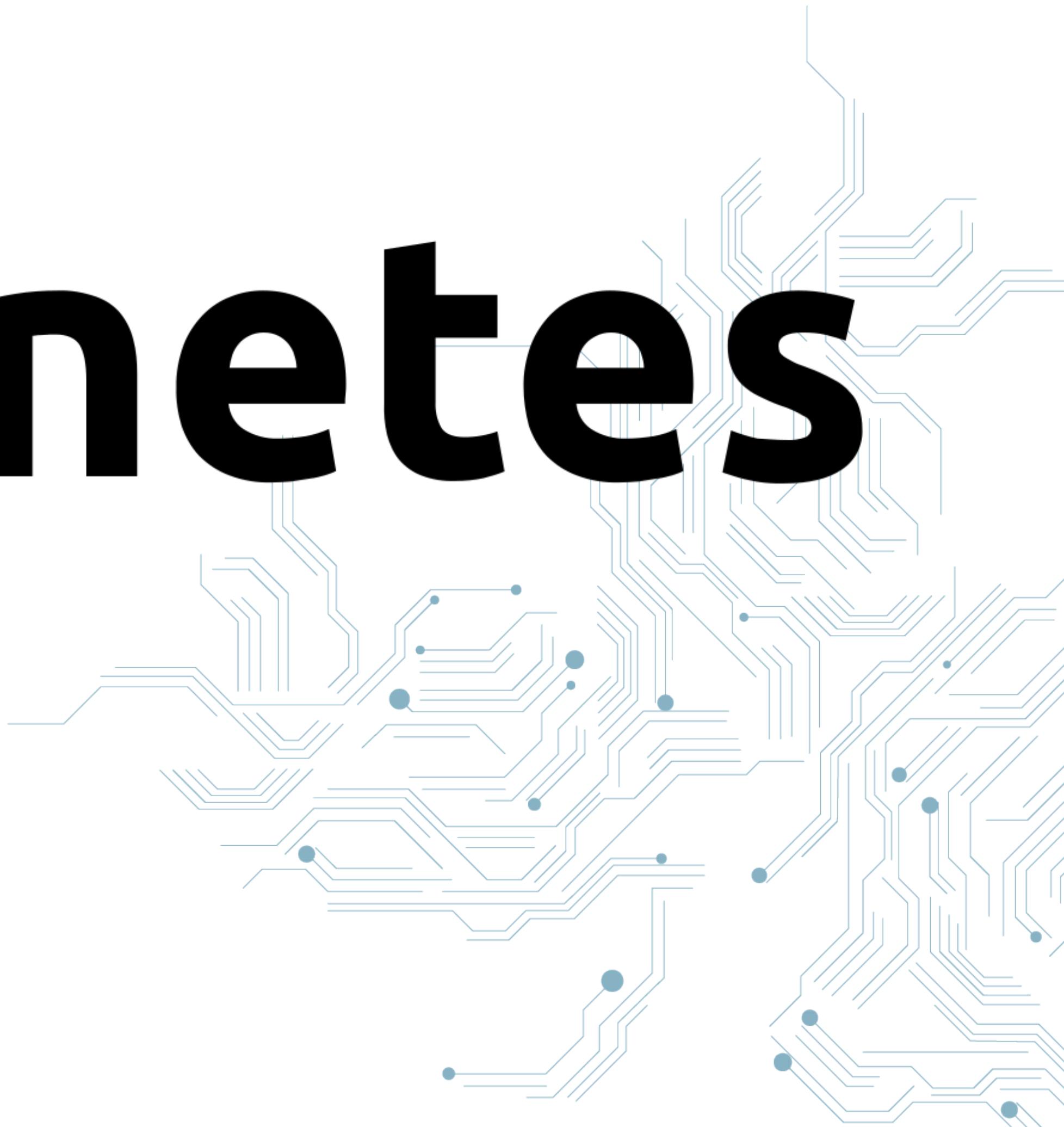


# Kubernetes Workshop

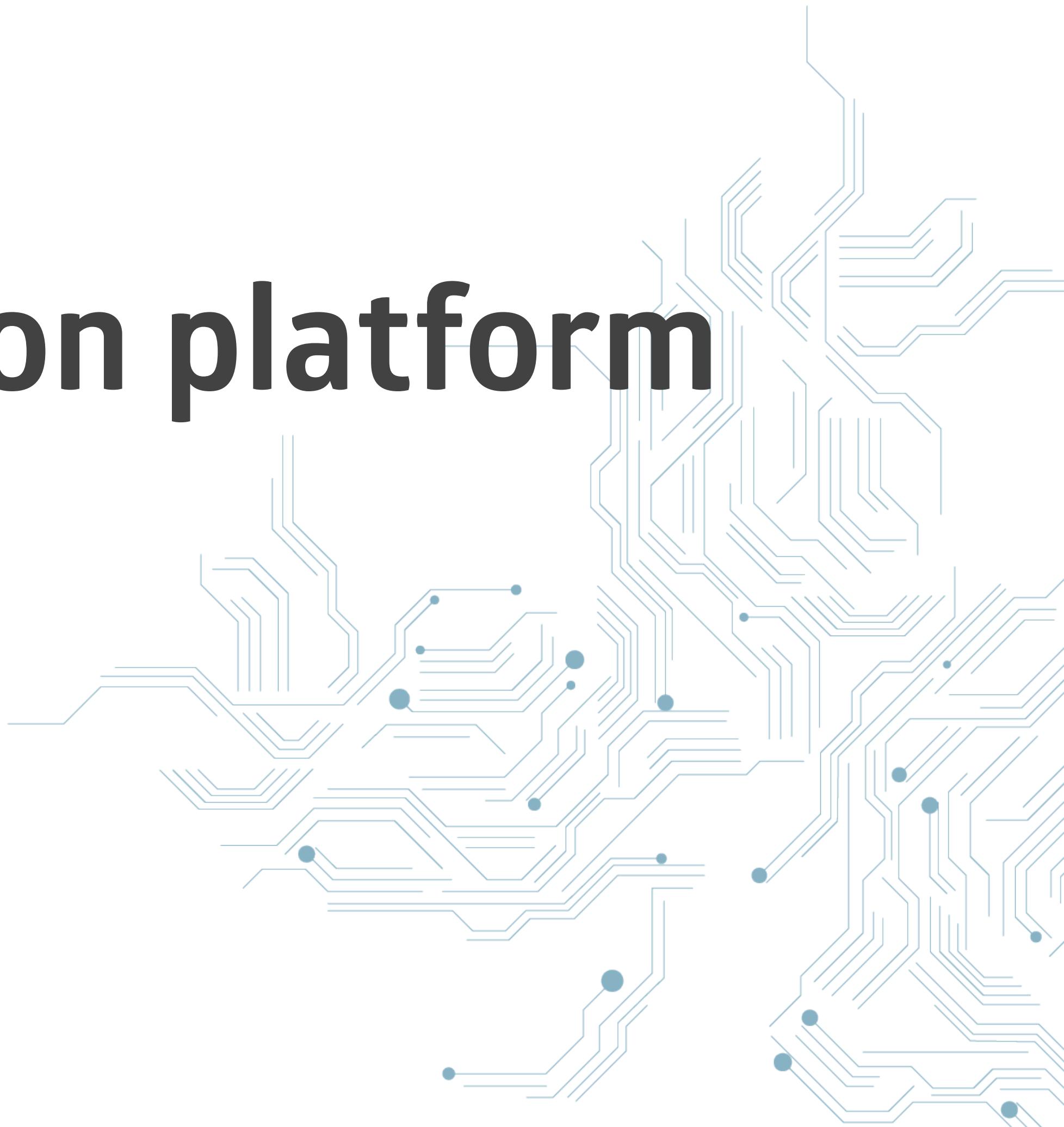




# kubernetes



# Container orchestration platform



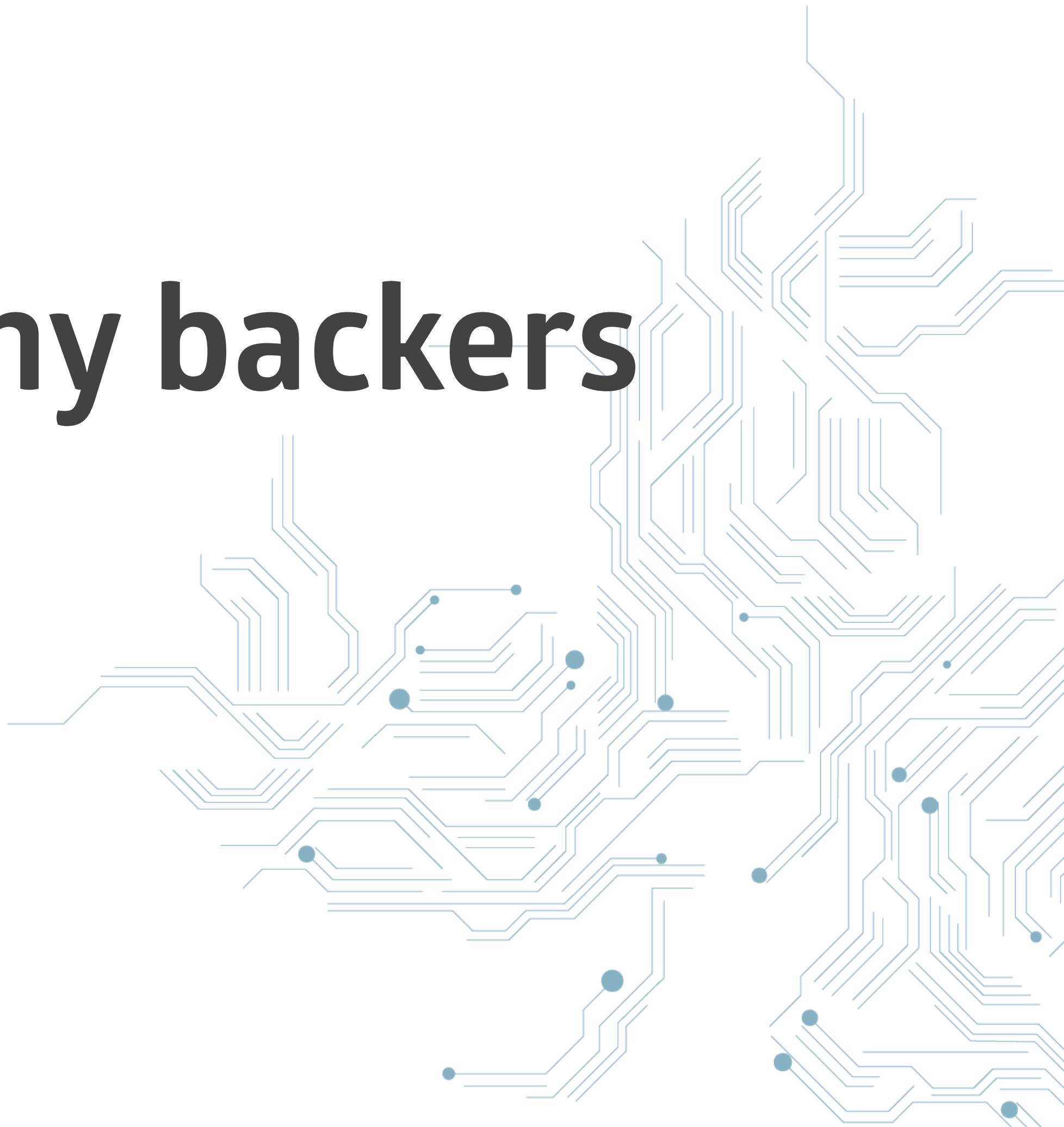
# **Deploy, run and scale your services in isolated containers**



# Very Powerful



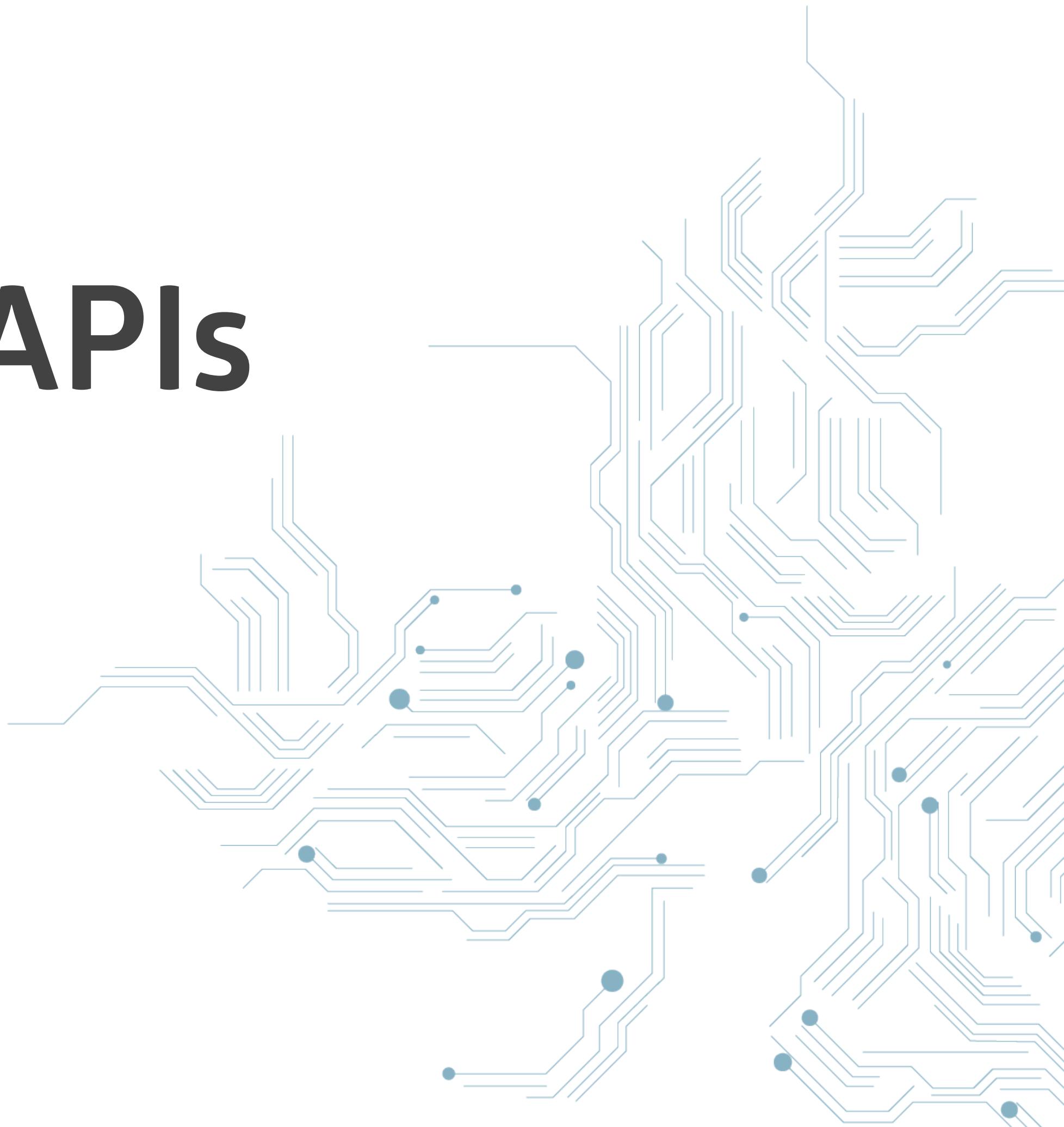
# Lot's of large company backers



# No vendor lock in



# Standardized APIs

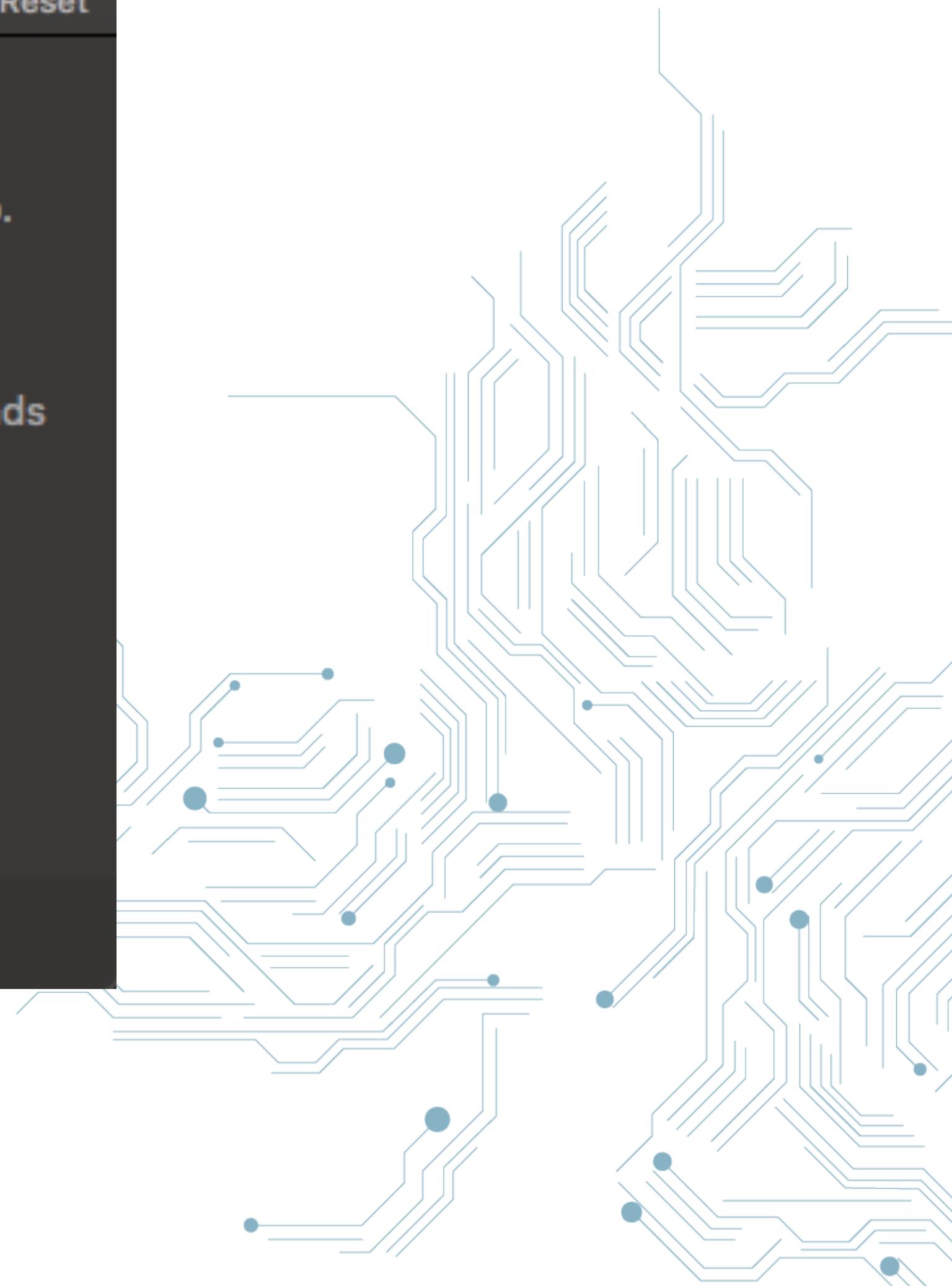
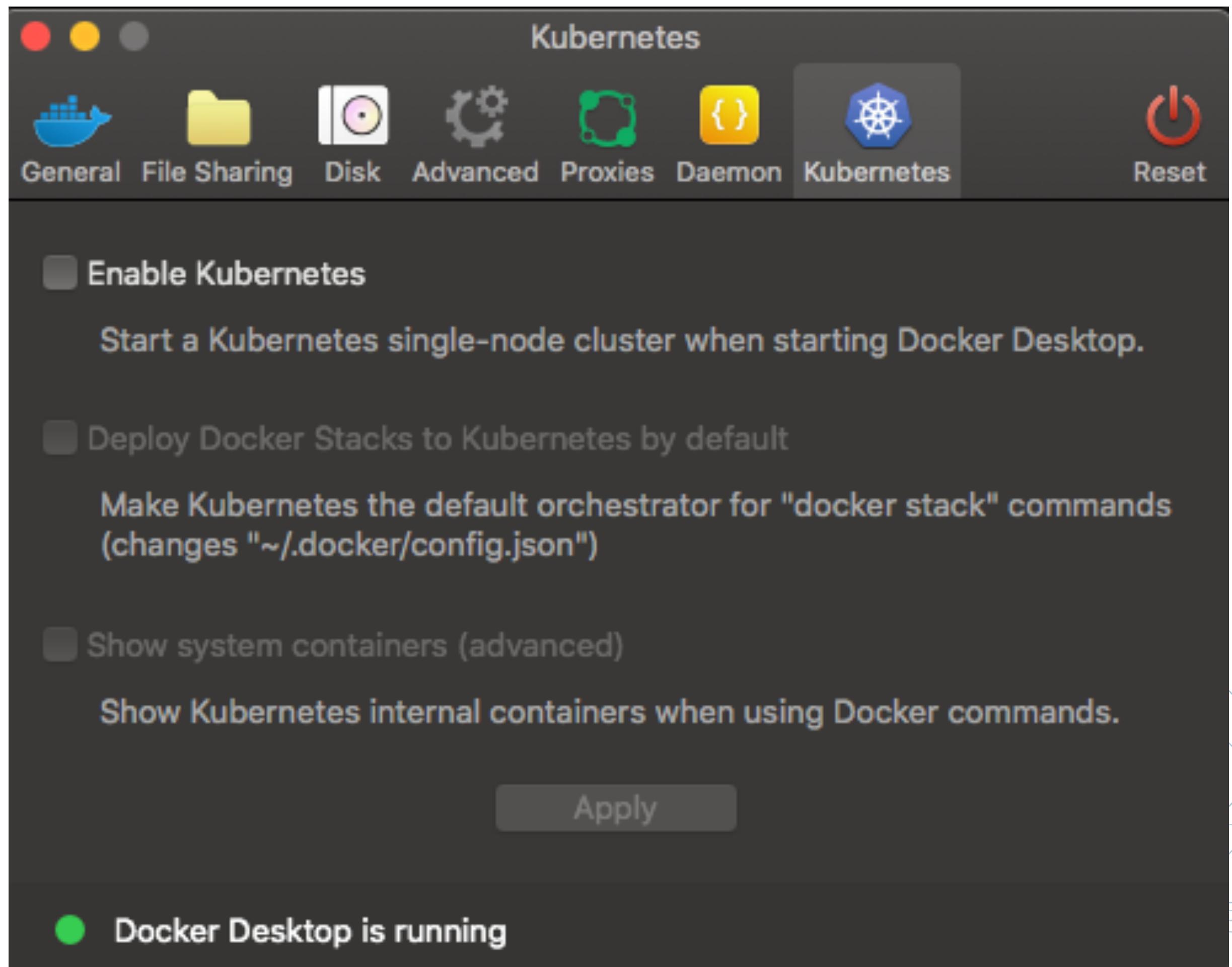


# Runs on



# Your laptop





# Bare metal



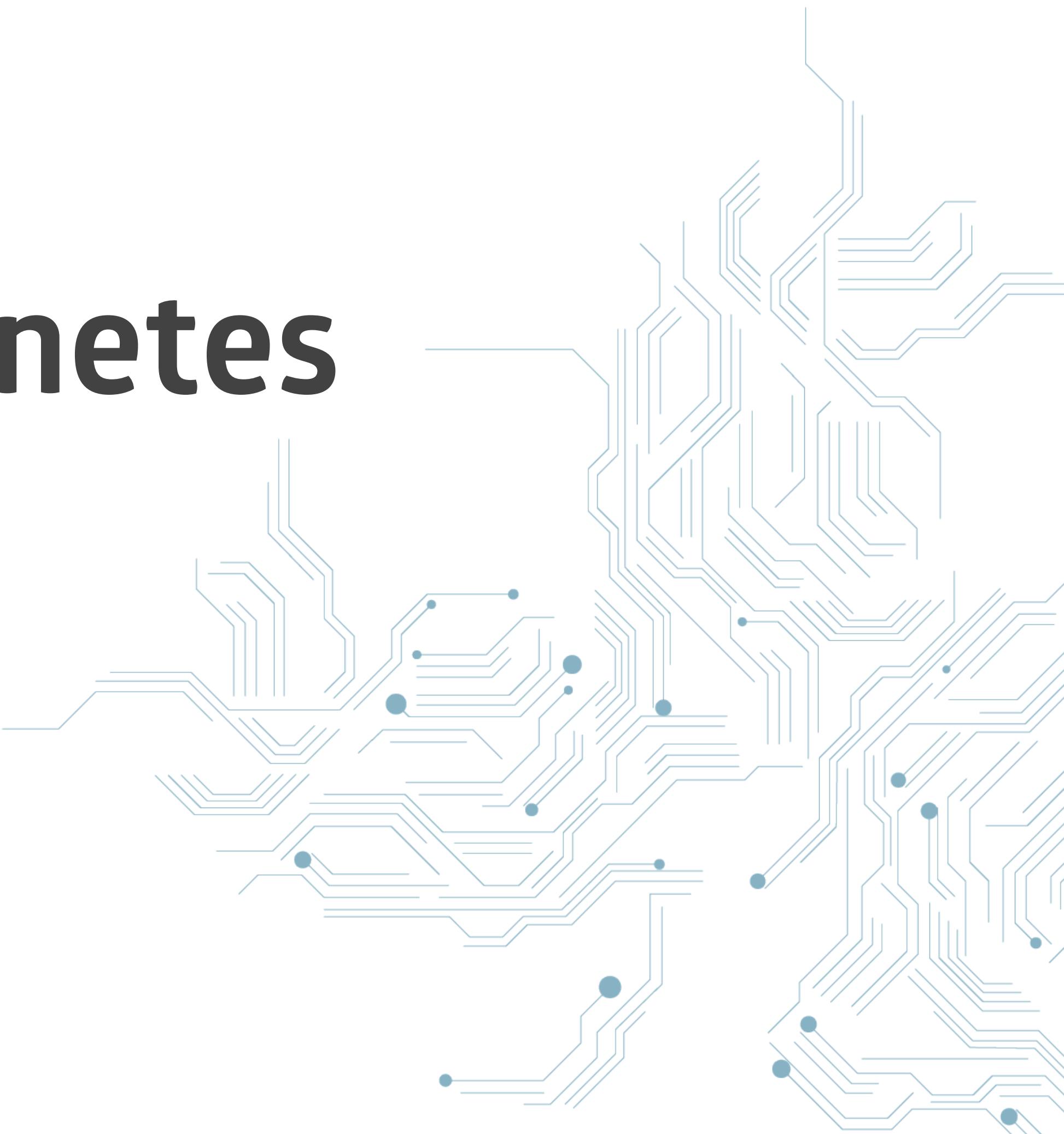
# Cloud Providers



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



# Managed Kubernetes



# CNCF Cloud Native Interactive Landscape

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: 2020-07-22 23:44:41Z

You are viewing 45 cards with a total of 746 stars, market cap of \$6.97T and funding of \$1.55B.

Landscape

Card Mode

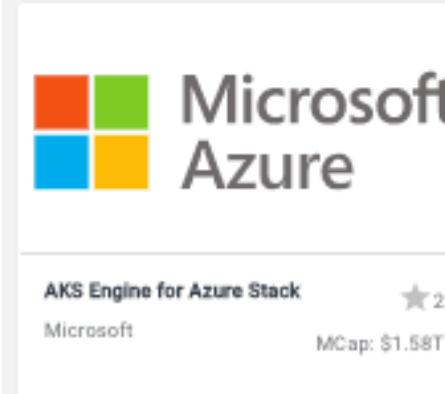
Serverless

Members



1163

Platform - Certified Kubernetes - Hosted (45)



Microsoft Azure

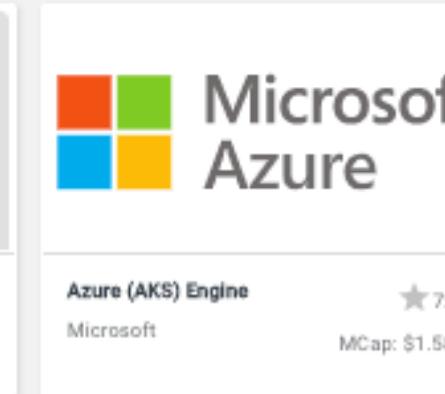
AKS Engine for Azure Stack  
Microsoft  
MCap: \$1.58T



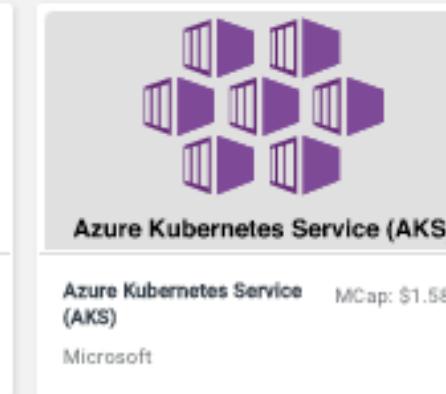
Alibaba Cloud Container Service for Kubernetes  
Alibaba Cloud



Amazon Elastic Container Service for Kubernetes (EKS)  
Amazon Web Services



Azure (AKS) Engine  
Microsoft  
MCap: \$1.58T



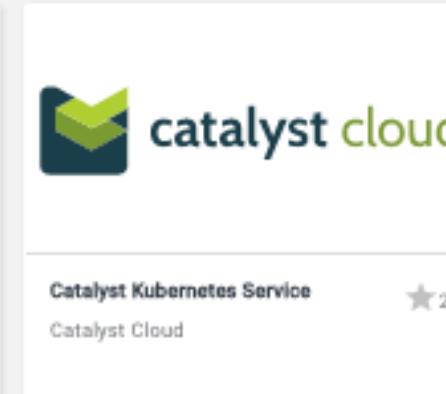
Azure Kubernetes Service (AKS)  
Microsoft  
MCap: \$1.58T



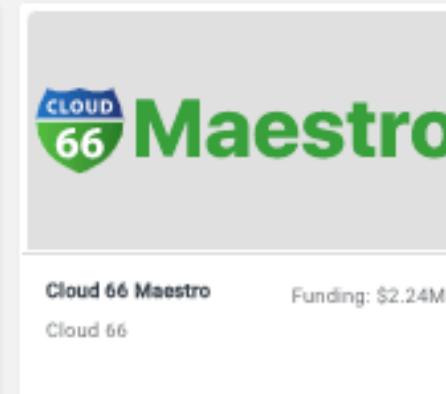
Baidu Cloud Container Engine  
Baidu  
MCap: \$43.16B



BoCloud BeyondContainer  
BoCloud  
Funding: \$29.77M



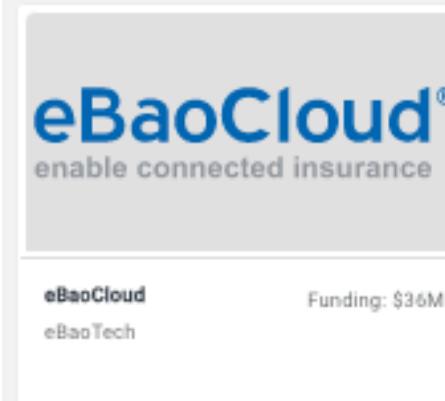
Catalyst Kubernetes Service  
Catalyst Cloud  
★ 2



Cloud 66 Maestro  
Cloud 66  
Funding: \$2.24M



DigitalOcean Kubernetes  
DigitalOcean  
Funding: \$455.41M

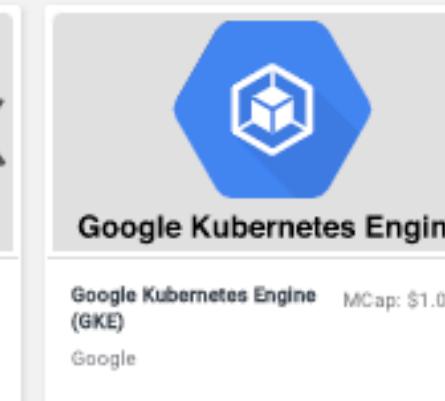


eBaoCloud®  
enable connected insurance

eBaoCloud  
eBaoTech  
Funding: \$36M



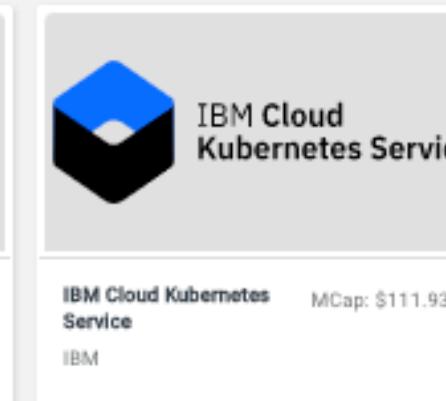
ELASTX Private Kubernetes  
ELASTX



Google Kubernetes Engine



HUAWEI



IBM Cloud Kubernetes Service  
Huawei Technologies  
MCap: \$111.93B



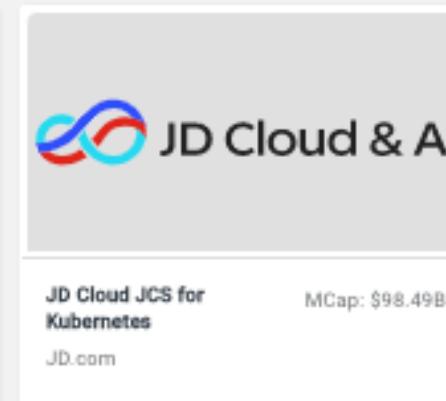
inspur



inspur



inspur



JD Cloud JCS for Kubernetes  
JD.com  
MCap: \$98.49B



JD.com JDOS Hosted  
JD.com  
MCap: \$98.49B



Kingsoft Cloud

Kingsoft Container Engine  
Kingsoft  
MCap: \$7.24B



Launcher Tech LStack Container Service for Kubernetes  
Hangzhou Launcher Technology



Linaro Developer Cloud Kubernetes Service  
Linaro  
★ 1



linode



MAIL.RU CLOUD SOLUTIONS



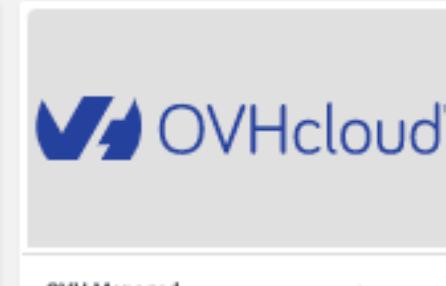
nirmata



Nutanix Karbon



Oracle Container Engine



OVH Managed Kubernetes Service  
OVHcloud  
Funding: \$736.63M



QingCloud KubeSphere Engine (QKE)  
QingCloud  
Funding: \$280.83M



RAFAY

Rafay  
Rafay Systems  
Funding: \$8M



Red Hat OpenShift Dedicated  
Red Hat  
MCap: \$111.93B



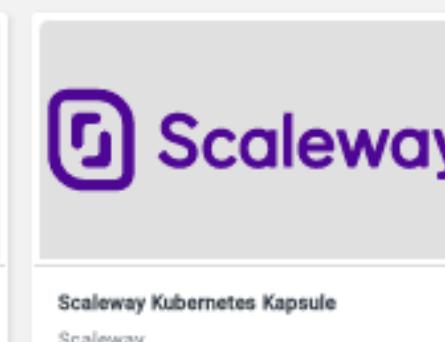
Red Hat OpenShift on IBM Cloud  
IBM  
MCap: \$111.93B



Samsung SDS Kubernetes Service  
Samsung SDS



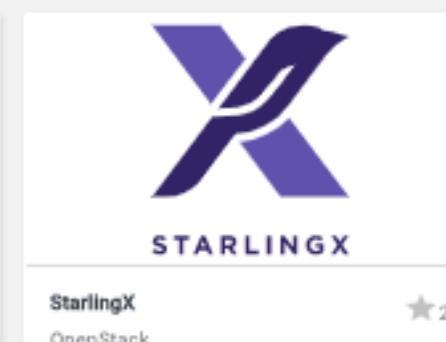
SAP Certified Gardener  
SAP  
MCap: \$188.79B



Scaleway Kubernetes Kapsule  
Scaleway



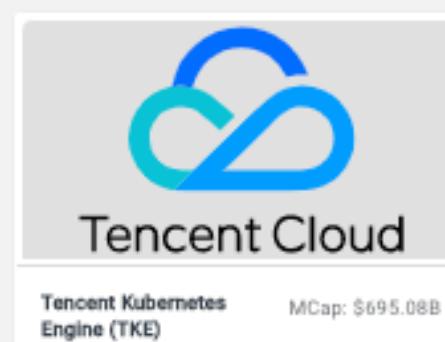
SOFASTACK Cloud Application Fabric Engine  
Ant Financial  
MCap: \$717.42B



StarlingX  
OpenStack  
★ 2



SysEleven MetaKube  
SysEleven

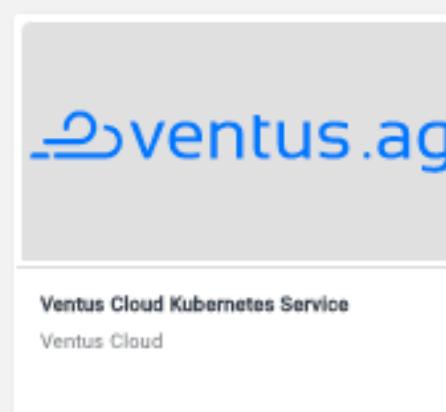


Tencent Kubernetes Engine (TKE)  
Tencent Holdings  
MCap: \$695.08B



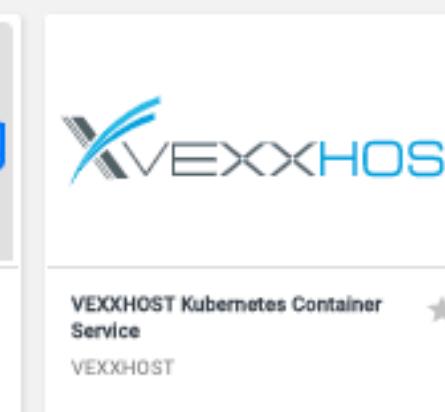
UCLOUD 优刻得

UCloud Kubernetes Service (UKBS)  
UCloud Information Technology  
MCap: \$4.04B



ventus.ag

Ventus Cloud Kubernetes Service  
Ventus Cloud  
MCap: \$4.04B



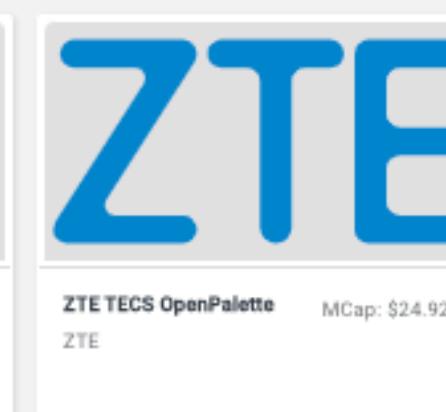
VEXXHOST

VEXXHOST Kubernetes Container Service  
VEXXHOST  
MCap: \$2.04B



网宿云

WANGSU CLOUD Container Service  
Wangsu Science & Technology  
MCap: \$3.16B



ZTE

ZTE TECS OpenPalette  
ZTE  
MCap: \$24.92B

# Easy setup



# Easy upgrades



# Easy scaling



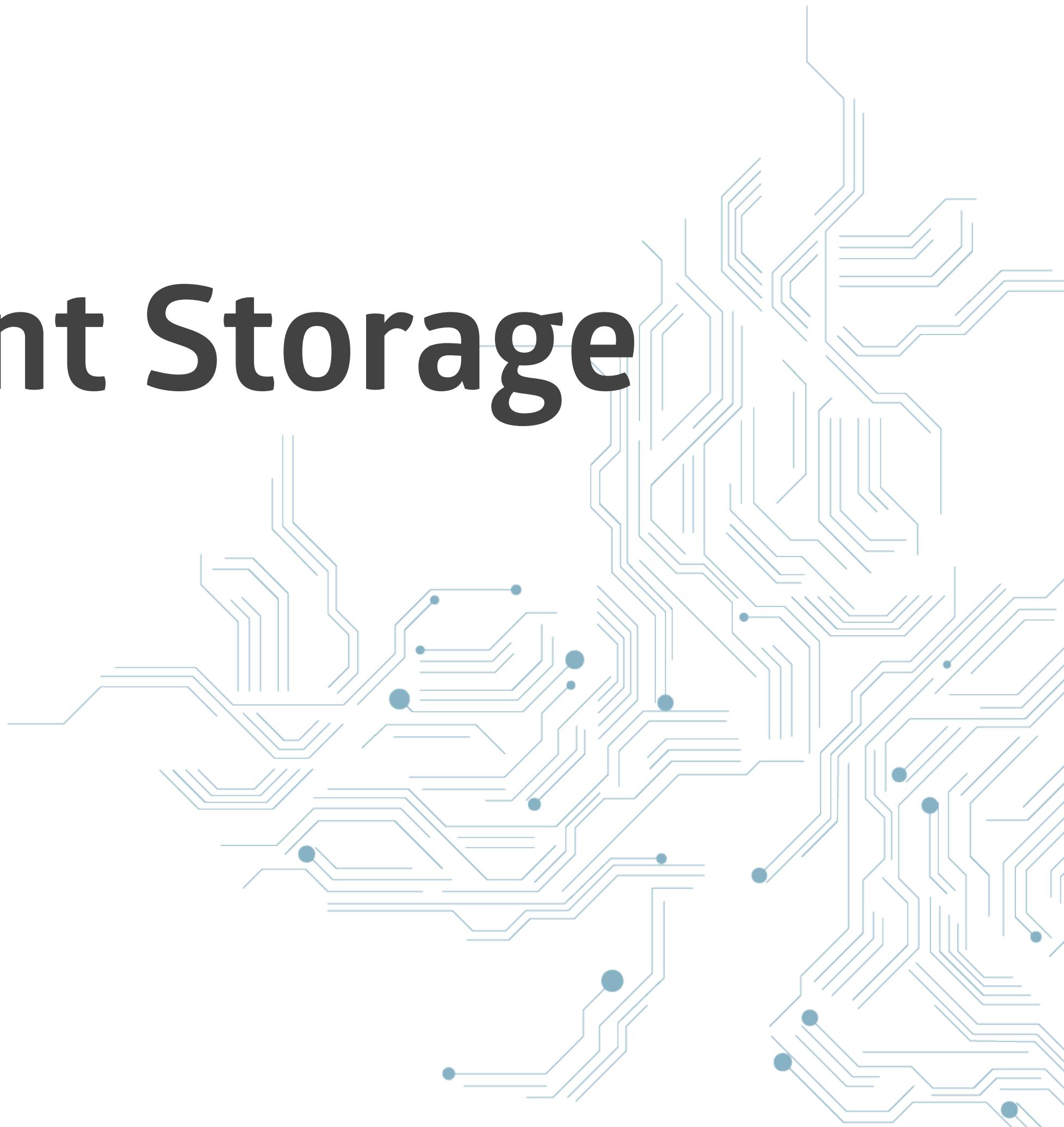
# Features



# Load Balancing



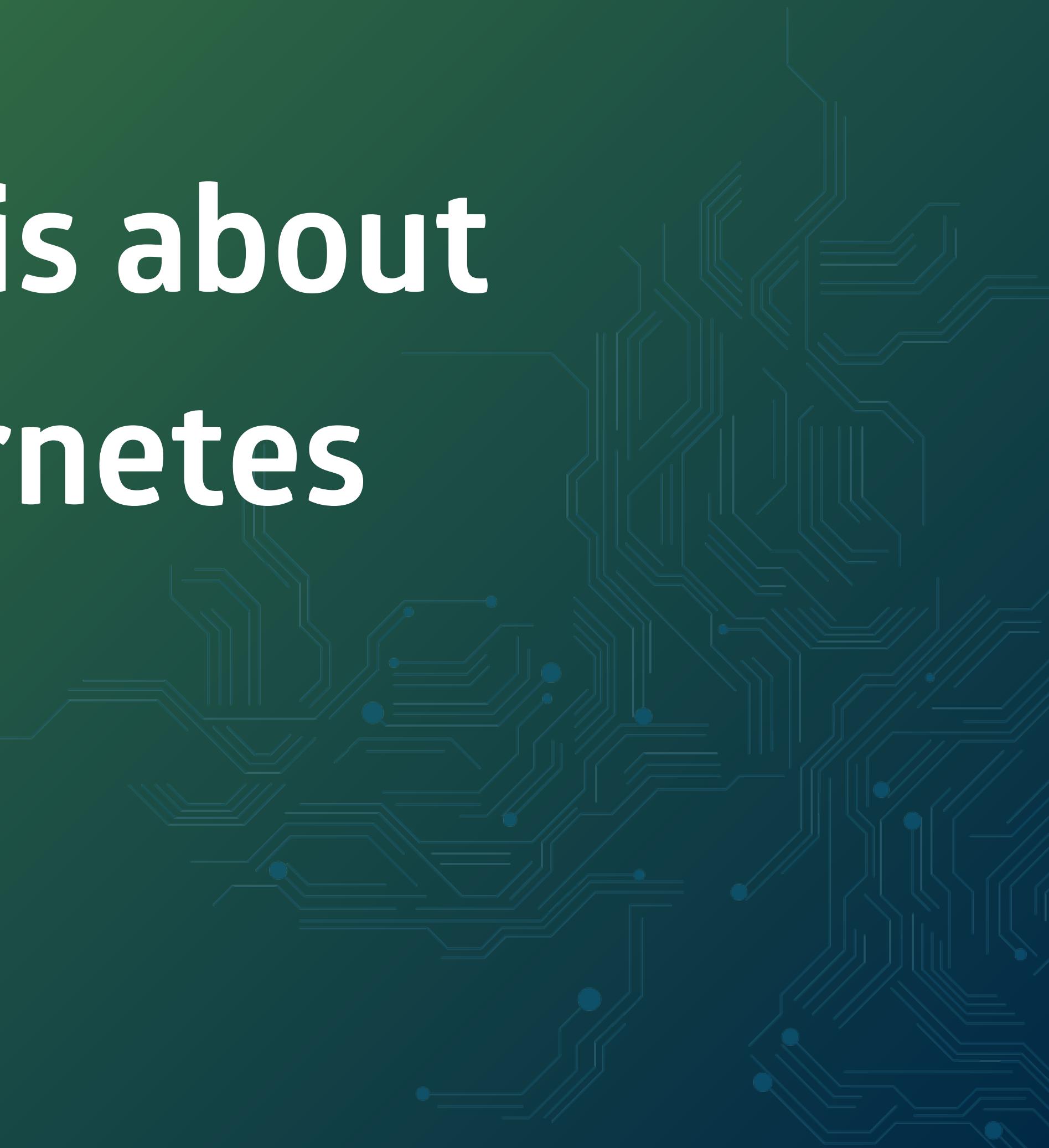
# Distributed Persistent Storage



# Backups



But this workshop is about  
how to use Kubernetes



# Learning curve

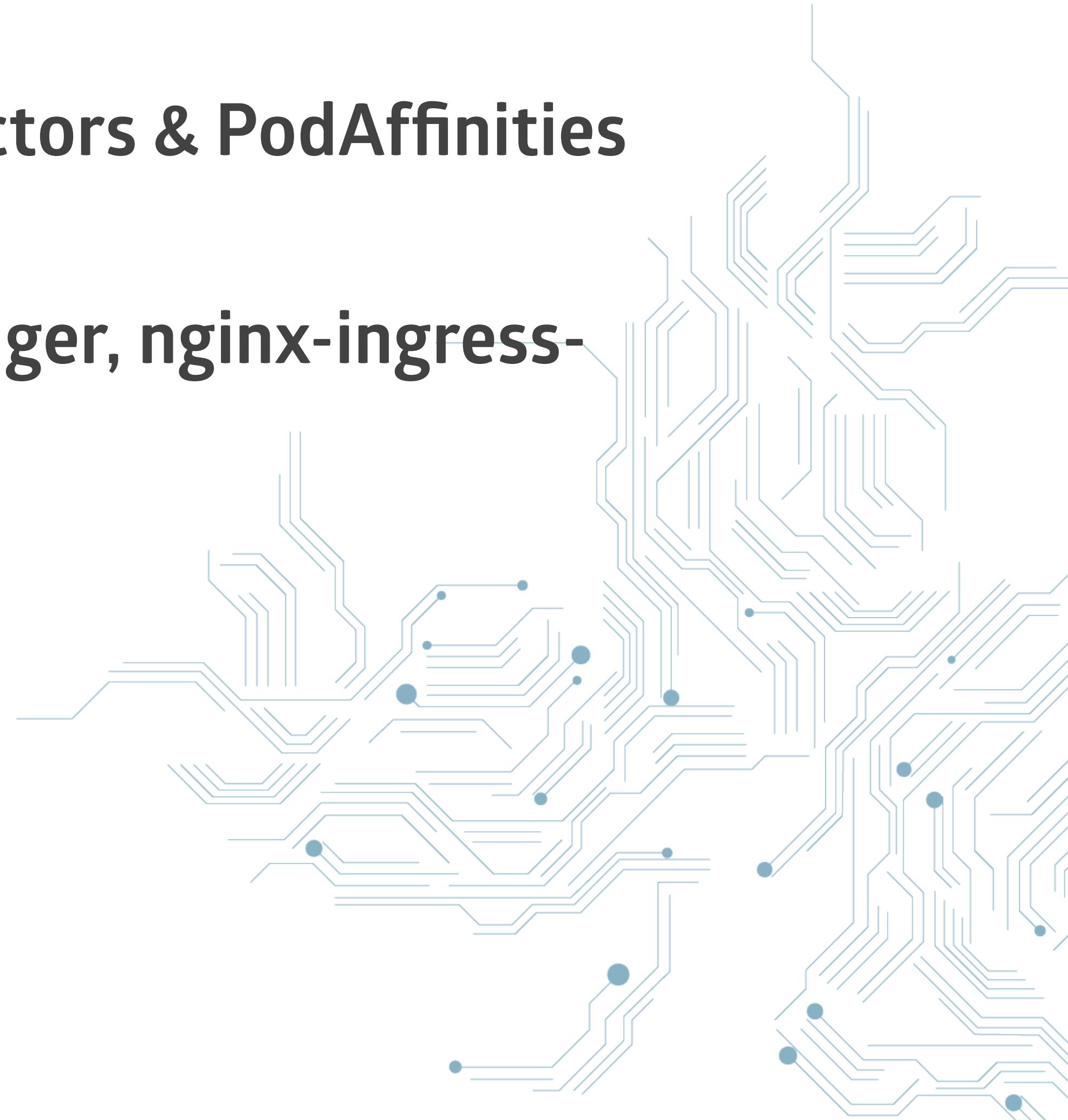


# Agenda





- **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**



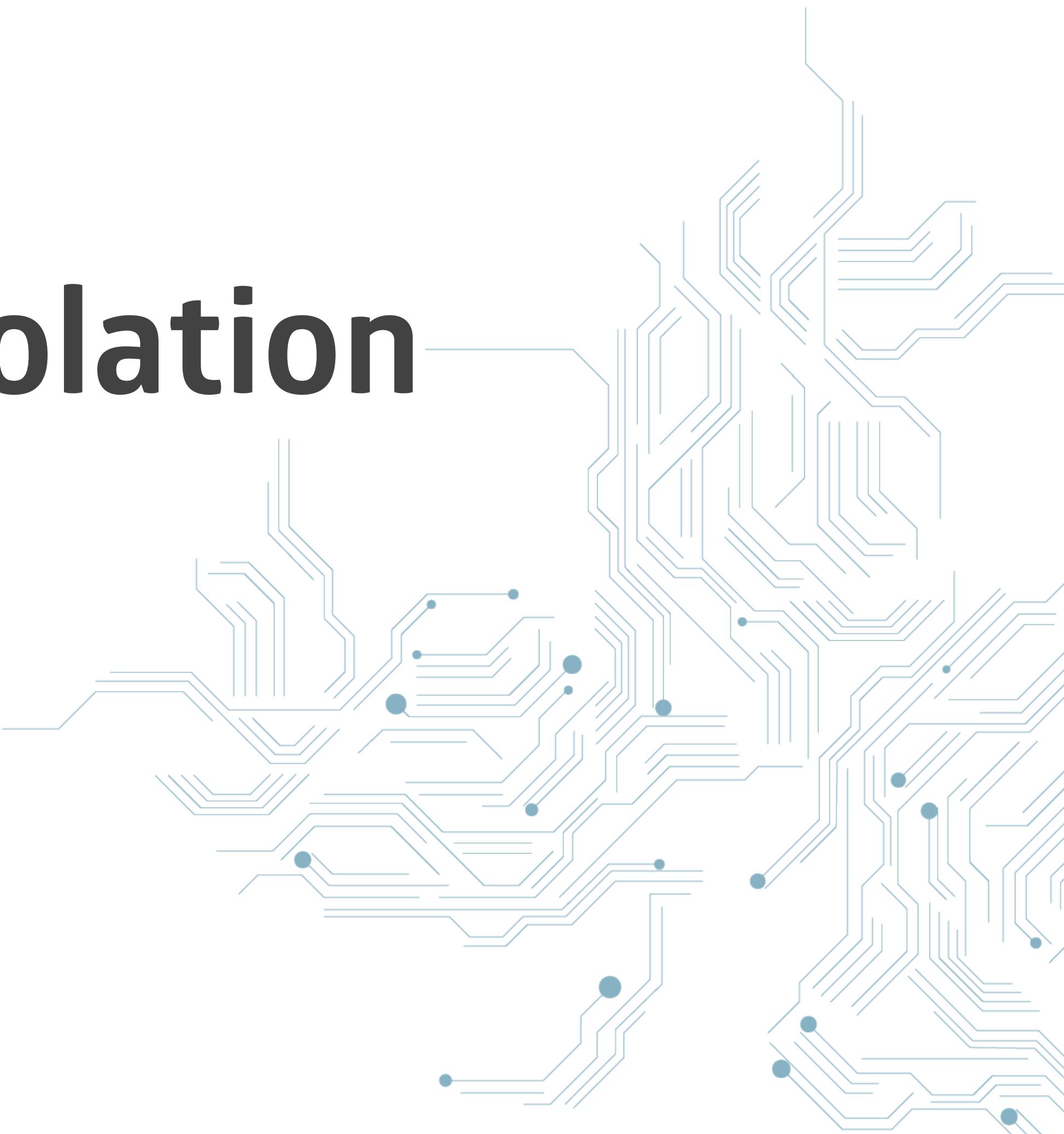
# But first



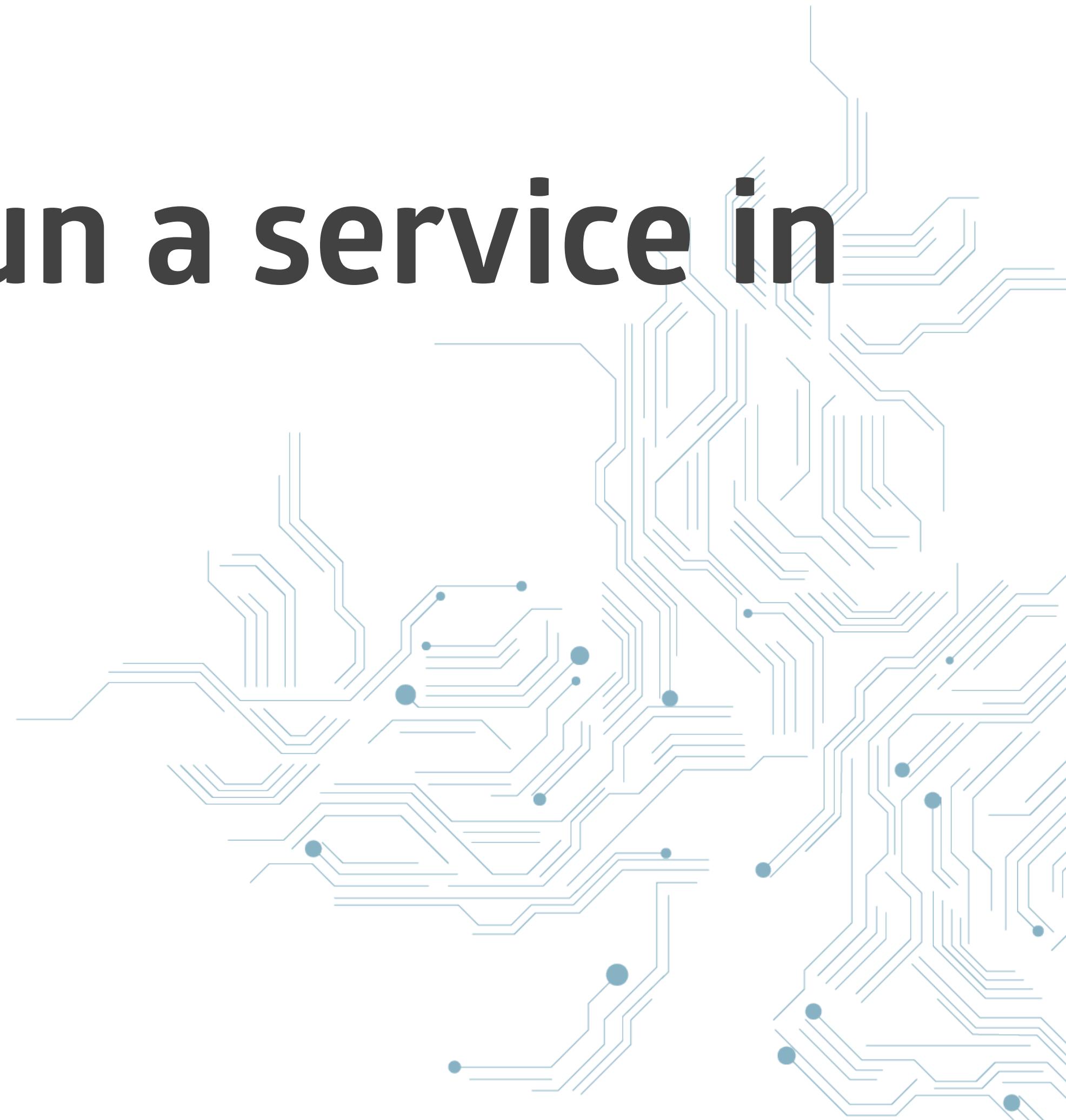
# Why containers?



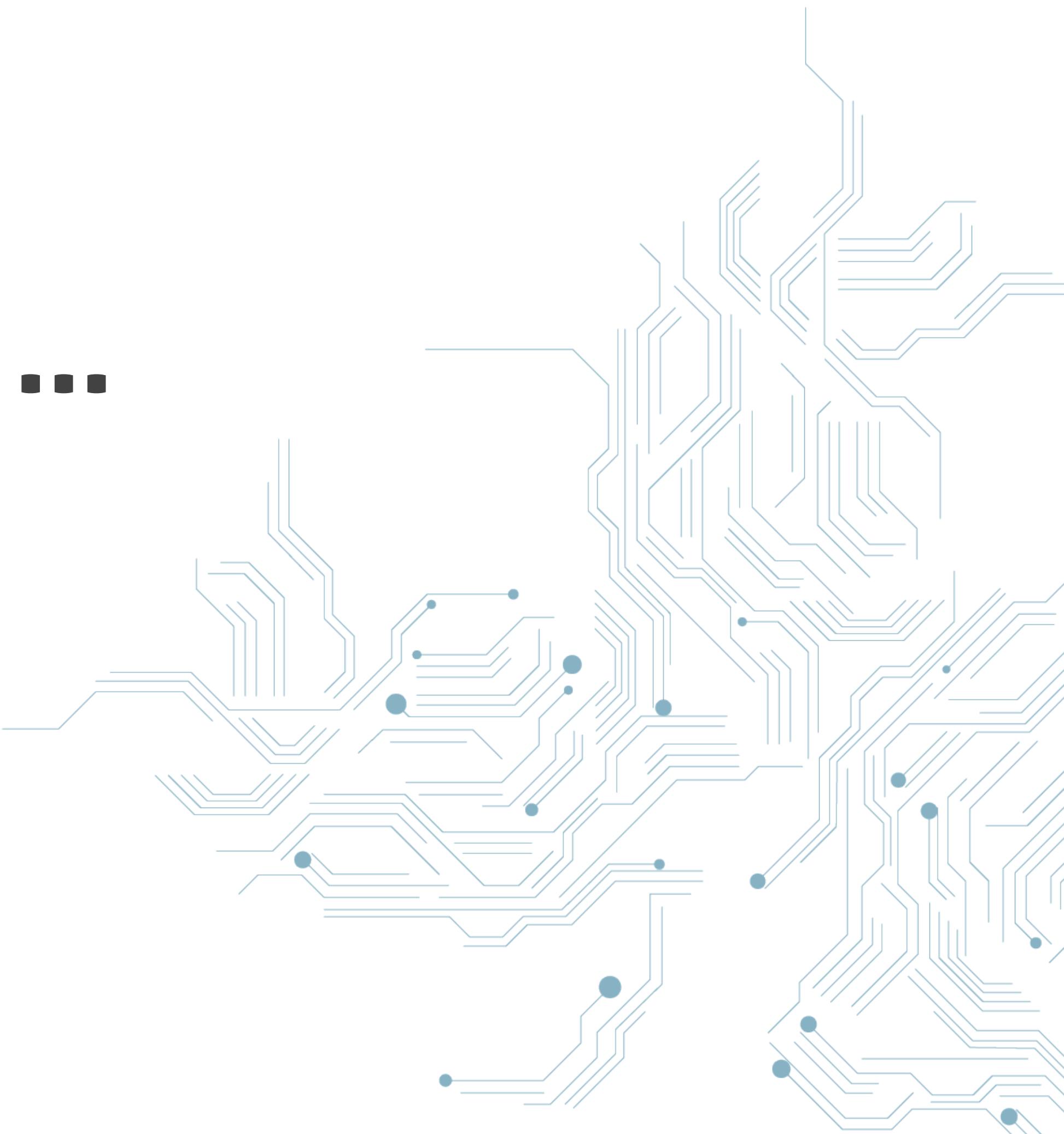
# Services run in isolation



**Everything needed to run a service in  
one image**



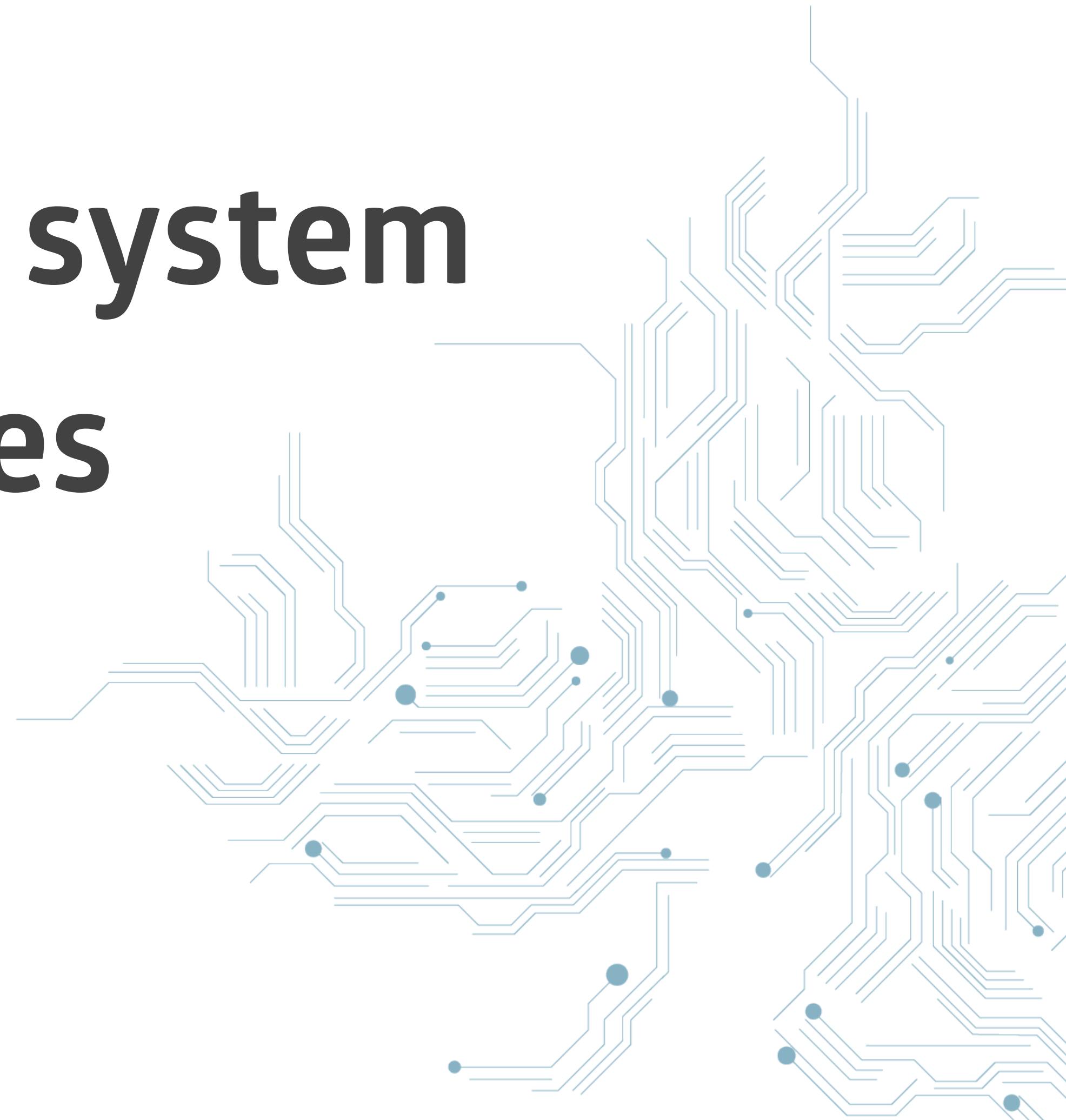
# Make things ...



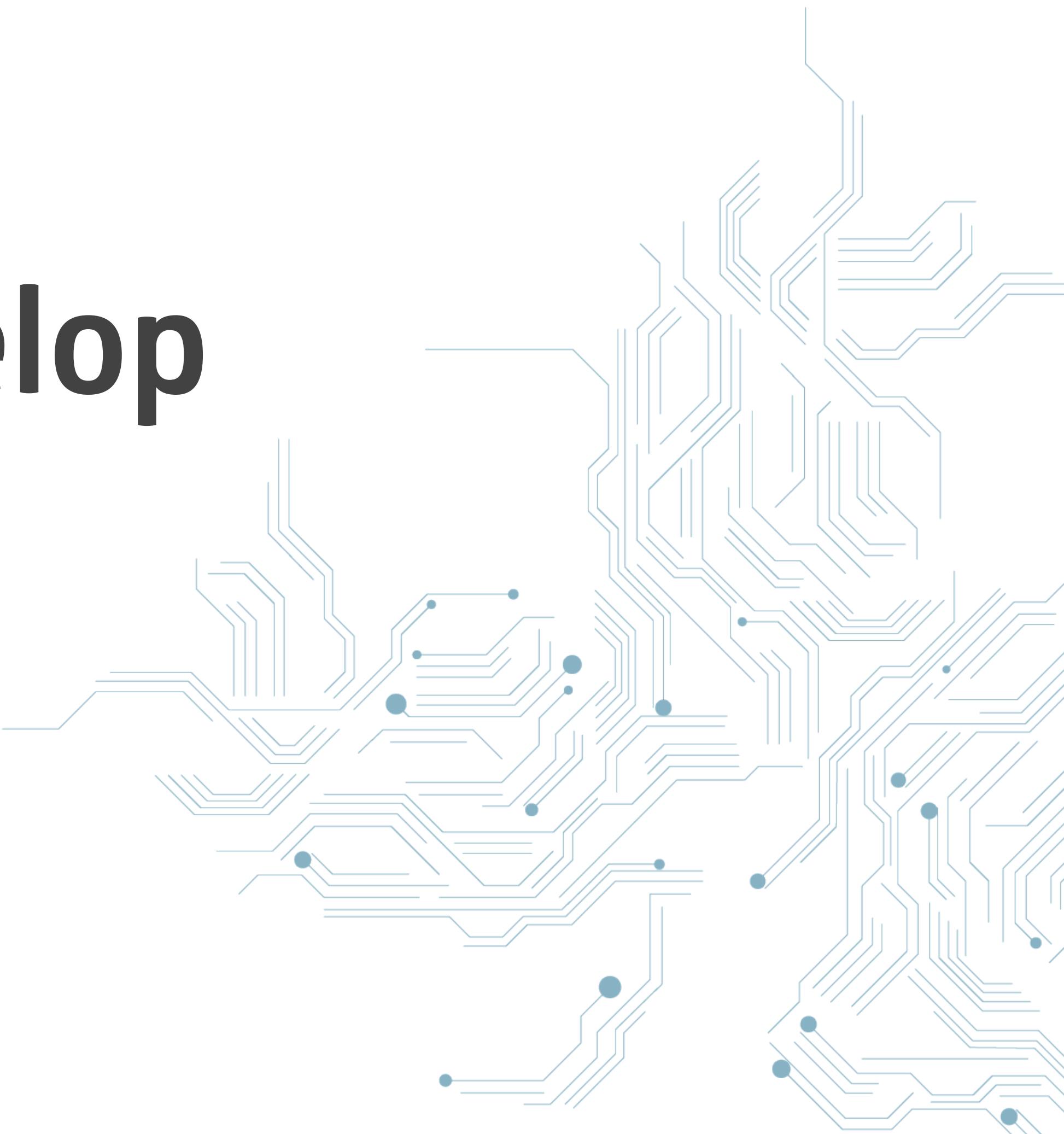
# Easier to deploy



# Easier to upgrade system dependencies



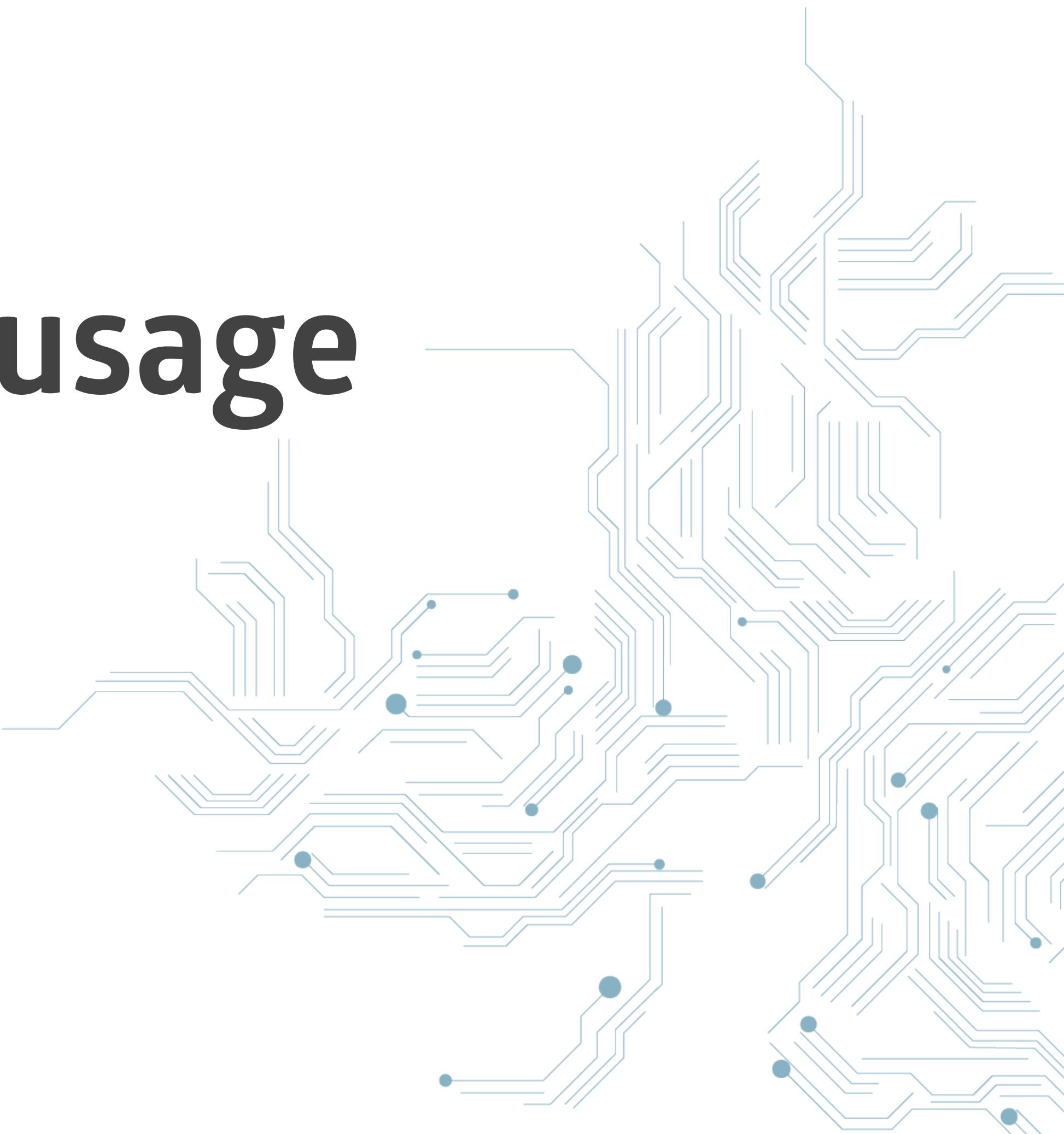
# Easier to develop



# Easier to scale

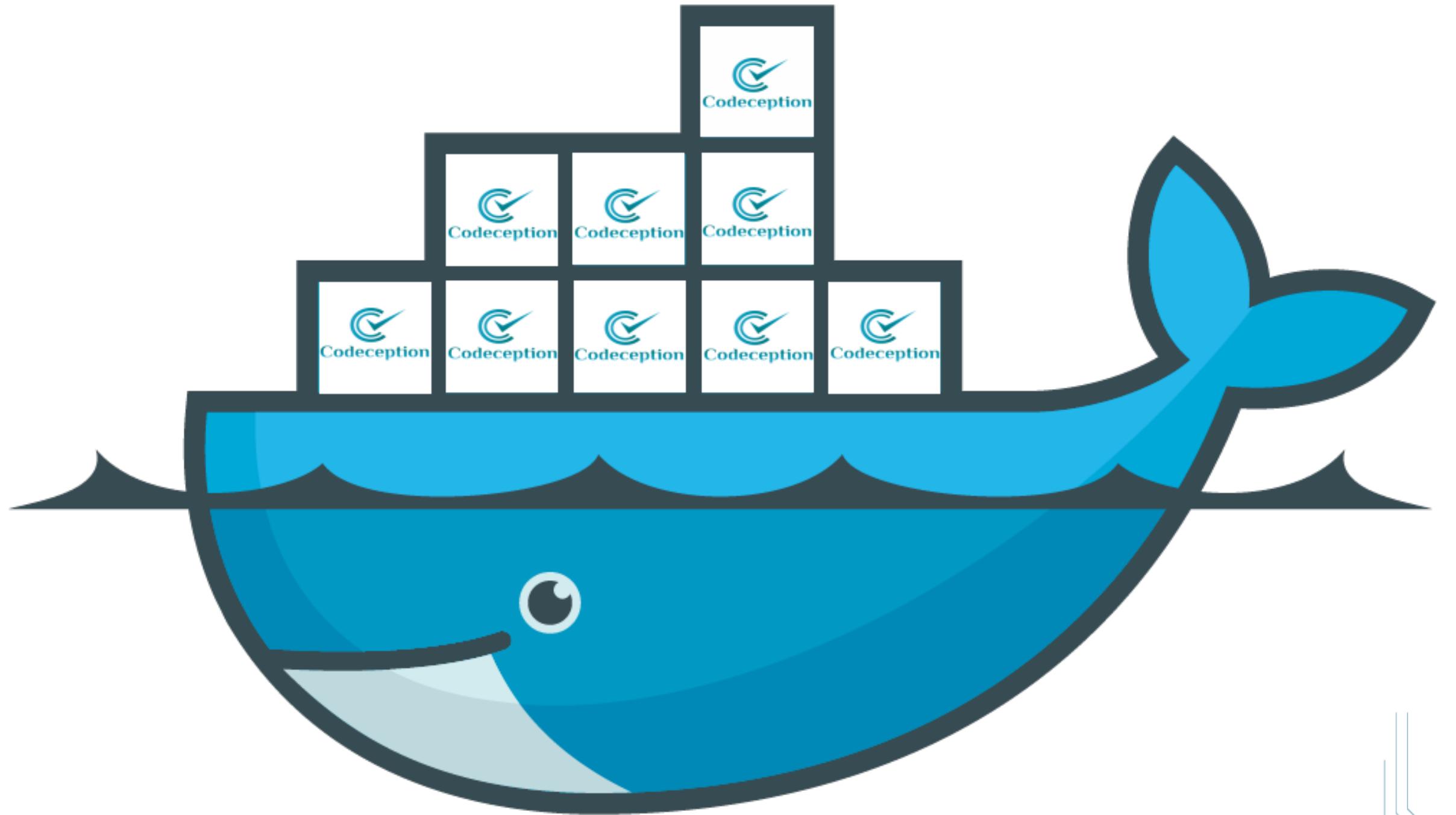


# Better resource usage

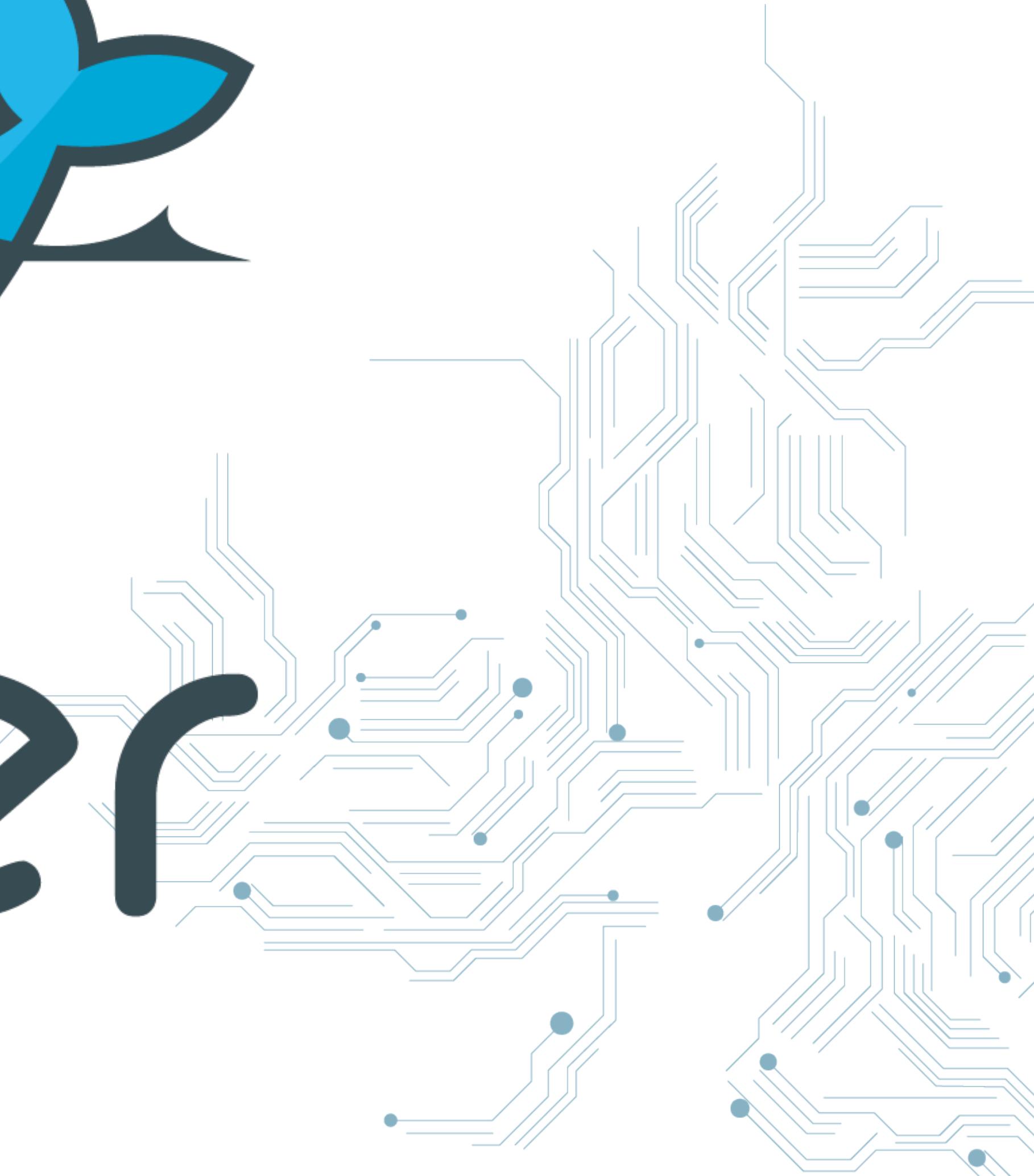


# #saveThePlanet





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.sys11.de/sys11/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

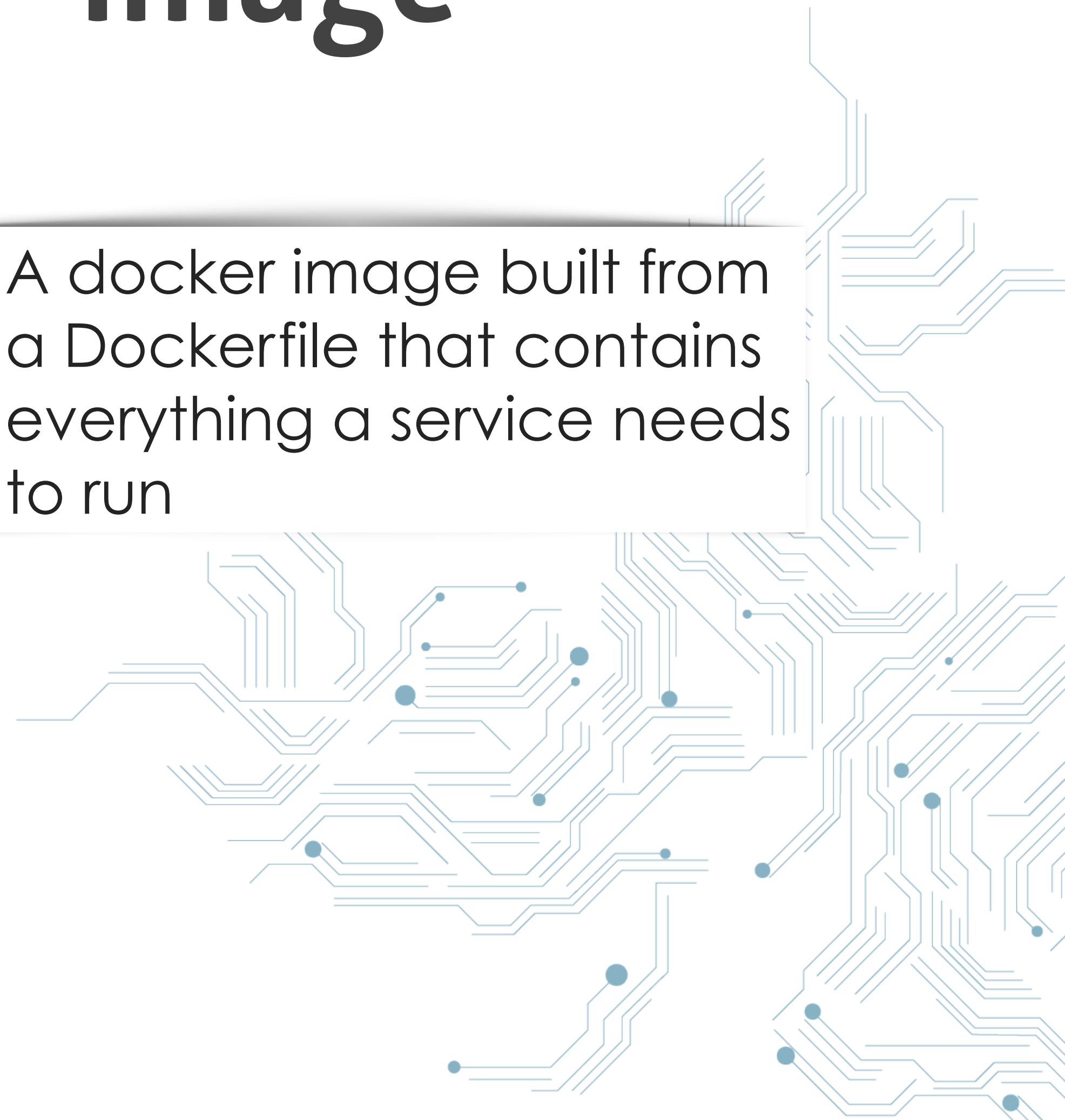


# Kubernetes Cluster



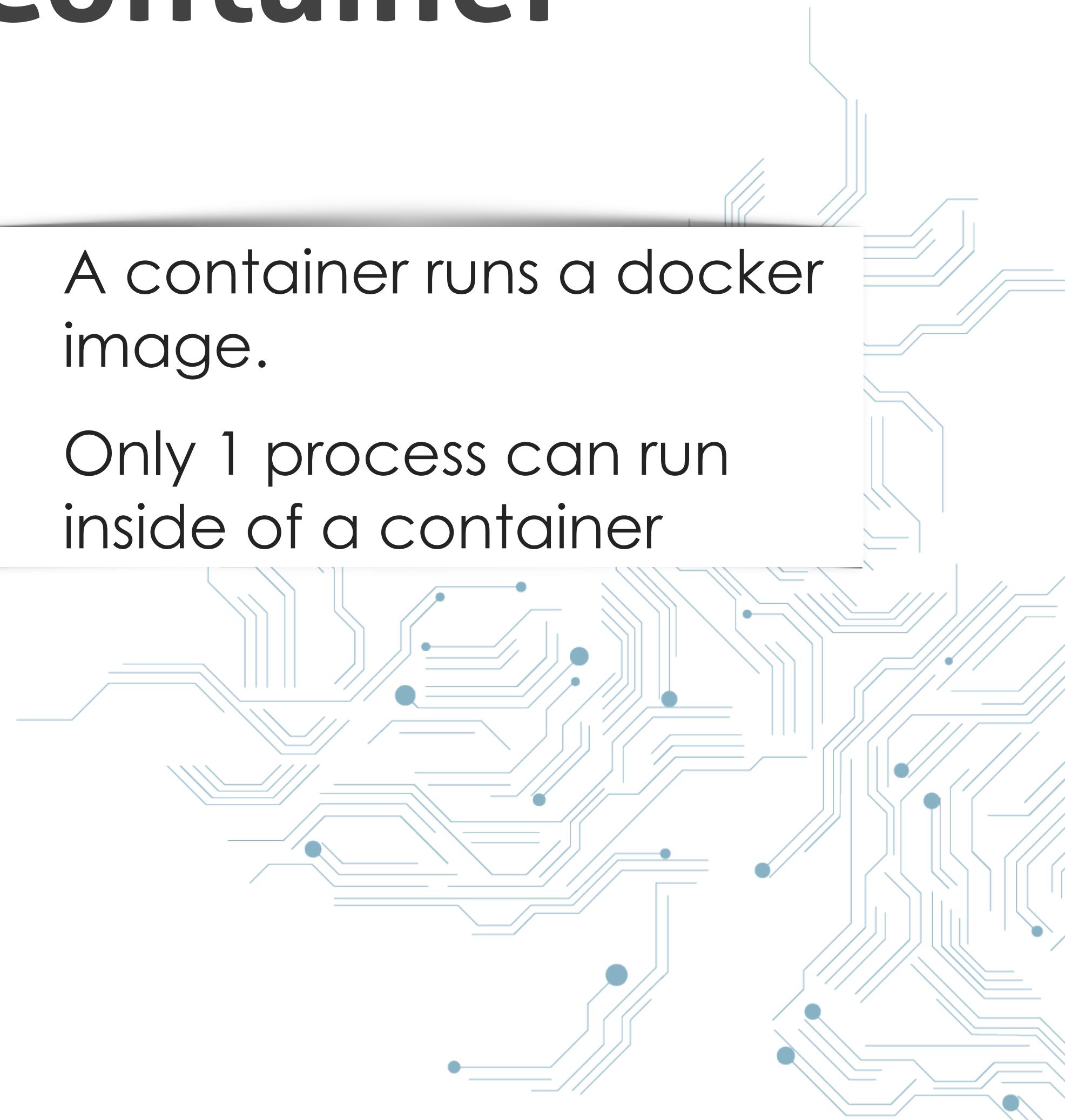
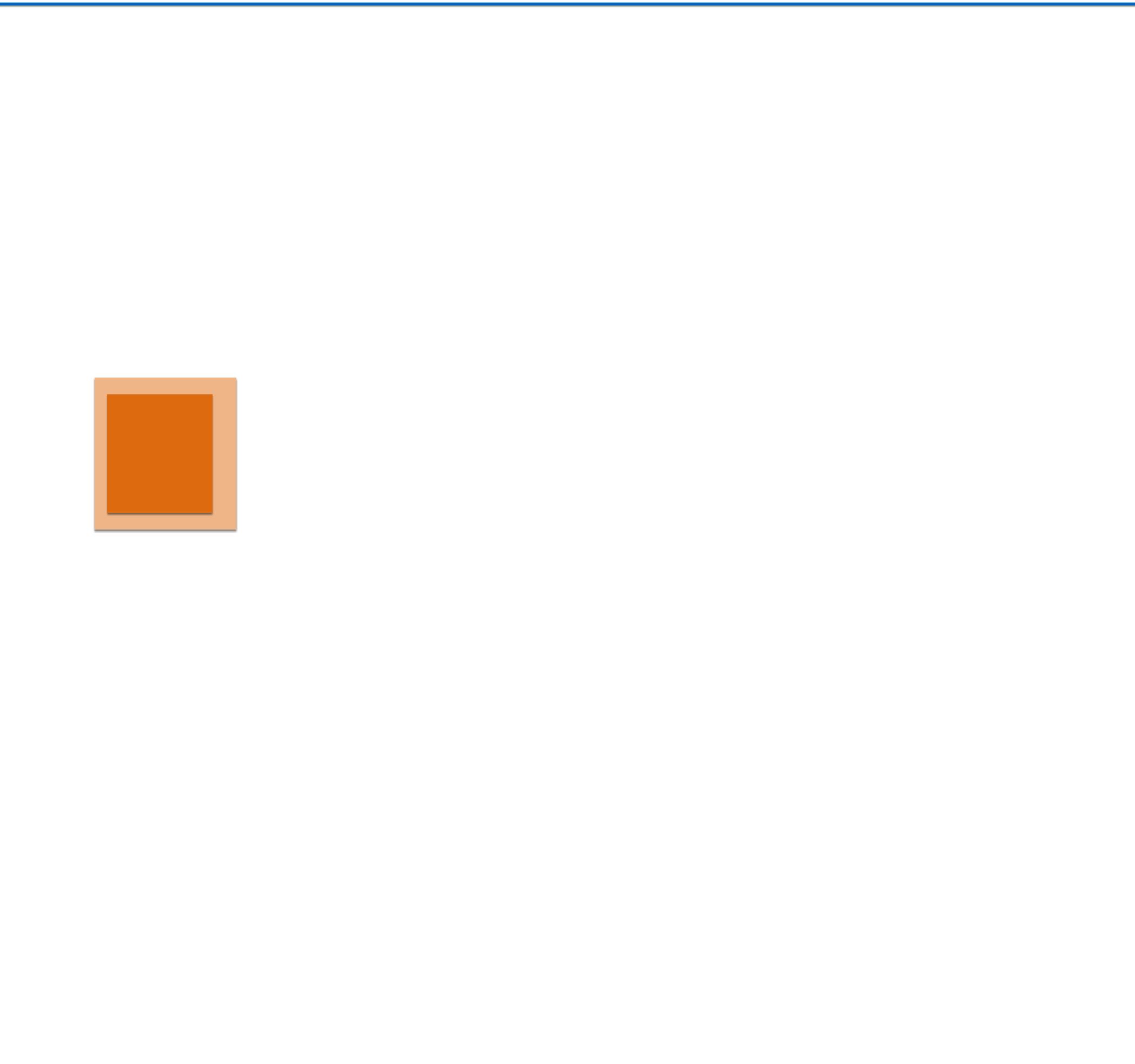
# Image

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

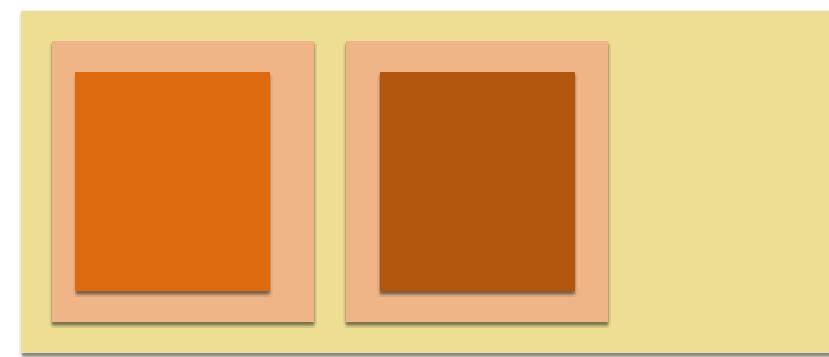


# Container

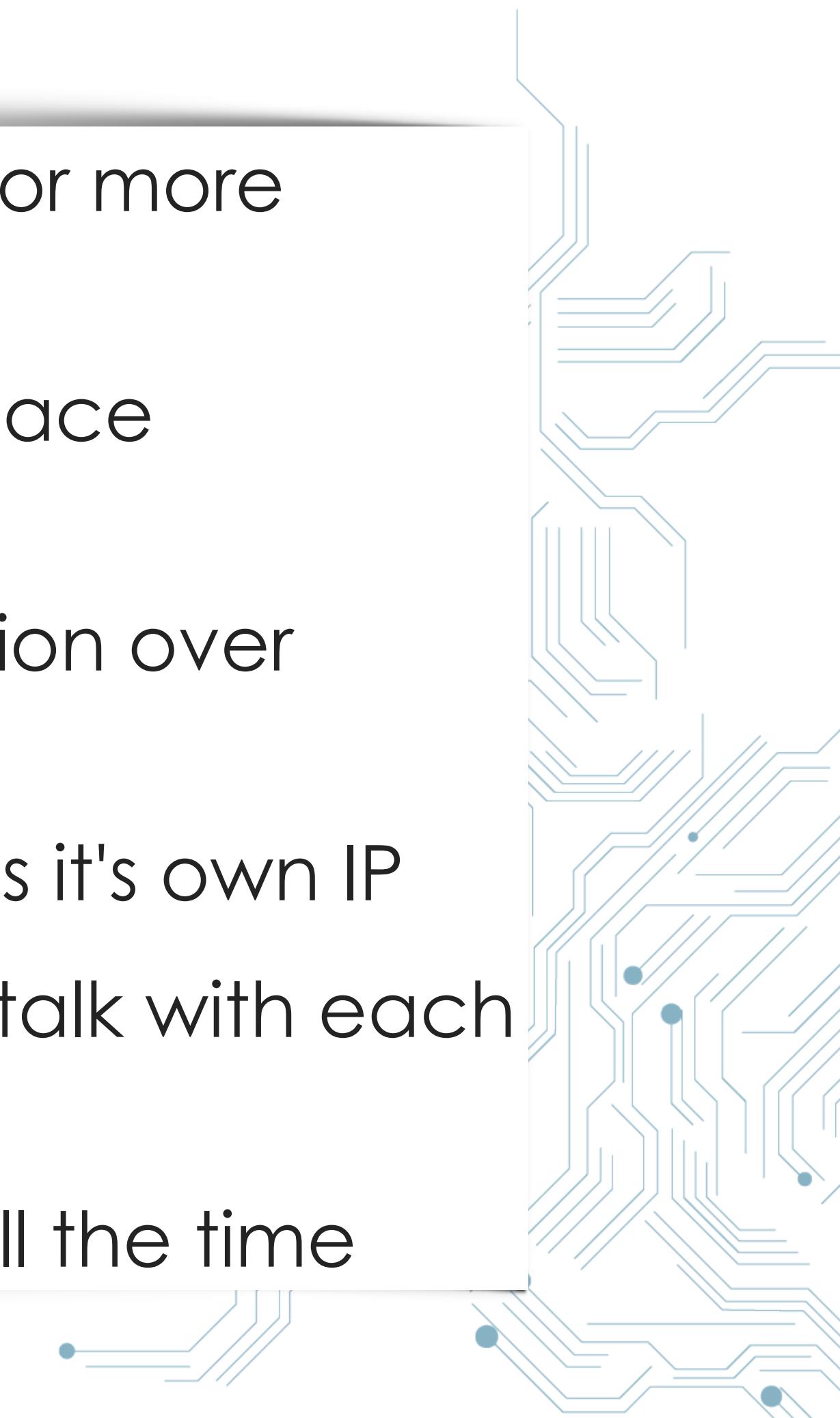
- A container runs a docker image.
- Only 1 process can run inside of a container



# Pod

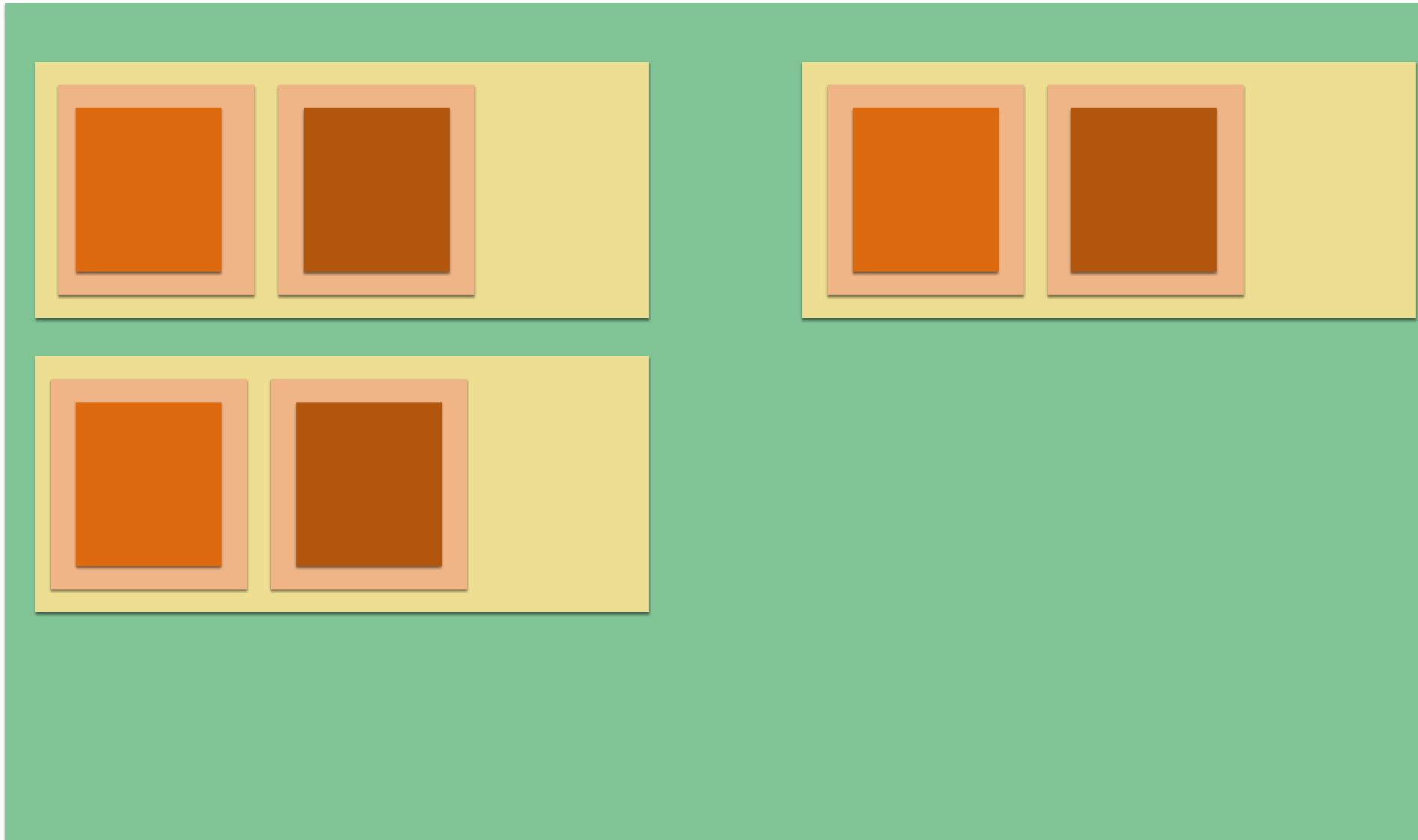


- A group of 1 or more containers
- Same port space
- Within a Pod:  
communication over localhost
- Every Pod has it's own IP
- All Pods can talk with each other
- IPs change all the time



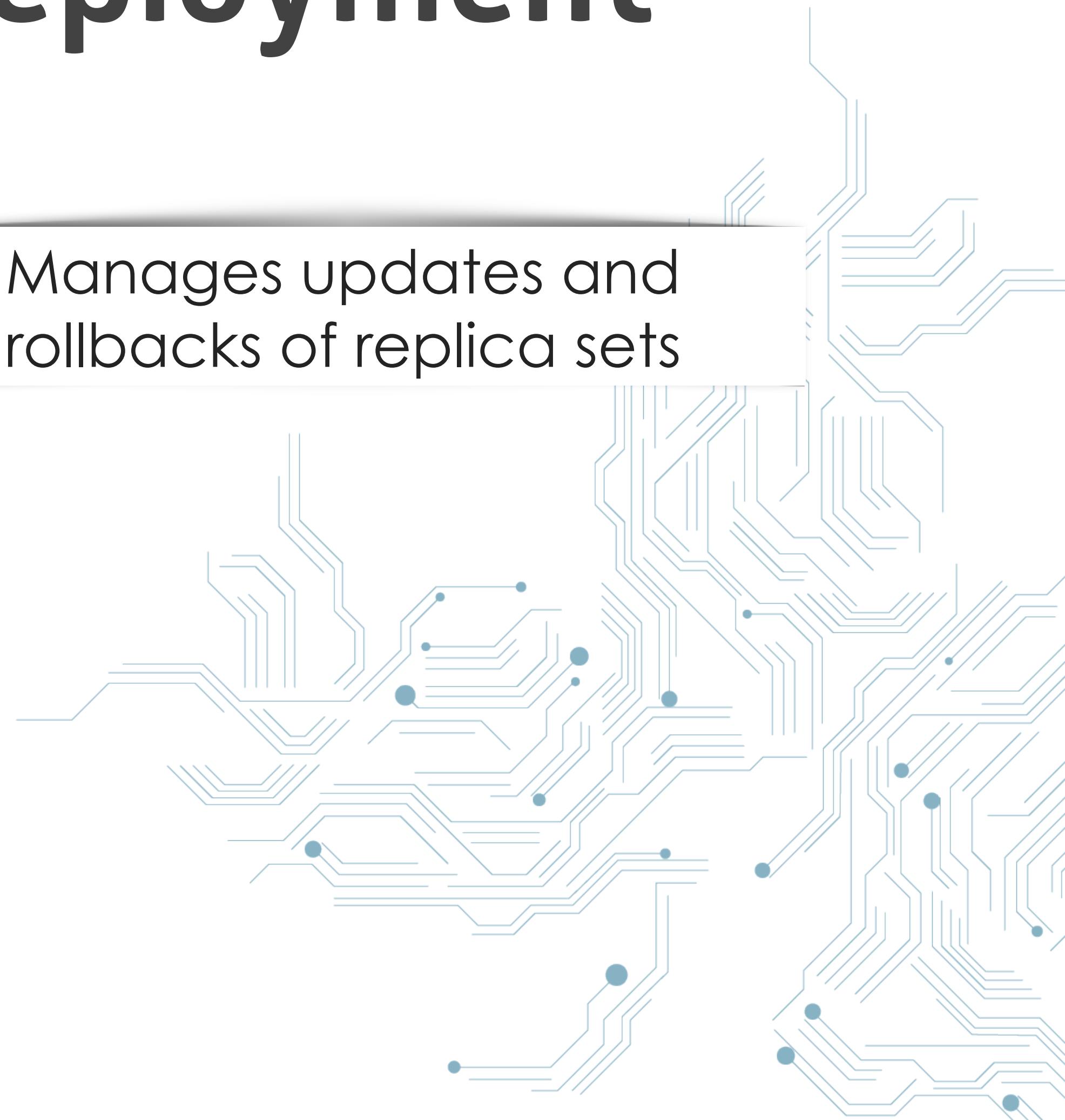
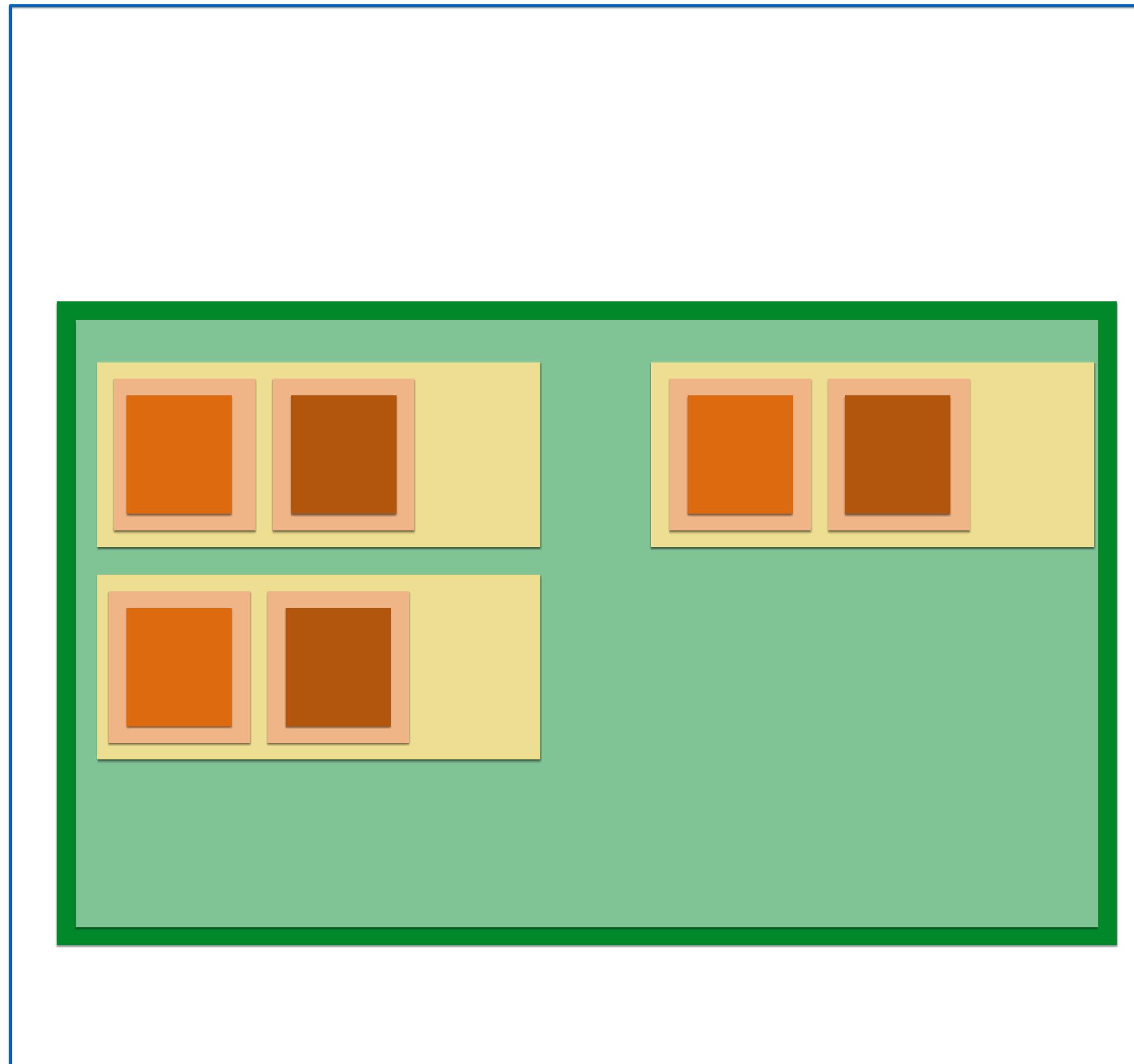
# Replica Set

- 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

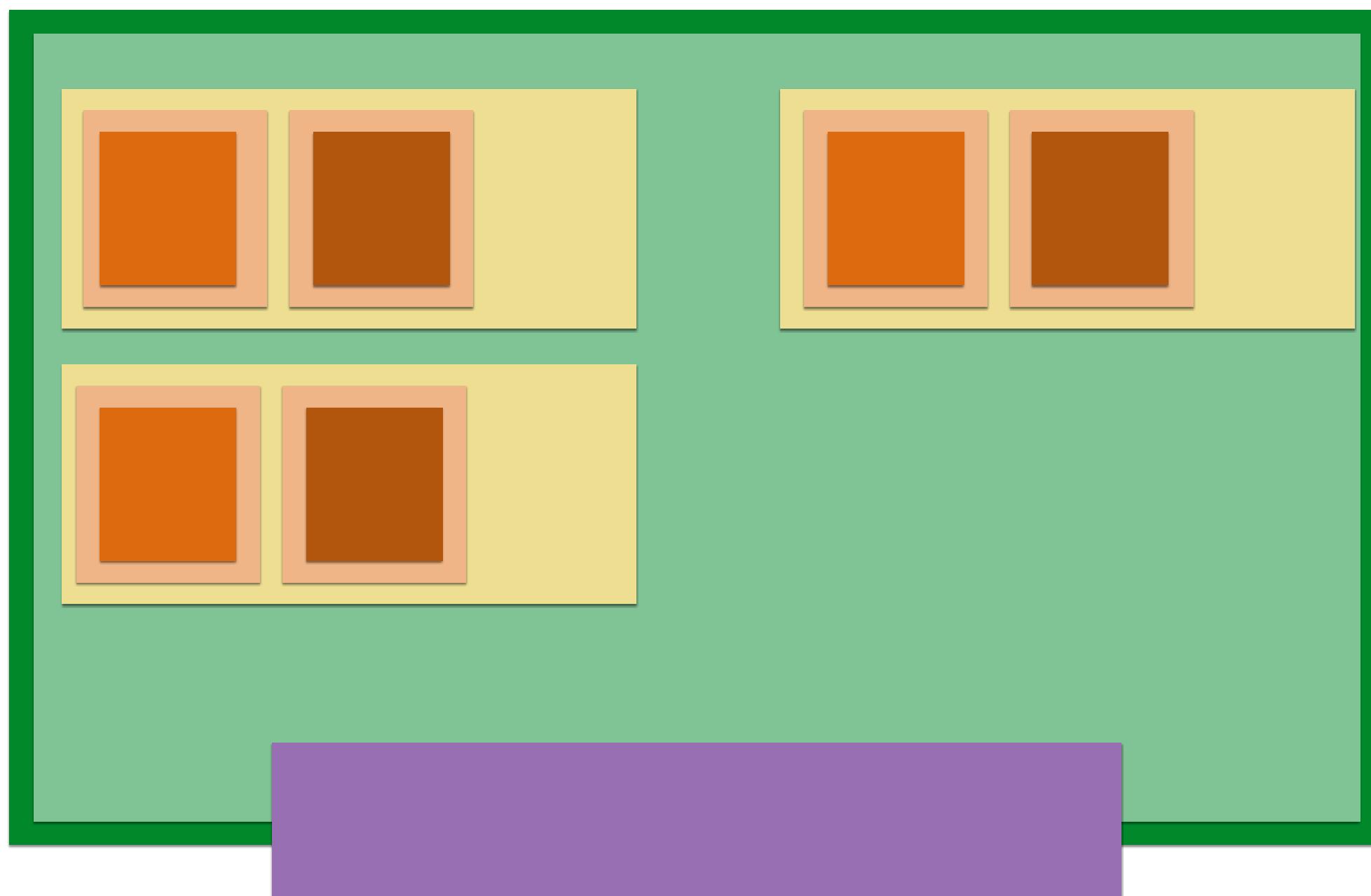


# Deployment

- Manages updates and rollbacks of replica sets

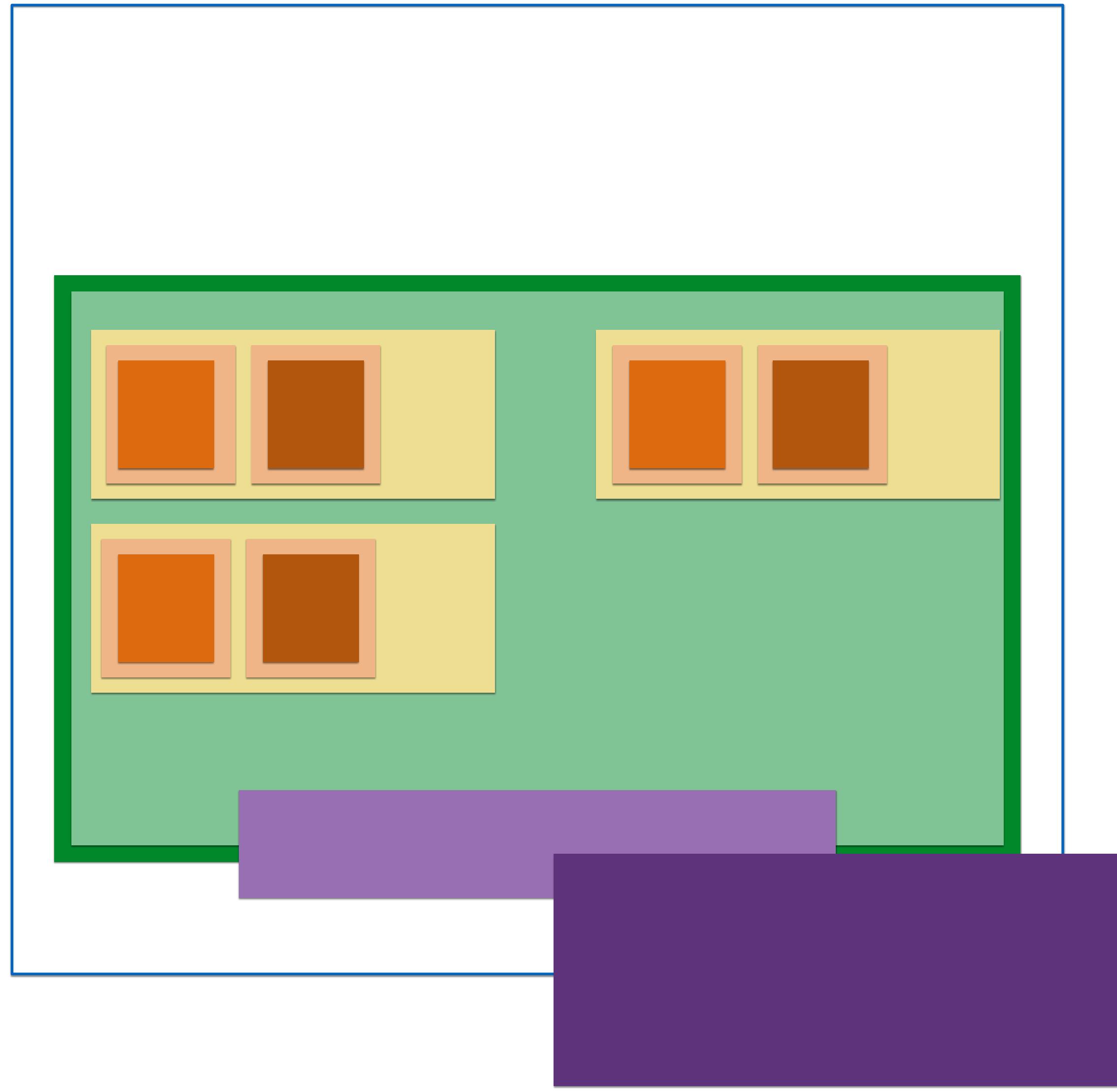


# Service

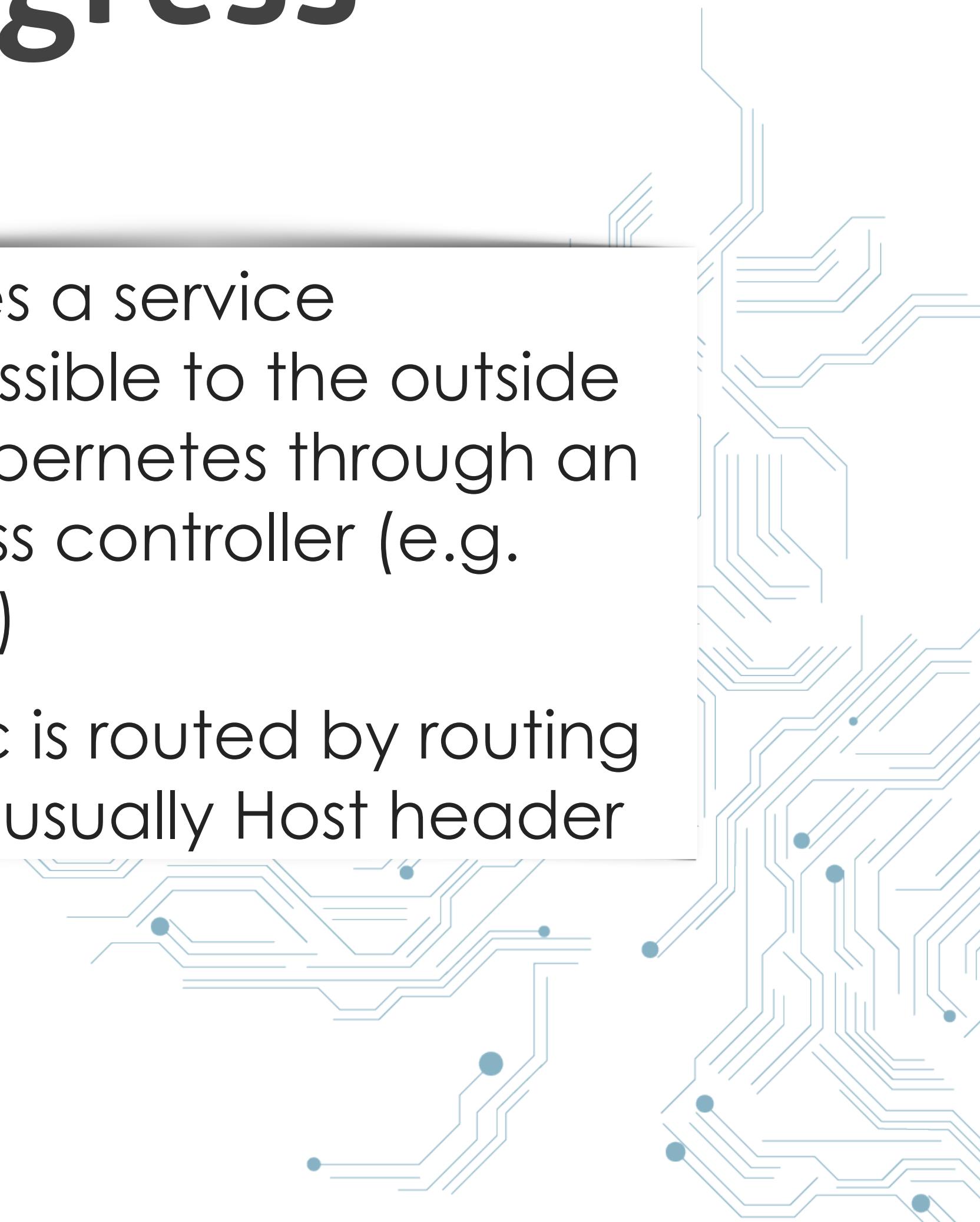


- Internal LoadBalancer
- Makes all pods matching a set of labels accessible through a stable, internal IP address
- You can attach external IP address through an cloud LoadBalancer

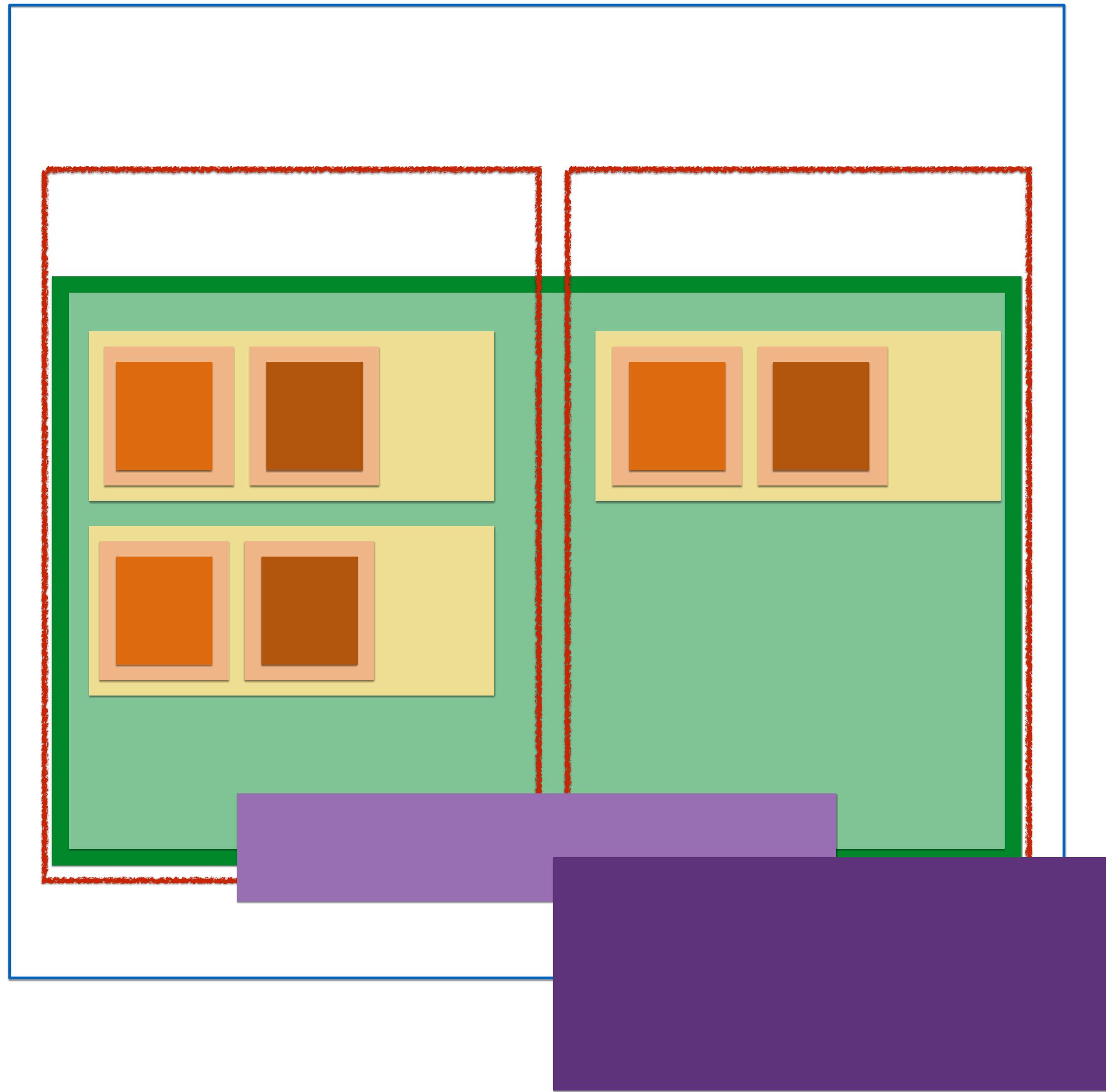
# Ingress



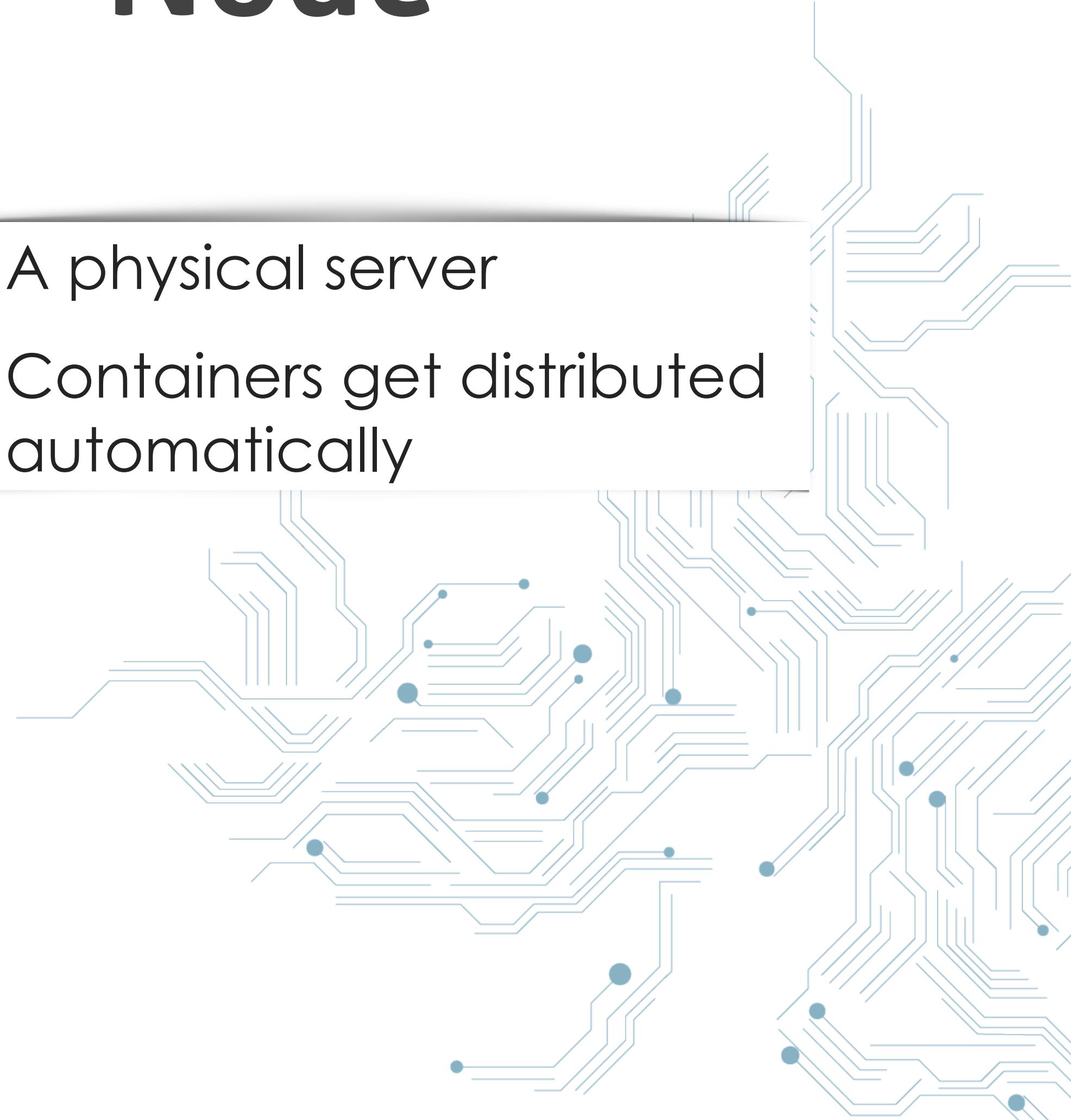
- Makes a service accessible to the outside of Kubernetes through an ingress controller (e.g. nginx)
- Traffic is routed by routing rules, usually Host header



# Node

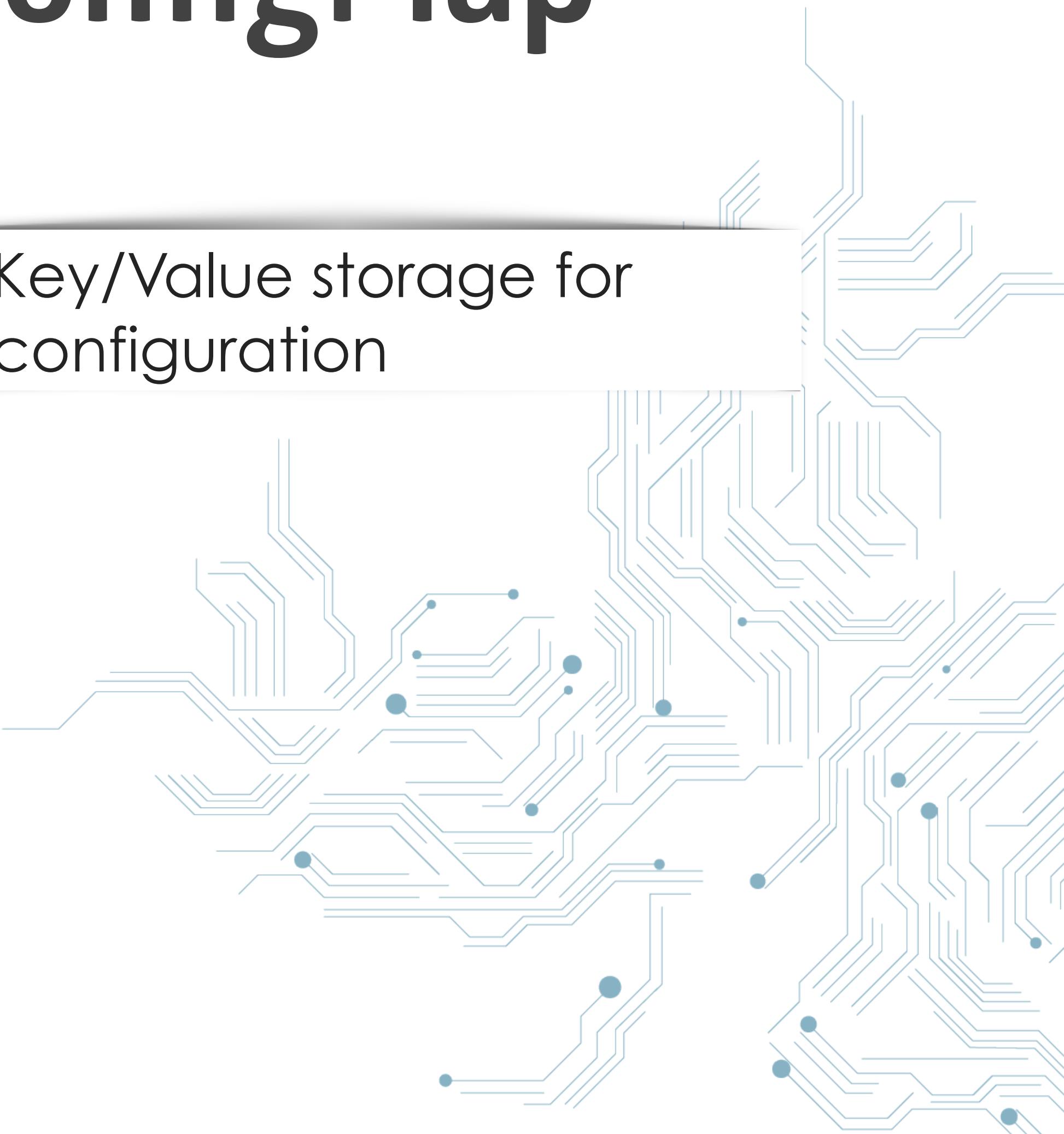
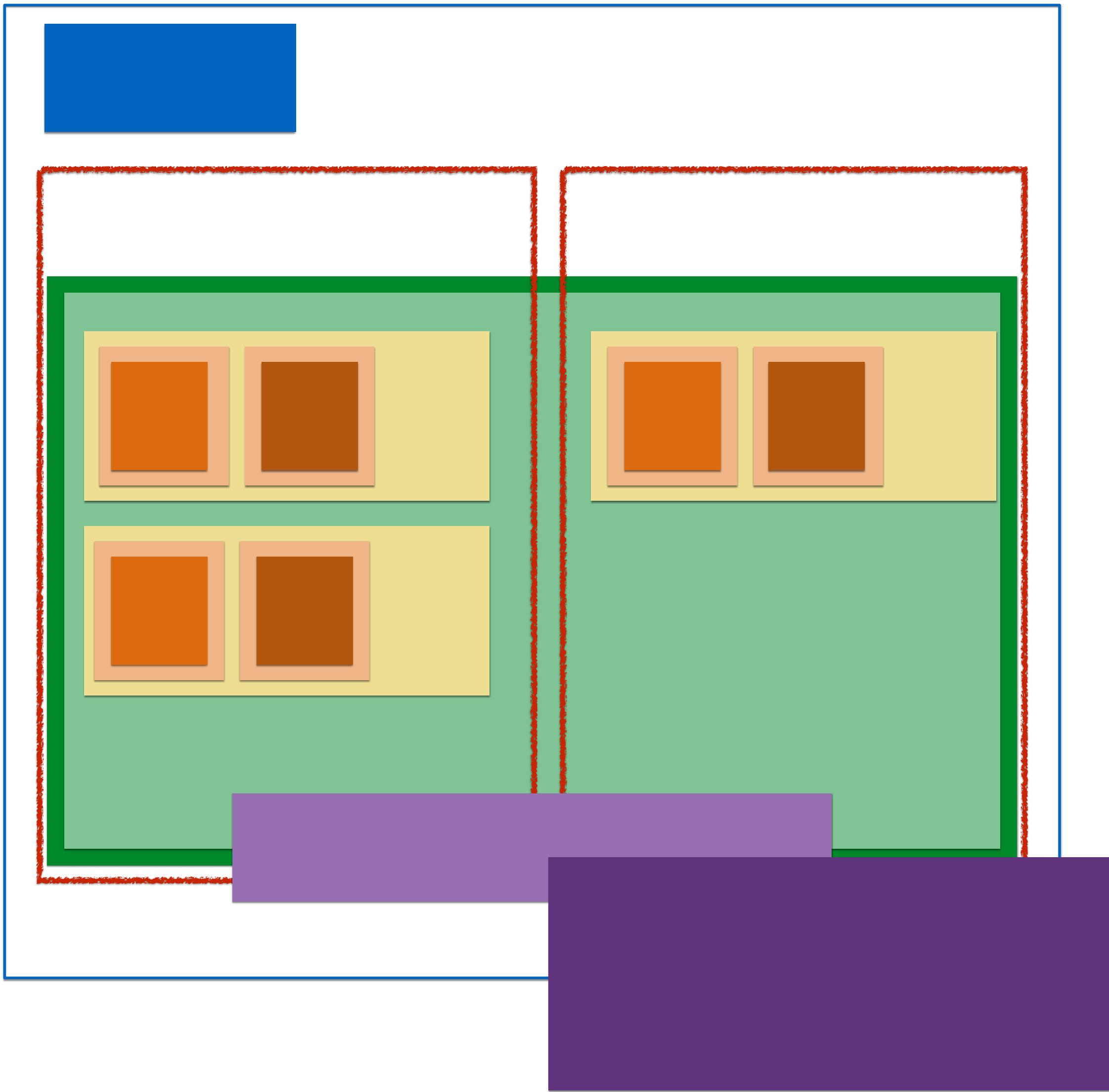


- A physical server
- Containers get distributed automatically



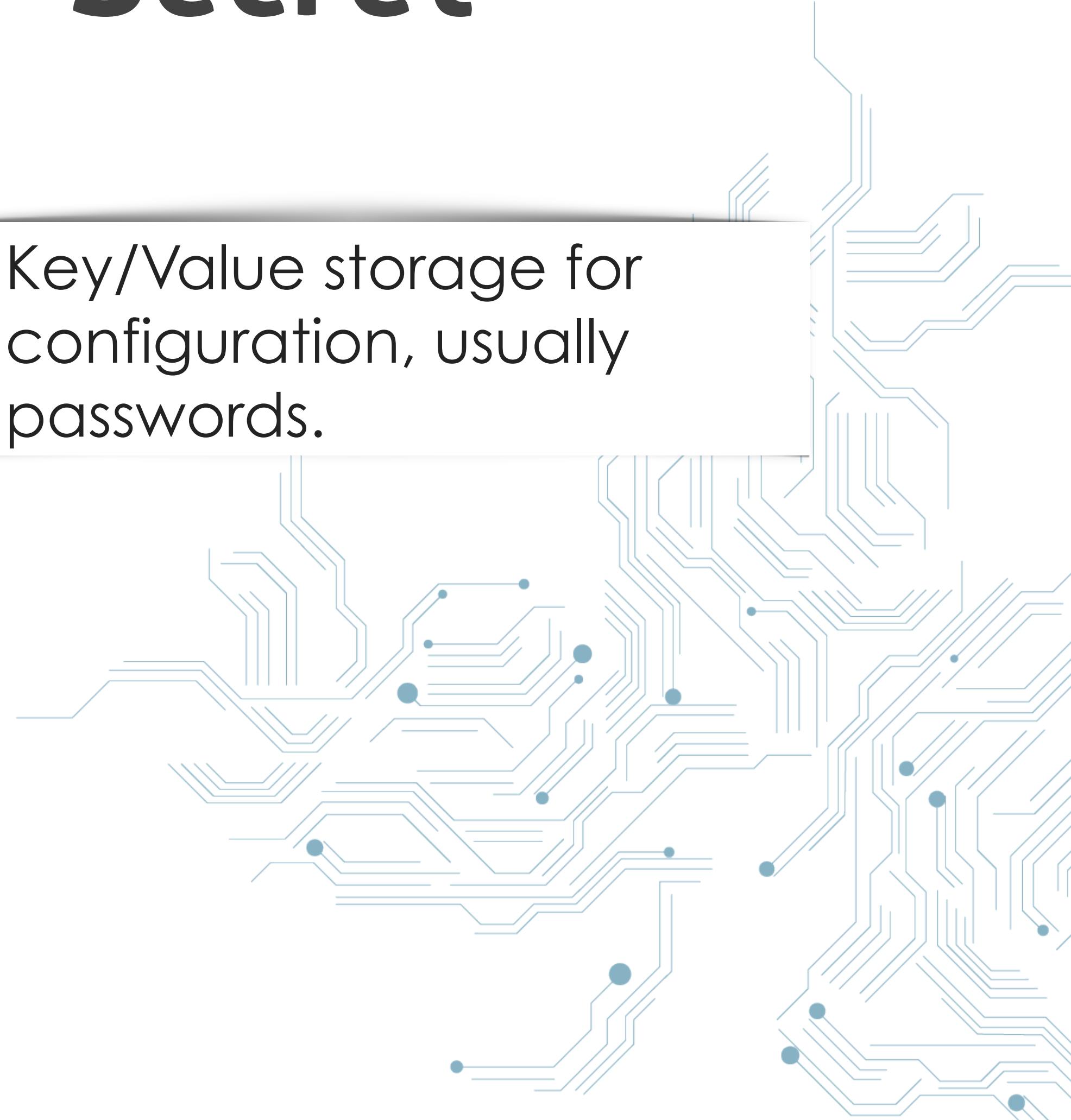
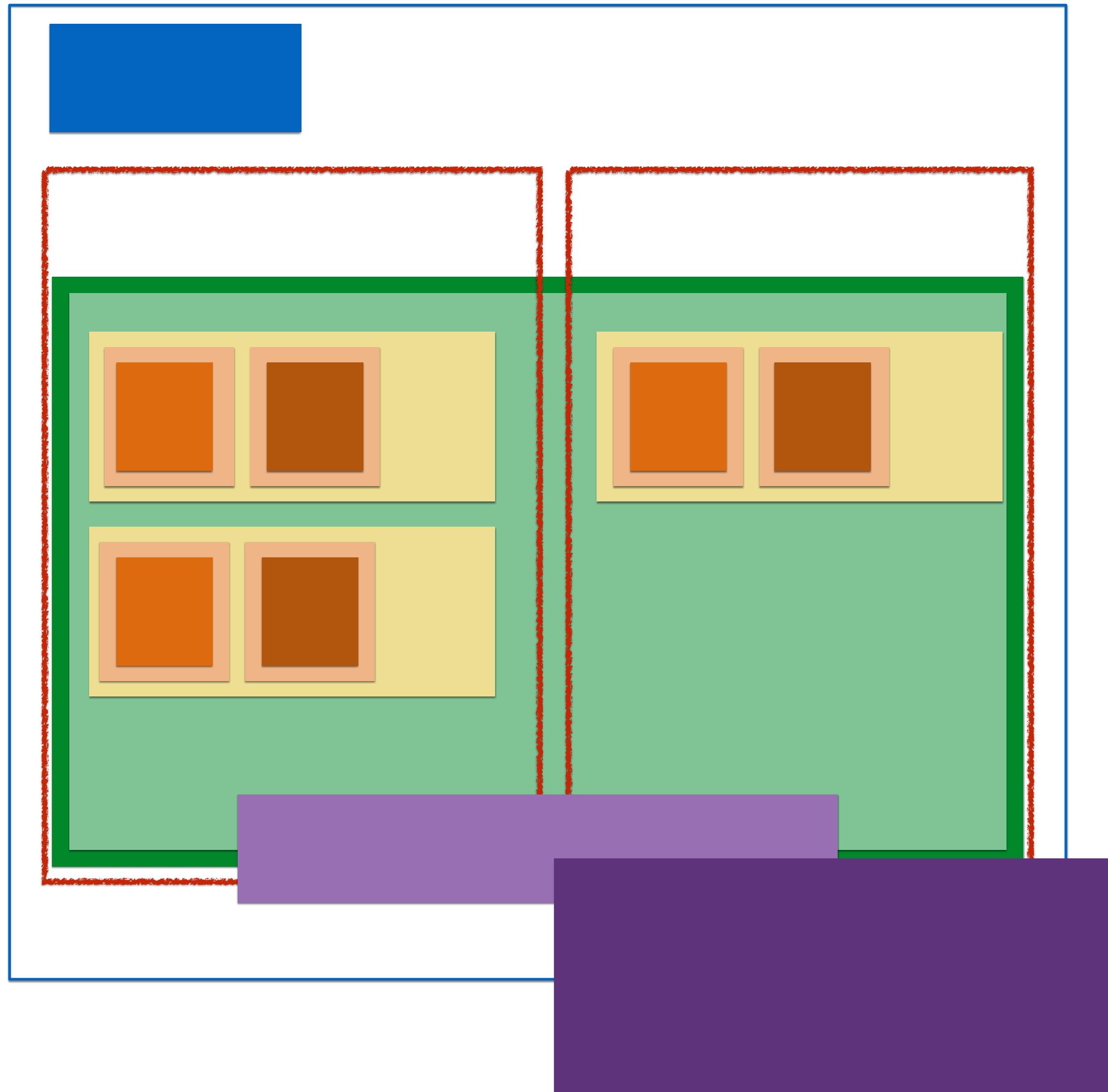
# ConfigMap

- Key/Value storage for configuration

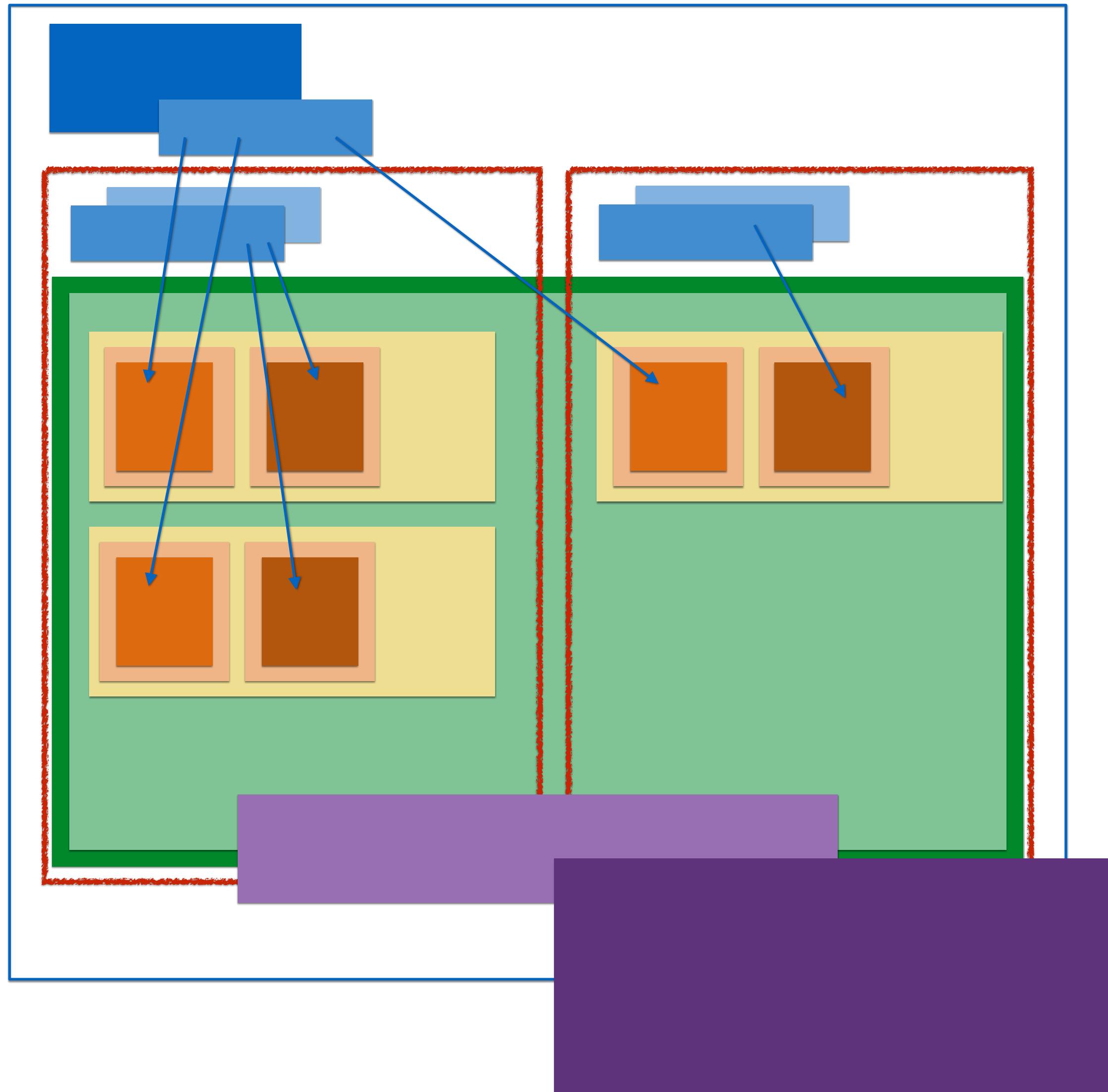


# Secret

- Key/Value storage for configuration, usually passwords.



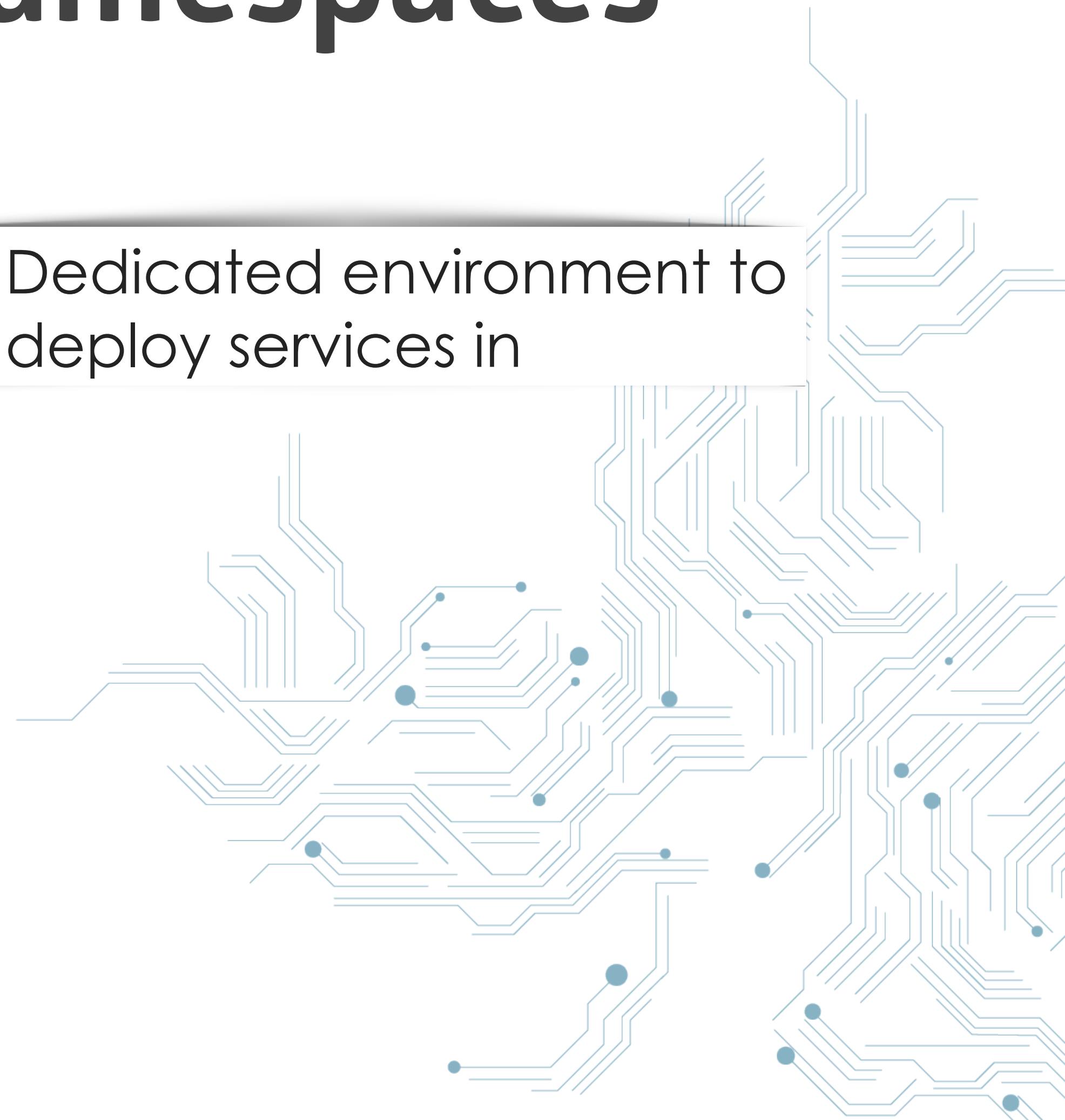
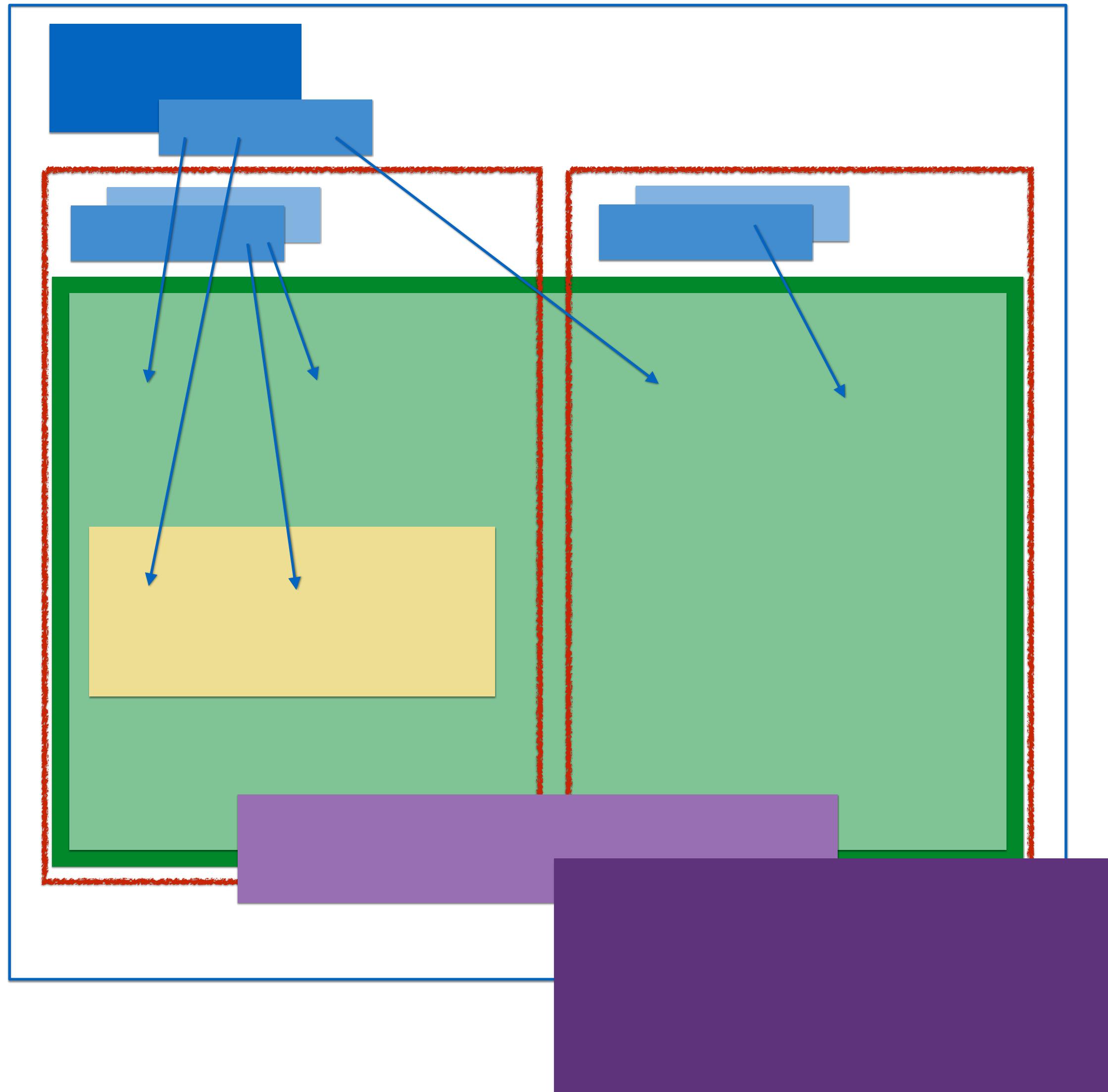
# Volumes



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

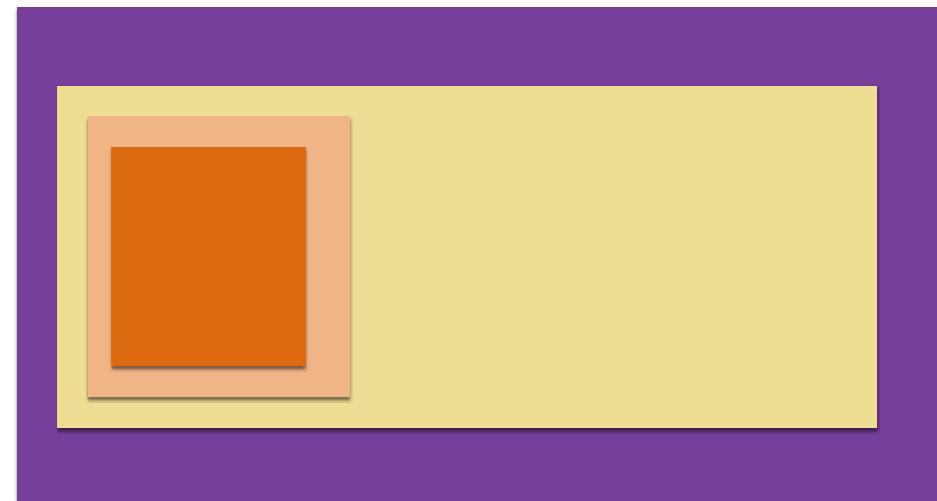
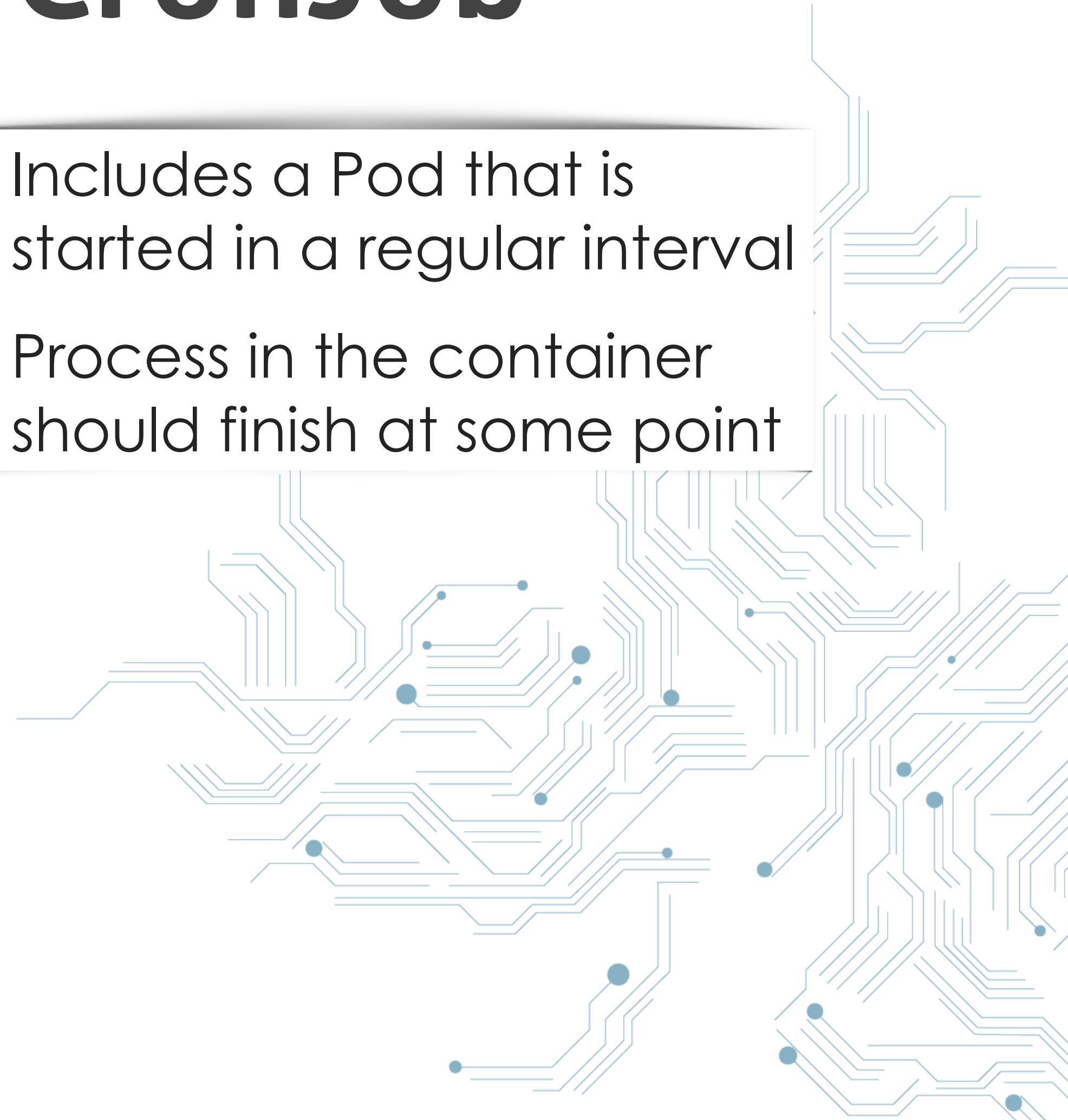
# Namespaces

- Dedicated environment to deploy services in



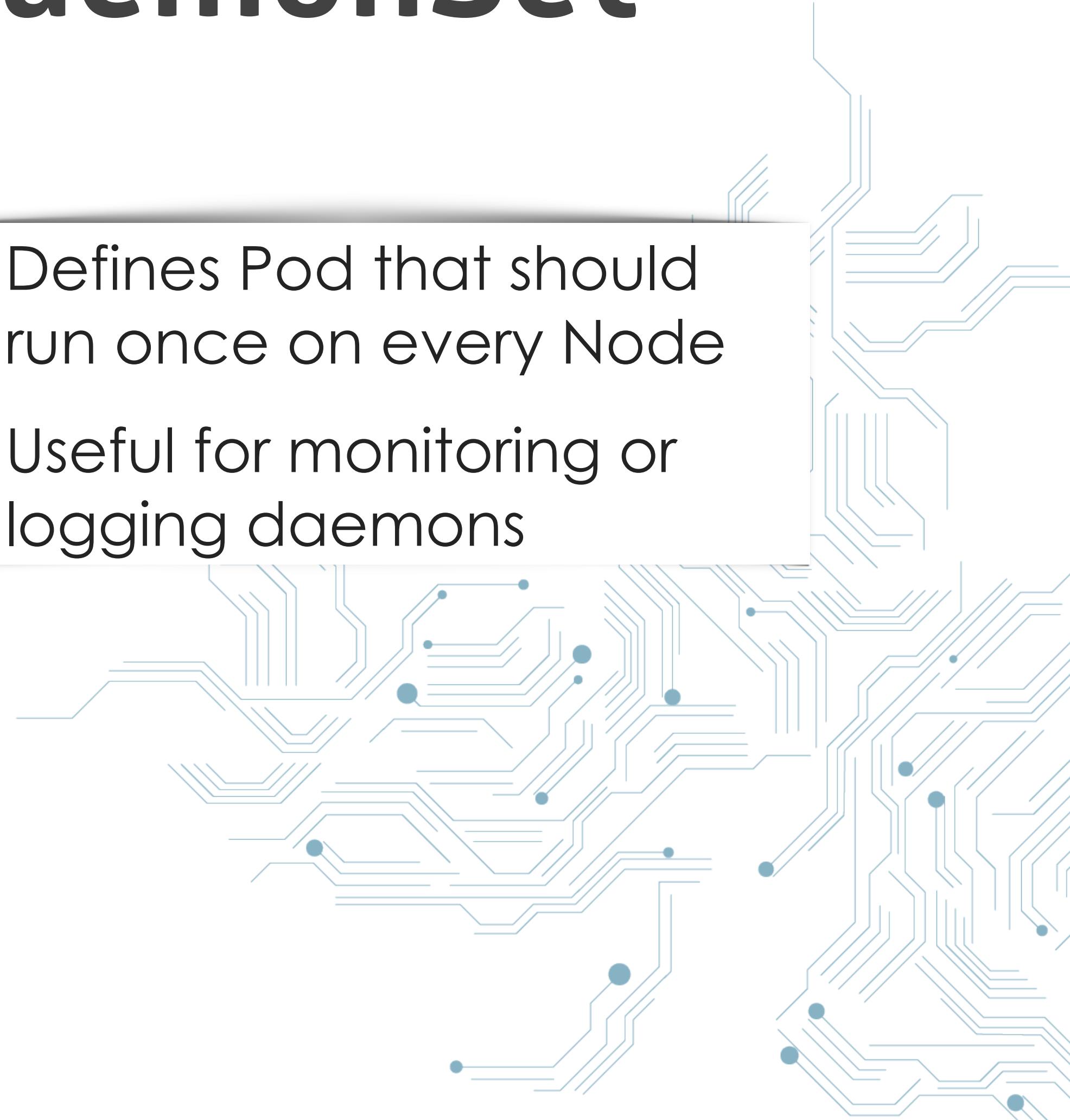
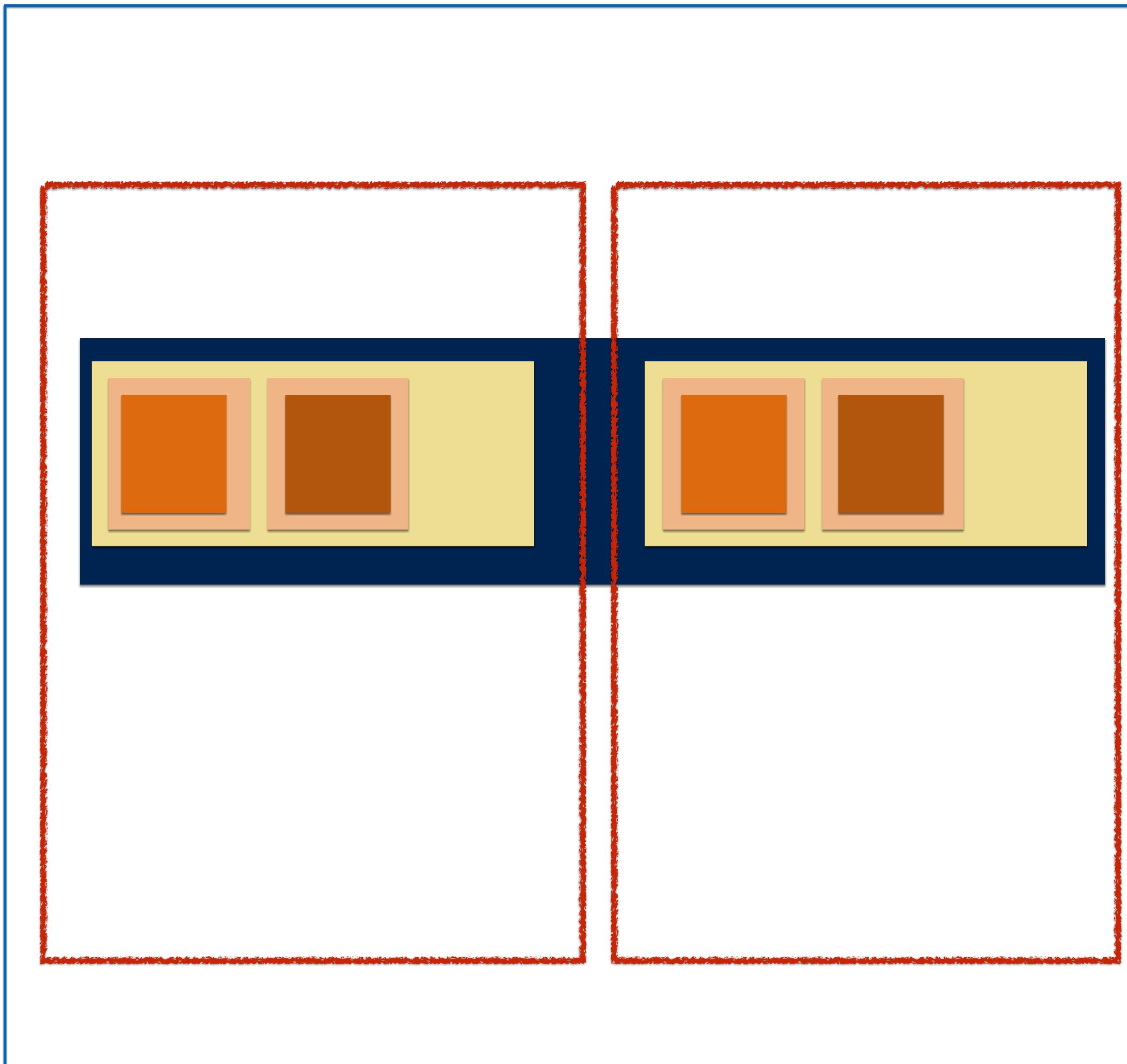
# CronJob

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

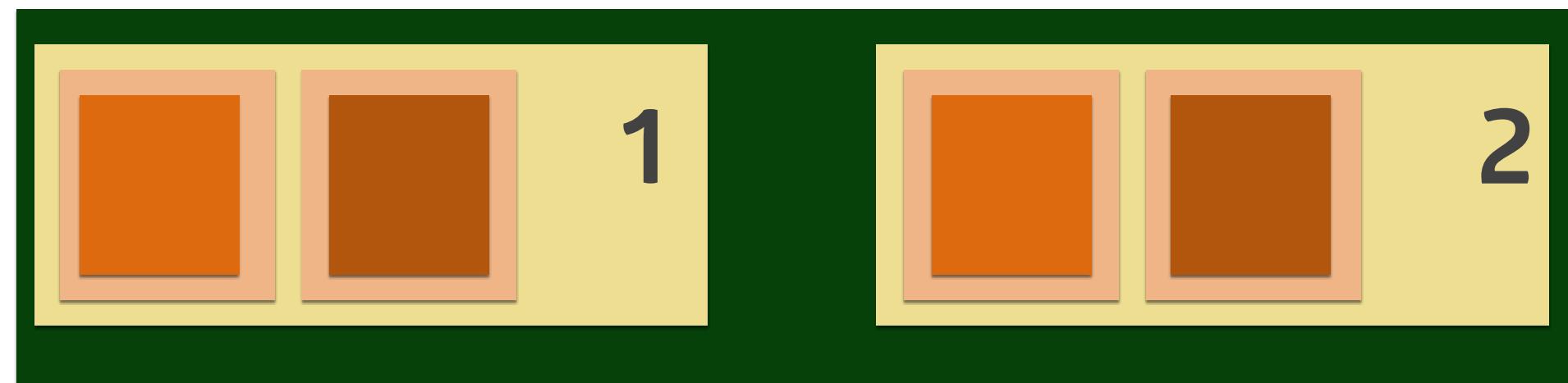


# DaemonSet

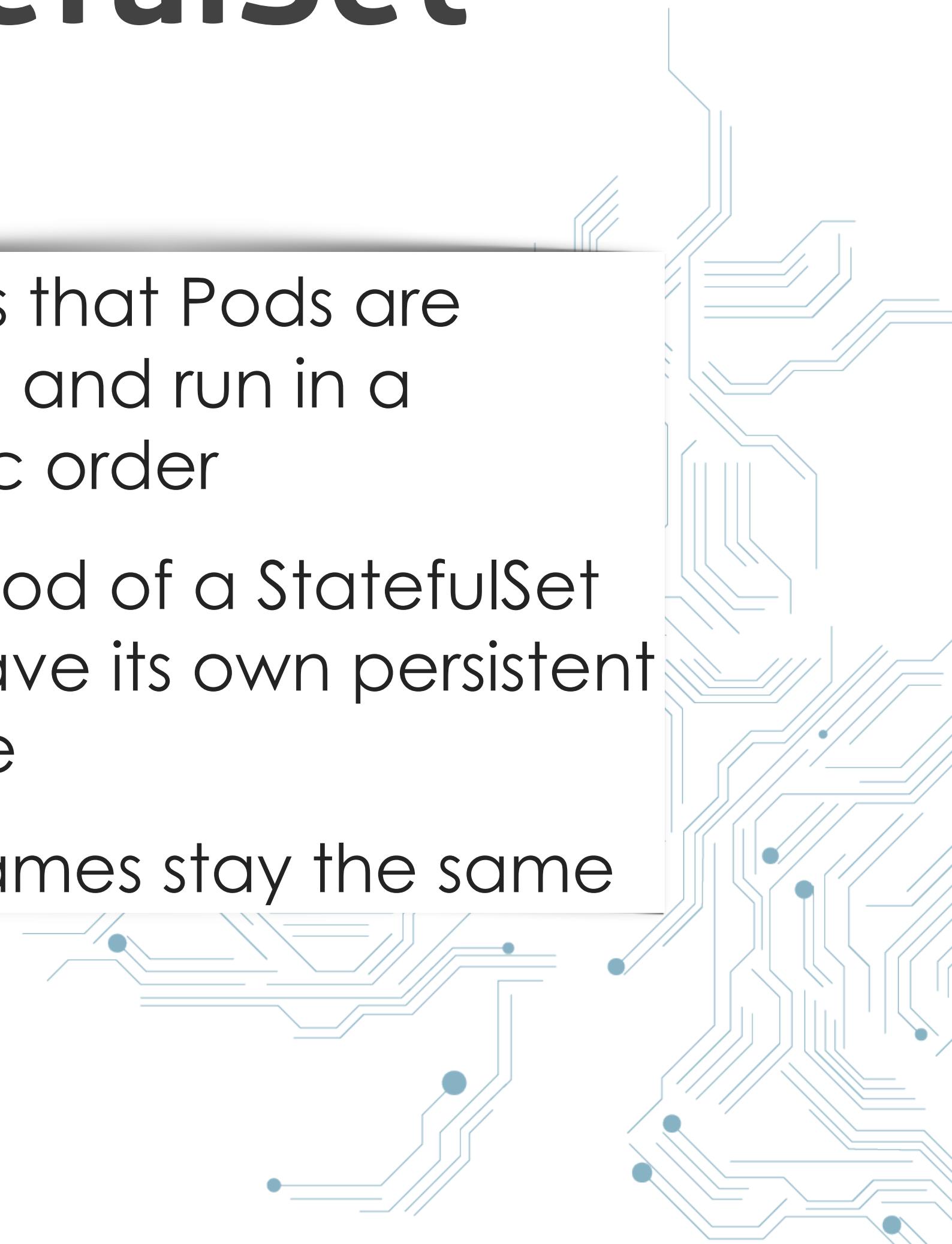
- Defines Pod that should run once on every Node
- Useful for monitoring or logging daemons



# StatefulSet



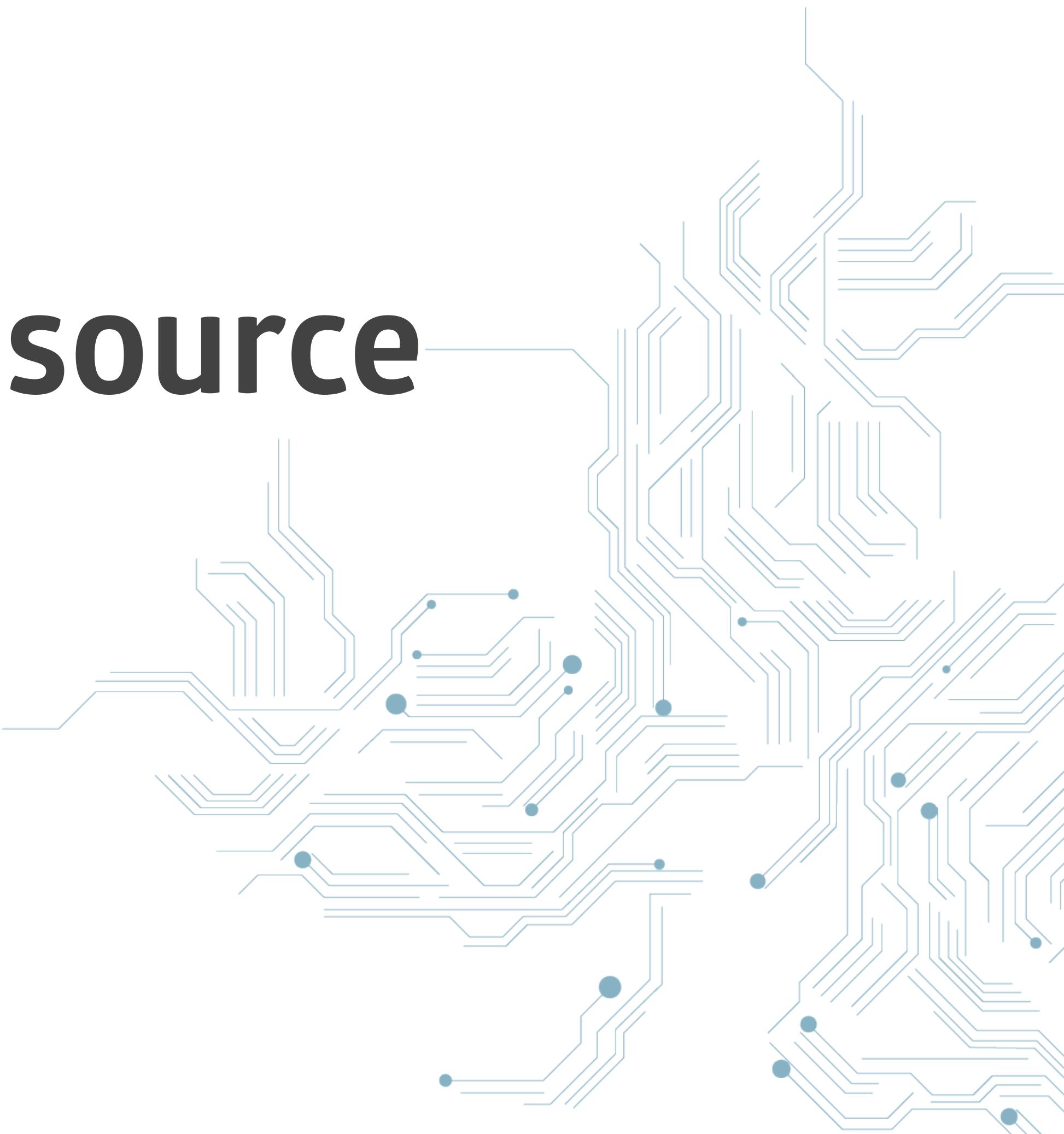
- Ensures that Pods are started and run in a specific order
- Each Pod of a StatefulSet can have its own persistent volume
- Pod names stay the same



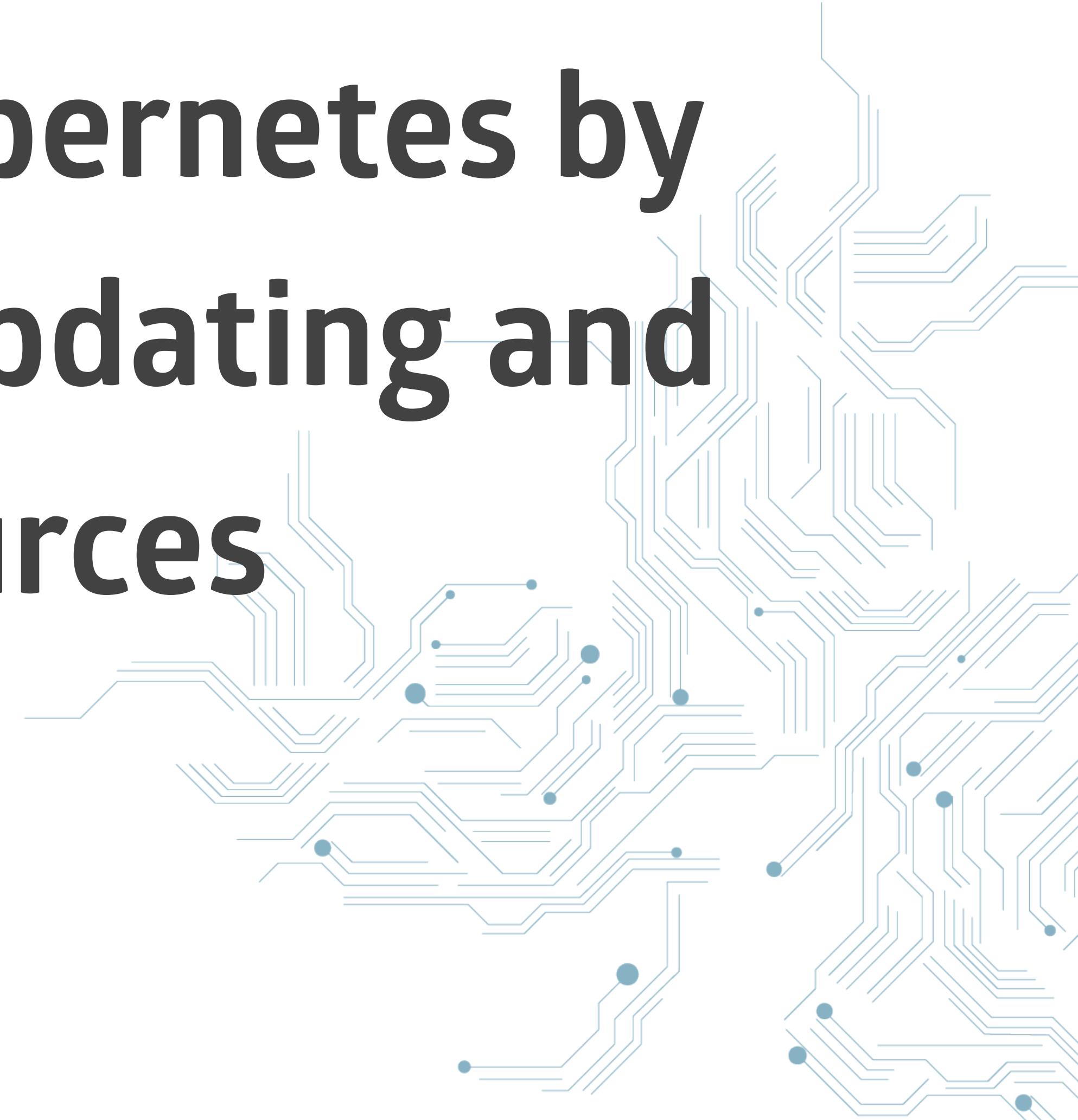
...



# Everything is a resource



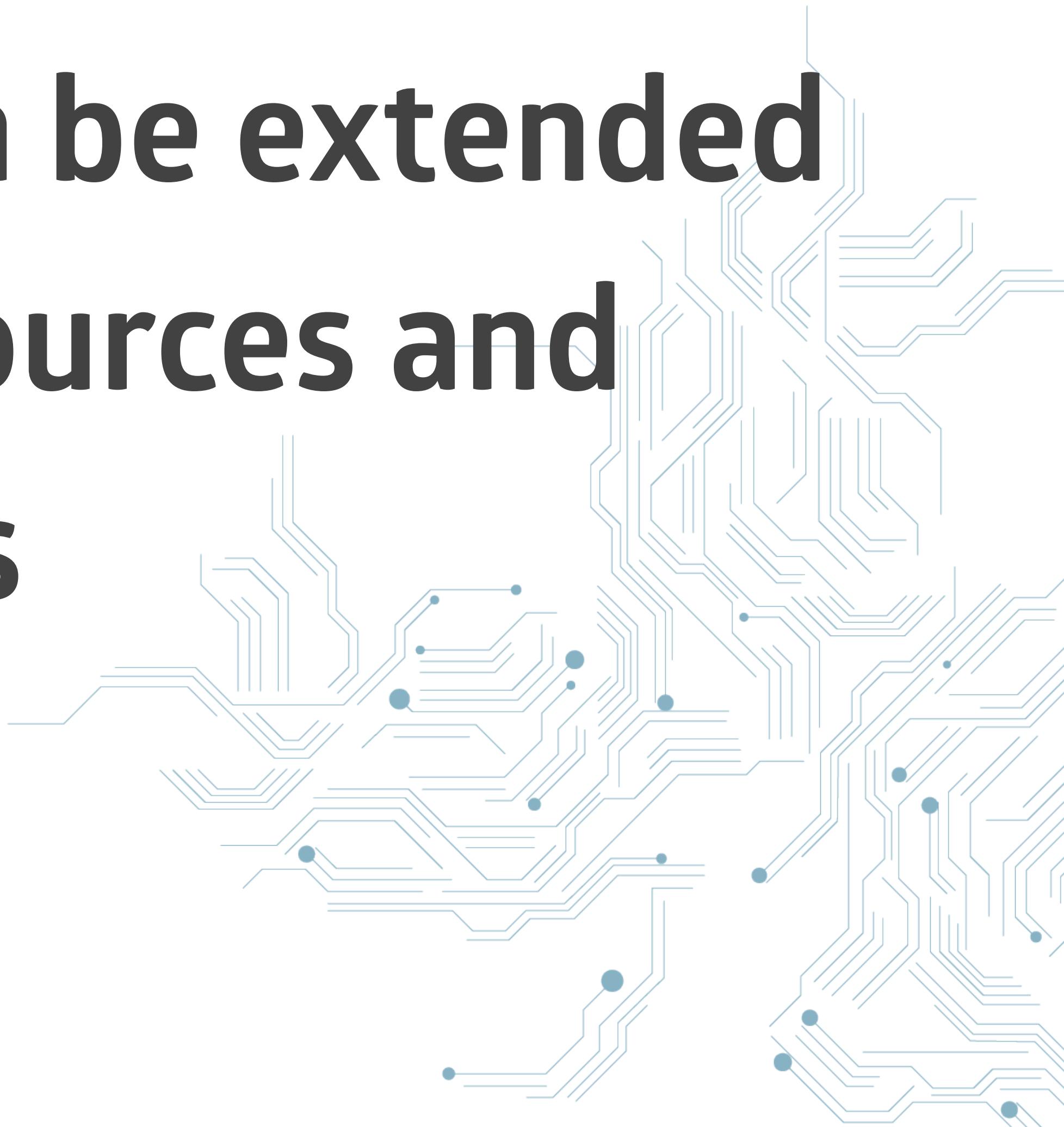
You interact with Kubernetes by  
creating, receiving, updating and  
deleting resources



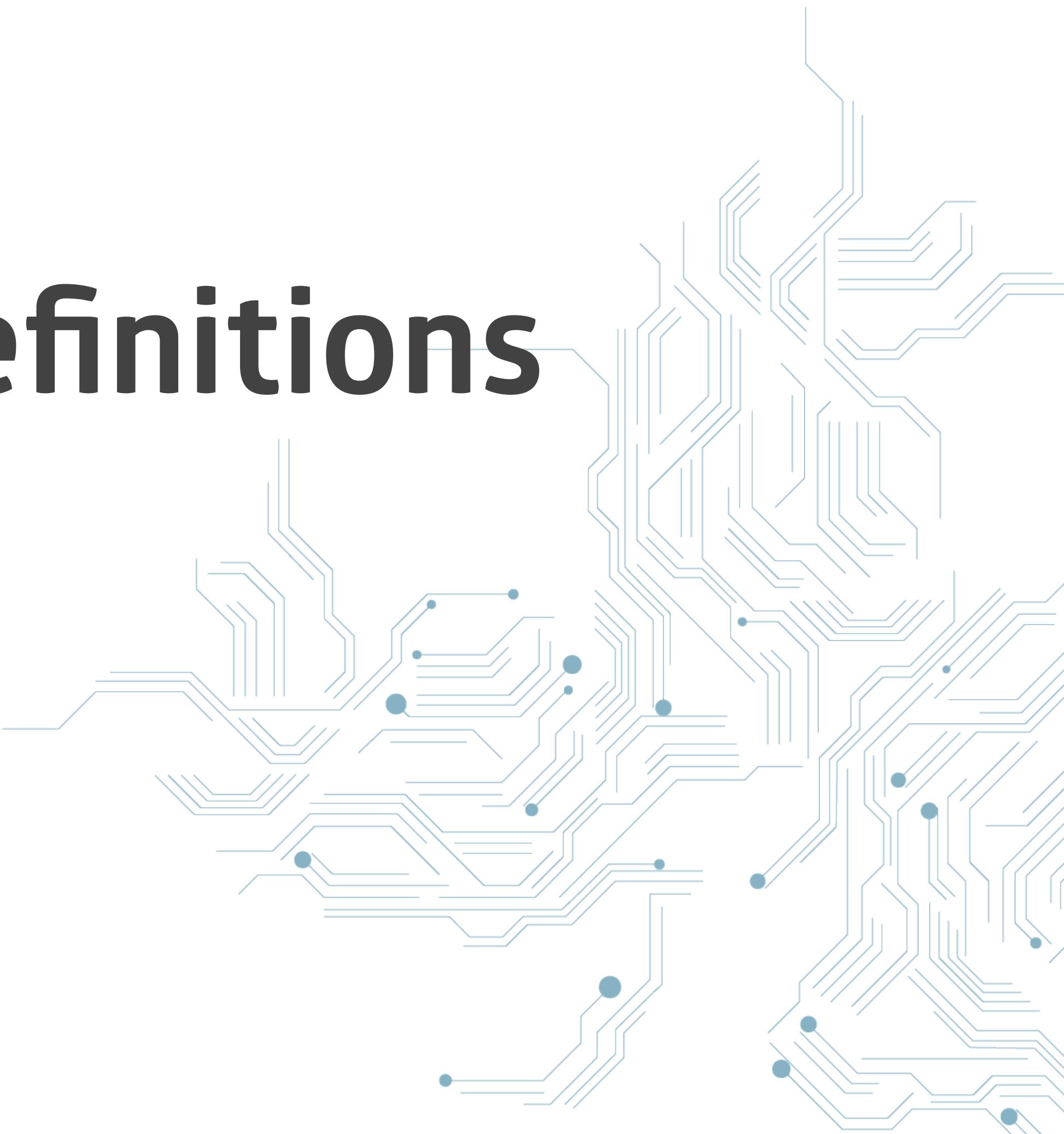
**Kubernetes has controllers to listen  
on these interactions and get the  
cluster in the desired state.**



**The Kubernetes API can be extended  
with additional Resources and  
Controllers**



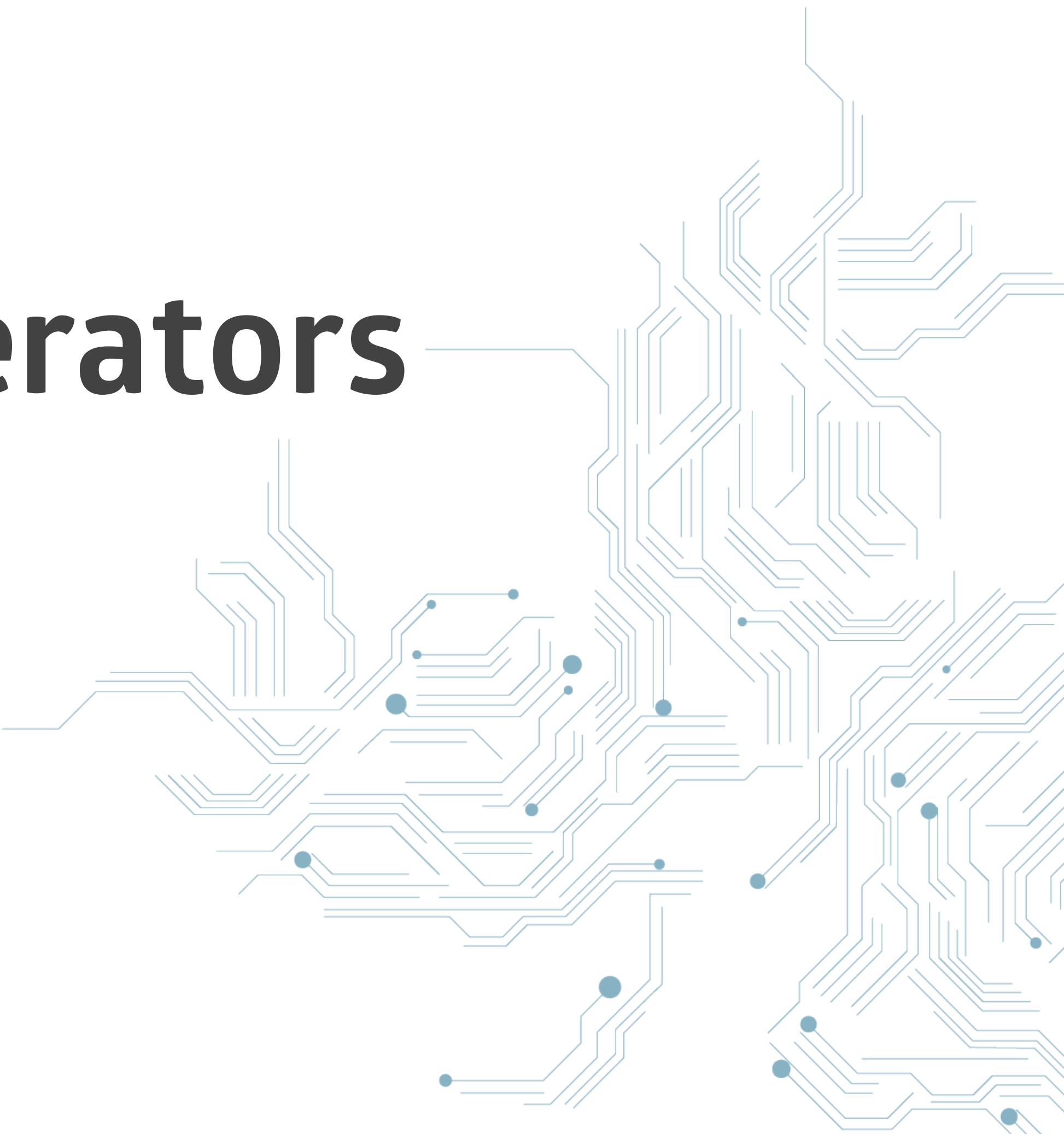
# CustomResourceDefinitions



# 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



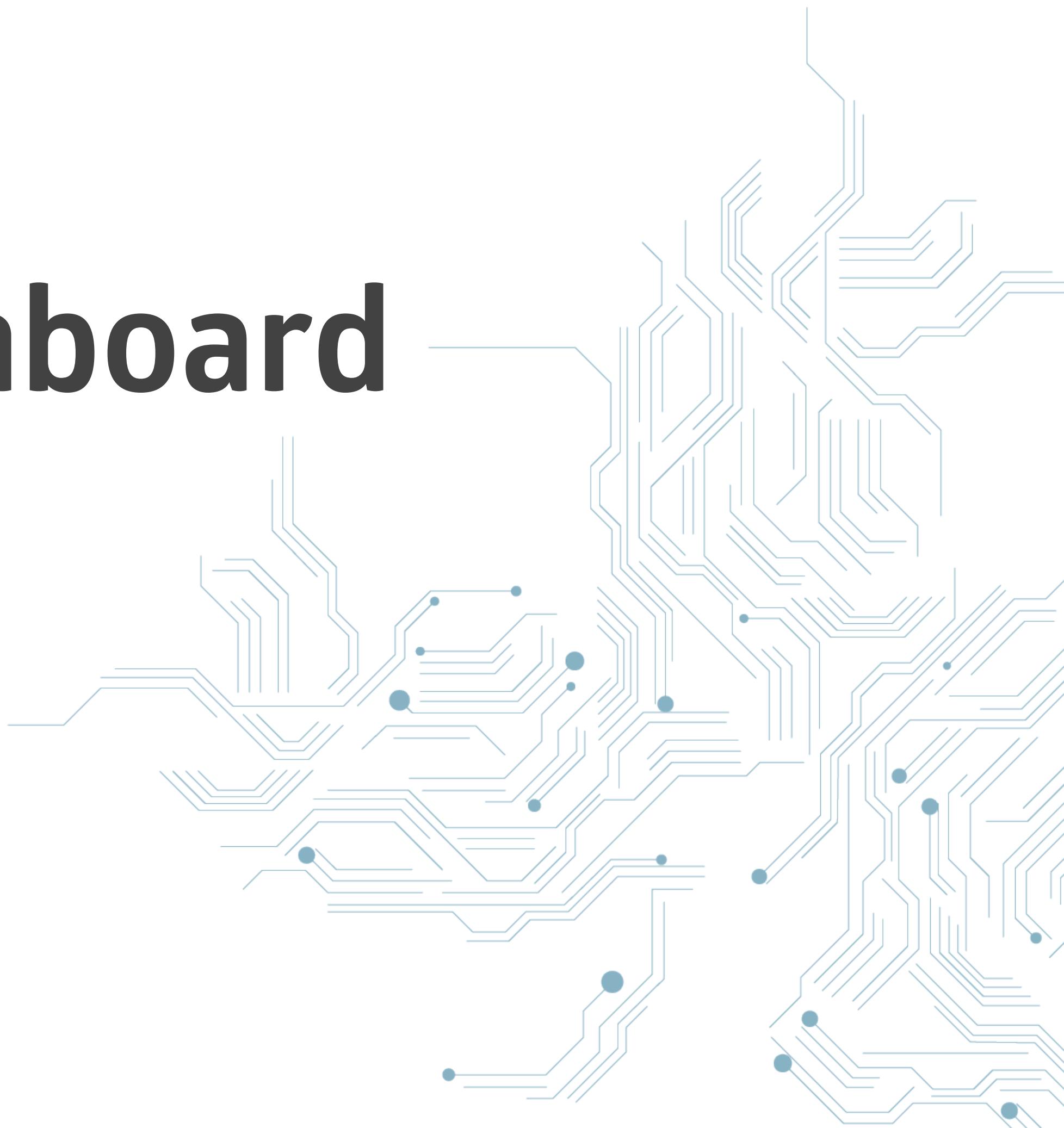
# REST API



*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": [
    {
      "status": {
        "conditions": [
          {
            "lastProbeTime": null,
            "lastTransitionTime": "2017-01-12T15:42:20Z",
            "status": "True",
            "type": "Ready"
          }
        ],
        "podIP": "10.244.1.10",
        "podIPs": [
          {
            "ip": "10.244.1.10"
          }
        ],
        "qosClass": "BestEffort"
      }
    }
  ]
}
```

# kubernetes-dashboard



## ☰ Overview

## Cluster

- Namespaces
- Nodes
- Persistent Volumes
- Roles
- Storage Classes

## Namespace

default ▾

Overview

## Workloads

- Daemon Sets
- Deployments
- Jobs
- Pods
- Replica Sets
- Replication Controllers
- Stateful Sets

## Discovery and Load Balancing

- Ingresses
- Services

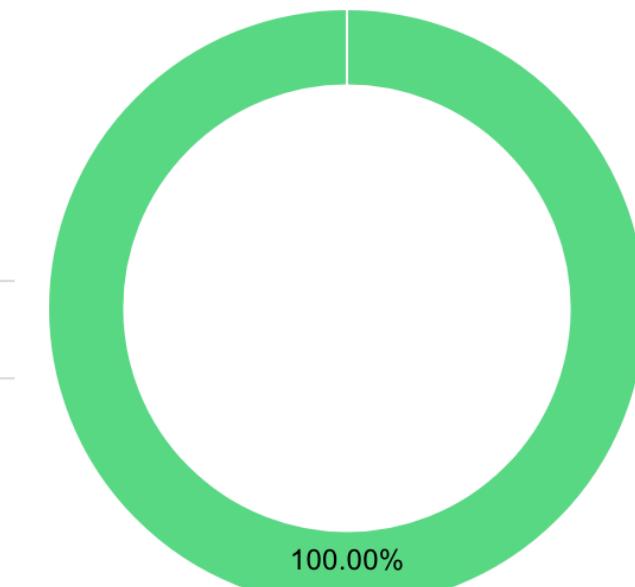
## Config and Storage

- Config Maps
- Persistent Volume Claims

## Resource Status

## Pods

- Running 6
- Pending 0
- Failed 0

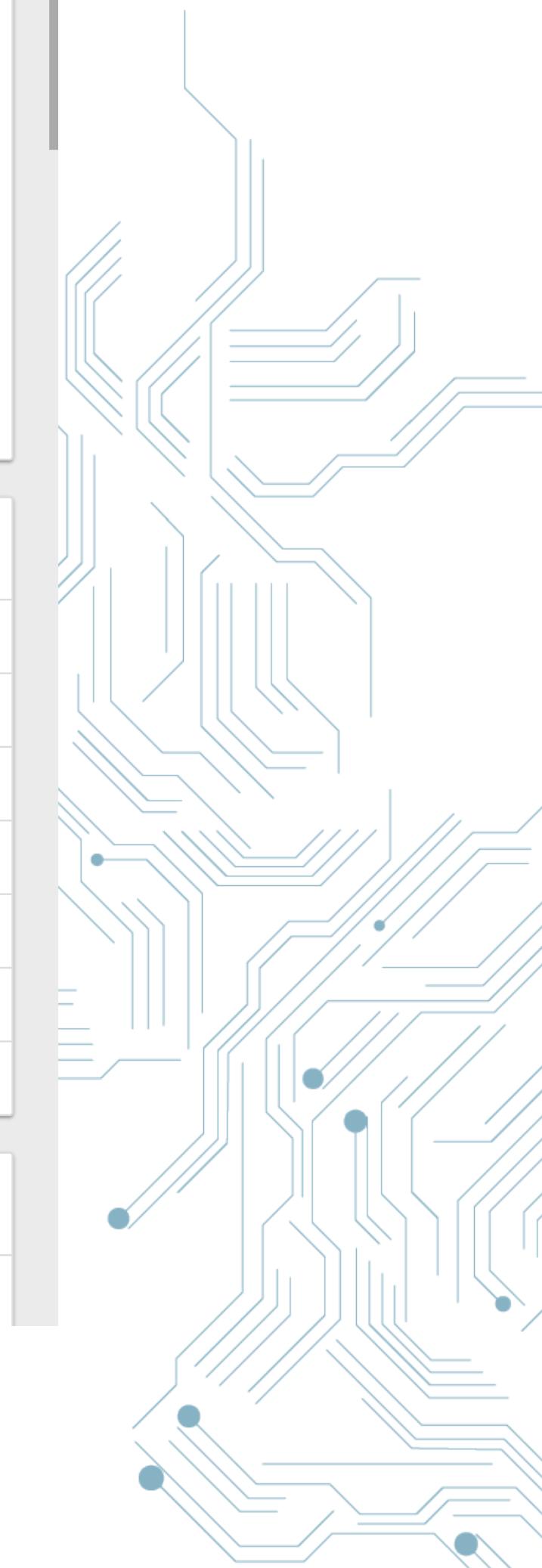


## Pods

Name	Node	Status	Restarts	Age	⋮
✓ symfony-demo-5b75f5fc6-jg8n4	docker-for-desktop	Running	23	8 days	⋮ ⋮
✓ symfony-demo-5b75f5fc6-c7wr9	docker-for-desktop	Running	0	8 days	⋮ ⋮
✓ silly-grizzly-mysql-556c9b5bcb-5jdrt	docker-for-desktop	Running	1	8 days	⋮ ⋮
✓ kindly-cardinal-nginx-ingress-controller-5549f5597c-97kcw	docker-for-desktop	Running	3	9 days	⋮ ⋮
✓ kindly-cardinal-nginx-ingress-default-backend-564d9d9477-tmnrr	docker-for-desktop	Running	4	9 days	⋮ ⋮
✓ idle-ferrit-kubernetes-dashboard-5b5bf59977-t9xb9	docker-for-desktop	Running	3	9 days	⋮ ⋮

## Deployments

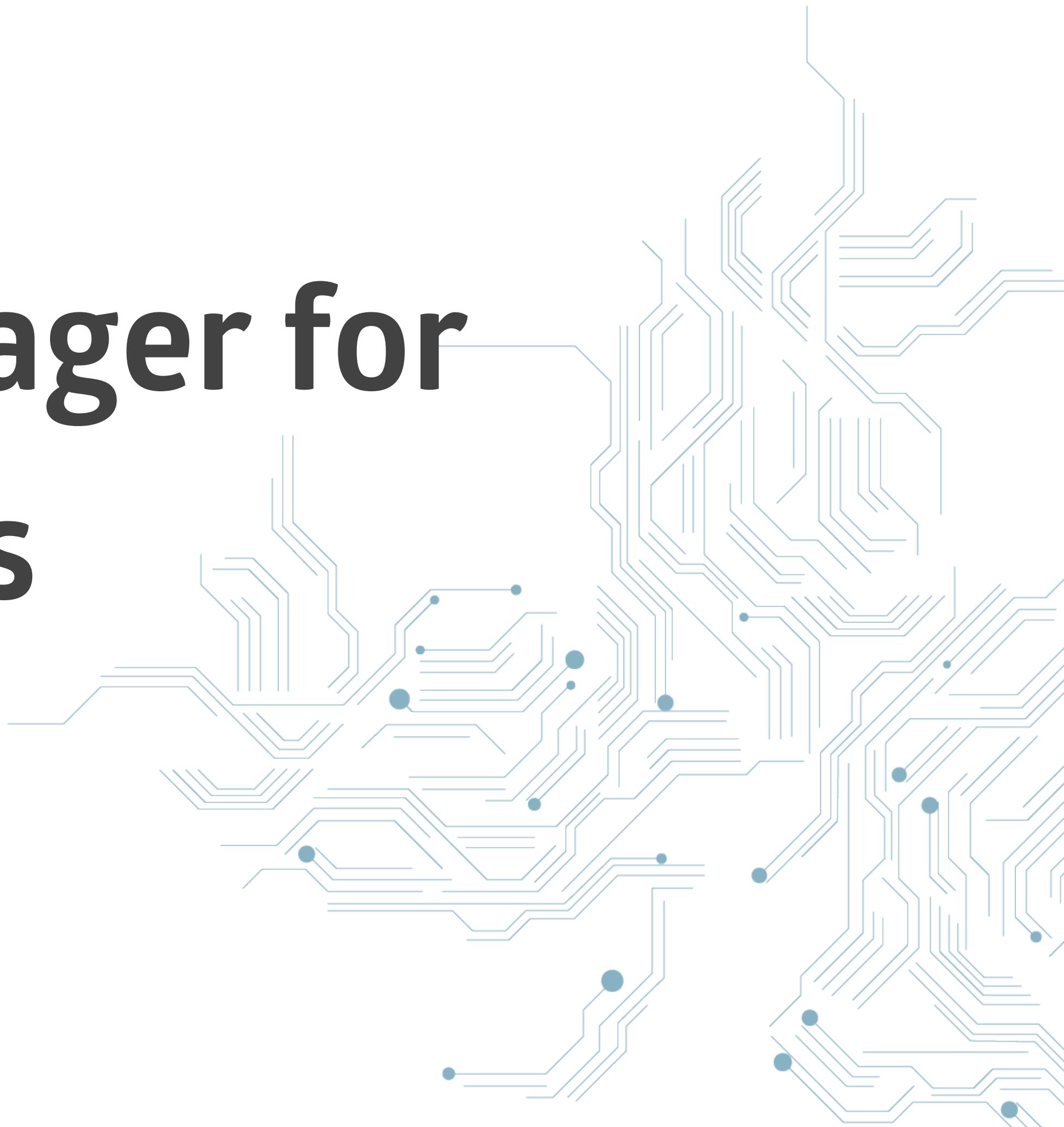
Name	Labels	Pods	Age	Images
------	--------	------	-----	--------



# Helm

## The package manager for

# Kubernetes



*bash*

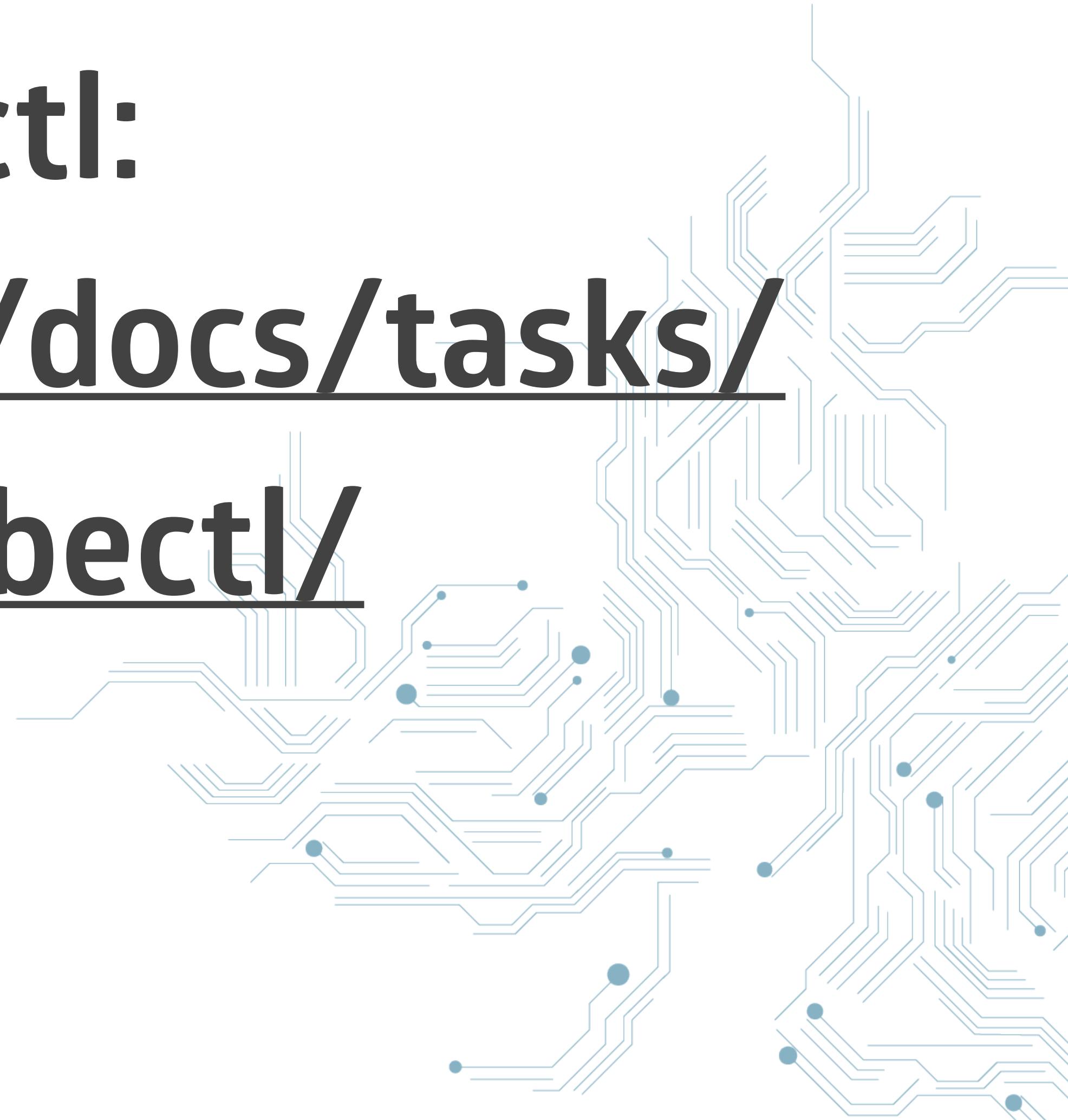
```
$ helm install stable/wordpress
```

# hands-on

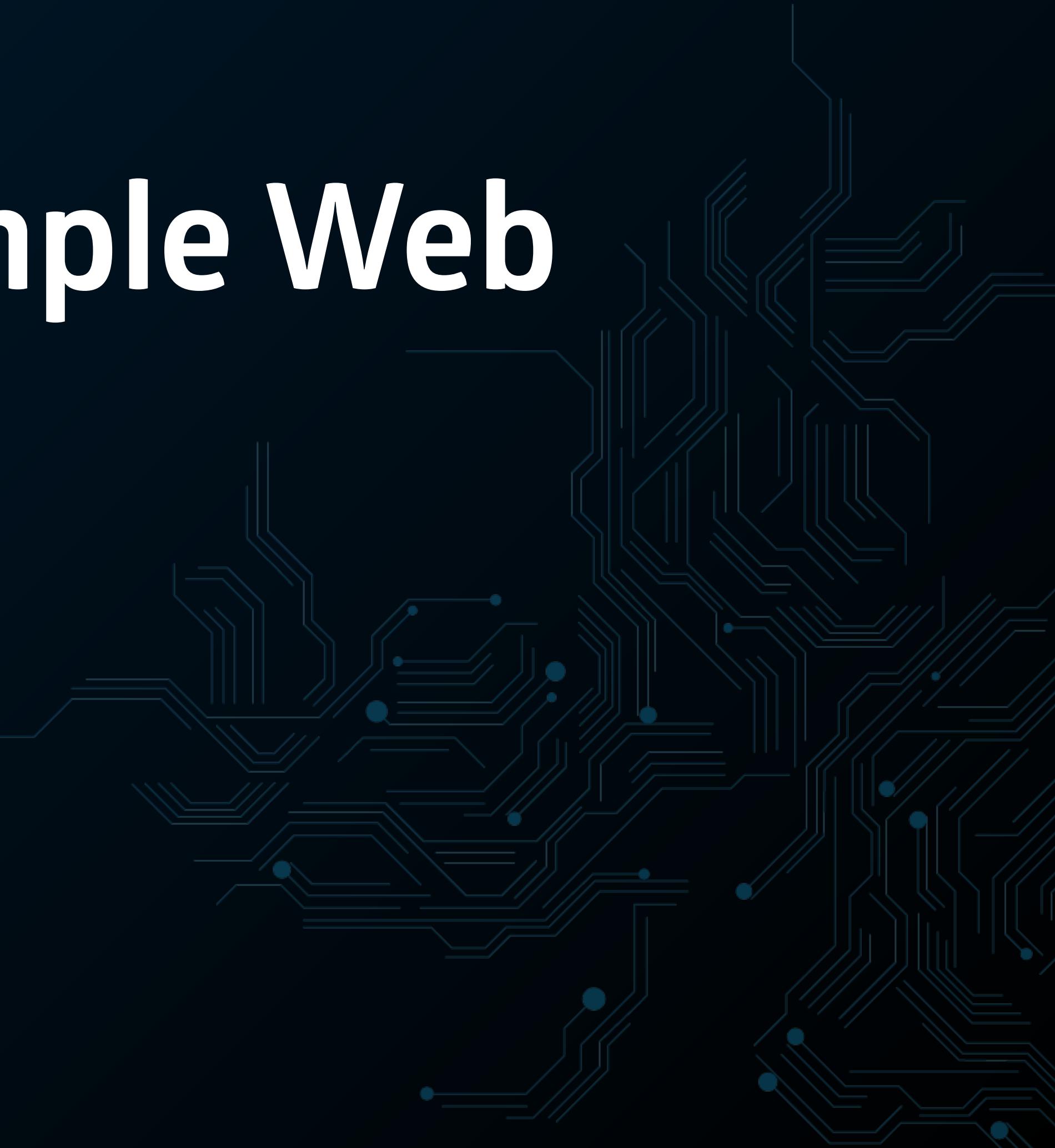


Install kubectl:

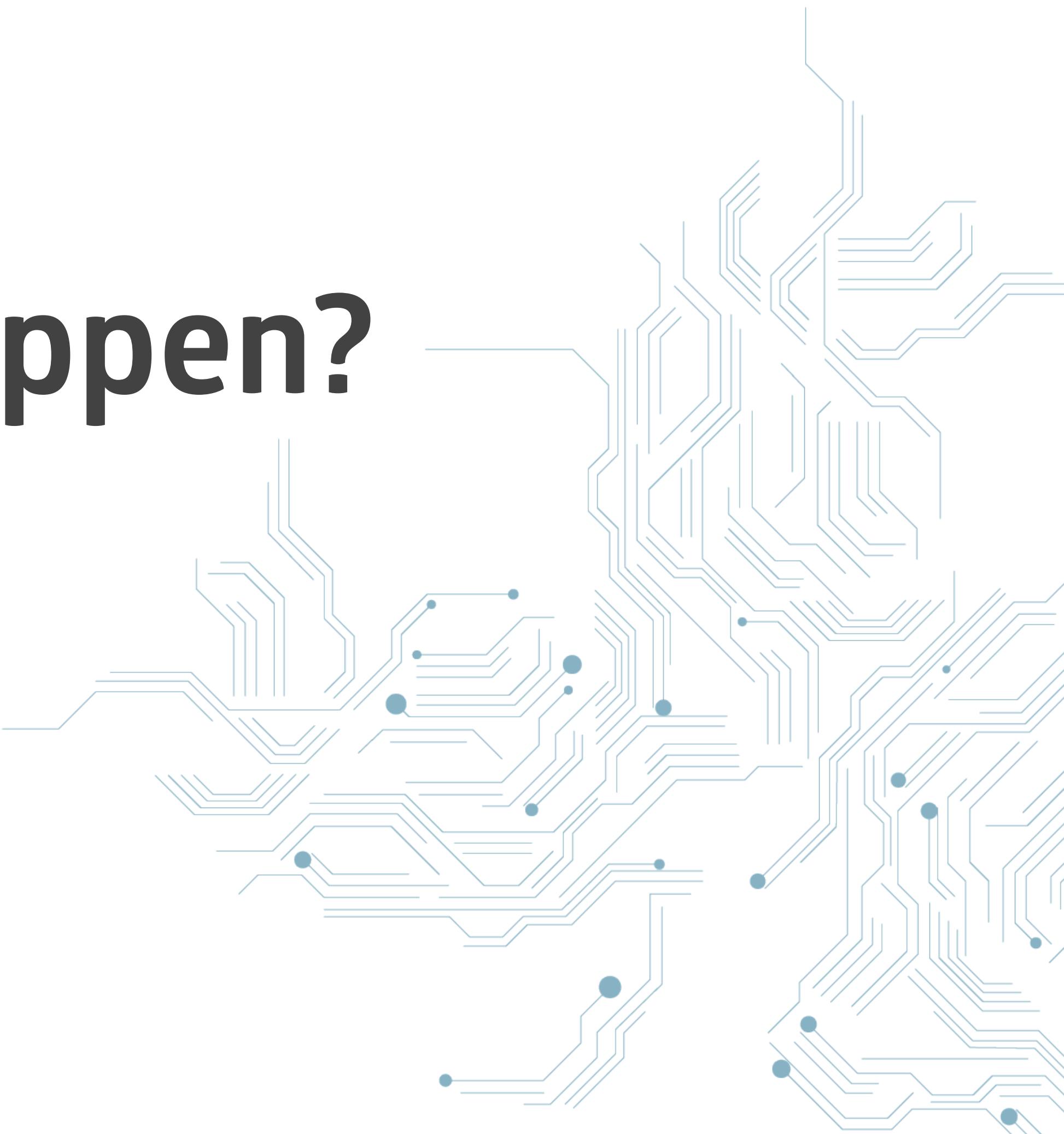
<https://kubernetes.io/docs/tasks/tools/install-kubectl/>

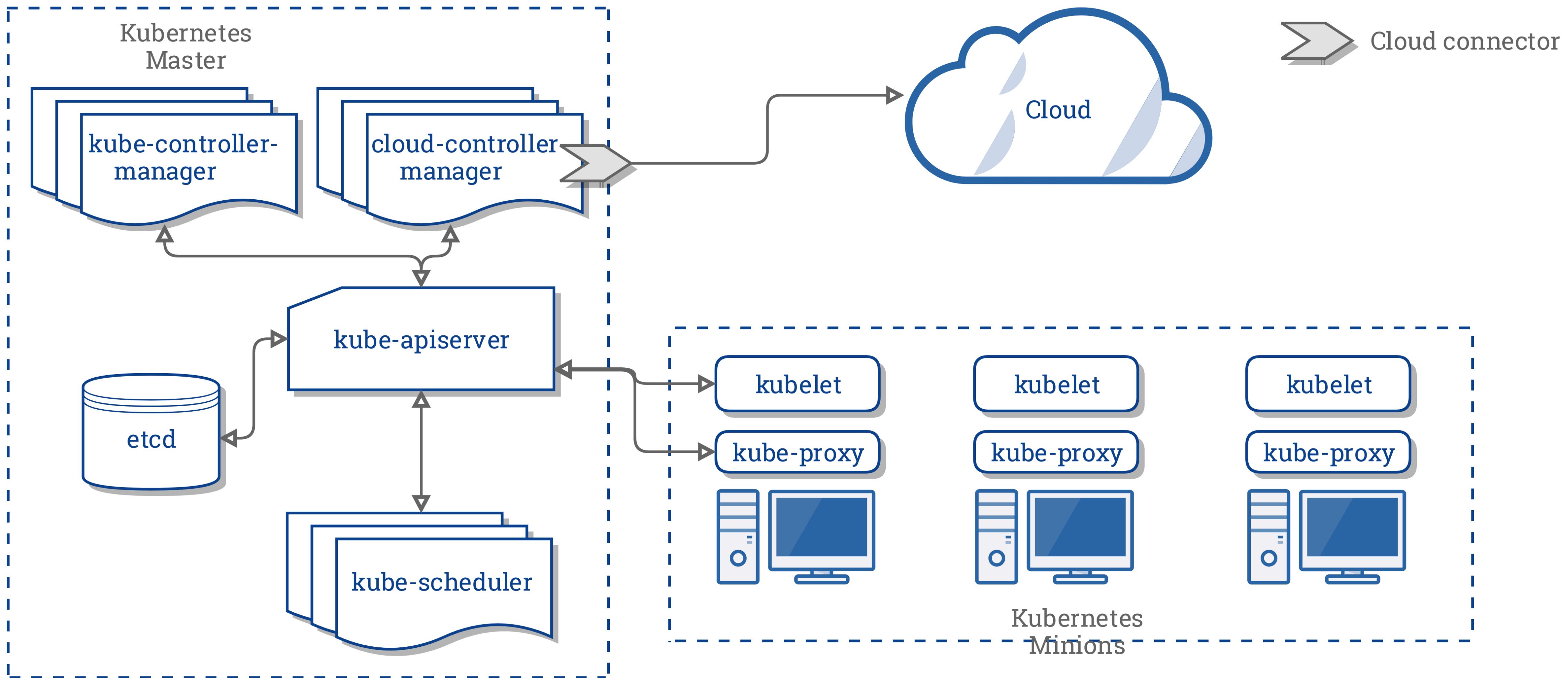


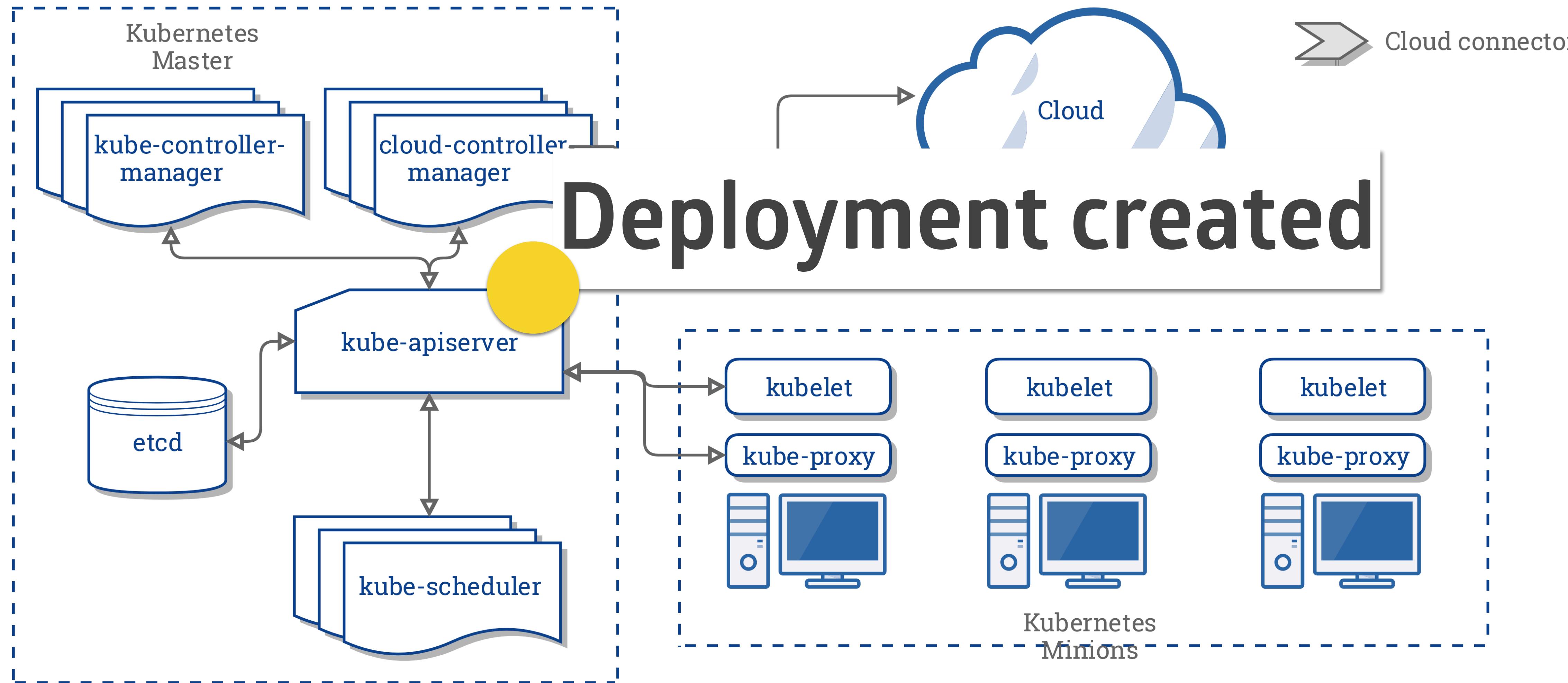
# # 1 - Deploying a simple Web Application



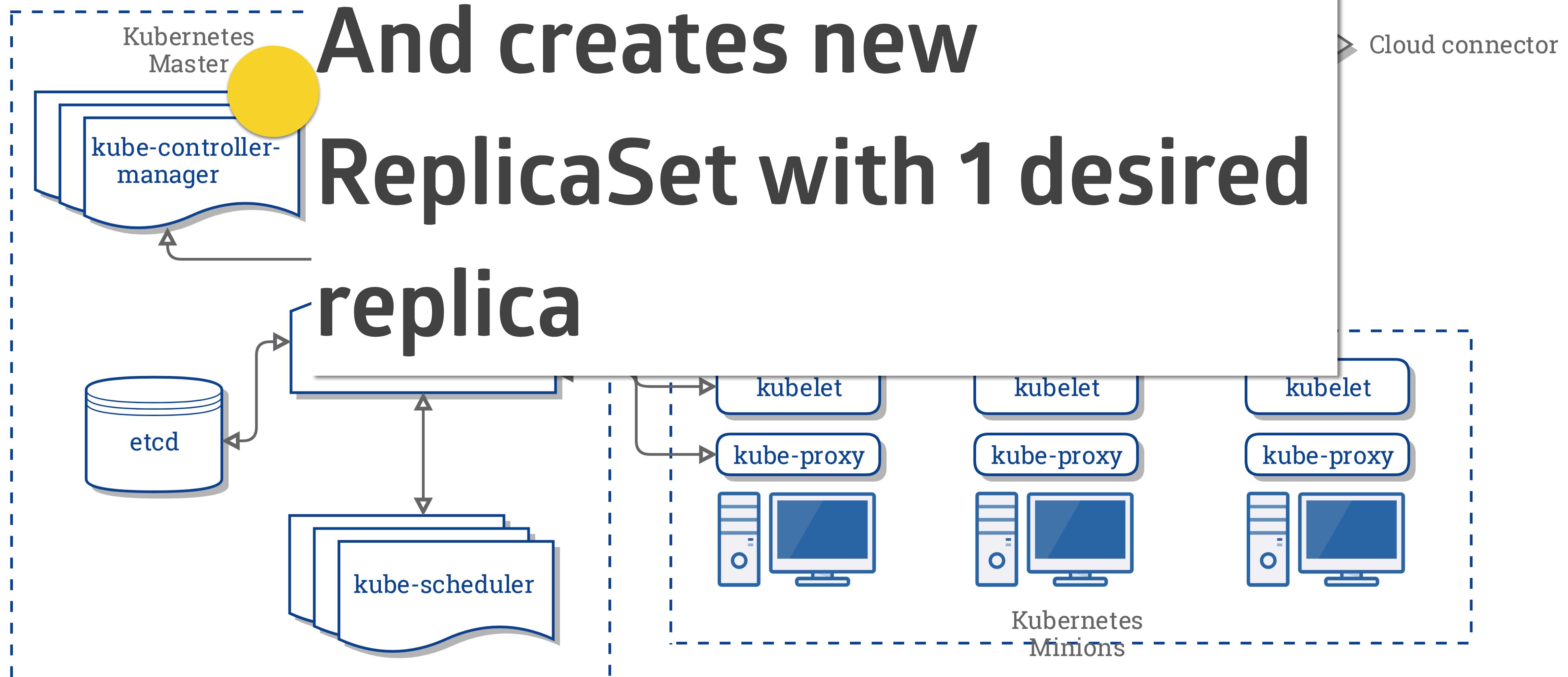
# What did just happen?



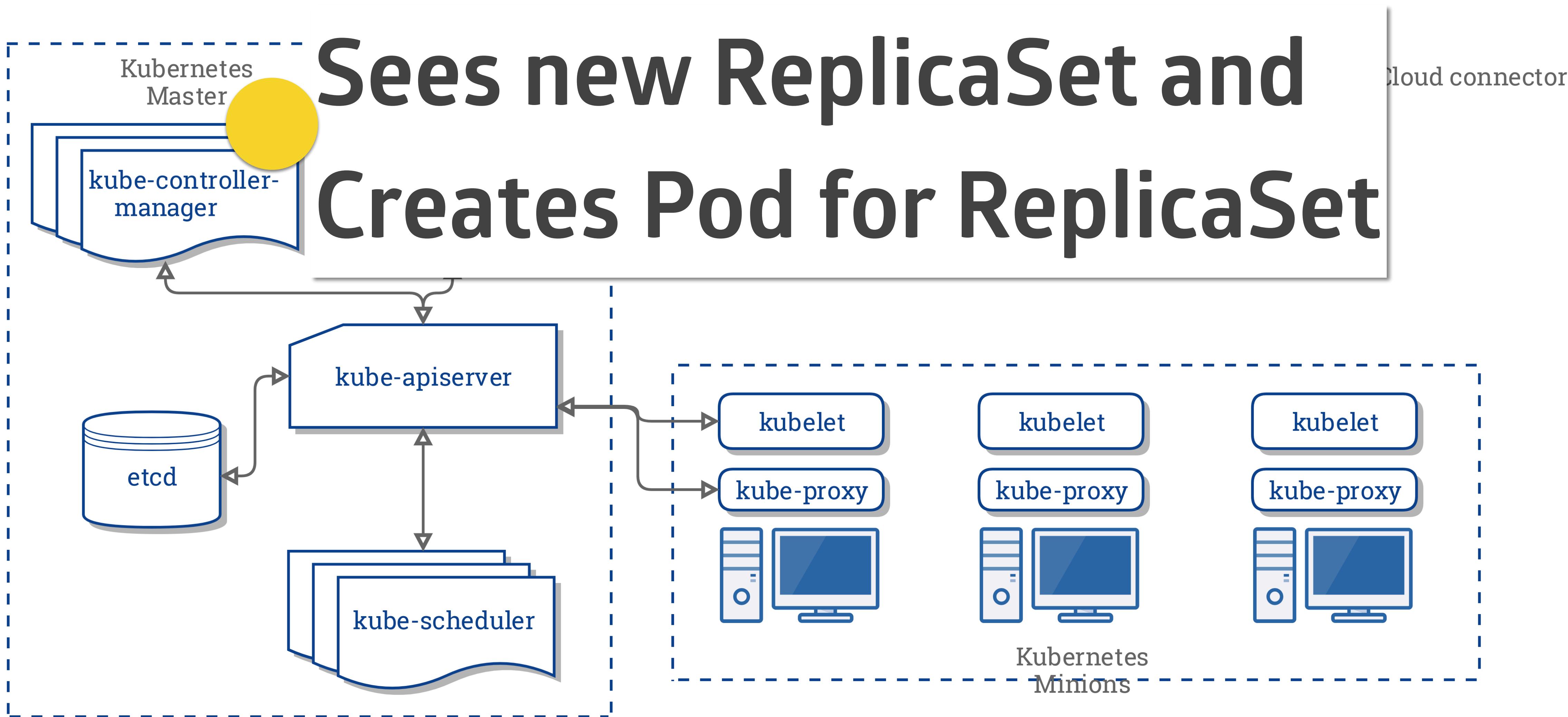


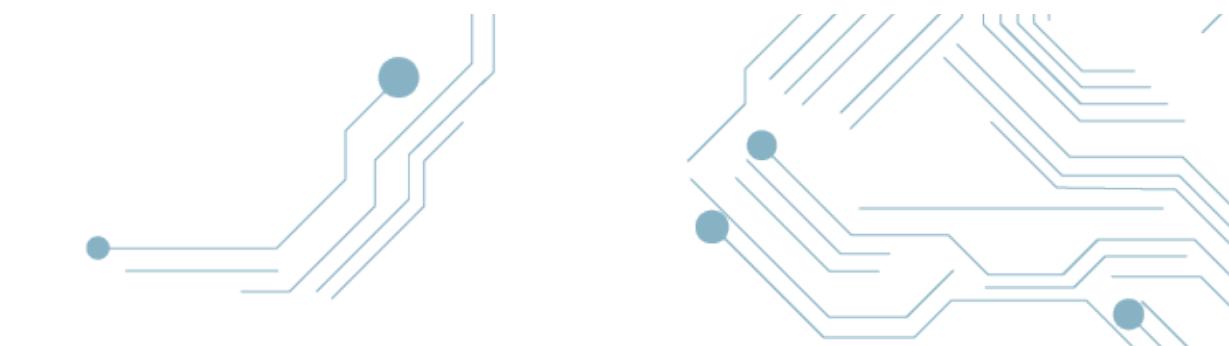
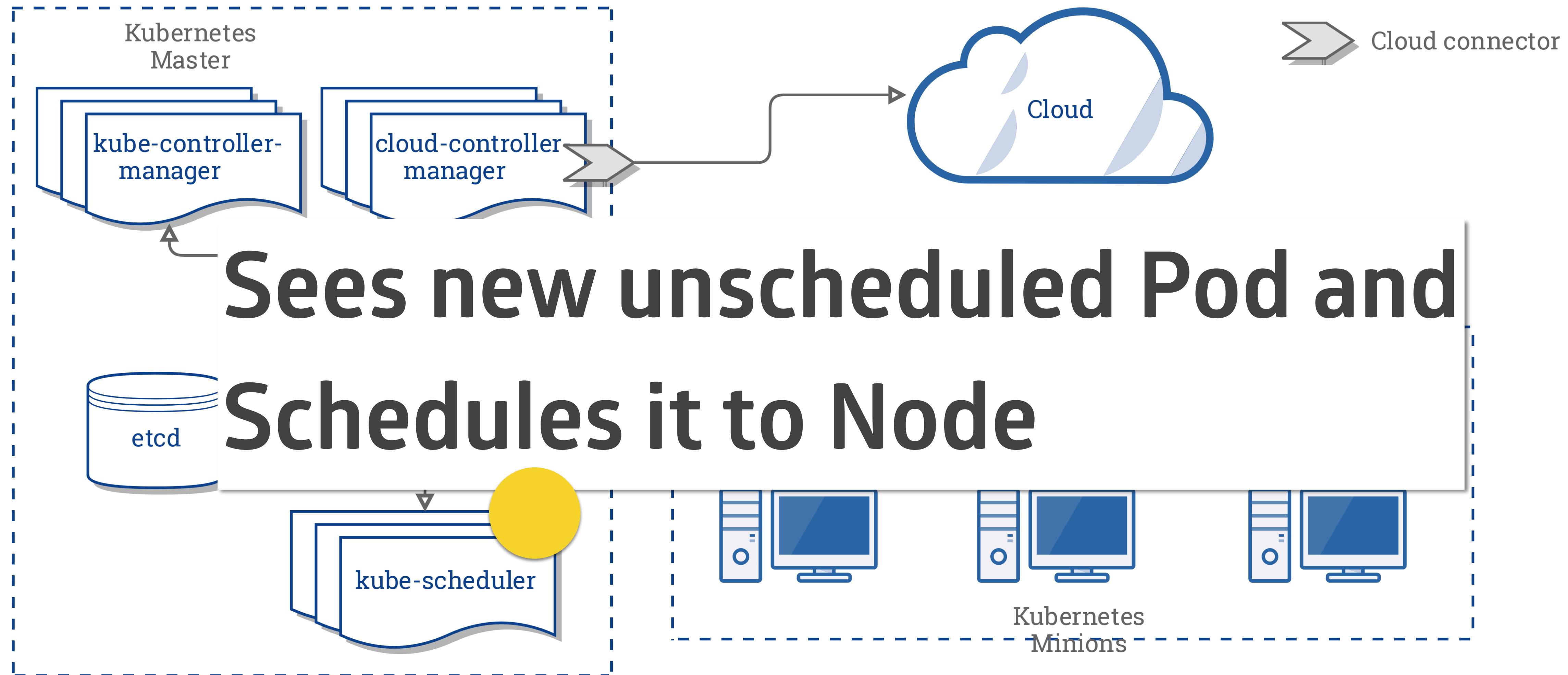


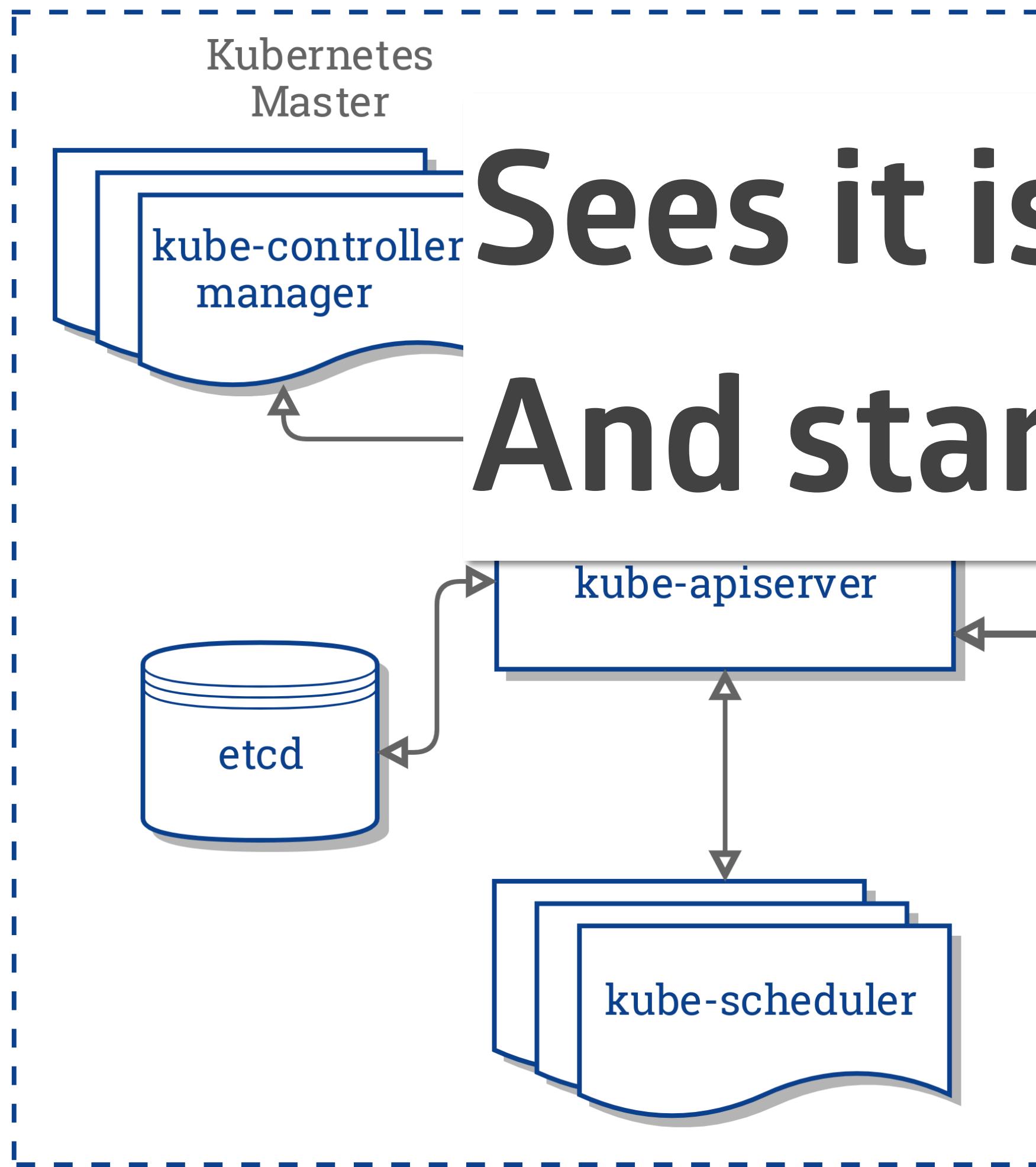
# Sees new Deployment



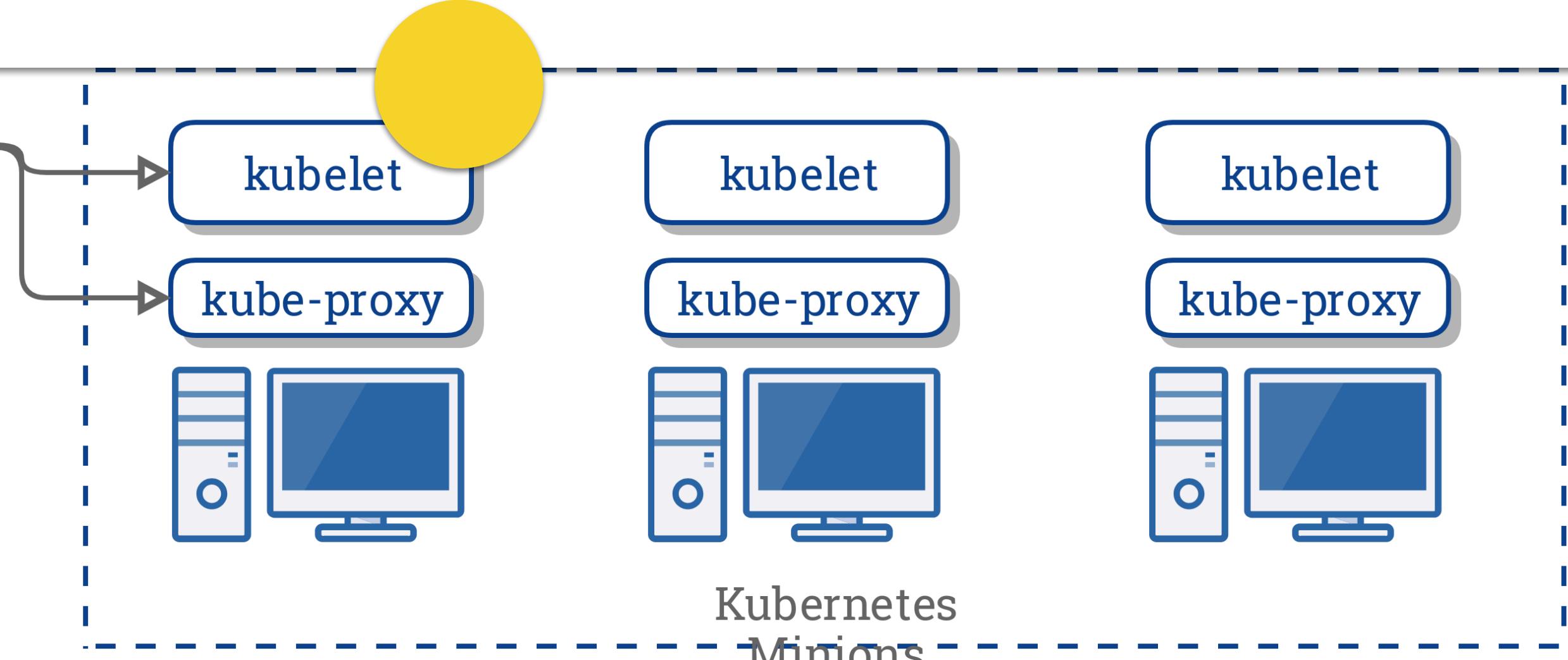
# Sees new ReplicaSet and Creates Pod for ReplicaSet



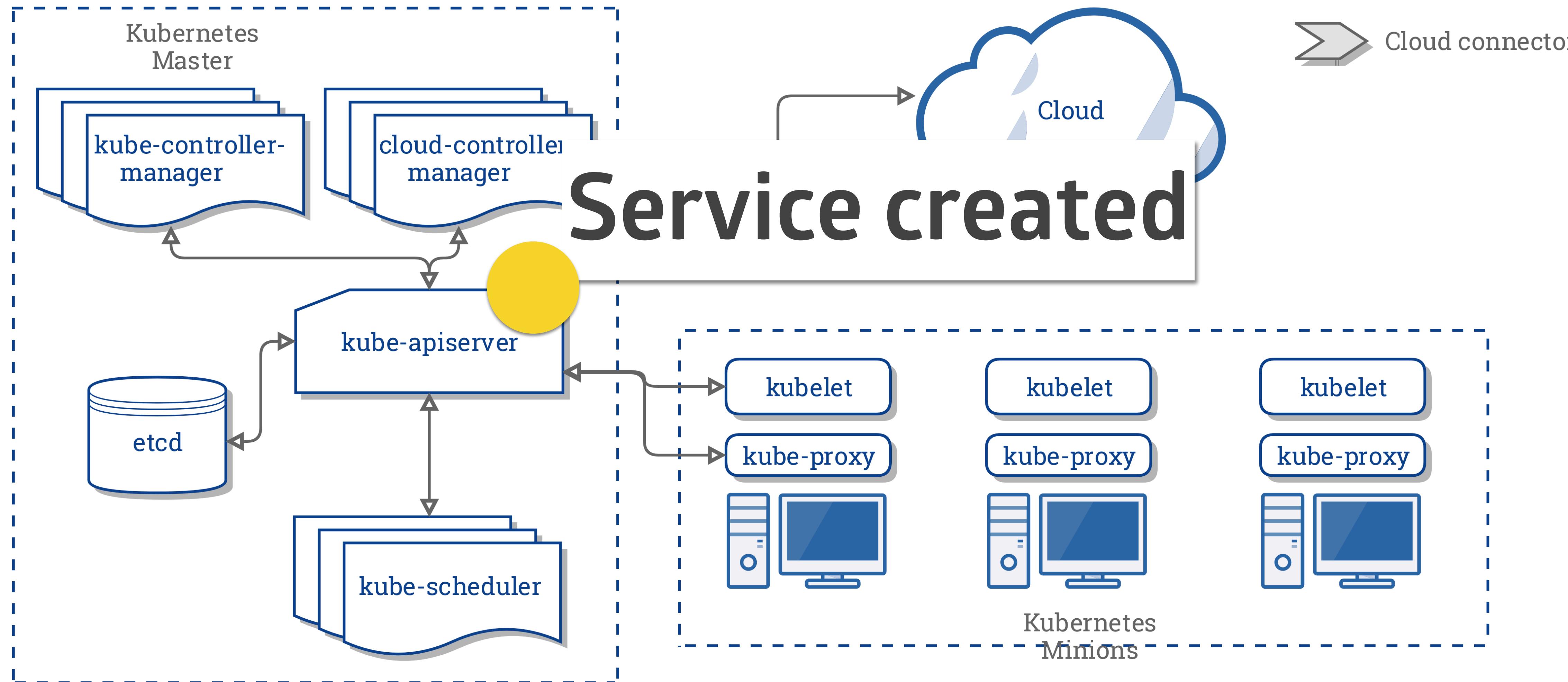




# Sees it is supposed to start a Pod And starts its Containers



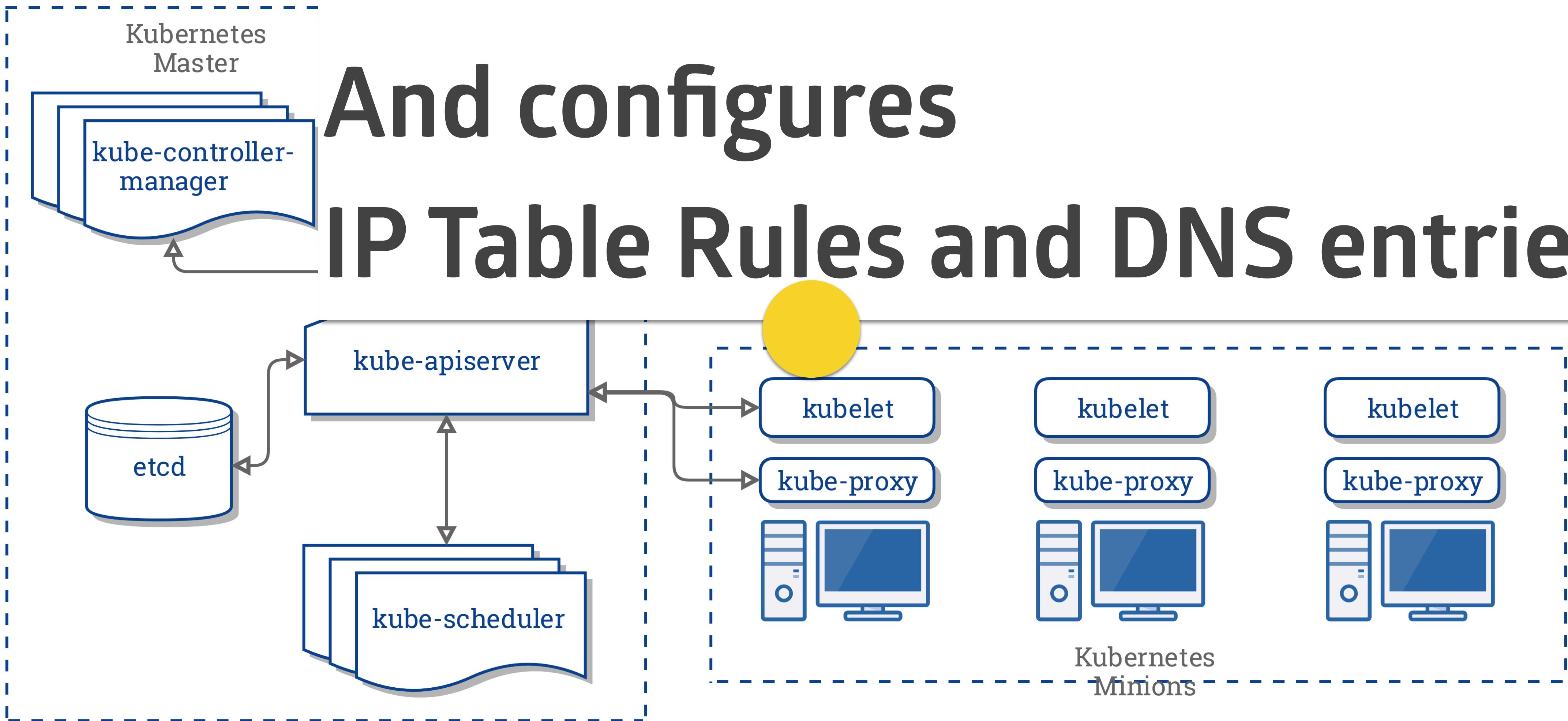
Cloud connector

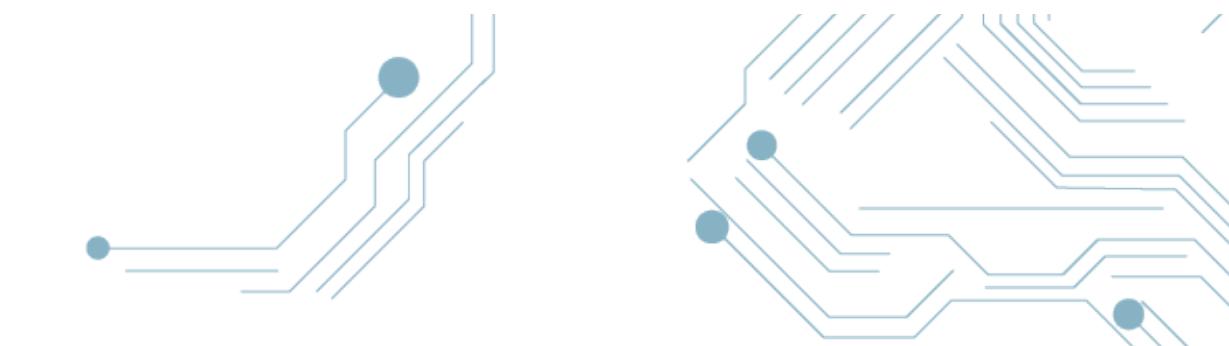
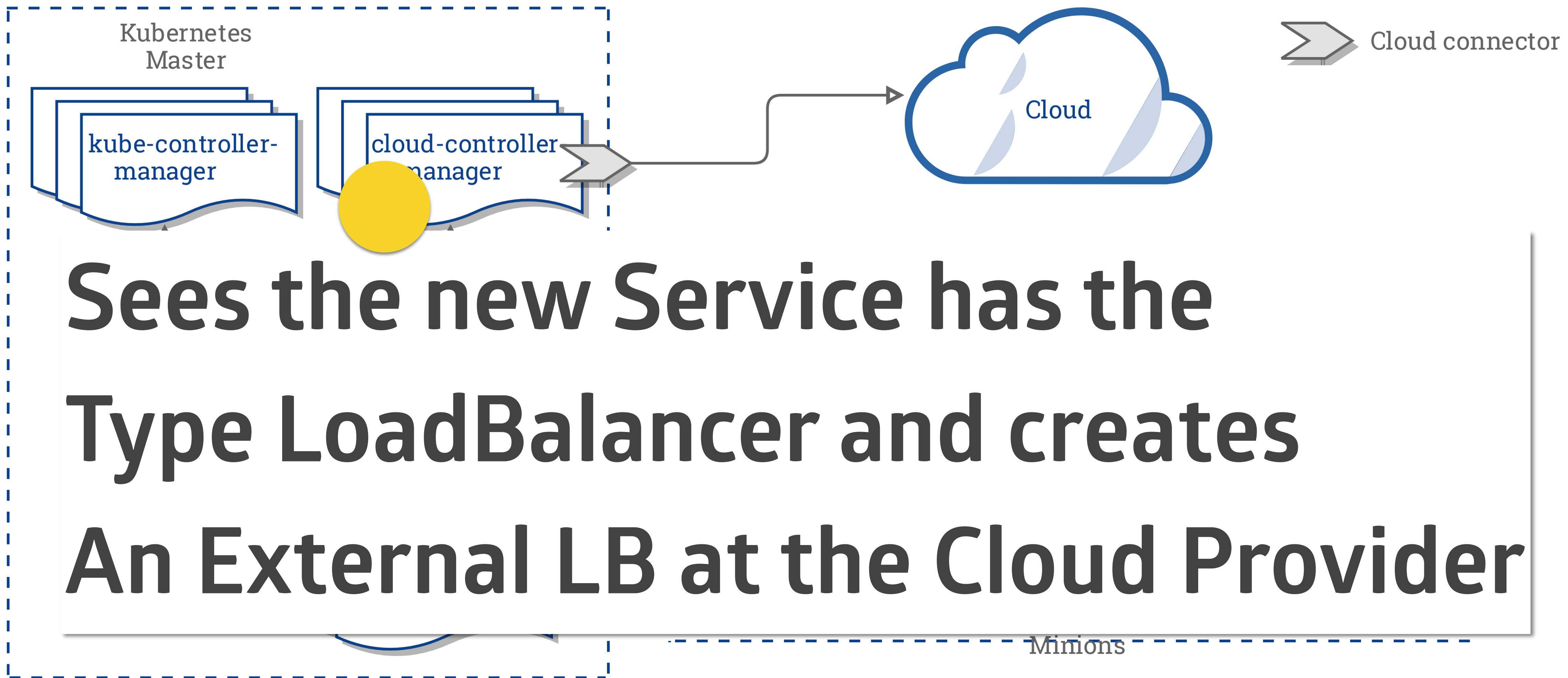


# Sees the new Service

## And configures

### IP Table Rules and DNS entries

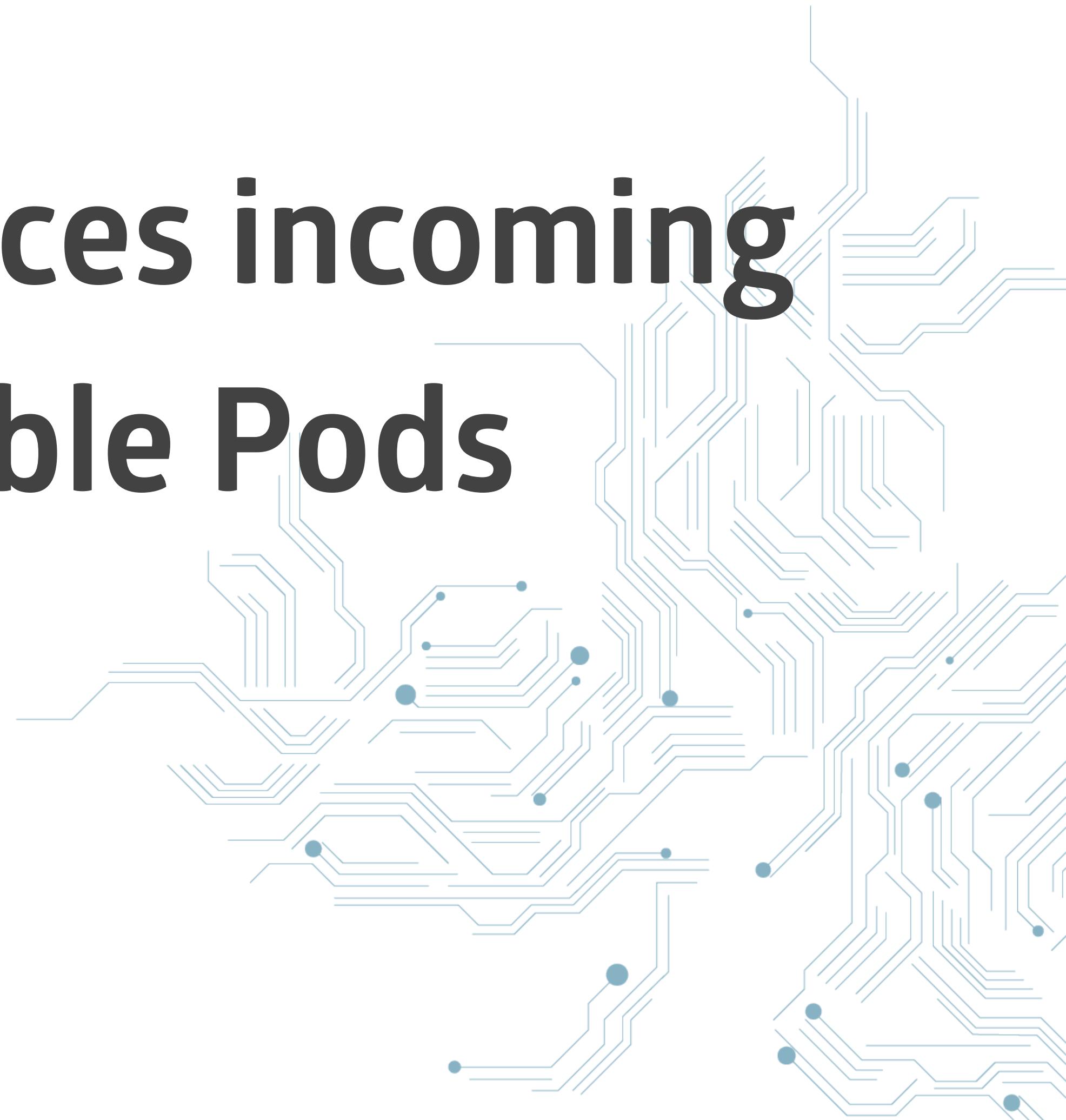




# How is traffic routed to the Pod



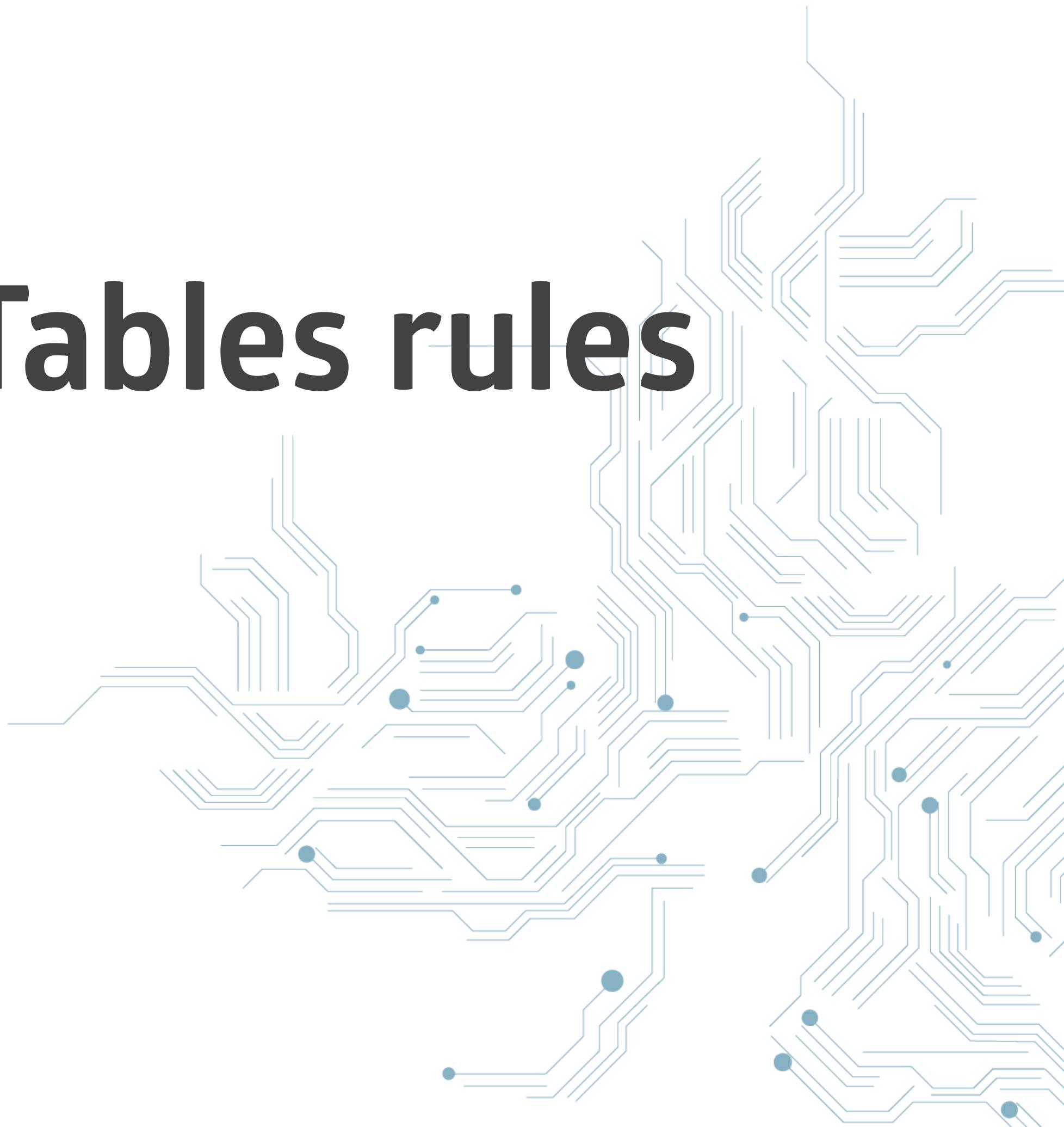
**The Service load-balances incoming  
traffic to all available Pods**



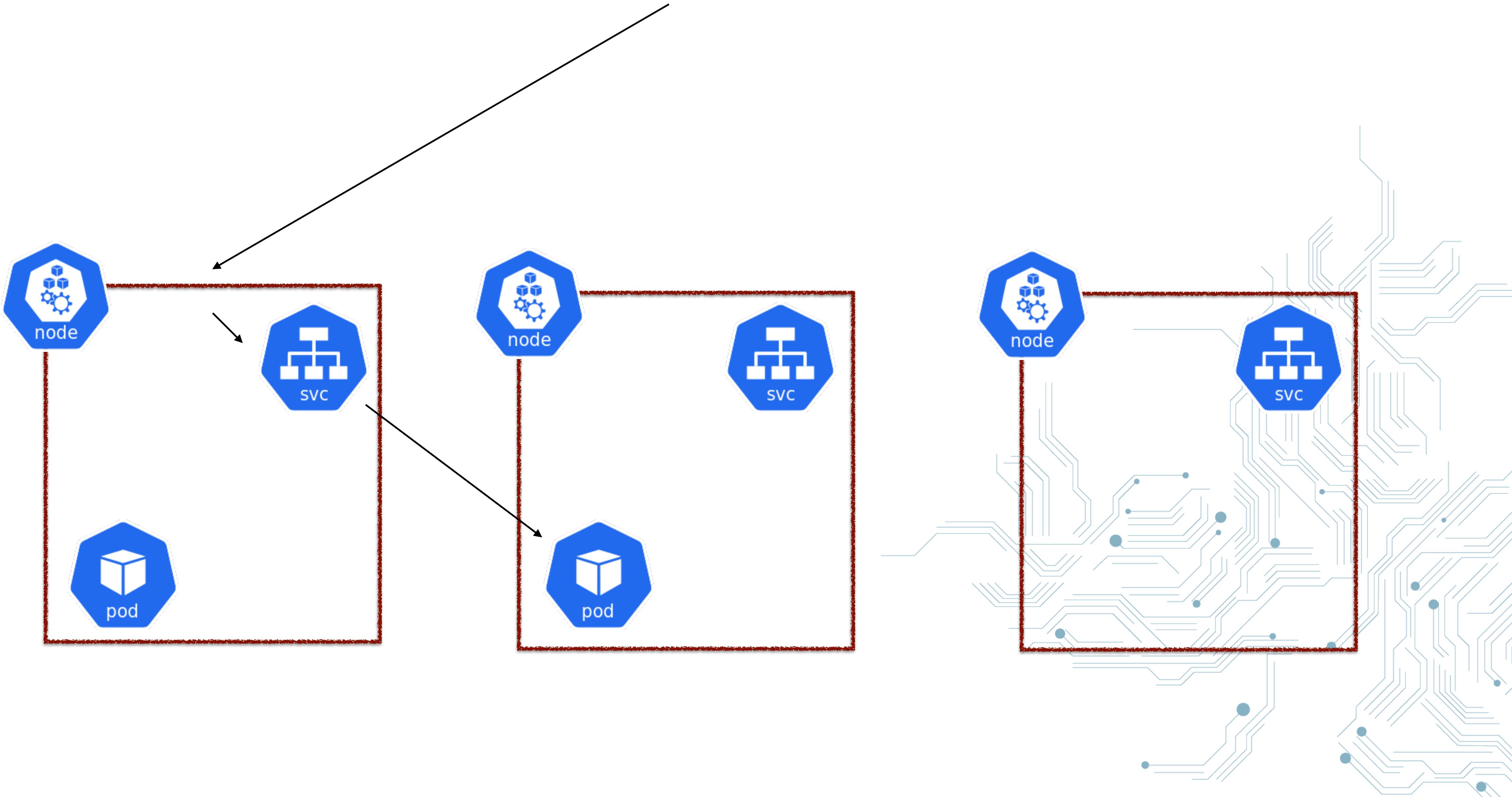
# **Every Service has a virtual IP**



# Round Robin with IP Tables rules



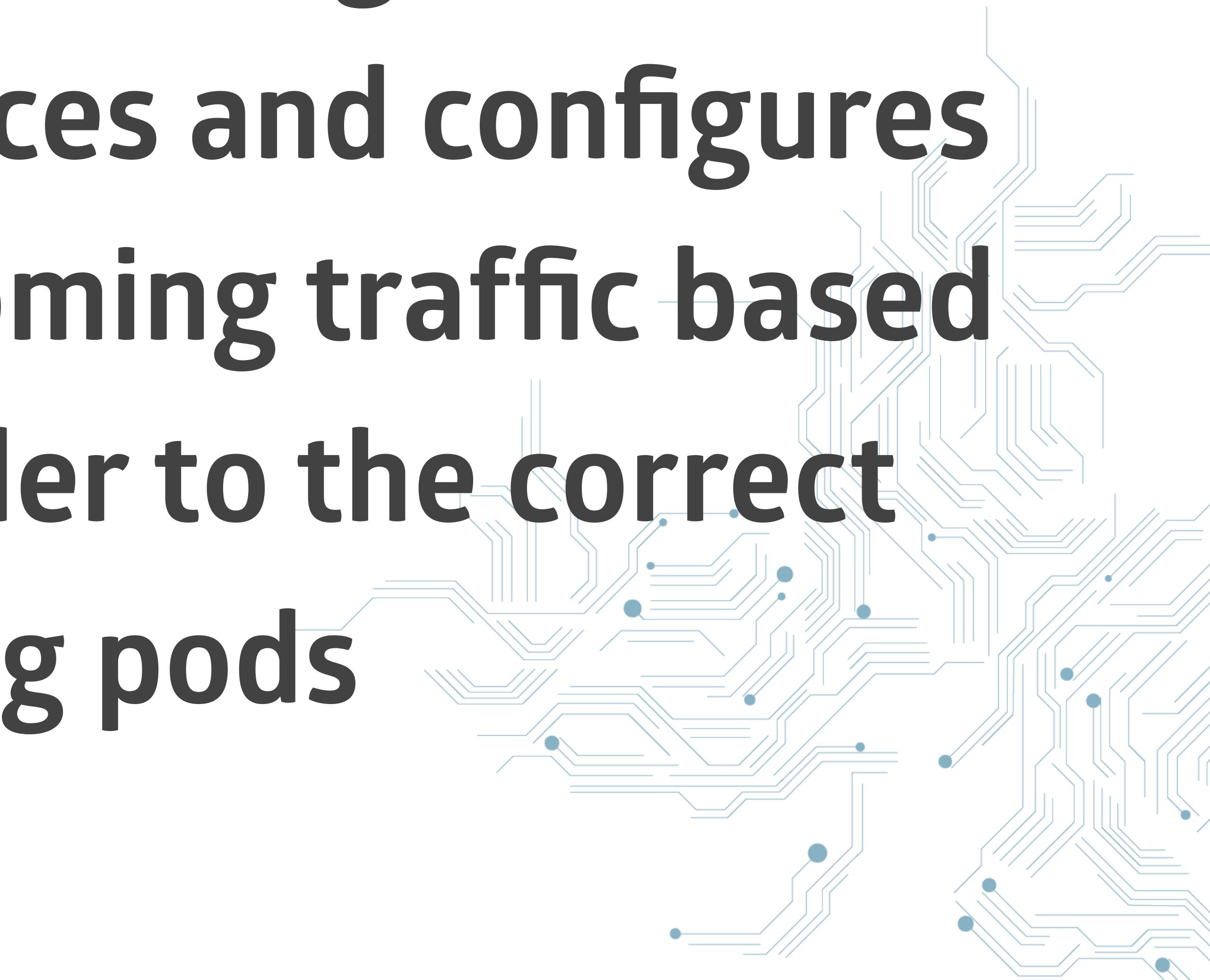
# OpenStack LoadBalancer



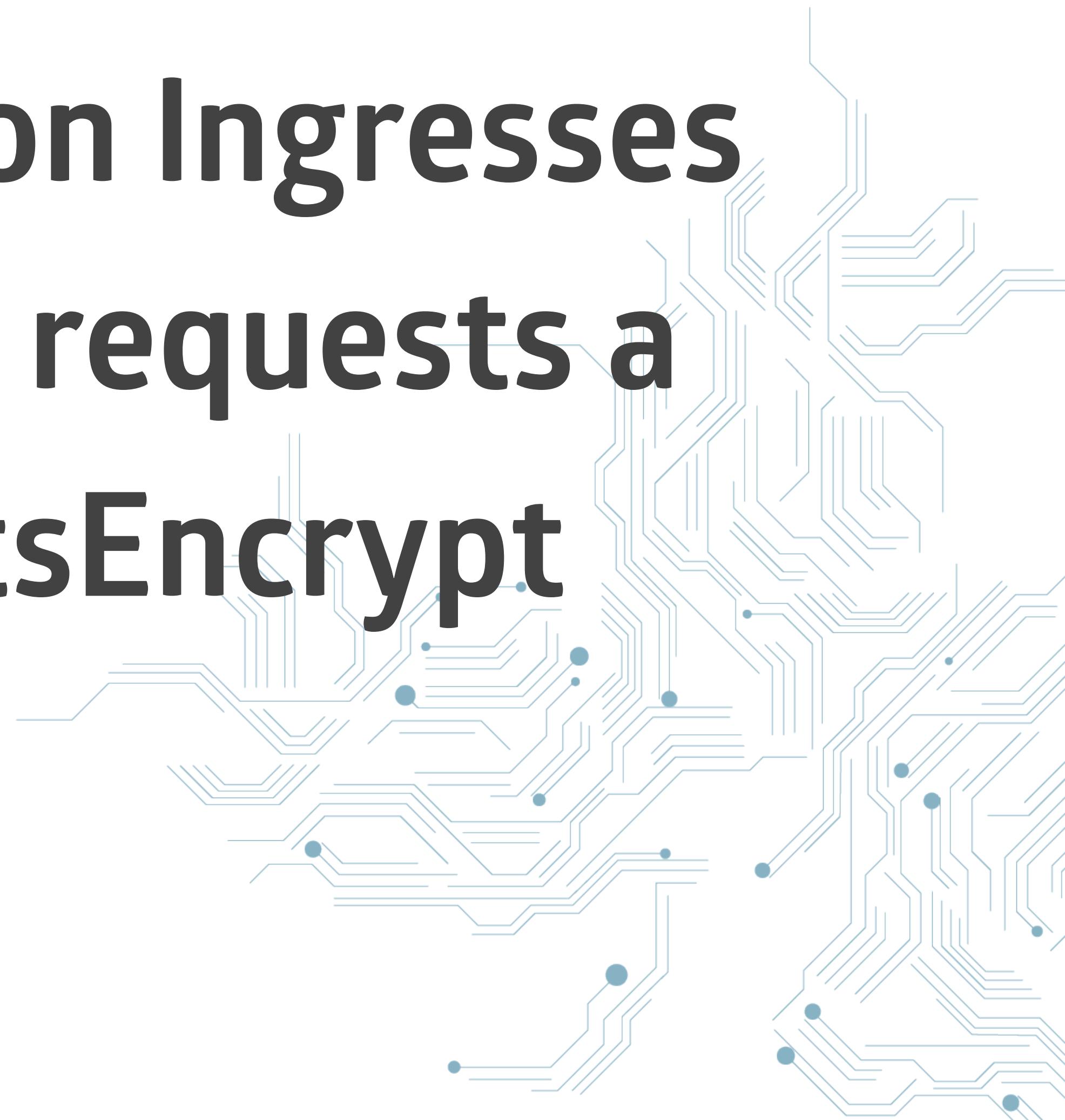
# # 9 - Using an Ingress with TLS



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.**



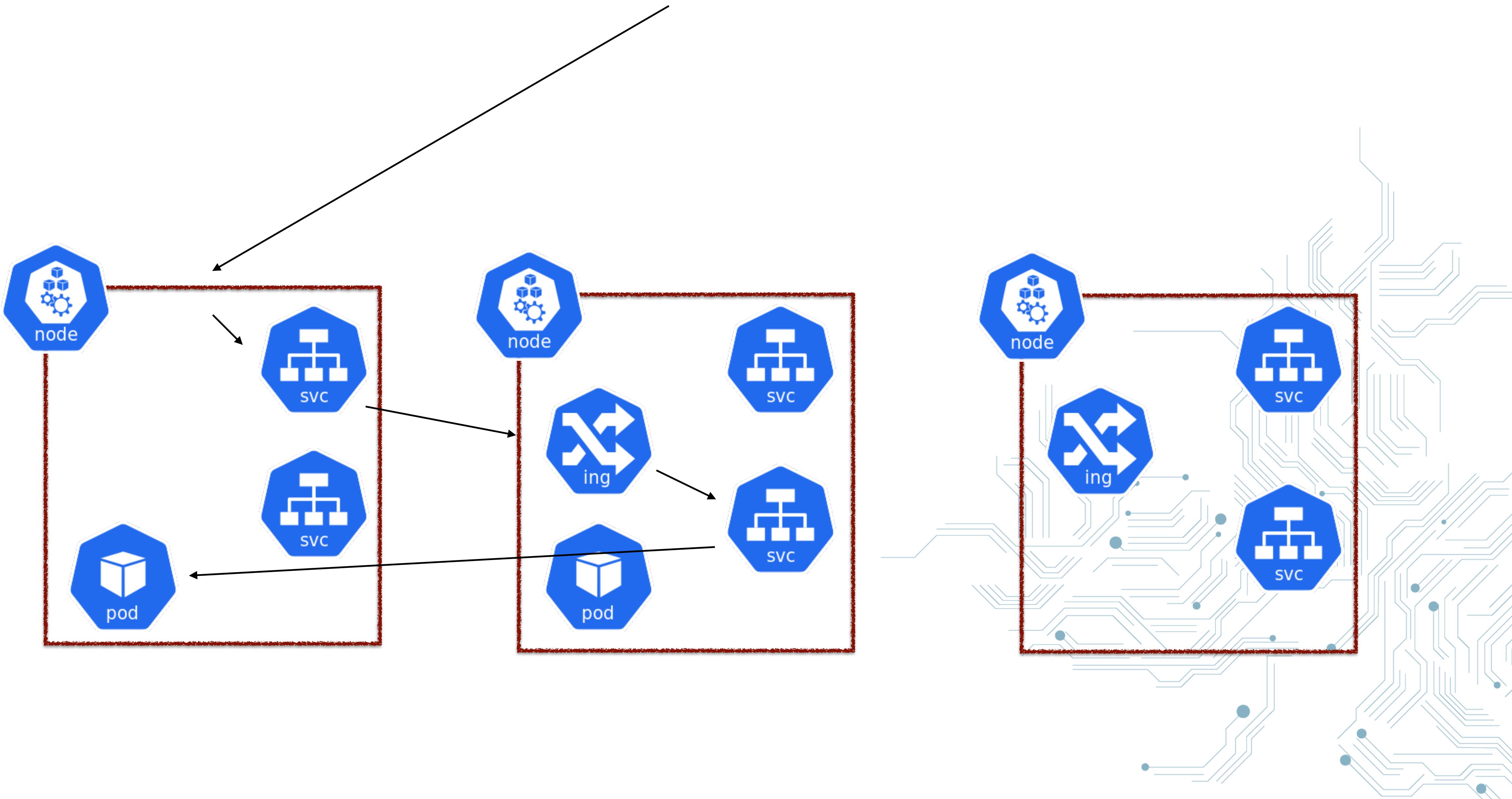
**External-DNS listens on Ingresses  
and creates DNS entries at AWS  
Route 53**



# How is traffic routed to the Pod

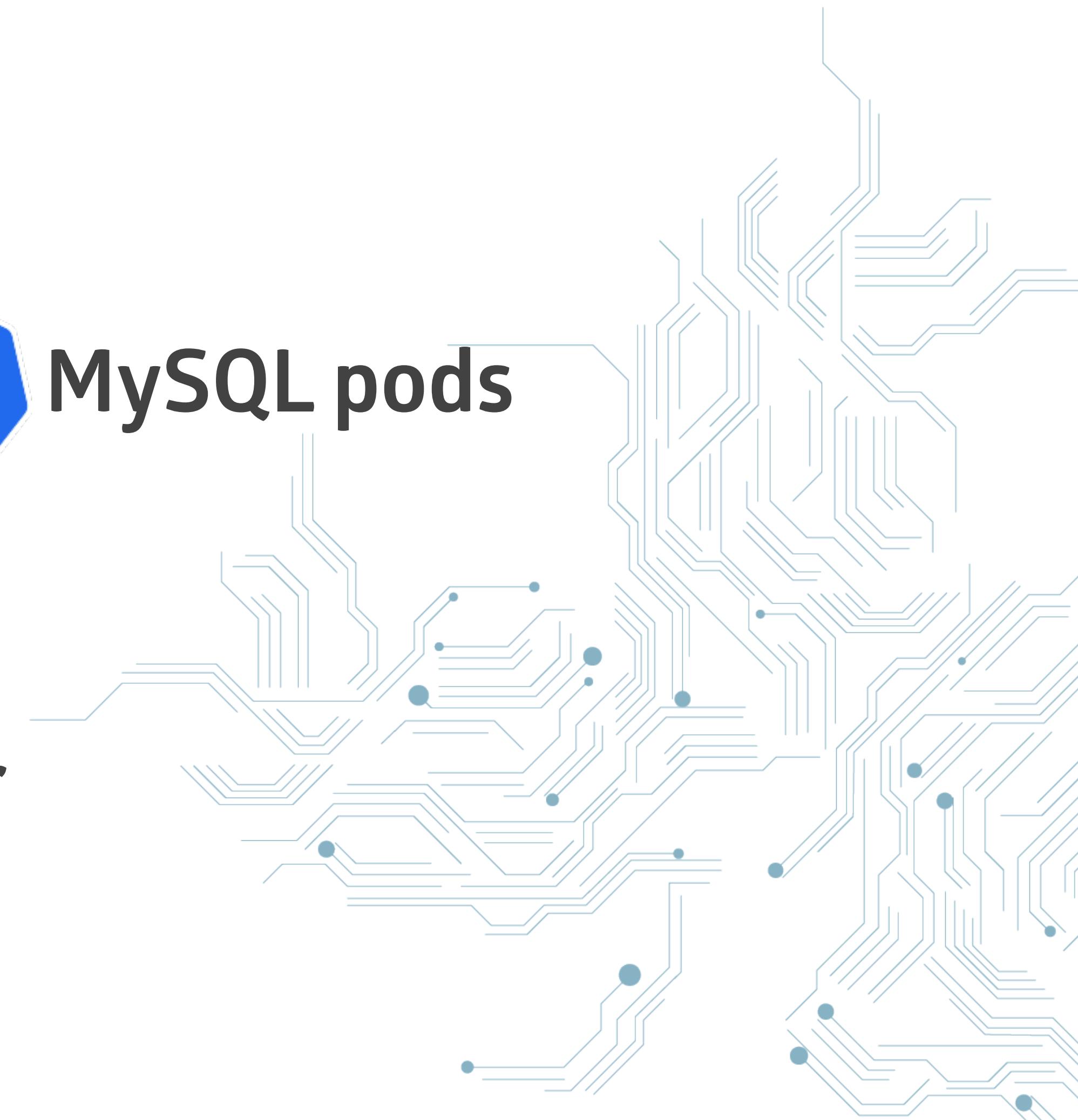
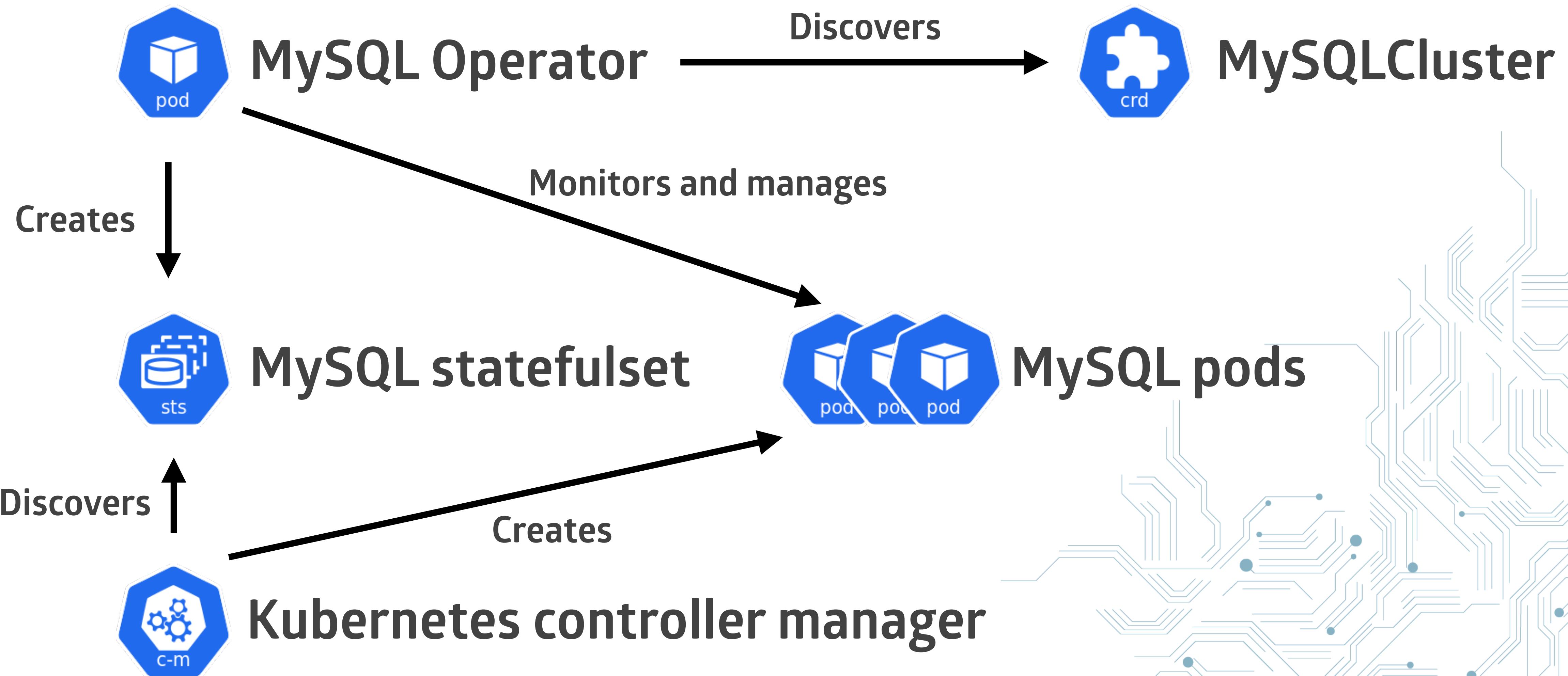


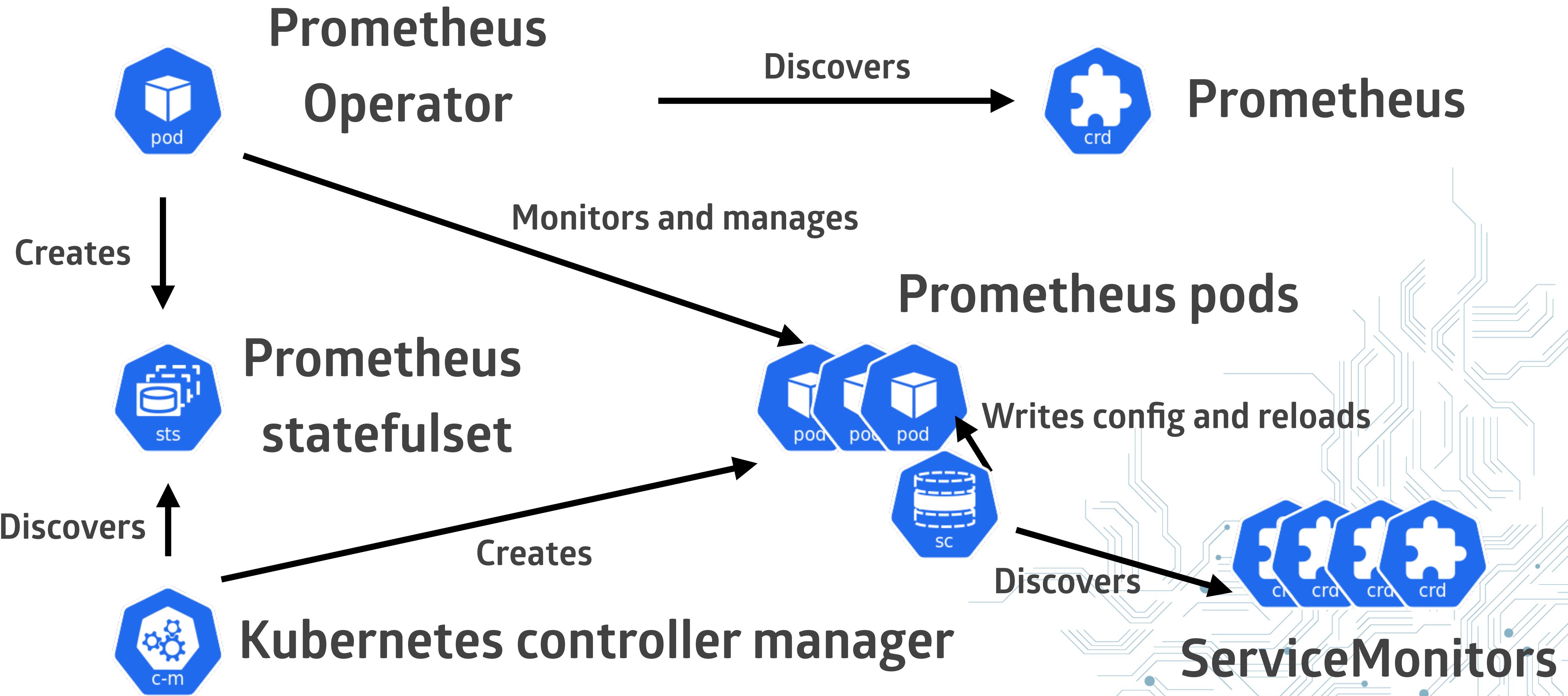
# OpenStack LoadBalancer



# # 10 - Operators



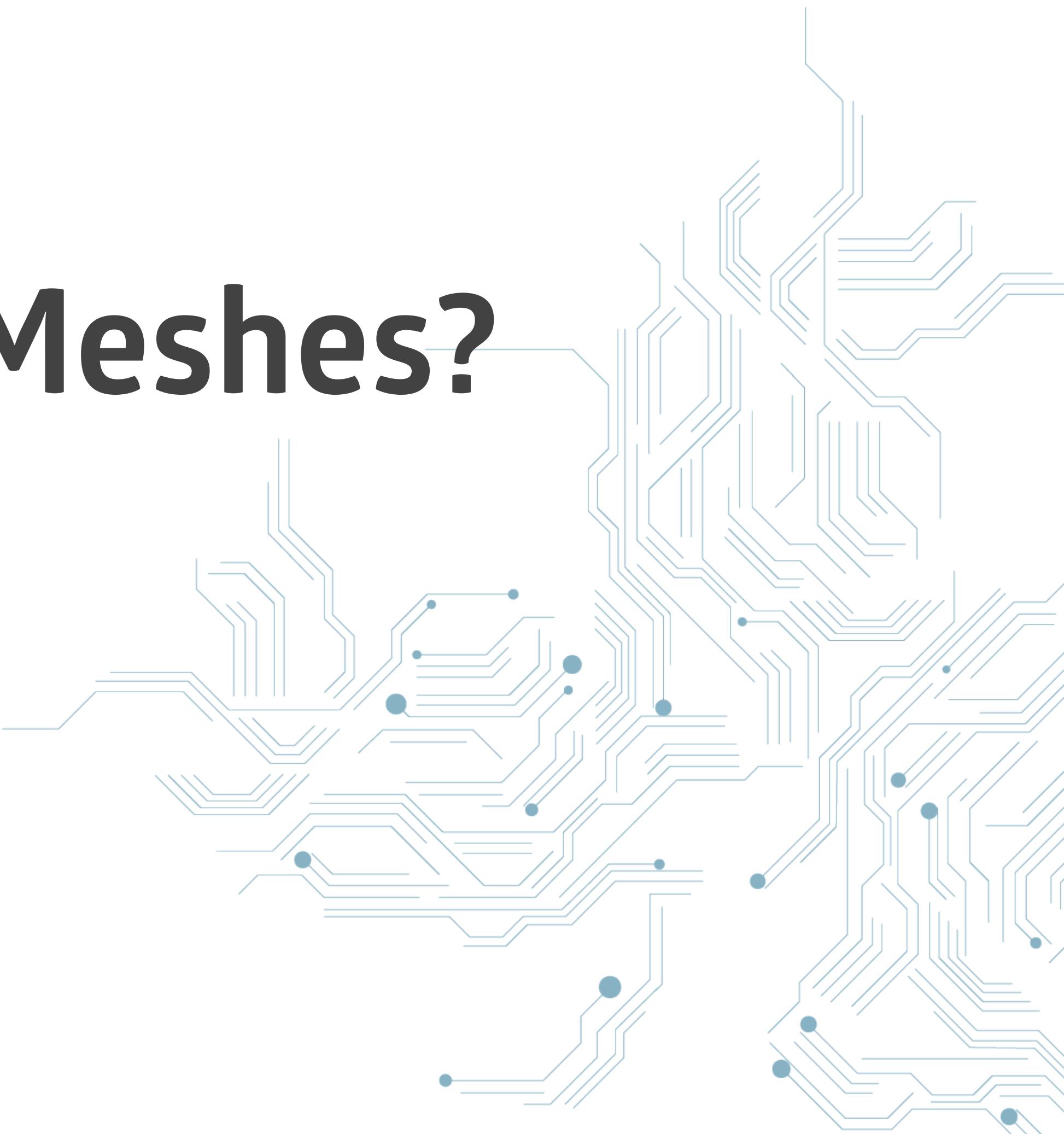


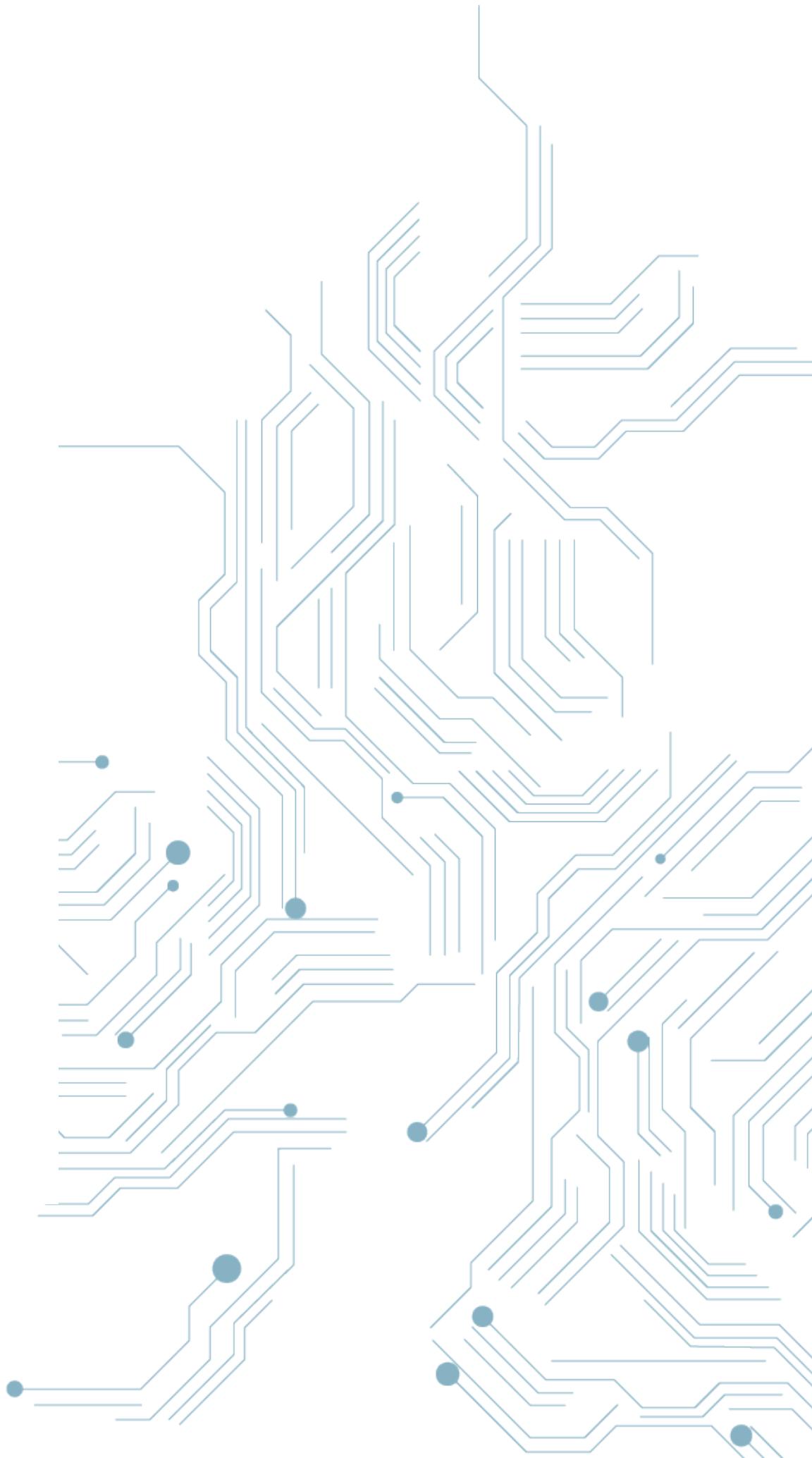
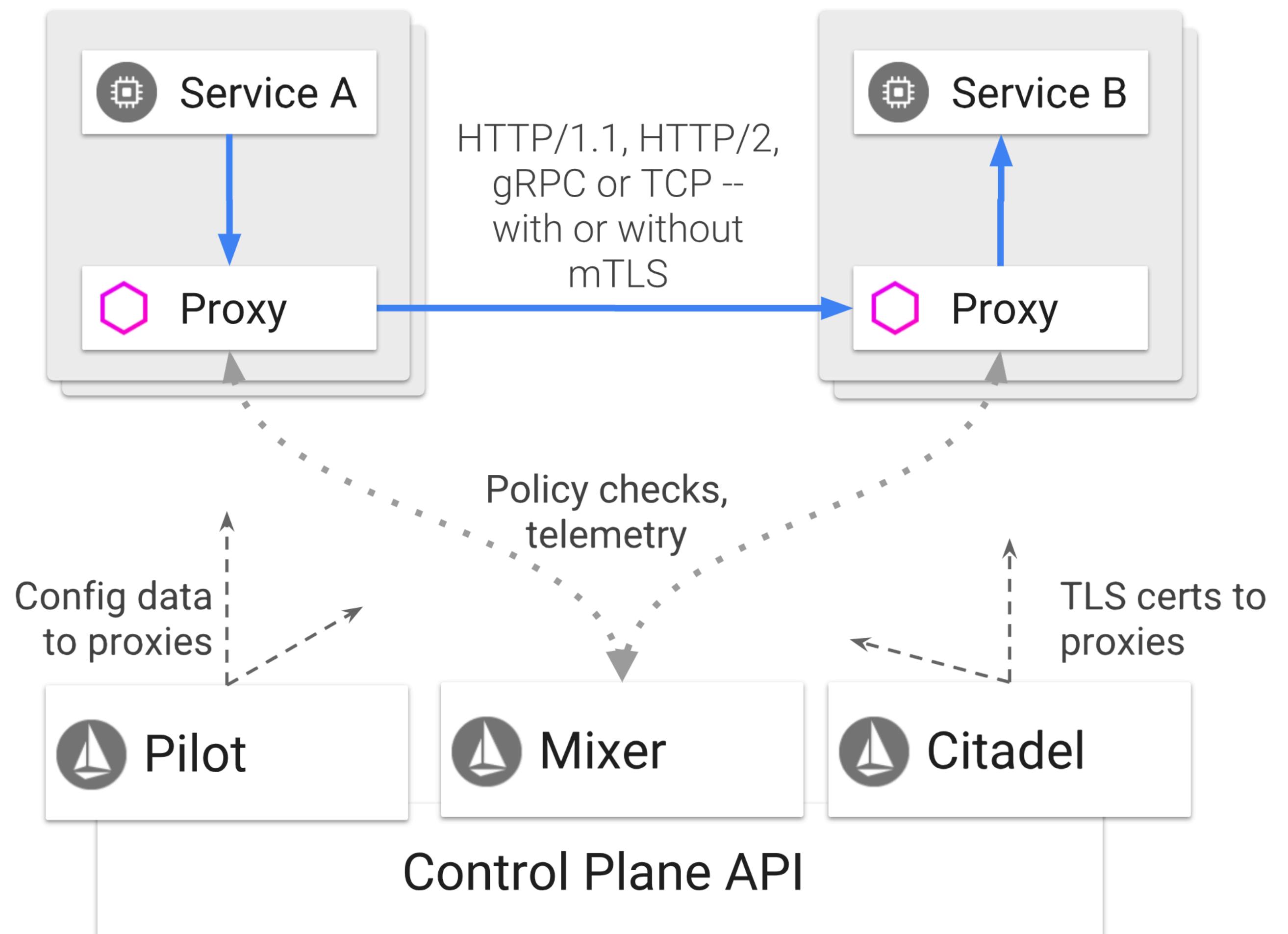


# # 14 - Service Meshes



# What are Service Meshes?





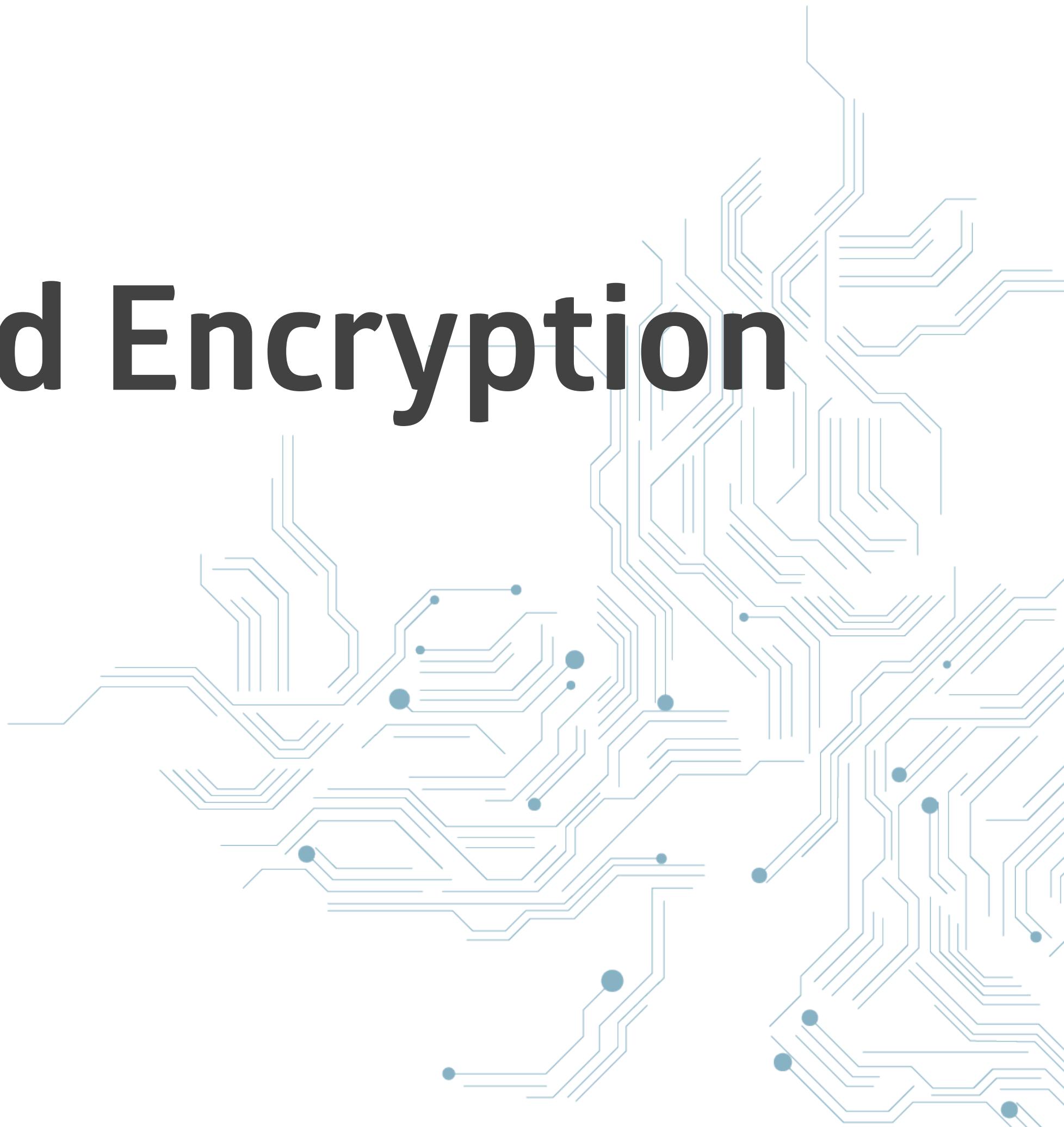
# They provide



# Metrics and Traces



# Transparent End-To-End Encryption



# Advanced Routing



# Istio



# LinkerD



# # 19 - Flux



