

## Pod 생명주기(lifecycle)

- Pod 생성부터 삭제까지의 과정에서 생명주기(lifecycle)가 있음

```
[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	<ul style="list-style-type: none"><li>• K8S 시스템에 pod를 생성하는 중임을 뜻함</li><li>• Container 이미지를 다운로드한 후 전체 container를 실행하는 도중</li><li>• Pod안에 전체 container가 실행 될 때까지 시간이 걸림</li></ul>
Running	<ul style="list-style-type: none"><li>• Pod 안에 모든 container가 실행 중인 상태</li><li>• 1개 이상의 container가 실행 중이거나 시작 또는 재시작 상태 임</li></ul>
Succeeded	<ul style="list-style-type: none"><li>• Pod 안 모든 container가 정상 실행 종료된 상태로 재시작 되지 않았음</li></ul>
Failed	<ul style="list-style-type: none"><li>• Pod 안 모든 container가 정상적으로 실행 종료되지 않은 container가 있는 상태</li><li>• Container 종료코드가 0이 아니면 비정상종료 또는 시스템이 직접 Container를 종료한 것</li></ul>
Unknown	<ul style="list-style-type: none"><li>• Pod의 상태를 알 수 없는 상태 (Pod가 있는 노드와 통신 할 수 없을 때 )</li></ul>

# Pod Conditions

- Pod의 현재 상태 정보를 나타냄 (Type과 Status로 구분)

```
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
```

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

```

[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 +0000
Labels:        app=myapp
Annotations:    <none>
Status:        Pending
IP:            10.5.1.2
IPs:
  IP: 10.5.1.2
Init Containers:

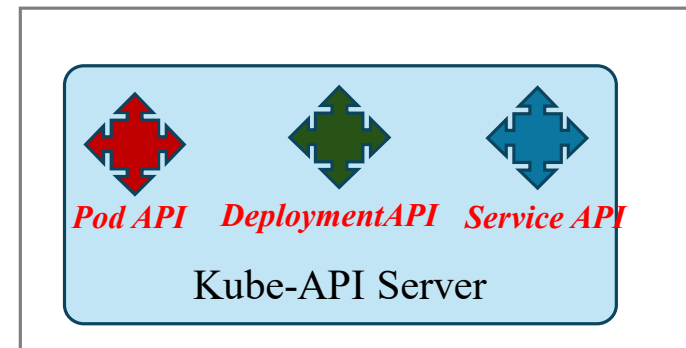
```

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

# API Version

- K8S Object 정의 시 API 버전 필요
- Update 된 API가 있다면 새로운 API로 생성
  - Alpha → Beta → Stable

Object API	Version
Pod	v1
Deployment	Apps/v1
ReplicaSet	Apps/v1
service	v1



Master(Control plane)

kubectl explain pod

kubectl explain deployment

```
controlplane $ kubectl explain pod
KIND:      Pod
VERSION:   v1
```

```
controlplane $ kubectl explain deployment
KIND:      Deployment
VERSION:   apps/v1
```

## kubectl api-resources

```
root@masternode:~# kubectl api-resources
NAME                SHORTNAMES  APIVERSION  NAMESPACED  KIND
bindings            v1          true        Binding
componentstatuses   cs          v1          false        ComponentStatus
configmaps           cm          v1          true         ConfigMap
endpoints            ep          v1          true         Endpoints
events              ev          v1          true         Event
limitranges          limits      v1          true         LimitRange
namespaces           ns          v1          false        Namespace
nodes               no          v1          false        Node
```

## kubectl api-resources | grep deploy

# 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**

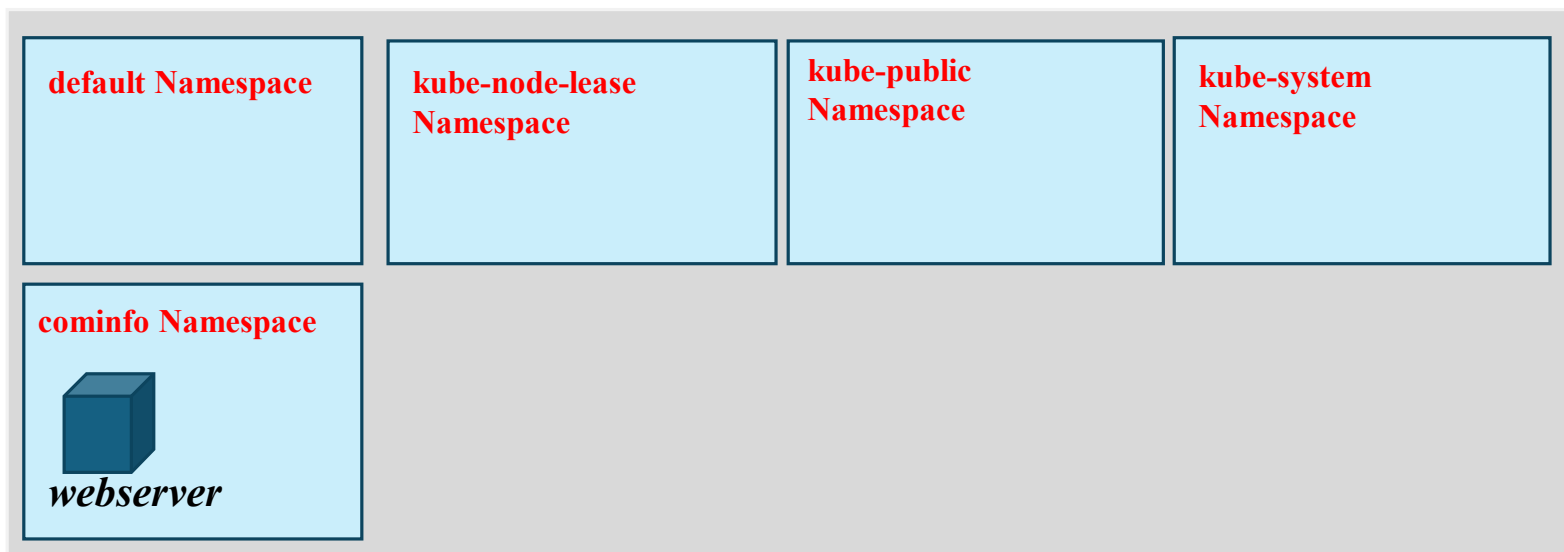
**cominfo Namespace**

## 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 $
```





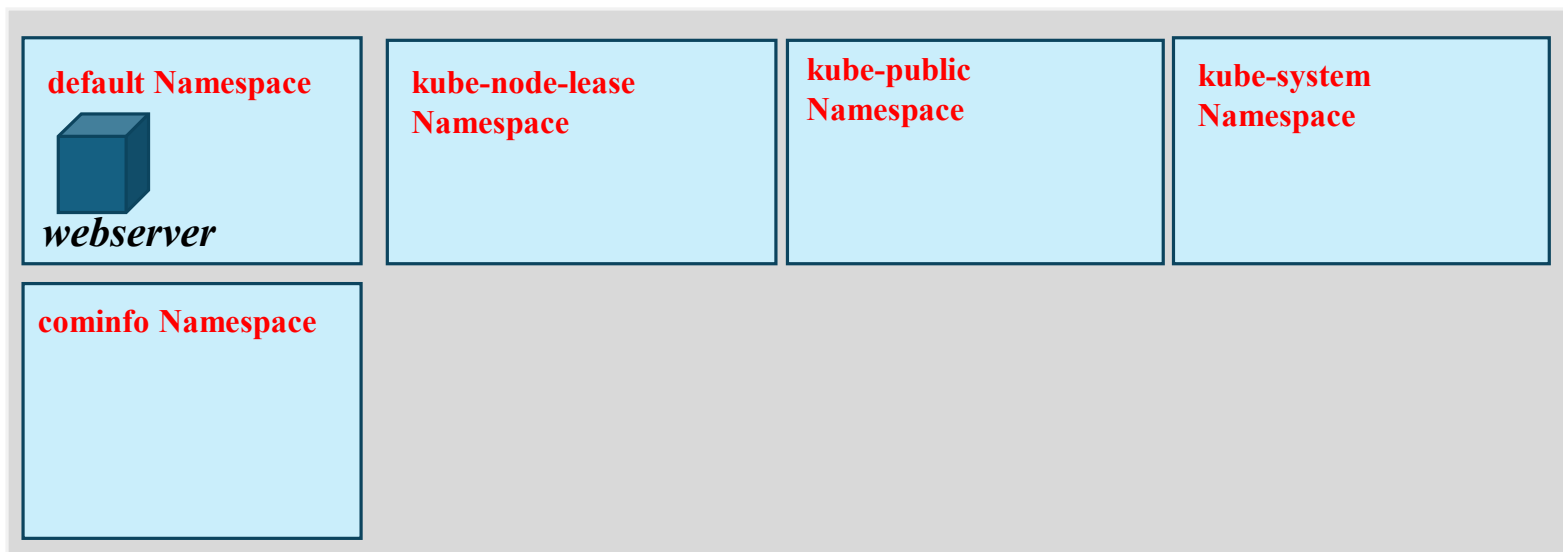
# namespace

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

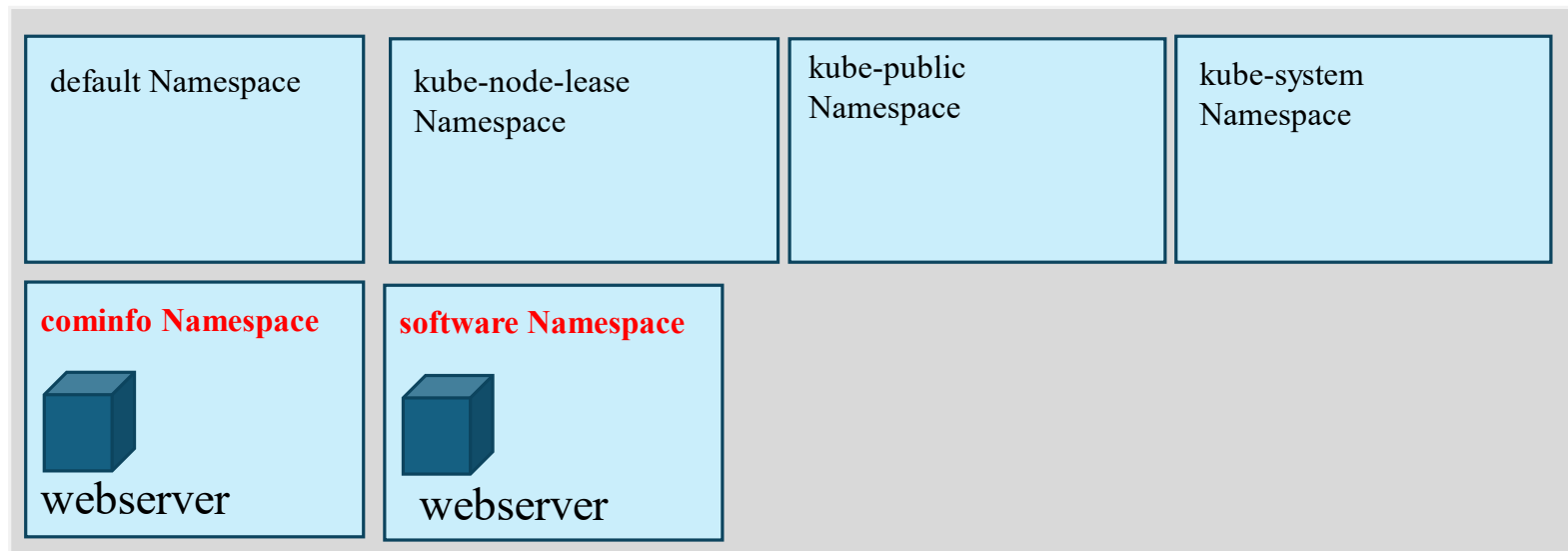
```
controlplane $ kubectl run web --image=nginx:1.14 --port 80
pod/web created
controlplane $ kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
web       1/1     Running   0           10s
```

kubectl get pod

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

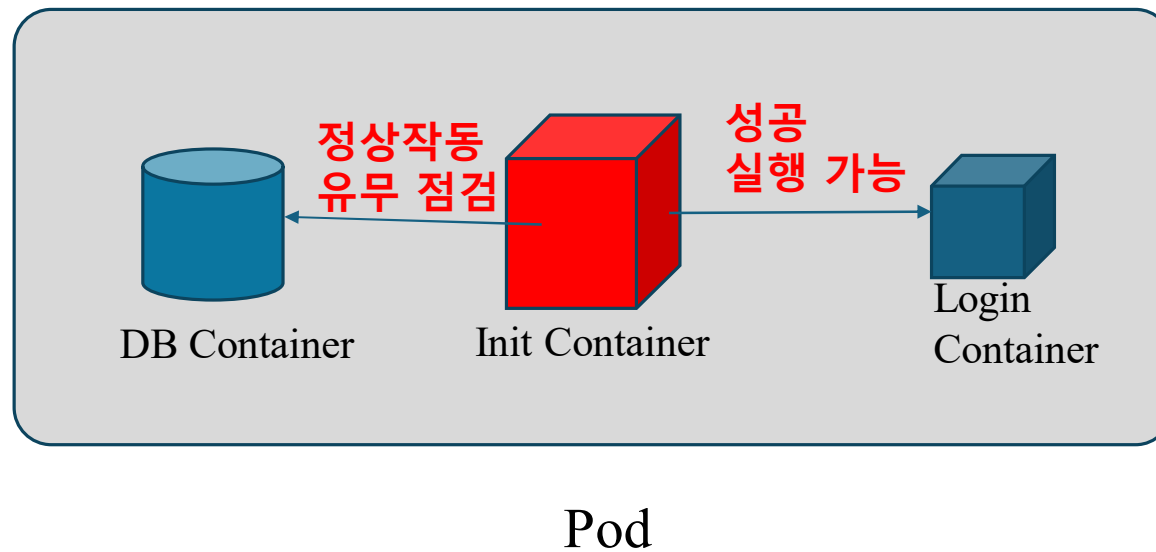


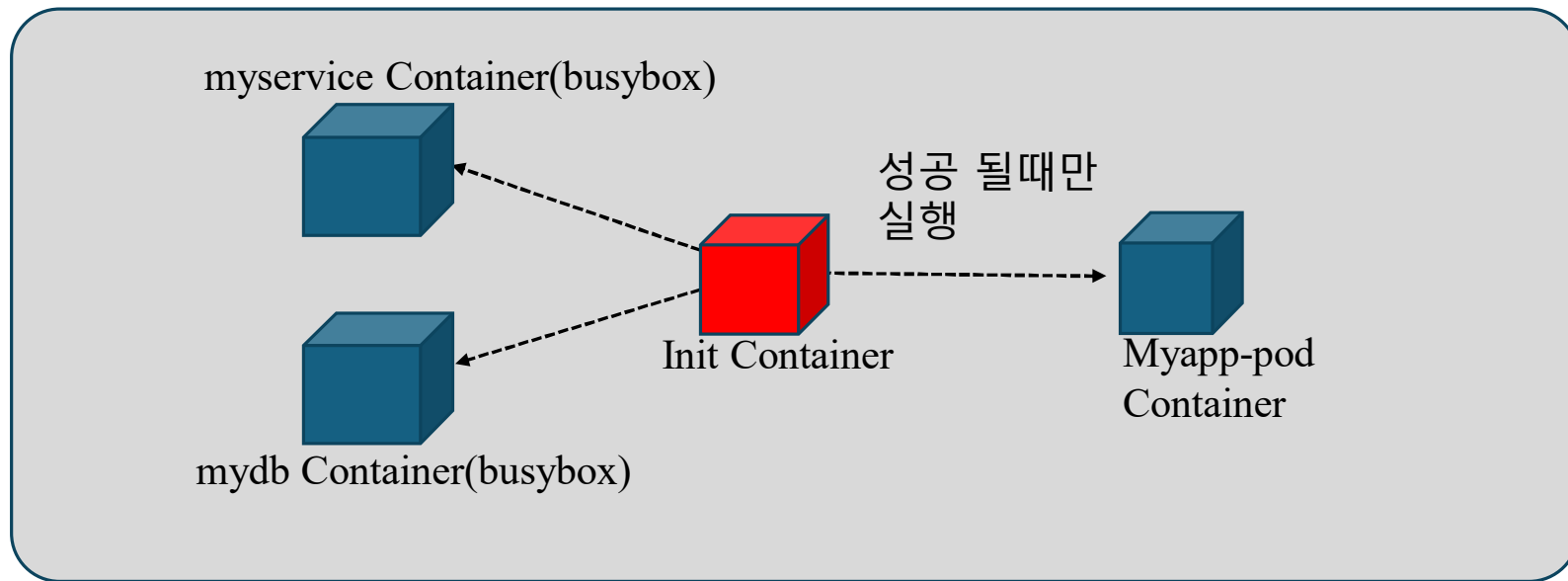
## 실습 1. Namespace 생성 후 지정된 cluster에 nginx server 생성 배치



- software Namespace 생성
- 생성된 software namespace에 nginx:1.14 Webserver pod 배치

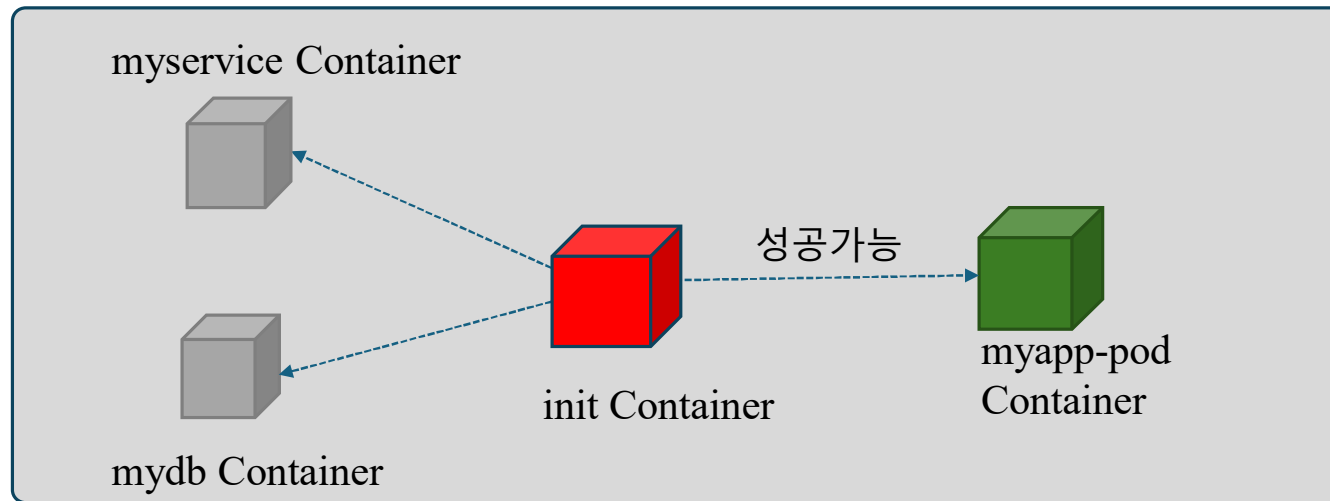
## 초기화 컨테이너(init container)





**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 --namespace cominfo`
- `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 `