

쿠버네티스 들어가기

- ▶▶ 쿠버네티스 소개
- ▶▶ 우분투에 쿠버네티스 클러스터 구성
- ▶▶ 쿠버네티스에서 앱 실행해보기
- ▶▶ GKE를 활용한 쿠버네티스 사용
- ▶▶ AWS EKS를 활용한 쿠버네티스 사용
- ▶▶ 쿠버네티스 관리자 인증 시험(CKA) 소개

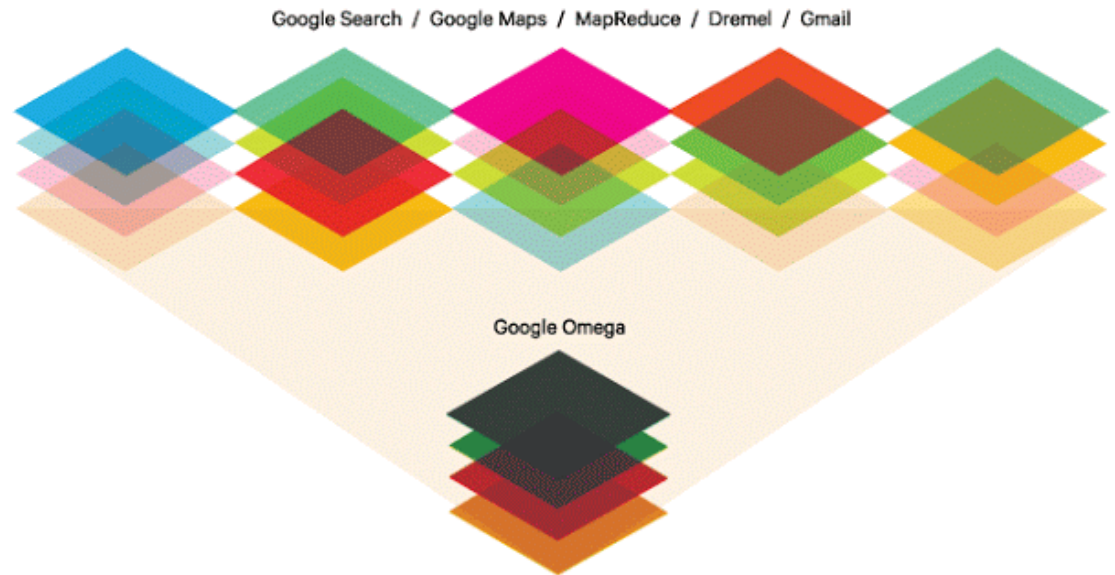
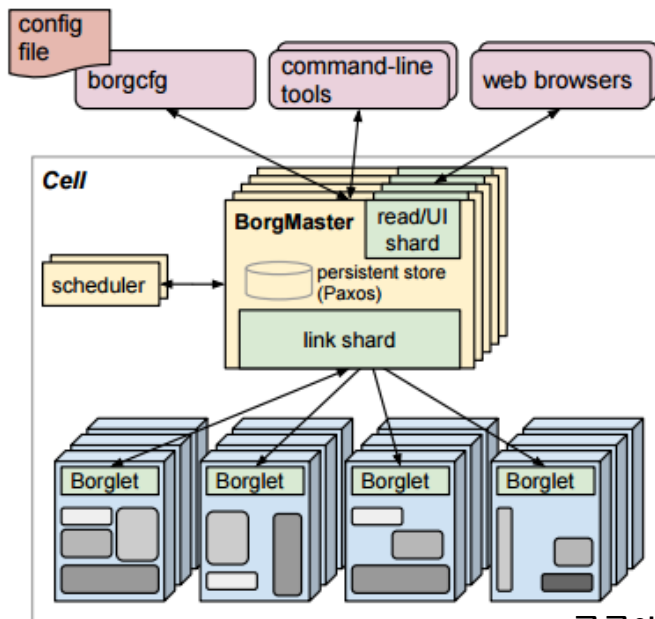


쿠버네티스 소개

쿠버네티스 소개

쿠버네티스 시작

- 오랜 세월 동안 구글은 보그(Borg)라는 내부 시스템을 개발
- 애플리케이션 개발자와 시스템 관리자가 수천 개의 애플리케이션과 서비스를 관리하는 데 도움
- 조직 규모가 클 때 엄청난 가치를 발휘
- 수십만 대의 시스템을 가동할 때 사용률이 조금만 향상돼도 수백만 달러의 비용 절감 효과
- 구글은 보그와 오메가를 15년 동안 비밀로 유지
- 2014년 구글 시스템을 통해 얻은 경험을 바탕으로 한 오픈소스 시스템인 '쿠버네티스'를 출시



구글에서 개발한 Borg와 Omega의 시스템

쿠버네티스 소개

인프라의 추상화

- 컨테이너 시스템에서 컨테이너 애플리케이션을 쉽게 배포, 관리하도록 돕는 소프트웨어 시스템
- 기본 인프라를 추상화해 개발 및 운영 팀의 개발, 배포, 관리를 단순화
- 모든 노드가 하나의 거대한 컴퓨터인 것처럼 수천개의 컴퓨터 노드에서 소프트웨어 애플리케이션을 실행



쿠버네티스 소개

쿠버네티스의 장점

- 애플리케이션 배포 단순화
 - 특정 베어메탈을 필요로 하는 경우(예: SSD/HDD)
- 하드웨어 활용도 극대화
 - 클러스터의 주변에 자유롭게 이동하여 실행중인 다양한 애플리케이션 구성 요소를 클러스터 노드의 가용 리소스에 최대한 맞춰 서로 섞고 매치
 - 노드의 하드웨어 리소스를 최상으로 활용
- 상태 확인 및 자가 치유
 - 애플리케이션 구성 요소와 실행되는 노드를 모니터링 하고 노드 장애 발생시 다른 노드로 일정을 자동으로 재조정
 - 운영자는 정규 근무 시간에만 장애가 발생한 노드를 처리(일이 편해진다!)
- 오토스케일링
 - 개별 애플리케이션의 부하를 지속적으로 모니터링할 필요 없이
 - 자동으로 리소스를 모니터링하고 각 애플리케이션에서 실행되는 인스턴스 수를 계속 조정하도록 지시 가능
- 애플리케이션 개발 단순화
 - 버그 발견 및 수정 (완전히 개발환경과 같은 환경을 제공하기 때문)
 - 새로운 버전 출시 시 자동으로 테스트, 이상 발견 시 롤 아웃

쿠버네티스 소개

개발자 돕기: 핵심 애플리케이션 기능에 집중

- 애플리케이션 개발자가 특정 인프라 관련 서비스를 애플리케이션에 구현하지 않아도 됨
- 쿠버네티스에 의존해 서비스 제공
 - 서비스 검색, 확장, 로드 밸런싱, 자가 치유, 리더 선출 등
- 애플리케이션 개발자는 애플리케이션의 실제 기능을 구현하는 데 주력
- 인프라와 인프라를 통합하는 방법을 파악하는데 시간을 낭비할 필요 없음

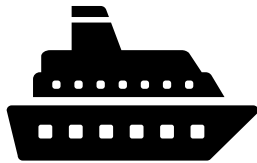
운영 팀 돕기: 이 효과적으로 리소스를 활용

- 실행을 유지하고 서로 통신할 수 있도록 컴포넌트에 정보를 제공
- 애플리케이션이 어떤 노드에서 실행되는 상관 없음(신경 쓰지 않아도 됨)
- 언제든지 애플리케이션을 재배포 가능
- 애플리케이션을 혼합하고 매칭시킴으로써 리소스를 매칭

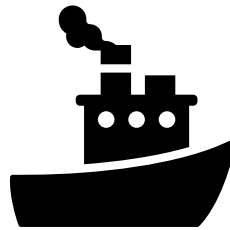
쿠버네티스 소개

쿠버네티스 클러스터 아키텍처

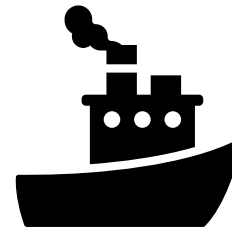
- 쿠버네티스의 클러스터는 하드웨어 수준에서 많은 노드로 구성되며 두 가지 유형 나뉨
 - 마스터 노드: 전체 쿠버네티스 시스템을 관리하고 통제하는 쿠버네티스 컨트롤 플레인을 관장
 - 워커 노드: 실제 배포하고자 하는 애플리케이션의 실행을 담당



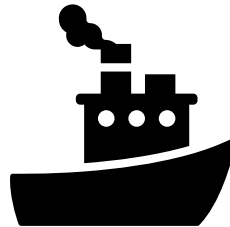
Master Node



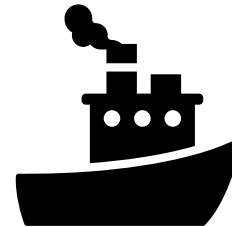
Worker Node1



Worker Node3



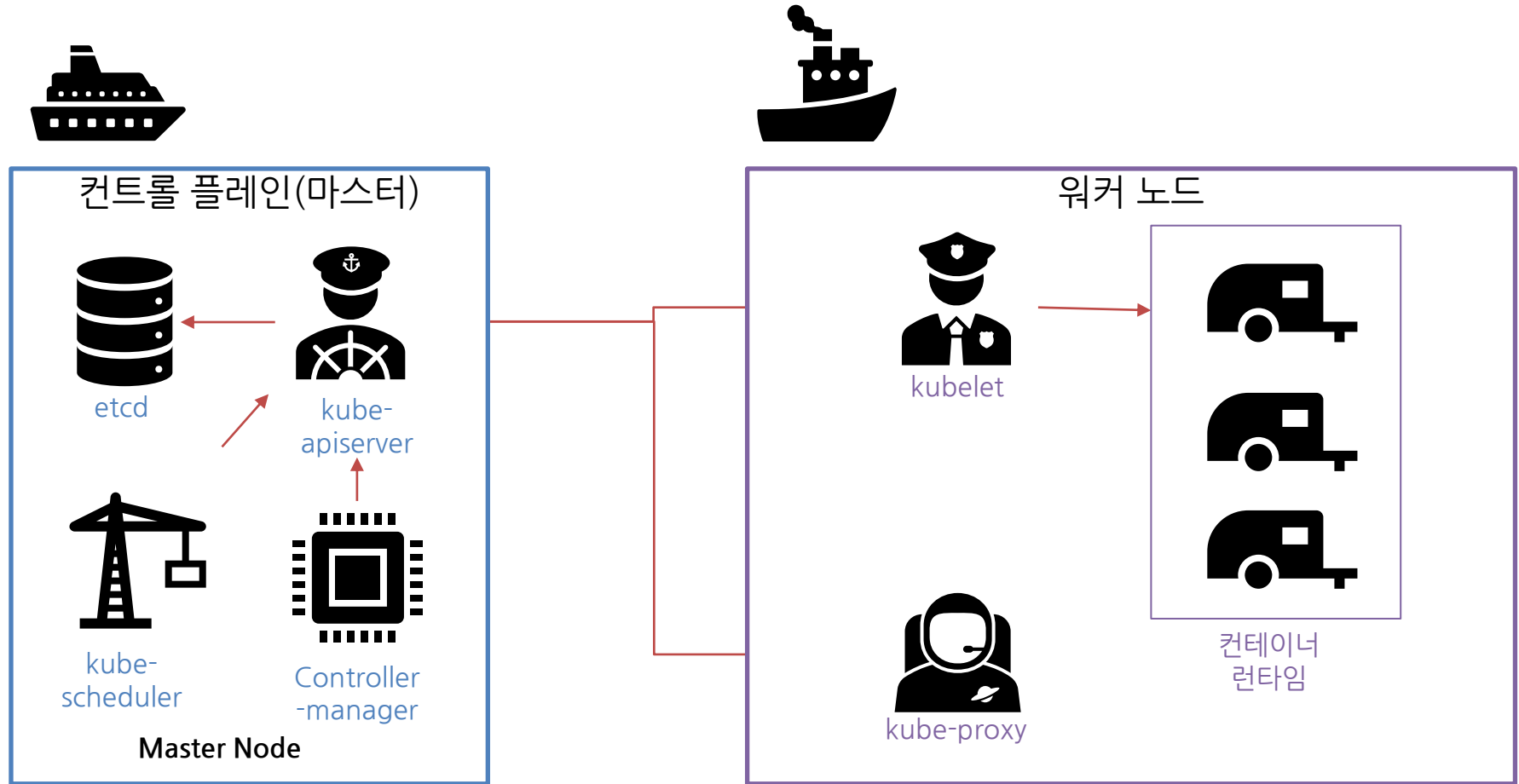
Worker Node2



Worker Node4

쿠버네티스 소개

쿠버네티스 클러스터 아키텍처



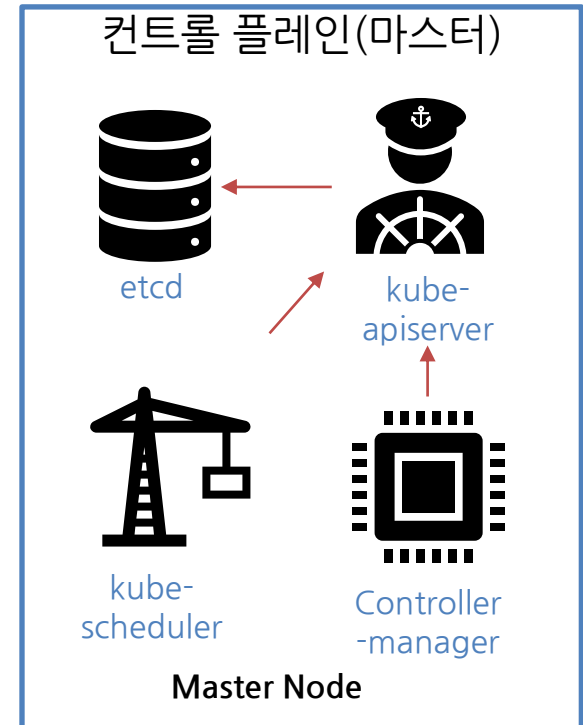
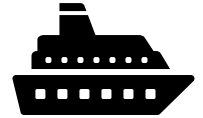
쿠버네티스 소개

컨트롤 플레인

- 컨트롤 플레인에서는 클러스터를 관리하는 기능
- 단일 마스터 노드에서 실행하거나 여러 노드로 분할되고 복제돼고가용성을 보장
- 클러스터의 상태를 유지하고 제어하지만 애플리케이션을 실행하지 않음

컨트롤 플레인의 구성요소

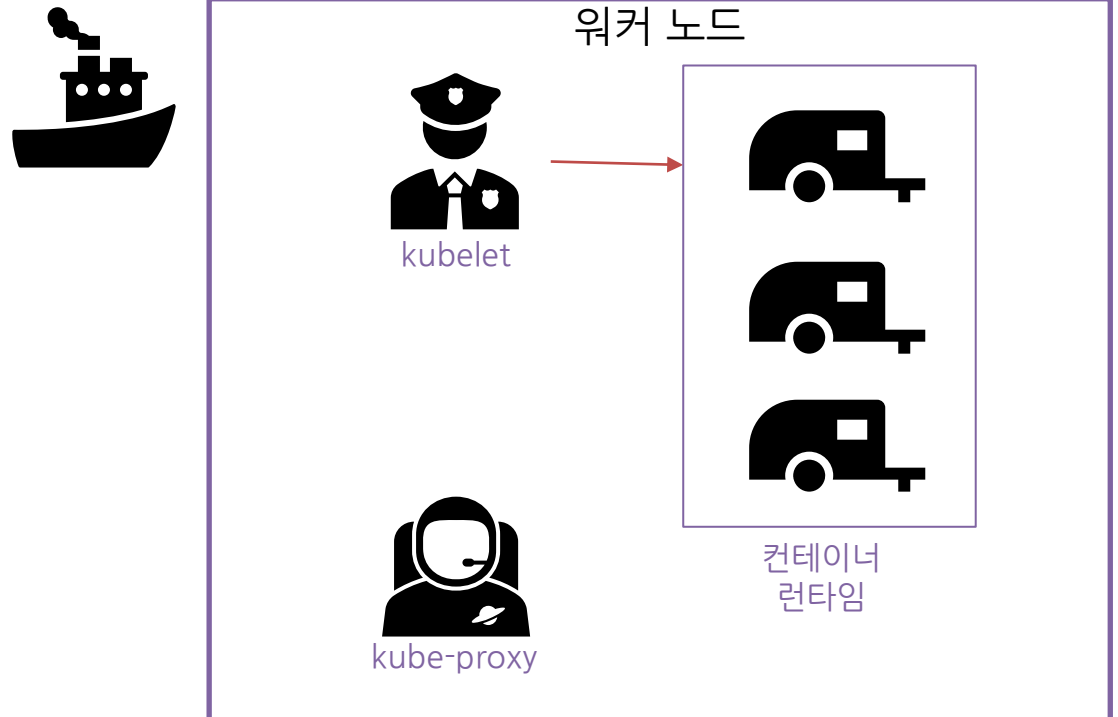
- 쿠버네티스 API 서버: 사용자, 컨트롤 플레인과 통신
- 스케줄러: 애플리케이션 예약(애플리케이션의 배포 가능한 각 구성 요소에 워커 노드를 할당)
- 컨트롤 매니저: 구성 요소 복제, 워커 노드 추적, 노드 장애 처리 등 클러스터 수준 기능을 실행
- 데이터 스토리지: etcd는 클러스터 구성을 지속적으로 저장하는 안정적인 분산



쿠버네티스 소개

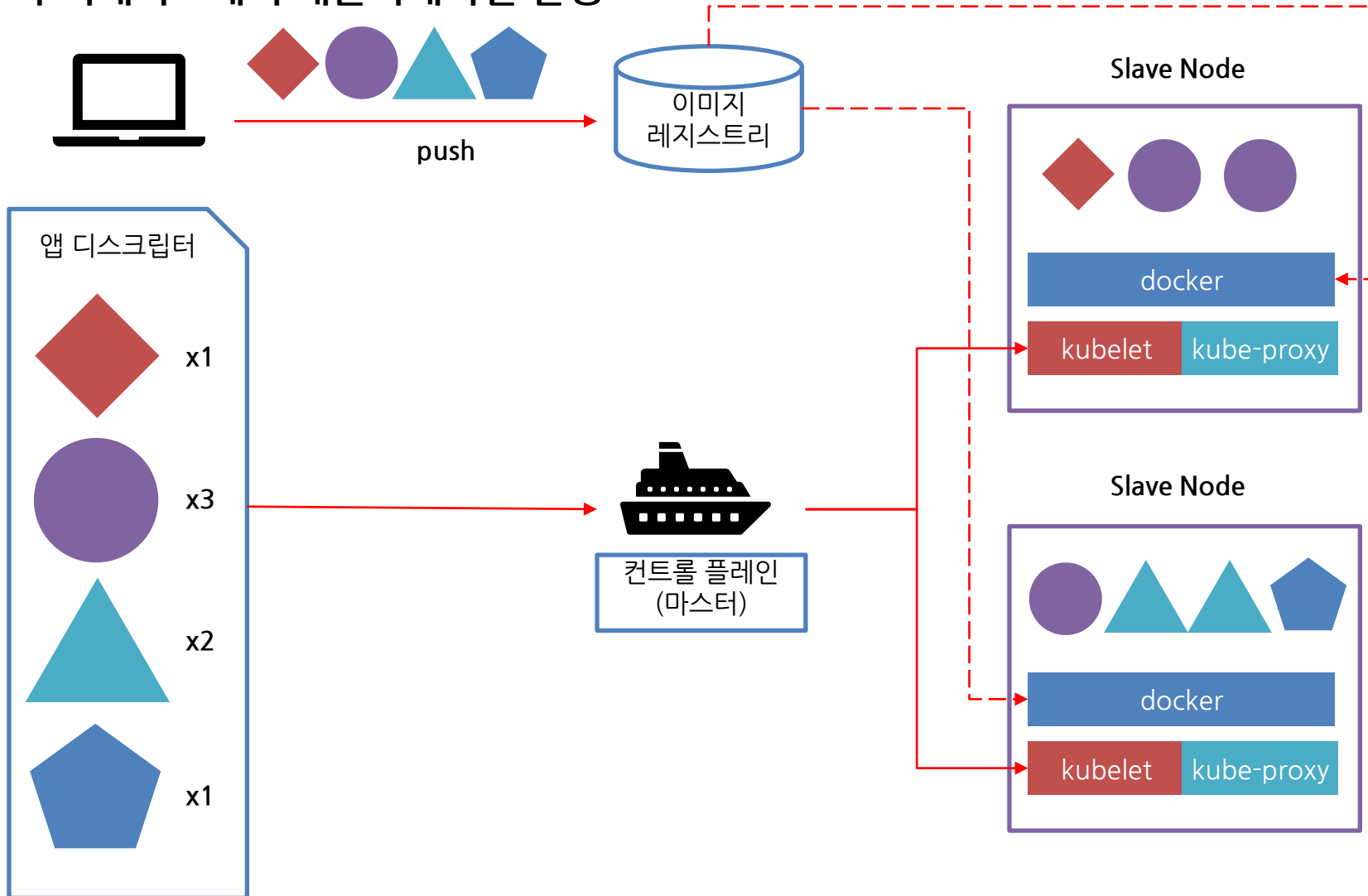
노드

- 워커 노드는 컨테이너화된 애플리케이션을 실행하는 시스템
- 애플리케이션에 서비스를 실행, 모니터링, 제공하는 작업은 다음과 같은 구성요소로 수행
- 컨테이너 런타임: 컨테이너를 실행하는 도커
- Kubelet: API 서버와 통신하고 노드에서 컨테이너를 관리
- 쿠버네티스 서비스(Kubernetes Service), 프록시(kube-proxy): 애플리케이션 간에 네트워크 트래픽을 분산 및 연결



쿠버네티스 소개

쿠버네티스에서 애플리케이션 실행



쿠버네티스 소개

▶ 쿠버네티스에서 애플리케이션 실행

● 쿠버네티스에서 애플리케이션을 실행 전

- 하나 이상의 컨테이너 이미지들을 패키지로 레지스트리에 푸시
- 쿠버네티스 API 서버에 애플리케이션의 디스크립션을 게시

● 디스크립션(YAML로 작성)

- 컨테이너 이미지 또는 애플리케이션 컴포넌트가 들어 있는 이미지가 존재
- 컴포넌트 간 관련성 및 노드 배치(동일 노드 또는 다른 노드) 정보 포함
- 각 컴포넌트의 실행 복제본 수를 지정
- 내부 클라이언트나 외부 클라이언트에 서비스를 제공하는 컴포넌트
- 단일 IP 주소로 노출해 다른 컴포넌트에서 검색

우분투에 쿠버네티스 클러스터 구성

우분투에 쿠버네티스 클러스터 구성

쿠버네티스 설치 필요 사항

- Master 우분투 : Kubernetes 의 마스터 노드가 설정될 호스트
- Work 노드(옵션) : 필수 사항은 아니지만, 클러스터에 Work 노드 추가 학습
- 버추얼박스에서 각 노드에서 복제하면서 반드시 변경해야 하는 설정
 - 호스트 이름 : /etc/hostname
 - 네트워크 인터페이스 변경
 - NAT 네트워크 설정(NAT랑 다름)
 - (호스트 이름 변경하려면 반드시 리붓)

Oracle VM VirtualBox 관리자

파일(F) 머신(M) 스냅샷(S) 도움말(H)



도구

[Master]ubuntu-18.04.2-desktop-amd64 (도커 설치 후)
전원 켜짐

[Slave]ubuntu-18.04.2-desktop-amd64 (도커 설치 후)
전원 켜짐

[Slave2]ubuntu-18.04.2-desktop-amd64 (도커 설치 후)
전원 켜짐

가상 머신 선택

새 머신의 이름과 경로

새 가상 머신의 이름 및 폴더(선택 사항)를 입력하십시오. 새 머신은 머신 [Slave]ubuntu-18.04.2-desktop-amd64-1의 복제본이 될 것입니다.

이름: [Slave2]ubuntu-18.04.2-desktop-amd64

경로: C:\Users\Wisc03\VirtualBox VMs

MAC 주소 정책(P): 모든 네트워크 어댑터의 새 MAC 주소 생성

추가 옵션: ☐ 디스크 이름 유지하기(D) ☐ 하드웨어 UUID 유지하기(H)

전문가 모드(E) 다음(N) 취소

우분투에 쿠버네티스 클러스터 구성

▶ 쿠버네티스 우분투에 설치

- 다음 내용을 install.sh 파일에 작성하고 chmod로 권한을 주고 실행
- 쿠버네티스 설치 사이트(아래 스크립트 내용이 있음):
 - <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>
- 신뢰할 수 있는 APT키를 추가
 - apt-get update && apt-get install -y apt-transport-https curl
 - curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -
- 그리고 아래의 명령어로 Repository를 추가하고, Kubernetes 를 설치
 - cat <<EOF >/etc/apt/sources.list.d/kubernetes.list
 - deb https://apt.kubernetes.io/ kubernetes-xenial main
 - EOF
 - apt-get update
 - apt-get install -y kubelet kubeadm kubectl
 - apt-mark hold kubelet kubeadm kubectl

우분투에 쿠버네티스 클러스터 구성

kubernetes를 관리하는 명령어

- **kubeadm**

- 클러스터를 부트스트랩하는 명령

- **kubelet**

- 클러스터의 모든 시스템에서 실행되는 구성 요소로, 창 및 컨테이너 시작과 같은 작업을 수행

- **kubectl**

- 커맨드 라인 util은 당신의 클러스터와 대화

우분투에 쿠버네티스 클러스터 구성

▶ Master 노드 초기화 (마스터 노드에서만 할 것!)

● Master 노드를 초기화를 가장 먼저 수행 (사용할 포트 네트워크 대역을 설정)

➤ `sudo kubeadm init`

● 스왑 에러 발생 시 스왑 기능 제거

➤ `sudo swapoff -a` // 현재 커널에서 스왑 기능 끄기

➤ `sudo sed -i 's/^swp(.*)$/#swp1/g' /etc/fstab` // 리붓 후에도 스왑 기능 유지

● Kubernetes에서 스왑을 비활성화하는 이유

➤ Kubernetes 1.8 이후, 노드에서 스왑을 비활성화해야 함(또는 `--fail-swap-on`을 `false`로 설정)

➤ kubernetes의 아이디어는 인스턴스를 최대한 100%에 가깝게 성능을 발휘하는 것

➤ 모든 배포는 CPU/메모리 제한을 고정하는 것이 필요

➤ 따라서 스케줄러가 포드를 머신에 보내면 스왑을 사용하지 않는 것이 필요

➤ 스왑 발생시 속도가 느려지는 이슈 발생

➤ 성능을 위한 것

➤ 참고문헌: <https://serverfault.com/questions/881517/why-disable-swap-on-kubernetes>

우분투에 쿠버네티스 클러스터 구성

Master 노드 초기화 성공 시 나오는 메시지 (마스터 노드에서만 할 것!)

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

```
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

<https://kubernetes.io/docs/concepts/cluster-administration/addons/>

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 10.0.2.15:6443 --token dwaoa1.4nf7b81nsfnkxctw ₩  
--discovery-token-ca-cert-hash  
sha256:b16367e80df58c3dbacfc3961126ae82d68519368d345fdb228addf04cb4ea2f
```

우분투에 쿠버네티스 클러스터 구성

▶ 클러스터를 사용 초기 세팅(마스터 노드에서만 할 것!)

● 다음을 일반 사용자 계정으로 실행 (콘솔에 출력된 메시지를 복붙)

- `mkdir -p $HOME/.kube`
- `sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config`
- `sudo chown $(id -u):$(id -g) $HOME/.kube/config`

● Pod Network 추가

- `kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"`
- 이것을 잘해야 노드 추가 명령어가 잘 실행됩니다!

우분투에 쿠버네티스 클러스터 구성

▶▶ 슬레이브 노드 추가 (슬레이브 노드에서만 할 것!)

- 앞서 설치한 대로 쿠버네티스 설치
- init 명령어 전까지만 수행(**init 명령어 실행하지 마세요**)
- 이후 각 노드에서 관리자 권한으로 워커 노드를 참가 시킴 (콘솔에 출력된 메시지를 복붙)
 - `sudo kubeadm join 10.0.2.15:6443 --token xwvbf.5xc67j8qc6ohl2it ₩`
 - `--discovery-token-ca-cert-hash`
`sha256:e19e9263aeb2340a602c2057966b71551e01a5e287d3f23b05073c7b248932e1`

우분투에 쿠버네티스 클러스터 구성

▶▶ 마지막으로 연결된 노드들의 상태 확인

- `kubectl get nodes`

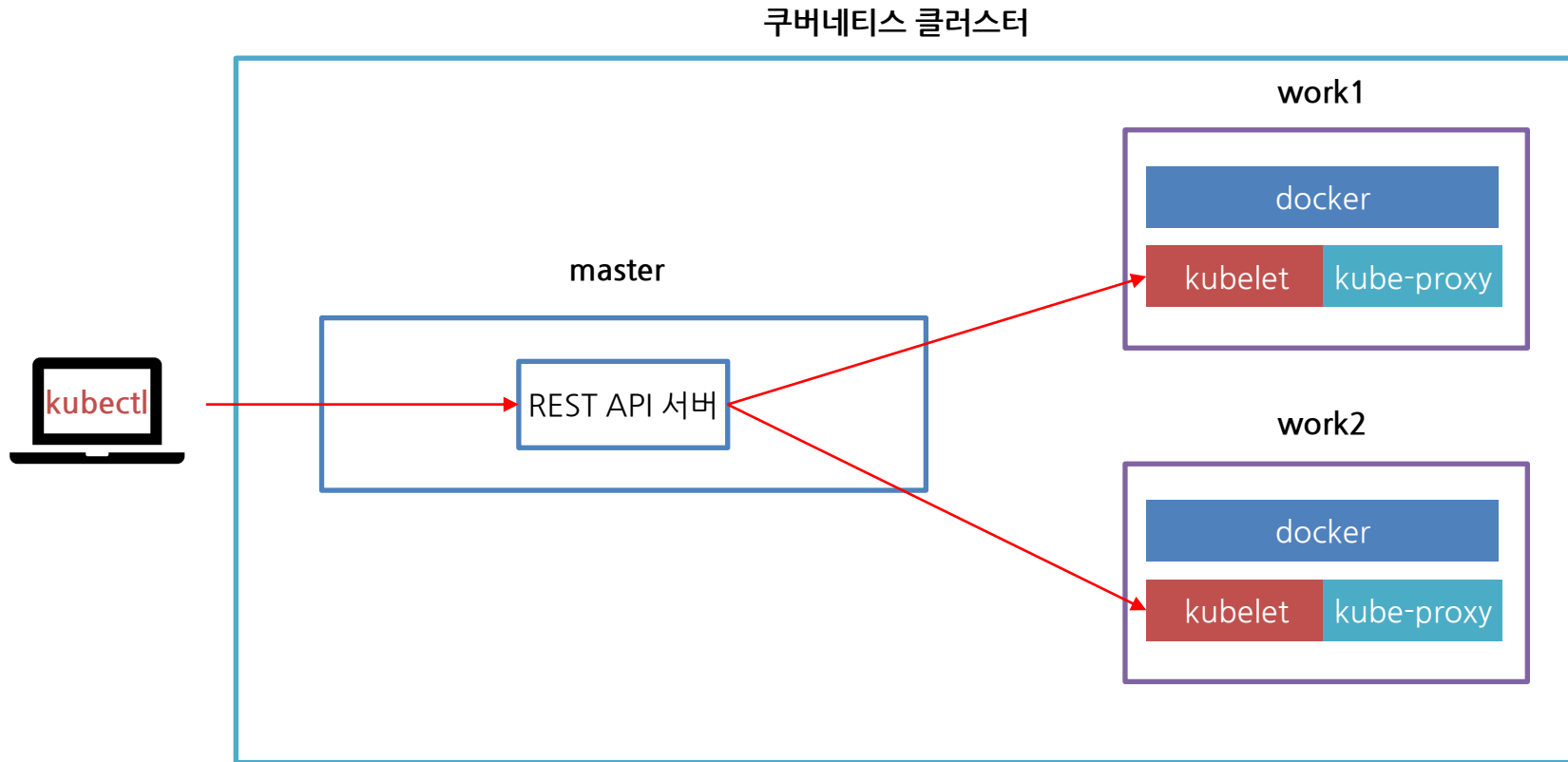
- STATUS 값이 NotReady 상태인 경우, Pod Network가 아직 deploy 되기 전일 수 있음
- 장시간 기다려도 변경되지 않으면 앞에서 설정한 “Pod Network 추가” 과정이 잘못 됐을 수 있음

```
$ kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
master-virtualbox	Ready	master	2m41s	v1.14.3
slave1-virtualbox	Ready	<none>	2m6s	v1.14.3
slave2-virtualbox	Ready	<none>	2m2s	v1.14.3

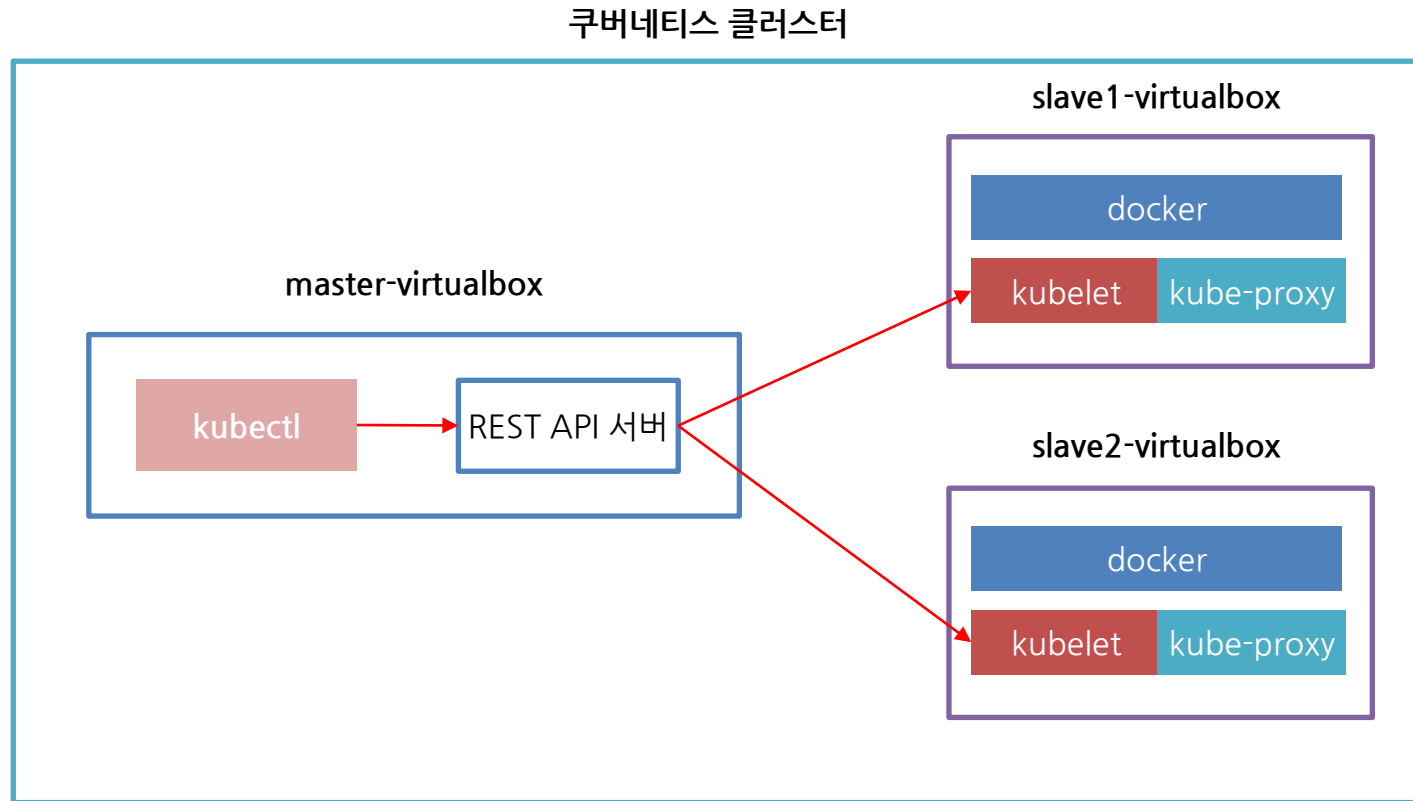
우분투에 쿠버네티스 클러스터 구성

일반적인 사용자와 마스터 노드, 워커 노드 연결 관계



우분투에 쿠버네티스 클러스터 구성

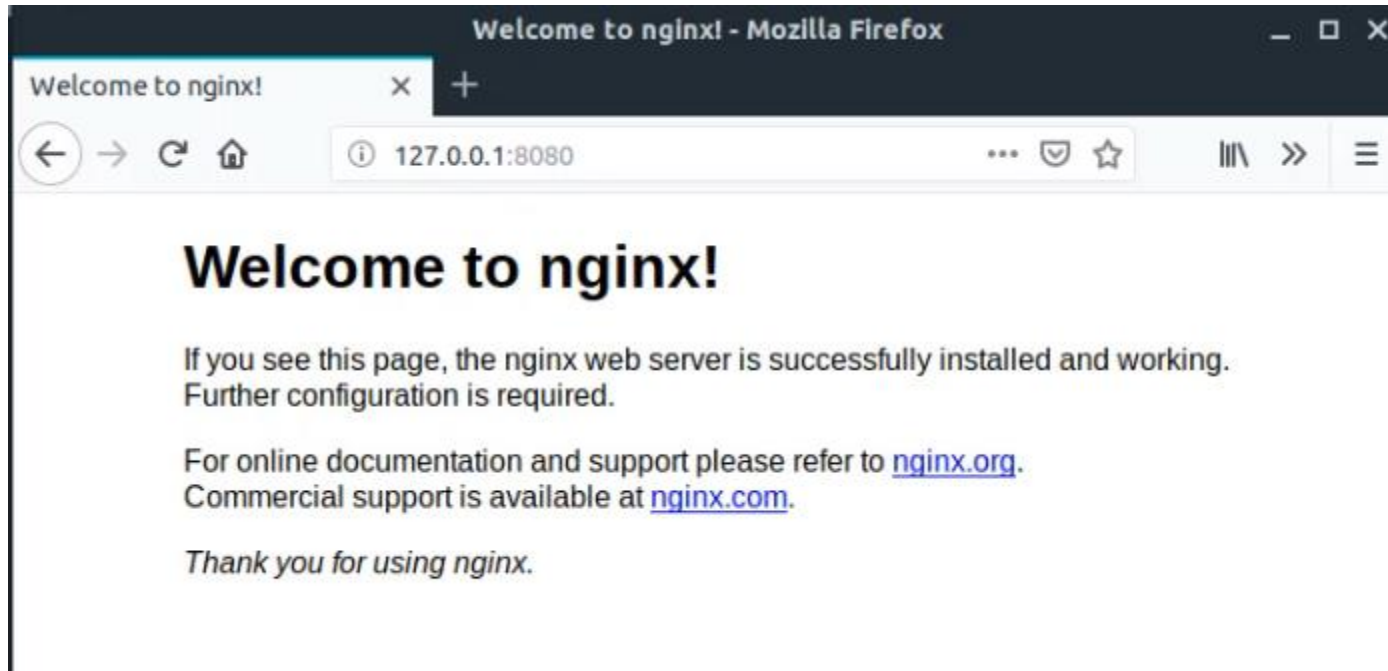
우리가 설정한 환경에서 마스터 노드와 슬레이브 노드 연결 관계



우분투에 쿠버네티스 클러스터 구성

간단한 애플리케이션 실행 및 확인

```
$ kubectl create deploy --image=nginx  
$ kubectl get pod  
$ kubectl port-forward nginx-XXXXXXXXXX-XXXXX 8080:80
```



우분투에 쿠버네티스 클러스터 구성

▶▶ 수강생 실습하기

- 가상머신을 하나만 놔두고 모든 노드를 삭제합니다.
- 모든 상태를 쿠버네티스를 설치한 뒤의 스냅샷으로 되돌립니다.
- Master, Worker1, Worker2의 가상머신을 각각 복제하여 생성하고 쿠버네티스를 설치하여 클러스터를 구성하십시오.
- Kubeadm init 또는 join 명령을 실행할 때 중복된 실행으로 문제가 생기는 경우 kubeadm reset 명령을 통해서 초기화한다.

GKE를 활용한 쿠버네티스 사용

GKE를 활용한 쿠버네티스 사용

Google cloud 의 관리형 kubernetes 서비스인 Google Kubernetes Engine(GKE)

- GKE 는 Kubernetes 를 쉽게 사용자가 활용할 수 있도록 관리형으로 제공
- 규모에 맞춘 컨테이너식 애플리케이션 관리
- 다양한 애플리케이션 배포
- 고가용성을 통한 원활한 운영
- 수요에 맞게 간편하게 확장
- Google 네트워크에서의 안전한 실행
- 온프레미스 및 클라우드 간의 자유로운 이동

Google Cloud Google을 사용해야 하는 이유 솔루션 제품 가격 책정 시작하기

컴퓨팅 제품

영업팀

KUBERNETES ENGINE

Kubernetes 클러스터를 안정적이고 효율적이며 안전하게 실행할 수 있는 방법



KUBERNETES ENGINE 문서 보기


내 콘솔 보기



GKE를 활용한 쿠버네티스 사용

▶ GCP 회원가입하기

- 회원가입 후 1년 \$300 무료 평가판 사용 가능
- 구글 ID 필요
- 회원가입 시 신용카드 필요
- <https://cloud.google.com/>

 Google Cloud Google을 사용해야 하는 이유 솔루션 제품 가격 책정 시작하기

이곳에서 펼치는 새로운 도약

Google Cloud는 가장 까다로운 비즈니스 과제를 해결할 수 있는 GCP 및 G Suite가 포함된 솔루션과 제품 모음입니다.

Google Cloud Platform

안전한 저장소, 강력한 컴퓨팅, 통합 데이터 분석 제품으로 비즈니스의 성장을 도모하세요.

[무료로 GCP 사용해 보기](#)

[영업팀에 문의하기](#)

모든 Cloud Platform 제품에 액세스

Firebase, Google Maps API 등을 포함해 앱, 웹사이트, 서비스를 구축하고 실행하는 데 필요한 모든 기능을 이용할 수 있습니다.

\$300의 무료 크레딧

가입하여 Google Cloud Platform에서 12개월간 사용할 수 있는 \$300 크레딧을 받아 보세요.

무료 평가판 종료 후 자동 청구되지 않음

신용카드를 요청하는 이유는 자동 가입을 방지하기 위해서입니다. 사용자가 유료 계정으로 직접 업그레이드하지 않는 한 요금이 청구되지 않습니다.

GKE를 활용한 쿠버네티스 사용

▶ GCP 가입 및 가입오류 해결

- 1) 회원 가입에서 다음이 눌리지 않는 경우 해결 방법

- <https://private-space.tistory.com/41>에서 자세히 설명
- 요약: <https://cloud.google.com/gcp/getting-started/?hl=ko>를 접속해서 “콘솔”로 접속하면 가입 오류 없이 회원 가입 가능

- 2) 회원가입 시 카드 정보 오류가 발생하는 경우

- 회원가입 시 카드 정보 오류가 발생하는 경우 모바일로 접속해서 등록하면 회원 가입 가능한 경우가 있음

- 3) 사용할 수 없는 결제 정보인 경우

- 다른 ID를 새로 생성하면 해결되는 경우가 있음

GKE를 활용한 쿠버네티스 사용

회원가입하기

- 신용카드 정보를 적고 무료 평가판 시작
- <주의> 모든 실습 후 클러스터를 반드시 삭제

Google Cloud Platform 무료로 사용해 보기

1/2단계

국가

서비스 약관

- ☐ [Google Cloud Platform 서비스 약관](#)과 [관련 서비스 및 API의 서비스 약관](#)에 동의합니다. 또한 [Google Cloud Platform 무료 평가판 서비스 약관](#)을 읽었으며 이에 동의합니다.

계속 진행하려면 체크박스를 선택하세요.

이메일 업데이트

- ☐ Google Cloud 및 Google Cloud 파트너가 보내는 뉴스, 제품 업데이트, 특별 이벤트에 대한 정기적인 이메일을 수신하겠습니다.

동의 및 계속하기

결제 옵션



자동 결제

비용이 발생한 후에만 서비스를 결제합니다. 청구 기준액에 도달하거나 지난 자동 결제일로부터 30일이 경과하면 둘 중 더 이른 날짜에 비용이 자동 청구됩니다.

결제 수단 ⓘ

카드 번호

#

MM / YY

CVC

카드 번호를 입력하세요.

카드 소유자 이름

카드 소유자 이름을 입력하세요.

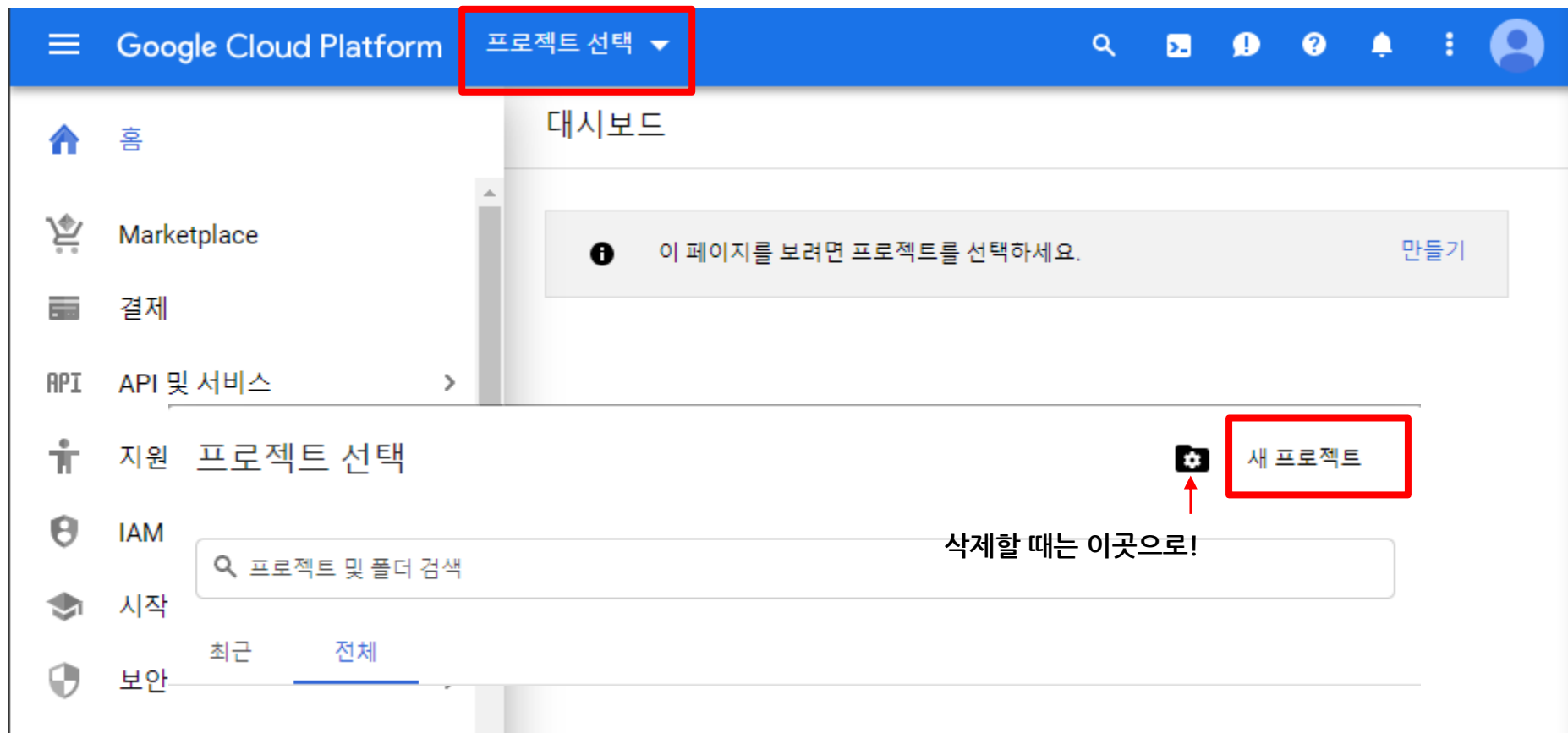
☒ 신용카드 또는 체크카드 주소가 위의 주소와 동일합니다.

무료 평가판 시작하기

GKE를 활용한 쿠버네티스 사용

프로젝트 생성

- 프로젝트 선택을 클릭하고 새 프로젝트 생성




GKE를 활용한 쿠버네티스 사용

▶ 프로젝트 생성

- 프로젝트 이름을 세팅하고 생성
- 홈-컴퓨팅-Kubernetes Engine 클릭

☰ Google Cloud Platform

새 프로젝트

 projects 할당량이 11개 남았습니다. 할당량 증가를 요청하거나 프로젝트를 삭제하세요. [자세히 알아보기](#)
[MANAGE QUOTAS](#)

프로젝트 이름 *

프로젝트 ID: gketest-244306입니다. 나중에 변경할 수 없습니다. [수정](#)


위치 *
 [찾아보기](#)


상위 조직 또는 폴더


☰ Google Cloud Platform


홈


컴퓨팅

 App Engine >

 Compute Engine >

 **Kubernetes Engine** >

 Cloud Functions

 Cloud Run

GKE를 활용한 쿠버네티스 사용

Kubernetes Engine API를 사용 설정(자동)

Google Cloud Platform gketest

Kubernetes Engine 클러스터

클러스터

작업 부하

서비스

애플리케이션

구성

저장소

Kubernetes Engine API를 사용 설정하는 중이며 1분 이상 걸릴 수 있습니다. 자세히 알아보기

Kubernetes Engine
Kubernetes 클러스터

컨테이너는 애플리케이션이 쉽게 배포되어 독립된 자체 환경에서 실행될 수 있도록 애플리케이션을 패키징합니다. 컨테이너는 VM 생성 및 유지관리를 자동화하는 클러스터에서 관리합니다. 자세히 알아보기

클러스터 만들기 컨테이너 배포 빠른 시작 사용

GKE를 활용한 쿠버네티스 사용

Kubernetes Engine 클러스터 만들기

- 클러스터를 생성하는 다양한 방법이 존재 (표준 클러스터 선택)
- 템플릿에 필요한 사항이 미리 채워져 있으나 확인 후 진행(도쿄 리전:asia-northeast1-a)
- 노드 수 확인(마스터 노드 포함)

클러스터 템플릿

사전 구성된 설정이 포함된 템플릿을 선택하거나 요구사항에 맞게 템플릿을 맞춤설정하세요.

☐ 기존 클러스터 복제

기존 클러스터 중 하나를 선택하여 필드를 채웁니다.

☒ 표준 클러스터

지속적 통합, 웹 제공, 백엔드용입니다. 추가 맞춤설정이 필요하거나 어떤 템플릿을 선택할지 확실하지 않은 경우에 선택하면 가장 적합합니다.

☐ 첫 번째 클러스터

Kubernetes Engine으로 실험하고 첫 번째 애플리케이션을 배포하세요. 저렴한 비용으로 시작할 수 있습니다.

'표준 클러스터' 템플릿

지속적 통합, 웹 제공, 백엔드용입니다. 추가 맞춤설정이 필요하거나 어떤 템플릿을 선택할지 확실하지 않은 경우에 선택하면 가장 적합합니다.

1 클러스터가 생성된 후에는 일부 필드를 변경할 수 없습니다. 자세히 알아보려면 도움말 아이콘 위에 마우스를 가져가세요.

닫기

이름 ?

standard-cluster-1

위치 유형 ?

☒ 영역

☐ 지역

영역 ?

asia-northeast1-a

마스터 버전

1.12.8-gke.6 (기본)

노드 풀

노드 풀은 클러스터에서 Kubernetes를 실행하는 별도의 인스턴스 그룹입니다. 여러 머신 유형의 노드 풀을 추가하거나 가용성을 높이기 위해 여러 영역에 노드 풀을 추가할 수 있습니다. 노드 풀을 추가하려면 '수정'을 클릭하세요. [자세히 알아보기](#)

default-pool

노드 수

3

머신 유형 ?

맞춤설정을 통해 코어, 메모리, CPU를 선택합니다.

...

자동 업그레이드: 사용 중

옵션 더보기

+ 노드 풀 추가

≈ 가용성, 네트워킹, 보안, 추가 기능

GKE를 활용한 쿠버네티스 사용

Kubernetes Engine 클러스터 만들기

- 생성 클릭 후 생성까지 약 10분 소요, 생성이 완료되면 연결 시작

Google Cloud Platform

gktest

Kubernetes Engine

Kubernetes 클러스터

클러스터 만들기

배포

새로고침

삭제

정보 패널 표시

클러스터

작업 부하

서비스

애플리케이션

구성

Kubernetes 클러스터는 컨테이너화된 애플리케이션을 실행하기 위한 VM 인스턴스의 관리형 그룹입니다. 자세히 알아보기

라벨 또는 이름으로 필터링

<input type="checkbox"/> 이름 ^	위치	클러스터 크기	총 코어수	총 메모리	알림	라벨
<input checked="" type="checkbox"/> standard-cluster-1	asia-northeast1-a	3	vCPU 3개	11.25GB		<div>연결</div>

클러스터에 연결

명령줄을 통하거나 대시보드를 사용하여 클러스터에 연결할 수 있습니다.

명령줄 액세스

다음 명령을 실행하여 `kubectl` 명령줄 액세스를 구성합니다.

```
$ gcloud container clusters get-credentials standard-cluster-1 --zone asia-northeast1-a --project gktest-244306
```

Cloud Shell에서 실행

GKE를 활용한 쿠버네티스 사용

Kubernetes Engine 클러스터 만들기

- 연결 후에 미리 입력된 명령어로 권한 설정

Cloud Shell

(gketest-244306) x + ▾

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to gketest-244306.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
isc0304@cloudshell:~ (gketest-244306)$ gcloud container clusters get-credentials standard-cluster-1 --zone asia-northeast1-a --project gketest-244306
Fetching cluster endpoint and auth data.
kubeconfig entry generated for standard-cluster-1.

To take a quick anonymous survey, run:
$ gcloud alpha survey

isc0304@cloudshell:~ (gketest-244306)$
```

- Kubectl 명령어로 현재 클러스터 확인

```
isc0304@cloudshell:~ (gketest-244306)$ kubectl get node
```

NAME	STATUS	ROLES	AGE	VERSION
gke-standard-cluster-1-default-pool-b1e2cd6b-3r3q	Ready	<none>	6m2s	v1.12.8-gke.6
gke-standard-cluster-1-default-pool-b1e2cd6b-cb83	Ready	<none>	6m4s	v1.12.8-gke.6
gke-standard-cluster-1-default-pool-b1e2cd6b-z2nd	Ready	<none>	6m4s	v1.12.8-gke.6

GKE를 활용한 쿠버네티스 사용

▶ 애플리케이션 배포

- 왼쪽 메뉴에 애플리케이션을 클릭하고 마켓플레이스에서 배포
- 원하는 앱 선택하고 “구성”

The screenshot displays the Google Cloud Platform (GCP) console interface. At the top, the navigation bar shows 'Google Cloud Platform' and a dropdown menu with 'gktest'. The left sidebar contains the 'Kubernetes Engine' section, where the '애플리케이션' (Applications) option is highlighted with a red box. The main content area is titled 'Kubernetes Engine 애플리케이션' and features a red box around the 'Marketplace에서 배포' (Deploy from Marketplace) button. Below this, a list of applications is displayed, including Gestalt, JFrog Artifactory Enterprise, GitLab, Magalix Agent, CloudBees Core, 1Password SCIM bridge, Aerospike Enterprise Edition for GKE, Agones, Airflow Operator, Aqua Security, Cassandra Cluster, Citrix ADC CPX, Citrix Ingress Controller, CloudBees Core, and Codefresh. Each application card includes its logo, name, and a brief description.

GKE를 활용한 쿠버네티스 사용

▶ 서비스 확인

- 서비스 탭에서 서비스 설정된 공인 IP 확인 가능

Google Cloud Platform

gktest

Kubernetes Engine

서비스

새로고침

삭제

클러스터

작업 부하

서비스

애플리케이션

구성

저장소

Kubernetes 서비스

브로커 서비스

베타

서비스는 탐색 및 부하 분산에 사용할 수 있는 네트워크 엔드포인트가 포함된 포트 집합입니다. 수신은 외부 HTTP(S) 트래픽을 서비스로 라우팅하기 위한 규칙 모음입니다.

시스템 개체 : False

리소스 필터링

이름	상태	유형	엔드포인트	포트	네임스페이스	클러스터
<input type="checkbox"/> application-controller-manager-service	확인	클러스터 IP	10.4.1.178	1 / 1	application-system	standard-cluster-1
<input type="checkbox"/> nginx-1-nginx-prometheus-svc	확인	클러스터 IP	None	3 / 3	default	standard-cluster-1
<input checked="" type="checkbox"/> nginx-1-nginx-svc	확인	부하 분산기	35.200.70.116:80 35.200.70.116:443	3 / 3	default	standard-cluster-1

← → ↺

주의 요함 | 35.200.70.116

Web site powered by NGINX

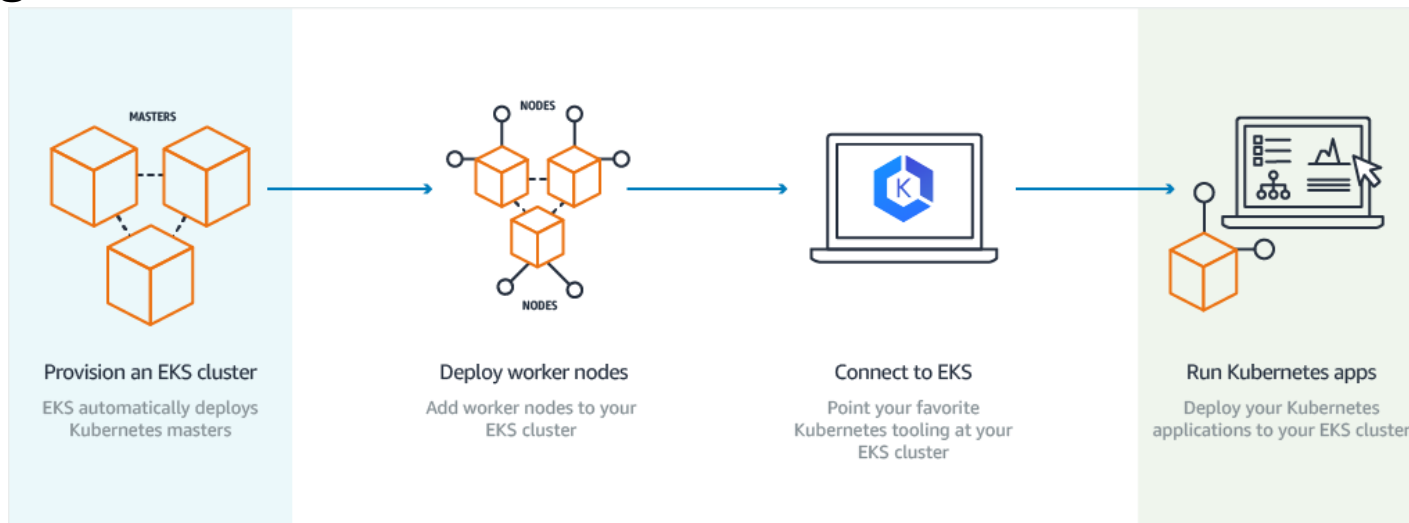
This page is under construction. Please come back soon!

AWS EKS를 활용한 쿠버네티스 사용

AWS EKS를 활용한 쿠버네티스 사용

Amazon Elastic Container Service for Kubernetes(Amazon EKS)

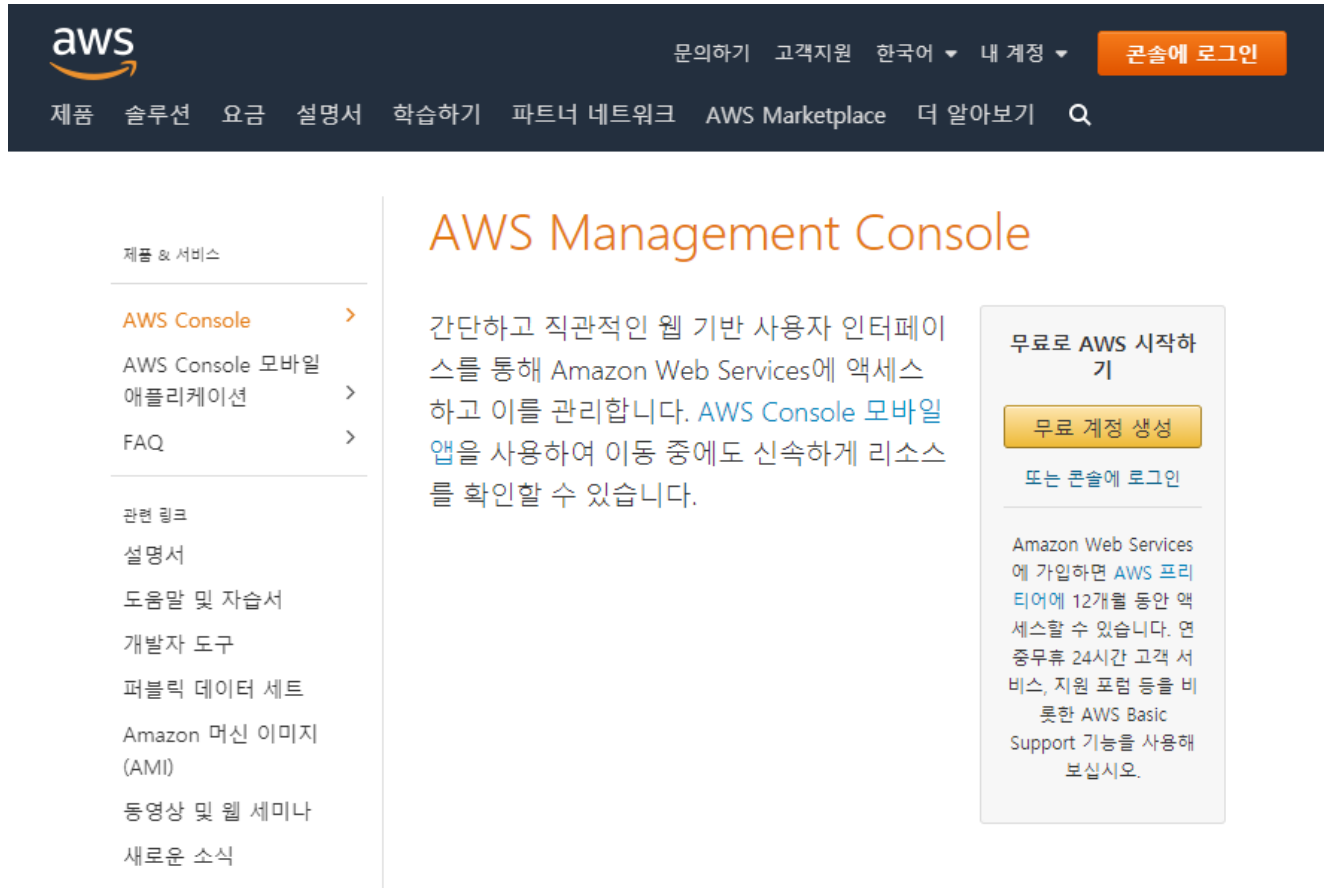
- AWS에서 Kubernetes를 손쉽게 실행하도록 하는 관리형 서비스
- 여러 가용 영역에서 Kubernetes 제어 플레인 인스턴스를 실행하여 고가용성을 보장
- 비정상 제어 플레인 인스턴스를 자동으로 감지하고 교체
- 자동화된 버전 업그레이드를 제공
- 여러 AWS 서비스와 통합되어 다음을 포함한 애플리케이션에 대한 확장성과 보안을 제공
 - 컨테이너 이미지용 Amazon ECR
 - 로드 배포용 Elastic Load Balancing
 - 인증용 IAM
 - 격리용 Amazon VPC



AWS EKS를 활용한 쿠버네티스 사용

▶ AWS 회원 가입 (무료 계정 생성)

- ASW 프리티어에 12개월 동안 액세스 가능
- <https://aws.amazon.com/ko/console/>



The screenshot shows the AWS Management Console homepage. At the top is a dark navigation bar with the AWS logo, links for '문의하기', '고객지원', '한국어', '내 계정', and a '콘솔에 로그인' button. Below this is a secondary navigation bar with links for '제품', '솔루션', '요금', '설명서', '학습하기', '파트너 네트워크', 'AWS Marketplace', and '더 알아보기'. The main content area is divided into two columns. The left column contains a '제품 & 서비스' section with links to 'AWS Console', 'AWS Console 모바일 애플리케이션', and 'FAQ', followed by a '관련 링크' section with links to '설명서', '도움말 및 자습서', '개발자 도구', '퍼블릭 데이터 세트', 'Amazon 머신 이미지 (AMI)', '동영상 및 웹 세미나', and '새로운 소식'. The right column features the 'AWS Management Console' title, a paragraph explaining that it's a simple and intuitive web-based interface for accessing and managing Amazon Web Services, and a '무료로 AWS 시작하기' section with a '무료 계정 생성' button and a link to '또는 콘솔에 로그인'. A detailed note about the 'AWS 프리티어' (AWS Free Tier) is also present, stating that users can access it for 12 months with 24/7 customer service and AWS Basic Support.

aws

문의하기 고객지원 한국어 ▼ 내 계정 ▼ 콘솔에 로그인

제품 솔루션 요금 설명서 학습하기 파트너 네트워크 AWS Marketplace 더 알아보기

AWS Management Console

간단하고 직관적인 웹 기반 사용자 인터페이스를 통해 Amazon Web Services에 액세스하고 이를 관리합니다. [AWS Console 모바일 앱](#)을 사용하여 이동 중에도 신속하게 리소스를 확인할 수 있습니다.

무료로 AWS 시작하기

무료 계정 생성

또는 콘솔에 로그인

Amazon Web Services에 가입하면 [AWS 프리티어](#)에 12개월 동안 액세스할 수 있습니다. 연중무휴 24시간 고객 서비스, 지원 포럼 등을 비롯한 AWS Basic Support 기능을 사용해 보십시오.

AWS EKS를 활용한 쿠버네티스 사용

▶▶▶ 아마존 EKS 시작하기

- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started.html
- AWS에서는 EKS를 시작하는 두 가지 방법을 제공
 - eksctl로 시작하기
 - ✓ Amazon EKS를 시작하는 가장 빠르고 쉬운 방법
 - ✓ 클러스터를 생성 및 관리하기 위한 간단한 명령줄 유틸리티인 eksctl 제공
 - ✓ 필요한 모든 리소스를 설치
 - ✓ kubectl 명령 줄 유틸리티
 - AWS Management
 - ✓ AWS Management 콘솔 사용
 - ✓ Amazon EKS를 시작할 때 필요한 모든 리소스를 생성 가능
 - ✓ Amazon EKS 또는 AWS CloudFormation 콘솔을 사용하여 각 리소스를 수동으로 생성
 - ✓ 각 리소스의 생성 방법 및 리소스 간의 상호 작용을 완벽하게 파악 가능
 - ✓ Amazon EKS를 시작하는 방법으로는 더 복잡하고 시간도 많이 걸림

AWS EKS를 활용한 쿠버네티스 사용

▶▶ 아마존 EKS 시작하기

- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started-eksctl.html
- aws 커맨드 설치 및 자격 증명

```
$ sudo apt install awscli
```

```
$ aws configure
```

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
```

```
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

```
Default region name [None]: ap-northeast-2
```

```
Default output format [None]: json
```

AWS EKS를 활용한 쿠버네티스 사용

▶▶ 아마존 EKS 시작하기

- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started-eksctl.html
- eksctl 설치

```
$ curl --silent --location  
"https://github.com/weaveworks/eksctl/releases/download/latest_release/eksctl_$(u  
name -s)_amd64.tar.gz" | tar xz -C /tmp  
$ sudo mv /tmp/eksctl /usr/local/bin  
$ eksctl version
```

- 설치 중에 kubectl이 없다는 명령어가 나오는 경우 다음 명령어를 실행
\$ sudo snap install kubectl --classic

AWS EKS를 활용한 쿠버네티스 사용

▶▶ 아마존 EKS 시작하기

- https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/getting-started-eksctl.html
- eksctl로 클러스터 생성하기

```
eksctl create cluster \  
--name prod \  
--version 1.12 \  
--nodegroup-name standard-workers \  
--node-type t3.medium \  
--nodes 3 \  
--nodes-min 1 \  
--nodes-max 4 \  
--node-ami auto
```

쿠버네티스에서 앱 실행해보기

쿠버네티스에서 앱 실행해보기

복습 시간!

- 쿠버네티스에 올릴 컨테이너 이미지를 하나 생성
- 포트 8080 에서 HTTP 서버를 시작
- /로 요청하는 경우 Index 함수 실행
- "Welcome! <hostname>"으로 응답

package main

main.go

import (

 "fmt"

 "github.com/julienschmidt/httprouter"

 "net/http"

 "log"

 "os"

)

func Index(w http.ResponseWriter, r *http.Request, _ httprouter.Params) {

 hostname, err := os.Hostname()

 if err == nil {

 fmt.Fprint(w, "Welcome! " + hostname + "\n")

 } else {

 fmt.Fprint(w, "Welcome! ERROR\n")

 }

}

func main() {

 router := httprouter.New()

 router.GET("/", Index)

 log.Fatal(http.ListenAndServe(":8080", router))

}

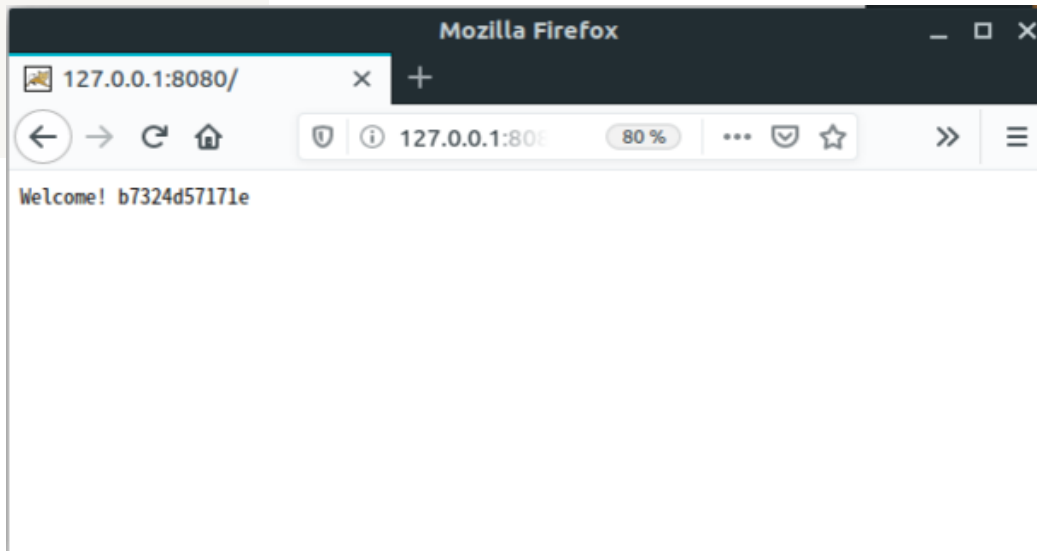
쿠버네티스에서 앱 실행해보기

복습 시간!

● Go 언어 설치 및 프로그램 빌드

- apt install golang를 사용해서 고 랭기지를 설치
- go get 명령을 사용해 외부 라이브러리 импорт
- go build 명령으로 main.go를 빌드하여 main 파일 생성
- main 명령 실행 후 정상 동작하는 지 테스트

```
apt install golang
go get github.com/julienschmidt/httprouter
go build main.go
main
```



쿠버네티스에서 앱 실행해보기

▶▶ 복습 시간!

● dockerfile 작성

- FROM golang:1.11 버전의 컨테이너 이미지를 사용
- 로컬 디렉터리의 main 파일을 이미지의 /usr/src/app 디렉터리에 동일한 이름으로 추가
- 이미지를 실행할 때 /usr/src/app/main 실행

dockerfile

```
FROM golang:1.11
WORKDIR /usr/src/app
COPY main /usr/src/app
CMD ["/usr/src/app/main"]
```

쿠버네티스에서 앱 실행해보기

▶▶ 복습 시간!

● 컨테이너 이미지 만들기

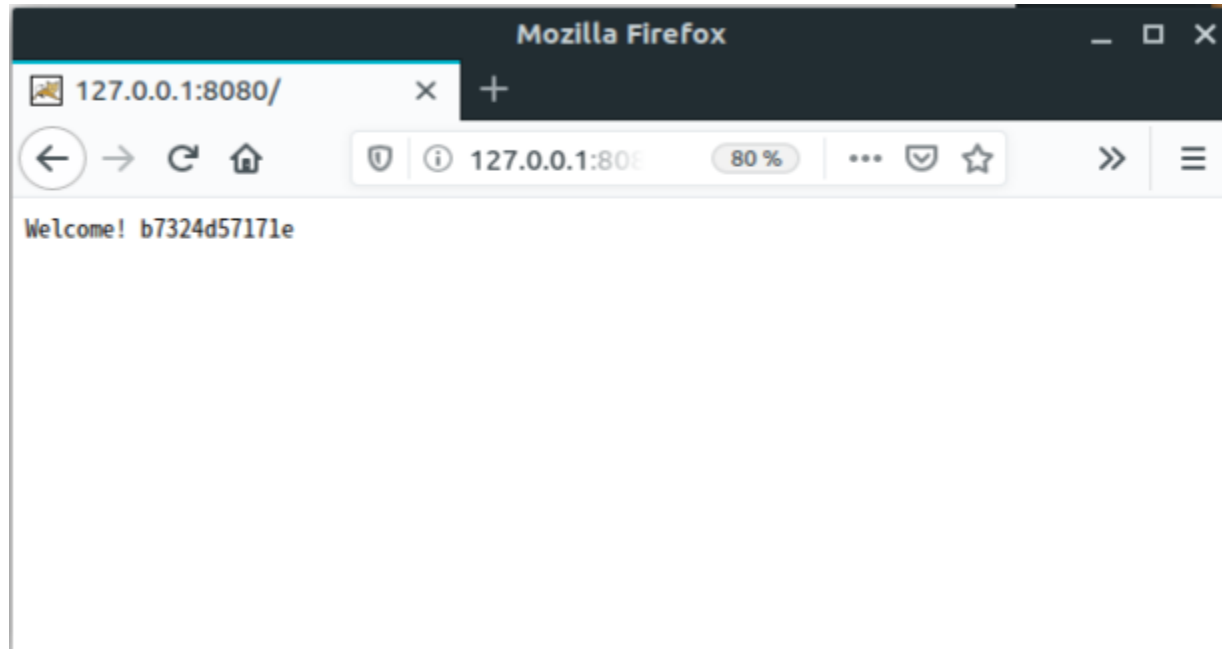
```
# docker build -t http-go .  
Sending build context to Docker daemon 6.681MB  
Step 1/4 : FROM golang:1.11  
----> 43a154fee764  
Step 2/4 : WORKDIR /usr/src/app  
----> Using cache  
----> 6c3e19fde006  
Step 3/4 : COPY main /usr/src/app  
----> Using cache  
----> aef49d15fc7e  
Step 4/4 : CMD ["/usr/src/app/main"]  
----> Using cache  
----> cefdc3afaac7  
Successfully built cefdc3afaac7  
Successfully tagged http-go:latest
```

쿠버네티스에서 앱 실행해보기

복습 시간!

- 컨테이너 실행하기

```
# docker run -d -p 8080:8080 --rm http-go
```



쿠버네티스에서 앱 실행해보기

복습 시간!

● 컨테이너 푸시하기

```
# docker tag http-go gasbugs/http-go
```

```
# docker login
```

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

Username: gasbugs

Password:

...

Login Succeeded

```
# docker push gasbugs/http-go
```

The push refers to repository [docker.io/gasbugs/http-go]

303fd7755a94: Pushed

cbd84fde8a9e: Pushed

18dccb3264f2: Mounted from library/golang

7ffc9f27ea7: Mounted from library/golang

39a8c34bbaf3: Mounted from library/golang

97e8dd85db4e: Mounted from library/golang

74e2ede3b29c: Mounted from library/golang

6d5a64ea8f37: Mounted from library/golang

660314270d76: Mounted from library/golang

latest: digest: sha256:584f70a5ccda2a7ca30d4b1eecbb5e46f980a3b2f8f9160e1282e167c5cdf4dc size: 2213

쿠버네티스에서 앱 실행해보기

▶▶ 처음으로 만드는 쿠버네티스 앱

- 보통 배포하려는 모든 컴포넌트의 설명이 기술된 JSON 또는 YAML 매니페스트를 준비 필요
- 이를 위해서는 쿠버네티스에서 사용되는 컴포넌트 유형을 잘 알아야 함
- 여기서는 명령어에 몇 가지 옵션으로 디스크립션을 간단히 전달하여 한 줄로 앱을 실행

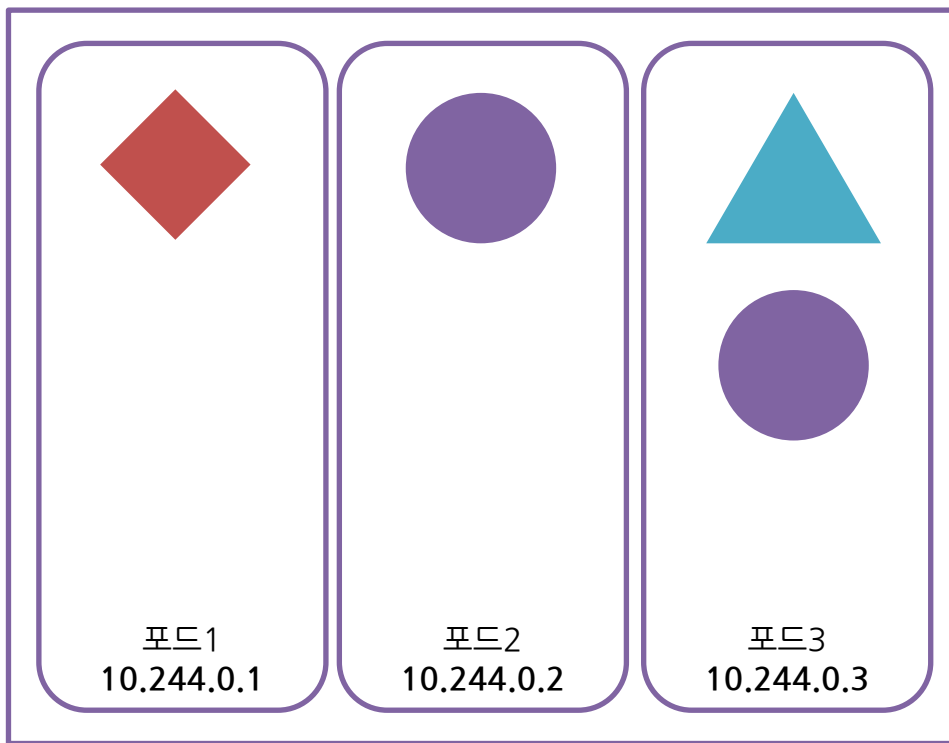
```
$ kubectl create deploy http-go --image=gasbugs/http-go  
deployment/http-go created
```

쿠버네티스에서 앱 실행해보기

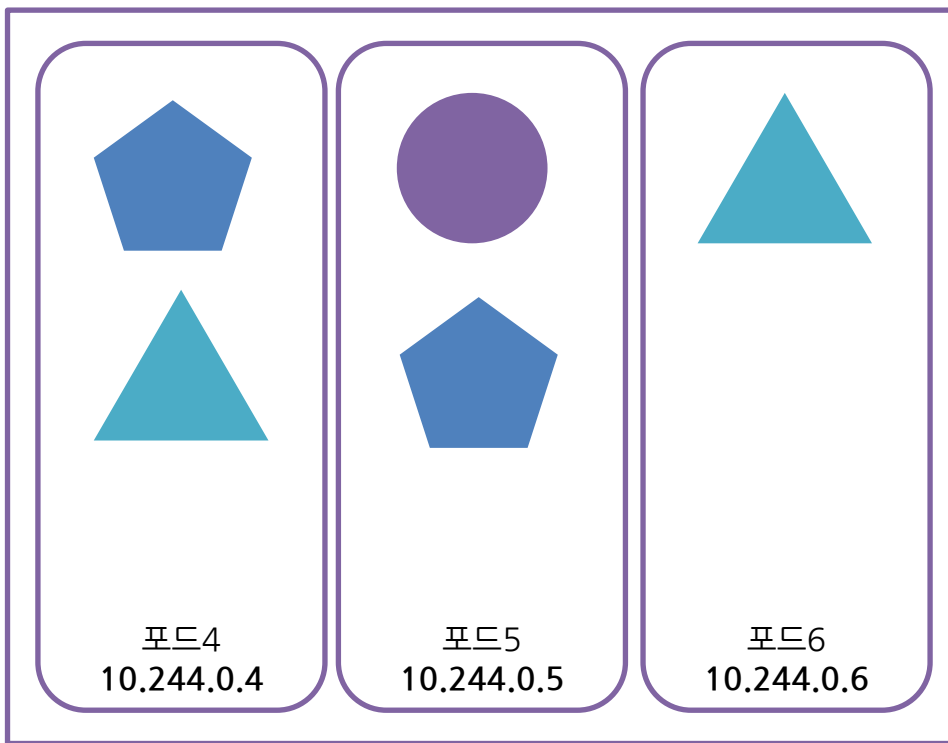
▶ 포드(pod)란?

- 쿠버네티스는 `kubectl get container`와 같이 컨테이너를 취급하지 않는다!
- 대신 여러 위치에 배치된 컨테이너 개념인 컨테이너 그룹을 포드(Pod)라는 개념을 사용

워커 노드1



워커 노드2



쿠버네티스에서 앱 실행해보기

▶ 포드의 특징

- 포드는 하나 이상의 밀접하게 관련된 컨테이너로 구성된 그룹
- 동일한 리눅스 네임스페이스와 동일한 워커 노드에서 항상 함께 실행
- 각 포드는 애플리케이션을 실행하는 자체 IP, 호스트 이름, 프로세스 등이 있는 별도의 논리적 시스템

▶ 포드 나열하기

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
http-go-XXXXXXXXXX-th8gg	0/1	Pending	0	5s

```
$ kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
http-go-XXXXXXXXXX-th8gg	1/1	Running	0	1m

쿠버네티스에서 앱 실행해보기

▶ 웹 애플리케이션 만들어보기

- 실행 중인 포드는 클러스터의 가상 네트워크에 포함돼 있음
- 어떻게 액세스 할 수 있을까?
- 외부에서 액세스하려면 서비스 객체를 통해 IP를 노출하는 것이 필요
- LoadBalancer라는 서비스를 작성하면 외부 로드 밸런서가 생성
- 로드 밸런서의 공인 IP를 통해 포드에 연결 가능
(하지만 로컬 쿠버네티스에서는 동작하지 않으며 externalDNS가 필요함,
이 기능은 GKE, EKS 같은 클라우드에서 사용 가능(구글, AWS 계정 필요))

```
$ kubectl expose deployment http-go --type=LoadBalancer --name http-go-svc --port=8080 -  
-target-port=8080  
service/http-go-svc exposed
```

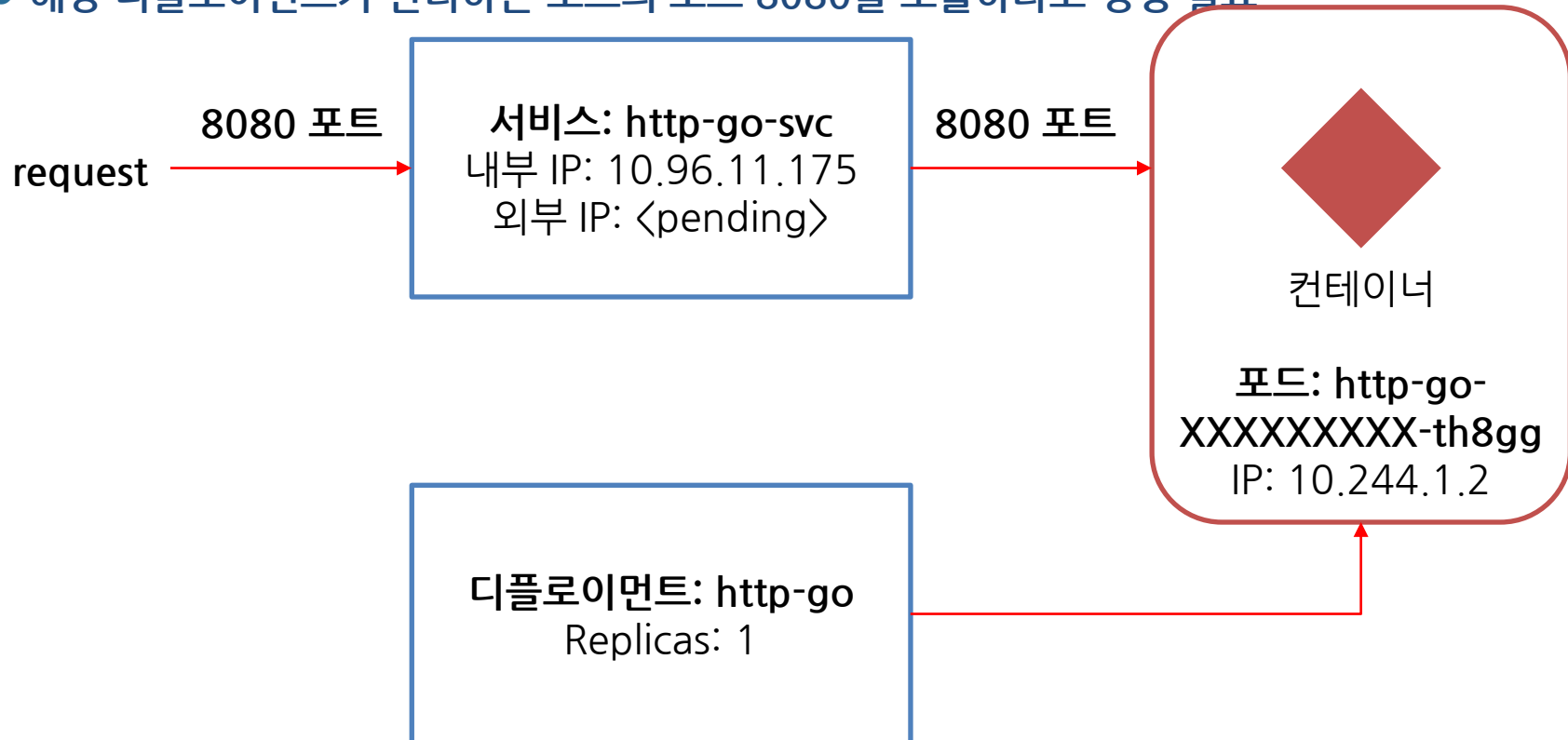
```
$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	33m
http-go-http	LoadBalancer	10.109.140.155	<pending>	8080:32464/TCP	21s

쿠버네티스에서 앱 실행해보기

디플로이먼트, 포드, 서비스가 동작하는 방식 이해

- 사실 실제로 포드도 직접 만들지 않음
- `kubectl create deploy` 명령을 실행하면 디플로이먼트가 생성
- 디플로이먼트가 실제 포드 객체를 생성
- 해당 디플로이먼트가 관리하는 포드의 포트 8080을 노출하라고 명령 필요



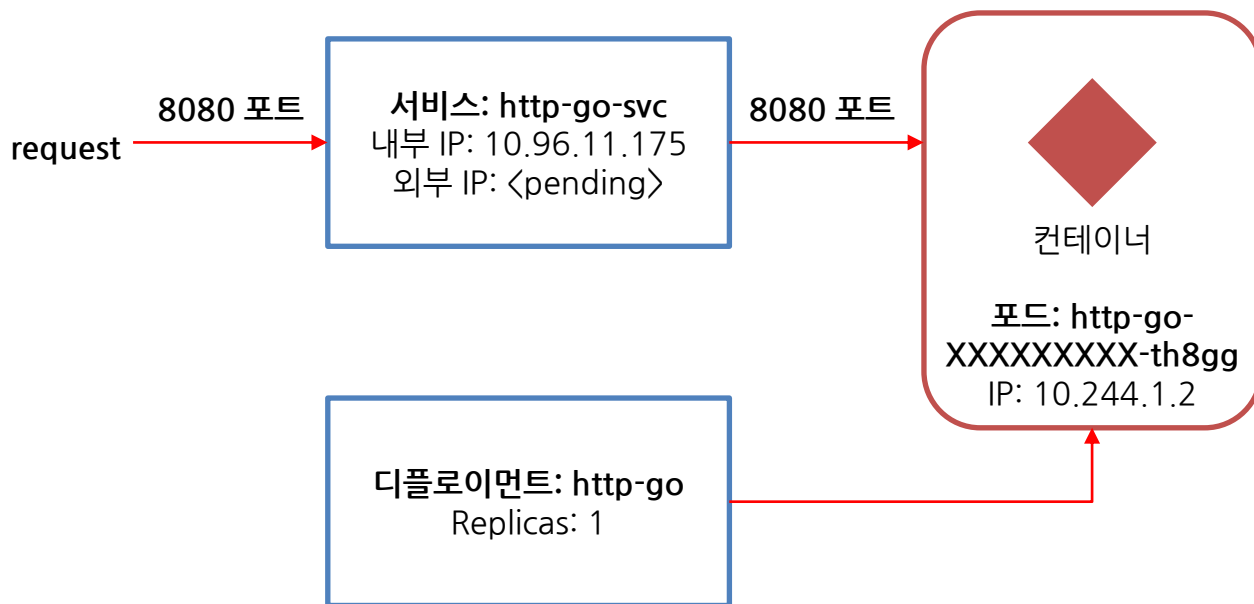
쿠버네티스에서 앱 실행해보기

디플로이먼트의 역할

- 디플로이먼트는 레플리카셋을 생성
- 레플리카셋은 수를 지정하여 알려주면 그 수만큼 포드를 유지
- 어떤 이유로든 포드가 사라지면 레플리카셋은 누락된 포드를 대체할 새로운 포드를 생성

```
$ kubectl get deployment
```

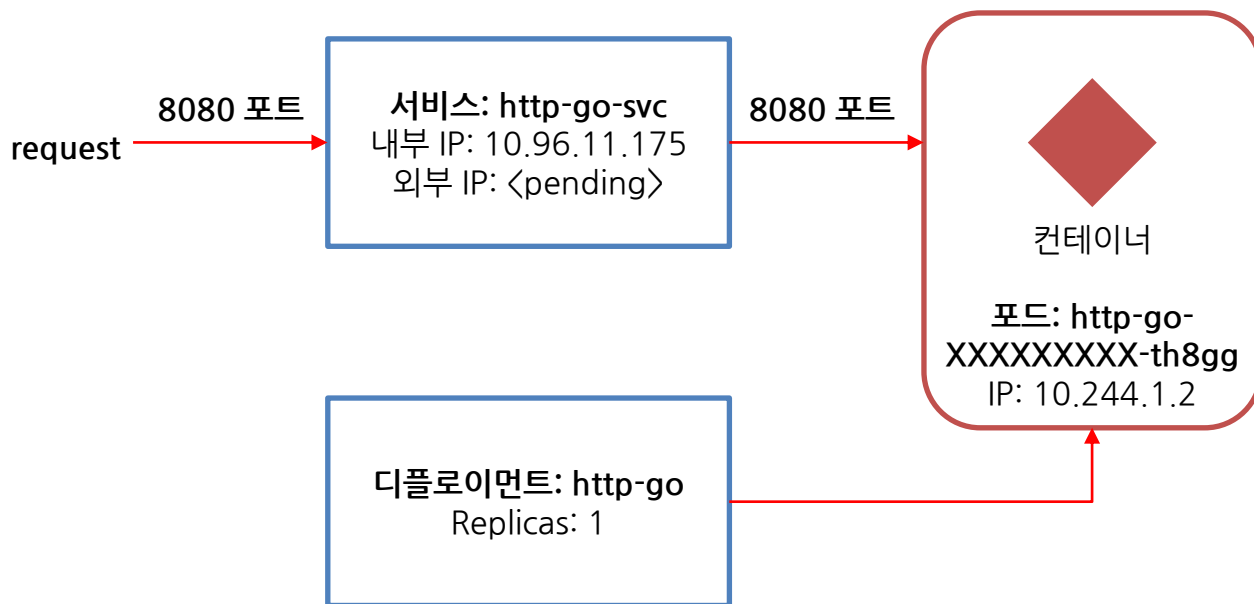
NAME	READY	UP-TO-DATE	AVAILABLE	AGE
http-go	1/1	1	1	16m



쿠버네티스에서 앱 실행해보기

서버의 역할

- 포드는 일시적이므로 언제든지 사라질 가능성 존재
- 포드가 다시 시작되는 경우에는 언제든지 IP와 ID 변경됨
- 서비스는 변화하는 포드 IP 주소의 문제를 해결하고 단일 IP 및 포트 쌍에서 여러 개의 포드 노출
- 서비스가 생성되면 정적 IP를 얻게 되고 서비스의 수명 내에서는 변하지 않음
- 클라이언트는 포드에 직접 연결하는 대신 IP 주소를 통해 서비스에 연결
- 서비스는 포드 중 하나로 연결을 포워딩



쿠버네티스에서 앱 실행해보기

애플리케이션의 수평 스케일링

- 쿠버네티스를 사용해 얻을 수 있는 큰 이점 중 하나는 간단하게 컨테이너의 확장이 가능하다는 점
- 포드의 개수를 늘리는 것도 쉽게 가능
- 포드는 디플로이먼트가 관리

```
$ kubectl scale deploy http-go --replicas=3  
deployment/http-go scaled
```

```
$ kubectl get deploy
```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
http-go	3/3	3	3	45m

```
$ kubectl get pod
```

NAME	READY	STATUS	RESTARTS	AGE
http-go-gpxj8	0/1	ContainerCreating	0	18s
http-go-t2qs6	1/1	Running	0	18s
http-go-th8gg	1/1	Running	0	45m

쿠버네티스에서 앱 실행해보기

직접 앱에 접근하기

- curl 명령어로 요청
- external IP를 할당 받지 못했기 때문에 포드의 힘을 빌려 요청

```
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-ccsc2
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-bt4xq
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-ccsc2
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-bt4xq
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-bt4xq
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-jvsg4
$ kubectl exec http-go-XXXXXX-bt4xq -- curl -s http://10.109.140.155:8080
Welcome! http-go-XXXXXX-ccsc2
```

쿠버네티스에서 앱 실행해보기

앱의 위치 확인

```
$ kubectl get pod -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	...
http-go-ccsc2	1/1	Running	0	7m18s	10.244.2.3	slave2-virtualbox	...
http-go-jvsg4	1/1	Running	0	7m	10.244.2.4	slave2-virtualbox	...

포드의 자세한 내용 살펴보기

- 예정된 노드, 시작된 시간, 실행 중인 이미지 등 유용한 정보 포함

```
$ kubectl describe pod http-go-XXXXXX-bt4xq
```

```
Name:                http-go-XXXXXX-bt4xq
Namespace:           default
Priority:             0
PriorityClassName:    <none>
Node:                slave1-virtualbox/10.0.2.11
Start Time:          Tue, 18 Jun 2019 17:23:04 +0900
Labels:              run=http-go
Annotations:         <none>
Status:              Running
IP:                  10.244.1.4
Controlled By:       ReplicationController/http-go
Containers: [중략]
```

쿠버네티스에서 앱 실행해보기

▶ 쿠버네티스를 활용해서 Jenkins 서비스하기

- `kubectl delete all --all` 를 실행해서 모든 서비스를 지우고 시작한다.

쿠버네티스 관리자 인증 시험(CKA) 소개

쿠버네티스 관리자 인증 시험(CKA) 소개

▶ Certified Kubernetes Administrator (CKA) 프로그램

- <https://www.cncf.io/certification/cka/>
- 리눅스 재단과 협력하여 쿠버네티스 생태계 개발을 돕기 위해 CNC(Cloud Native Computing Foundation) 재단에 의해 시작
- CNC 재단은 쿠버네티스 관리자 커뮤니티를 성장시키기 위해 노력
- 쿠버네티스를 사용하는 광범위한 회사 및 조직에서 지속적으로 성장을 도모
- \$300의 비용으로 온라인 시험을 치룸 (쿠폰을 검색하면 나오니 쿠폰을 사용합시다!)
- WebCam을 통해 수험자를 모니터링하고 크롬으로 시험을 진행
- 재시험 1회 무료로 응시 가능



쿠버네티스 관리자 인증 시험(CKA) 소개

▶ Certified Kubernetes Administrator (CKA) 출제 범위

- Application Lifecycle Management - 8%
- Installation, Configuration & Validation - 12%
- Core Concepts - 19%
- Networking - 11%
- Scheduling - 5%
- Security - 12%
- Cluster Maintenance - 11%
- Logging / Monitoring - 5%
- Storage - 7%
- Troubleshooting - 10%



쿠버네티스 관리자 인증 시험(CKA) 소개

▶ 시험에서 자주 사용되는 문법 <http://bitly.kr/Mn1B8e>

- <https://www.notion.so/gasbugs/CKA-Kubectl-070fc3ece01848f2b8d32ae3fe310819>
- short url : <http://bitly.kr/Mn1B8e>

CKA 테스트 팁: Kubectl를 사용할 때 알아야 할 필수 간편 명령어

시험을 볼 때 정의 파일을 사용하여 주로 선언적 방식으로 작업하는 것보다 커맨드 명령을 사용하면 한 번에 작업을 빠르게 완료 할 수 있고 정의 템플릿을 쉽게 생성 가능하다. 이런 기능을 사용하면 시험 중 상당한 시간을 절약 할 수 있다.

기본적으로 알아둬야 하는 편의 옵션 두 가지

`--dry-run`

- 이 옵션을 추가하고 명령을 실행하면 리소스를 생성하지 않음
- 단순히 명령을 테스트하려는 경우에 사용
- 사용자의 명령어가 올바른지 알려줌

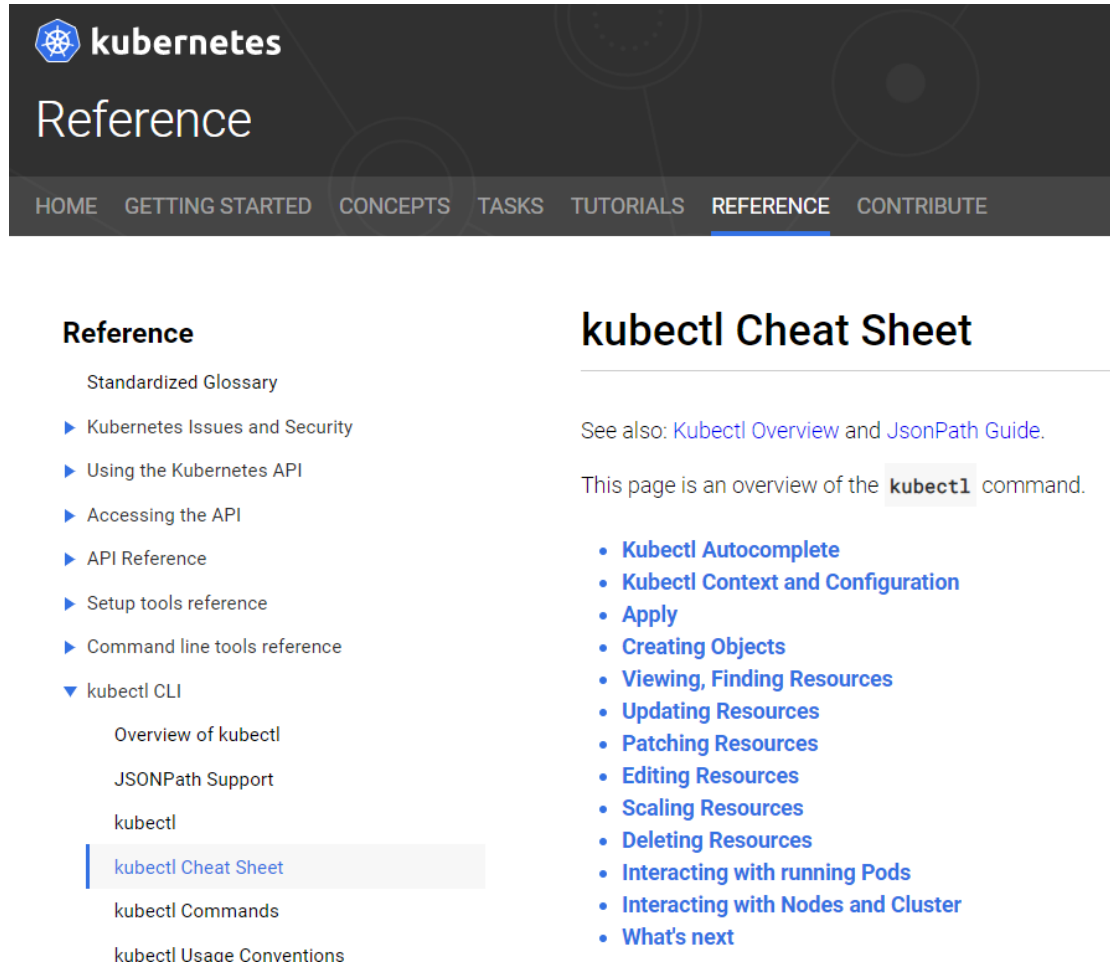
`-o yaml`

화면에 YAML 형식의 리소스 정의를 출력

쿠버네티스 관리자 인증 시험(CKA) 소개

▶ 쿠버네티스 cheatsheet

- <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>



The screenshot shows the Kubernetes Reference page. The header includes the Kubernetes logo and the word "kubernetes" in white on a dark background. Below the header is a navigation bar with links: HOME, GETTING STARTED, CONCEPTS, TASKS, TUTORIALS, REFERENCE (highlighted with a blue underline), and CONTRIBUTE. The main content area is titled "Reference" and contains a list of links: Standardized Glossary, Kubernetes Issues and Security, Using the Kubernetes API, Accessing the API, API Reference, Setup tools reference, Command line tools reference, and kubectl CLI. The kubectl CLI link is expanded, showing a list of sub-links: Overview of kubectl, JSONPath Support, kubectl, kubectl Cheat Sheet (highlighted with a blue bar), kubectl Commands, and kubectl Usage Conventions. To the right of the kubectl Cheat Sheet link, the title "kubectl Cheat Sheet" is displayed in a large font. Below the title, there is a link to "See also: Kubectl Overview and JsonPath Guide." and a paragraph stating "This page is an overview of the `kubectl` command." followed by a list of topics: Kubectl Autocomplete, Kubectl Context and Configuration, Apply, Creating Objects, Viewing, Finding Resources, Updating Resources, Patching Resources, Editing Resources, Scaling Resources, Deleting Resources, Interacting with running Pods, Interacting with Nodes and Cluster, and What's next.

Reference

- Standardized Glossary
- ▶ Kubernetes Issues and Security
- ▶ Using the Kubernetes API
- ▶ Accessing the API
- ▶ API Reference
- ▶ Setup tools reference
- ▶ Command line tools reference
- ▼ kubectl CLI
 - Overview of kubectl
 - JSONPath Support
 - kubectl
 - kubectl Cheat Sheet**
 - kubectl Commands
 - kubectl Usage Conventions

kubectl Cheat Sheet

See also: [Kubectl Overview](#) and [JsonPath Guide](#).

This page is an overview of the `kubectl` command.

- [Kubectl Autocomplete](#)
- [Kubectl Context and Configuration](#)
- [Apply](#)
- [Creating Objects](#)
- [Viewing, Finding Resources](#)
- [Updating Resources](#)
- [Patching Resources](#)
- [Editing Resources](#)
- [Scaling Resources](#)
- [Deleting Resources](#)
- [Interacting with running Pods](#)
- [Interacting with Nodes and Cluster](#)
- [What's next](#)