

쿠알못을 위한 쿠버네티스 입문

2021. 6. 19

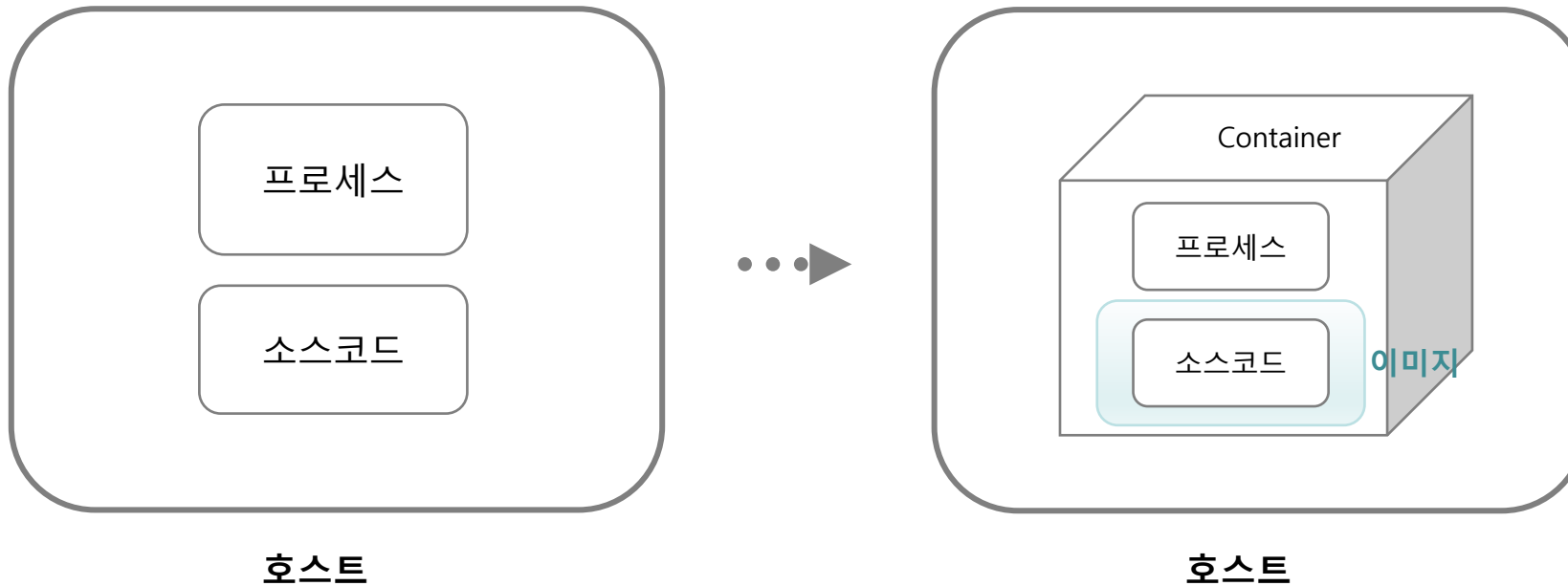
Kubernetes Korea Group 안승규

Container

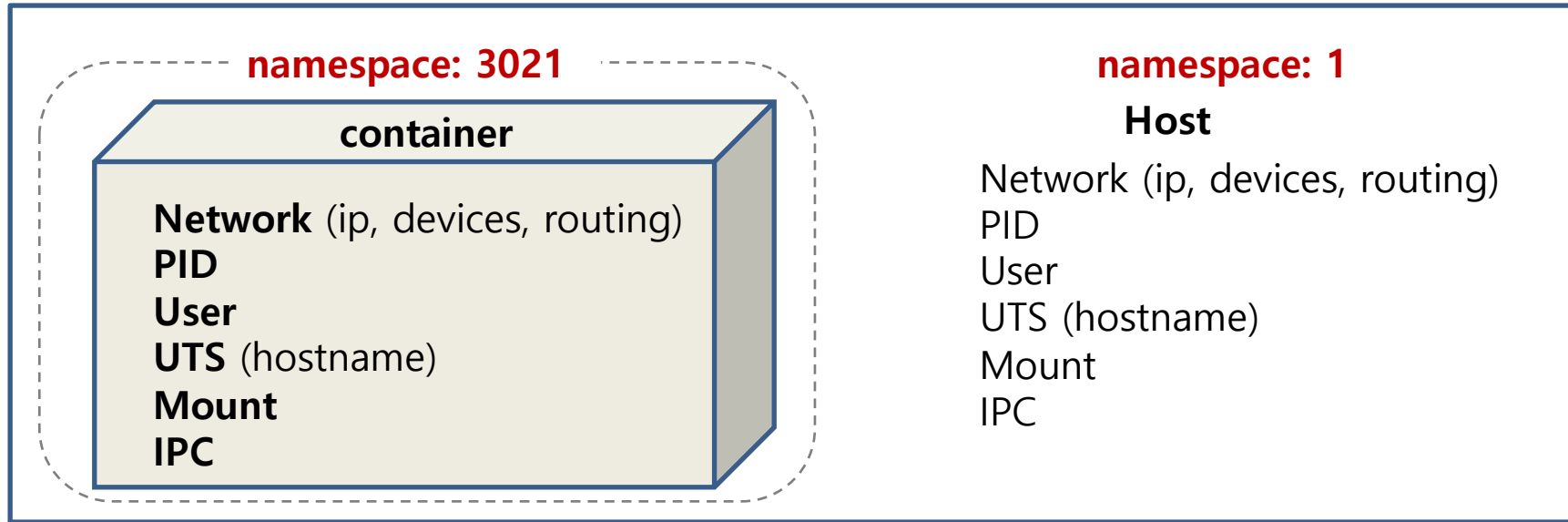
호스트에서 실행되는 소스코드, 프로세스 형태를 관리하기 쉬운 구조로 변경
소스코드는 이미지로 패키징되어 호스트에서 isolation 프로세스로 실행

필요 기술

자원 분배(cgroup), 격리(namespace)



Container



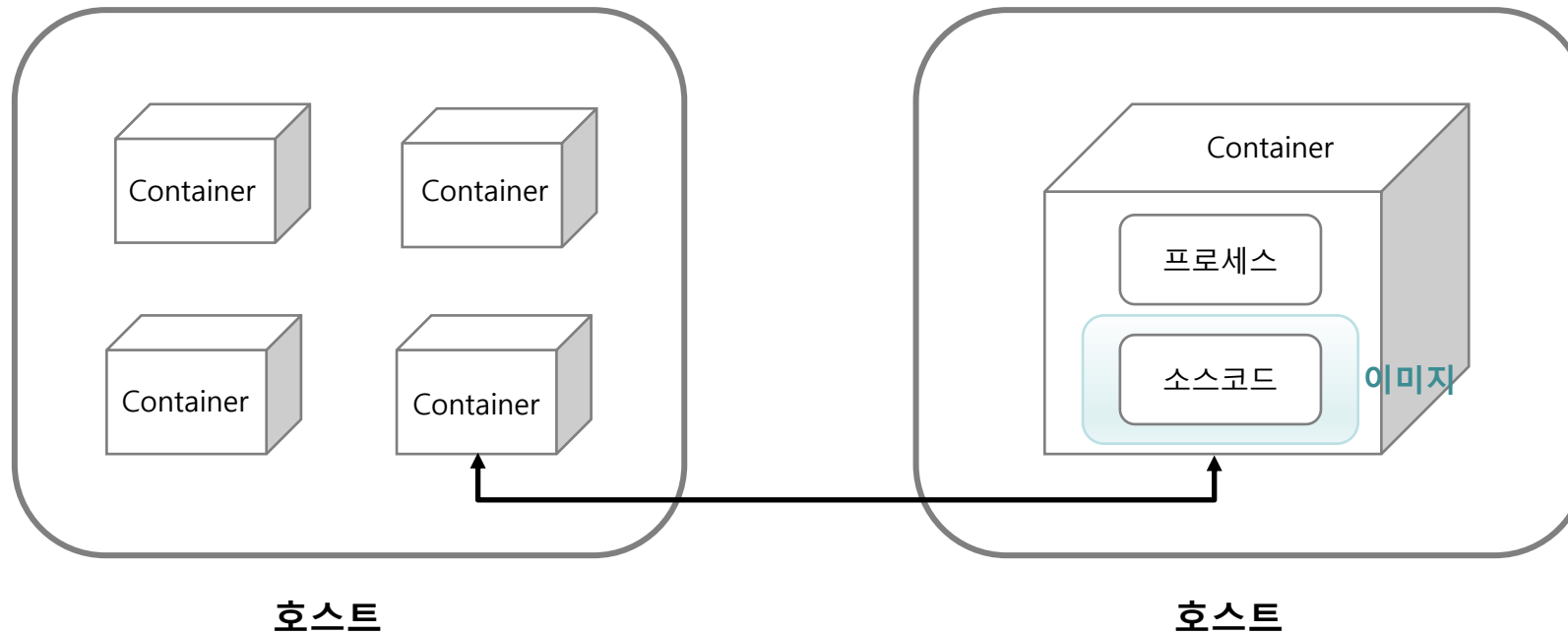
- cgroups 와 namespace 로 구분
- 프로세스가 독립적인 net(ip, devices, routing, port), process id, uid, hostname, mount 을 있음
- net: 독립적인 ip 어드레스, 네트워크 디바이스, port
- process id: 독립적인 process id 를 가짐
- uid: 독립적인 user id 를 가짐
- hostname: 독립적인 호스트네임을 가짐
- mount : host 의 디렉토리를 이용하거나 tmpfs (가상 메모리(vm) 파일 시스템) 를 이용

Container

하나의 호스트 내에서도 멀티 컨테이너가 독립적으로 고립되어 실행될 수 있고,
멀티 호스트에서 멀티 컨테이너가 실행될 수 있음

필요 기술

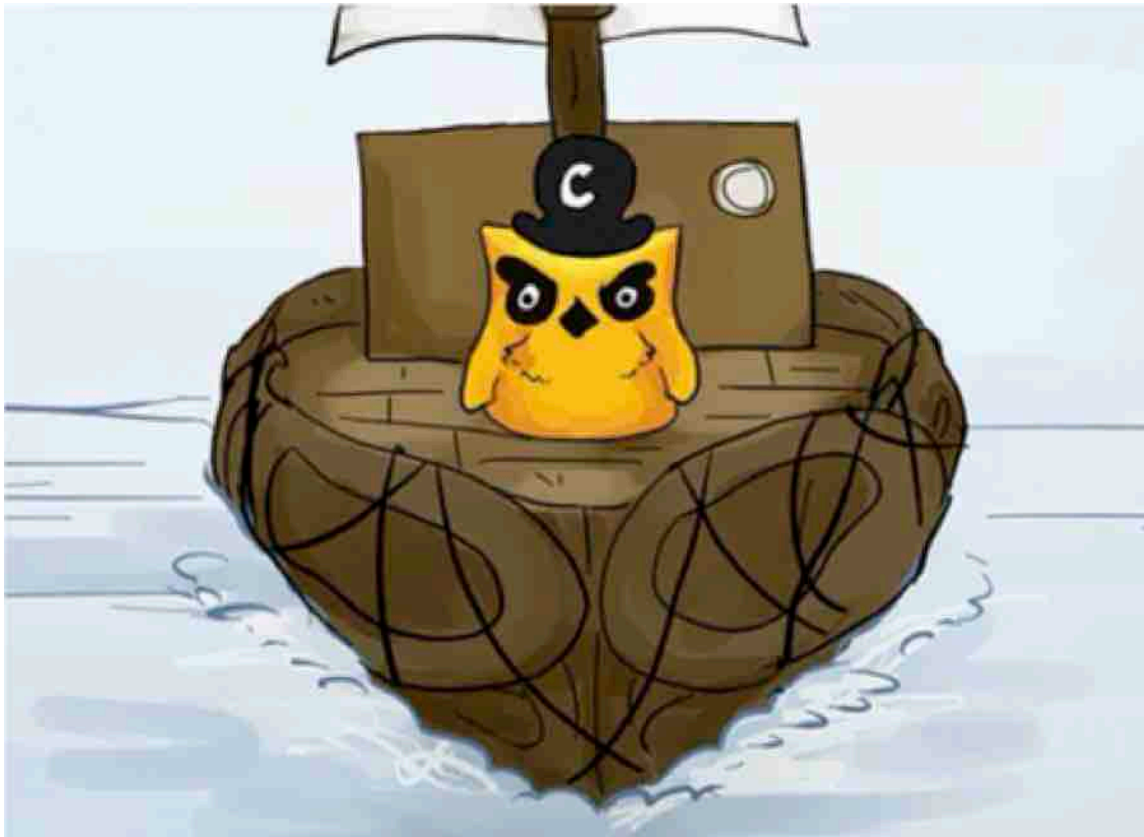
스케줄링과 배치(Placement), 관리(실행,삭제,재실행), 네트워크 통신, Scale out, 스토리지(Volume)



Kubernetes 란?

- Containerized App을 자동 배포, Scaling, 관리하는 오픈 소스 (Apache License 2.0)
- 그리스어: κυβερνήτης [kyvernitis], 뜻: helmsman or ship pilot
- 줄여서 k8s (k 와 s 사이 8글자)
- Google 의 Borg (컨테이너 관리 s/w) 에서 영감 받음
- 2015년 7월 21일 정식 버전 1.0 Release (By Google)
이후 CNCF (Cloud Native Computing Foundation) 에 기증
- Joe Beda, Brendan Burns, Craig McLuckie 가 시작
- CNCF 프로젝트는 Graduate, Incubator, Sandbox 프로젝트로 구성
- CNCF 최초의 graduated 프로젝트
- 3개월에 stable 버전 Release
- 이전 버전은 v1.21.0 (2021년 4월 8일)
- 최신 Release 버전은 v1.21.1 (2021년 5월 13일)

The Illustrated Children's Guide to Kubernetes



Captain Kube



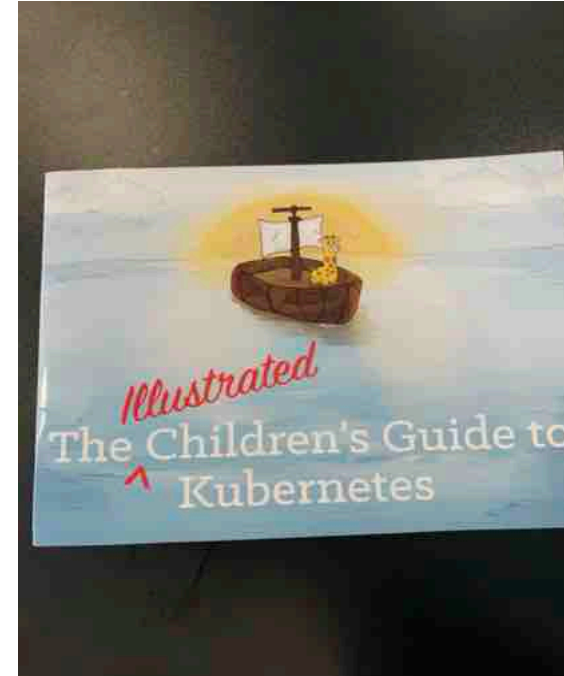
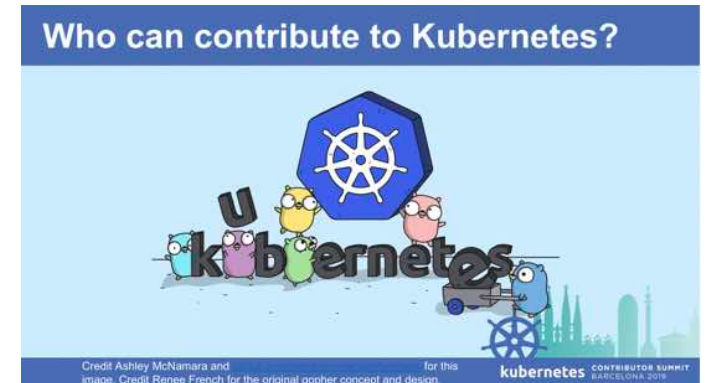
Phippy

- 컨테이너 배포 / 실행 (Immutable, Declarative, Scheduling)
- Service discovery & Load balancing (Internal DNS)
- Secret & Configuration
- Self-repairing (Reconciliation, Readiness probe, Liveness probe)
- Horizontal scaling (Reconciliation)
- Automated rollout & rollback (History)
- Storage orchestration (block storage)
- Batch execution

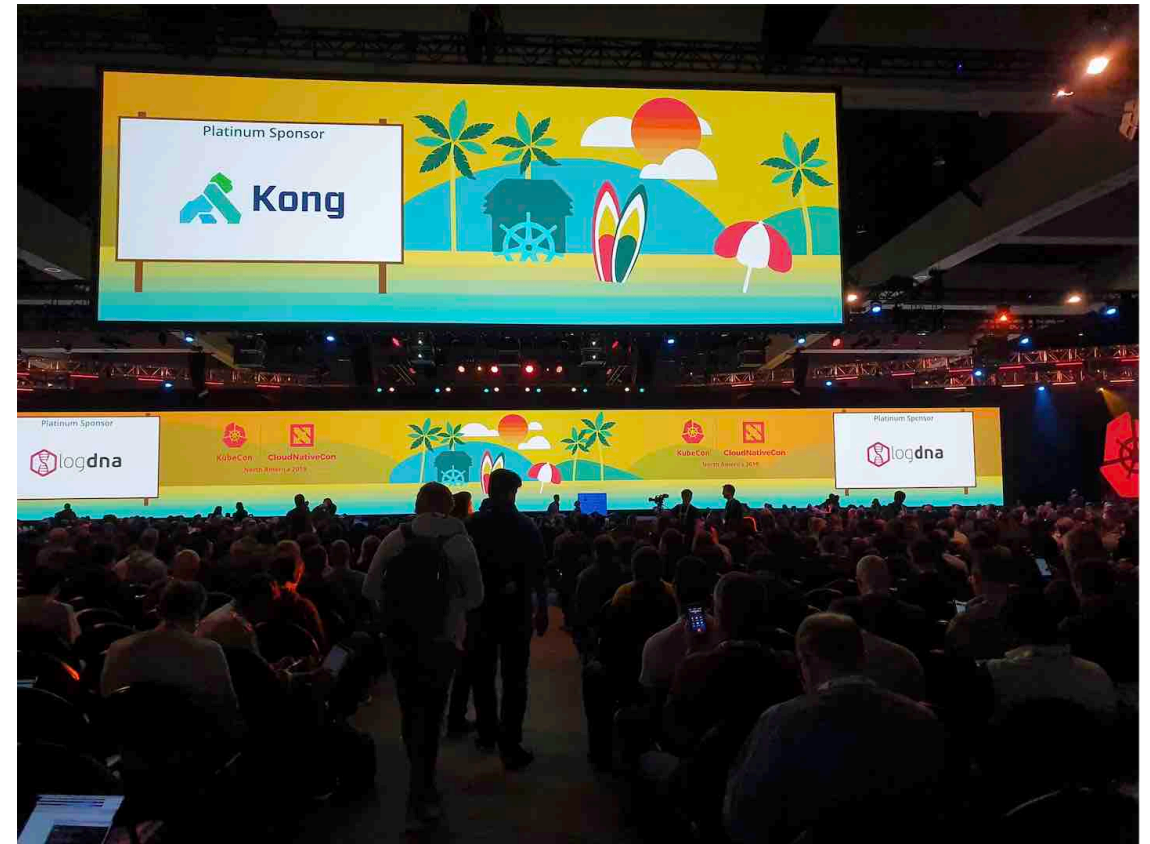
Cloud Native Application 으로 개발된 고객의 서비스에 대한 Reliability, Scalability, Resilience 를 제공

- Declarative deployment (yaml) -> 장점???
- API 확장 (Adminssion Controller)
- Immutable Infrastructure (Software as a Service)
- Custom Controller (CRD – Custom Resource Definition)
- Ecosystem

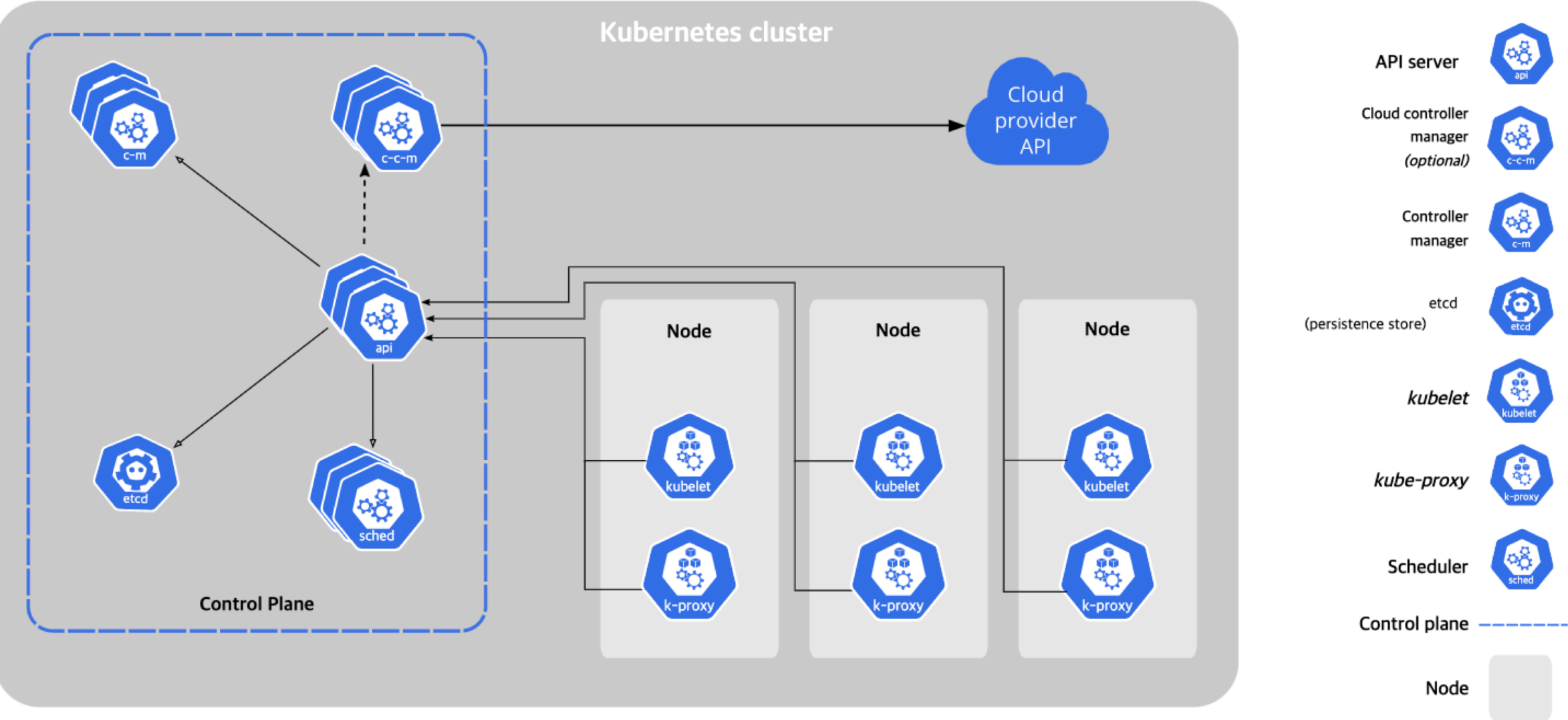
KubeCon 2019 Barcelona



KubeCon 2019 San Diego



Kubernetes Architecture



쿠버네티스는 네트워크 통신 관련 컴포넌트는 third party 에 맡기며, CNI 만 지원
테스트용 네트워크 컴포넌트는 있으나 운영에 적합하지 않음

Scheduler

- 컨테이너를 배치(Placement)
- 호스트를 그룹으로 묶어서 컨테이너 배치를 지정할 수 있음

Controller

- Scale out
- 이것들을 모두 통틀어서 Reconciliation (desired state == current state)

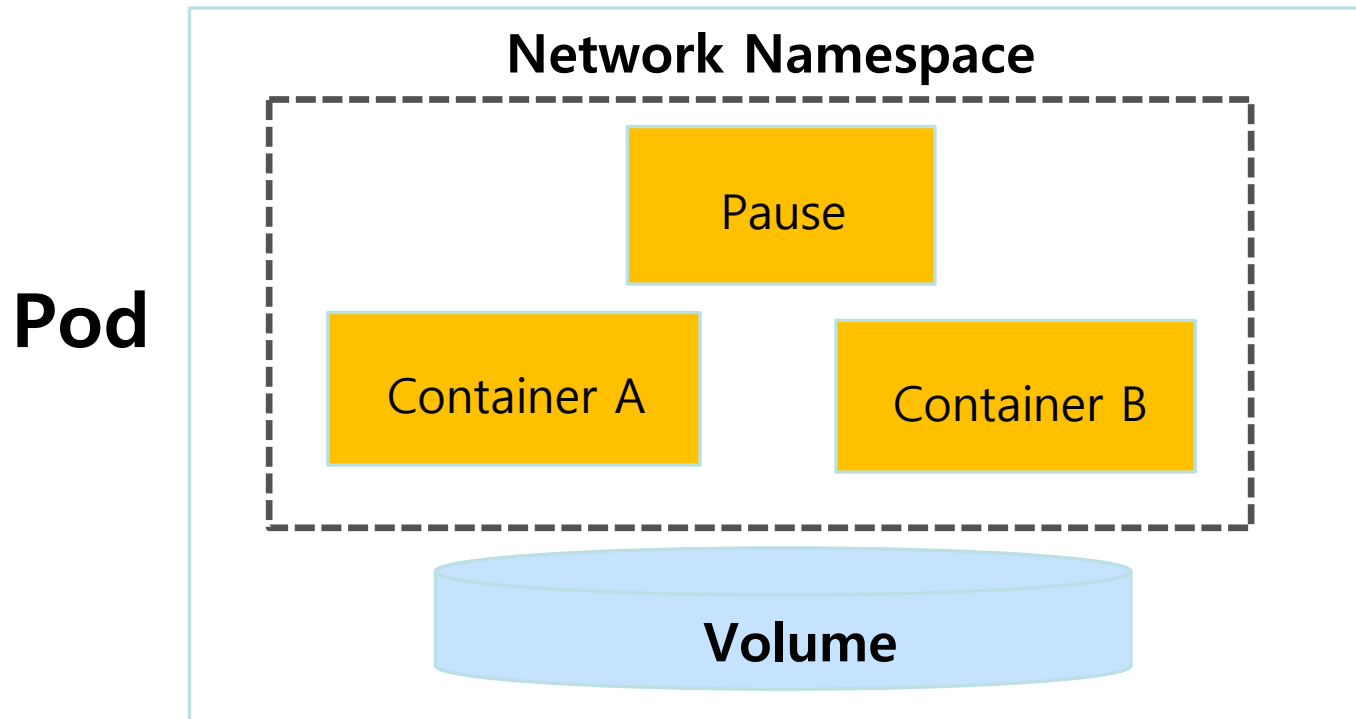
Kubelet

- 컨테이너를 실행, 삭제, 재실행
- 네트워크 모듈과 컨테이너를 연결 (CNI plugin)

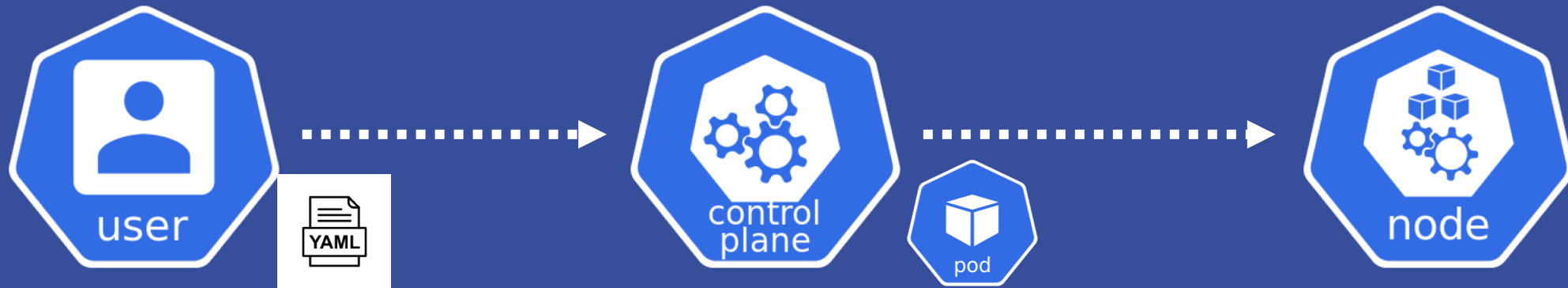
- **Namespace**
- **Pod**
- **Deployment**
- **StatefulSet**
- **Service / Endpoint**
- **ConfigMap / Secret**
- **PV / PVC**
- **Custom Resource Definition / Custom Resource**

POD

- Container 의 묶음으로 스케줄링 되는 단위
- Pod 단위로 Network Namespace 와 IP 를 가짐 (Container Network Interface - CNI by CoreOS)
- Pod 안의 컨테이너는 같은 Volume 을 볼 수 있음
- Pod 안에는 Pause 라는 Infra container 가 기본으로 실행됨

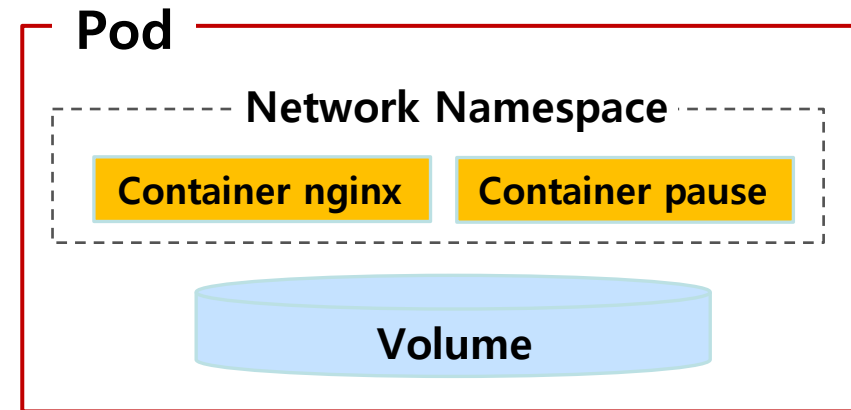


Pod 배포



Pod 배포

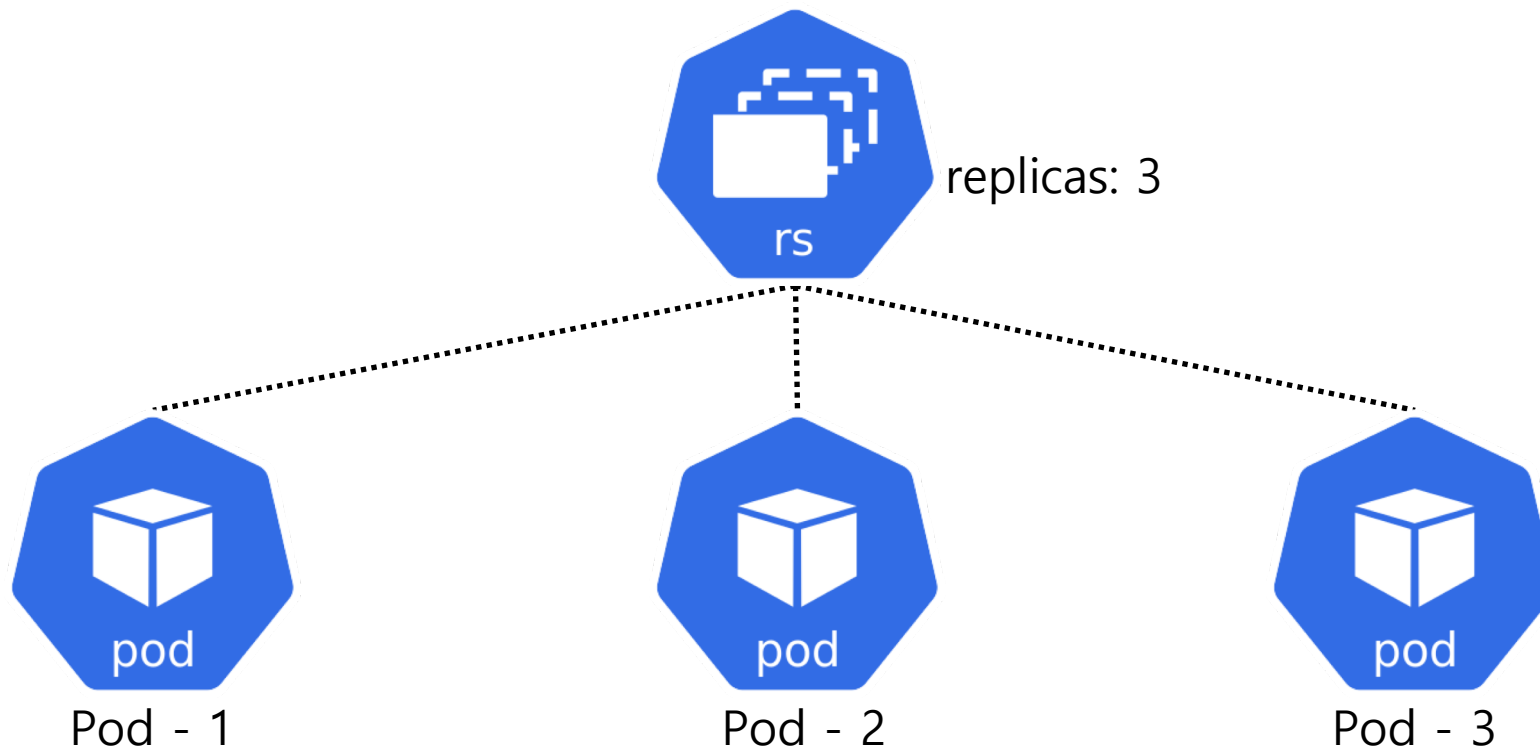
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  namespace: default
spec:
  containers:
  - name: nginx
    image: nginx:1.21.0
    ports:
    - containerPort: 80
```



Node1

ReplicaSet

- ReplicaSet = Pod + replicas 수 (Pod 개수)
- Pod 의 개수를 관리
- current state 개수와 desired state 개수가 같을 때 까지 관리 (실행 중인 Pod 의 수를 항상 보장)
- Horizontal Pod Autoscaler가 autoscale을 할 때 ReplicaSet 활용

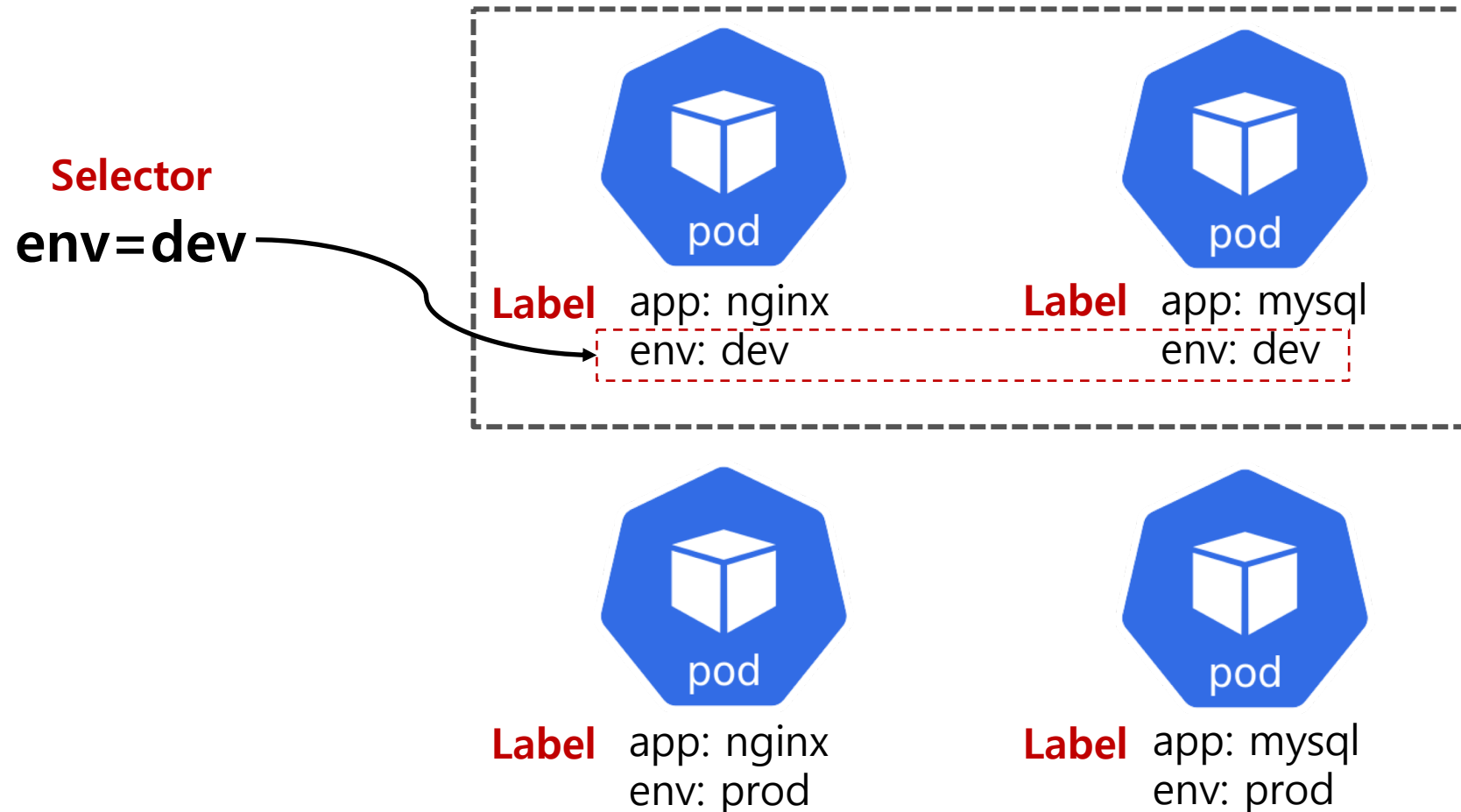


Deployment

- Deployment = ReplicaSet + History (Revision)
- Pod 배포에 대한 버전 관리가 가능
- spec 과 spec.template.spec 으로 정의

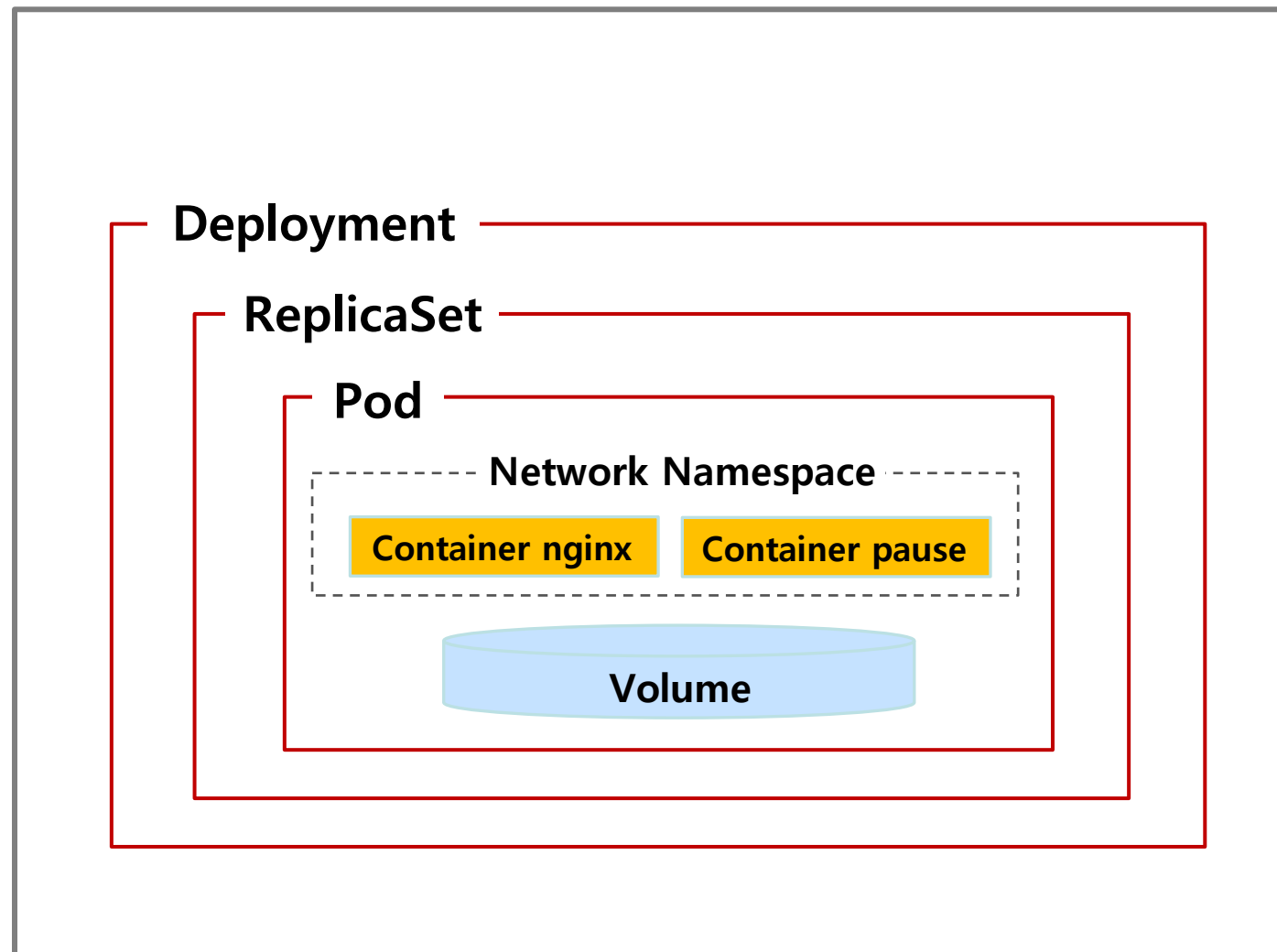
Labels & Selector

- Key: value 형태로 Kubernetes Object 에 붙힐 수 있음
- Pod, Deployment, Service, Node, Namespace 등



Deployment 배포 / 실행

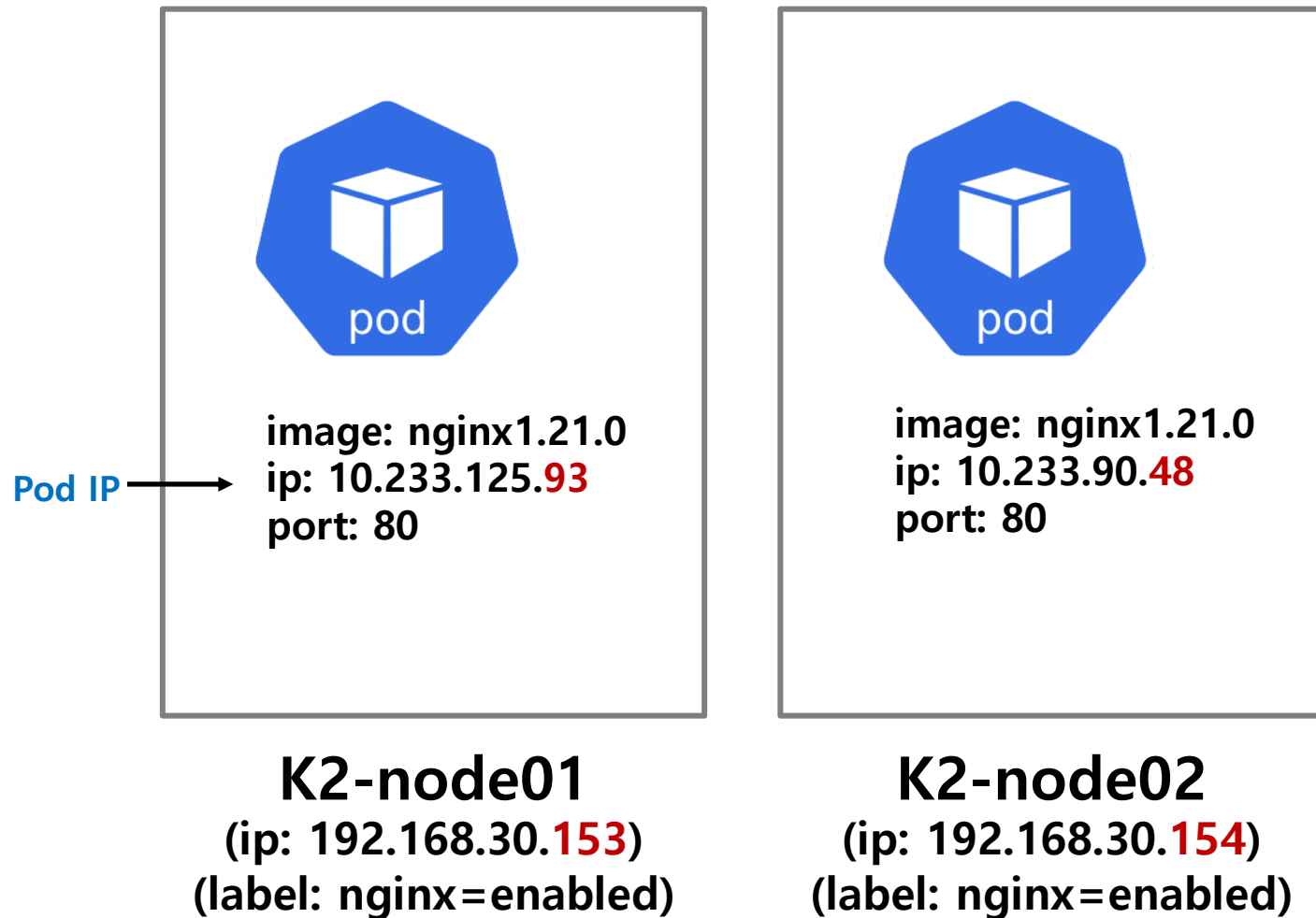
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.0
          ports:
            - containerPort: 80
      nodeSelector:
        nginx: enabled
```



K2-node01 (nginx=enabled)

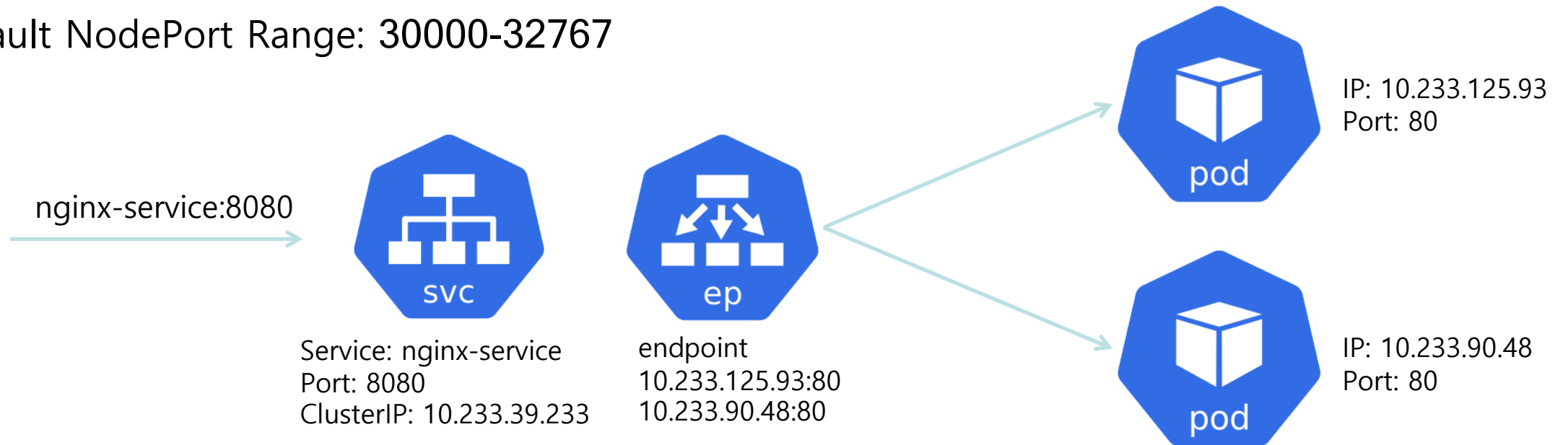
Deployment 배포 / 실행

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  namespace: default
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.21.0
          ports:
            - containerPort: 80
      nodeSelector:
        nginx: enabled
```



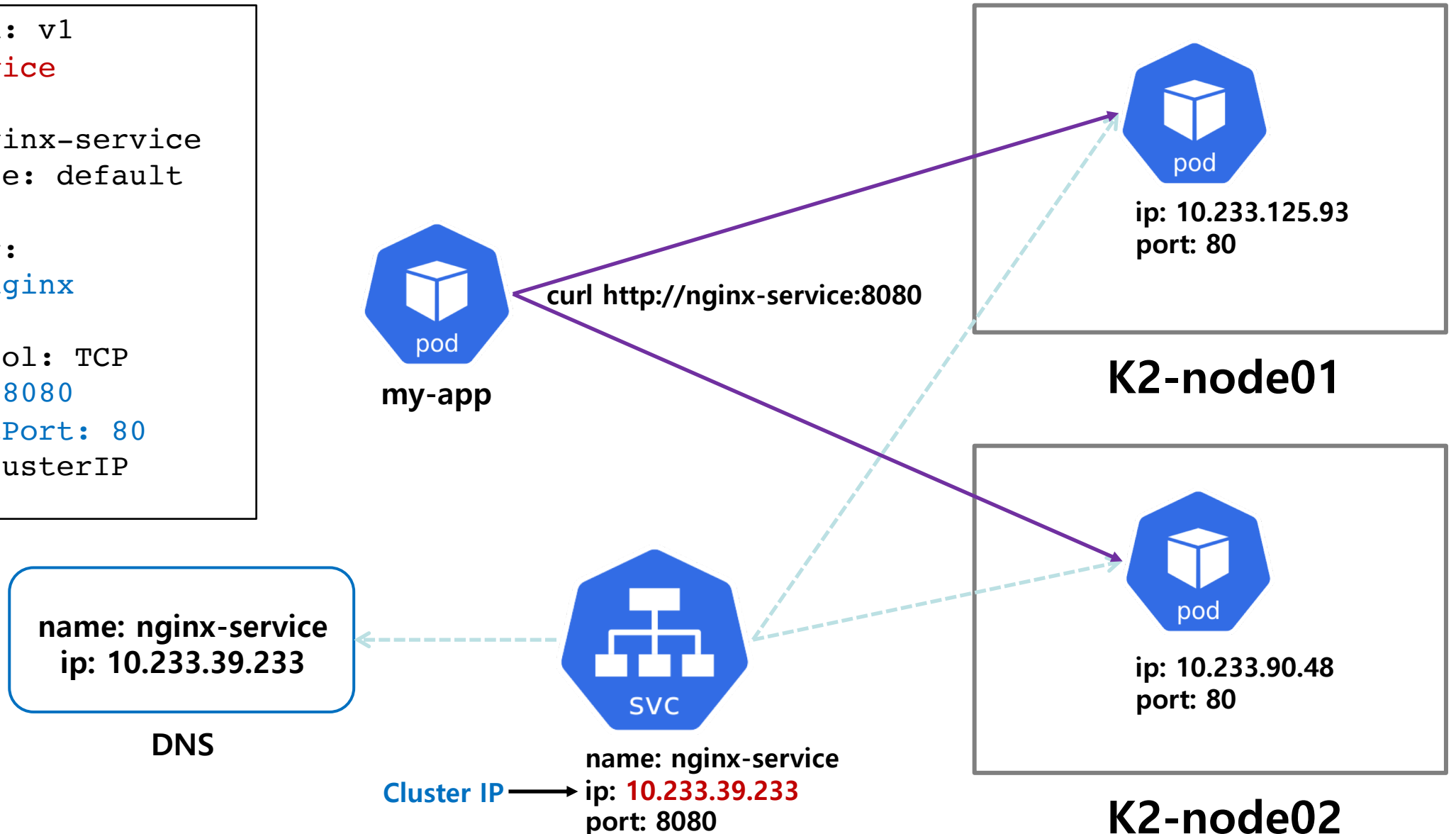
Service / Endpoint

- IP 주소(ClusterIP 로 Virtual IP 임)와 DNS 명이 생성되어 내부 DNS 서비스에 등록됨
- DNS 이름으로 (서비스 이름) Pod를 연결: Label 과 Selector 를 활용
- Simple Load Balance (default : Round Robin)
- Type: ClusterIP(default), LoadBalancer, NodePort, ExternalName, ExternalIP
- kube-proxy 가 iptables 에 ClusterIP 세팅
- selector 로 Pod를 연결하면 Endpoint 가 자동으로 생성됨
- Default NodePort Range: 30000-32767



Service & Load balancing

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
  namespace: default
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 80
  type: ClusterIP
```

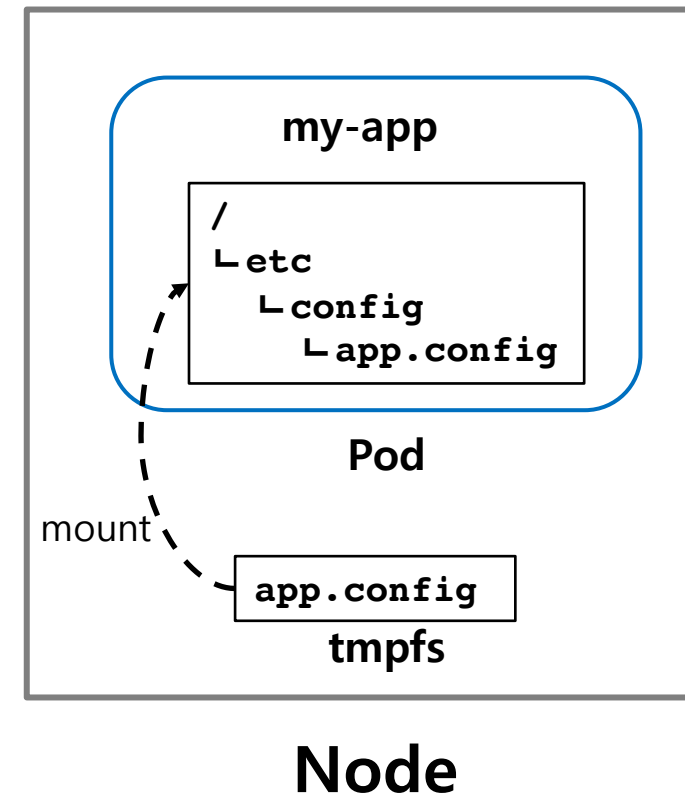


ConfigMap & Secret

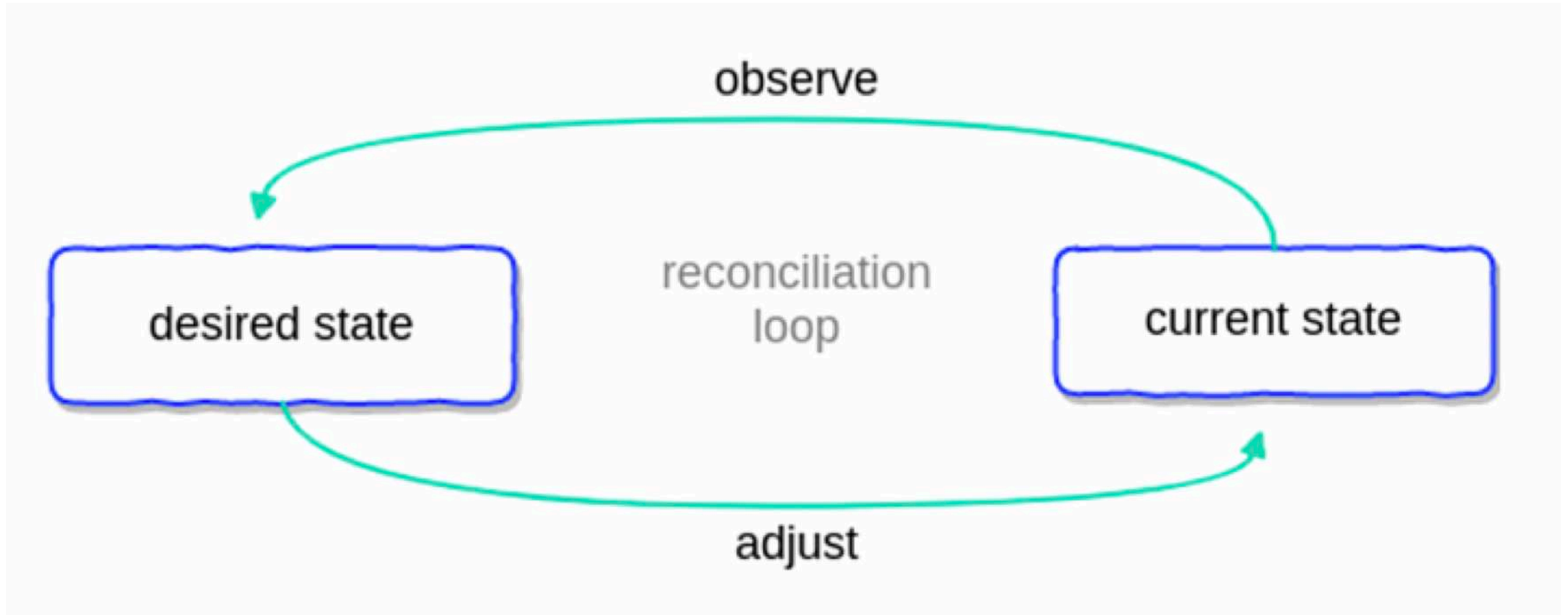
```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
  namespace: default
data:
  app.config : |
    url: nginx-service:8080
    log_level: debug
```

```
volumeMounts:
- name: config-volume
  mountPath: /etc/config
volumes:
- name: config-volume
  configMap:
    name: app-config
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: front
  template:
    metadata:
      labels:
        app: front
    spec:
      containers:
        - name: my-app
          image: nginx:1.21.0
          ports:
            - containerPort: 80
          volumeMounts:
            - ...
          volumes:
            - ...
```



Reconciliation



Cloud Native Computing Foundation (CNCF)

Graduated

[View on the CNCF landscape »](#)



containerd

Container Runtime



CoreDNS

Service Discovery



Envoy

Network Proxy



etcd

Key/Value Store



Fluentd

Logging



Harbor

Registry



Helm

Package Management



Jaeger

Distributed Tracing



Kubernetes

Orchestration



Open Policy Agent

Policy



Prometheus

Monitoring



Rook

Storage



TiKV

Key/Value Store



TUF

Software Update Spec



Vitess

Storage



Cloud Native Computing Foundation (CNCF)

Incubating

[View on the CNCF landscape »](#)

 Argo Continuous Integration & Deployment       	 Buildpacks Packaging Spec    	 CloudEvents Serverless      	 CNI Networking API      	 Contour High performance ingress controller  
 Cortex Monitoring     	 CRI-O Container Runtime      	 Dragonfly Image Distribution    	 Emissary-ingress	 Falco Container Security    
 Flux GitOps       	 gRPC Remote Procedure Call      	 KubeEdge Edge      	 Linkerd Service Mesh      	 NATS Messaging      
 Notary Security     	 OpenTracing Distributed Tracing API      	 Operator Framework Operator Lifecycle Manager (OLM) + Operator SDK     	 SPIFFE Identity Spec      	 SPIRE Identity    
 Thanos Monitoring      				

