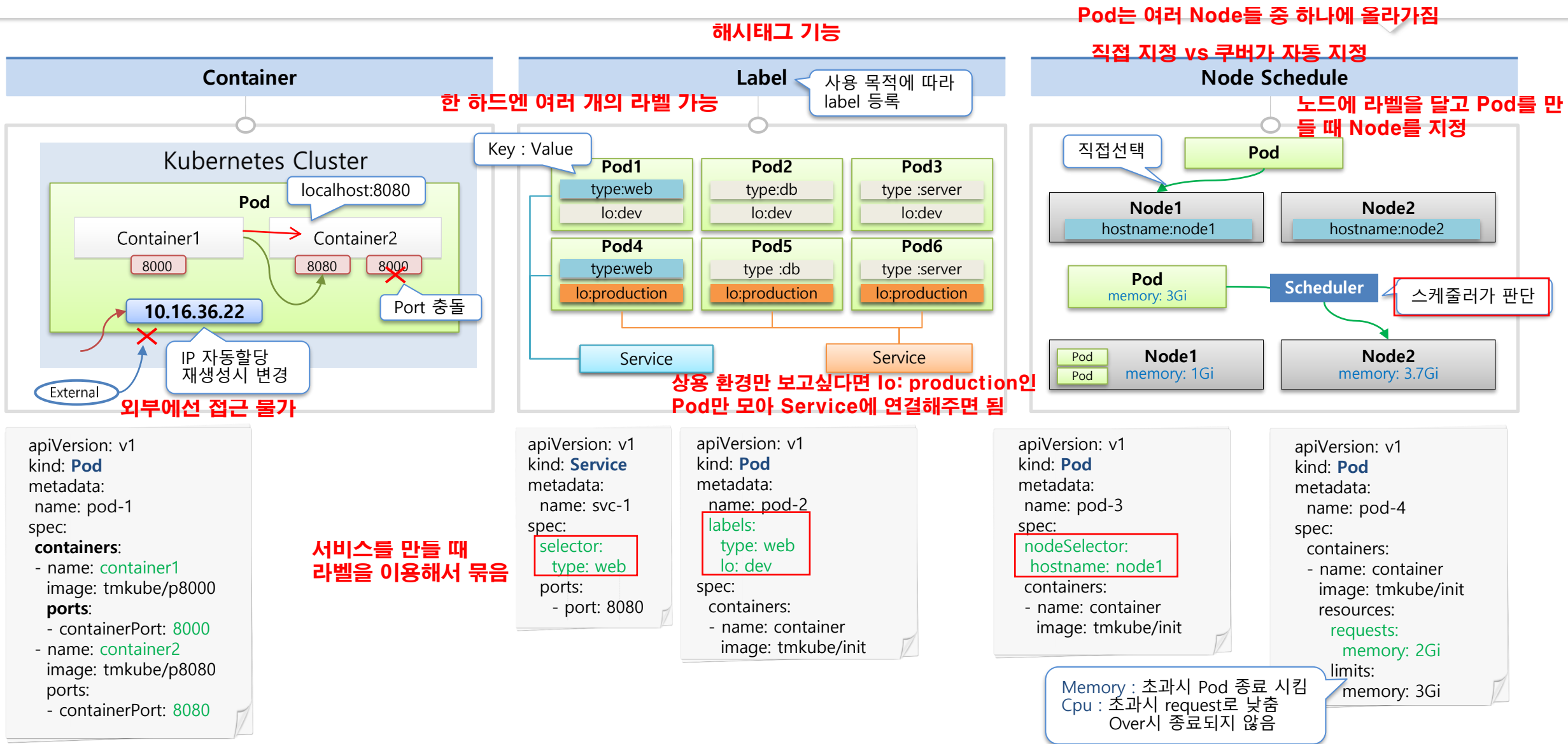


5. Object - Pod



5. Object Service

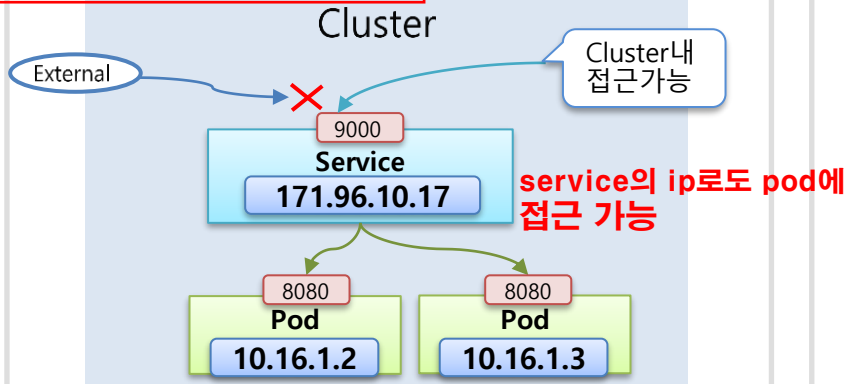
- 서비스 디버깅
- 내부 트래픽/대쉬보드
- 인증된 사용자 연결

- 데모나 임시 연결용
- 프로덕션 환경(X)

- 외부 시스템 노출용

ClusterIP

클러스터 내에서만 접근 가능



```
apiVersion: v1
kind: Service
metadata:
  name: svc-1
spec:
  selector:
    app: pod
  ports:
    - port: 9000
      targetPort: 8080
  type: ClusterIP
```

기본값: clusterip임

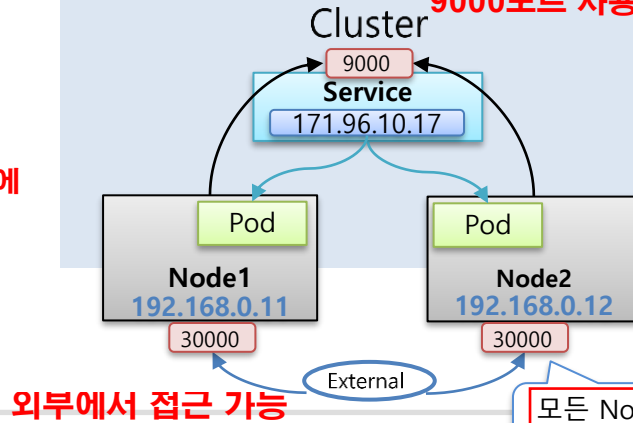
```
apiVersion: v1
kind: Pod
metadata:
  name: pod-1
labels:
  app: pod
spec:
  containers:
    - name: container
      image: tmkube/app
      ports:
        - containerPort: 8080
```

service를 통해 접근하는 이유

- pod는 죽으면 언제든지 재 생성 된다. 그때 ip가 바뀌기 때문

NodePort

클러스터 내부에서 service에 접근할 때 9000포트 사용



```
apiVersion: v1
kind: Service
metadata:
  name: svc-2
spec:
  selector:
    app: pod
  ports:
    - port: 9000
      targetPort: 8080
      nodePort: 30000
  type: NodePort
```

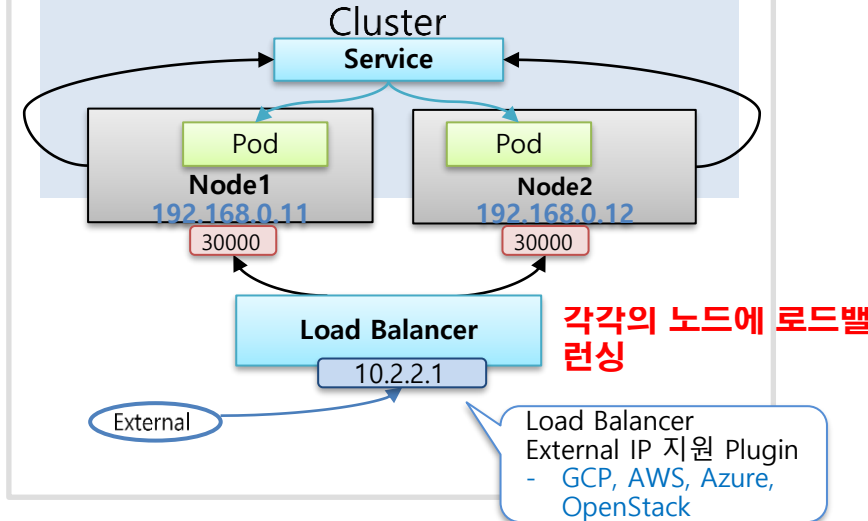
externalTrafficPolicy: Local

이게 없으면 Node1에 접근해도 Node1, 2 다 갈 수 있음 => 로드밸런싱 있으면 접근한 Node만 접근 가능

모든 Node에 Port할당

30000~32767

Load Balancer



```
apiVersion: v1
kind: Service
metadata:
  name: svc-3
spec:
  selector:
    app: pod
  ports:
    - port: 9000
      targetPort: 8080
  type: LoadBalancer
```

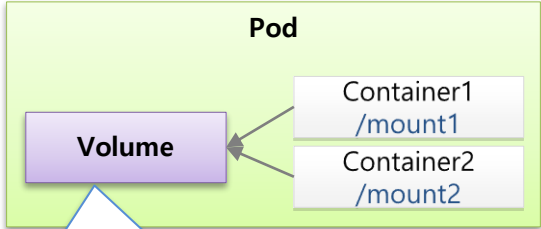
5. Object - Volume

최초 Admin이 PV를 만들고 사용자가 PVC를 만들면
쿠버네티스가 PVC내용에 맞는 적절한 PV에 연결해줌
그 다음 Pod를 만들 때 PVC를 사용

일시적인 사용 목적에 의한 데이터만 넣기

emptyDir

컨테이너 1,2 가 서로 파일을 주고 받을
필요 없이 mount된 걸 쓰면 됨



Pod 생성시 만들어지고
삭제시 없어짐

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-volume-1
spec:
```

containers:

- name: container1
- image: tmkub/init

volumeMounts:

- name: empty-dir
- mountPath: /mount1
- name: container2
- image: tmkub/init
- volumeMounts:
- name: empty-dir
- mountPath: /mount2

volumes:

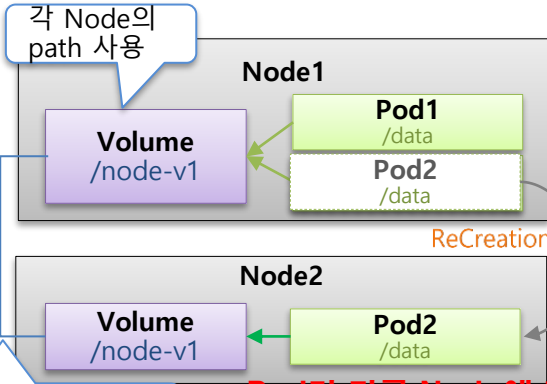
- name: empty-dir
- emptyDir: {}

mountPath는 다르지
만 name이 같아서
공유

이게 같아야함

hostPath

Pod가 죽어도 괜찮



Node추가시마다
Mount걸어줌

Pod가 다른 Node에 재생성
시켜준다면 볼륨 연결 실패

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-volume-2
spec:
```

containers:

- name: container
- image: tmkub/init
- volumeMounts:
- name: host-path
- mountPath: /mount1

volumes:

- name: host-path
- hostPath:
- path: /node-v
- type: Directory

DirectoryOrCreate로 하면
만들어짐

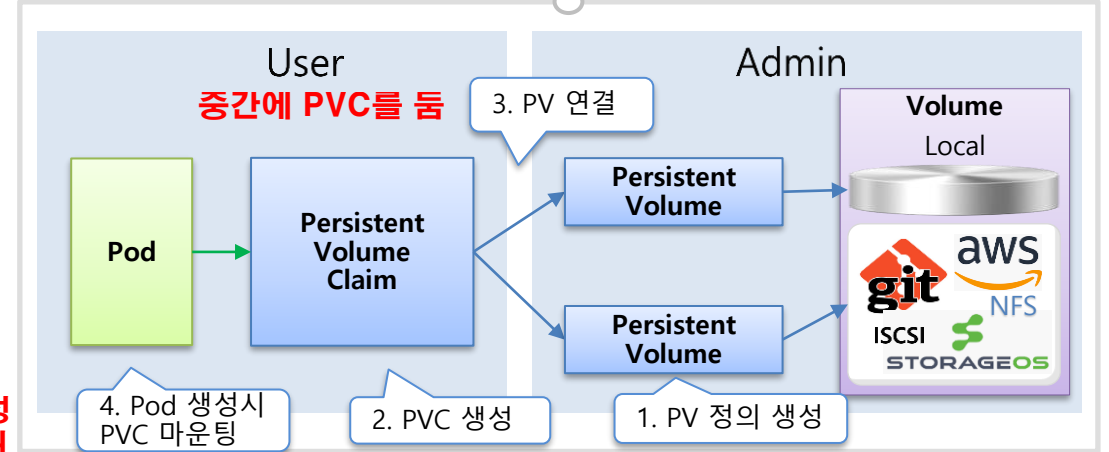
사전에 해당 Node에
경로가 있어야함

Pod가 만들어지기 전에 생성되어있어야 함

PVC / PV

서비스 담당자가 관리

쿠버 운영자가 관리



```
apiVersion: v1
kind: Pod
metadata:
  name: pod-volume-3
spec:
```

containers:

- name: container
- image: tmkub/init
- volumeMounts:
- name: pvc-pv
- mountPath: /volume

volumes:

- name: pvc-pv
- persistentVolumeClaim:
- claimName: pvc-01

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pvc-01
spec:
```

accessModes:

- ReadWriteOnce

resources:

- requests:
- storage: 1G

storageClassName: ""

읽기 쓰기
용량 1기가
현재 만들어진 PV들 중
선택이 됨

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-01
spec:
```

capacity:

- storage: 1G
- accessModes:
- ReadWriteOnce

local:

- path: /node-v
- nodeAffinity:
- required:
- nodeSelectorTerms:
- matchExpressions:
- {key: node, operator:
- In, values: [node1]}

Pod가 해당
Node에 만들어짐

nfs:

serve

path:

iscsi:

target:

iqn: i

lun: 0

fsType:

readOnly:

chap:

gitRepo:

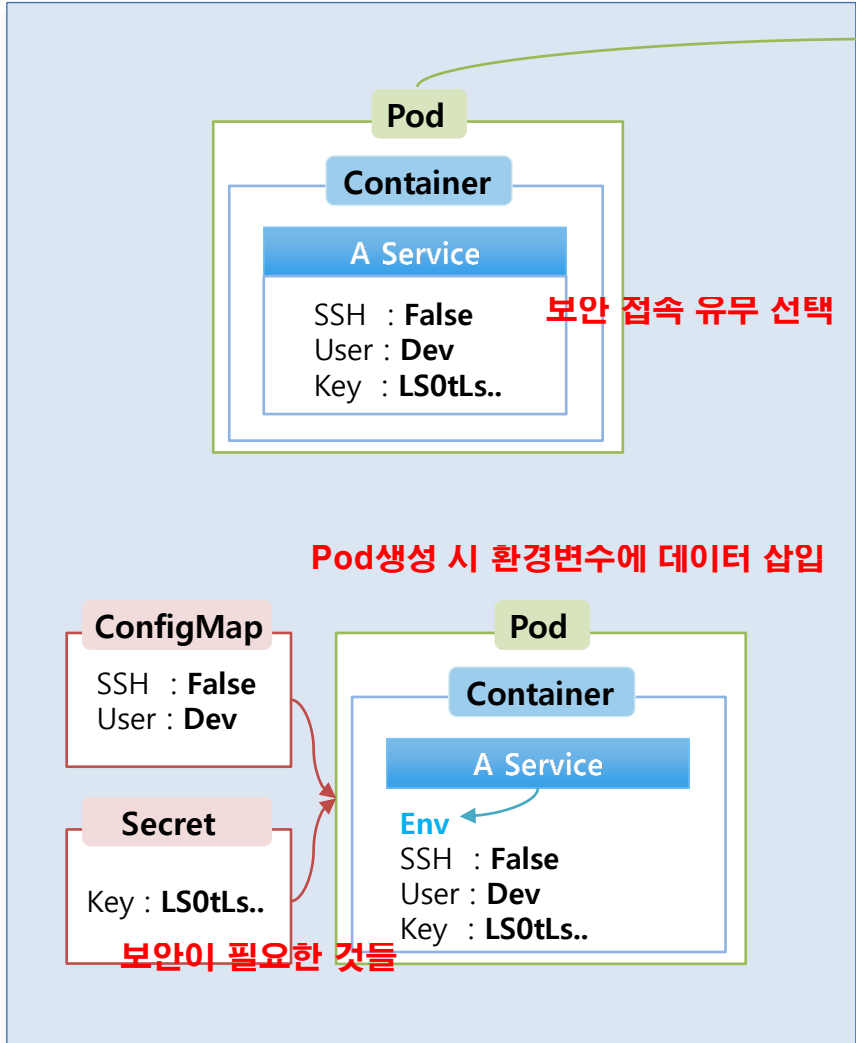
repo:

revision:

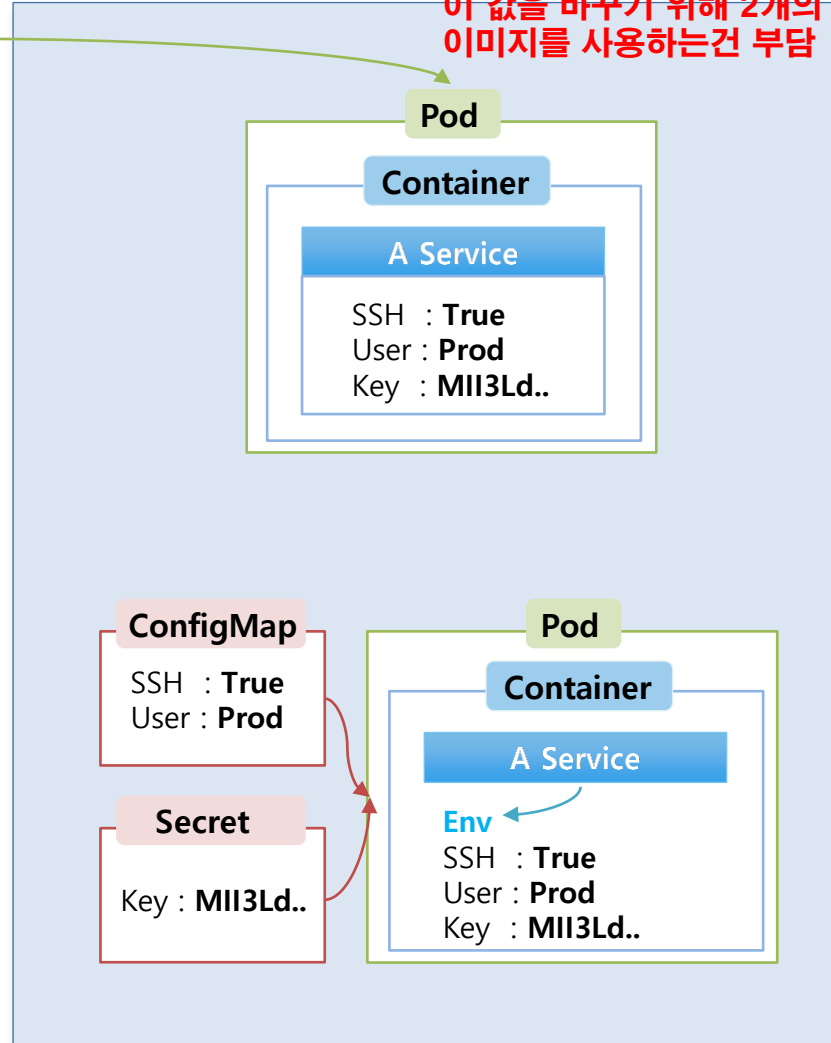
direct:

5. Object - ConfigMap, Secret 환경에 따라 변하는 값들을 외부에서 설정

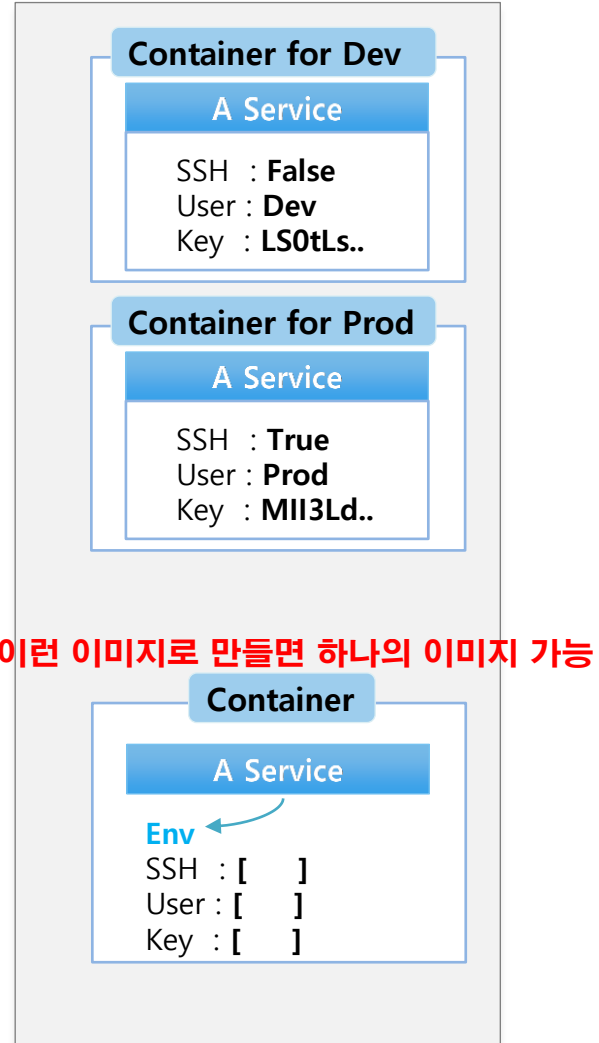
Dev



Production

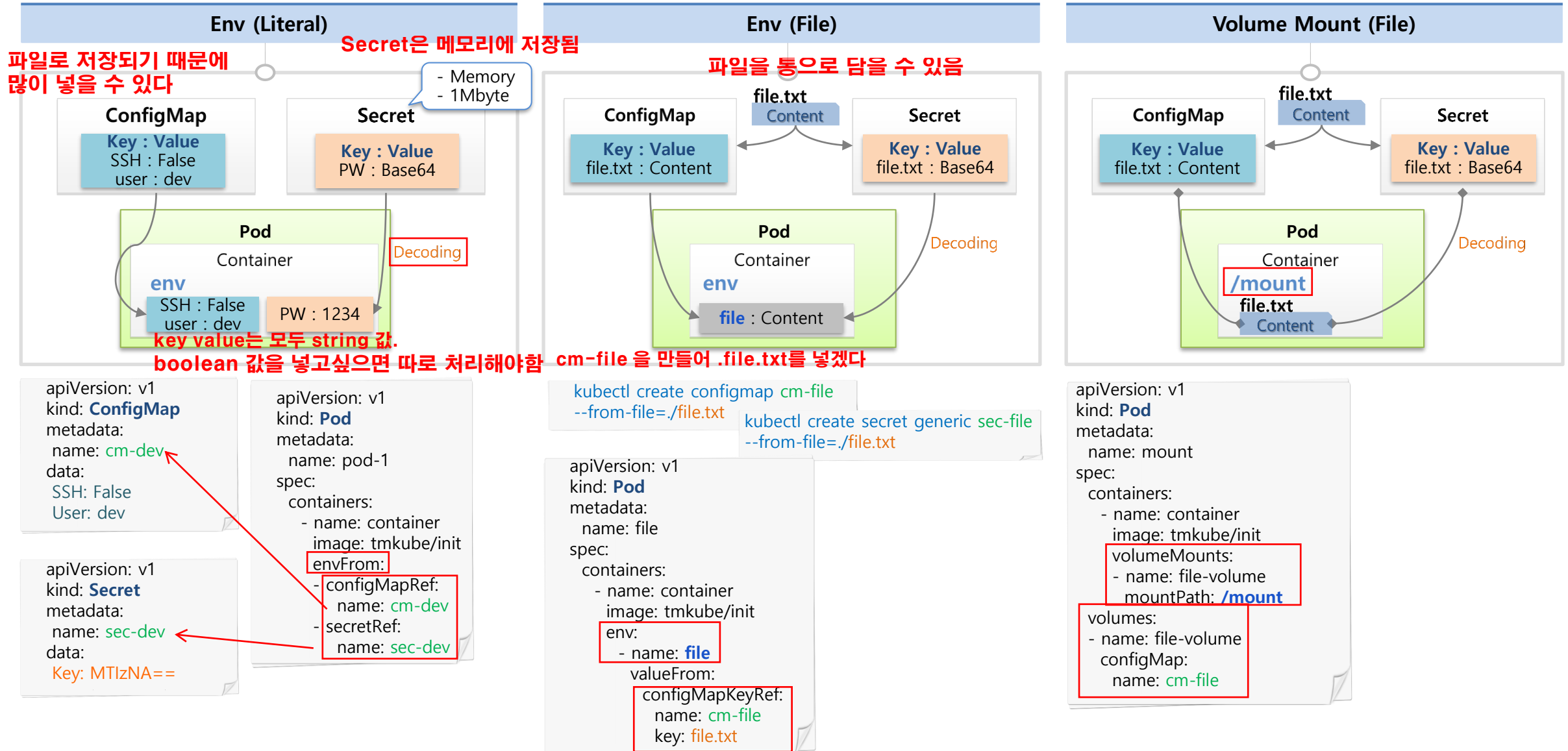


Image



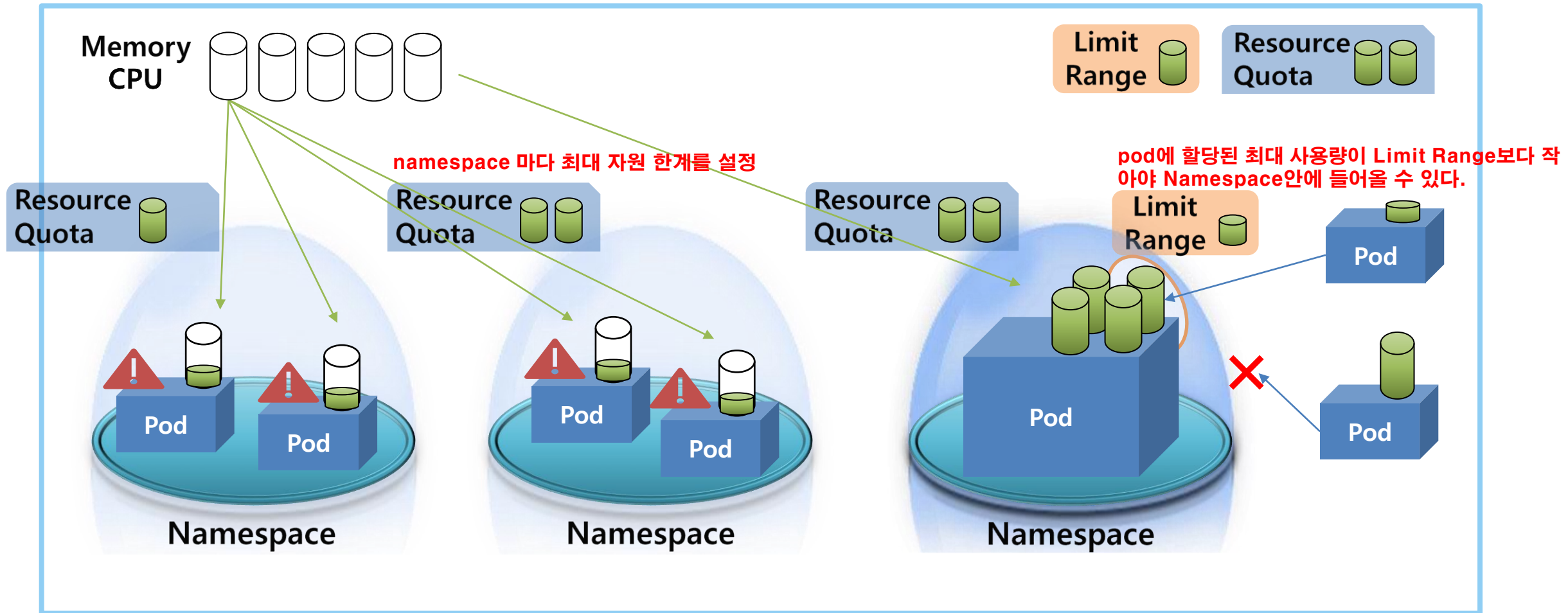
5. Object - ConfigMap, Secret

추후에 ConfigMap에 데이터가 변경된다면
Env는 Pod에 반영되지 않는 반면 Mount는 반영된다.



5. Object - Namespace, ResourceQuota, LimitRange

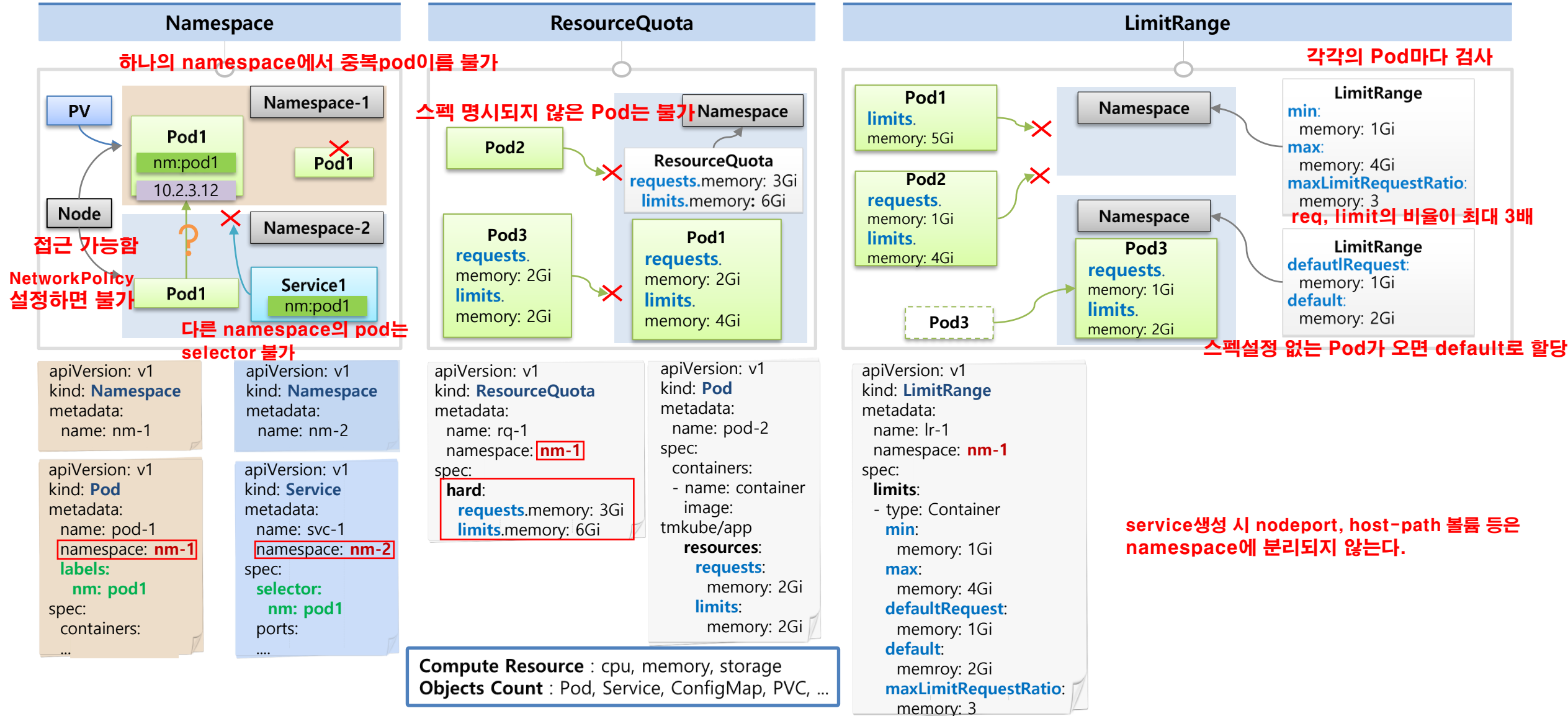
Kubernetes Cluster



5. Object - Namespace, ResourceQuota, LimitRange

다른 namespace와의 자원 분리 가능

하나의 namespace엔 하나의 limitrange를 지정하자



Compute Resource : cpu, memory, storage
Objects Count : Pod, Service, ConfigMap, PVC, ...