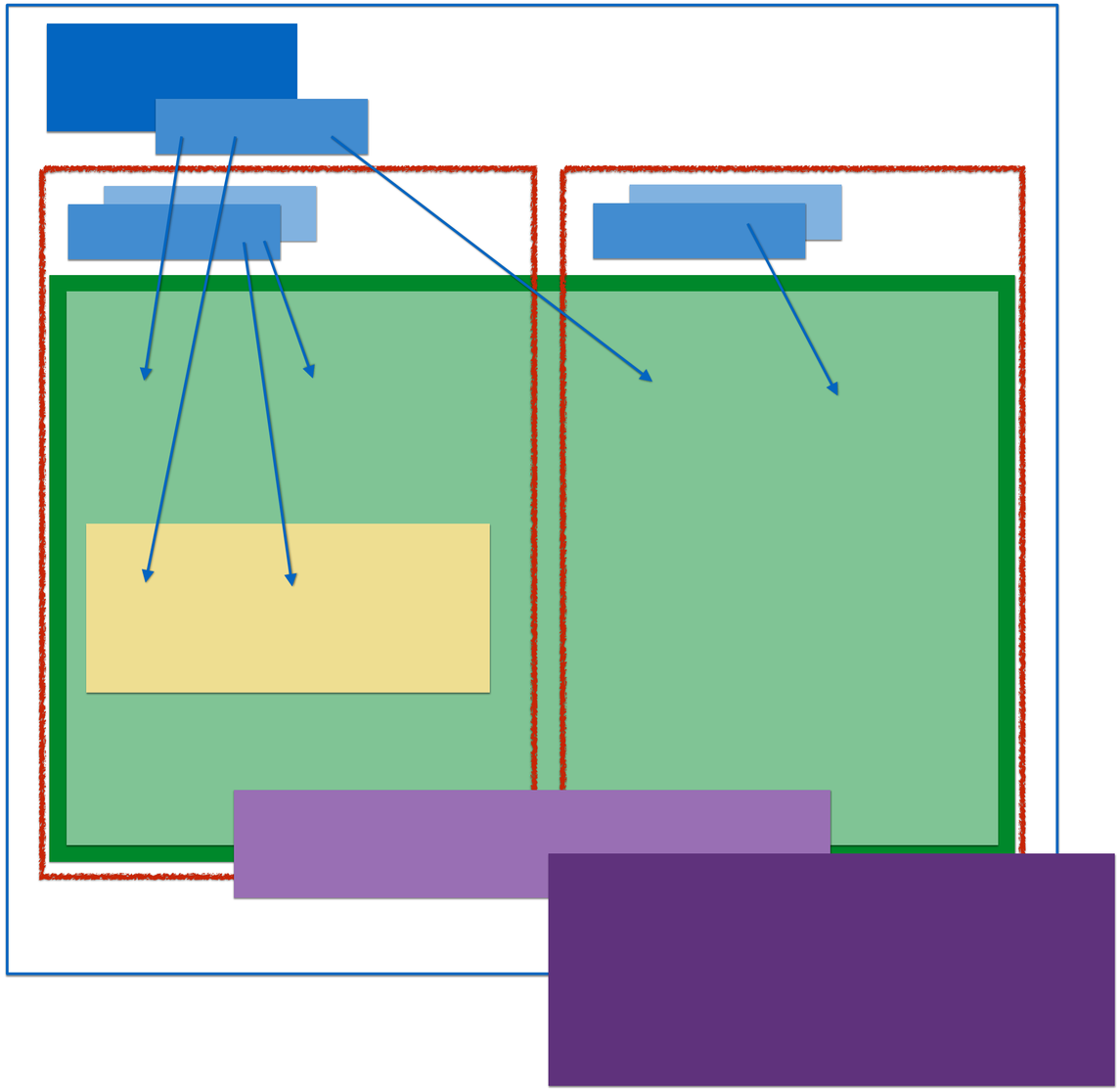
Secret

- Key/Value storage for configuration, usually passwords.



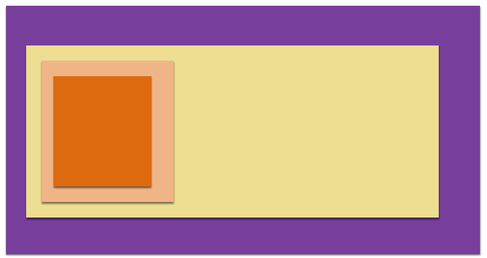
Volumes

- Volumes can be mounted into a container to access a ConfigMap, Secret, persistent volumes with network storage or a folder on the node



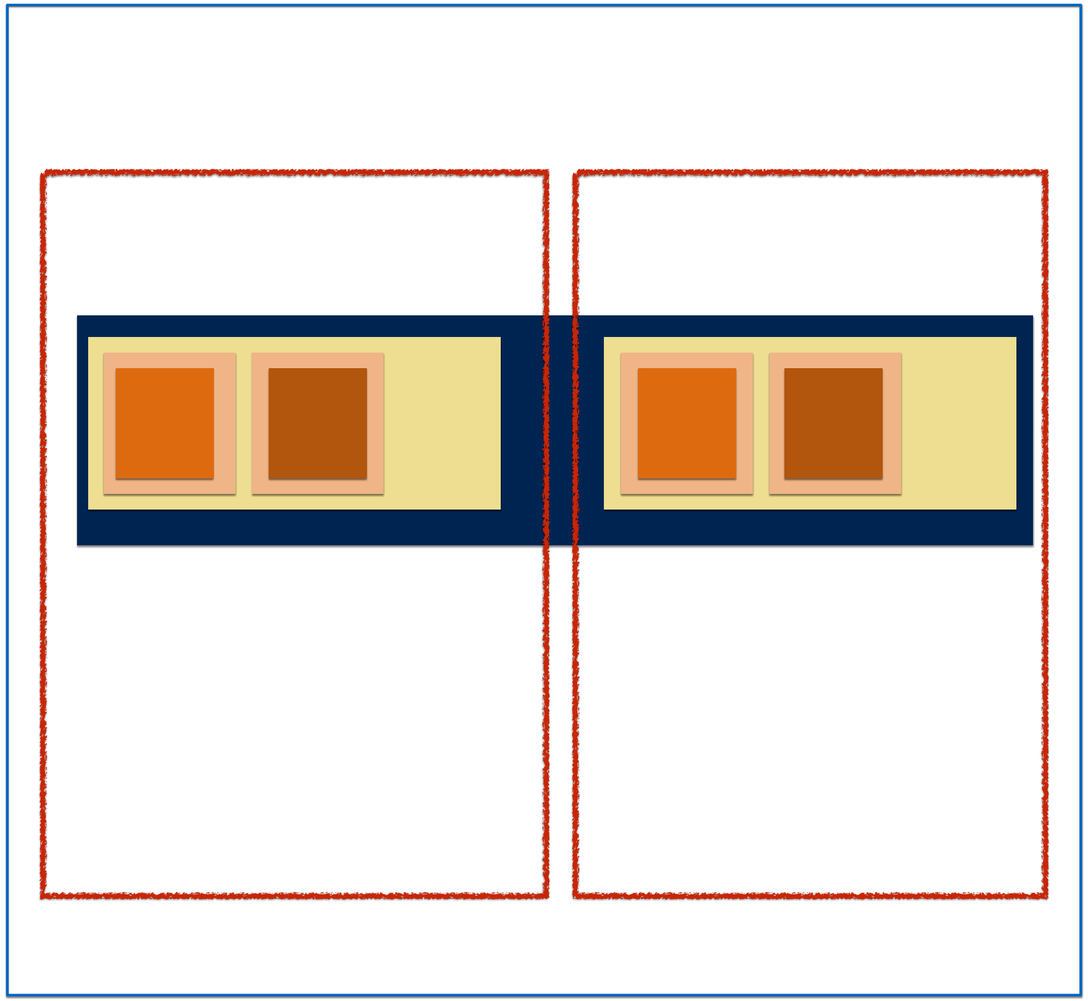
Namespaces

- Dedicated environment to deploy services in



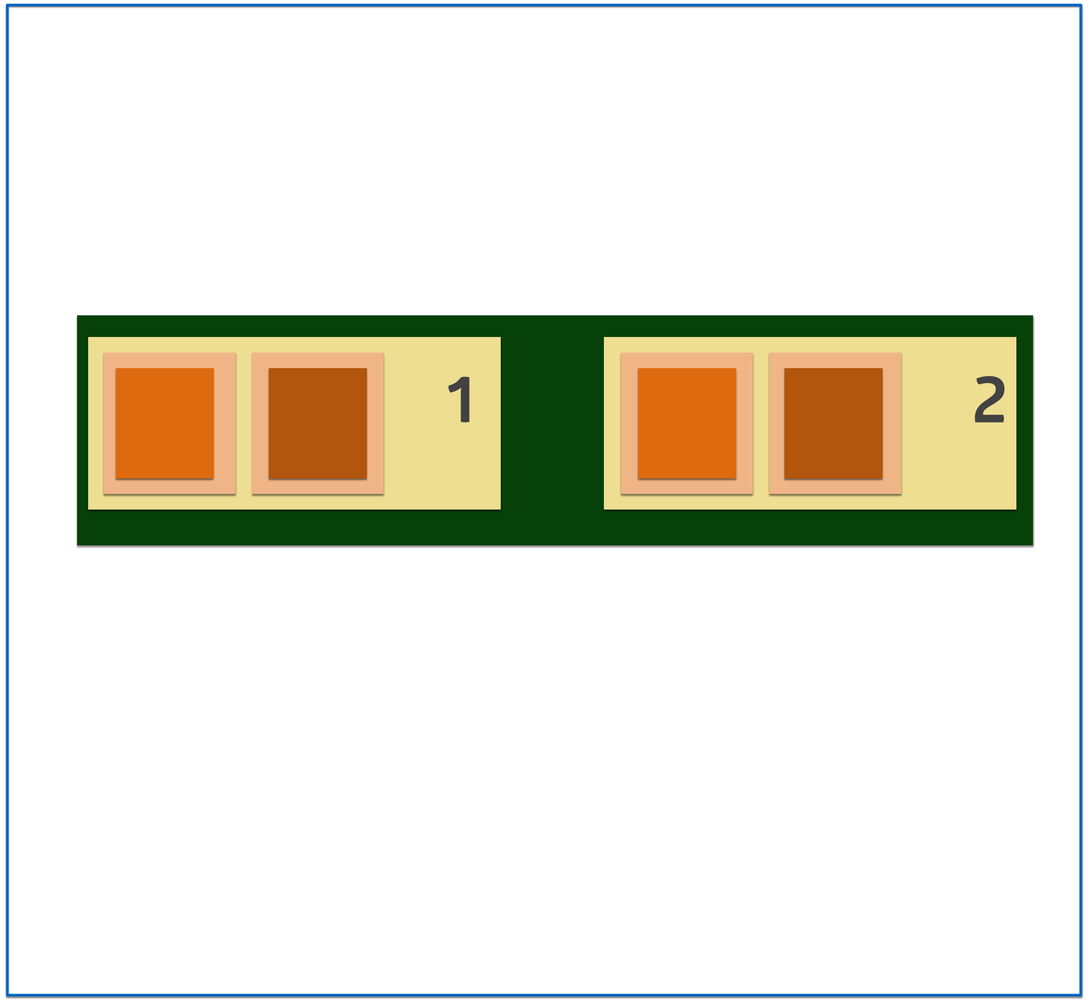
CronJob

- Includes a Pod that is started in a regular interval
- Process in the container should finish at some point



DaemonSet

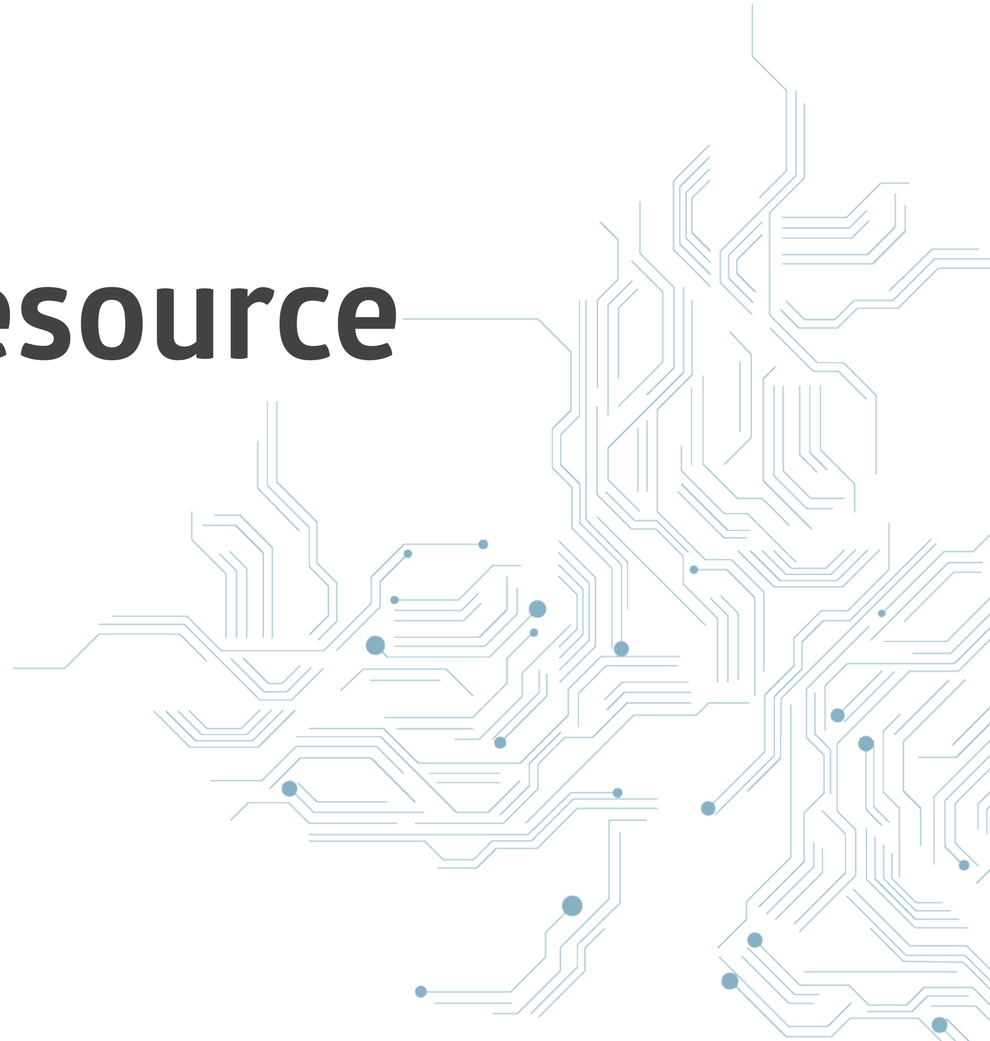
- Defines Pod that should run once on every Node
- Useful for monitoring or logging daemons



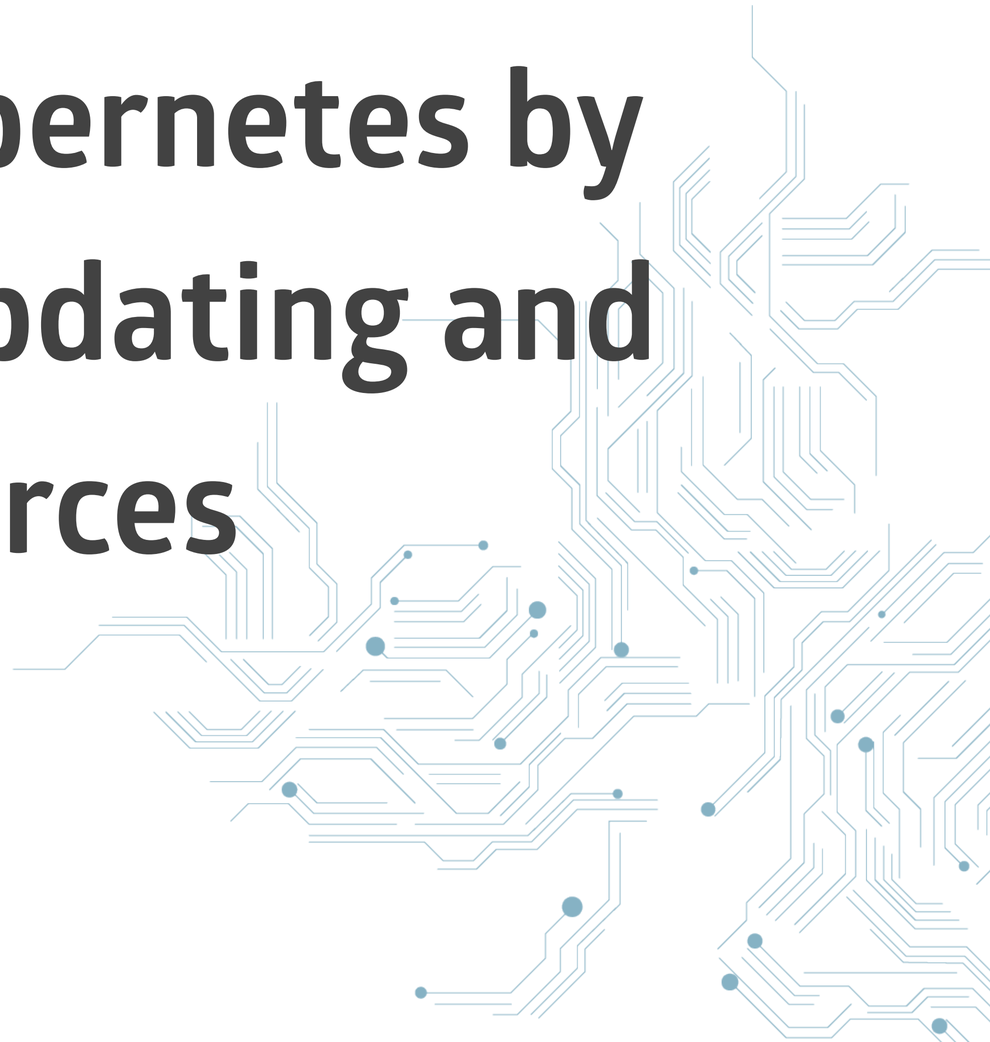
StatefulSet

- Ensures that Pods are started and run in a specific order
- Each Pod of a StatefulSet can have its own persistent volume
- Pod names stay the same

Everything is a resource

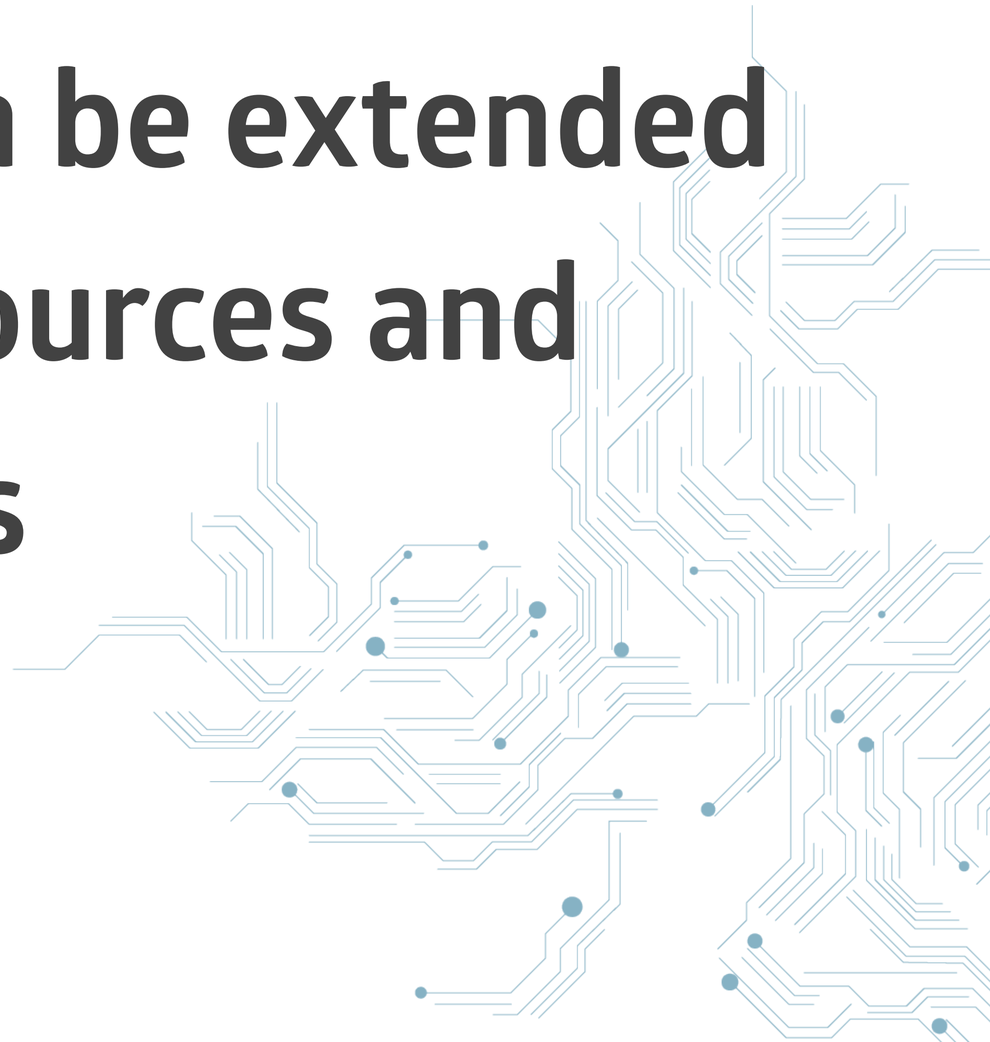


You interact with Kubernetes by creating, receiving, updating and deleting resources



Kubernetes has controllers to listen on these interactions and get the cluster in the desired state

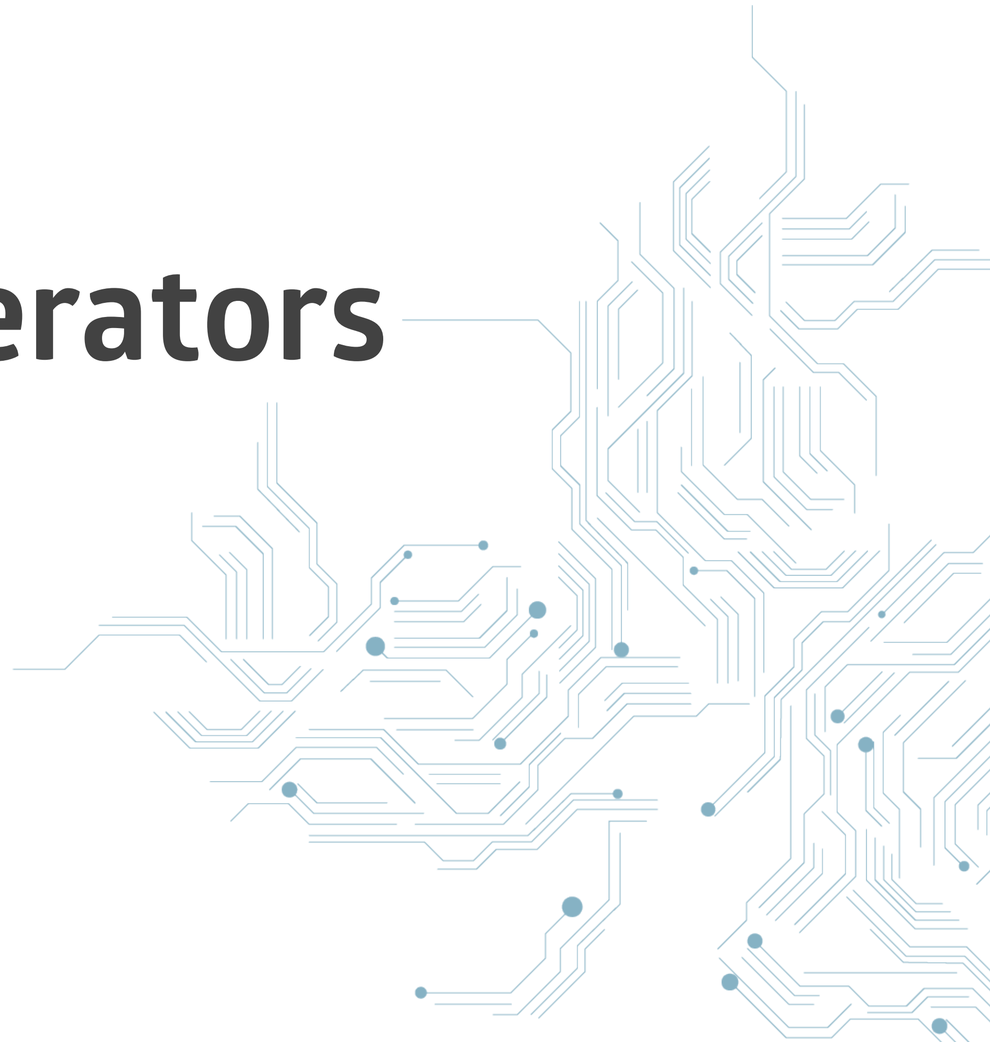
The Kubernetes API can be extended with additional Resources and Controllers



CustomResourceDefinitions

Certificate, Backup, Restore, MySQLCluster, Function, ...

Controllers / Operators



deployment.yaml

```
kind: Deployment
apiVersion: extensions/v1beta1
metadata:
  name: symfony-demo
spec:
 template:
    spec:
      containers:
        - name: symfony-demo
          image: symfony-demo:1.1.0
          ports:
```

bash

```
$ kubectl apply -f deployment.yaml
```

bash

```
$ kubectl get deployments
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
symfony-demo   1         1         1            1           21h
```

bash

```
$ kubectl get deployment symfony-demo -o yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  annotations:
    ...
spec:
 ...
 template:
     ...
    spec:
```

bash

```
$ kubectl delete deployment symfony-demo
```

Tooling

kubectl

REST API

bash

```
$ kubectl proxy --port=8080
$ curl http://localhost:8080/api/v1/namespaces/default/pods
{
  "kind": "PodList",
  "apiVersion": "v1",
  "metadata": {
    "selfLink": "/api/v1/namespaces/default/pods",
    "resourceVersion": "336834"
  },
  "items": [
    {
```

Helm

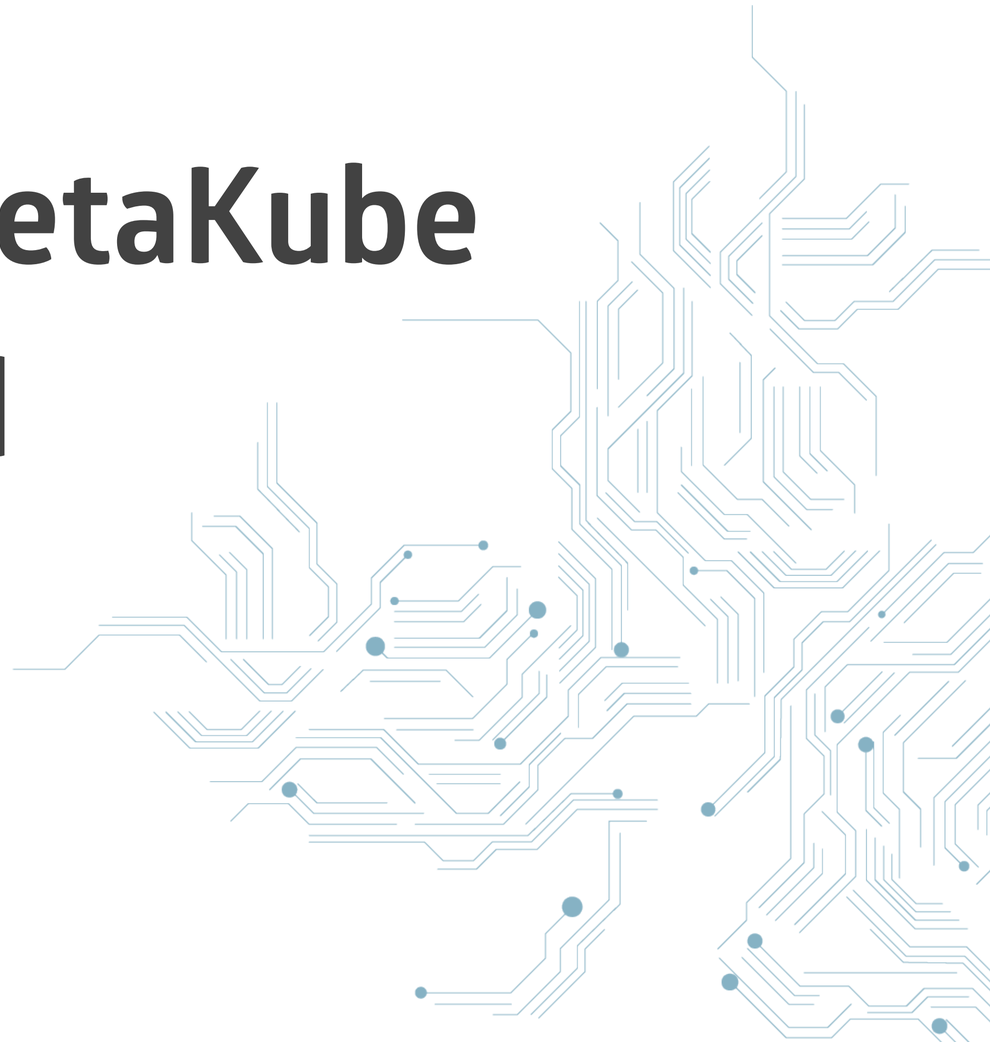
The package manager for Kubernetes

bash

```
$ helm install stable/wordpress
```

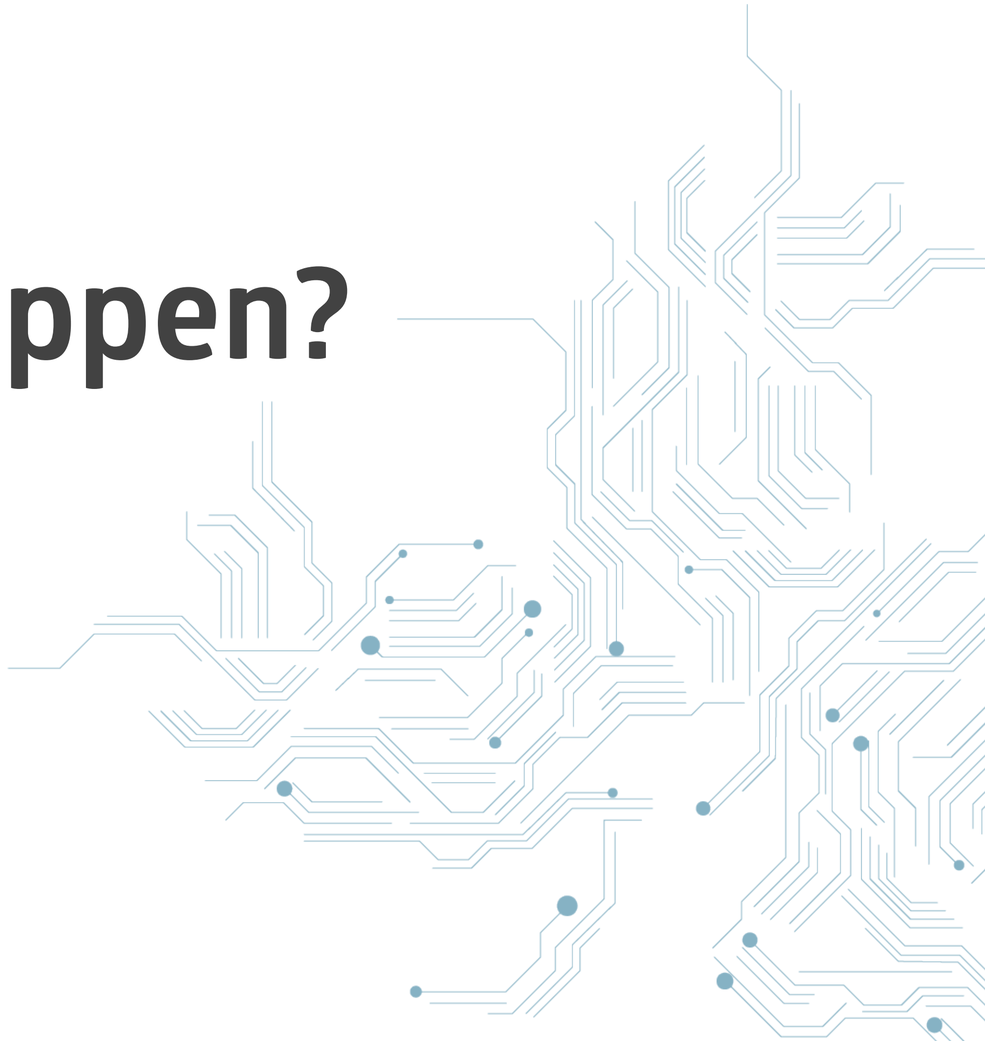
hands-on

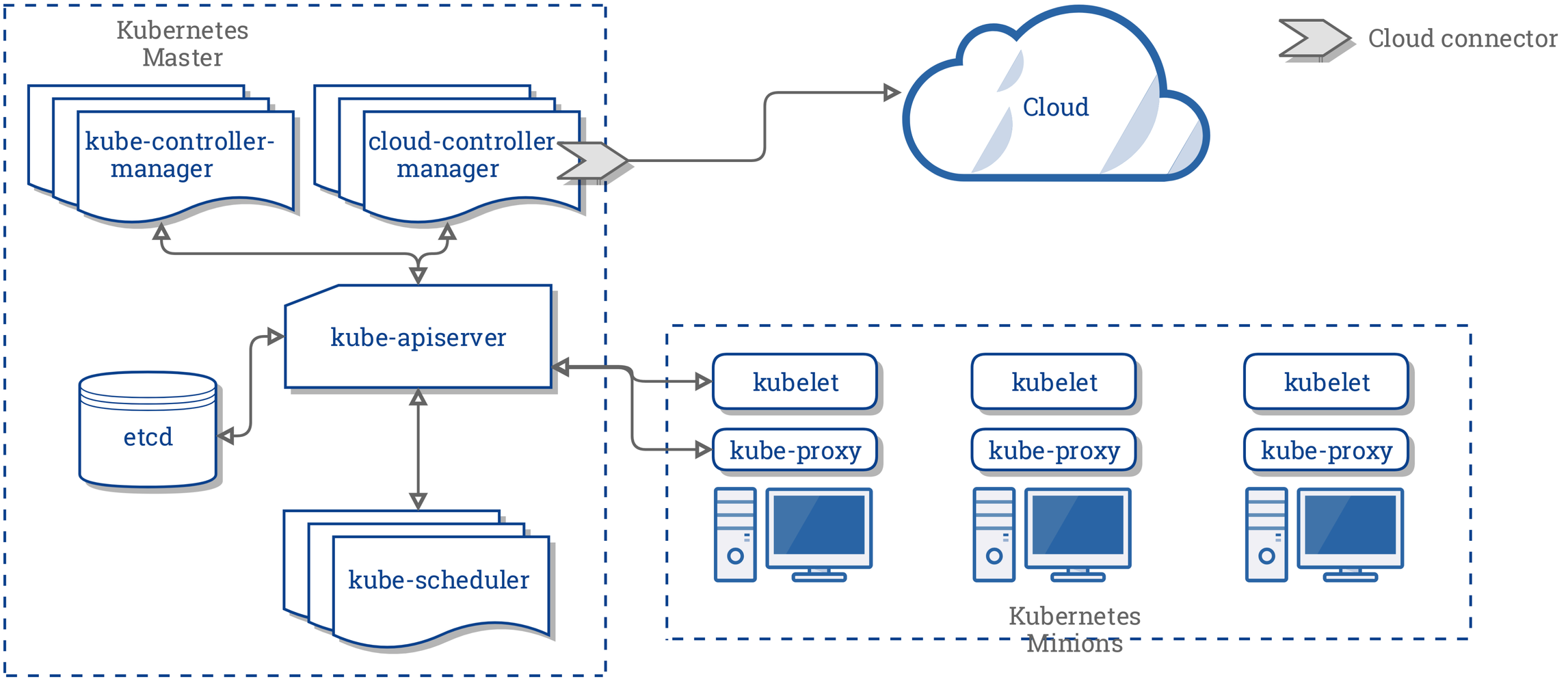
Let’s look at the MetaKube dashboard

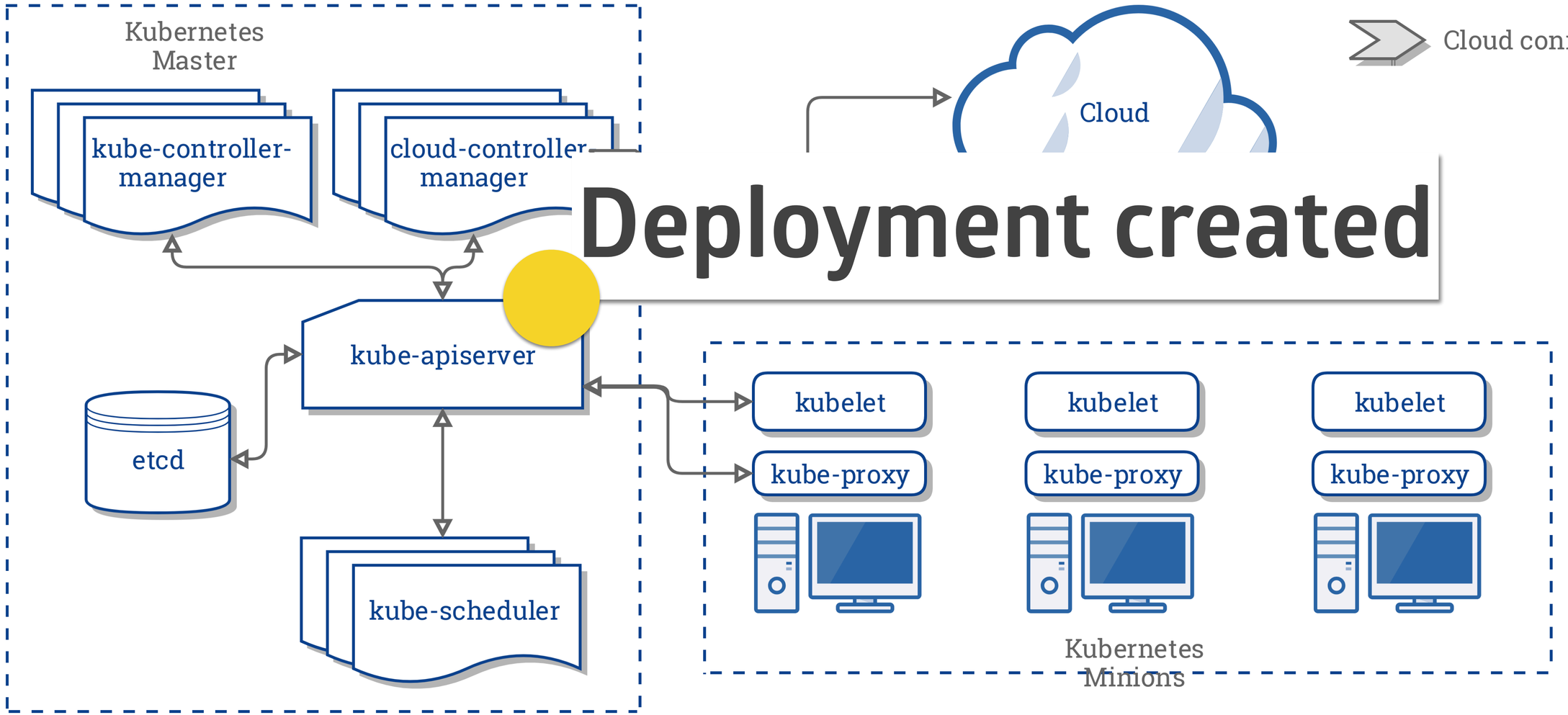


1 - Deploying a simple Web Application

What did just happen?

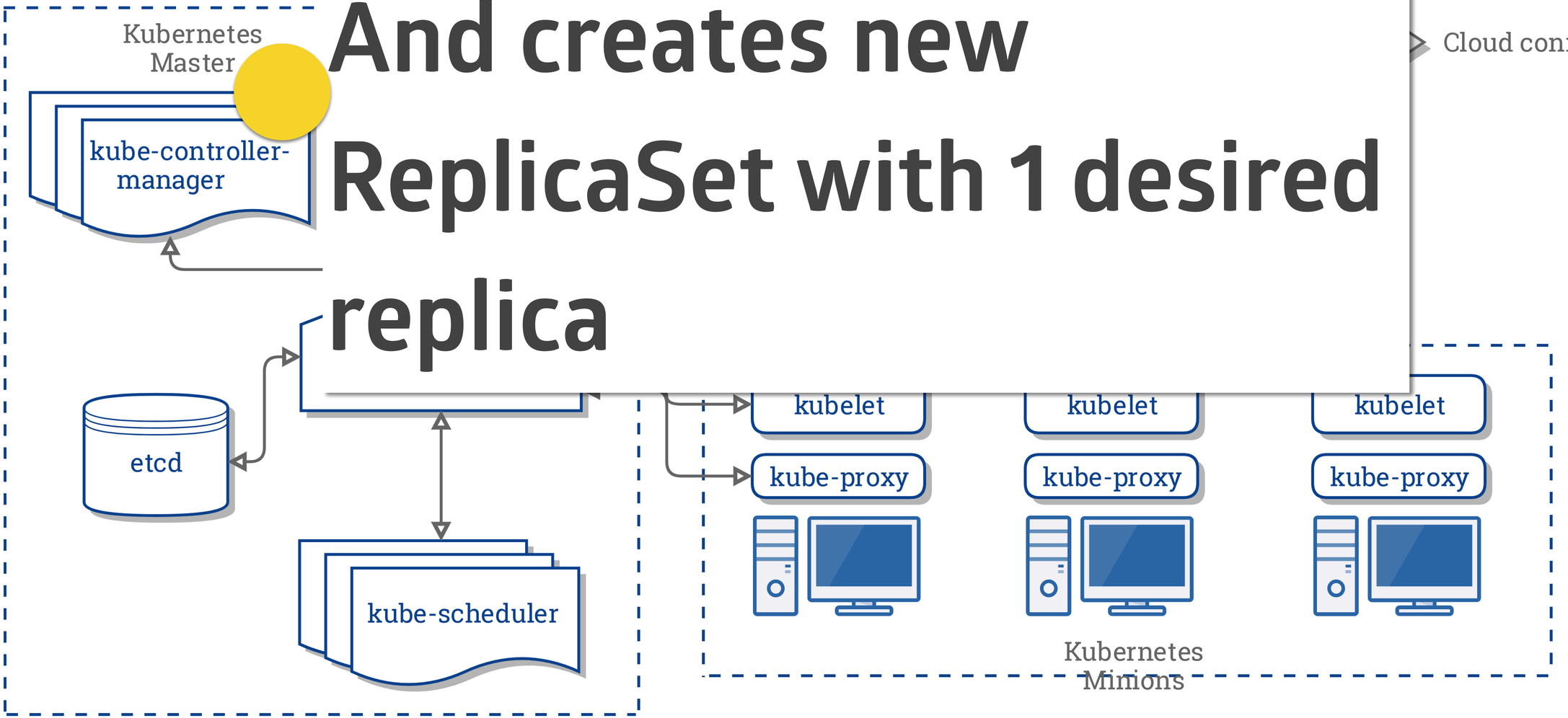



Kubernetes
Master
kube-controller-
manager
cloud-controller
manager
Cloud
Cloud connector
kube-apiserver
etcd
kube-scheduler
kubelet
kube-proxy
kubelet
kube-proxy
kubelet
kube-proxy
Kubernetes
Minions

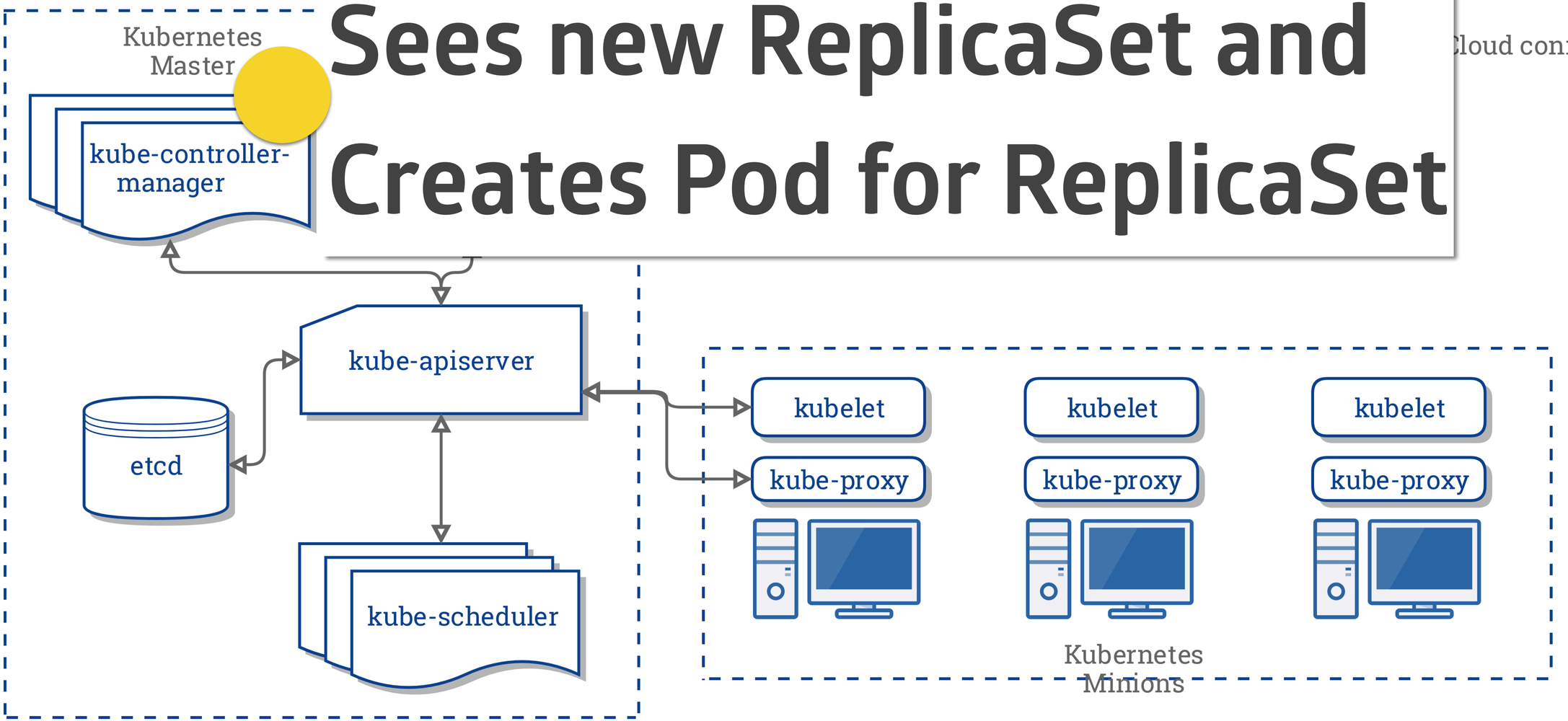

Kubernetes
Master
kube-controller-
manager
cloud-controller-
manager
Cloud
Cloud connector
Deployment created
kube-apiserver
etcd
kube-scheduler
kubelet
kubelet
kubelet
kube-proxy
kube-proxy
kube-proxy
Kubernetes
Minions

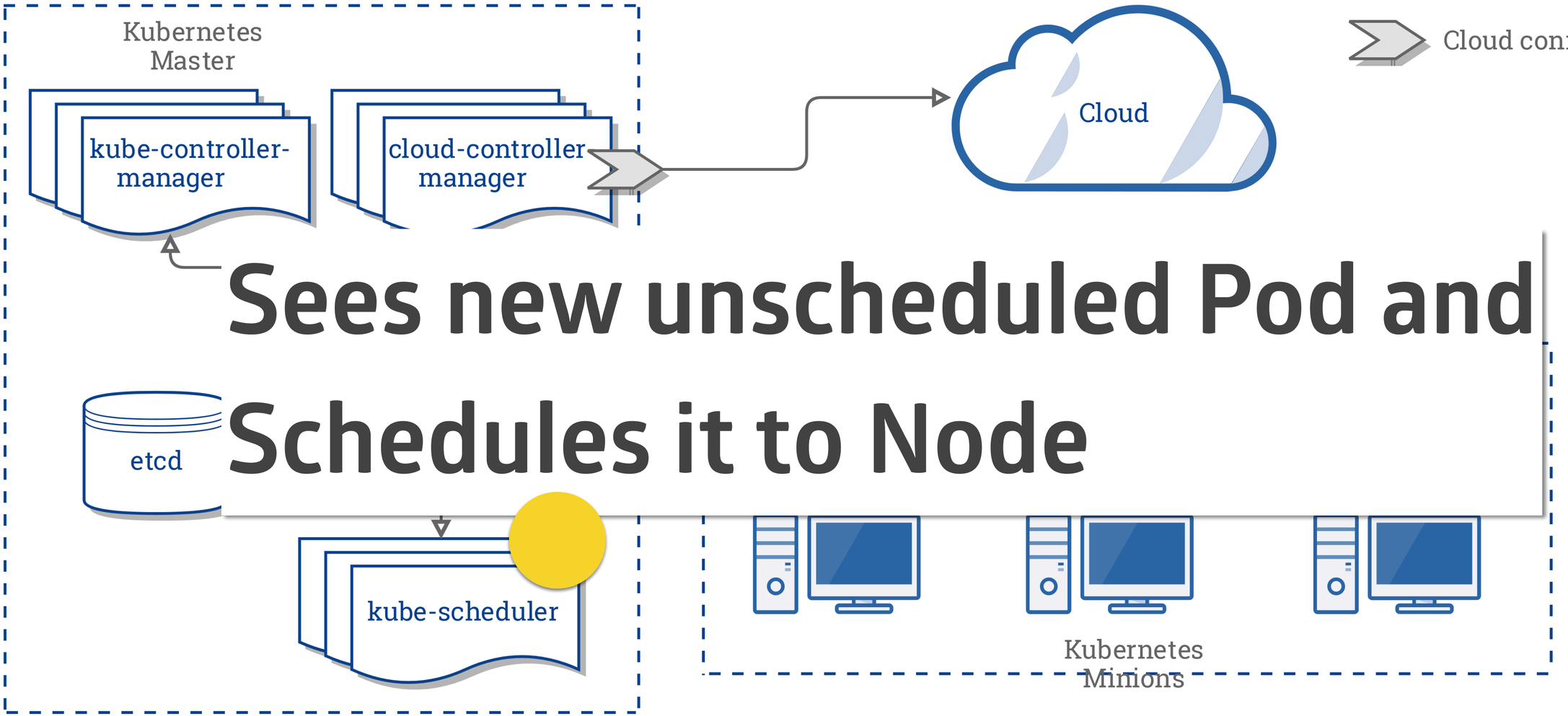
Sees new Deployment

And creates new ReplicaSet with 1 desired replica

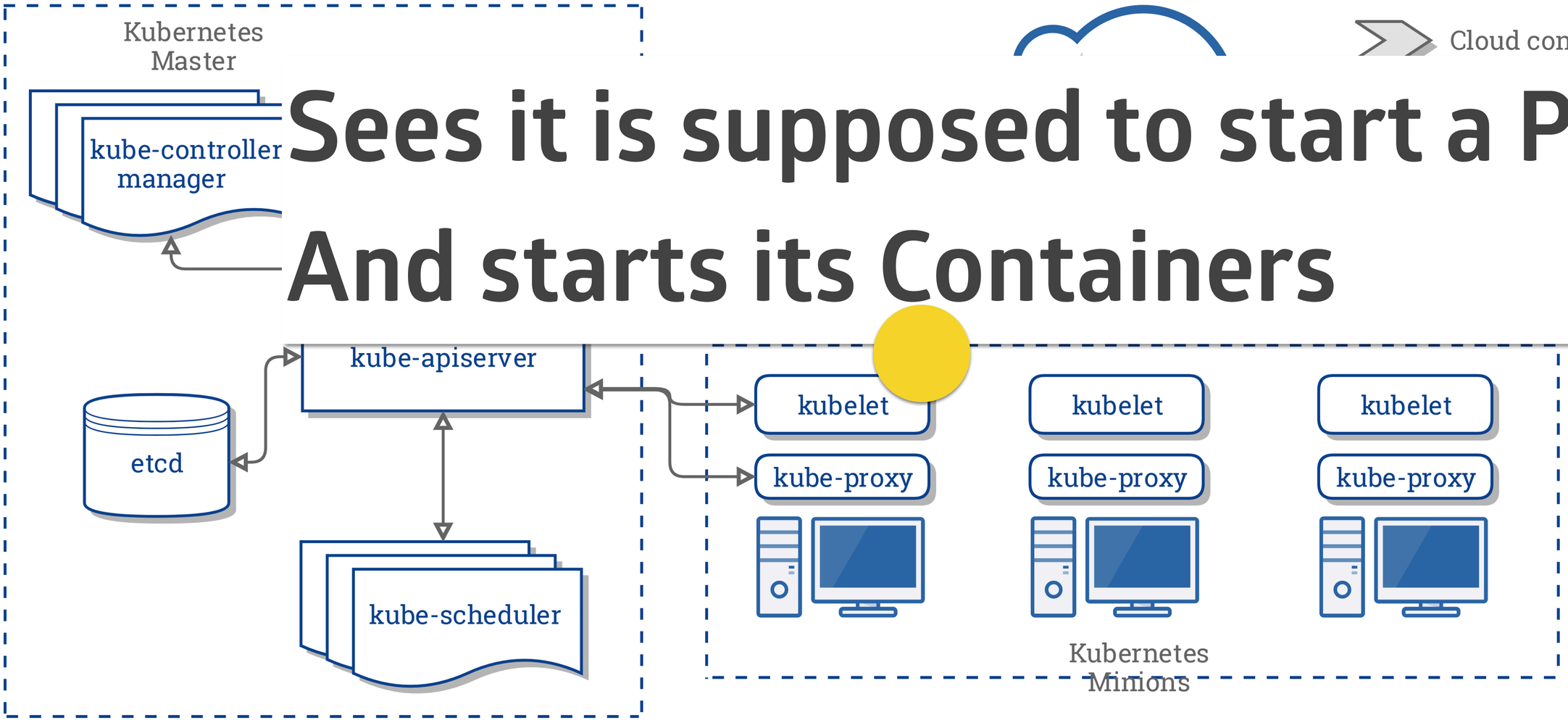


Sees new ReplicaSet and Creates Pod for ReplicaSet



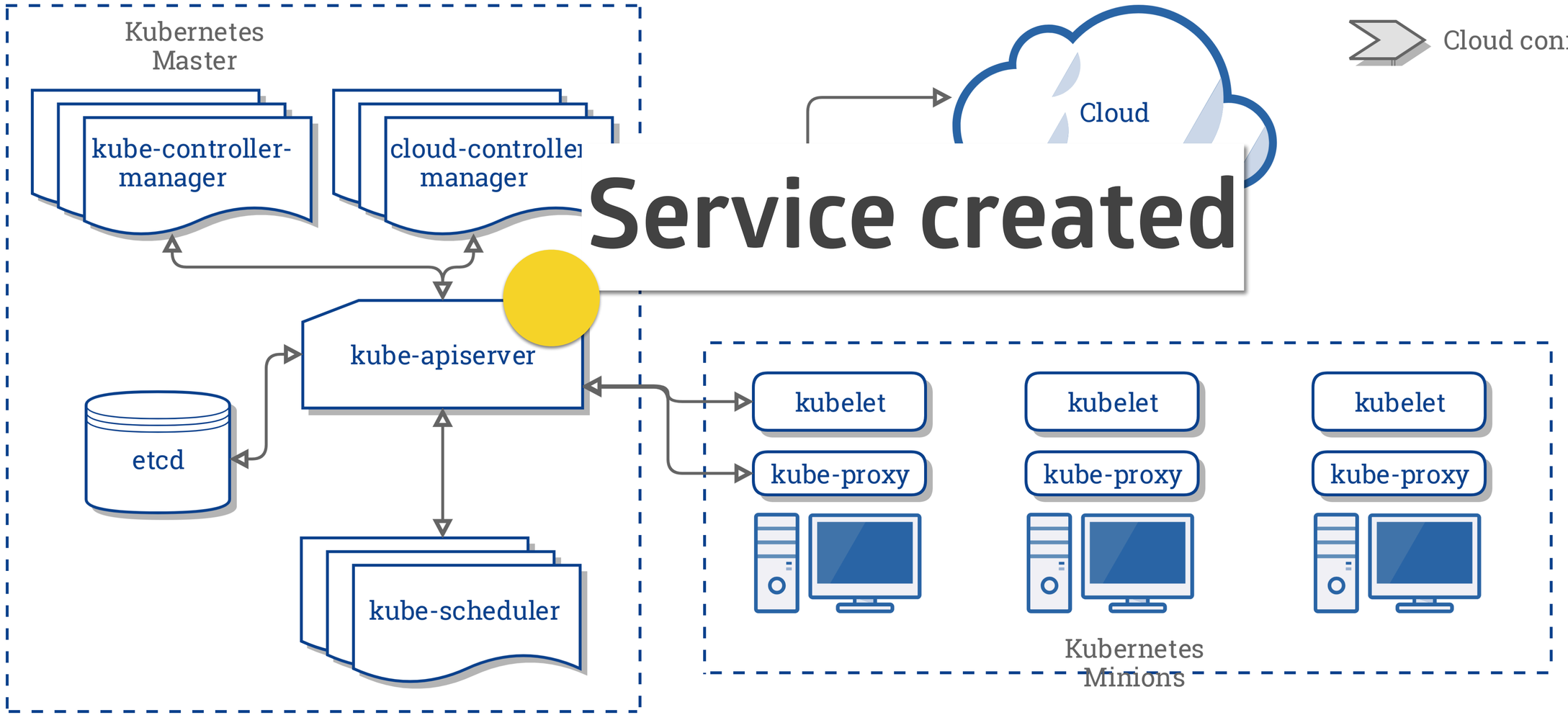


Sees new unscheduled Pod and Schedules it to Node

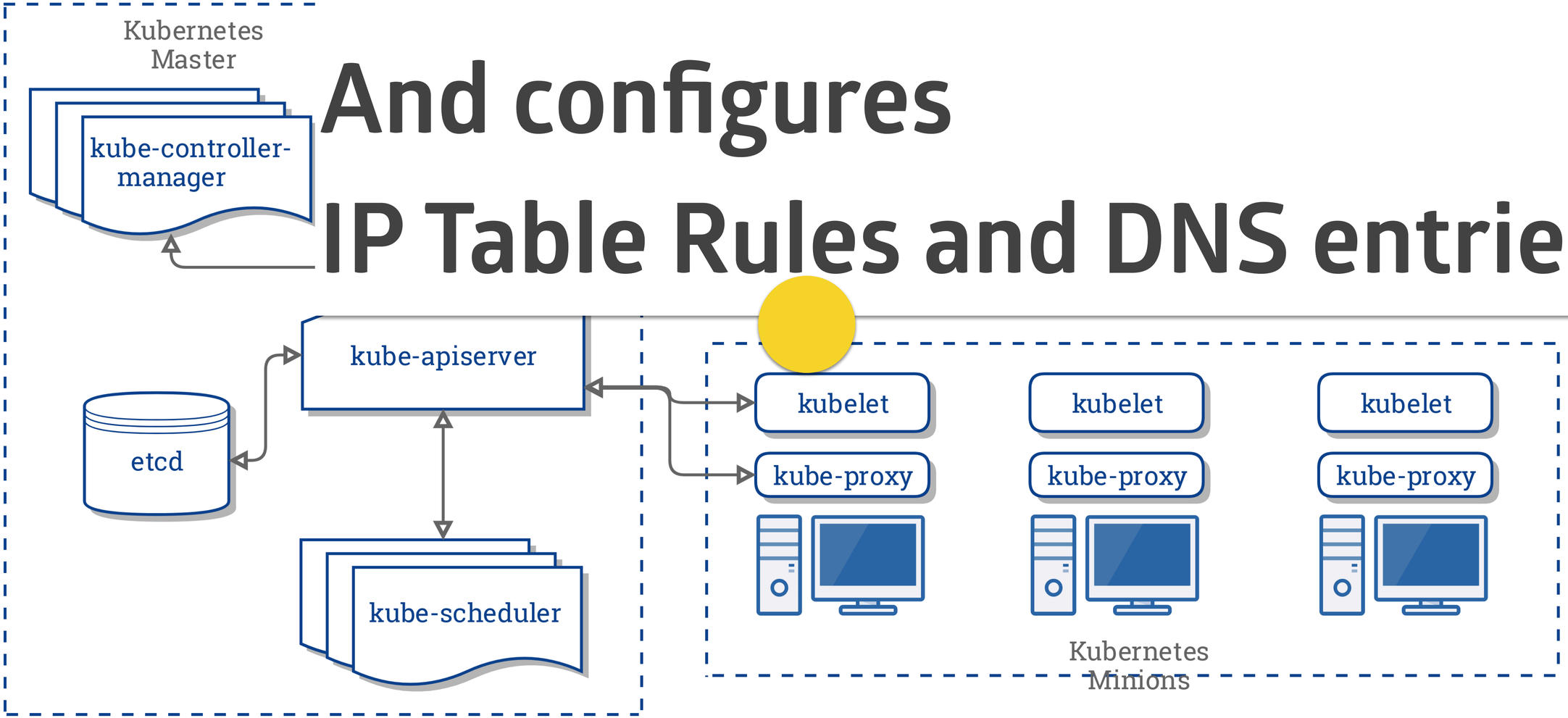


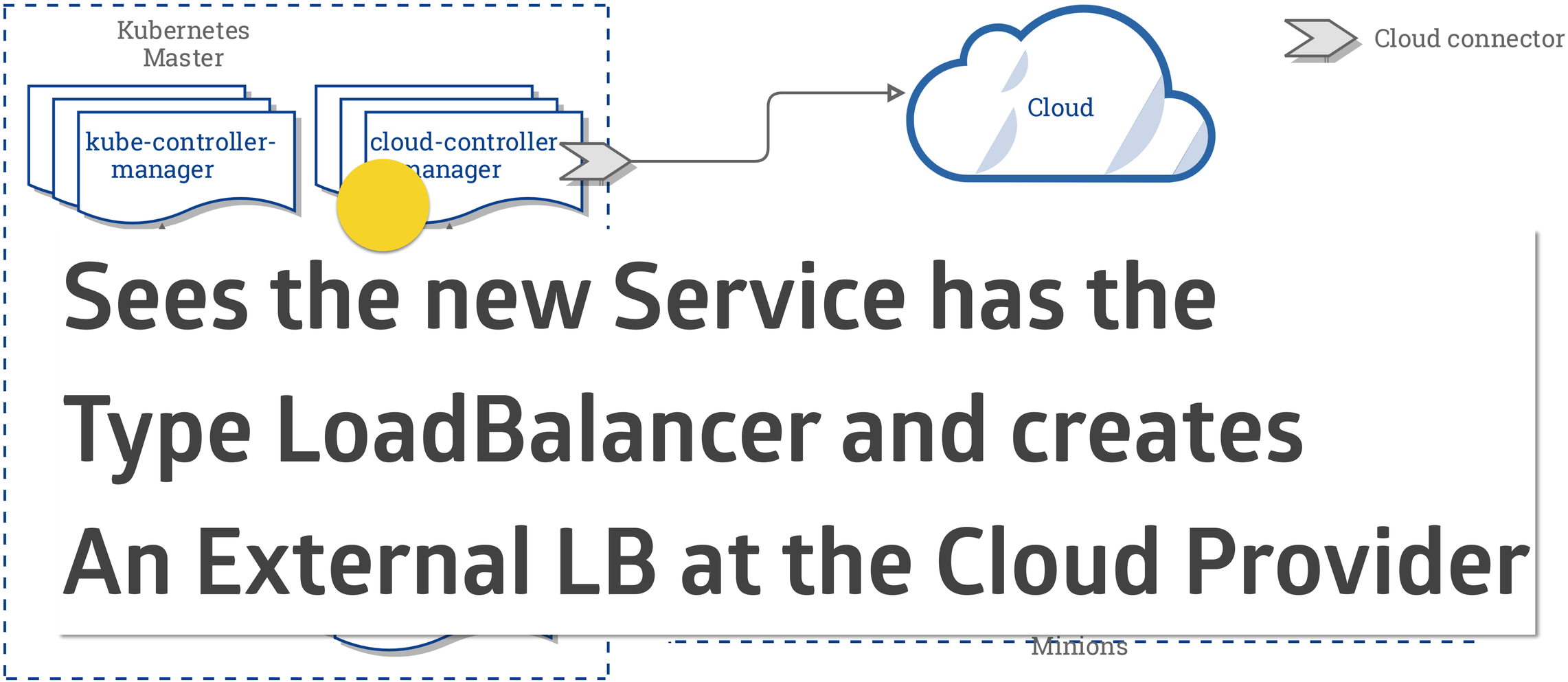
Sees it is supposed to start a Pod

And starts its Containers


Kubernetes
Master
kube-controller-
manager
cloud-controller
manager
Cloud
Service created
Cloud connector
kube-apiserver
etcd
kube-scheduler
kubelet
kube-proxy
kubelet
kube-proxy
kubelet
kube-proxy
Kubernetes
Minions

Sees the new Service
And configures
IP Table Rules and DNS entries

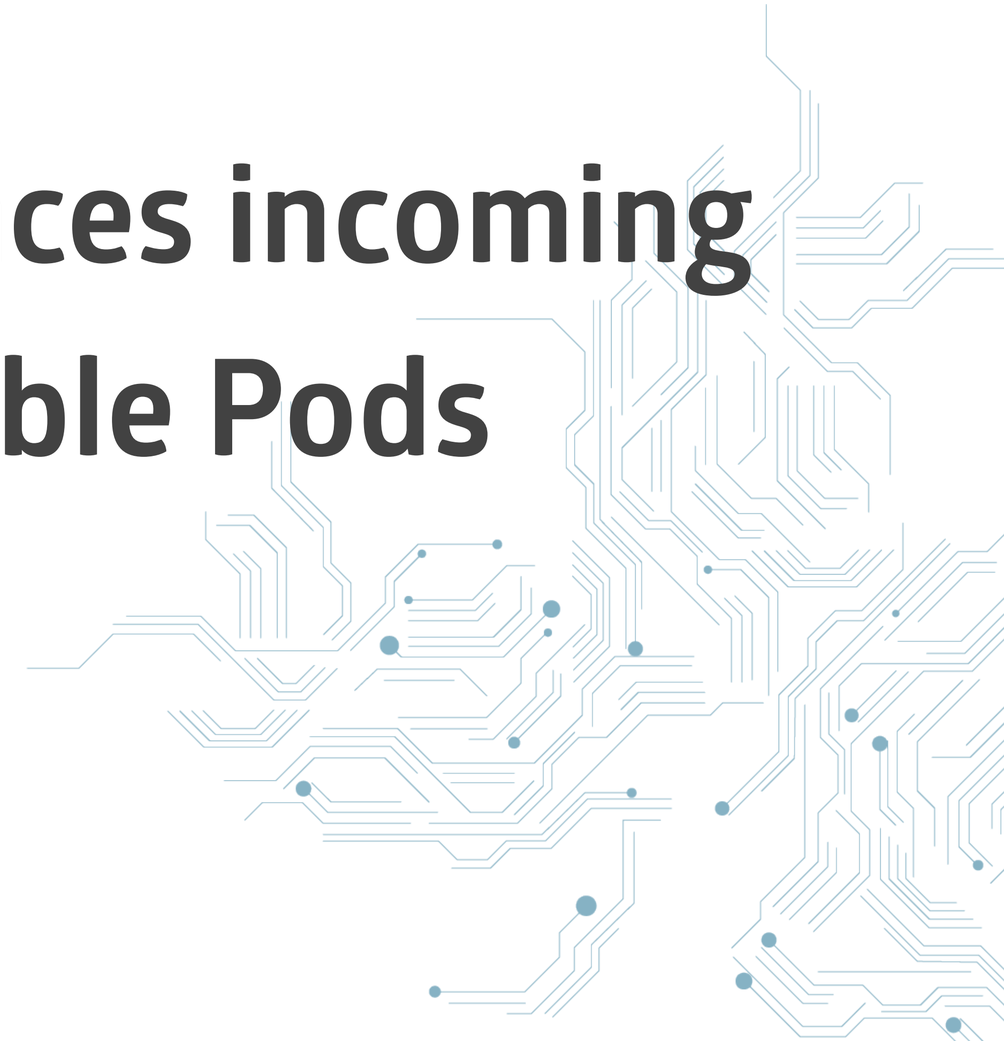




Sees the new Service has the Type LoadBalancer and creates An External LB at the Cloud Provider

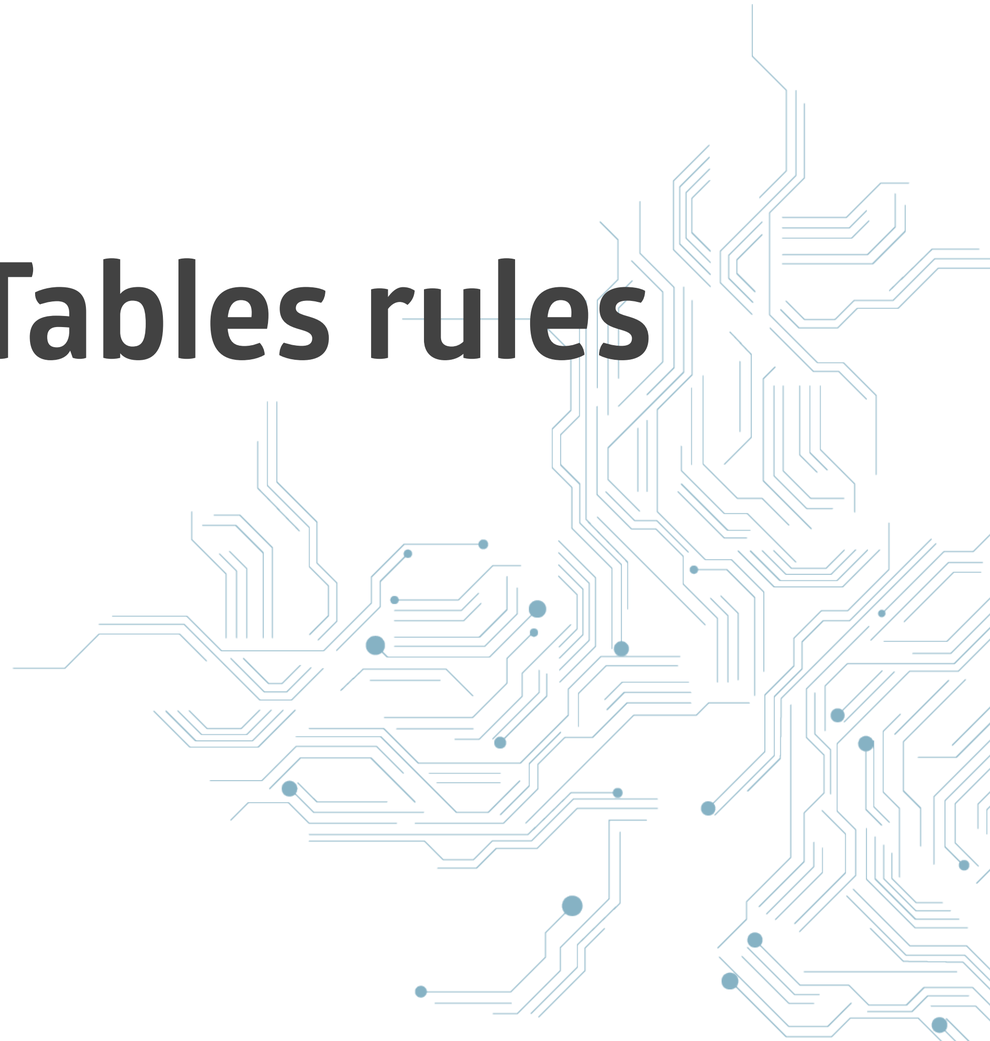
How is traffic routed to the Pod

The Service load-balances incoming traffic to all available Pods

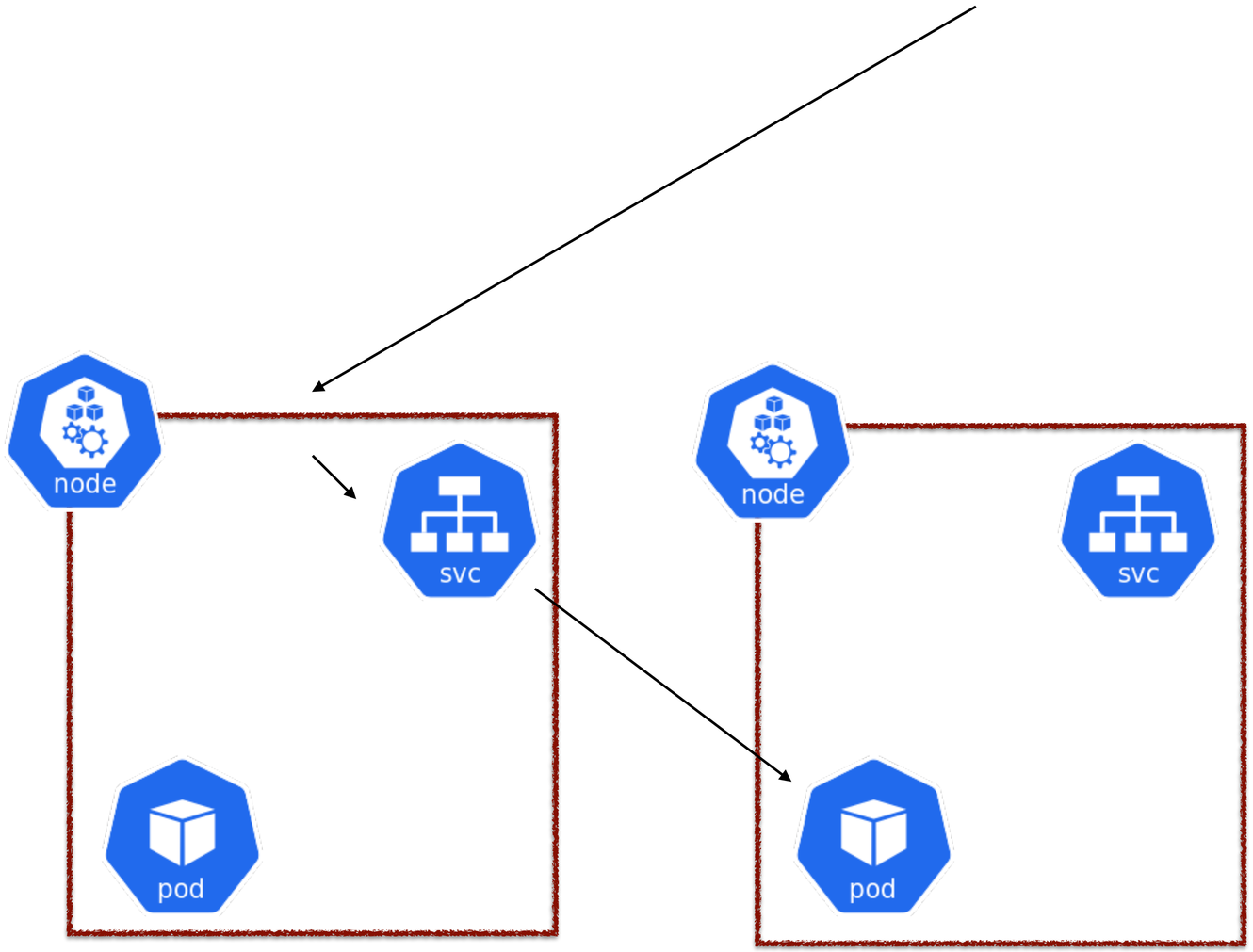


Every Service has a virtual IP

Round Robin with IP Tables rules



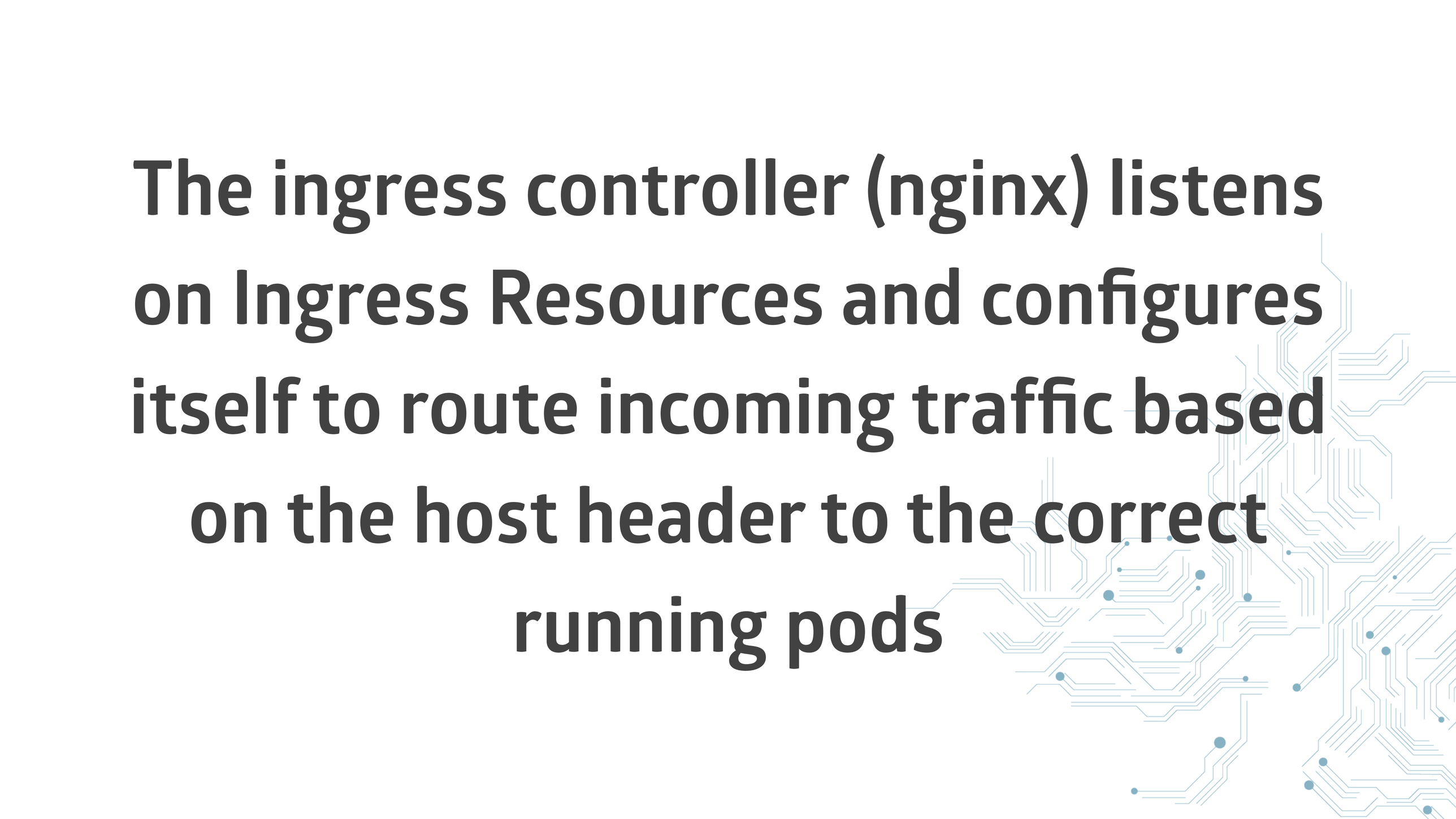
OpenStack LoadBalancer



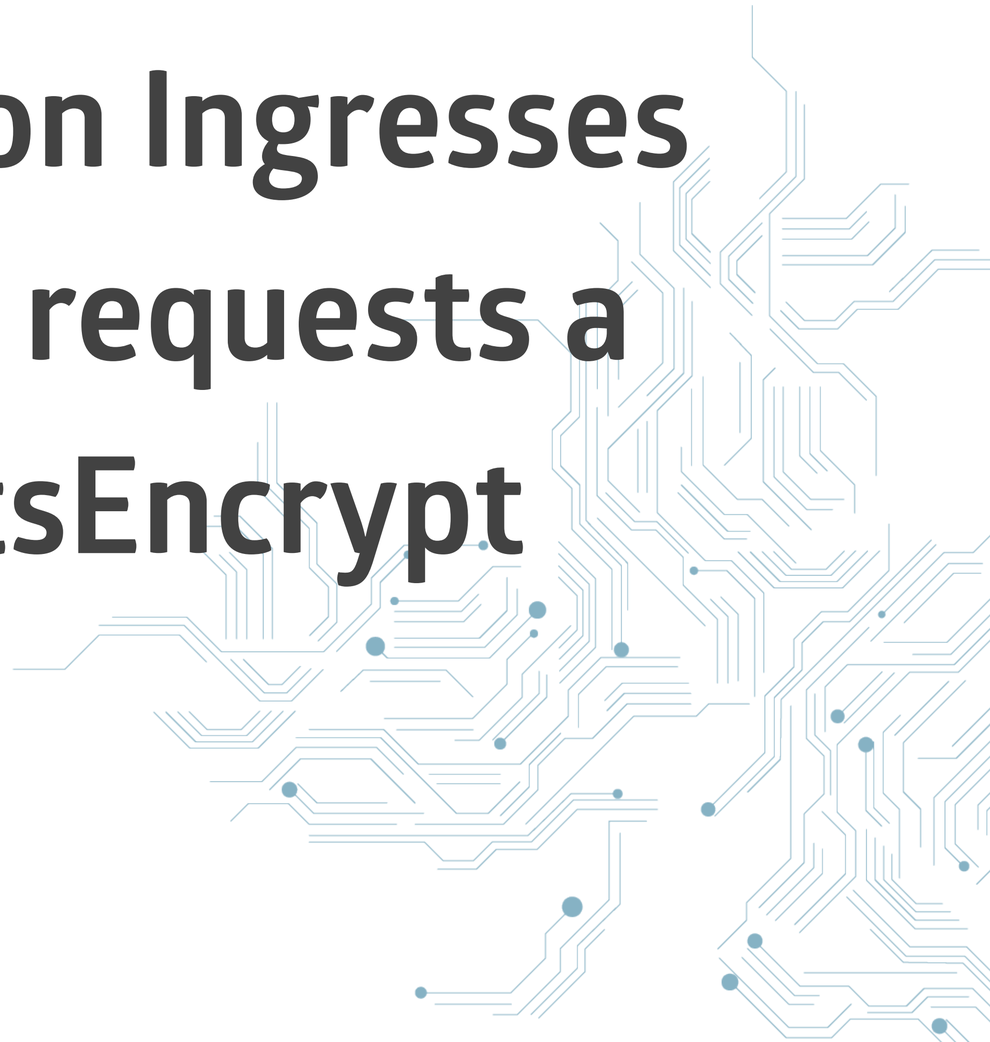
7 - Operators

9 - Using an Ingress with TLS

The ingress controller (nginx) listens on Ingress Resources and configures itself to route incoming traffic based on the host header to the correct running pods



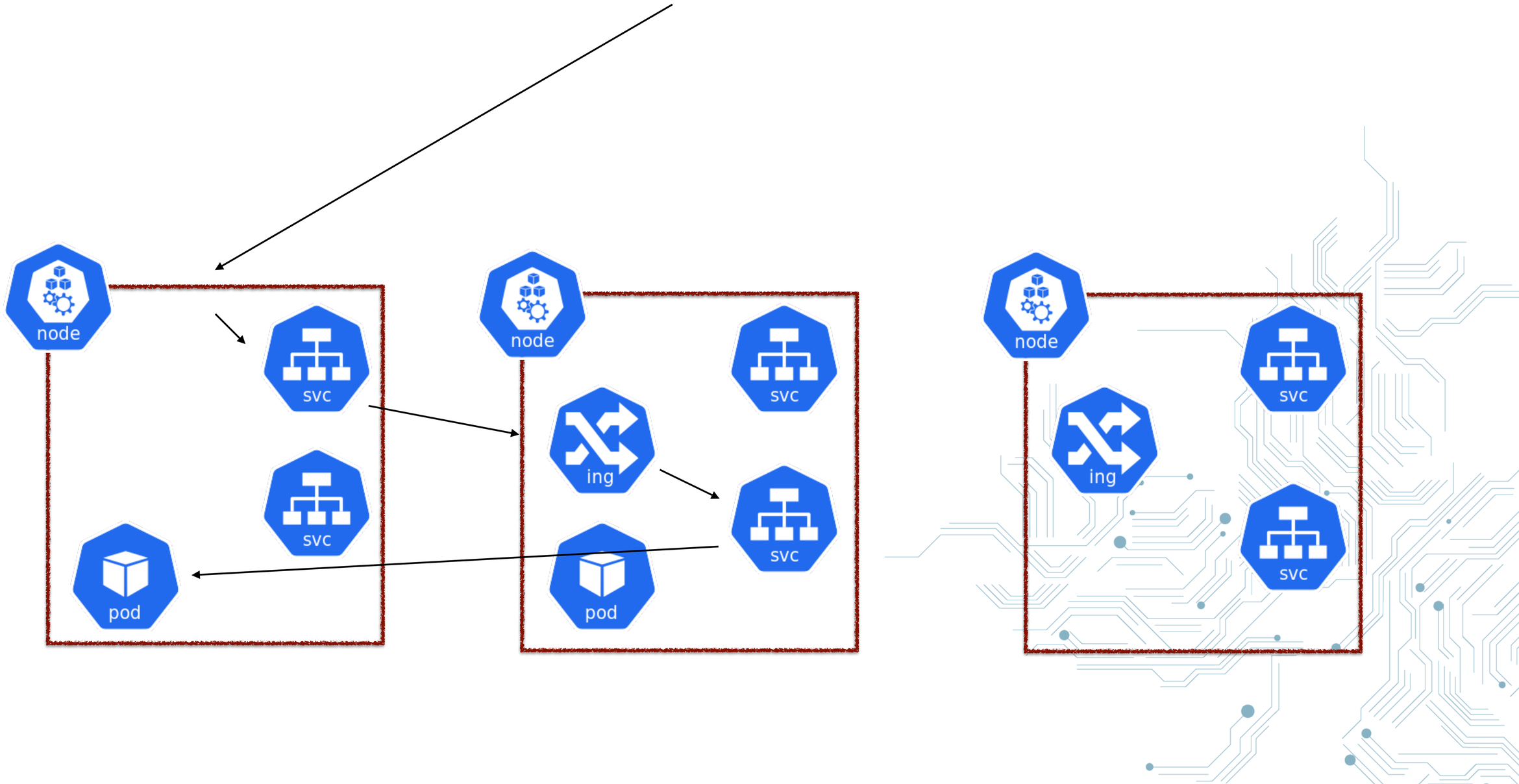
Cert-manager listens on Ingresses and if they want TLS, requests a certificate from LetsEncrypt

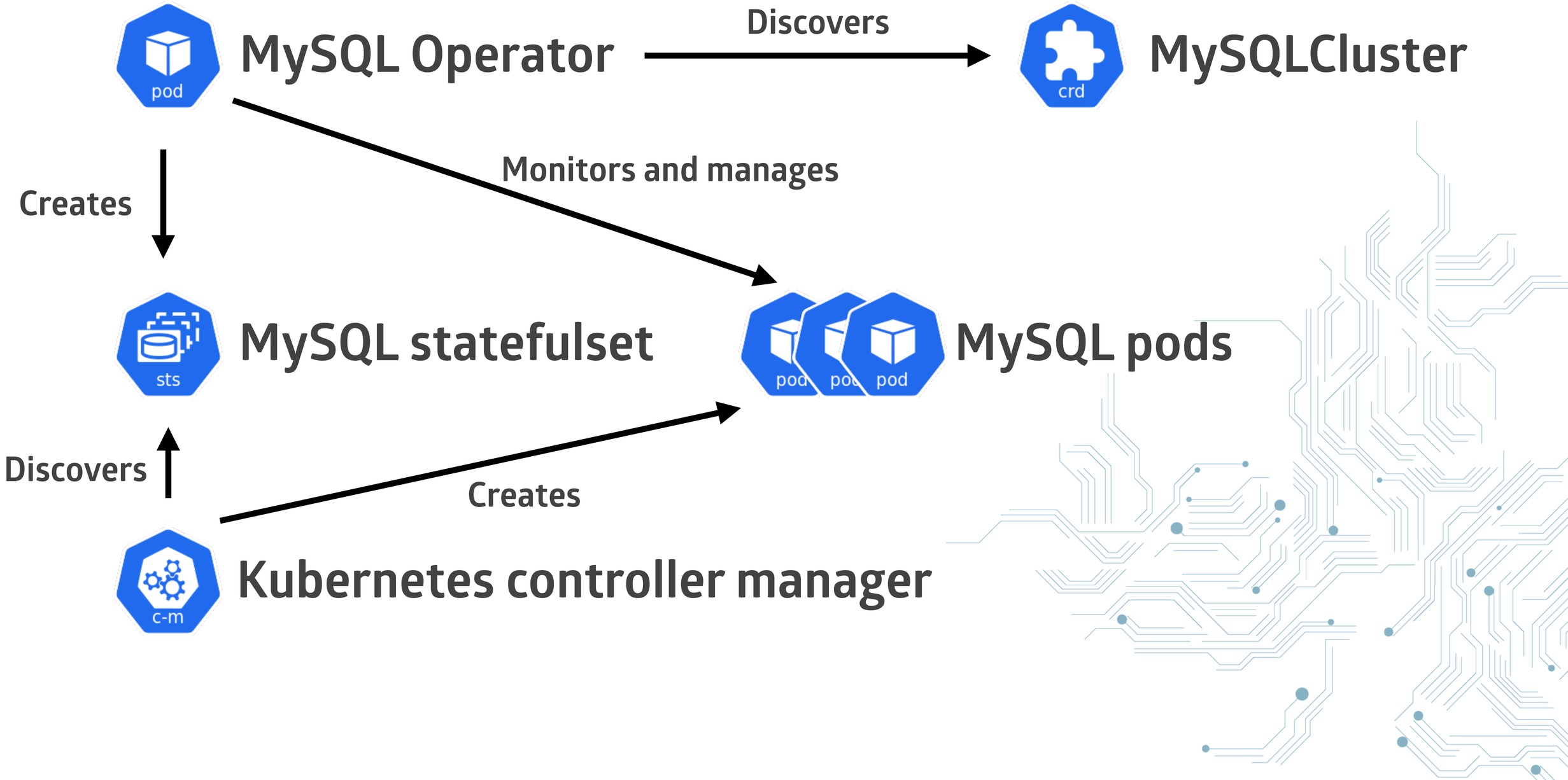


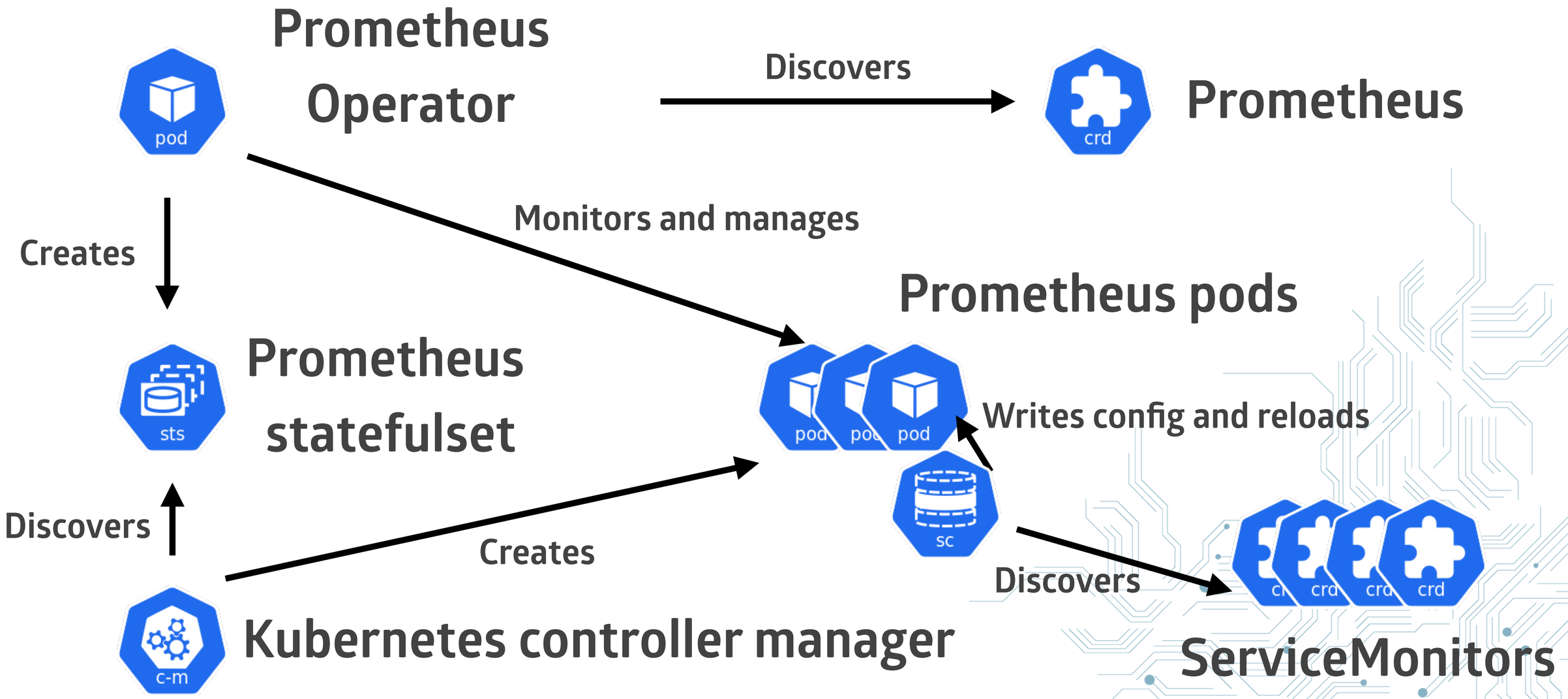
External-DNS listens on Ingresses and creates DNS entries at AWS Route 53

How is traffic routed to the Pod

OpenStack LoadBalancer

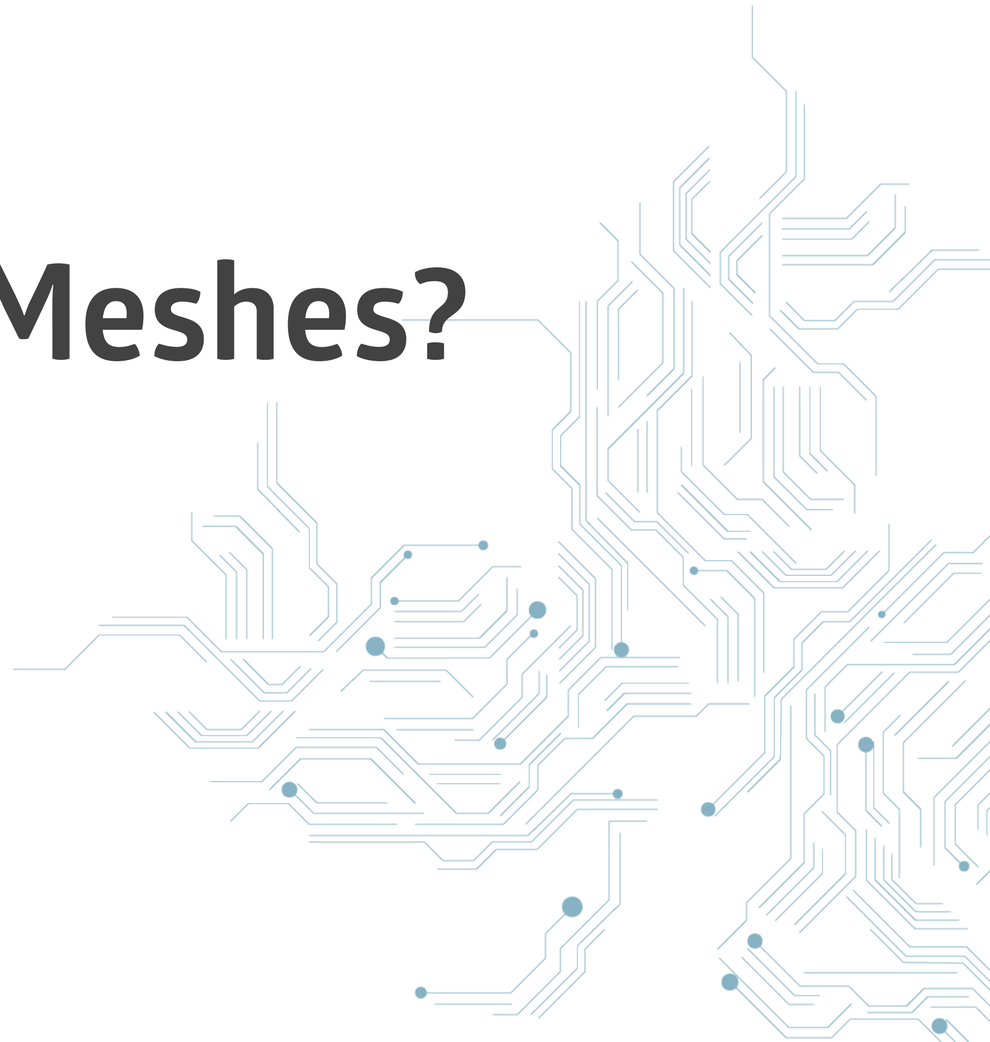


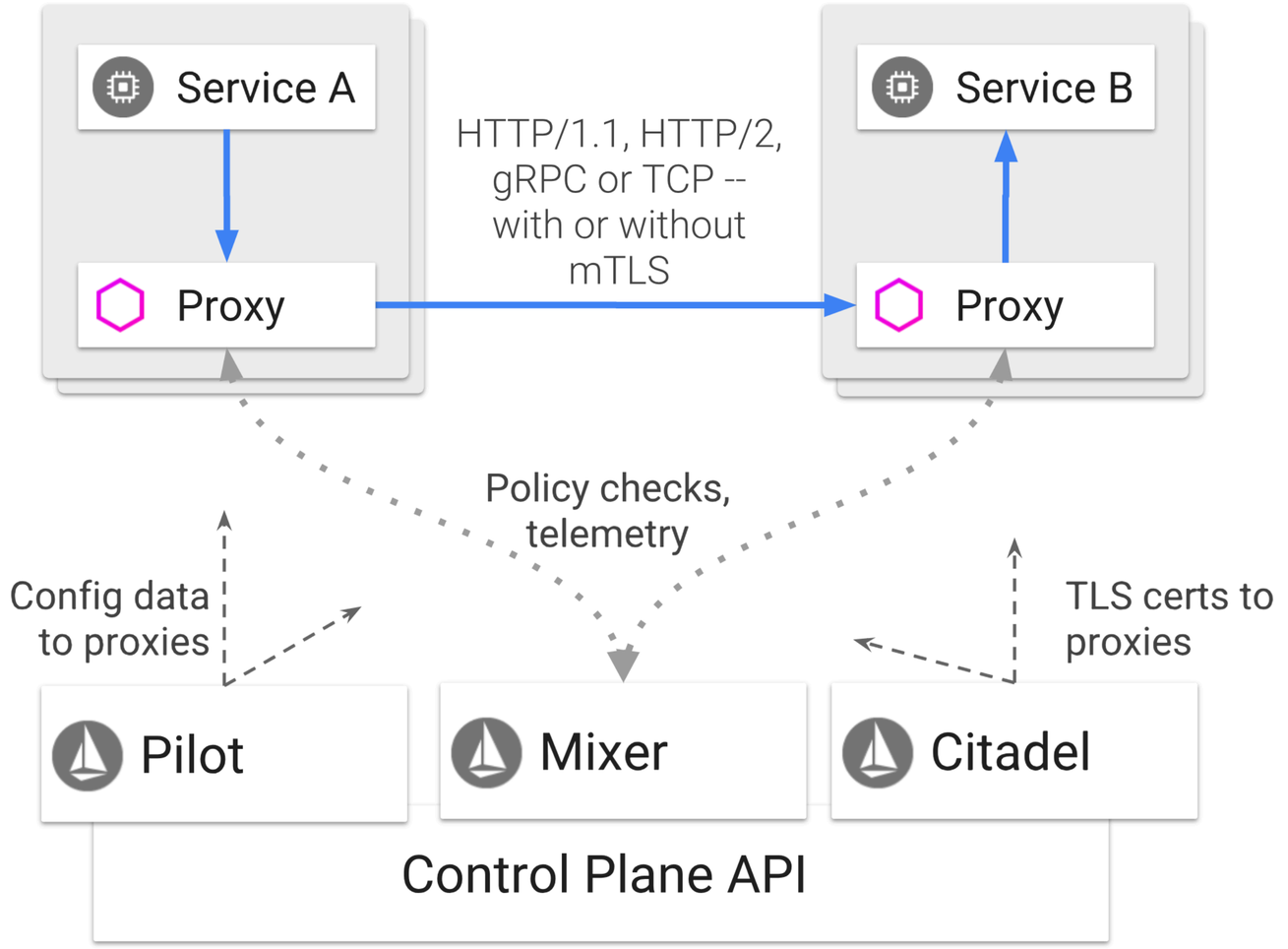

MySQL Operator
Discovers
MySQLCluster
Creates
Monitors and manages
MySQL statefulset
MySQL pods
Discovers
Creates
Kubernetes controller manager
pod
crd
sts
c-m


Prometheus Operator
pod
Discovers
crd
Prometheus
Monitors and manages
Creates
Prometheus pods
Prometheus statefulset
sts
pod
pod
pod
Writes config and reloads
sc
Discovers
Creates
Discovers
crd
crd
crd
crd
Kubernetes controller manager
c-m
ServiceMonitors

17 - Service Meshes

What are Service Meshes?

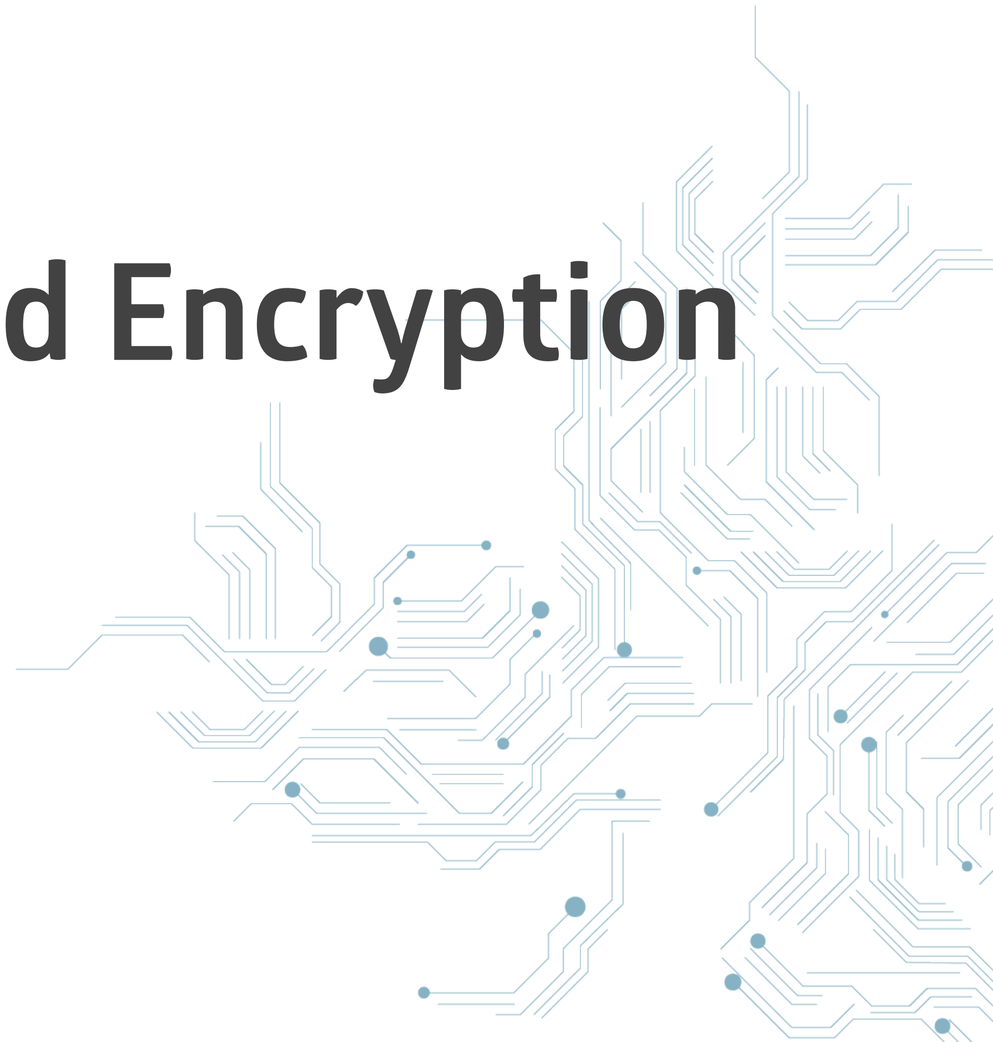



Service A
Service B
HTTP/1.1, HTTP/2,
gRPC or TCP --
with or without
mTLS
Proxy
Proxy
Policy checks,
telemetry
Config data
to proxies
TLS certs to
proxies
Pilot
Mixer
Citadel
Control Plane API

They provide

Metrics and Traces

Transparent End-To-End Encryption



Advanced Routing

Istio

LinkerD