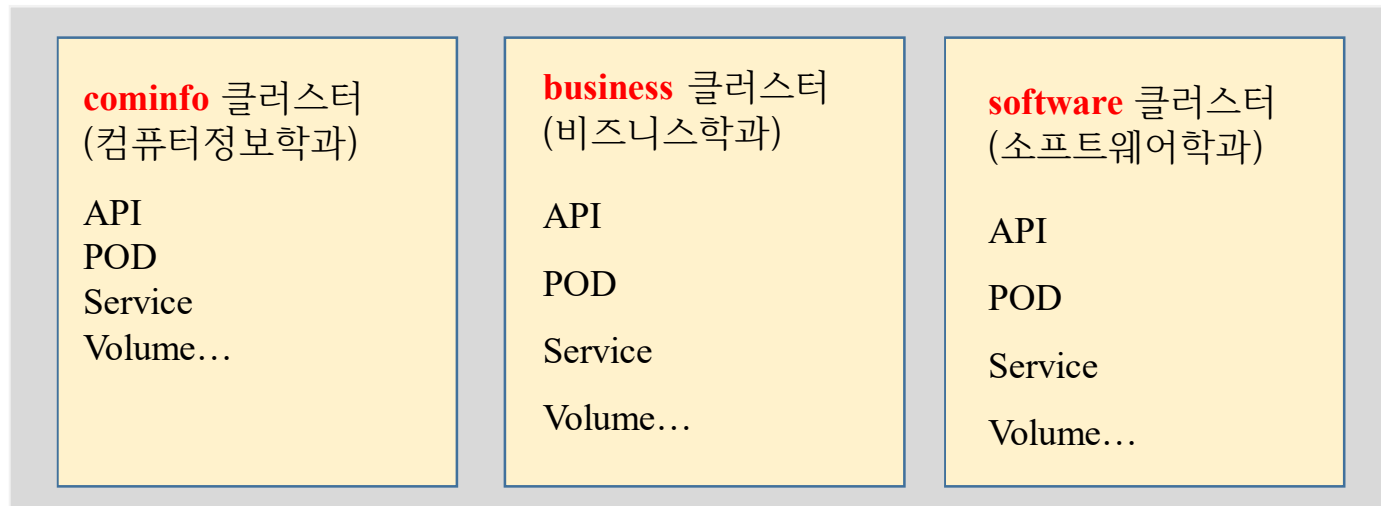


Namespace

Namespace

- K8S API 종류 중 하나
- K8S cluster 하나를 여러 개 논리적인 단위로 나눠 사용하는 것
- K8S Cluster 하나를 여러 개 팀이나 사용자가 함께 공유 할 수 있음



물리적 K8S 클러스터

현재 생성된 Namespace 확인

kubectl get namespaces

```
Terminal Host 1 +
controlplane $ kubectl get namespaces
NAME                STATUS    AGE
default             Active    42s
kube-node-lease     Active    43s
kube-public         Active    43s
kube-system         Active    43s
controlplane $
```

default Namespace

**kube-node-lease
Namespace**

**kube-public
Namespace**

**kube-system
Namespace**

Namespace 생성

kubectl create namespace **cominfo**

```
controlplane $ kubectl create namespace cominfo
namespace/cominfo created
controlplane $ kubectl get namespaces
NAME                STATUS    AGE
cominfo             Active    11s
default             Active    28m
kube-node-lease     Active    28m
kube-public         Active    28m
kube-system         Active    28m
controlplane $
```

default Namespace

**kube-node-lease
Namespace**

**kube-public
Namespace**

**kube-system
Namespace**

cominfo Namespace

지정된 Namespace에 pod 생성

kubectl run webserver --image=nginx:1.14 --port 80 --namespace **cominfo**

kubectl get pods --namespace cominfo

```
controlplane $ kubectl run webserver --image=nginx:1.14 --port 80 --namespace cominfo
pod/webserver created
controlplane $
controlplane $ kubectl get pods --namespace cominfo
NAME          READY   STATUS    RESTARTS   AGE
webserver     1/1     Running   0           30s
controlplane $
```

default Namespace

kube-node-lease
Namespace

kube-public
Namespace

kube-system
Namespace

cominfo Namespace



webserver

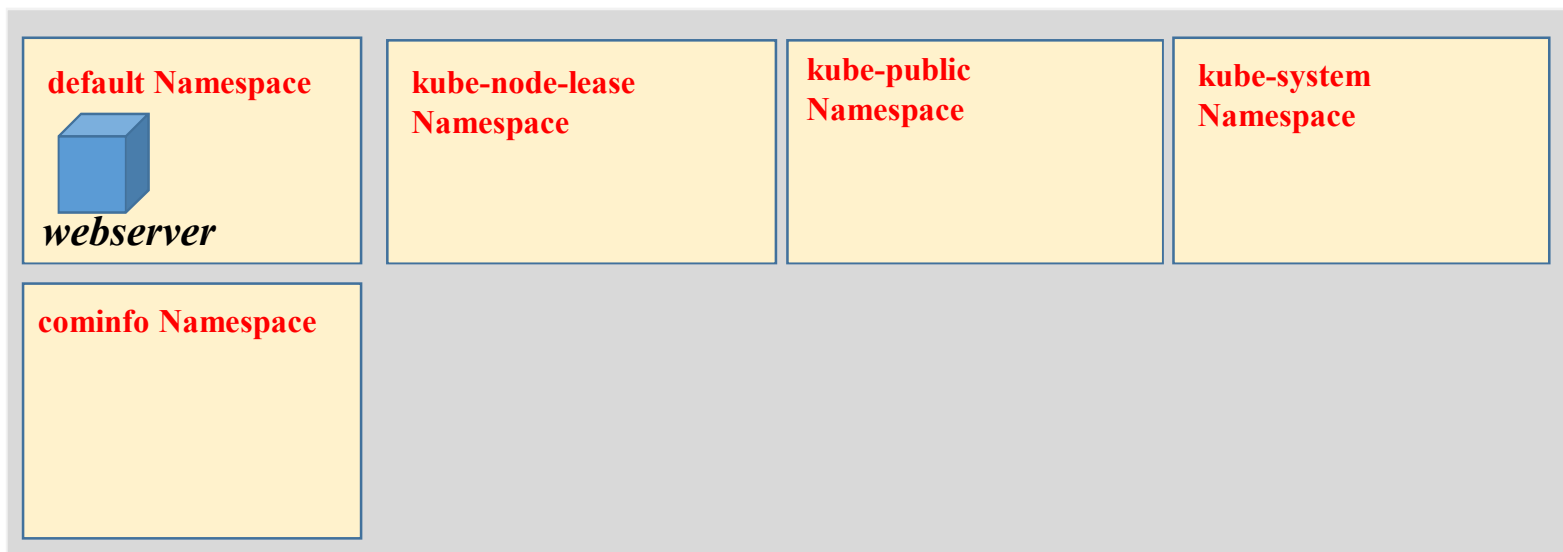
기본 namespace

kubectl run web --image=nginx:1.14 --port 80

```
controlplane $ kubectl run web --image=nginx:1.14 --port 80
pod/web created
controlplane $ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
web        1/1     Running   0           10s
```

kubectl get pod

```
controlplane $ kubectl get pod
No resources found in default namespace.
controlplane $
```

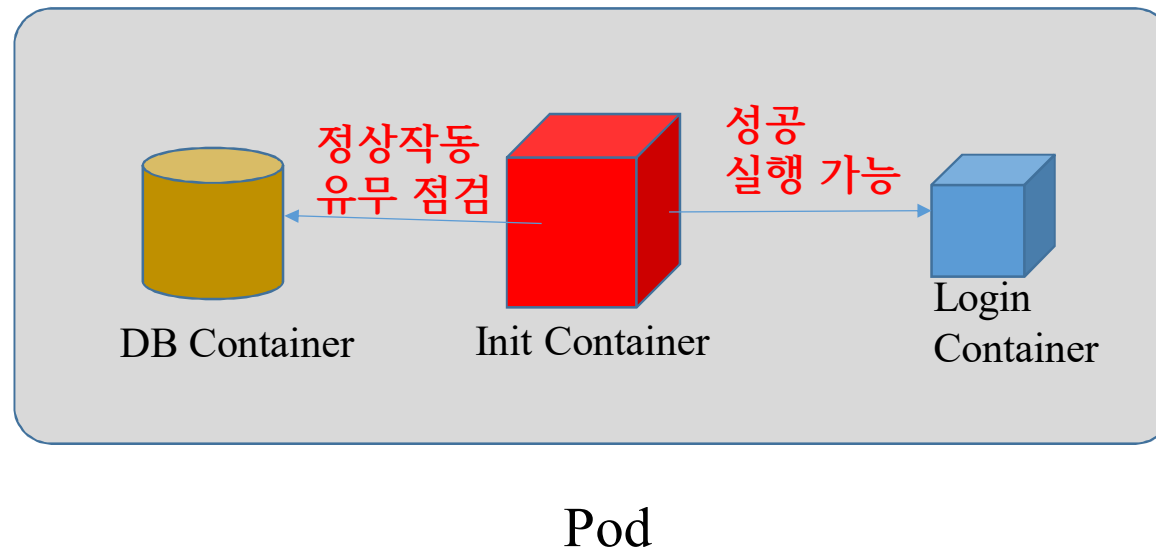


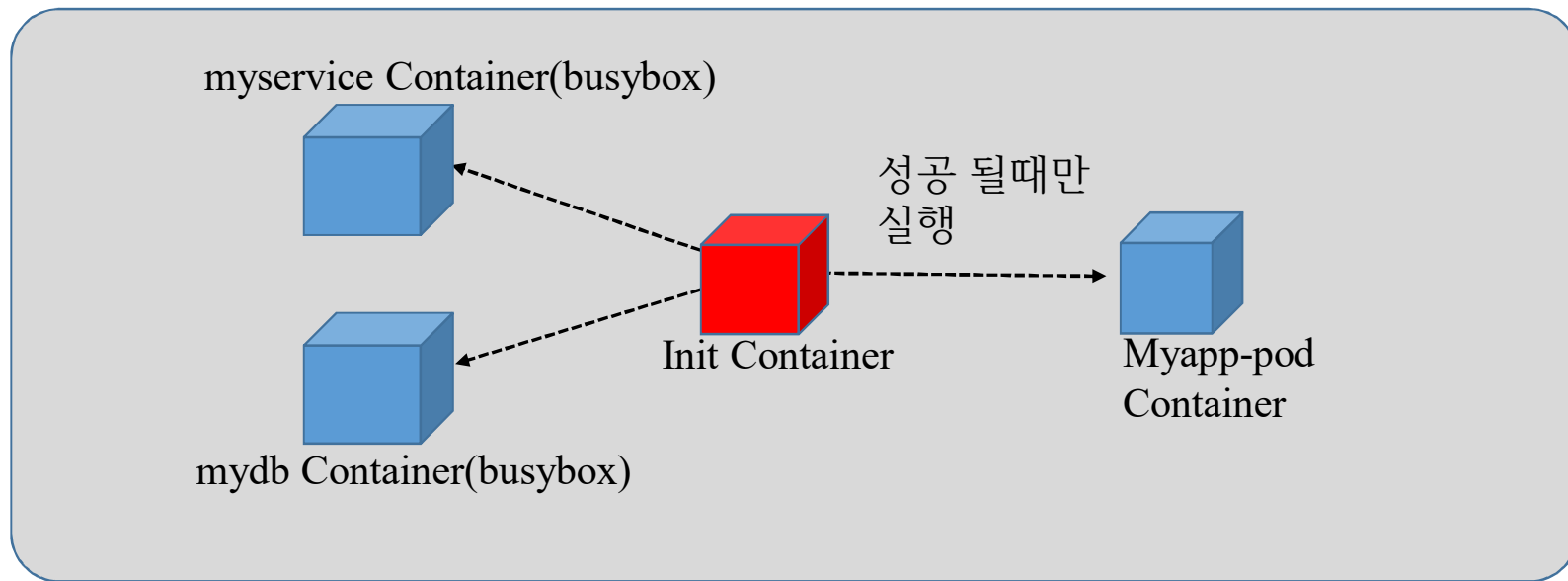
Init Container

초기화 컨테이너(init container)

- Main container(App container)가 실행되기 전 동작시킨 컨테이너
- Main Container가 실행되기 전에 사전 작업이 필요한 경우 사용
- 초기화 컨테이너가 모두 실행된 후에 App container를 실행
 - Init container는 여러 개 구성할 수 있음
 - Init container 실행이 실패하면 성공할 때까지 재시작함
 - Init container가 모두 실행된 후 App Container 실행이 시작

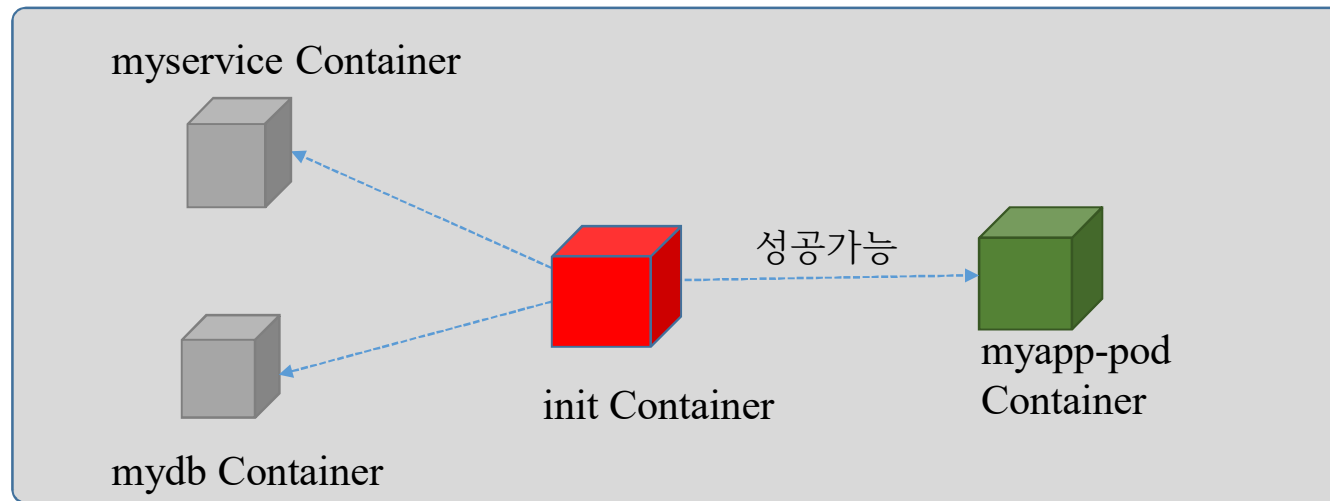
초기화 컨테이너(init container)





Alt+Enter

```
[node1 ~]$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE   NOMINATED NODE   READINESS GATES
myapp-pod     0/1     Init:0/2   0          48s   10.5.1.5    node2   <none>           <none>
[node1 ~]$
```



```

apiVersion: v1
kind: Service
metadata:
  name: myservice
spec:
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376

```

myservice.yaml

```

apiVersion: v1
kind: Service
metadata:
  name: mydb
spec:
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9377

```

mydb.yaml


init Container

```

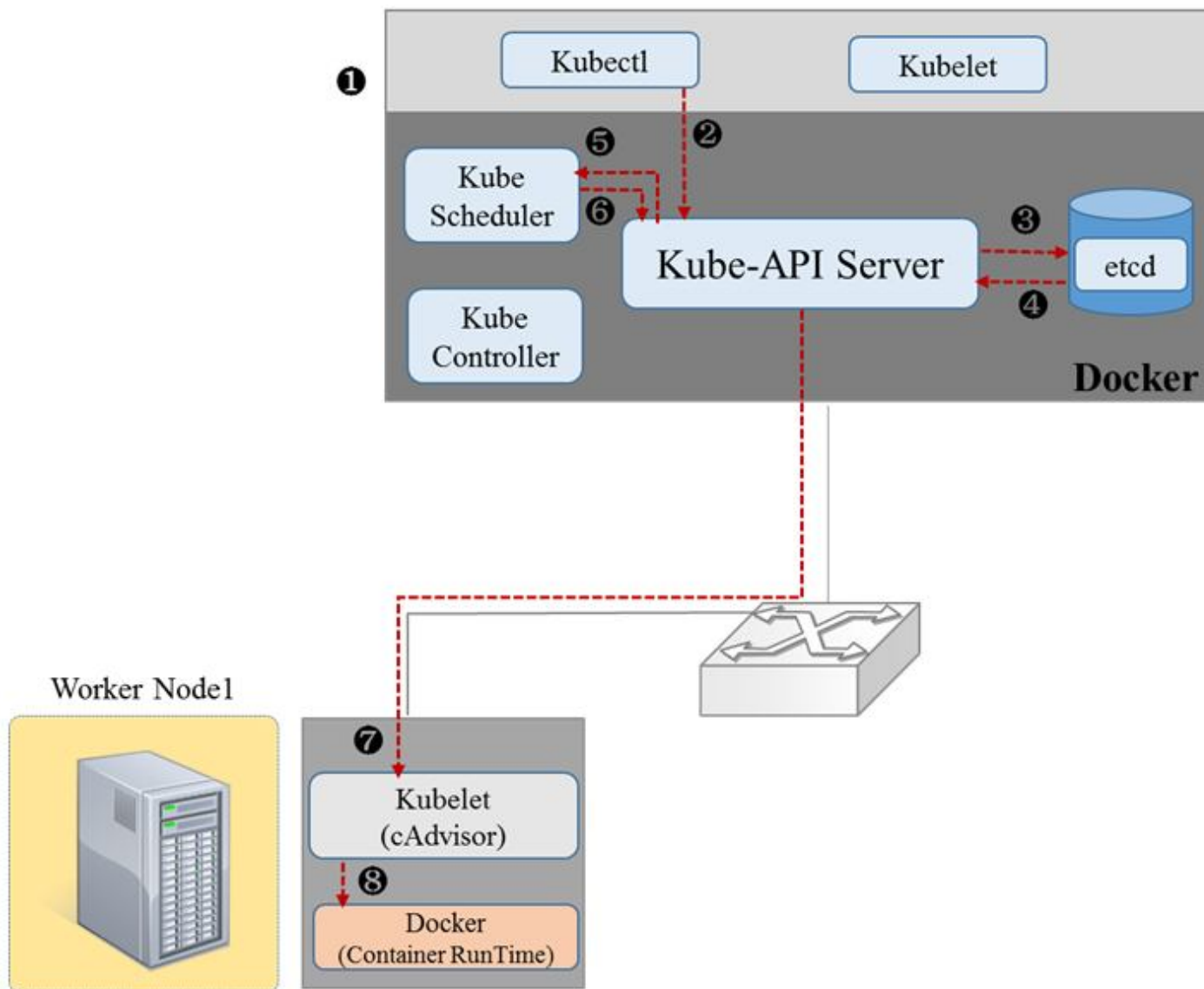
apiVersion: v1
kind: Pod
metadata:
  name: myapp-pod
spec:
  containers:
    - name: myapp-container
      중간 생략~~~
    initContainers:
    - name: myservice
      중간 생략~~~
    - name: mydb
      중간 생략~~~

```

myapp-pod.yaml

- `kubectl delete pod --all`
- `kubectl get pods -o wide --watch`
- `kubectl create -f init-container-exam.yaml`
- `kubectl create -f init-container-exam-svc.yaml`
- `kubectl create -f init-container-exam-db.yaml`
- `kubectl get pods -o wide`
- `kubectl describe pods pod `

Static Pod




Static Pod

- Kube-apiserver를 통하지 않고 kubelet이 직접 실행하는 pod
 - API 서버 없이 특정 노드에 있는 kubelet 데몬에 의해 직접관리
 - kubelet daemon에 의해 관리되는 container
- Kubelet이 직접관리하면서 이상이 생기면 재시작

실습. Static Pod

```
[node1 ~]$ cd /var/lib/kubelet  
[node1 kubelet]$ ls  
config.yaml  cpu_manager_state  device-plugins  kubeadm-flags.env  
[node1 kubelet]$ cat config.yaml  
apiVersion: kubelet.config.k8s.io/v1beta1  
authentication:
```



```
kind: KubeletConfiguration  
logging: {}  
nodeStatusReportFrequency: 0s  
nodeStatusUpdateFrequency: 0s  
rotateCertificates: true  
runtimeRequestTimeout: 0s  
shutdownGracePeriod: 0s  
shutdownGracePeriodCriticalPods: 0s  
staticPodPath: /etc/kubernetes/manifests  
streamingConnectionIdleTimeout: 0s  
syncFrequency: 0s  
volumeStatsAggPeriod: 0s  
[node1 kubelet]$
```


Pod lifecycle

- Pod 생성부터 삭제까지의 과정에서 생명주기(lifecycle)가 있음

kubectl get pods -o wide --watch

```
[node1 ~]$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
multiple	2/2	Running	0	19s	10.5.1.3	node2	<none>		<none>	
single	1/1	Running	0	11m	10.5.2.2	node3	<none>		<none>	

\$kubectl describe pods *nginx-pod-live*

```
[node1 ~]$ kubectl describe pods nginx-pod-live
Name:          nginx-pod-live
Namespace:     default
Priority:       0
Node:          node3/192.168.0.11
Start Time:    Sun, 01 May 2022 04:19:01 +0000
Labels:        <none>
Annotations:   <none>
Status:        Running
IP:            10.5.2.3
IPs:
  IP: 10.5.2.3
Containers:
```

Pod 생명주기(lifecycle)

Pending	K8S 시스템에 pod를 생성하는 중임을 뜻함 Container 이미지를 다운로드한 후 전체 container를 실행하는 도중 Pod안에 전체 container가 실행될때까지 시간이 걸림
Running	Pod 안에 모든 container가 실행 중인 상태 1개 이상의 container가 실행 중이거나 시작 또는 재시작 상태 임
Succeeded	Pod 안 모든 container가 정상 실행 종료된 상태로 재시작 되지 않았음
Failed	Pod 안 모든 container가 정상적으로 실행 종료되지 않은 container가 있는 상태 Container 종료코드가 0이 아니면 비정상종료 또는시스템이 직접 Container를 종료한것
Unknown	Pod의 상태를 알 수 없는 상태 (Pod가 있는 노드와 통신 할 수 없을 때)

```
[node1 kubelet]$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
myapp-pod     0/1     Init:0/2   0           11s

[node1 kubelet]$ kubectl describe pods myapp-pod
Name:          myapp-pod
Namespace:     default
Priority:       0
Node:          node2/192.168.0.12
Start Time:    Tue, 03 May 2022 04:01:37
Labels:        app=myapp
Annotations:    <none>
Status:        Pending
IP:            10.5.1.2
IPs:
  IP: 10.5.1.2
Init Containers:
  Conditions:
    Type              Status
    Initialized        False
    Ready              False
    ContainersReady    False
    PodScheduled       True
  Volumes:
    default-token-hg45s:
      Type:          Secret (a volume populated by a Secret)
      SecretName:    default-token-hg45s
      Optional:      false
    QoS Class:        BestEffort
```

Condition kubectl describe pods pod명

Initialized	모든 초기화 컨테이너가 성공적으로 시작 완료
Ready	Pod는 요청을 실행 할 수 있음 연결된 모든 서비스의 로드밸런싱 Pool에 추가되어야 한다는 뜻
ContainersReady	Pod 안 모든 컨테이너가 준비상태
PodScheduled	Pod가 하나의 노드로 스케줄을 완료 했음
UnSchedulable	스케줄러가 자원의 부족이나 다른 제약 등으로 지금 당장 Pod를 스케줄 할 수 없음