

2.2 쿠버네티스 클러스터 설치

2.1 을 통해 배운 것

- 컨테이너 이미지에 애플리케이션 패키징
- 도커 허브를 통해 이미지 공유
- 다른 머신들에서도 손쉽게 이미지 실행

쿠버네티스 클러스터

이전 과정에서는 도커에서 직접 이미지를 실행했다.

대신 이제 쿠버네티스 클러스터에 배포해 실행하는 방법을 진행해보자.

완전한 다중 노드 쿠버네티스 클러스터의 설치 는 리눅스/네트워크 관리에 능숙하지 않다면 어렵다.

- 올바른 쿠버네티스 설치 는 여러 물리 머신 또는 가상머신에 걸쳐 수행된다.
- 쿠버네티스 클러스터 내 실행되는 모든 컨테이너가 동일한 플랫 네트워킹 공간을 통해 서로 연결되도록 네트워크를 올바르게 설정해야 한다.

쿠버네티스 설치하는 방법은 다양 → <https://kubernetes.io/> 참조

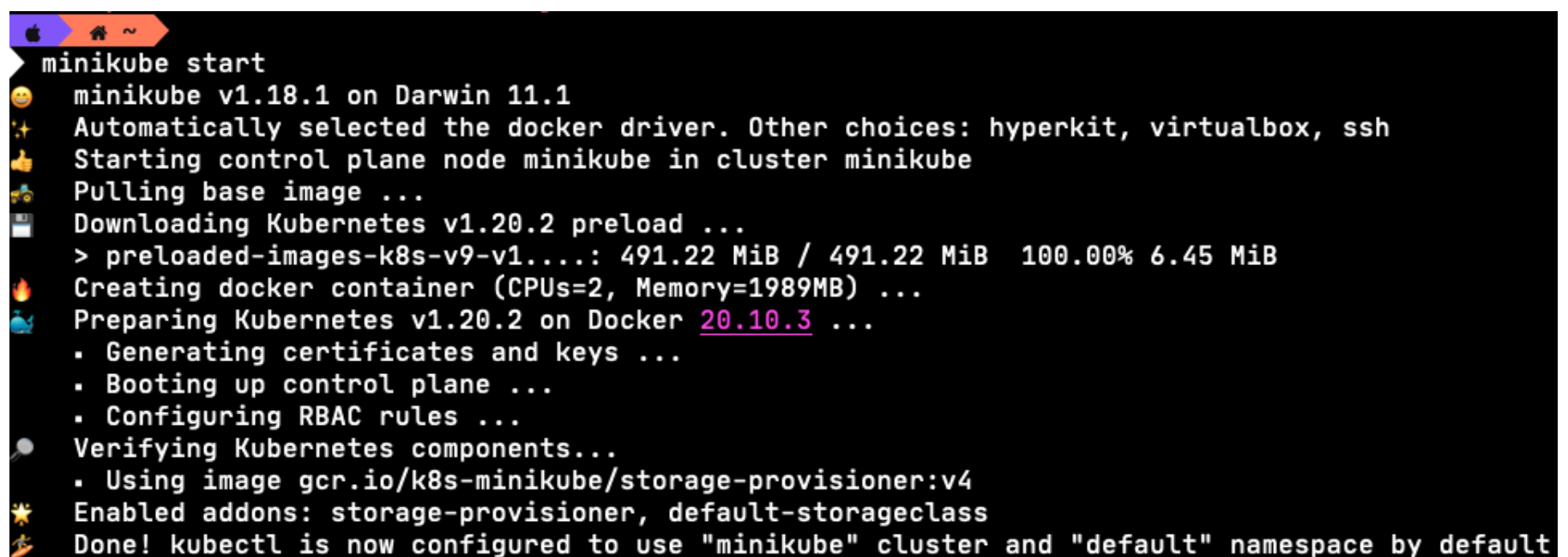
- 로컬 개발 머신, 클라우드 공급자 등을 통해 쿠버네티스 클러스터를 사용할 수 있음.

이 장에서는 쿠버네티스를 손쉽게 실행할 수 있는 2가지 옵션을 진행할 것

- 로컬에서 단일 노드 쿠버네티스 클러스터를 실행하는 방법 [minikube]
- 구글 쿠버네티스 엔진에 실행 중인 클러스터에 접근하는 방법 [gcloud]

Minikube를 활용한 단일 노드 쿠버네티스 클러스터 실행하기

- `brew install --cask virtualbox`
- `brew install kubectl`
- `brew install minikube`
- `minikube start`



```
minikube start
minikube v1.18.1 on Darwin 11.1
Automatically selected the docker driver. Other choices: hyperkit, virtualbox, ssh
Starting control plane node minikube in cluster minikube
Pulling base image ...
Downloading Kubernetes v1.20.2 preload ...
> preloaded-images-k8s-v9-v1....: 491.22 MiB / 491.22 MiB 100.00% 6.45 MiB
Creating docker container (CPUs=2, Memory=1989MB) ...
Preparing Kubernetes v1.20.2 on Docker 20.10.3 ...
. Generating certificates and keys ...
. Booting up control plane ...
. Configuring RBAC rules ...
Verifying Kubernetes components...
. Using image gcr.io/k8s-minikube/storage-provisioner:v4
Enabled addons: storage-provisioner, default-storageclass
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

클러스터 작동 여부 확인

```
kubectl cluster-info
```

```
▶ kubectl cluster-info
Kubernetes master is running at https://127.0.0.1:55001
KubeDNS is running at https://127.0.0.1:55001/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

- API 콘솔, 웹 콘솔 비롯한 다양한 쿠버네티스 구성요소 URL 표시

구글 쿠버네티스 엔진 활용한 관리형 쿠버네티스 클러스터 사용하기

- 모든 클러스터 노드와 네트워킹을 수동으로 설정할 필요가 없다. → 부담을 줄일 수 있다.
- 구글 클라우드 플랫폼에서 쿠버네티스 엔진 API 활성화한 후 SDK 다운로드 받아서 활용하면 된다.

```
gcloud components install kubectl
```

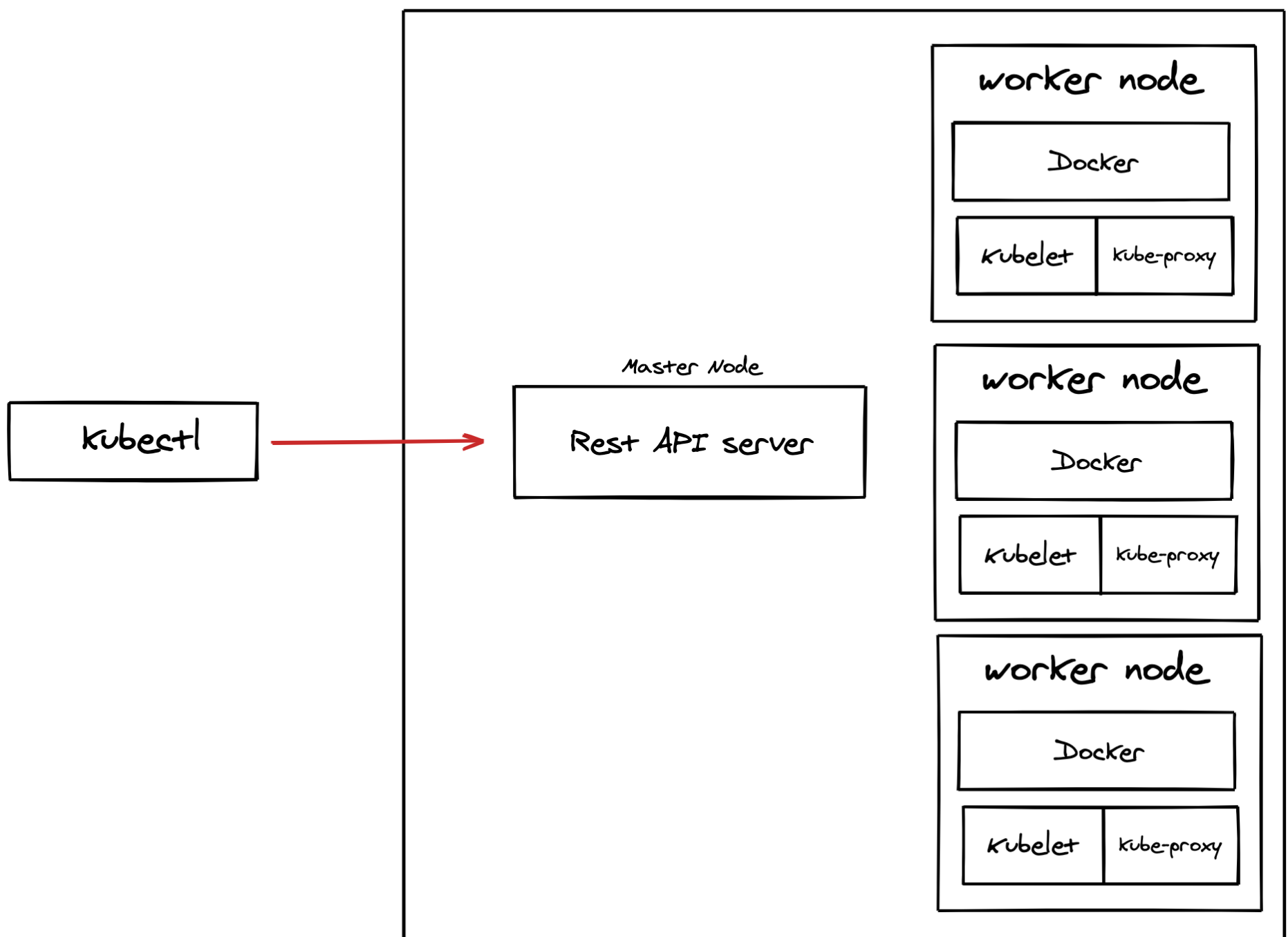
- kubectl 명령도구 설치

```
gcloud container clusters create kubia --num-nodes 3
```

- `--num-nodes 3`: 워커노드 3개를 가진 쿠버네티스 클러스터 설치

그럼 이제 클러스터와 상호작용하는 방법에 대해 알아보자.

kubernetes cluster



- 각 노드는 Docker, kublet, kube-proxy로 구성
- kubectl 클라이언트 명령어를 통해 마스터 노드인 쿠버네티스 API 서버에 요청을 보내서 클러스터와 상호작용한다.

```
kubectl get nodes
```

- 쿠버네티스 오브젝트 조회 → 즉, 워커 노드 조회
- 이름과 상태, 버전 등의 정보를 보여준다.

```
kubectl describe node [object이름]
```

- 오브젝트에 대한 상세정보 조회
- CPU, Memory, 시스템 정보등의 상세정보를 확인할 수 있다.