

자동 배포(Docker&Jenkins) + k8s

2023.02.03 실습코치 한진성



가이드 순서

- ✓ Docker
- ✓ 자동 배포 (Docker 활용)
- ✓ 쿠버네티스



Docker

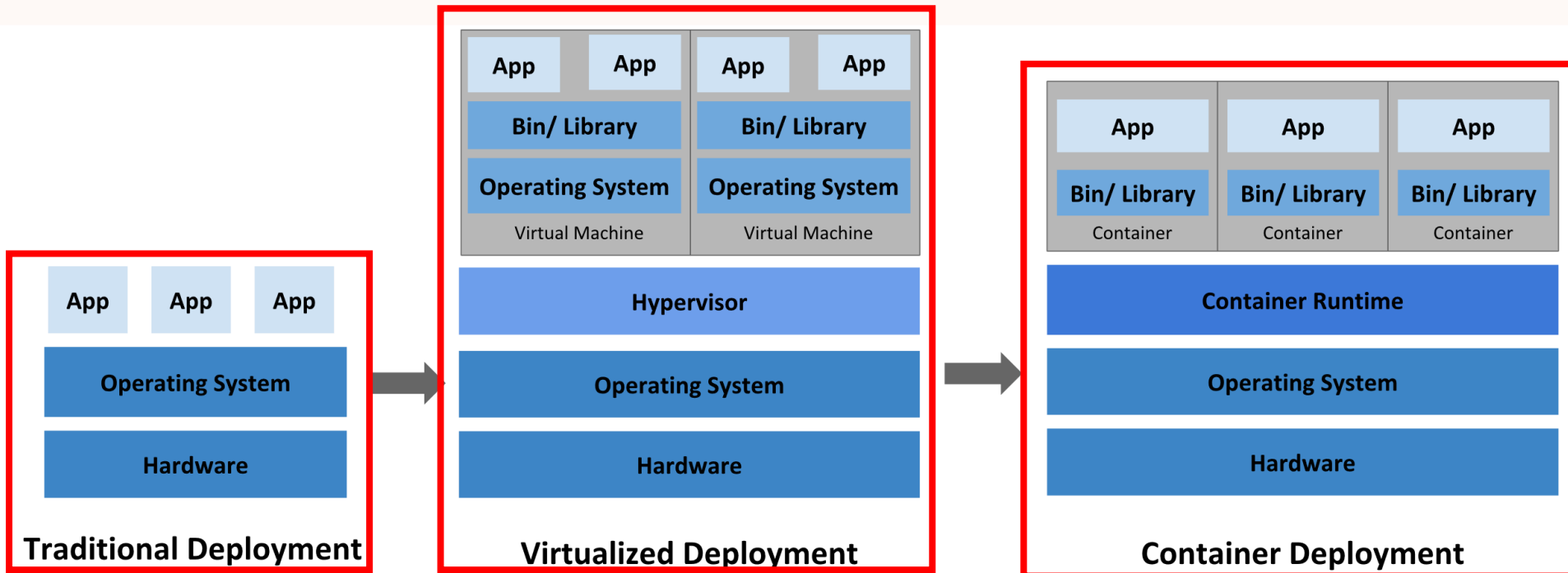


서버 관리 방식의 변화



함께가요 미래로!
Enabling People

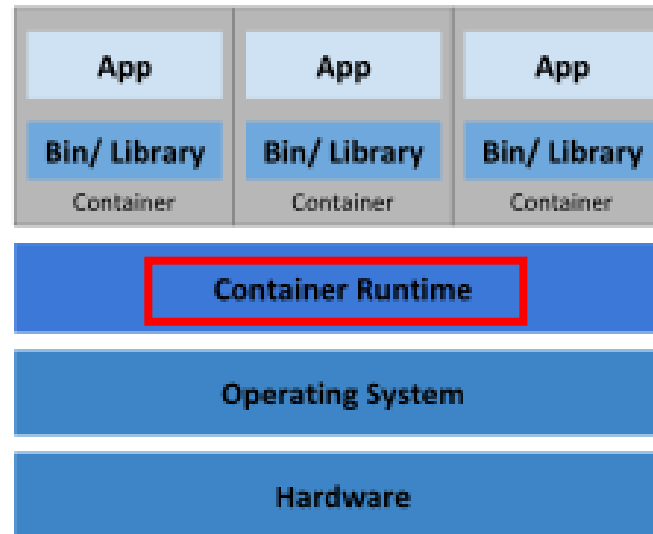
- ✓ 전통적인 배포 : 1개의 OS에 여러 앱을 배포
- ✓ 가상화 배포 : 1개의 OS에 Virtual Machine 으로 새로운 OS를 띄워 각각의 OS에 앱을 배포
- ✓ 컨테이너 배포 : 컨테이너의 개념을 사용하여 1개의 OS에 여러 개 앱을 배포



컨테이너란?



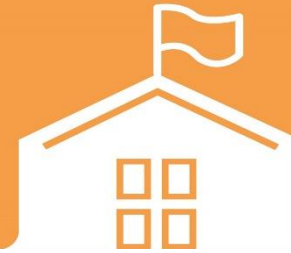
- ✓ 어디서든 실행될 수 있도록 App 코드가 의존성 라이브러리와 함께 패키징 된 것
- ✓ 각각의 컨테이너를 마치 별도의 서버처럼 사용할 수 있게 함
- ✓ 이러한 컨테이너를 다루는 도구를 **컨테이너 런타임** 이라고 부른다.



Container Deployment

- ✓ 가볍다!
- ✓ 일관성 있는 환경 보장!
- ✓ 배포 편리!

도커란?



- ✓ 컨테이너 런타임 도구의 대표 주자.
- ✓ 리눅스 기반 APP들을 컨테이너로 실행하고 관리할 수 있도록 하는 오픈소스 프로젝트

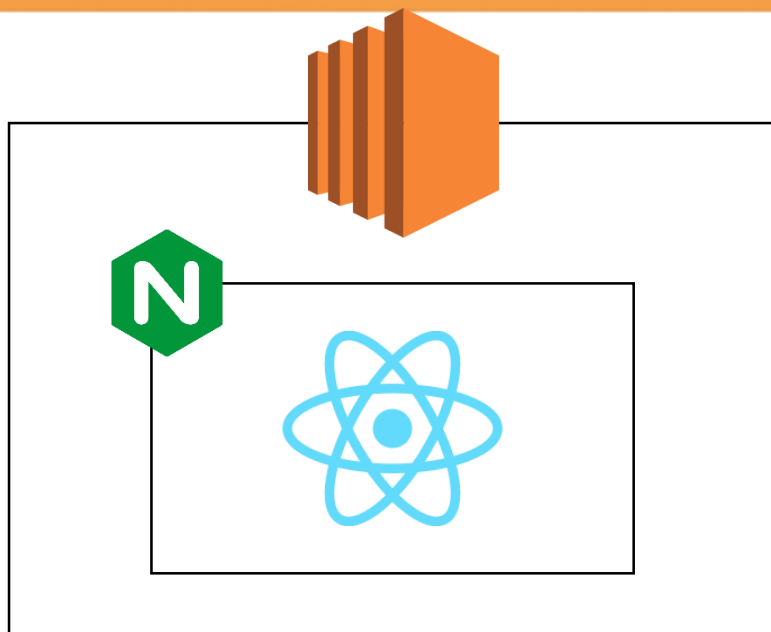


docker

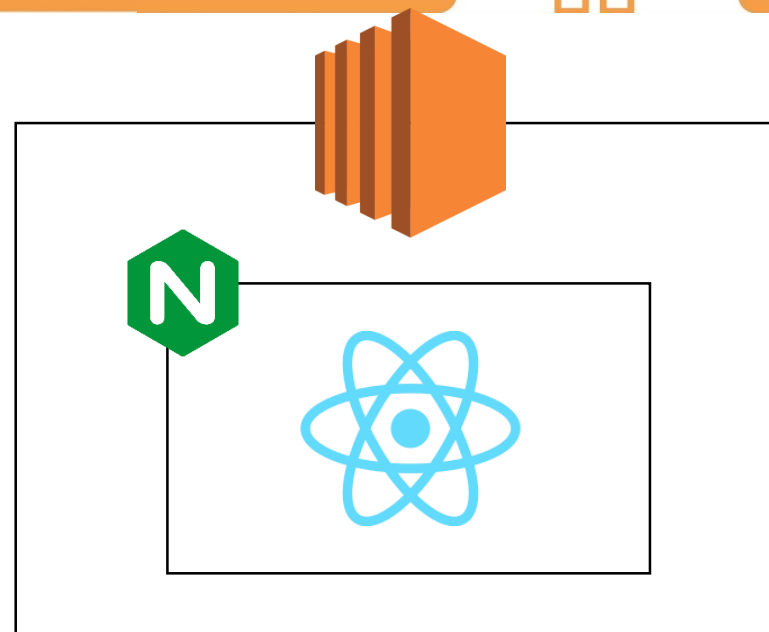
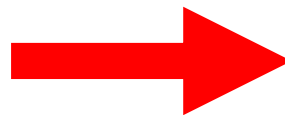
도커로 할 수 있는 배포 상태



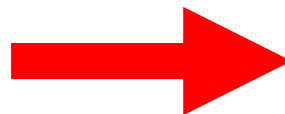
함께가요 미래로!
Enabling People



npm install
npm run build
웹 서버 설치 (Ex : Nginx)
Build 파일 이동
Nginx 설정 후 Start



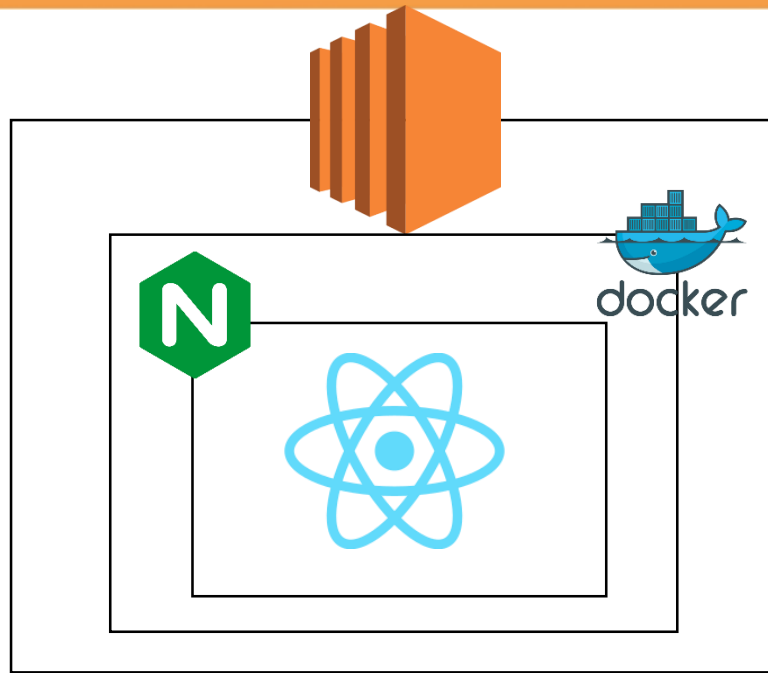
npm install
npm run build
웹 서버 설치 (Ex : Nginx)
Build 파일 이동
Nginx 설정 후 Start



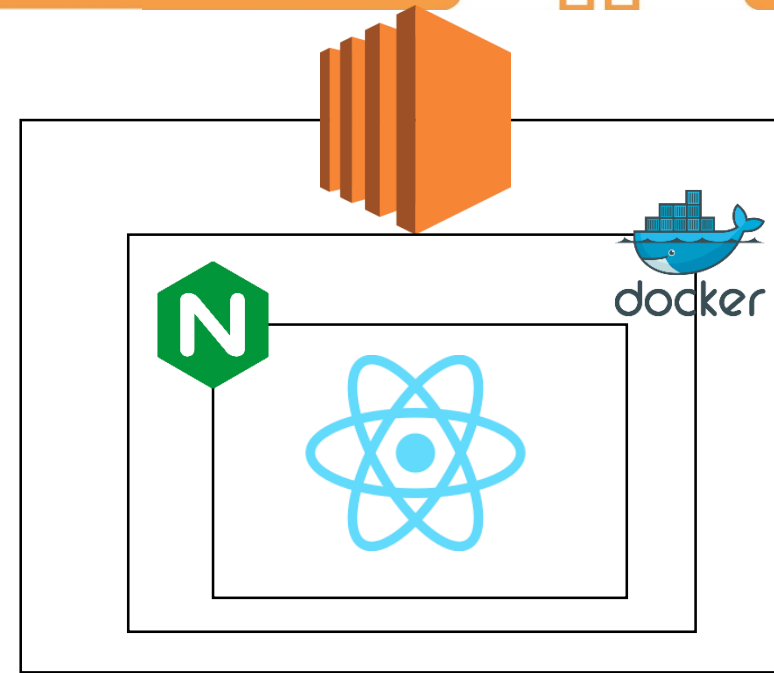
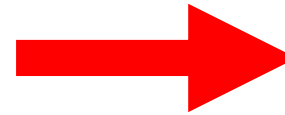
도커로 할 수 있는 배포 상태



함께가요 미래로!
Enabling People



docker push
docker save



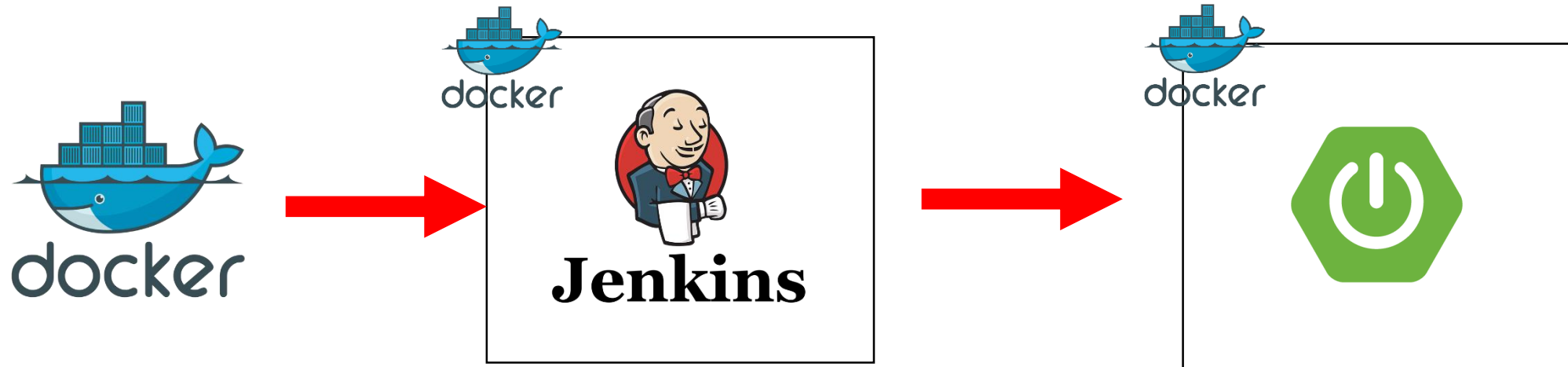
docker pull
docker run



도커 에서 또 도커를?



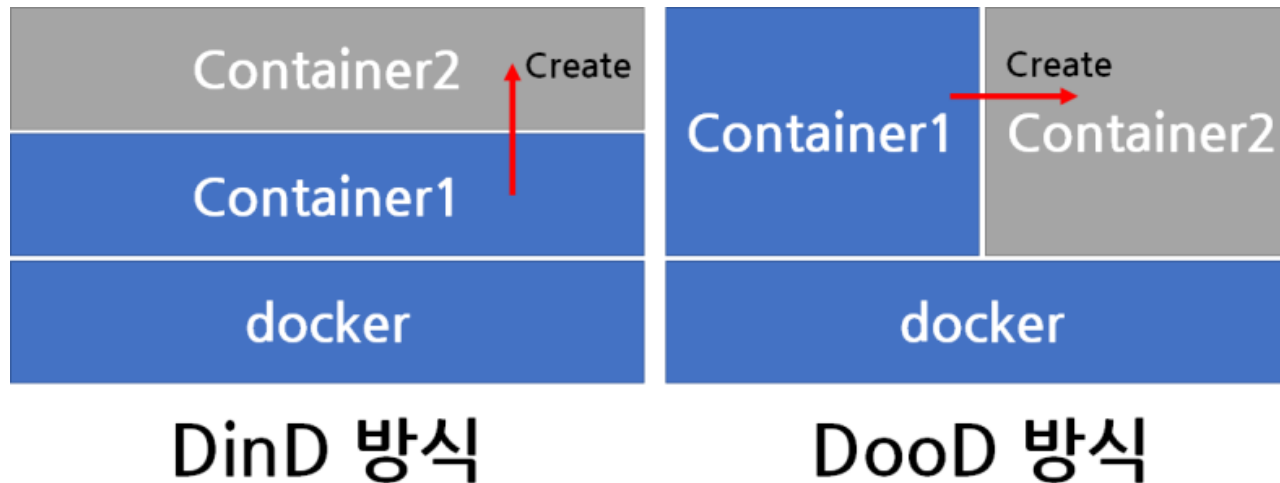
- ✓ Docker 컨테이너 내부에서 또다른 Docker image를 build, run 할 수 있다.
- ✓ Docker 컨테이너로 젠킨스를 띄우고, 젠킨스에서 도커를 빌드해서 또 띄우고...





DinD? DooD?

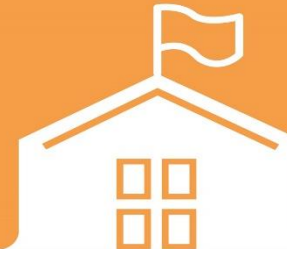
- ✓ DinD : Docker in Docker (--privileged 옵션을 이용해 권한을 부여)
- ✓ DooD : Docker out of Docker (/var/run/docker.sock 파일을 Volume으로 연결)



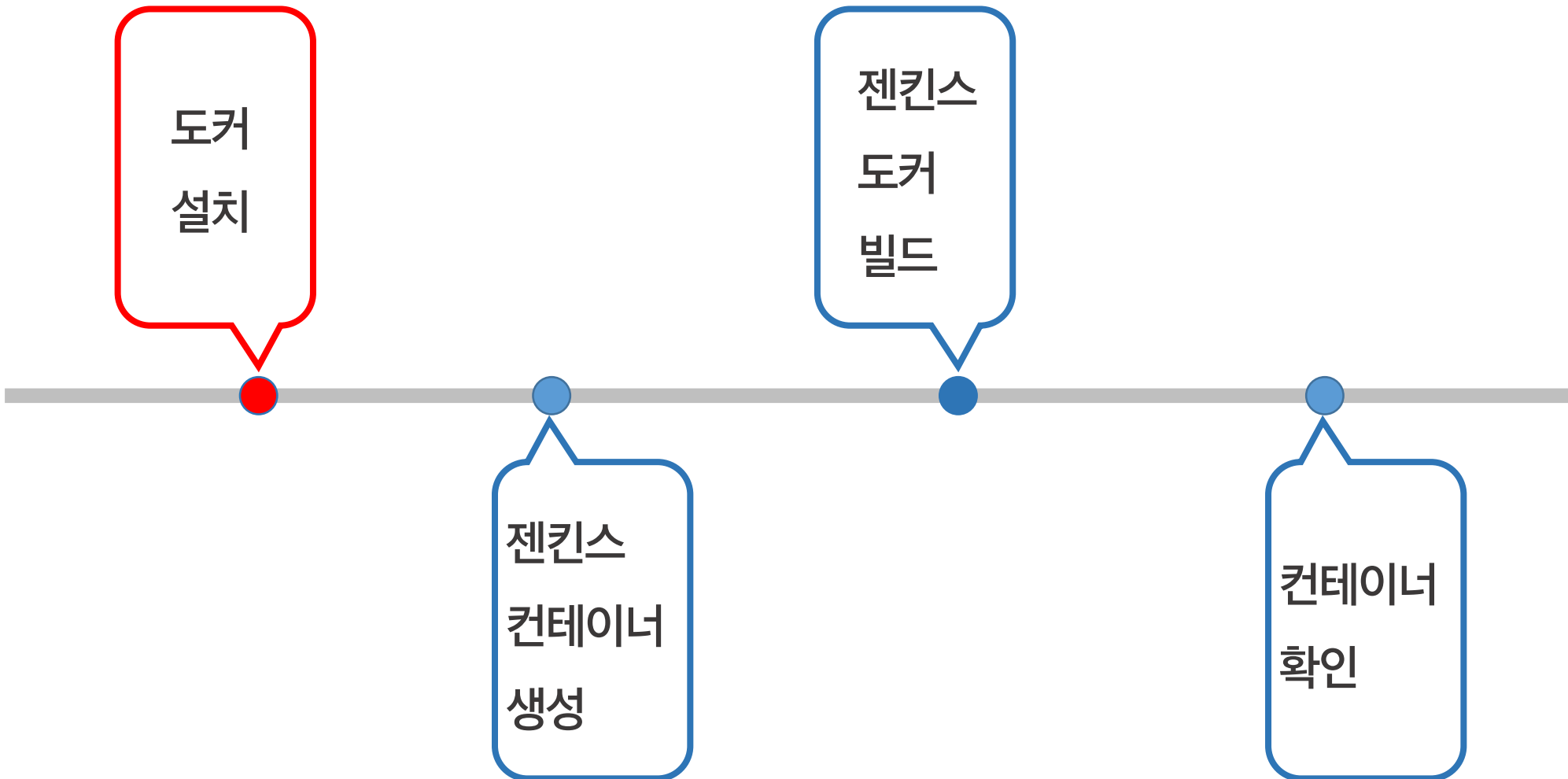
자동 배포



Docker 컨테이너로 젠킨스 설치



함께가요 미래로!
Enabling People



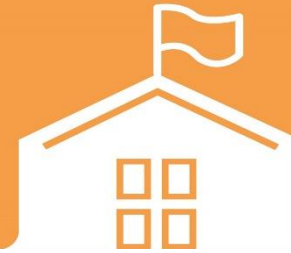
Docker 설치하기



✓ 사전 패키지 설치하기

```
sudo apt update
sudo apt-get install -y ca-certificates \
    curl \
    software-properties-common \
    apt-transport-https \
    gnupg \
    lsb-release
```

Docker 설치하기

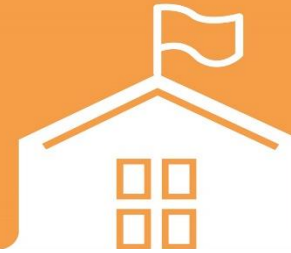


✓ GPG Key 다운로드 : 패키지 유효성 검증

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Docker 설치하기



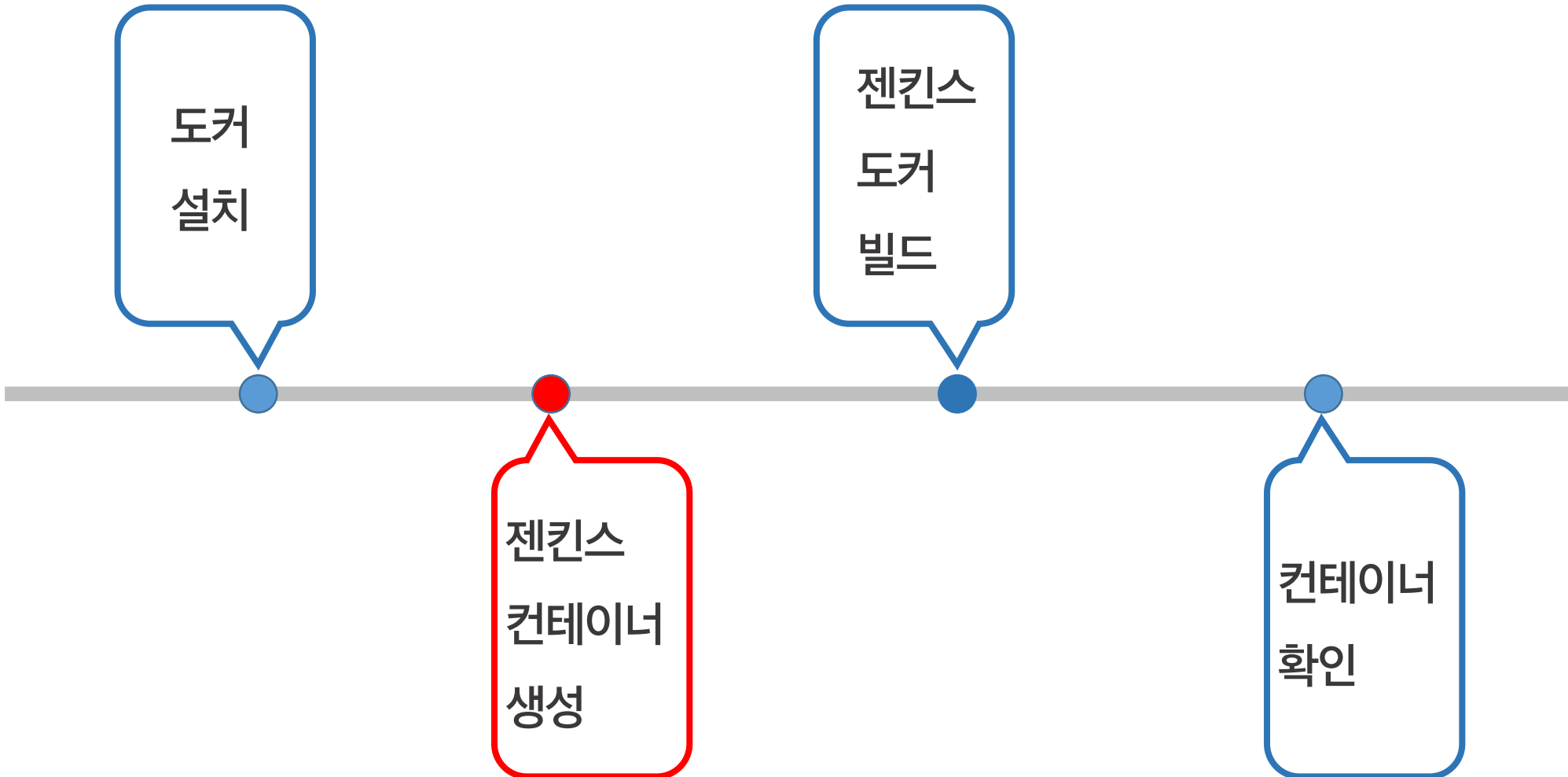
✓ Docker 설치

```
sudo apt update  
sudo apt install docker-ce docker-ce-cli containerd.io docker-compose
```

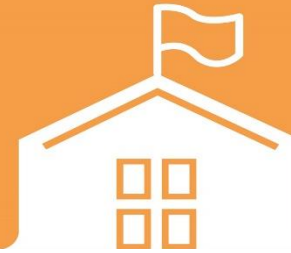
Docker 컨테이너로 젠킨스 설치



함께가요 미래로!
Enabling People



Docker 컨테이너로 젠킨스 설치



✓ Docker-compose 이용하여 컨테이너 생성

```
version: '3'

services:
  jenkins:
    image: jenkins/jenkins:lts
    container_name: jenkins
    volumes:
      - /usr/bin/docker:/usr/bin/docker
      - /var/run/docker.sock:/var/run/docker.sock
      - /jenkins:/var/jenkins_home
    ports:
      - "9090:8080"
    privileged: true
    user: jenkins
```

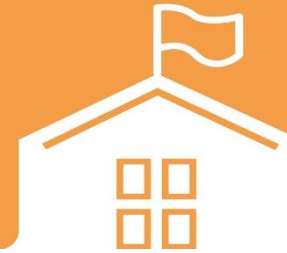
Docker 컨테이너로 젠킨스 설치



✓ Docker-compose 이용하여 컨테이너 생성

```
ubuntu@ip-172-31-62-140: ~$ vim docker-compose.yml
ubuntu@ip-172-31-62-140: ~$ ls
docker-compose.yml
ubuntu@ip-172-31-62-140: ~$ docker-compose up -d
```

Docker 컨테이너로 젠킨스 설치



함께가요 미래로!
Enabling People

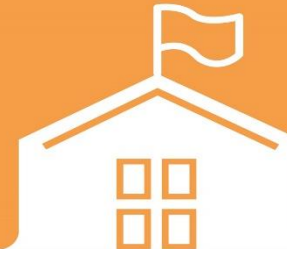
✓ Docker-compose 이용하여 컨테이너 생성

```
ubuntu@ip-172-31-62-140: ~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
3dc6f6b83177	jenkins/jenkins:lts	"/usr/bin/tini -- /u..."	About a minute ago	Up About a minute	50000/tcp, 0.0.0.0:9090->8080/tcp, :::9090->8080/tcp	jenkins

```
ubuntu@ip-172-31-62-140: ~$
```

Docker 컨테이너로 젠킨스 설치



- ✓ Jenkins administrator password 찾기 : `docker logs <container_name>`

```
*****
*****
*****

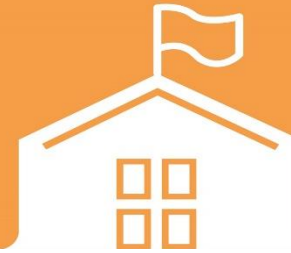
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
```

```
613e613f92044ca98db1adc4e2aa38f
```

```
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
```

```
*****
*****
*****
```

젠킨스에 도커 플러그인 설치



함께가요 미래로!
Enabling People

Plugin Manager

업데이트된 플러그인 목록 설치 가능 설치된 플러그인 목록 고급

Q. docker

Install Name ↓

Docker 1.2.9

Cloud Providers Cluster Management docker

☒ This plugin integrates Jenkins with Docker

This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

Docker Commons 1.21

☒ Library plugins (for use by other plugins) docker

Provides the common shared functionality for various Docker-related plugins.

Docker Pipeline 521.v1a_a_dd2073b_2e

☒ pipeline DevOps Deployment docker

28 days ago

Build and use Docker containers from pipelines.

Docker API 3.2.13-37.vf3411c9828b9

Library plugins (for use by other plugins) docker

☒ This plugin provides `docker-java` API for other plugins.

4 mo 20 days ago

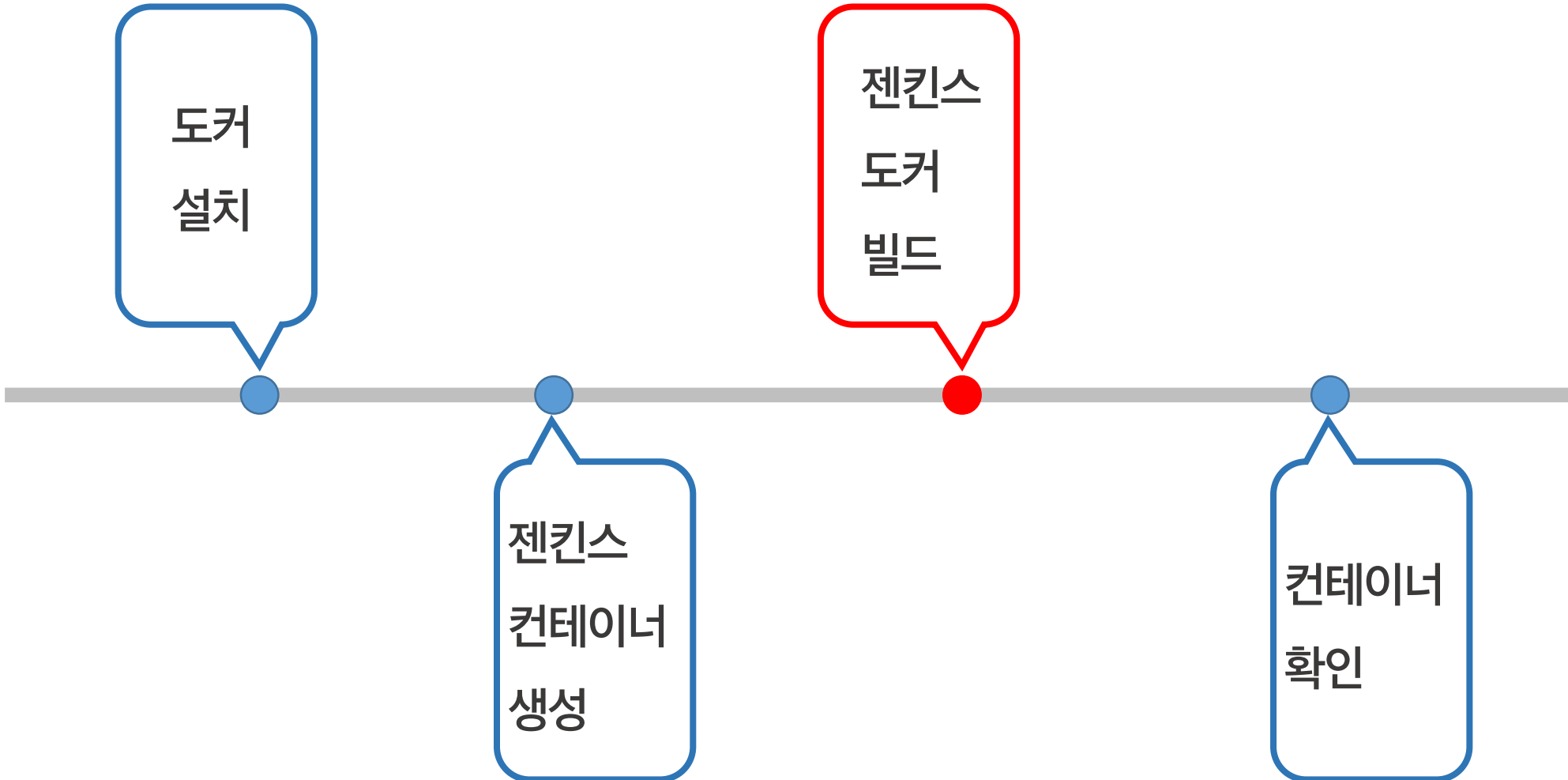
This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.

docker-build-step 2.8

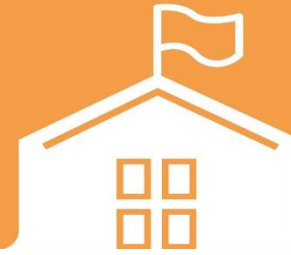
Docker Build & Run



함께가요 미래로!
Enabling People

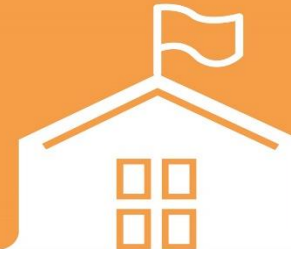


Dockerfile과 Docker 명령어



- ✓ Dockerfile : Docker에서 이미지를 생성할 때 작성하는 파일, 여러 지시어를 사용해 이미지를 생성할 수 있다.
- ✓ 서버를 도커를 통해 빌드를 할 때, 빌드할 디렉토리에 해당 Dockerfile이 있어야 한다.

Dockerfile 생성



```
FROM adoptopenjdk/openjdk11 AS builder
COPY gradlew .
COPY gradle gradle
COPY build.gradle .
COPY settings.gradle .
COPY src src
RUN chmod +x ./gradlew
RUN ./gradlew bootJAR

FROM adoptopenjdk/openjdk11
COPY --from=builder build/libs/*.jar app.jar
EXPOSE 10080
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

- ✓ FROM : docker에서 base image 지정
- ✓ WORKDIR : shell의 cd와 유사, 작업 디렉토리를 지정함
- ✓ RUN : 커맨드를 실행시킴
- ✓ ENTRYPOINT : 이미지를 컨테이너로 올릴 때, 꼭 실행되어야 하는 커맨드 지정
- ✓ COPY : 파일 또는 폴더를 복사함 순서는 <SRC> <DEST>
- ✓ EXPOSE : 컨테이너로 들어오는 트래픽을 listen 하기 위한 포트 지정

Docker Build Steps



Build Steps

≡ Execute shell ?

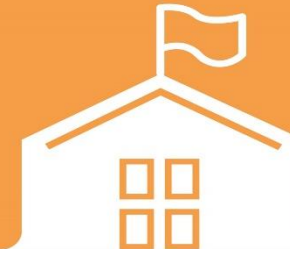
Command

See [the list of available environment variables](#)

```
docker image prune
docker build -t backing ./crudtest-1
if (docker ps | grep "backing"); then docker stop backing; fi
docker run -it -d --rm -p 10080:10080 --name backing backing
echo "Run testproject"
```

- ✓ \$docker ps : 현재 실행중인 컨테이너 확인
- ✓ \$docker build : 도커 이미지 빌드
- ✓ \$docker run : 도커 컨테이너 실행
- ✓ \$docker logs : 도커 로그 확인

Docker Build Steps



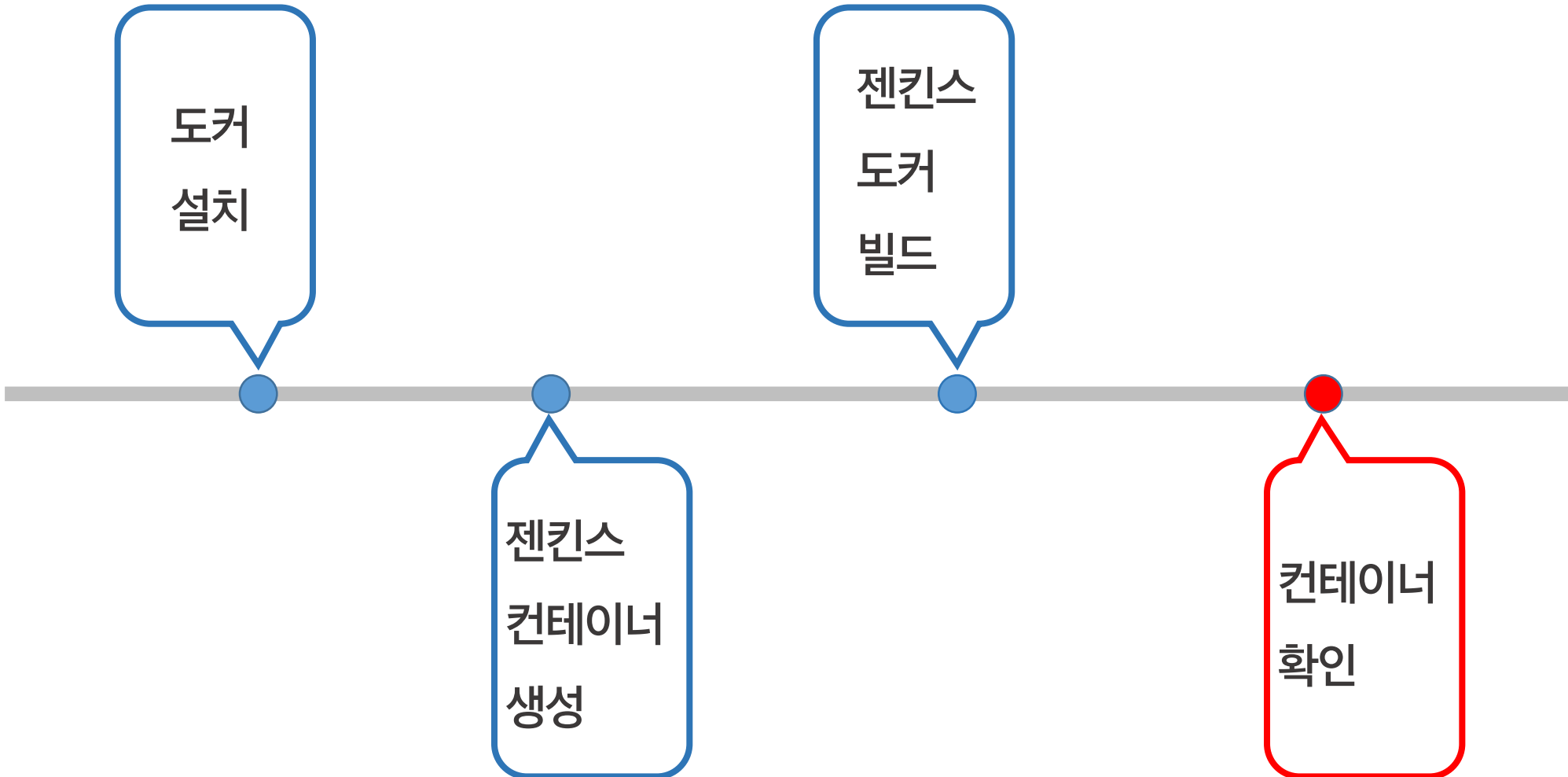
Build History		추이 ▼
Filter builds...		
✓ #64	2023. 2. 2. 오후 9:45	
✓ #63	2023. 2. 2. 오전 1:38	
✓ #62	2023. 2. 1. 오전 2:08	
✓ #61	2023. 1. 31. 오전 4:56	
✓ #60	2023. 1. 31. 오전 4:56	
✓ #59	2023. 1. 31. 오전 4:55	

✓ Jenkins Build하기

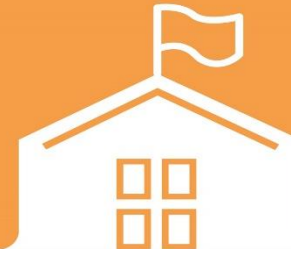
생성된 컨테이너 확인



함께가요 미래로!
Enabling People



생성된 컨테이너 확인



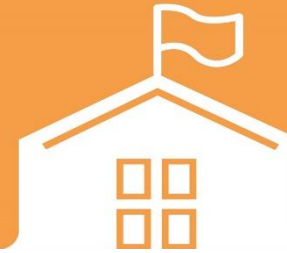
```
ubuntu@ip-172-26-8-156:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
23d1f74a289d	backimg	"java -jar /app.jar"	20 hours ago	Up 20 hours	0.0.0.0:10080->10080/tcp, :::10080->10080/tcp	backimg
35b4931ae1df	jenkins/jenkins:ls	"/usr/bin/tini -- /u..."	44 hours ago	Up 44 hours	0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp	jenkins

쿠버네티스



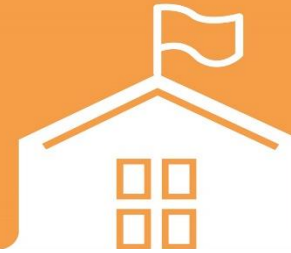
쿠버네티스란?



- ✓ 컨테이너화 된 App의 배포, 확장, 관리를 자동화하는 오픈소스 기반 시스템



쿠버네티스와 도커



함께가요 미래로!
Enabling People

- ✓ 도커 : 컨테이너를 다루는 도구 -> 컨테이너 런타임
- ✓ 쿠버네티스 : 컨테이너 런타임을 이용해 컨테이너를 다루는 도구

쿠버네티스, 도커 지원 중단! 어떻게 해야 할까요?

#DevOps

#쿠버네티스

#도커

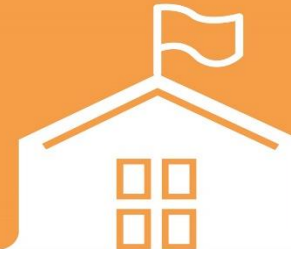


podman



skopeo

쿠버네티스, 왜 쓰죠?



- ✓ 컨테이너들을 관리할 수 있다면?
- ✓ 여러 대의 서버에 컨테이너를 적절히 배치하고 관리하는 오케스트레이션 기능

무중단

자원 효율

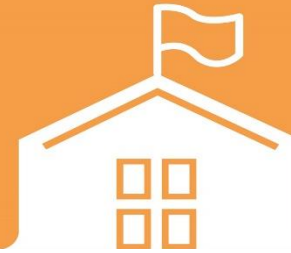
호환성

확장성

쿠버네티스 실습



클러스터 생성



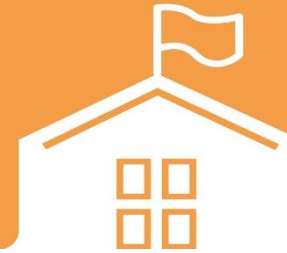
쿠버네티스
설치

쿠버네티스
대시보드

클러스터 생성

Jenkins Pod
생성

기본 세팅



- ✓ WSL2 사용
- ✓ Ubuntu 20.04
- ✓ DockerDesktop 가용 상태

클러스터 생성



함께가요 미래로!
Enabling People

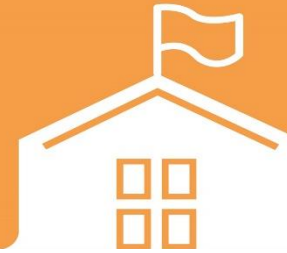
쿠버네티스
설치

쿠버네티스
대시보드

클러스터 생성

Jenkins Pod
생성

Ubuntu update



✓ 패키지 업데이트

```
sudo apt update  
sudo apt upgrade -y
```

Docker, Kubectl 확인



함께가요 미래로!
Enabling People

✓ docker version

```
master@DESKTOP-96CQMV:~$ sudo docker version
Client: Docker Engine - Community
 Cloud integration: v1.0.29
 Version:           20.10.22
 API version:       1.41
 Go version:        go1.18.9
 Git commit:        3a2c30b
 Built:             Thu Dec 15 22:28:22 2022
 OS/Arch:           linux/amd64
 Context:           default
 Experimental:      true

Server: Docker Desktop
Engine:
 Version:           20.10.22
 API version:       1.41 (minimum version 1.12)
 Go version:        go1.18.9
 Git commit:        42c8b31
 Built:             Thu Dec 15 22:26:14 2022
 OS/Arch:           linux/amd64
 Experimental:      false
containerd:
 Version:           1.6.14
 GitCommit:         9ba4b250366a5ddde94bb7c9d1def331423aa323
```

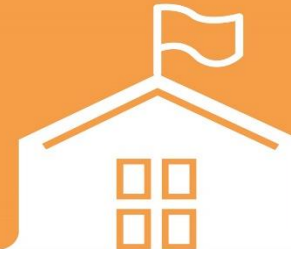
Docker, Kubectl 확인



✓ kubectl version

```
master@DESKTOP-96CQMV:~$ kubectl version
WARNING: This version information is deprecated and will be replaced with the output from kubectl version --
short. Use --output=yaml|json to get the full version.
Client Version: version.Info{Major:"1", Minor:"25", GitVersion:"v1.25.4", GitCommit:"872a965c6c6526caa949f0c
6ac028ef7aff3fb78", GitTreeState:"clean", BuildDate:"2022-11-09T13:36:36Z", GoVersion:"go1.19.3", Compiler:"
gc", Platform:"linux/amd64"}
Kustomize Version: v4.5.7
The connection to the server localhost:8080 was refused - did you specify the right host or port?
```

KinD 설치



- ✓ KinD : Docker Container를 노드로 사용하여 로컬 쿠버네티스 클러스터를 실행하기 위한 도구

```
# kind 최신버전 다운
curl -Lo ./kind https://github.com/kubernetes-
sigs/kind/releases/download/v0.17.0/kind-linux-amd64
# 실행 권한 부여
chmod +x ./kind
# 파일 이동
sudo mv ./kind /usr/local/bin/
```


클러스터 생성



함께가요 미래로!
Enabling People

쿠버네티스
설치

쿠버네티스
대시보드

클러스터 생성

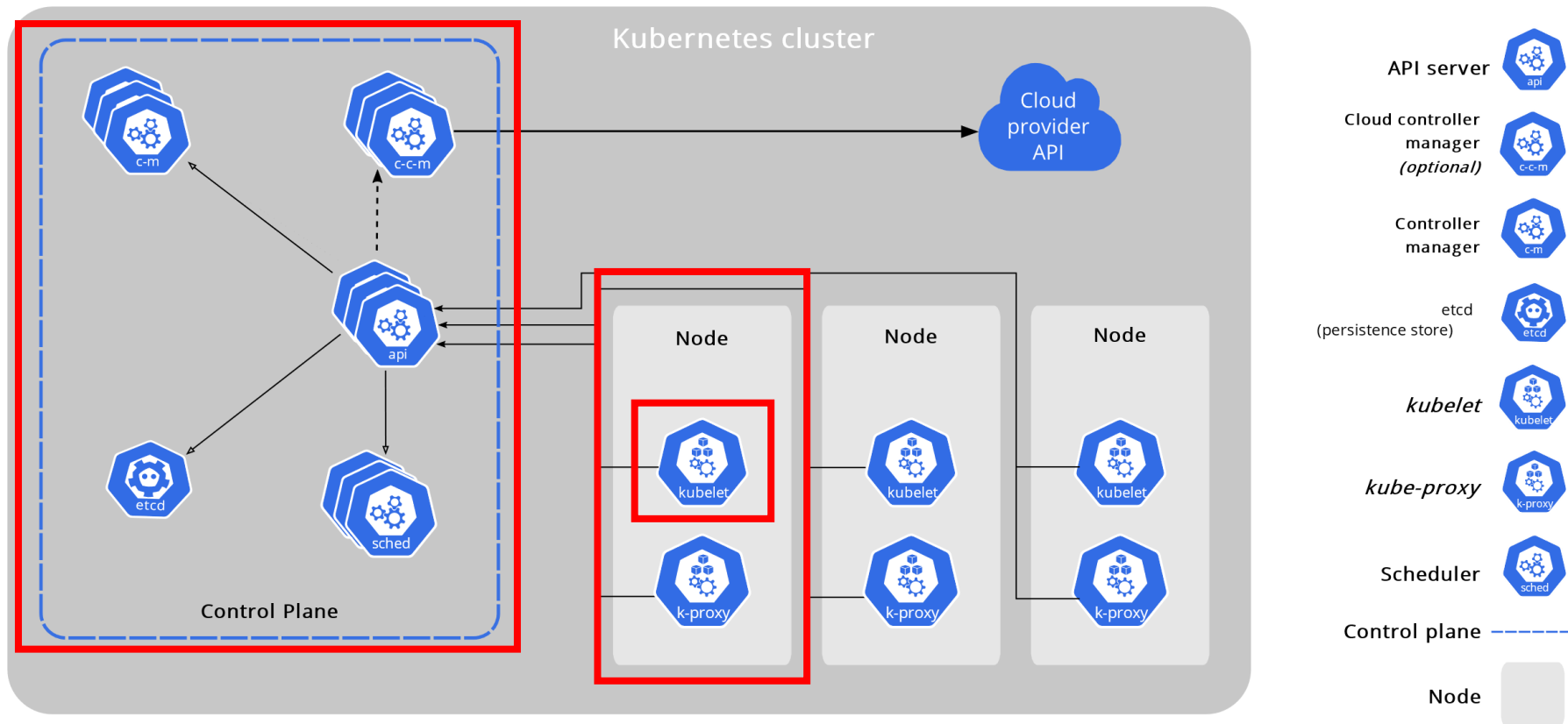
Jenkins Pod
생성

쿠버네티스 클러스터 생성



함께가요 미래로!
Enabling People

✓ 클러스터 : App 컨테이너를 실행하기 위한 일련의 Node 머신



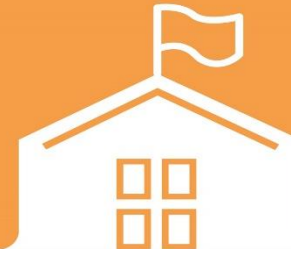
쿠버네티스 클러스터 생성



✓ yaml 파일로 클러스터 생성하기

```
# 2node 클러스터 사용 yaml 파일 생성
cat << EOF > kind-2nodes.yaml
kind: Cluster
apiVersion: kind.x-k8s.io/v1alpha4
nodes:
  - role: control-plane
  - role: worker
EOF
# 생성한 설정 파일로 클러스터 생성하기
kind create cluster --name wslkindmultinodes --config ./kind-2nodes.yaml
# 생성된 노드 확인
kubectl get nodes
```

쿠버네티스 클러스터 생성



- ✓ 클러스터 : App 컨테이너를 실행하기 위한 일련의 Node 머신

```
master@DESKTOP-96CQMVL:~$ kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
wslkindmultinodes-control-plane	Ready	control-plane	5m14s	v1.25.3
wslkindmultinodes-worker	Ready	<none>	4m38s	v1.25.3

쿠버네티스 대시보드



쿠버네티스
설치

쿠버네티스
대시보드

클러스터 생성

Jenkins
컨테이너 생성

쿠버네티스 대시보드



✓ 쿠버네티스 관리 현황을 GUI로 확인

```
# 최신버전 쿠버네티스 대시보드 추가하기
kubectl apply -f
https://raw.githubusercontent.com/kubernetes/dashboard/v2.6.1/aio/deploy/recommended.yaml
# 새로운 namespace 생성 및 내부 Pod 확인하기
kubectl get all -n kubernetes-dashboard
```

쿠버네티스 대시보드



✓ 쿠버네티스 관리 현황을 GUI로 확인

```
master@DESKTOP-96CQMV:~$ kubectl get all -n kubernetes-dashboard
```

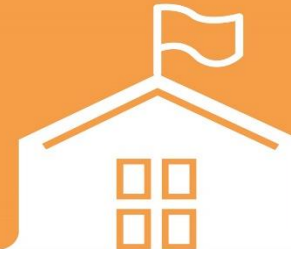
NAME	READY	STATUS	RESTARTS	AGE
pod/dashboard-metrics-scraper-64bcc67c9c-qbb4l	1/1	Running	0	52s
pod/kubernetes-dashboard-66c887f759-lzr9x	1/1	Running	0	52s

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/dashboard-metrics-scraper	ClusterIP	10.96.188.162	<none>	8000/TCP	52s
service/kubernetes-dashboard	ClusterIP	10.96.219.36	<none>	443/TCP	52s

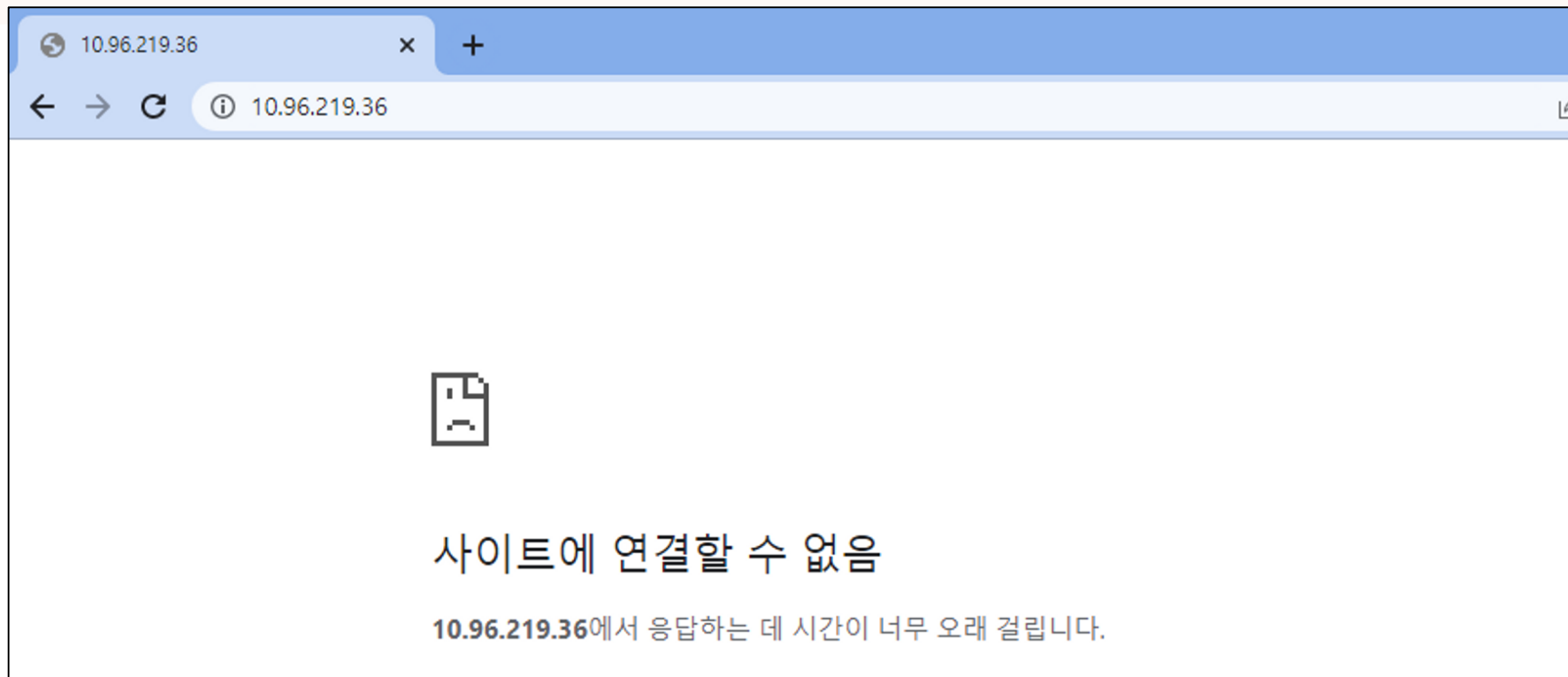
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/dashboard-metrics-scraper	1/1	1	1	52s
deployment.apps/kubernetes-dashboard	1/1	1	1	52s

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/dashboard-metrics-scraper-64bcc67c9c	1	1	1	52s
replicaset.apps/kubernetes-dashboard-66c887f759	1	1	1	52s

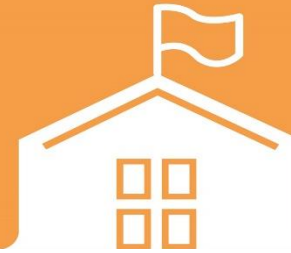
쿠버네티스 대시보드



- ✓ 클러스터 IP : 내부 네트워크 주소, 실제로 존재하지 않는 IP, kube-proxy에 의해 관리된다.



쿠버네티스 대시보드



- ✓ 클러스터 IP : 내부 네트워크 주소, 실제로 존재하지 않는 IP, kube-proxy에 의해 관리된다.

```
kubectl proxy
```

```
http://localhost:8001/api/v1/namespaces/kubernetes-  
dashboard/services/https:kubernetes-dashboard:/proxy/
```

쿠버네티스 대시보드



함께가요 미래로!
Enabling People

Kubernetes Dashboard

localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/#/login

Kubernetes Dashboard

☐ 토큰

모든 서비스 어카운트는 시크릿을 가지고 있고, 시크릿에는 대시보드에 로그인할 때 사용할 수 있는 유효한 베어러(Bearer) 토큰이 있습니다. 베어러(Bearer) 토큰을 설정 및 사용하는 방법은 [인증](#) 섹션에서 알 수 있습니다.

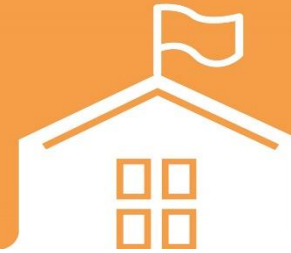
☒ Kubeconfig

클러스터에 접근을 설정하기 위해 생성한 kubeconfig 파일을 선택하세요. kubeconfig 파일을 설정 및 사용하기 위한 방법은 [멀티 클러스터에 접근 설정하기](#) 섹션에서 확인할 수 있습니다.

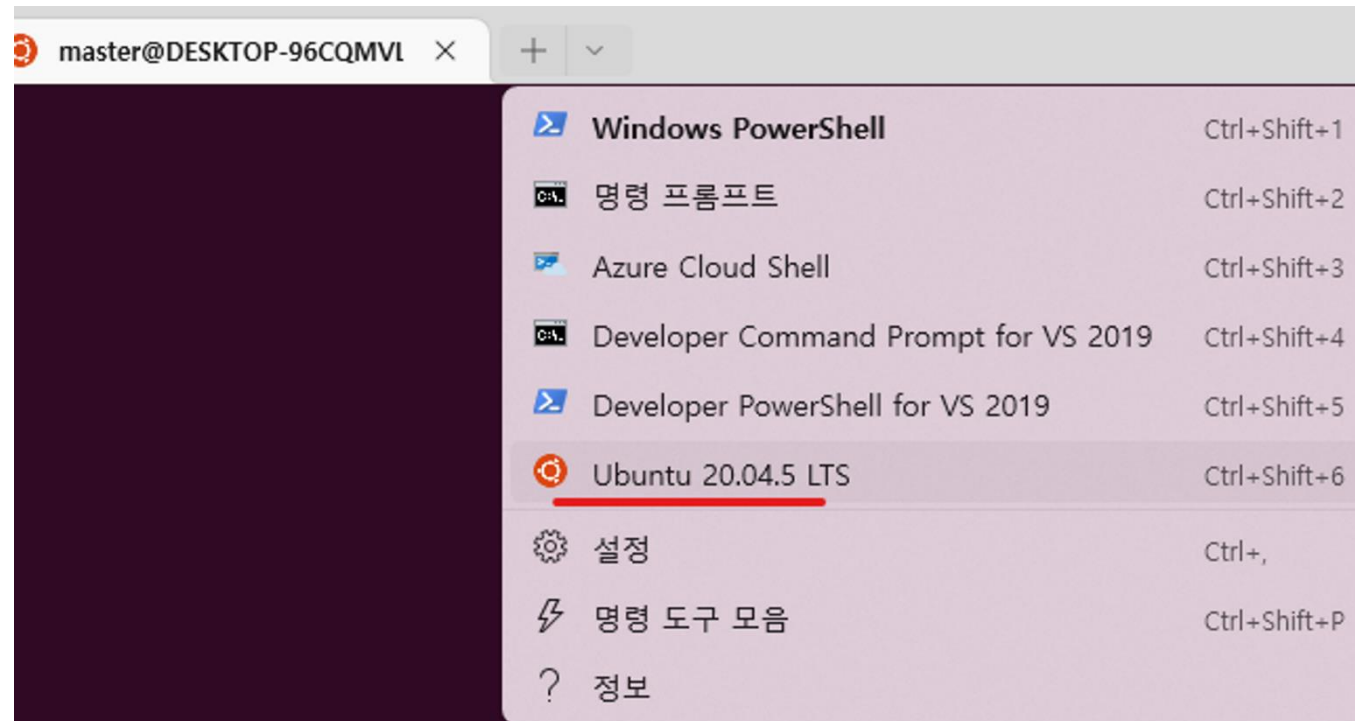
kubeconfig 파일 선택

로그인

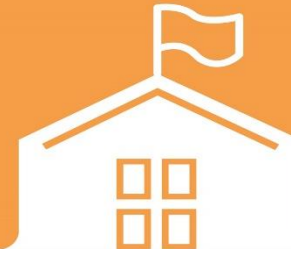
쿠버네티스 대시보드



- ✓ 자격 증명 추가: 새 Ubuntu 세션 열기(proxy 열어 둔 세션 그대로 두기)



쿠버네티스 대시보드



✓ 새 serviceAccount 생성

```
kubectl apply -f - <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
  name: admin-user
  namespace: kubernetes-dashboard
EOF
# 새 serviceAccount에 대해 ClusterRoleBinding 생성
kubectl apply -f - <<EOF
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: admin-user
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: cluster-admin
subjects:
- kind: ServiceAccount
  name: admin-user
  namespace: kubernetes-dashboard
EOF
```

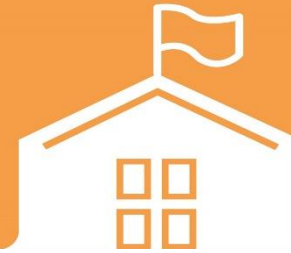


✓ 대시보드 토큰 발행

```
kubectl -n kubernetes-dashboard create token admin-user
```

```
master@DESKTOP-96CQMVL:~$ kubectl -n kubernetes-dashboard create token admin-user
eyJhbGciOiJSUzI1NiIsImtpZCI6IjkyZWwvXVZrdGE1NTctVlBQY21YbDNsTlg0WF9wbzBuQWNEZGllZlRlLnMRLmF1bHQuc3ZjLmNsdXN0ZXIubG9yYWwiXSwiZXhwIjoxNjc1MDEyMzk3LCJpYXQiOjE2NzUwMDg3OTUwLnN2Yy5jbHVzdGVyLmxvY2FsIiwia3ViZXJuZXRlcy5pbyI6eyJuYW1lc3BhY2UiOiJrdWJlcm5ldGVzLmF1bGUiOiJhZG1pb11c2VyIiwidWlkIjoiazDhkMGUzYzgtOTk0Yy00NmMyLTgzZWYtNDQ0ZTVmMTQxZmJKInRpdzZXJ2aWNlYWNib3VudDprdwJlcm5ldGVzLWRhc2hib2FyZDphZG1pb11c2VyIn0. jknUxbsETs6MvtXjQ2Cae1Lnyw5GT7IHMTWRtIFFj02jKhg0TVI1hWvzTCIZjBIHss-4av0W1Wq01mW0aVrdbQ2aaIRcb0q-SLnfbf3uSWDRDdC7jhVDxWBM0D80PUKGmU-bGTFmD0LE2USEP8aD0qw7MZCijTM6Fk_n_vPoekAJi3wiHQnMz-f-Rzdd2LEuloWQNNXgJ9W1wn4AszQw
```

쿠버네티스 대시보드



함께가요 미래로!
Enabling People

✓ 토큰 복사해서 입력하기

Kubernetes Dashboard

☒ 토큰

모든 서비스 어카운트는 시크릿을 가지고 있고, 시크릿에는 대시보드에 로그인할 때 사용할 수 있는 유효한 베어러(Bearer) 토큰이 있습니다. 베어러(Bearer) 토큰을 설정 및 사용하는 방법은 [인증](#) 섹션에서 알 수 있습니다.

☐ Kubeconfig

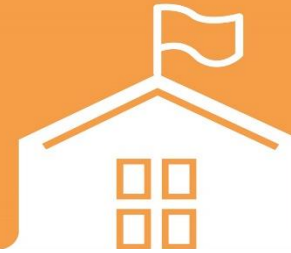
클러스터에 접근을 설정하기 위해 생성한 kubeconfig 파일을 선택하세요. kubeconfig 파일을 설정 및 사용하기 위한 방법은 [멀티 클러스터에 접근 설정하기](#) 섹션에서 확인할 수 있습니다.

토큰 입력 *

.....

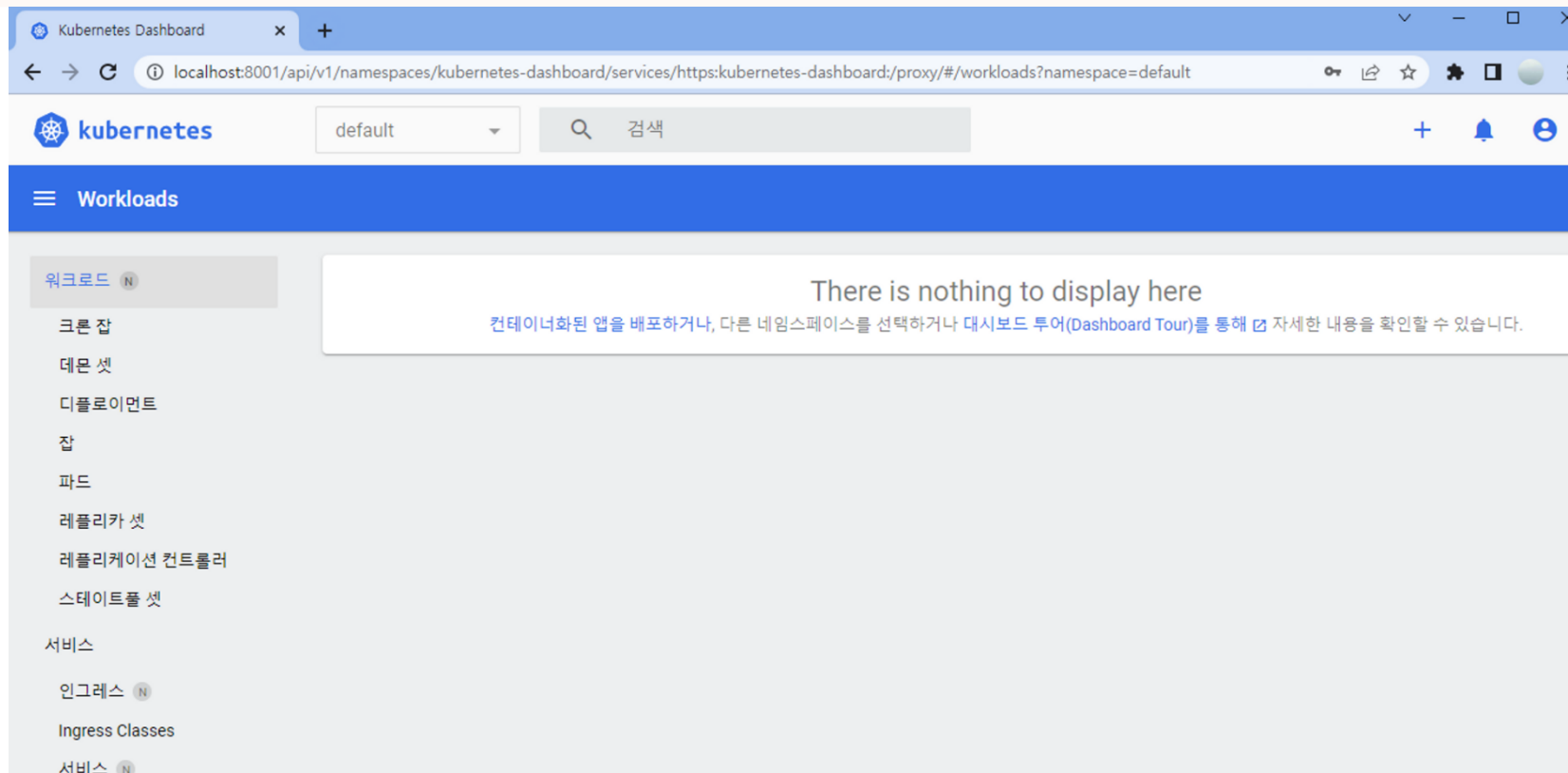
로그인

쿠버네티스 대시보드



함께가요 미래로!
Enabling People

✓ 대시보드 로그인 성공



쿠버네티스 대시보드



함께가요 미래로!
Enabling People

✓ 클러스터, 네임스페이스,
노드 등 대시보드로 확인 가능

워크로드

크론 잡
데몬 셋
디플로이먼트
잡
파드
레플리카 셋
레플리케이션 컨트롤러
스테이트풀 셋

서비스

인그레스
서비스

컨피그 및 스토리지

컨피그 맵
퍼시스턴트 볼륨 클레임
시크릿
스토리지 클래스

클러스터

클러스터 롤 바인딩
클러스터 롤
네임스페이스
네트워크 플리시
노드

워크로드 상태

데몬 셋

디플로이먼트

파드

레플리카 셋

데몬 셋

이름	네임스페이스	레이블	파드	생성 시간 ↑	이미지
✓ weave-net	kube-system	name: weave-net	3 / 3	14 hours ago	docker.io/weaveworks/weave-kube:2.8.1
✓ kube-proxy	kube-system	k8s-app: kube-proxy	3 / 3	14 hours ago	k8s.gcr.io/kube-proxy:v1.22.3

1 - 2 of 2 |< < > >|

디플로이먼트

이름	네임스페이스	레이블	파드	생성 시간 ↑	이미지
✓ dashboard-metrics-scraper	kubernetes-dashboard	k8s-app: dashboard-metrics-scraper	1 / 1	30 minutes ago	kubernetes/metrics-scraper:v1.0.6

클러스터 생성



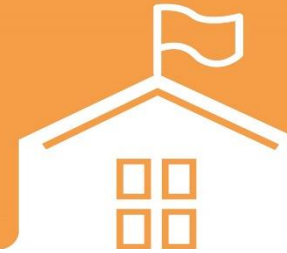
쿠버네티스
설치

쿠버네티스
대시보드

클러스터 생성

Jenkins Pod
생성

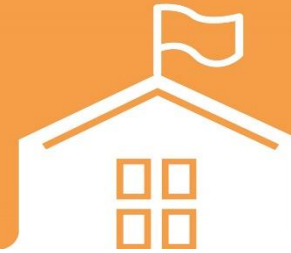
Jenkins 설치 해보기



✓ Jenkins namespace 생성

```
master@DESKTOP-96CQMV:~$ kubectl create namespace jenkins
namespace/jenkins created
master@DESKTOP-96CQMV:~$
```

Jenkins 설치 해보기

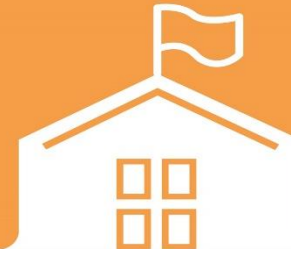


✓ Jenkins.yaml 파일 생성

```
apiVersion: v1
kind: Pod
metadata:
  name: jenkins
  namespace: jenkins
  labels:
    app: jenkins
spec:
  containers:
  - name: jenkins
    image: jenkins:2.60.3
    ports:
    - containerPort: 8080
    volumeMounts:
    - mountPath: /var/jenkins_home
      name: jenkins-home
  volumes:
  - name: jenkins-home
    hostPath:
      path: /jenkins
      type: DirectoryOrCreate
```

```
---
apiVersion: v1
kind: Service
metadata:
  name: jenkins
  namespace: jenkins
spec:
  type: NodePort
  selector:
    app: jenkins
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
    nodePort: 30080
```

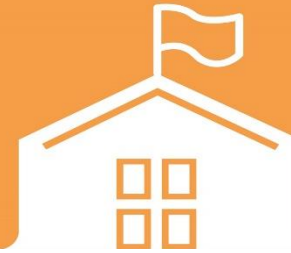
Jenkins 설치 해보기



- ✓ Jenkins.yaml 파일 이용하여 Jenkins 배포

```
kubectl apply -f jenkins.yaml
```

Jenkins 설치 해보기



✓ 포트포워딩

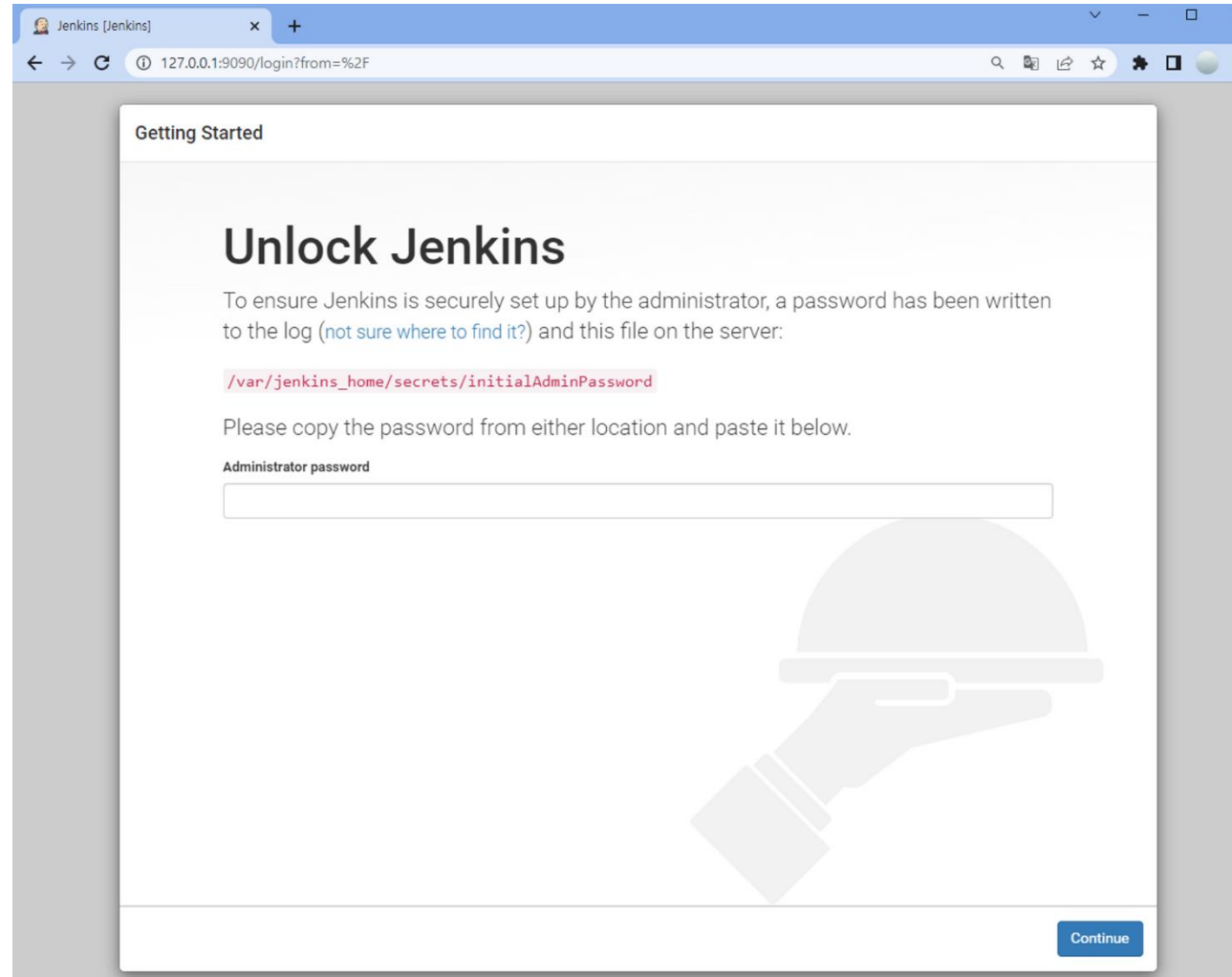
```
kubectl port-forward jenkins -n jenkins 9090:8080
```

```
master@DESKTOP-96CQMV:~$ kubectl port-forward jenkins -n jenkins 9090:8080
Forwarding from 127.0.0.1:9090 -> 8080
Forwarding from [::1]:9090 -> 8080
```

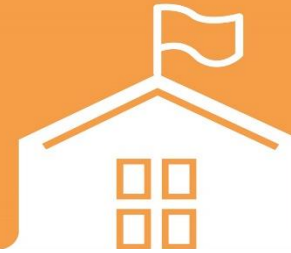
Jenkins 설치 해보기



함께가요 미래로!
Enabling People



쿠버네티스가 너무 무겁다면



함께가요 미래로!
Enabling People



minikube



K3S

PORTAINER

감사합니다.

