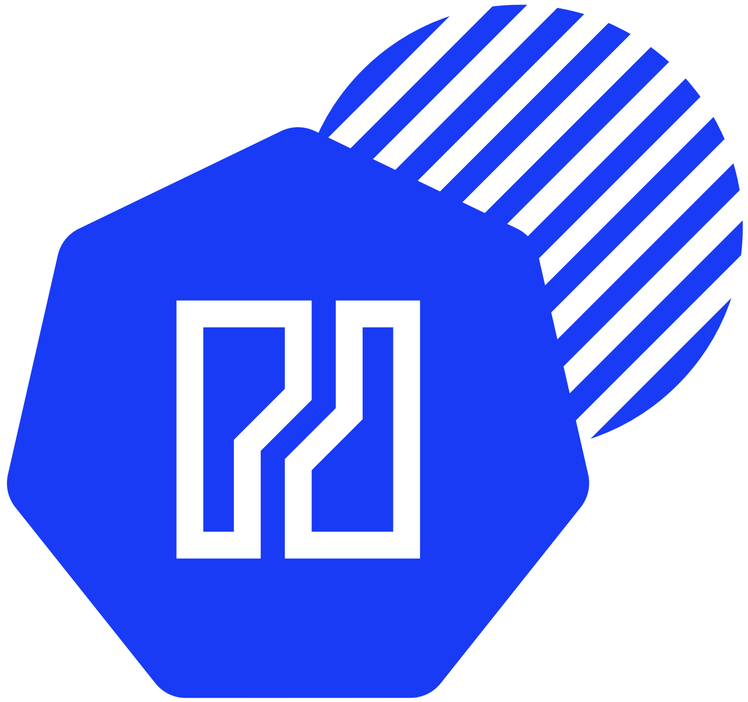
Kubernetes Workshop

Roadmap to Kubernetes



About Natron / Stepping Stone

M1

M2

Kubernetes Basics

Kubernetes Concepts

M3

M4

Kubernetes Communication

Kubernetes Storage

M5

M6

Further Topics

Natron Tech Gmbh
info@natron.io

Introduction to Kubernetes

- Interactive Hands On Tutorials
- Core Concepts
- Helm - Kustomize
- Monitoring
- GitOps

Natron Tech Gmbh
info@natron.io

M1

About us

@janlauber
@natrongmbh

https://natron.io
https://natron.ch

https://stepping-stone.ch

Natron Tech Gmbh
info@natron.io

M2 Kubernetes Basics

- What is Kubernetes?

Kubernetes is a powerful **open-source** system, initially developed by **Google**, for managing **containerized applications** in a **clustered** environment. It aims to provide better ways of managing related, distributed components and services across varied infrastructure.

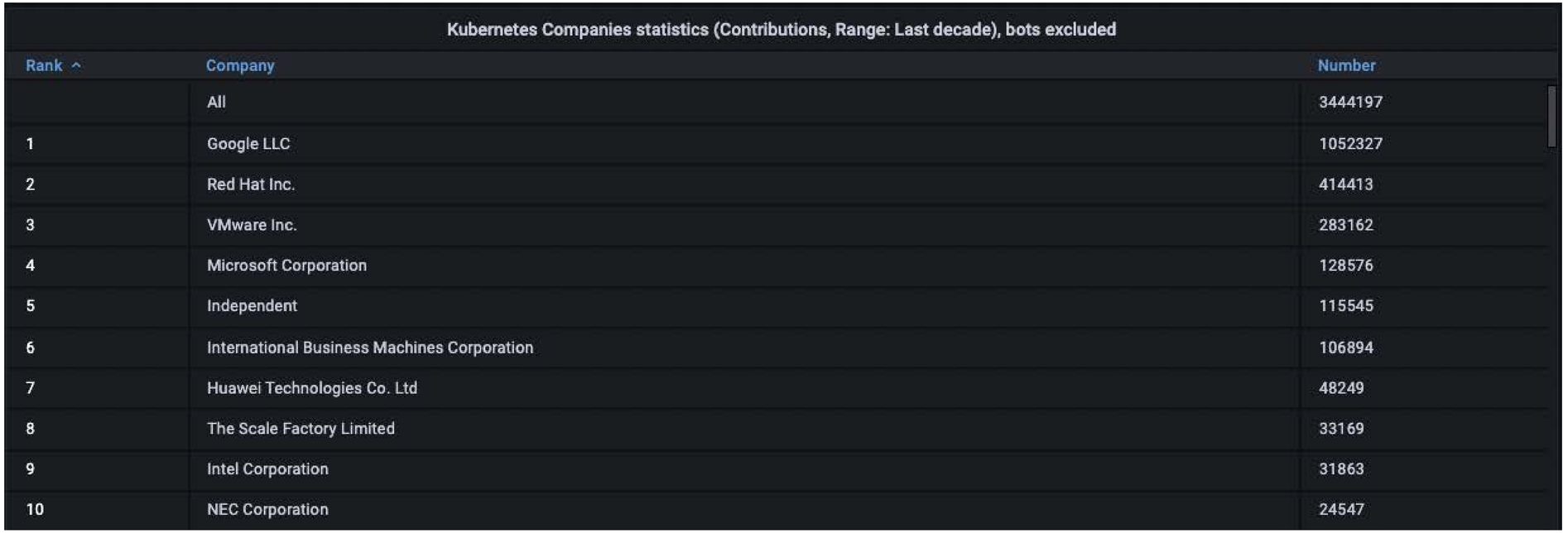
Often seen: **K8S** (**K**uberneteS)

Natron Tech Gmbh
info@natron.io

M2-1 History

- Founded by Google
- Development and design are heavily influenced by Google's Borg system
- **Documentary**: https://www.youtube.com/watch?v=BE77h7dmoQU
- Original code name for Kubernetes within Google was Project Seven of Nine, the « **friendlier** » Borg
- Kubernetes **v1.0** was released on **July 21, 2015**. Google partnered with the Linux Foundation to form the Cloud Native Computing Foundation (**CNCF**)

M2-2 Community Contributions



Kubernetes Companies statistics (Contributions, Range: Last decade), bots excluded		
Rank ^	Company	Number
	All	3444197
1	Google LLC	1052327
2	Red Hat Inc.	414413
3	VMware Inc.	283162
4	Microsoft Corporation	128576
5	Independent	115545
6	International Business Machines Corporation	106894
7	Huawei Technologies Co. Ltd	48249
8	The Scale Factory Limited	33169
9	Intel Corporation	31863
10	NEC Corporation	24547

https://k8s.devstats.cncf.io/d/9/companies-table?orgId=1

Natron Tech Gmbh
info@natron.io

M2-3 Certified Kubernetes

- Consistency
  - when interacting with any installation of Kubernetes
- Confirmability
  - by running identical open source conformance applications
- Timely Updates
  - updates yearly or more frequently

minikube

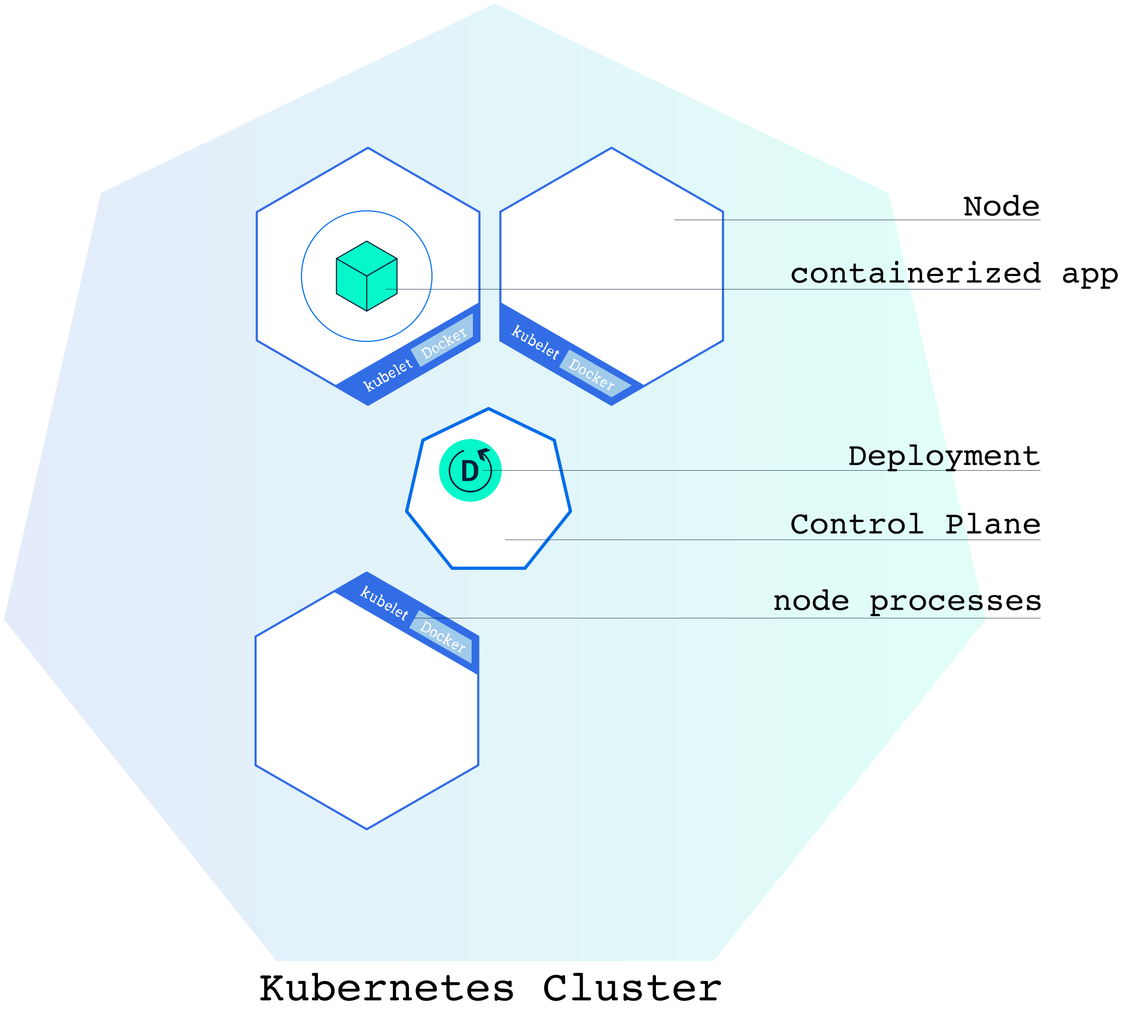
Red Hat OpenShift

K3S

Natron Tech Gmbh
info@natron.io

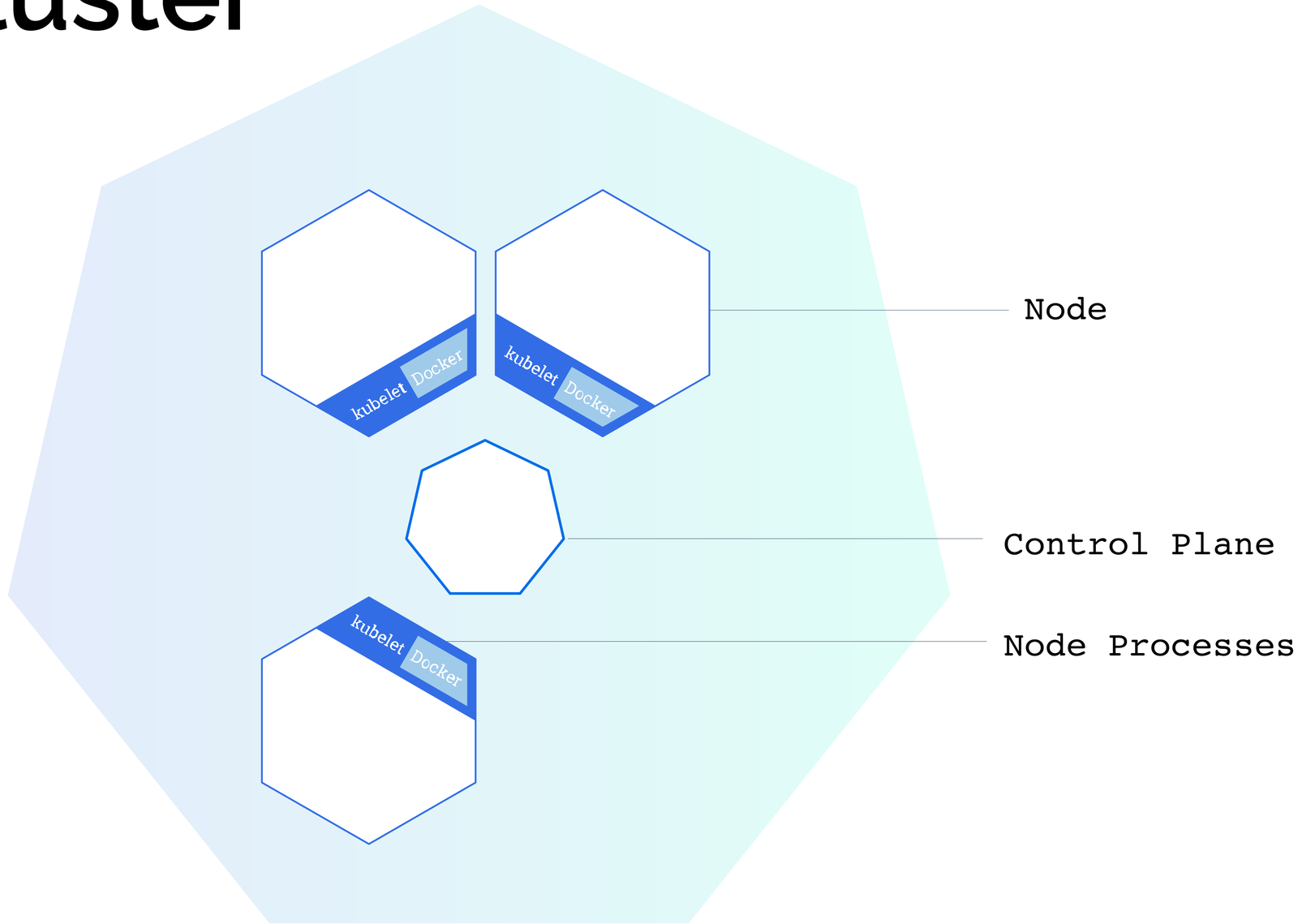
M2-4 What is it?

Kubernetes, at its basic level, is a system for **running and coordinating** containerized applications across a cluster of machines. It is a platform designed to completely manage the life cycle of containerized applications and services using methods that provide **predictability**, **scalability**, and **high availability**.



Natron Tech Gmbh
info@natron.io

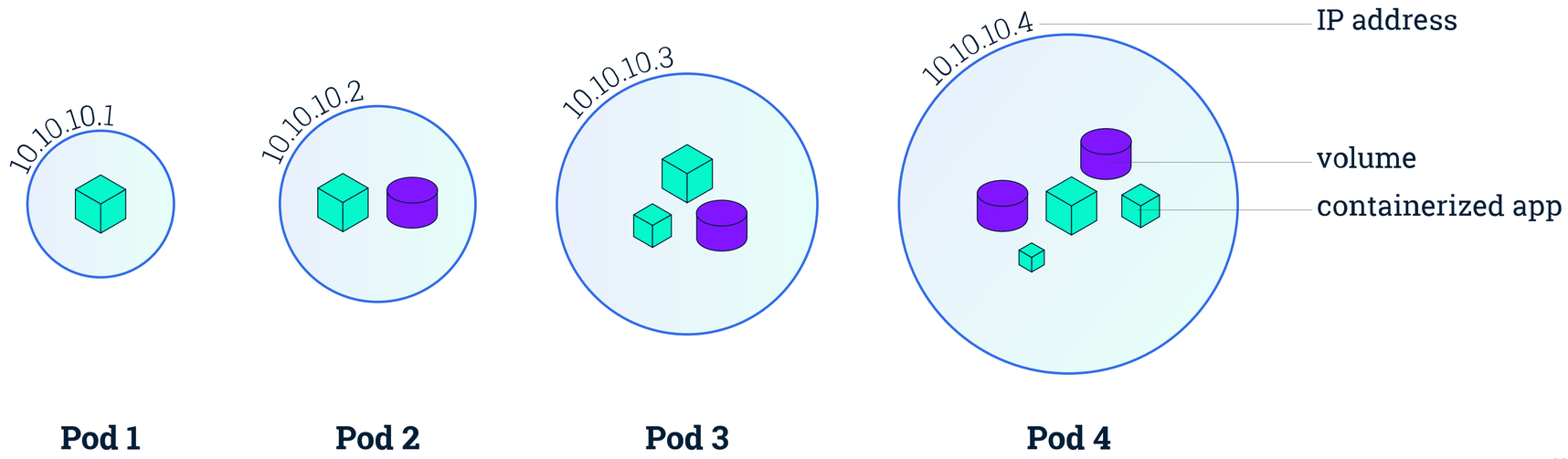
M3 Cluster



Natron Tech Gmbh
info@natron.io

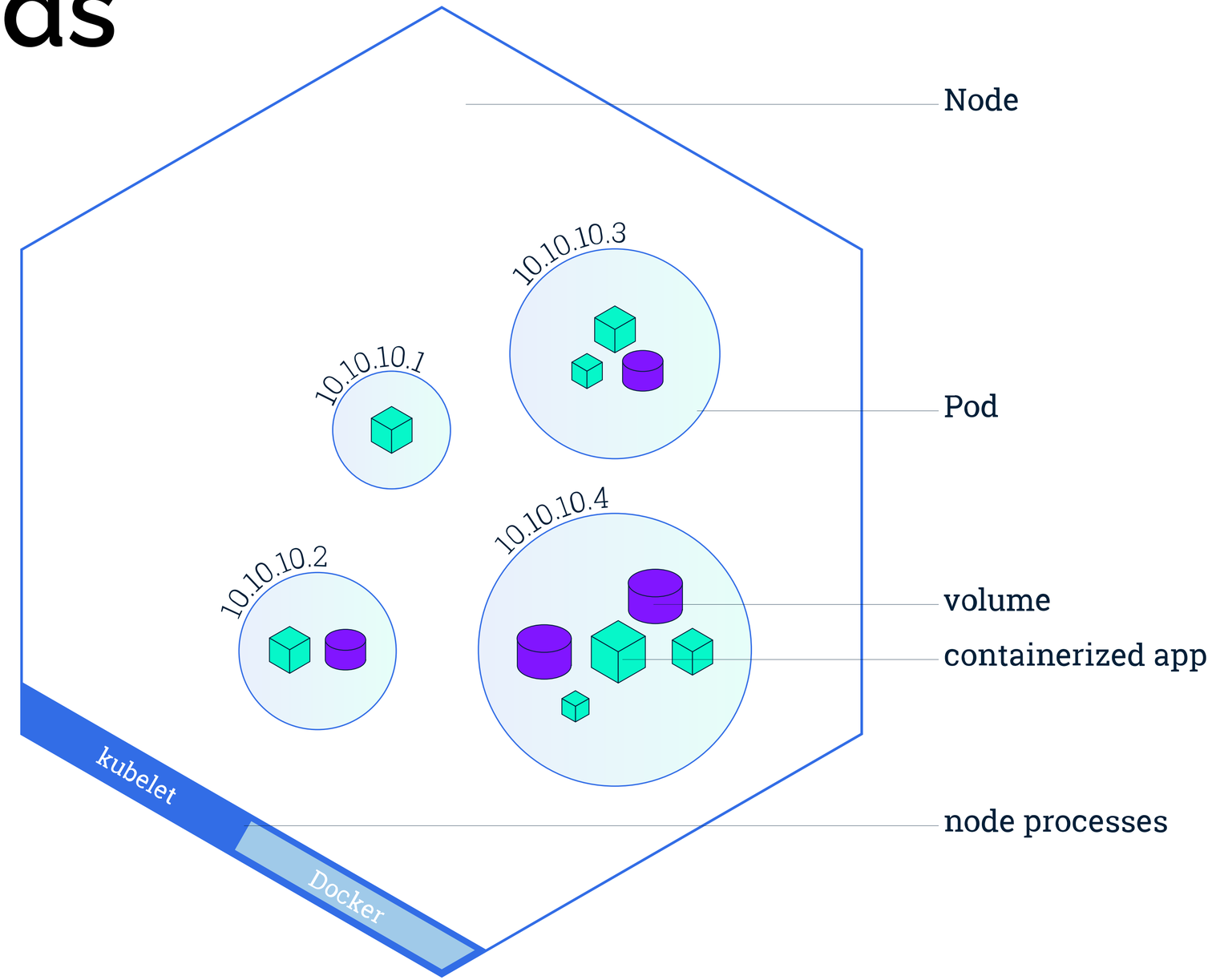
M3-1 Kubernetes Objects

- **Pod** is the smallest deployable unit on a Node. It's a group of containers which must run together. Quite often, but not necessarily, a Pod usually contains one container.
- **Volume** is essentially a directory accessible to all containers running in a Pod.



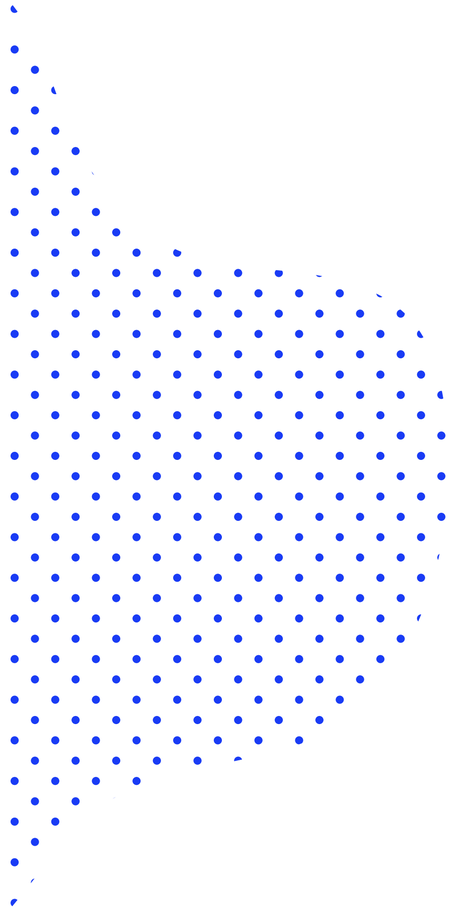
Natron Tech Gmbh
info@natron.io

M3-2 Pods



Natron Tech Gmbh
info@natron.io

M3-2 Pods - YAML



pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    name: nginx
spec:
  containers:
    - name: nginx
image: nginx
```

Natron Tech Gmbh
info@natron.io

M3-3 Replication Sets

- A **ReplicaSet's** purpose is to maintain a stable set of **replica** Pods running at any given time. As such, it is often used to **guarantee** the availability of a specified **number** of **identical** Pods.
- defines **pod template**
- control parameters to **scale replicas**
- scaling **horizontally**

replicaset.yaml

```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
template:
    metadata:
      labels :
        tier: frontend
    spec:
      containers:
      - name: nginx
image: nginx
```

Natron Tech Gmbh
info@natron.io

M3-4 Deployments

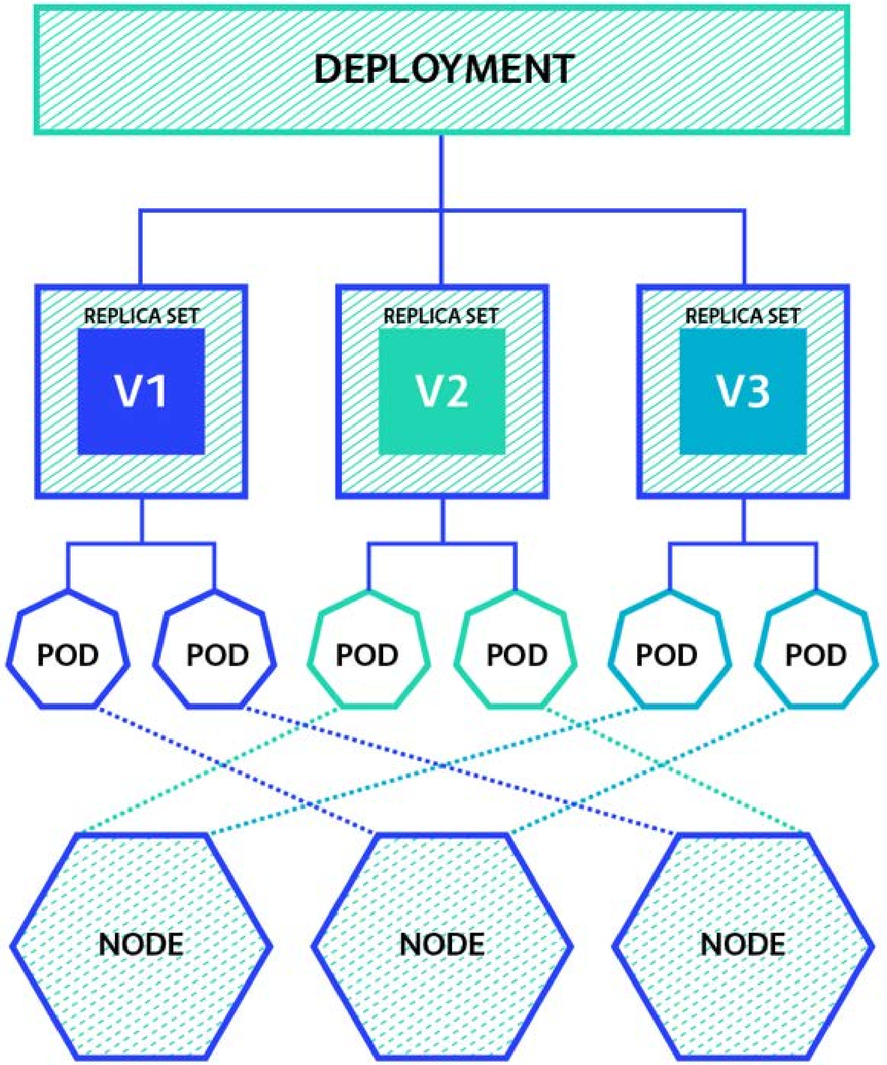
- A **Deployment** provides declarative updates for **Pods** and **ReplicaSets**.
- Deployments use replication sets as a **building block**, adding **flexible** life cycle management functionality to the mix.

deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
replicas: 3
  strategy:
    type: RollingUpdate
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels :
        tier: frontend
    spec:
      containers:
      - name: nginx
image: nginx
```

Natron Tech Gmbh
info@natron.io

M3-5 Deployments & Replicasets



Natron Tech Gmbh
info@natron.io

M3-6 Services

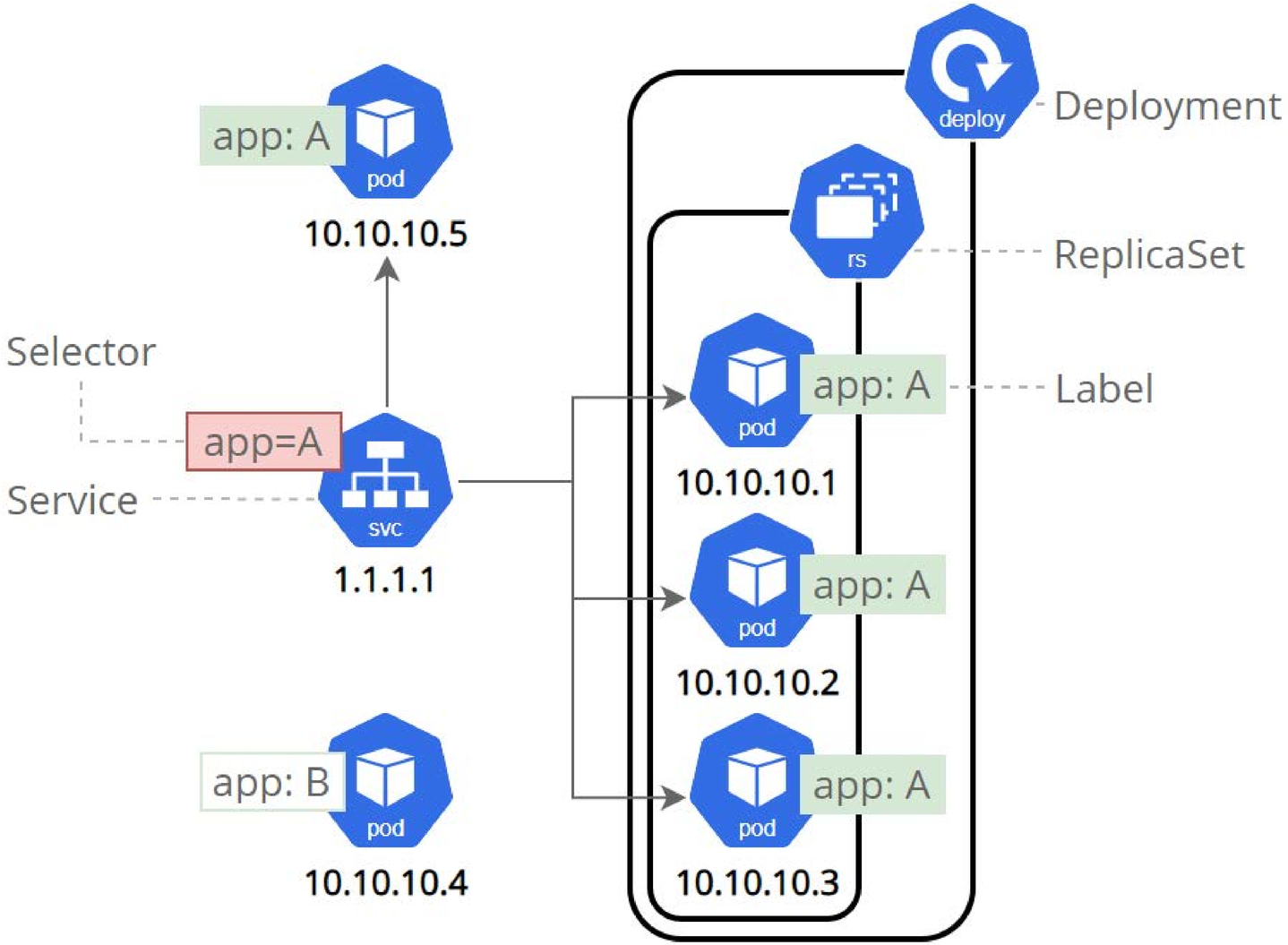
- An abstract way to **expose** an application running on a set of **Pods** as a network service.
- Kubernetes gives Pods their own I**P addresses** and a single **DNS name** for a set of Pods, and can **load-balance** across them.

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: example-service
spec:
  selector:
    app: guestbook
  ports:
    - protocol: TCP
      port: 3000
      targetPort: 3000
```

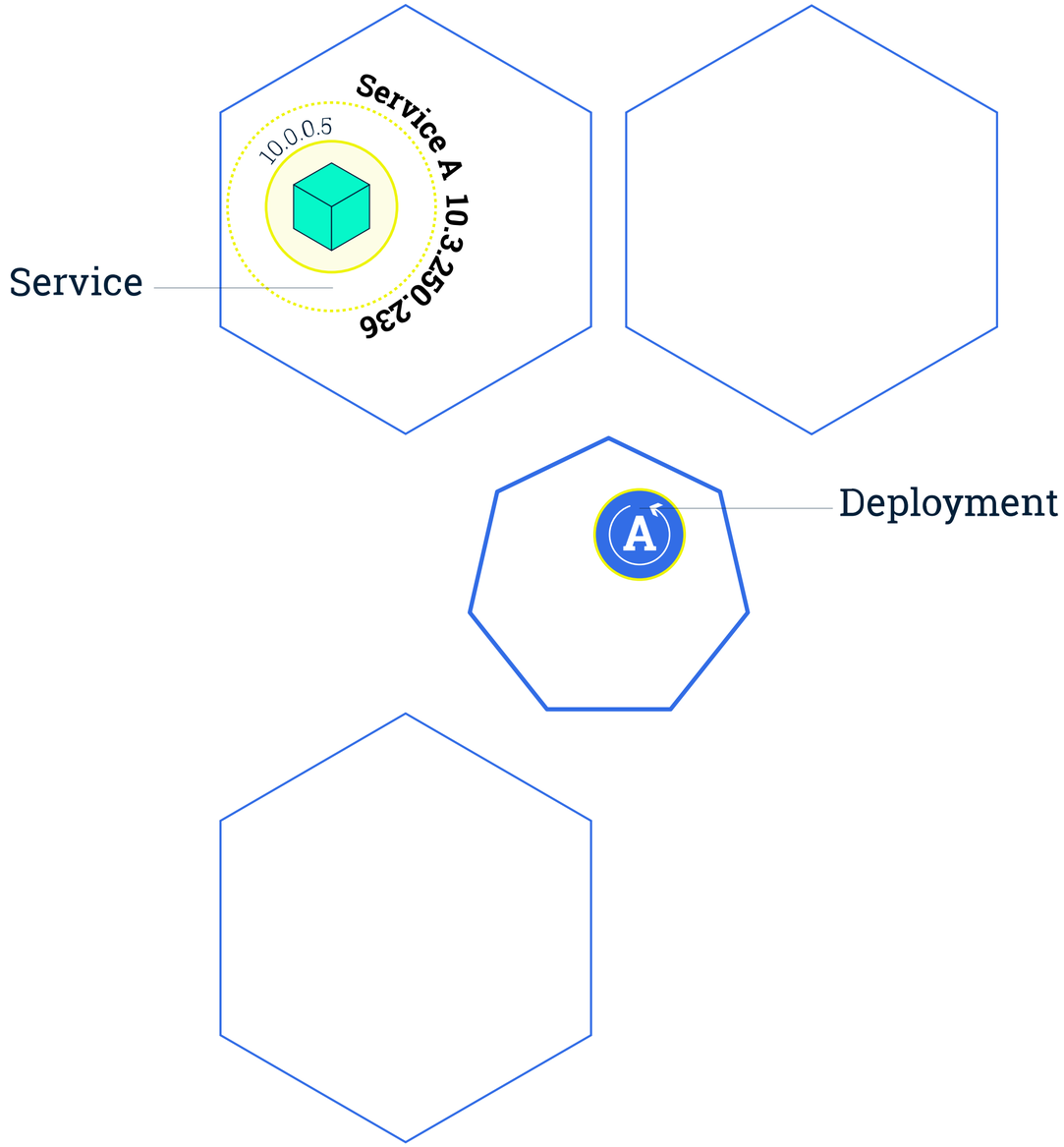
Natron Tech Gmbh
info@natron.io

M3-6 Services Diagram



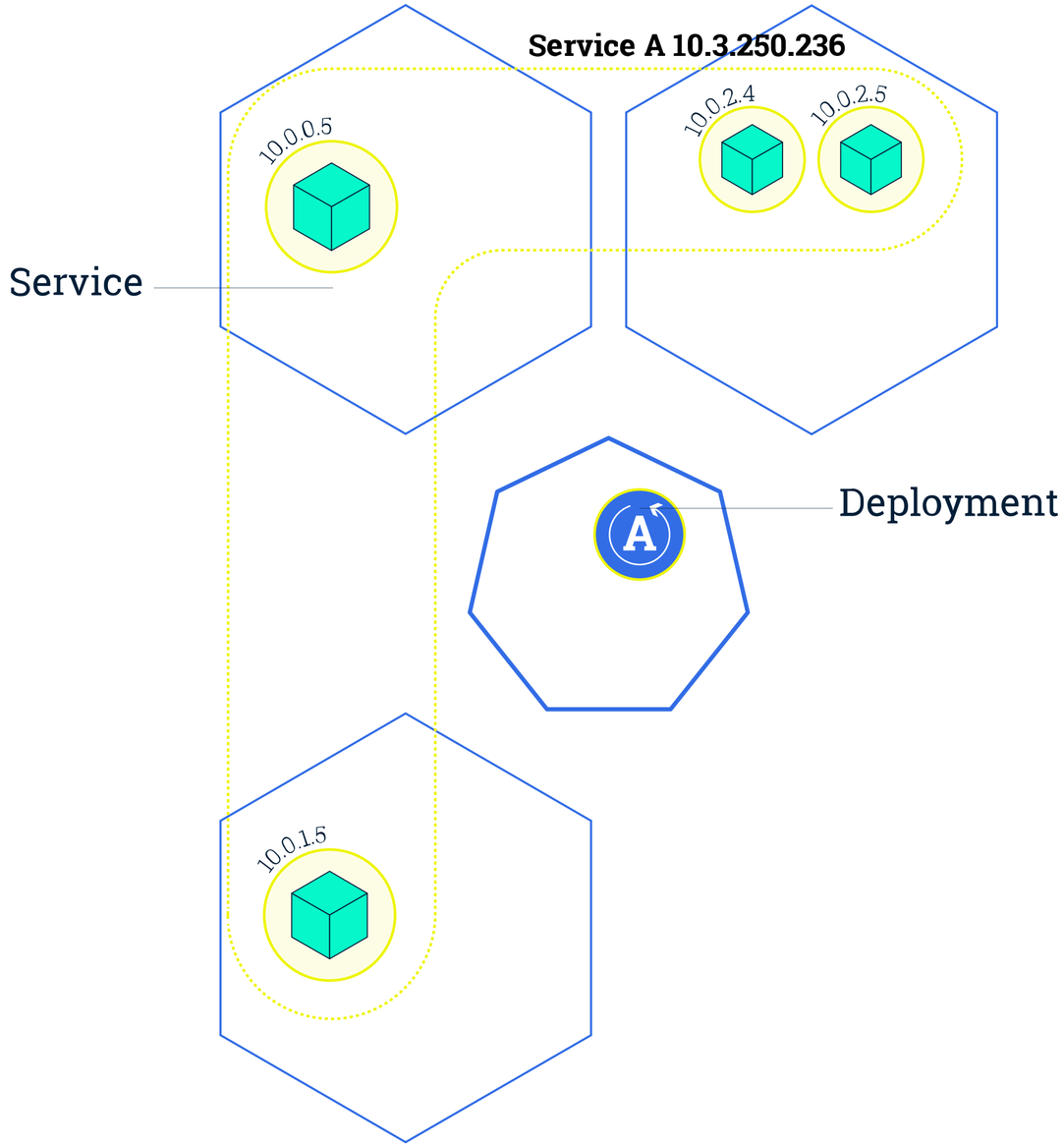
Natron Tech Gmbh
info@natron.io

M3-7 Scaling



Natron Tech Gmbh
info@natron.io

M3-7 Scaling



Natron Tech Gmbh
info@natron.io

M3-8 Ingress

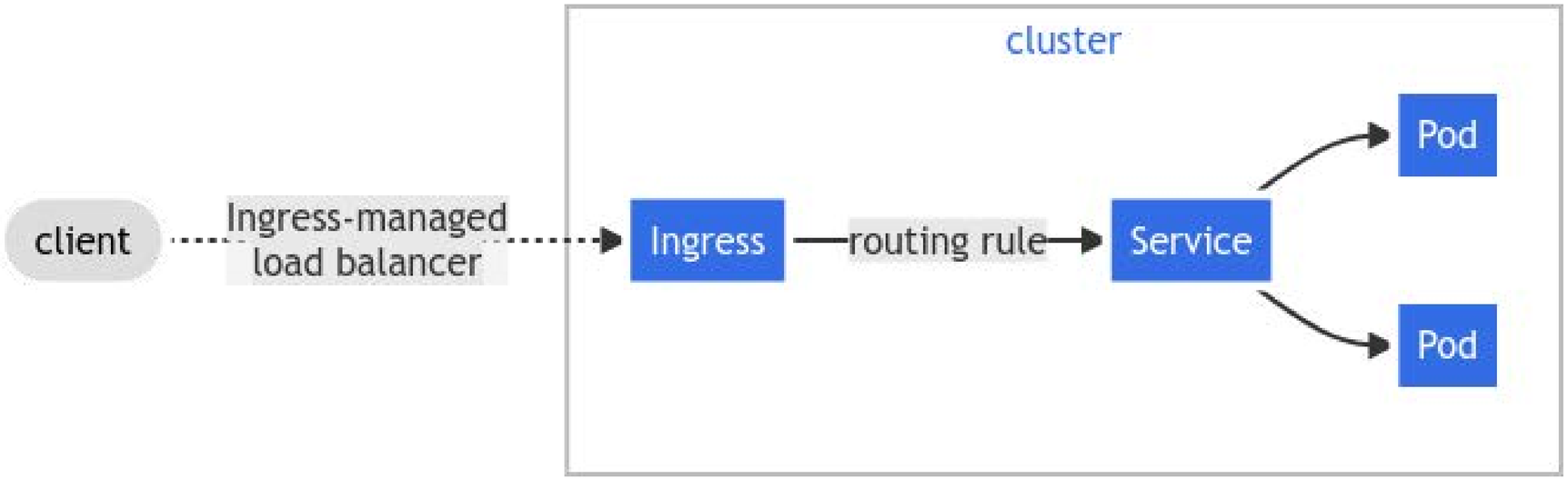
- An API object that manages **external access** to the services in a cluster, typically **HTTP**.
- **Ingress** may provide load balancing, SSL termination and name-based **virtual hosting**.

ingress.yaml

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
spec:
  rules:
  - host: example.com
    http:
      paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: example-service
              port:
                number: 3000
  tls:
  - hosts:
      - example.com
    secretName: example-tls
```

Natron Tech Gmbh
info@natron.io

M3-8 Ingress

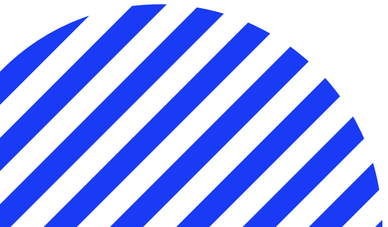


Natron Tech Gmbh
info@natron.io

M3-9 Namespaces

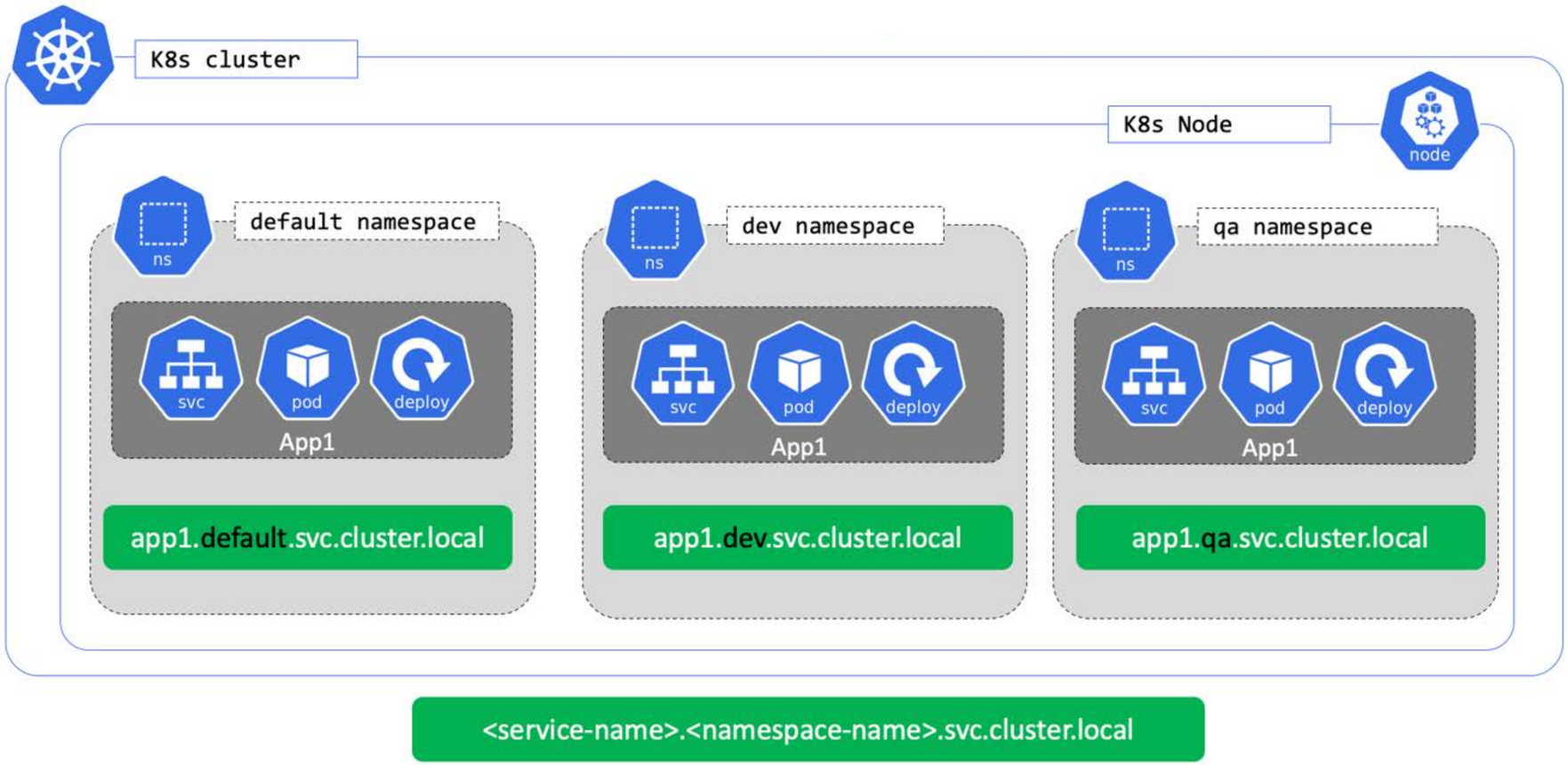
Namespaces provides a mechanism for **isolating groups** of resources within a single cluster.

- Namespaces are intended for use in environments with many users spread across multiple teams, or projects.
- Namespaces provide a scope for names. Names of resources need to be **unique within** a namespace, but **not across namespaces**. Namespaces **can not be nested** inside one another and each Kubernetes resource can only be **in one namespace**.
- Namespaces are a way to divide cluster resources between multiple users (via **resource quota**).



Natron Tech Gmbh
info@natron.io

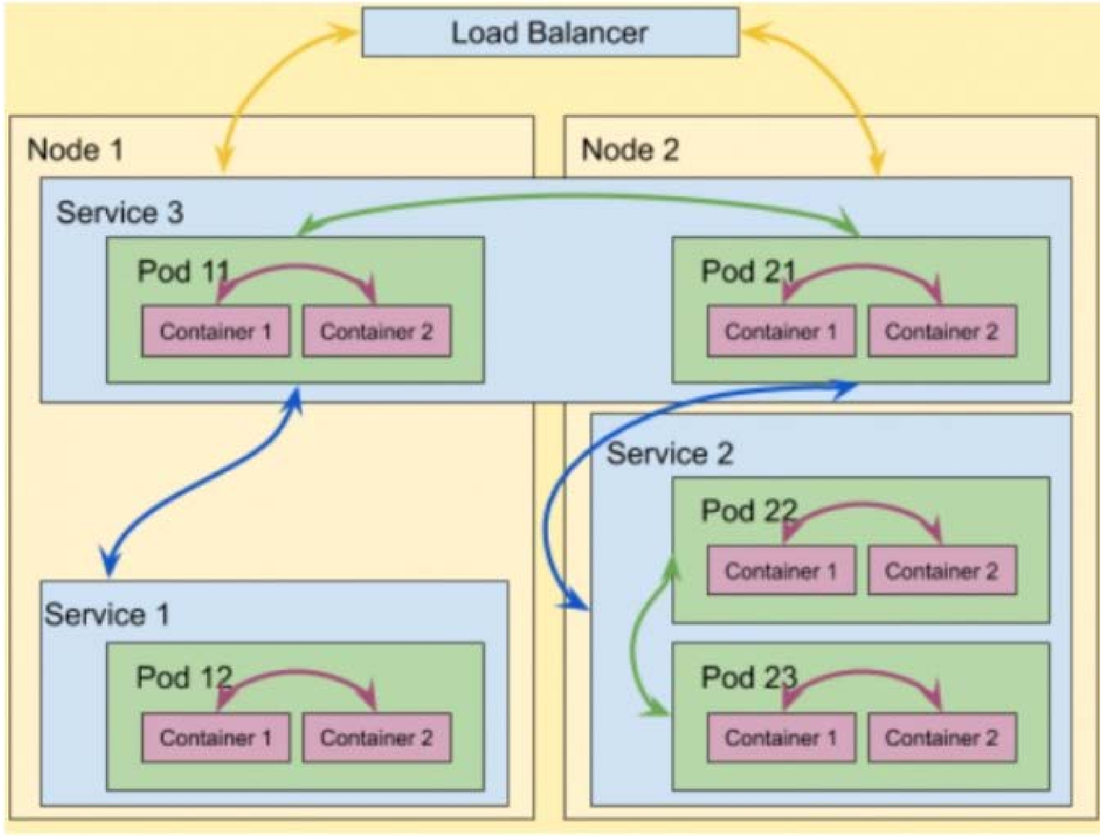
M3-9 Namespaces



Natron Tech Gmbh
info@natron.io

M4 Communication

- **Container** to **Container** communication
- **Pod** to **pod** communication
- **Pod** to **Service** communication
- **External** to **service** communication

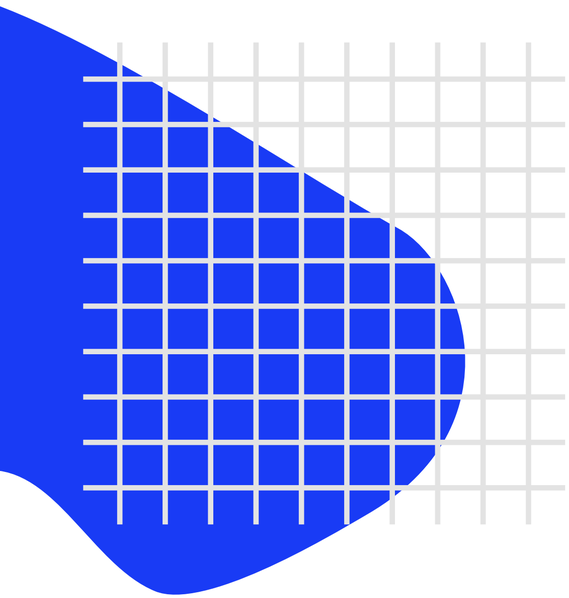


Natron Tech Gmbh
info@natron.io

M4-1 Network Model

- All **containers** can communicate with each other **without NAT**
- All **nodes** can communicate with containers **without NAT**
- The IP address a **container** sees for itself is the **same address** everyone else sees

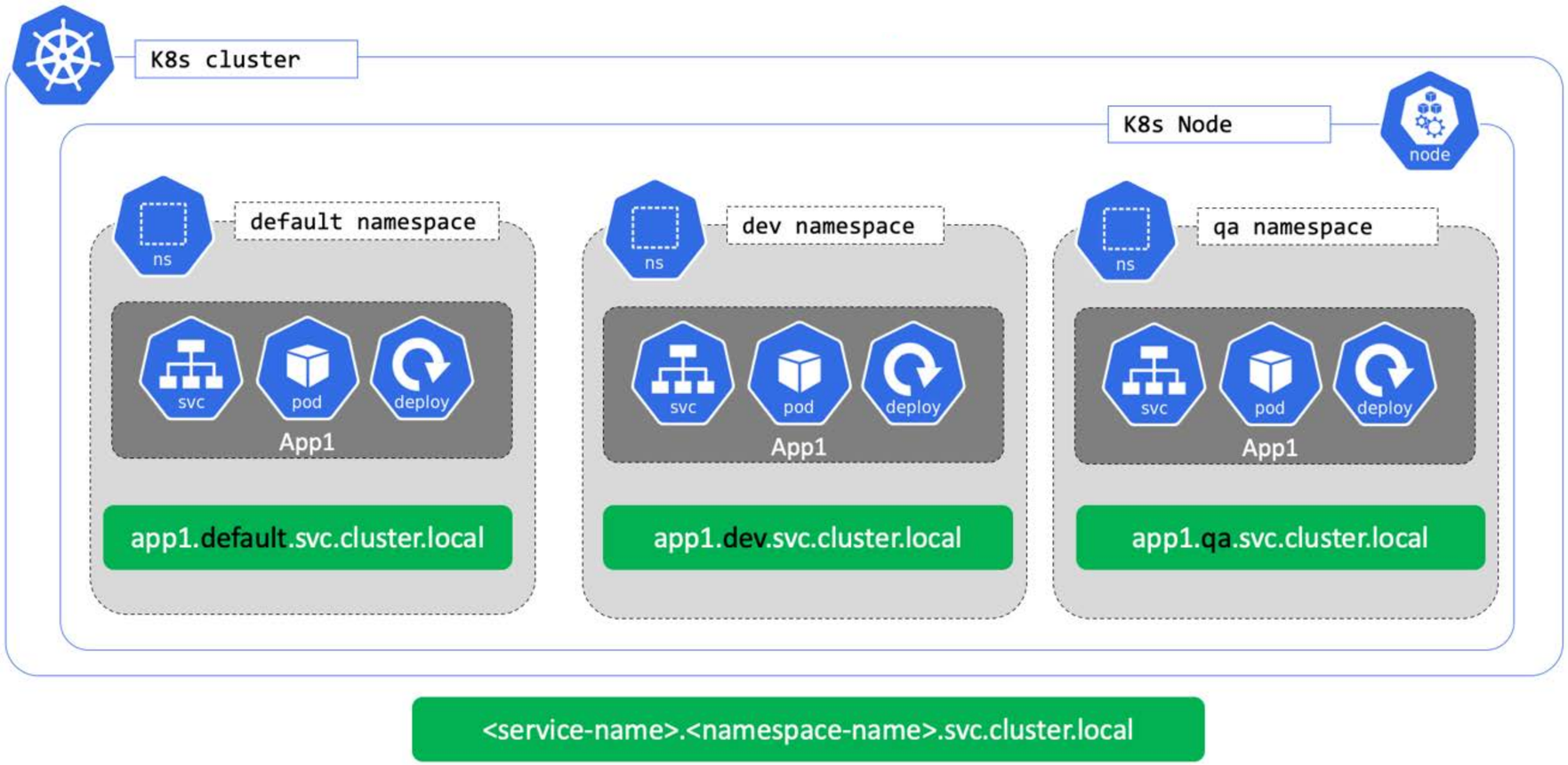
M4-2 Container Network Interface (CNI)



- Kubernetes enables networking through **2** different network plugins:
  - **Kubenet**
  - **CNI**
- **CNI** is only responsible for network connectivity of **containers** and **removing** allocated resources when the container is deleted.
- Initially the **container/pod** has **no network interface**. To connect containers Kubernetes calls the CNI plugin with commands like **ADD**, **DEL**, **CHECK**, **VERSION**.

Natron Tech Gmbh
info@natron.io

M4-3 DNS



Natron Tech Gmbh
info@natron.io

M4-3 DNS

- **“Normal” (not headless) Services are assigned a DNS A or AAAA record**
  - my-svc.my-namespace.svc.cluster.local
- **Pods are assigned a DNS A or AAA record**
  - pod-ip-address.my-namespace.pod.cluster.local
- **Pods exposed by a Service have the following DNS resolution:**
  - pod-ip-address.service-name.my-namespace.svc.cluster.local

Natron Tech Gmbh
info@natron.io

M5 Storage

A **PersistentVolume** (**PV**) is a piece of storage in the cluster that has been provisioned by an administrator or dynamically provisioned using **Storage Classes**. It is a resource in the cluster just like a node is a cluster resource. PVs are **volume plugins** like **Volumes**, but have a lifecycle independent of any individual pod that uses the PV. This API object captures the details of the implementation of the storage, be that **NFS**, iSCSI, or a **cloud-provider-specific storage system**.

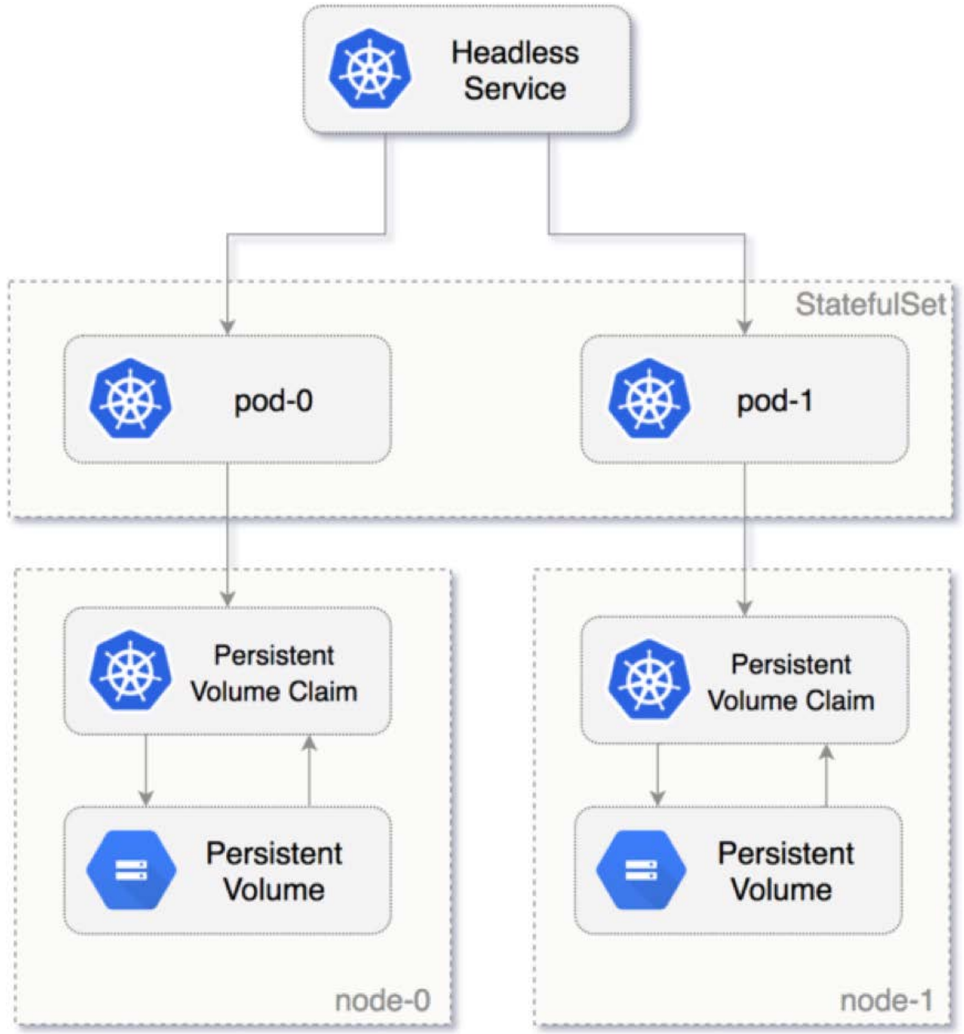
Natron Tech Gmbh
info@natron.io

M5-1 PVC

A **PersistentVolumeClaim** (**PVC**) is a request for storage by a user. It is **similar** to a **pod**. Pods **consume node resources** and **PVCs** consume **PV** resources. **Pods** can request specific levels of resources (**CPU** and **Memory**). **Claims** can request specific **size** and **access modes** (e.g., can be mounted once **read/write** or **many times read-only**).

Natron Tech Gmbh
info@natron.io

M5-2 Overview



Natron Tech Gmbh
info@natron.io

M6 Further Topics

- **Monitoring**
  - Kubernetes Dashboard
  - Kube Prometheus Stack
- **Gitops**
  - Gitlab CI/CD
  - Flux
  - ArgoCD

Natron Tech Gmbh
info@natron.io

M6-1 Image Material

- https://www.weave.works/blog/kubernetes-faq-configure-storage-for-bare-metal-cluster
- https://stacksimplify.com/azure-aks/azure-kubernetes-service-namespaces-imperative/
- https://kubernetes.io/docs/home/
- https://ray.so
- https://undraw.co

Natron Tech Gmbh
info@natron.io

M6-2 Useful Links

- **Official Kubernetes Docs**
  - https://kubernetes.io/docs/home/
- **CNCF Landscape**
  - https://landscape.cncf.io
- **Kubernetes API Reference**
  - https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.24/

M6-3 Get certified

- **Certified Kubernetes Application Developer (CKAD)**
  - https://www.cncf.io/certification/ckad/
- **Certified Kubernetes Administrator (CKA)**
  - https://www.cncf.io/certification/cka/
- **Certified Kubernetes Security Specialist (CKS)**
  - https://www.cncf.io/certification/cks/

Natron Tech Gmbh
info@natron.io

Keep on learning!

@janlauber
@natrongmbh

Natron Tech Gmbh
info@natron.io