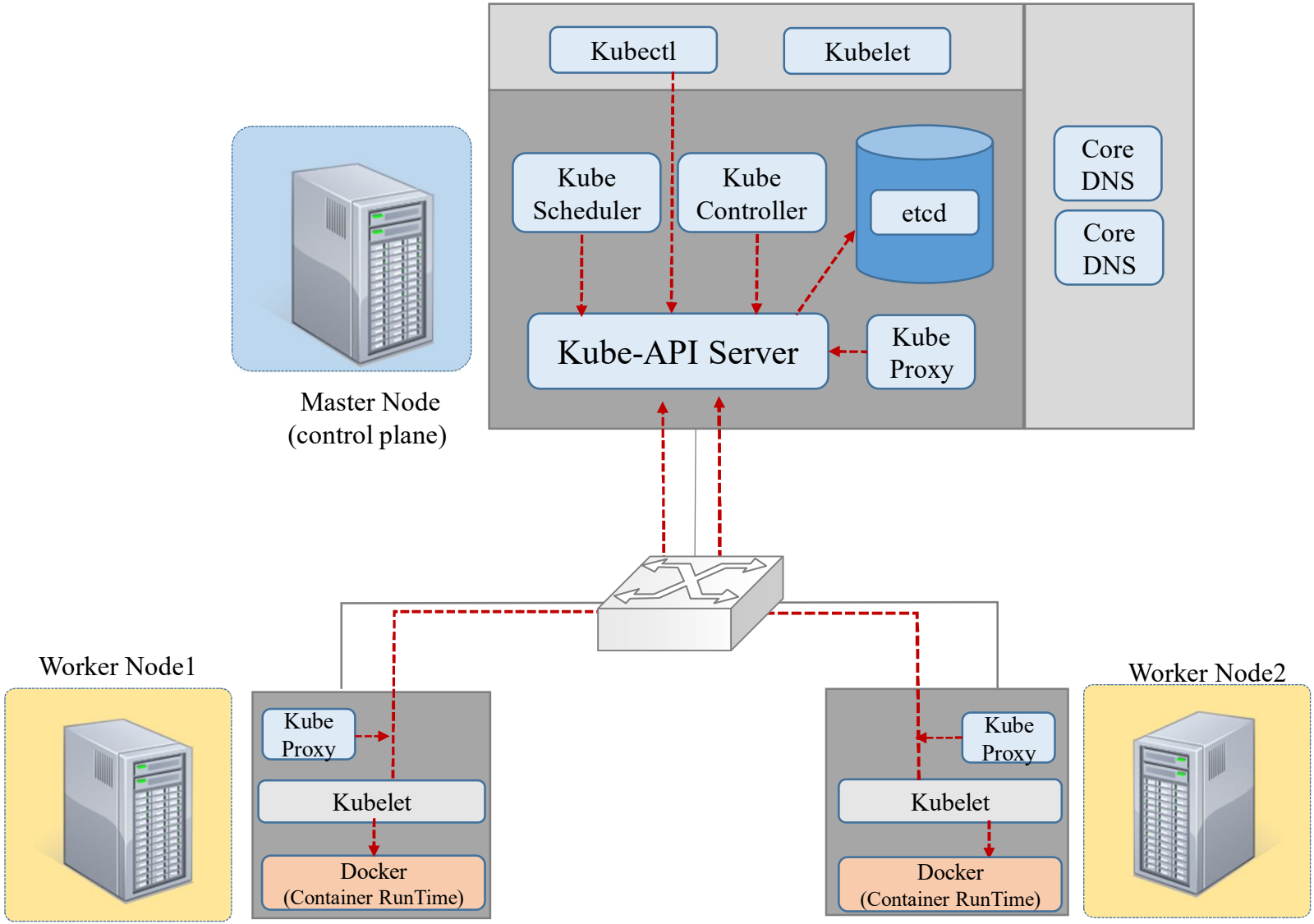
명령어 kubectl

#kubectl create deploy web --image=hub.test.com/nginx



kubectl

- k8s에게 원하는 작업을 요청 시 사용하는 명령어
- k8s cluster를 관리하는 동작은 kubectl이라는 Command line interface로 실행
  - K8s 자원들의 생성, 업데이트, 삭제 (create, update, delete)
  - 디버그, 모니터링, 장애처리(log, exec, cp, top, attach..)
  - 클러스터 관리(cordon, top, drain, traint…)

kubectl --help

```
root@masternode:~# kubectl --help
kubectl controls the Kubernetes cluster manager.

 Find more information at: https://kubernetes.io/docs/reference/kubectl/ove

Basic Commands (Beginner):
  create        Create a resource from a file or from stdin
  expose        Take a replication controller, service, deployment or pod a
  run           Run a particular image on the cluster
  set           Set specific features on objects

Basic Commands (Intermediate):
  explain       Get documentation for a resource
  get           Display one or many resources
  edit          Edit a resource on the server
  delete        Delete resources by file names, stdin, resources and names,
```

kubectl 명령어 형식

kubectl [command] [TYPE] [NAME] [flags]

Command	자원에 실행되는 동작	create, get, delete
TYPE	자원타입	pod, service, ingress
NAME	자원이름	
Flags	부가적으로 설정할 옵션	--help, -o wide

(ex) kubectrl get pod WEBServer –o wide

➔ WEBServer 이름을 가진 Pod 자원정보를 자세히 확인

[참고] 명령어 자동 완성 자동완성

```
source <(kubectl completion bash)

source <(kubeadm completion bash)

echo 'source <(kubectl completion bash)' >>~/.bashrc

echo 'source <(kubeadm completion bash)' >>~/.bashrc
```

kubectl --help

kubuctl run --help

```
root@masternode:~# kubectl run --help
Create and run a particular image in a pod.

Examples:
  # Start a nginx pod
  kubectl run nginx --image=nginx

  # Start a hazelcast pod and let the container expose port 5701
  kubectl run hazelcast --image=hazelcast/hazelcast --port=5701

  # Start a hazelcast pod and set environment variables "DNS_DOMAIN=clus
container
  kubectl run hazelcast --image=hazelcast/hazelcast --env="DNS_DOMAIN=cl

  # Start a hazelcast pod and set labels "app=hazelcast" and "env=prod"
  kubectl run hazelcast --image=hazelcast/hazelcast --labels="app=hazelc

  # Dry run; print the corresponding API objects without creating them
  kubectl run nginx --image=nginx --dry-run=client
```

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
```

실습 1.

- kubectl get nodes
- kubectl get nodes -o wide
- kubectl describe node master

- watch kubectl get pod –o wide

실습 2.

- kubectl run web --image=nginx:1.14 --port 80
- kubectl get pods
- kubectl describe pod web
- kubectl get pods -o wide
- kubectl get pods web -o wide
- curl 10.42.0.1

실습 3.

- kubectl create deployment mainserver --image=httpd --replicas=3
- kubectl get deployments.apps
- kubectl describe deployments.apps mainserver
- kubectl get pods
- kubectl get pods -o wide
- kubectl get pod mainserver-6c9cbf6cb7-nrxl7
- kubectl get pod mainserver-6c9cbf6cb7-nrxl7 -o wide
- curl 10.40.0.1

실습 4.

[Home 페이지 수정]

- kubectl exec web -it -- /bin/bash/
  cd /usr/share/nginx/html
  cat index.html
  echo "HEllo~~"" > index.html
- exit

- curl 10.42.0.1
- kubectl logs web

ufw disable

실습 5.

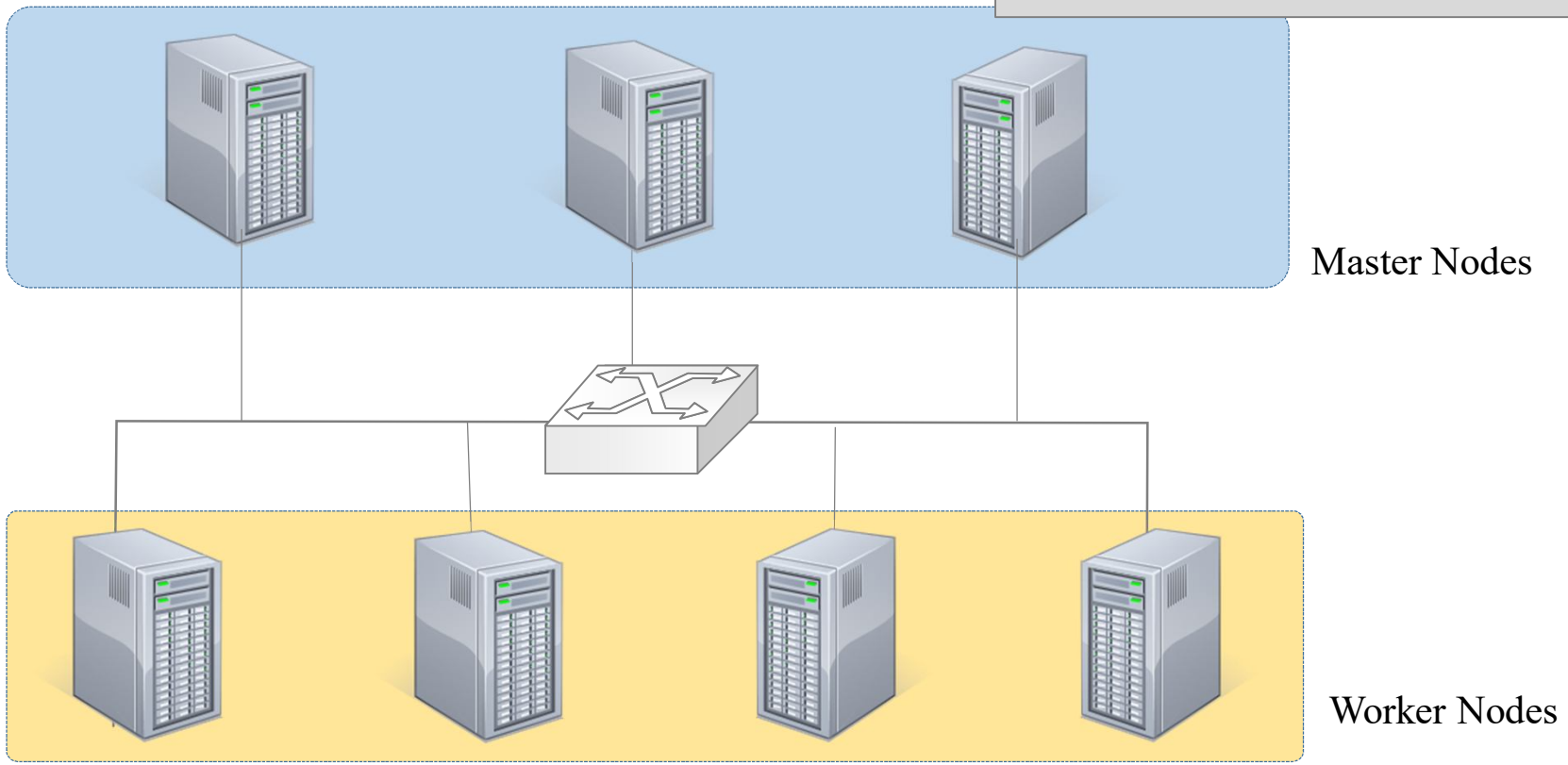
- kubectl delete pod web
- kubectl get pods
- kubectl delete deployment.apps mainserver

K8S Architecture

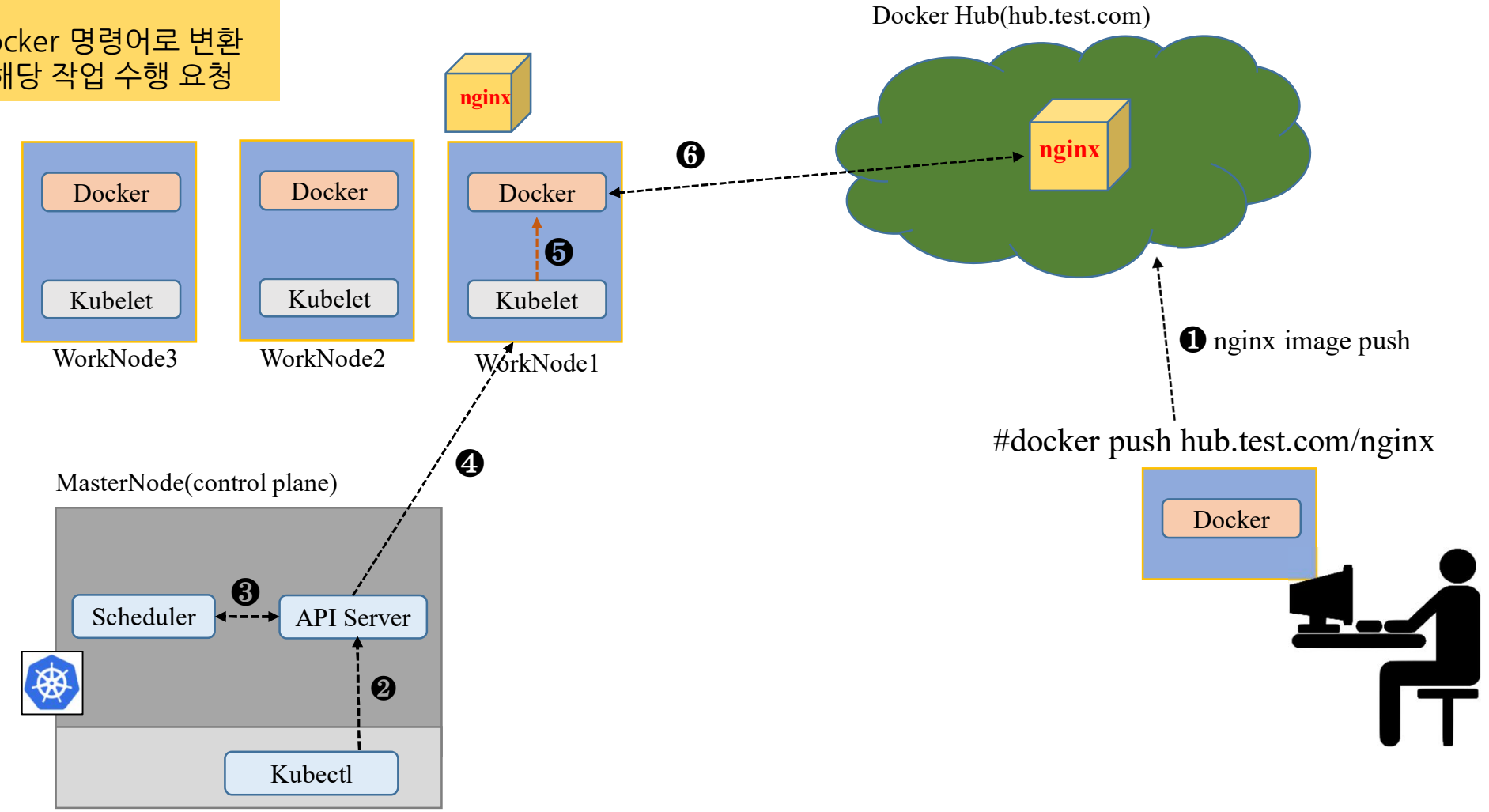
K8S 클러스터 전체구조

- K8S 클러스터는 크기 두 종류의 서버로 구성
  - Master Nodes : Cluster 관리 노드들
  - Worker Nodes : Container를 실행시키는 노드들

- Leader master : 1대
- Standby master : 2대
- 안정적으로 운영하기 위해 5대 구성도 가능

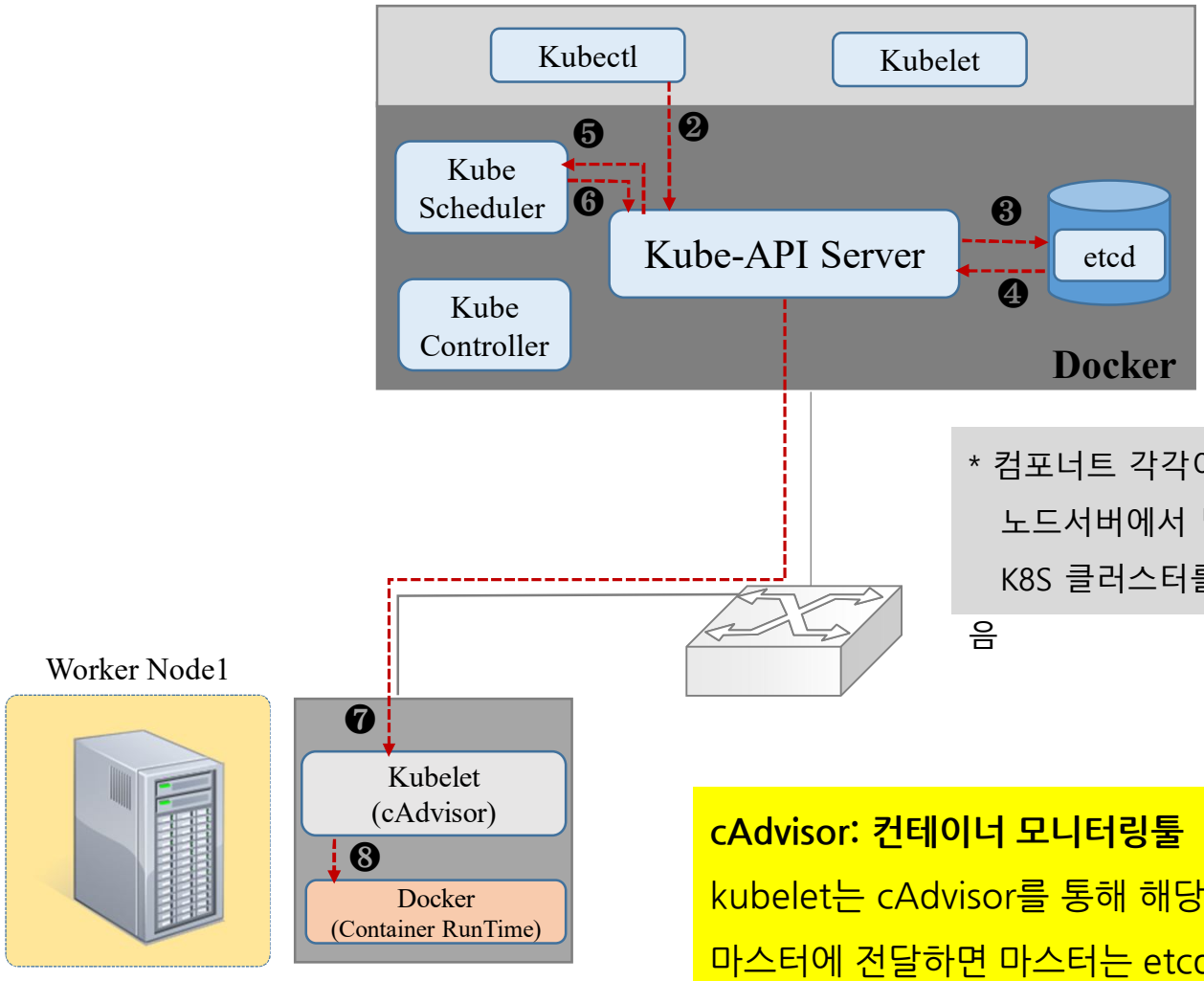


Kubelet
- API요청을 docker 명령어로 변환
- docker에게 해당 작업 수행 요청



#kubectl create deploy web --image=hub.test.com/nginx

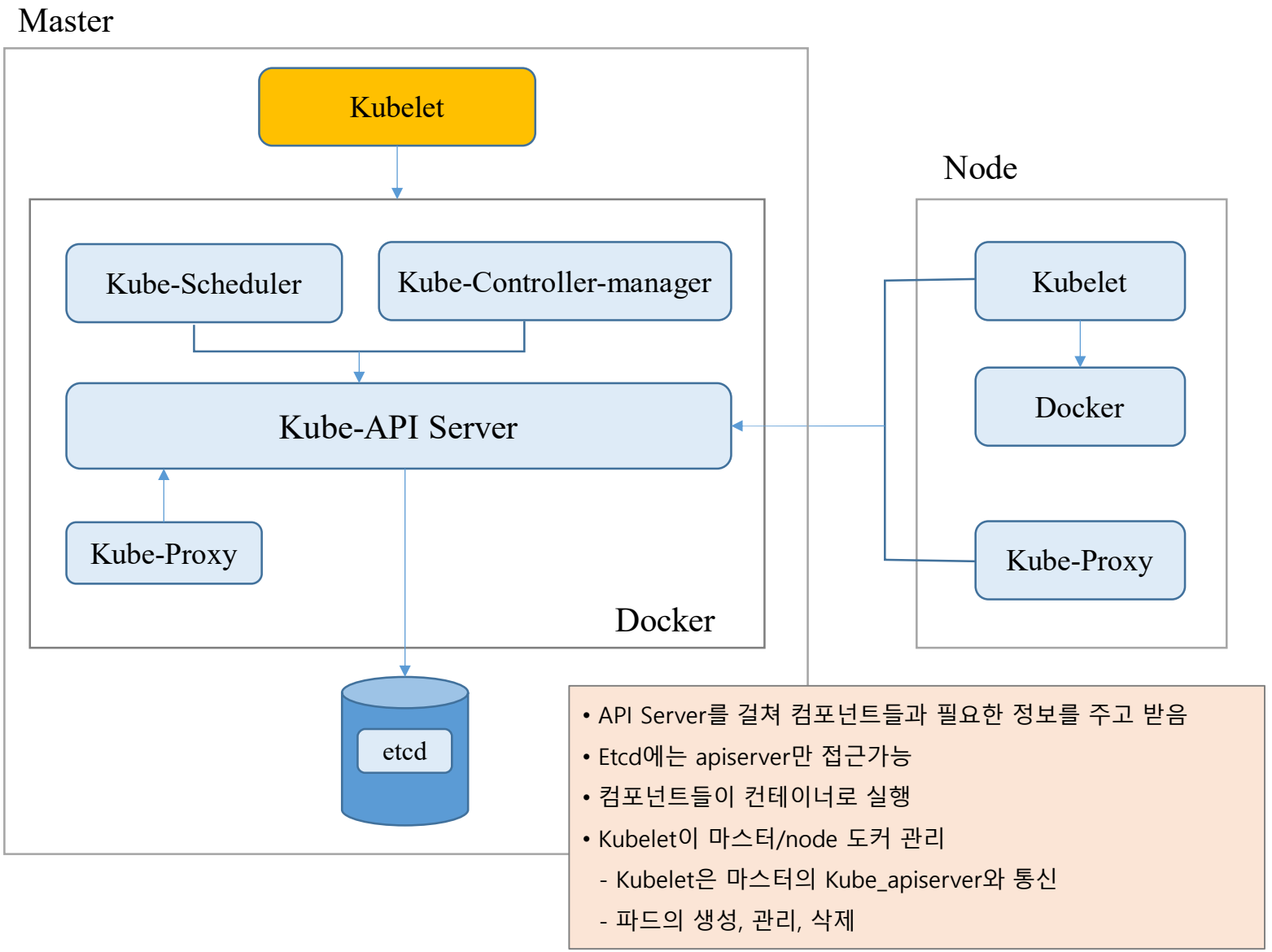
❶ *#kubectl create deploy web --image=hub.test.com/nginx*



* 컴포넌트 각각이 다른 마스터나 노드서버에서 별개로 실행되어도 K8S 클러스터를 운영하는데 이상이 없음

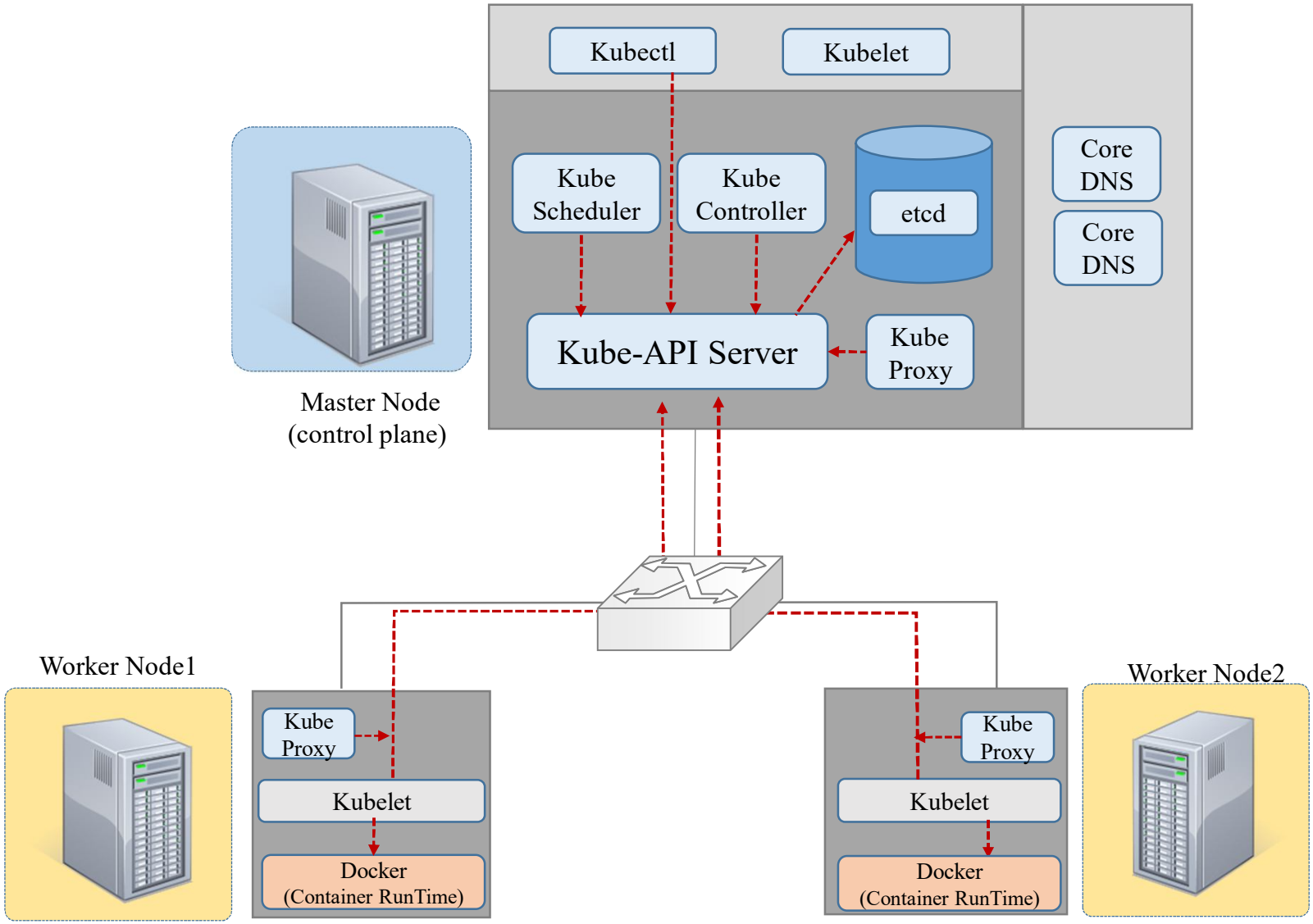
cAdvisor: 컨테이너 모니터링툴

kubelet는 cAdvisor를 통해 해당 노드의 상태정보를 수집하여 마스터에 전달하면 마스터는 etcd에 해당 정보를 저장

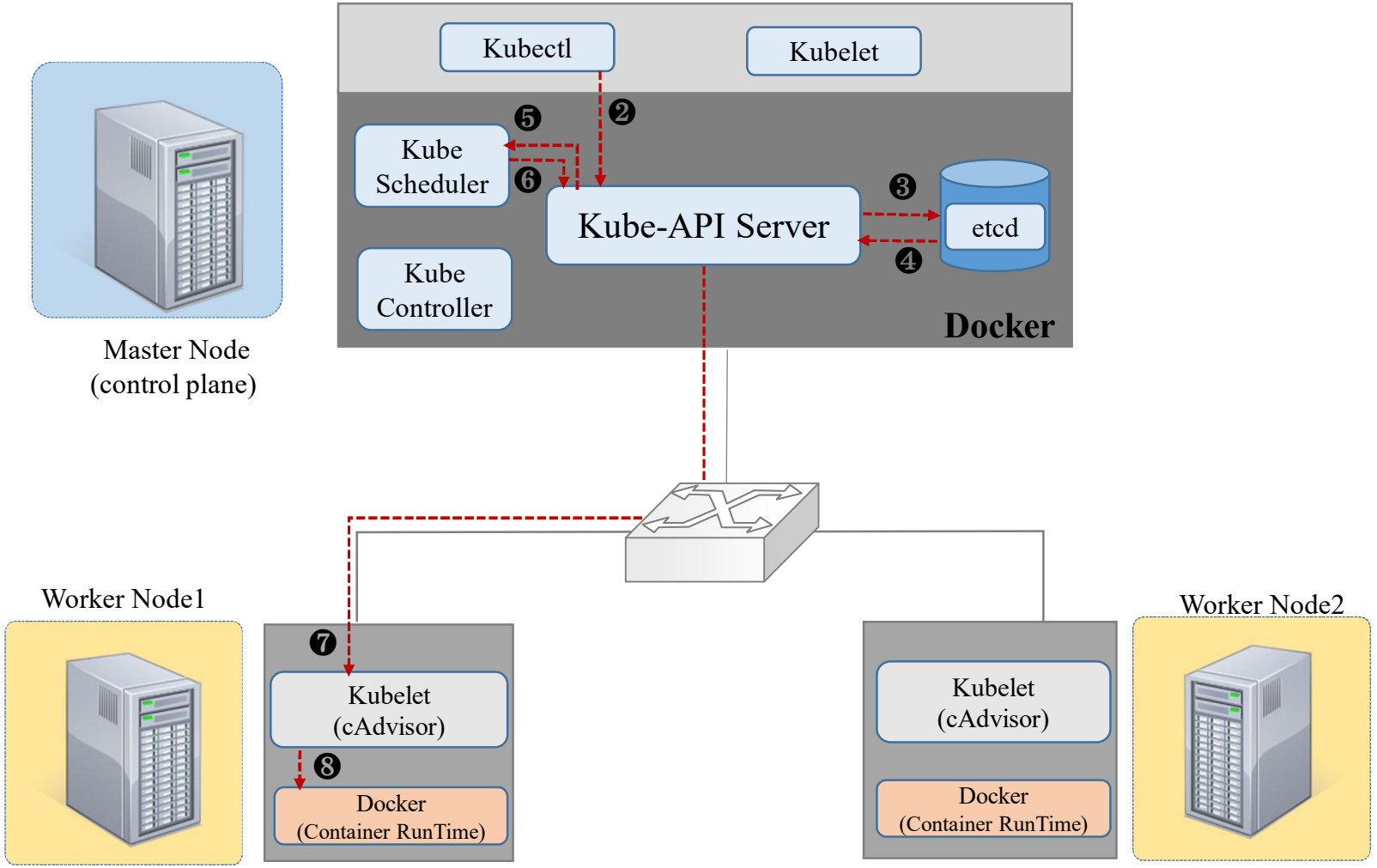


- API Server를 걸쳐 컴포넌트들과 필요한 정보를 주고 받음
- Etcd에는 apiserver만 접근가능
- 컴포넌트들이 컨테이너로 실행
- Kubelet이 마스터/node 도커 관리
  - Kubelet은 마스터의 Kube_apiserver와 통신
  - 파드의 생성, 관리, 삭제

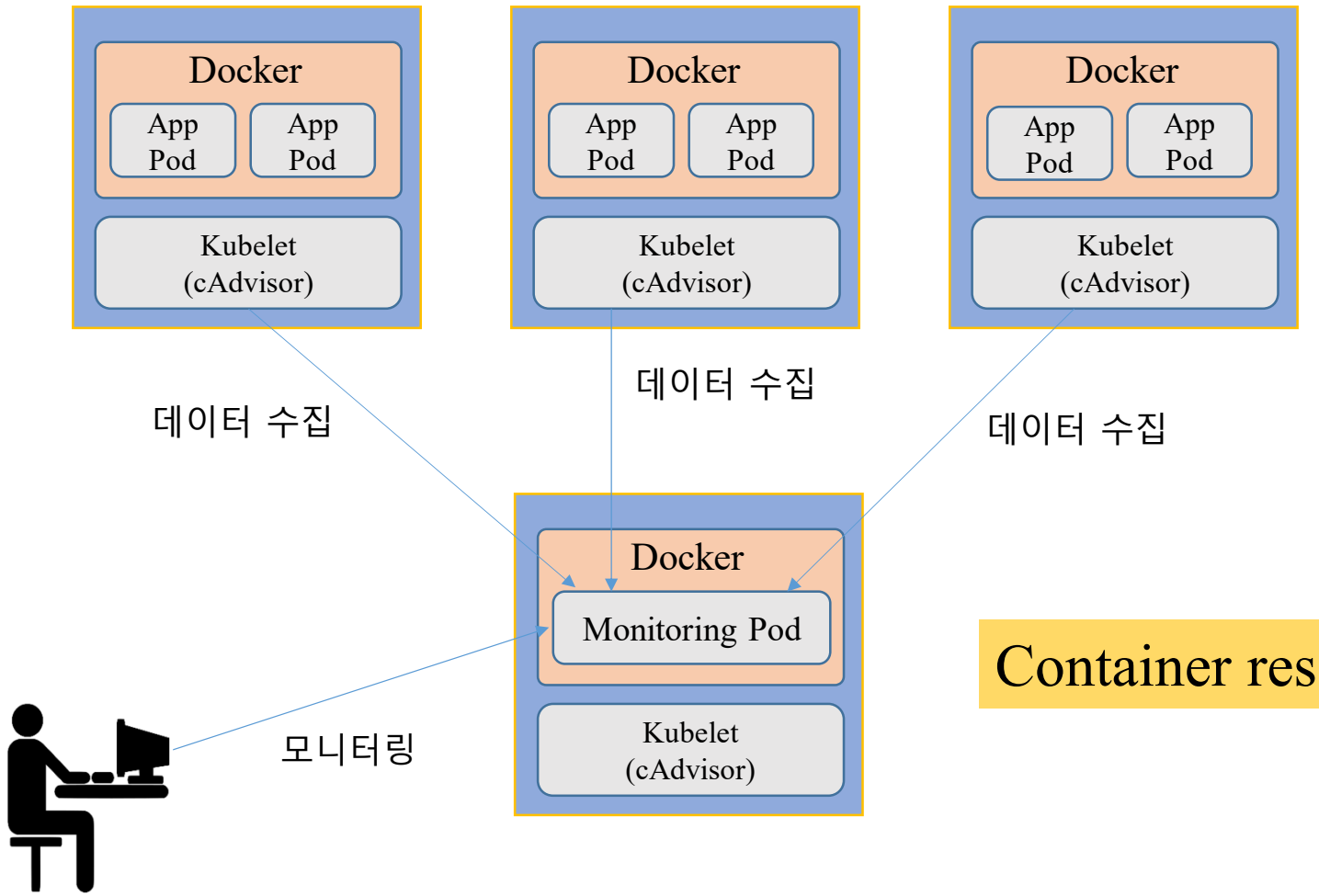
#kubectl create deploy web --image=hub.test.com/nginx



❶ *#kubectl create deploy web --image=hub.test.com/nginx*

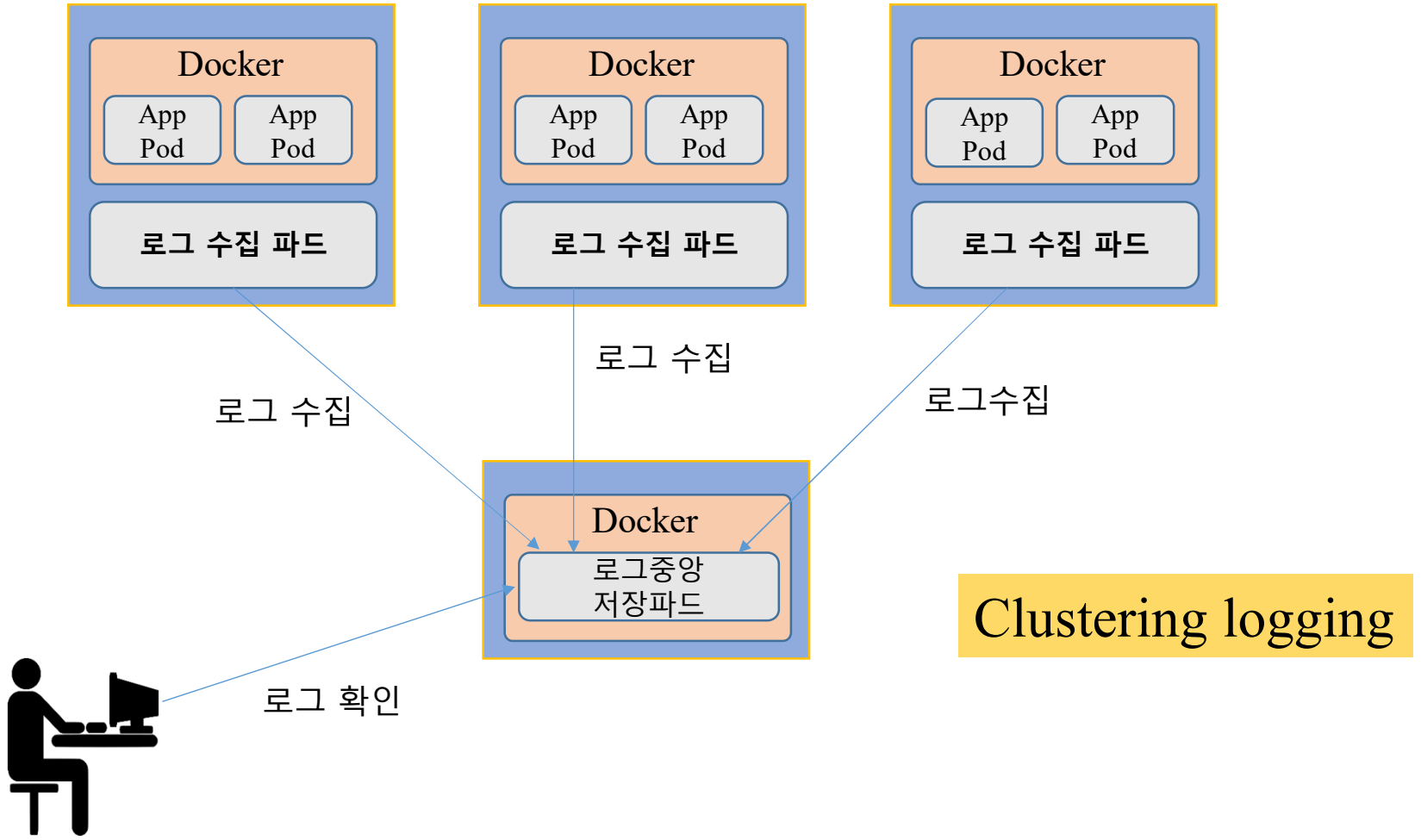


* cAdvisor 동작원리



Container resource monitoring

* 클러스터 로깅 동작원리


Docker
App Pod
App Pod
로그 수집 파드
Docker
App Pod
App Pod
로그 수집 파드
Docker
App Pod
App Pod
로그 수집 파드
로그 수집
로그 수집
로그수집
Docker
로그중앙 저장파드
로그 확인
Clustering logging