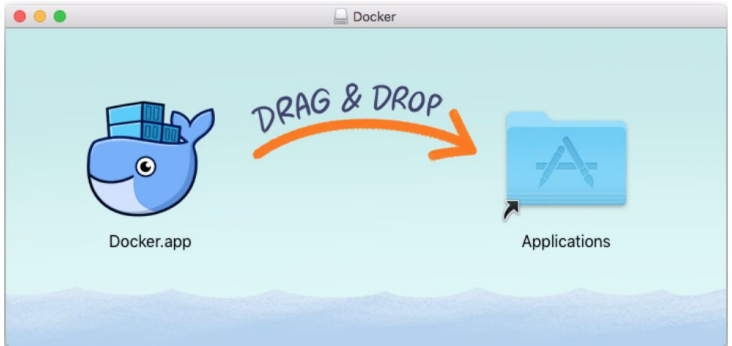
Kubernetes in Action chapter-2

Sunggon Song

Docker 설치(Mac)

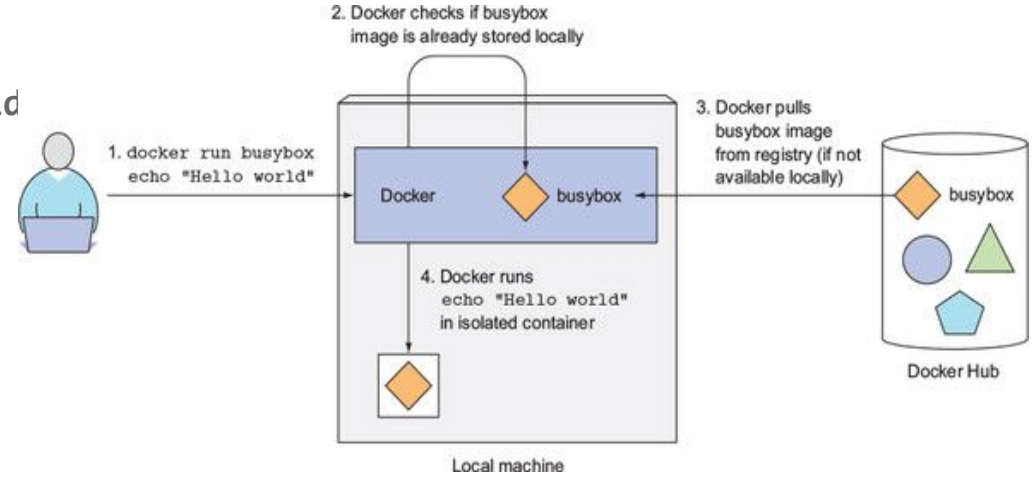
Install Docker for Mac

- To download Docker for Mac, head to Docker Hub.
  - https://download.docker.com/mac/stable/Docker.dmg
- Double-click Docker.dmg to open the installer, then drag Moby the whale to the Applications folder.
- Double-click Docker.app in the Applications folder to start Docker.
- 

Hello World 컨테이너 실행

```
$ docker run busybox echo "Hello World"
Unable to find image 'busybox:latest' locally
latest: Pulling from library/busybox
57c14dd66db0: Pull complete
Digest: sha256:7964ad52e396a6e045c39b5a44438424ac52e12e4d5a25d94895f2058cb863a0
Status: Downloaded newer image for busybox:latest
Hello World
```

```
$docker run busybox echo "Hello World
Hello World
```



간단한 Node.js app 생성

```
$ cat <<'EOF' > app.js
const http = require('http');
const os = require('os');

console.log("Kubia server starting...");

var handler = function(request, response) {
  console.log("Received request from " + request.connection.remoteAddress);
  response.writeHead(200);
  response.end("You've hit " + os.hostname() + "\n");
};

var www = http.createServer(handler);
www.listen(8080);
EOF
```

Node.js app 실행

```
$ node app.js
Kubia server starting...
Received request from ::1
```

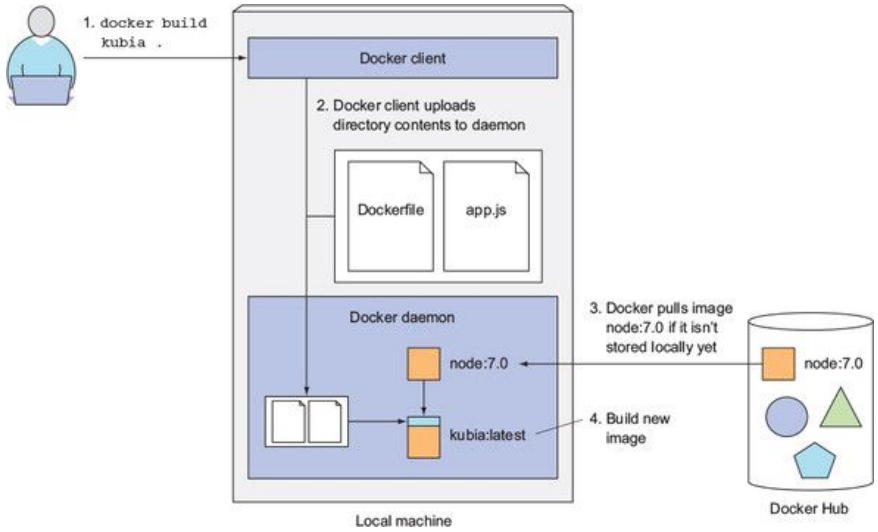
```
$ curl http://localhost:8080/
You've hit users-MacBook-Pro.local
```

Docker 이미지를 위한 Dockerfile 생성

```
$ cat <<'EOF' > Dockerfile
  FROM node:7
  ADD app.js /app.js
  CMD node app.js
  EOF
```

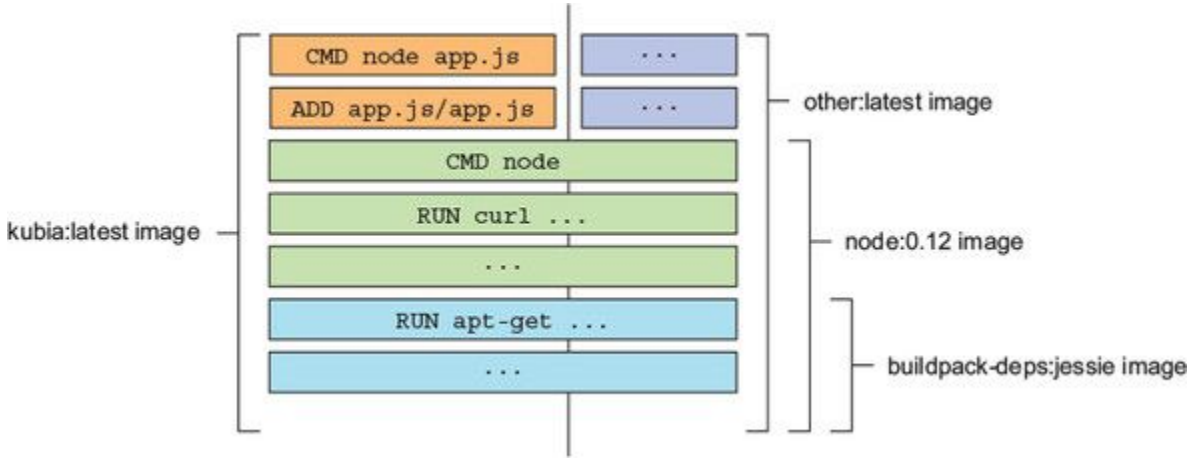
컨테이너 이미지 빌드하기

```
$ docker build -t kubia .
Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM node:7
 ---> d9aed20b68a4
Step 2/3 : ADD app.js /app.js
 ---> Using cache
 ---> fb87dfeb507d
Step 3/3 : ENTRYPOINT ["node", "app.js"]
 ---> Using cache
 ---> 368abcc4bef2
Successfully built 368abcc4bef2
Successfully tagged kubia:latest
```



이미지 layer 이해하기

```
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
kubia               latest              368abcc4bef2        3 days ago          660MB
busybox             latest              3a093384ac30        13 days ago         1.2MB
node                7                   d9aed20b68a4        17 months ago       660MB
```



컨테이너 이미지 실행하기

```
$ docker run --name kubia-container -p
8080:8080 -d kubia
```

- --name : 컨테이너의 이름을 지정
- -d : backgroud(detach mode)로 실행
- -p 8080:8080 : 컨테이너의 포트 8080(뒤)이 호스트 시스템의 포트 8080(앞)에 바인드

실행중인 컨테이너 확인

```
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS                    NAMES
831a05a16037        kubia               "node app.js"       3 days ago
Up 8 seconds        0.0.0.0:8080->8080/tcp   kubia-container
```

```
$ curl localhost:8080
You've hit 831a05a16037
```

컨테이너의 자세한 정보 확인하기

```
$ docker inspect kubia-container
...
```

```
$ docker inspect kubia-container |egrep "Port|8080"
        "PortBindings": {
            "8080/tcp": [
                    "HostPort": "8080"
        "PublishAllPorts": false,
        "ExposedPorts": {
            "8080/tcp": {}
        "Ports": {
            "8080/tcp": [
                    "HostPort": "8080"
```

실행 중인 컨테이너 내부 보기

```
$ docker exec -it kubia-container bash
  -i, --interactive    Keep STDIN open even if not attached
  -t, --tty            Allocate a pseudo-TTY
```

```
root@831a05a16037:/# ps -ef
UID        PID  PPID  C STIME TTY          TIME CMD
root         1     0  0 11:44 ?        00:00:00 node app.js
root        30     0  1 11:55 pts/0    00:00:00 bash
root        35    30  0 11:55 pts/0    00:00:00 ps -ef
root@831a05a16037:/# exit
```

실행 중인 컨테이너 내부 보기

```
$ docker exec -it kubia-container ps aux | grep node
root         1  0.0  1.2 614432 26452 ?        ssl  11:44   0:00 node app.js

$ docker exec -it kubia-container ls /
app.js   boot  etc   lib    media  opt   root  sbin  sys  usr
bin  dev   home  lib64  mnt    proc  run   srv   tmp  var
```

컨테이너의 종료 및 삭제

```
$ docker stop kubia-container
$ docker ps
$ docker ps -a
$ docker rm kubia-container
```

이미지 레지스트리에 컨테이너 이미지 올리기

```
$ docker tag kubia sunggonsong/kubia
```
⇒ 이미지에 태그(Tag) 추가

```
$ docker login
```
⇒ docker hub login

```
$ docker push sunggonsong/kubia
```
⇒ 레지스트리에 이미지 올리기

다른 시스템에서 이미지 실행하기

```
$ docker run -p 8080:8080 -d sunggonsong/kubia
Unable to find image 'sunggonsong/kubia:latest' locally
latest: Pulling from sunggonsong/kubia
ad74af05f5a2: Downloading [==============================================>    ]
48.49MB/52.61MB
2b032b8bbe8b: Download complete
a9a5b35f6ead: Downloading [================================================>  ]
41.42MB/43.23MB
3245b5a1c52c: Downloading [=======================>                           ]
64.55MB/131.9MB
afa075743392: Waiting
9fb9f21641cd: Waiting
3f40ad2666bc: Waiting
49c0ed396b49: Waiting
9a3e1e0242fc: Waiting
```

Kubernates 클러스터 설정

Minikube로 진행

Minikube설치(Mac)

```
$ brew cask install minikube
```

Minikube설치(Ubuntu 18.04)

```
$ sudo apt-get update
$ sudo apt-get install apt-transport-https
$ sudo apt-get upgrade
⇒ update & upgrade system
$ sudo apt install virtualbox virtualbox-ext-pack
⇒ install virtualbox
$ wget
https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64
chmod +x minikube-linux-amd64 --no-check-certificate
sudo mv minikube-linux-amd64 /usr/local/bin/minikube
$ minikube version
⇒ download & install minikube
```

Kubectl 설치(Ubuntu 18.04)

```
$ curl -s
https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo
apt-key add -
$ echo "deb http://apt.kubernetes.io/ kubernetes-xenial main"
| sudo tee /etc/apt/sources.list.d/kubernetes.list
$ sudo apt update
$ sudo apt -y install kubectl
$ kubectl version -o json
```

Minikube로 단일노드 Kubernetes cluster 시작

```
$ minikube start
⇒ Start minikube cluster
```

cluster 정보 표시

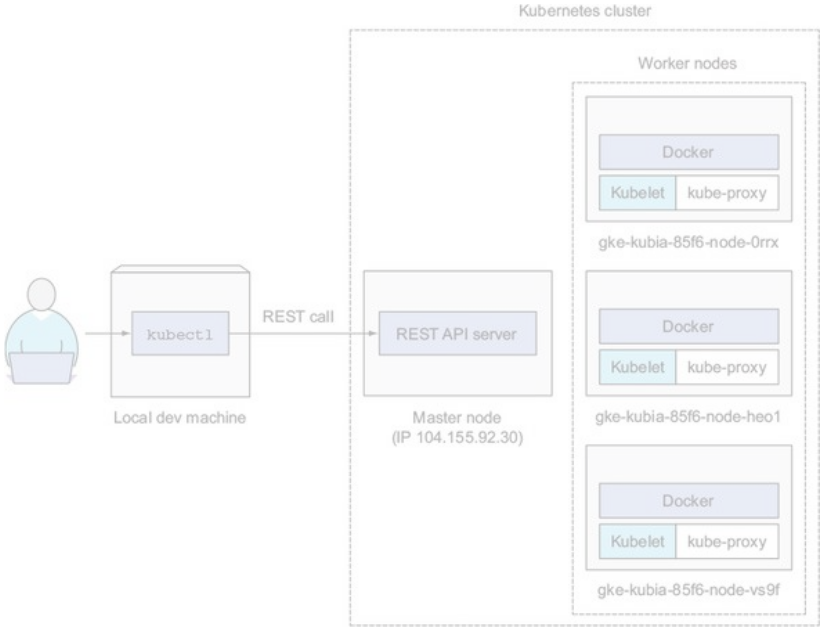
```
$ kubectl cluster-info
Kubernetes master is running at https://192.168.99.101:8443
KubeDNS is running at
https://192.168.99.101:8443/api/v1/namespaces/kube-system/services/kube-dns:dns
/proxy
```

Using a hosted Kubernetes cluster with Google Kubernetes Engine

. . .

Creating a Kubernetes cluster with three nodes- using GKE

```
$ gcloud container clusters create kubia --num-nodes 3 --machine-type f1-micro
```



cluster nodes 목록으로 cluster 동작 확인

```
$ kubectl get nodes
NAME       STATUS    ROLES     AGE       VERSION
minikube   Ready     master    2d        v1.12.4
```

객체의 세부 사항 확인

```
$ kubectl describe node minikube
Name:                   minikube
Roles:                  master
Labels:                 beta.kubernetes.io/arch=amd64
                        beta.kubernetes.io/os=linux
...
```

alias와 커맨드 라인 completion 셋업하기

```
$ alias k=kubectl
```

```
$ source <(kubectl completion zsh)
Or
$ source <(kubectl completion bash)
```

```
$ source <(kubectl completion zsh | sed s/kubectl/k/g)
```

Node.js app 배포

```
$ kubectl run kubia --image=sunggonsong/kubia --port=8080
--generator=run/v1
```

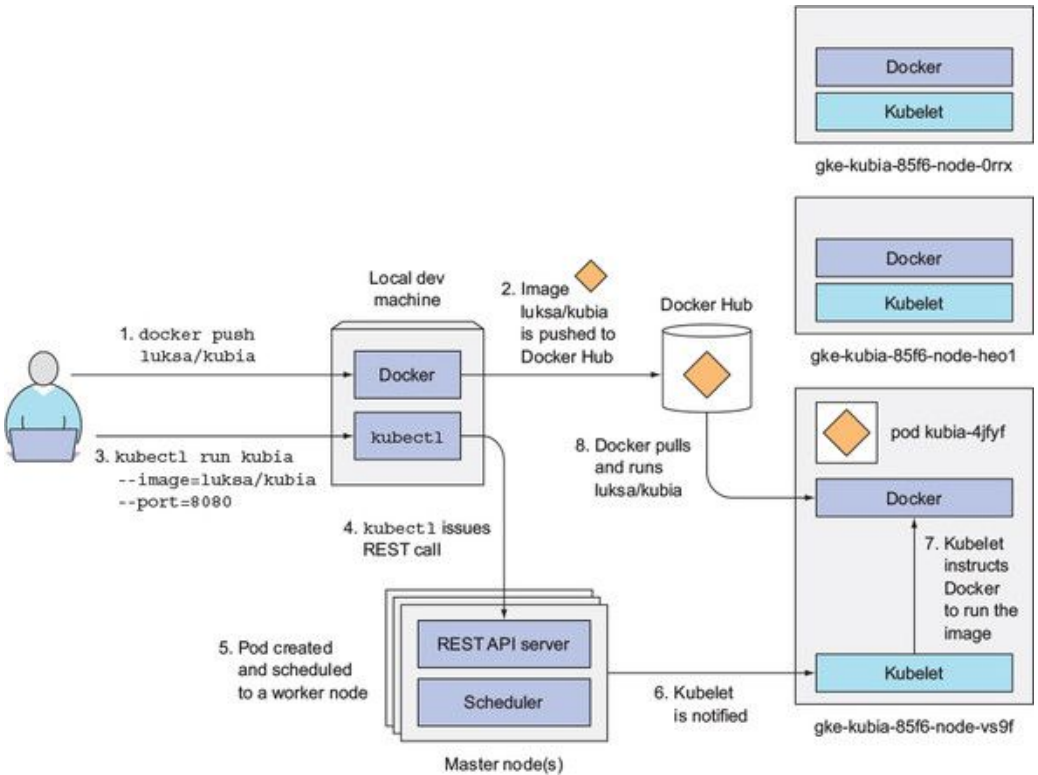
```
replicationcontroller "kubia" created
```

Listing Pods

```
$ kubectl get pods

NAME          READY     STATUS    RESTARTS   AGE
kubia-k99gg   1/1       Running   0          16m
```

Understanding what happened behind the scenes



Deleting deployment

```
$ kubectl delete deployment kubia
```

Deleting RC

```
$ kubectl delete rc kubia
```

Service 객체 생성

```
$ kubectl expose rc kubia --type=LoadBalancer --name kubia-http
service "kubia-http" exposed
```

Listing services

```
$ kubectl get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes   ClusterIP      10.96.0.1       <none>        443/TCP          2d
kubia-http   LoadBalancer   10.107.37.211   <pending>     8080:30701/TCP   1m
```

External IP를 통한 서비스 액세스

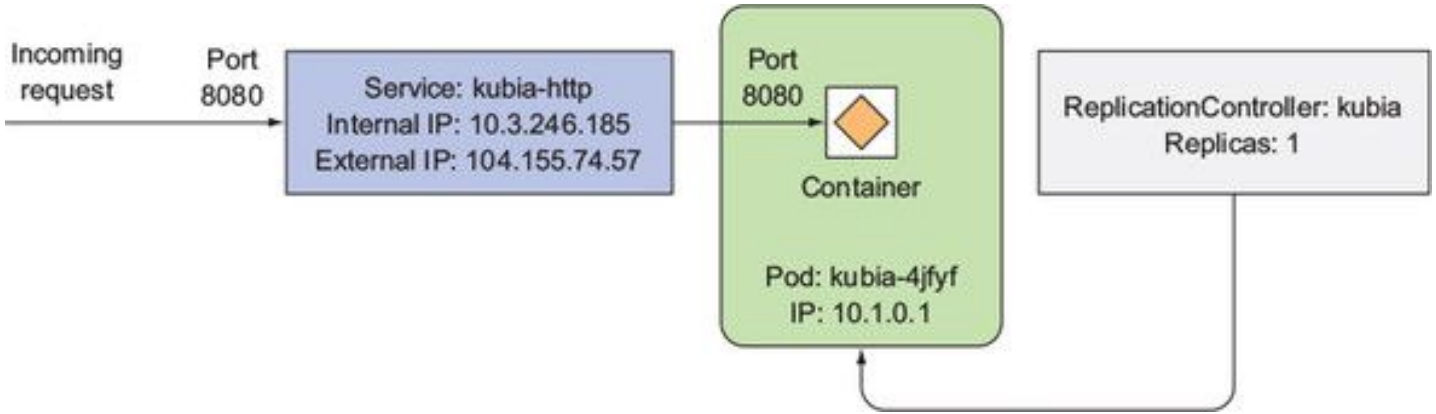
```
$ minikube service kubia-http
Opening kubernetes service default/kubia-http in default browser...
```

⇒ 서비스에 액세스할 수 있는 IP와 포트를 얻음

```
$ curl http://192.168.99.101:30701/
You've hit kubia-k99gg
```

ReplicationController, Pod, Service가 동작하는 방식의 이해

Kubectl run을 통해서 **rc시작하고,**
RC는 pod를 관리하고(복제, 누락된 포드 대체 등),
외부에서 access하려면 **pod**들을 **서비스**로 **노출**



Pod와 컨테이너 이해

- 쿠버네티스 시스템 주요 구성요소는 Pod
- Pod는 원하는 만큼 컨테이너를 포함
- 컨테이너 내에는 어플리케이션 프로세스 존재
- Pod는 고유한 private IP와 호스트 이름을 갖음

ReplicationController의 역할 이해

- **Pod를 복제**하는데 복제수를 지정하지 않으면 단일 Pod를 생성
- Pod가 누락되면 **대체 Pod**를 새로 **생성**

서비스가 필요한 이유

서비스가 필요한 이유 이해

Pod는 일시적이다. (IP 또한 일시적)

누군가 Pod를 삭제했거나, Pod가 정상 노드에서 제거되었을수 있다. RC가 손실된 Pod를 새로운 Pod로 대체하고 그러면 그 포드의 IP도 달라질 수 있다.

서비스가 생성되면 **고정IP를** 얻게되고 **서비스 수명내에 결코 변하지 않는다.**

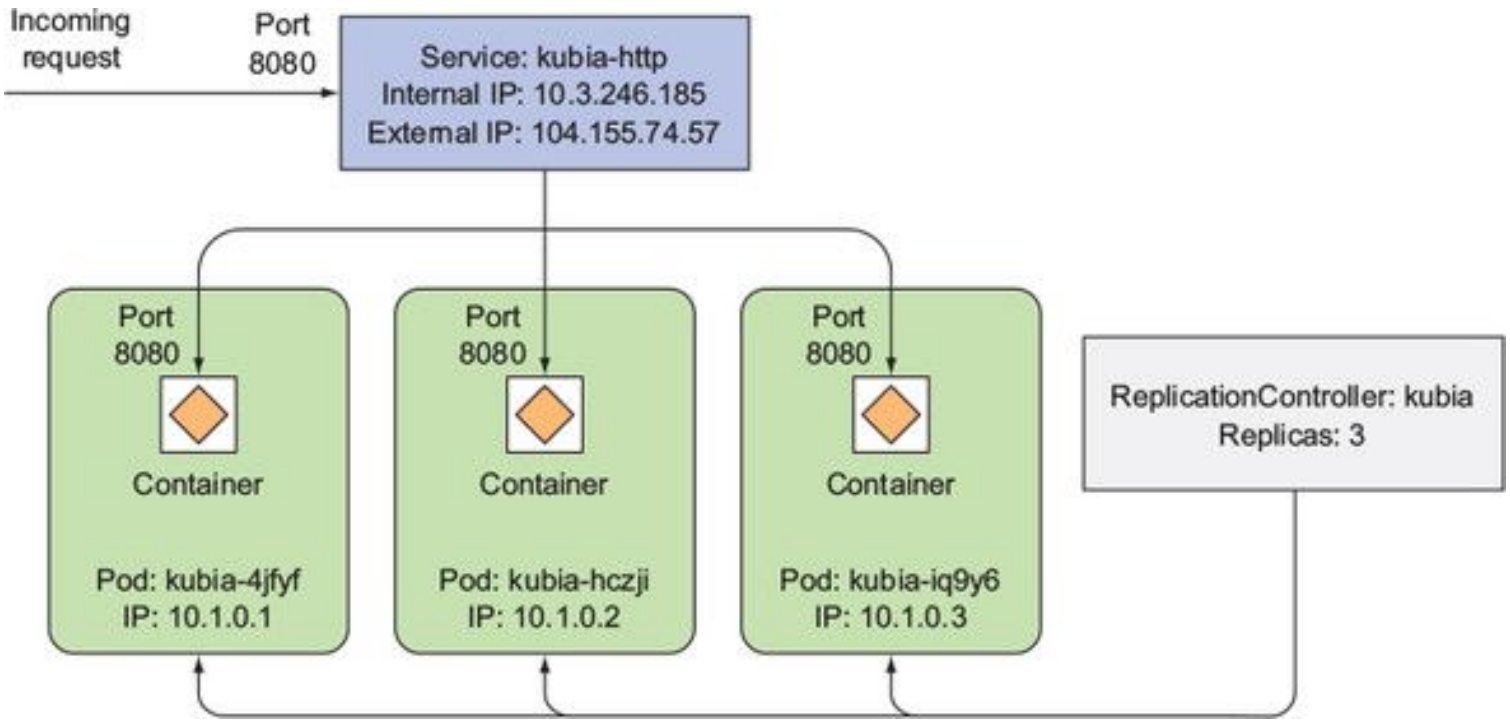
Application의 수평 스케일링

```
$ kubectl get replicationcontrollers
NAME      DESIRED   CURRENT   READY     AGE
kubia     1         1         1         33m
$ kubectl scale rc kubia --replicas=3
replicationcontroller "kubia" scaled
$ k get rc
NAME      DESIRED   CURRENT   READY     AGE
kubia     3         3         2         35m
$ kubectl get pods
NAME          READY     STATUS    RESTARTS   AGE
kubia-57zlr   1/1       Running   0          1m
kubia-k99gg   1/1       Running   0          36m
kubia-sl82g   1/1       Running   0          1m
```

서비스 요청 시 모든 포드에 요청하는지 확인

```
$ curl http://192.168.99.101:30701/
You've hit kubia-57zlr
$ curl http://192.168.99.101:30701/
You've hit kubia-k99gg
$ curl http://192.168.99.101:30701/
You've hit kubia-k99gg
$ curl http://192.168.99.101:30701/
You've hit kubia-sl82g
```

시스템의 새로운 상태 시각화



Pods를 나열할때 IP와 노드 표시하기

```
$ k get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP           NODE
kubia-57zlr   1/1     Running   0          9m    172.17.0.5   minikube
kubia-k99gg   1/1     Running   0          43m   172.17.0.7   minikube
kubia-sl82g   1/1     Running   0          9m    172.17.0.6   minikube
```

kubectl describe를 이용한 Pod세부 내용

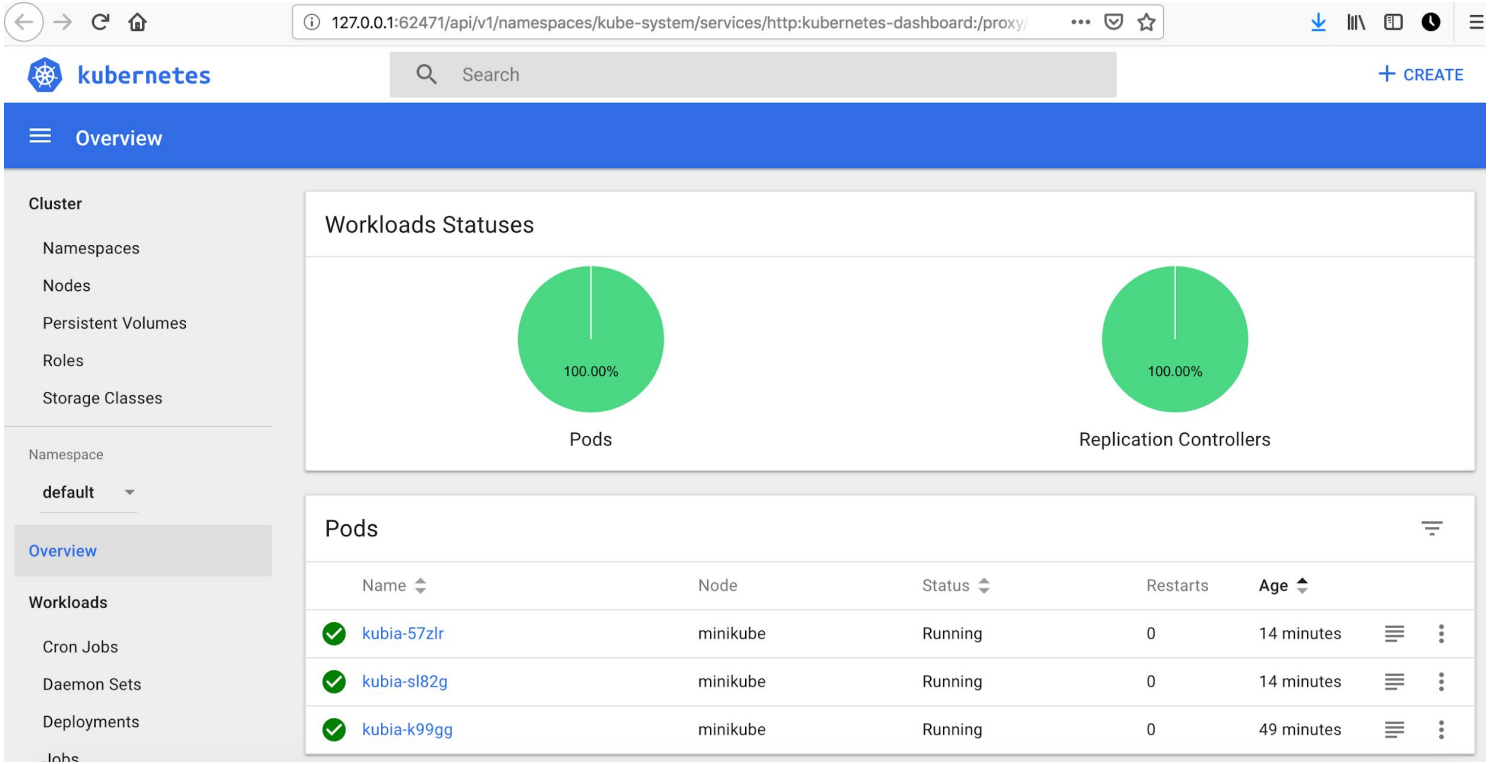
```
$ kubectl describe pod kubia-57zlr
Name:               kubia-57zlr
Namespace:          default
Priority:           0
PriorityClassName:  <none>
Node:               minikube/10.0.2.15
Start Time:         Mon, 14 Jan 2019 22:24:05 +0900
Labels:             run=kubia
Annotations:        <none>
Status:             Running
IP:                 172.17.0.5
...
```

Kubernetes dashboard 소개

```
$ kubectl cluster-info | grep dashboard
```

```
$ minikube dashboard
Opening
http://127.0.0.1:62471/api/v1/namespaces/kube-system/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```

Kubernetes dashboard



Referece

https://livebook.manning.com/#!/book/kubernetes-in-action/chapter-2/24

https://docs.docker.com/docker-for-mac/install/