Kubernetes In Action

Hands On

written by whatwant

2020.10.10

실습 기본 환경


Oracle VM VirtualBox 관리자
파일(F) 머신(M) 도움말(H)
도구
새로 만들기(N) 설정(S) 삭제 표시(H)
Ubuntu 18.04 Server (2020.01.12 21:40)
전원 꺼짐
Ubuntu 18.04 Desktop (2020.1...)
실행 중
일반
이름: Ubuntu 18.04 Desktop
운영 체제: Ubuntu (64-bit)
시스템
기본 메모리: 8192 MB
프로세서: 2
가속: VT-X/AMD-V, 네스티드 페이징, KVM 반가상화
디스플레이
비디오 메모리: 32 MB
그래픽 컨트롤러: VMSVGA
원격 데스크톱 서버: 사용 안 함
녹화: 사용 안 함
미리 보기
저장소
컨트롤러: IDE
IDE 세컨더리 마스터: [광학 드라이브] 비어 있음
컨트롤러: SATA
SATA 포트 0: Ubuntu 18.04 Desktop.vdi (일반, 50.00 GB)
오디오
호스트 드라이버: Windows DirectSound
컨트롤러: ICH AC97
네트워크
어댑터 1: Intel PRO/1000 MT Desktop(82540EM)
어댑터 2: Intel PRO/1000 MT Desktop(82540EM)

Node.js 설치 및 실습 #1

- https://nodejs.org/
- https://github.com/nodesource/distributions/blob/master/README.md#debinstall

```
$ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -
$ sudo apt-get install -y nodejs
```

- node.js 샘플 작성

```
$ nano /srv/workspace/html/index-1.js
```

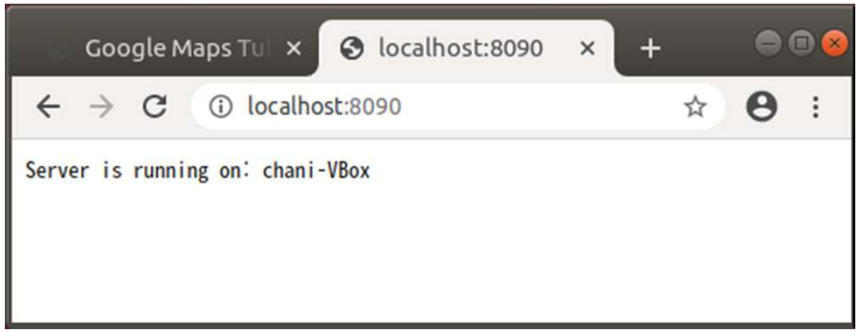
```
var http = require('http');
var os = require('os');

var host = os.hostname();

var handleRequest = function(request, response) {
    console.log('Received request for URL: ' + request.url);
    response.writeHead(200);
    response.end('Server is running on: ' + host);
};

var www = http.createServer(handleRequest);
www.listen(8090);
```

- node 실행

```
$ cd /srv/workspace/html/
$ node ./index-1.js
```

- 웹페이지 접속

http://localhost:8090/



Node.js 설치 및 실습 #2

- http-server 설치

```
$ sudo npm install http-server -g
```

- html 샘플 작성

```
$ nano /srv/workspace/html/index-1.html
```

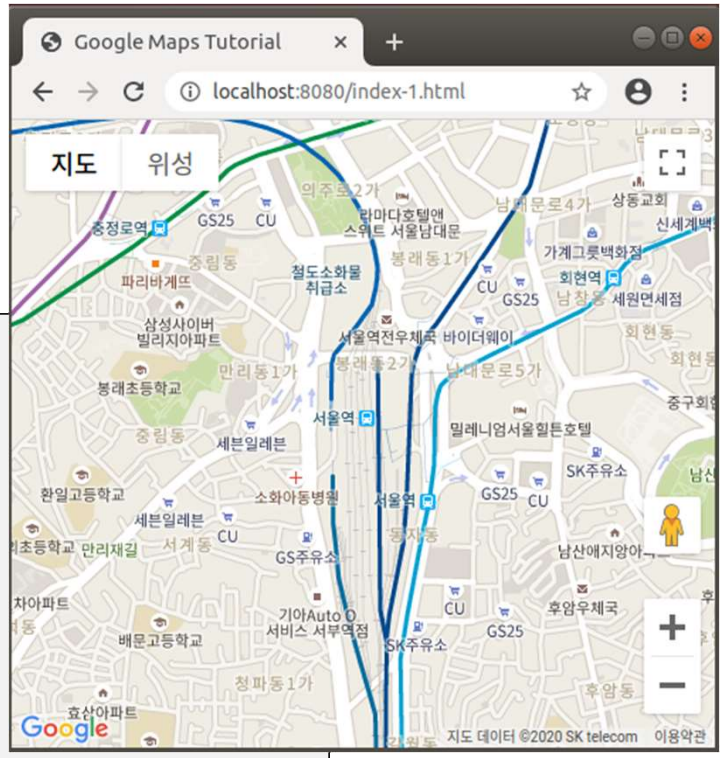
- http-server 실행

```
$ http-server /srv/workspace/html/ -p 8080
```

- 웹페이지 접속

http://localhost:8080/index-1.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Google Maps Tutorial</title>
    <style type="text/css">
      /* Always set the map height explicitly to define the size of the div
       * element that contains the map. */
      #map {
        height: 100%;
      }
      /* Makes the sample page fill the window. */
      html,
      body {
        height: 100%;
        margin: 0;
        padding: 0;
      }
    </style>
    <script>
    // Initialize and add the map
    function initMap() {
      // The location of seoul station
      var seoul = { lng: 126.9706673, lat: 37.5547787 };
      var options = {
        center: seoul,
        zoom: 15,
      };
      // The map, centered at seoul station
      var map = new google.maps.Map(document.getElementById('map'), options);
    }
    </script>
    <script async defer
    src="https://maps.googleapis.com/maps/api/js?key=AIzaSyBSYZj72nircNhp5RnkjltwEVijUMa8db4&callback=initMap">
    </script>
  </head>
  <body>
    <div id="map"></div>
  </body>
</html>
```



Docker 설치 및 실습 #1

- Ubuntu 18.04 환경에서 최신 버전 docker 설치
- 최신 버전 확인 후 Download
  - https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/

```
$ wget https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/containerd.io_1.3.7-1_amd64.deb
$ wget https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/docker-ce-cli_19.03.13~3-0~ubuntu-bionic_amd64.deb
$ wget https://download.docker.com/linux/ubuntu/dists/bionic/pool/stable/amd64/docker-ce_19.03.13~3-0~ubuntu-bionic_amd64.deb
```

- 설치

```
$ sudo dpkg --install ./containerd.io_1.3.7-1_amd64.deb
$ sudo dpkg --install ./docker-ce-cli_19.03.13\~3-0\~ubuntu-bionic_am
$ sudo dpkg --install ./docker-ce_19.03.13\~3-0\~ubuntu-bionic_amd6
```

- 현재 계정에서 바로 docker 사용할 수 있도록 권한(그룹) 설정

```
$ sudo usermod -aG docker $USER && newgrp docker
```

- hello-world

```
$ docker run hello-world
```

```
❯ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:4cf9c47f86df71d48364001ede3a4fcd85ae80ce02ebad74156906caff5378bc
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Docker 설치 및 실습 #2

- dockerfile 만들기

```
$ nano ./dockerfile1
```

```
FROM node:14.13.1
EXPOSE 8090
COPY index-1.js .
CMD node index-1.js
```

- build

```
$ docker build -t hostname:v1 -f ./dockerfile1 .
```

- 결과 확인

```
$ docker images
```

```
❯ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hostname            v1                  c64a791a11e8        30 seconds ago      943MB
node                14.13.1             f47907840247        2 days ago          943MB
hello-world         latest              bf756fb1ae65        9 months ago        13.3kB
```

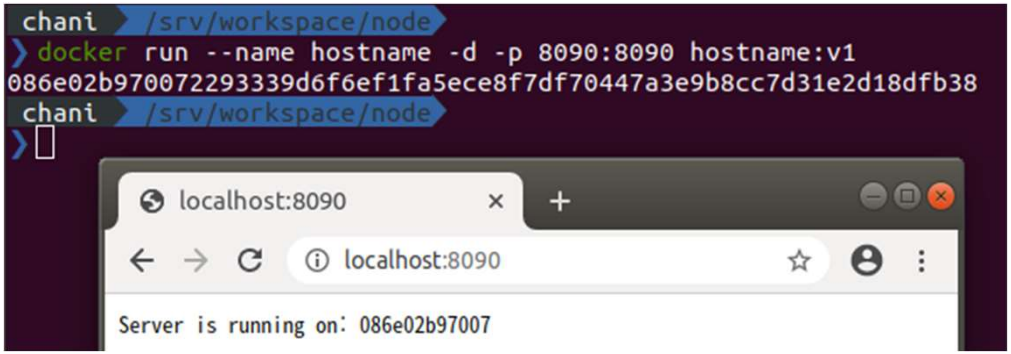
- 실행

```
$ docker run --name hostname -d -p 8090:8090 hostname:v1
```

```
chani  /srv/workspace/node
❯ docker run --name hostname -d -p 8090:8090 hostname:v1
086e02b970072293339d6f6ef1fa5ece8f7df70447a3e9b8cc7d31e2d18dfb38
chani  /srv/workspace/node
❯
```

- 웹페이지 접속

http://localhost:8090/



Minikube 설치 및 실행 #1

- VirtualBox 안에 설치한 Ubuntu 18.04 에서, docker를 이용해서 minikube 실행을 할 것이다.

- kbectl 다운로드 및 설치
- 최신 버전 확인
  - https://storage.googleapis.com/kubernetes-release/release/stable.txt

```
$ wget https://storage.googleapis.com/kubernetes-release/release/v1.19.2/bin/linux/amd64/kubectl
$ chmod +x ./kubectl
$ sudo cp ./kubectl /usr/local/bin/kubectl
```

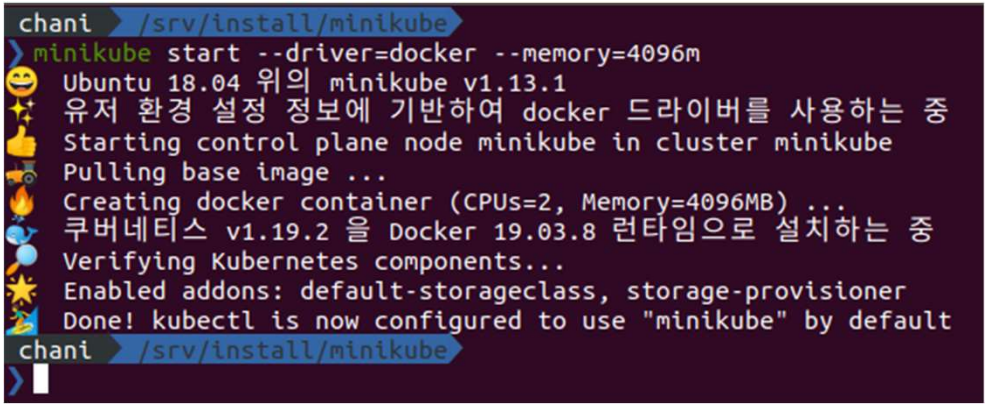
- minikube 다운로드 및 설치

```
$ wget https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 -O minikube
$ chmod +x ./minikube
$ sudo install minikube /usr/local/bin
```

Minikube 설치 및 실행 #2

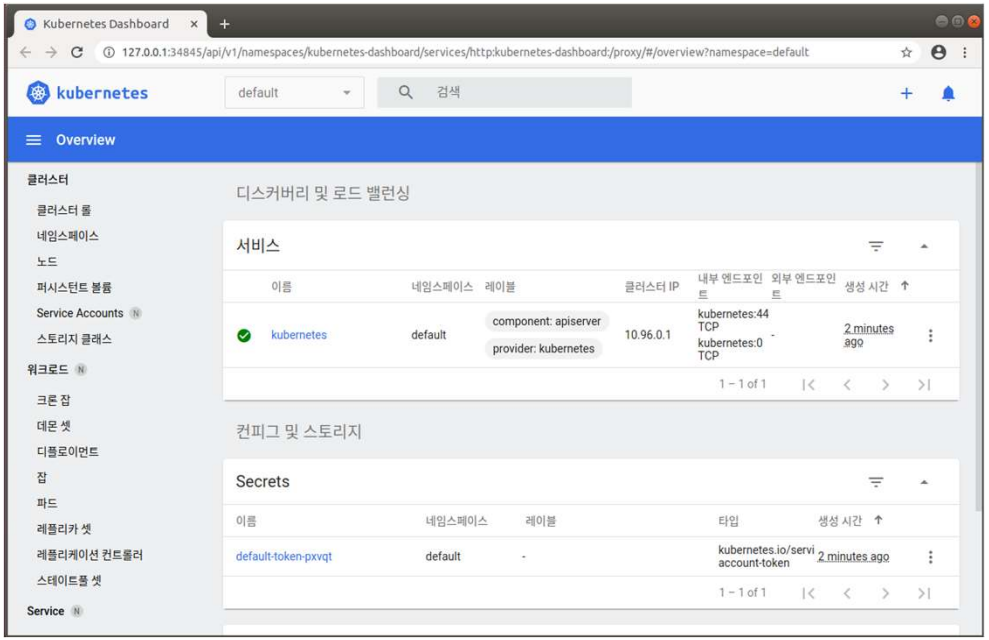
- minikube 실행

```
$ minikube start --driver=docker –memory=4096m
```



- dashboard 실행

```
$ minikube dashboard
```



Docker in Minikube #1

- minikube를 위한 Docker 환경 설정 방법 확인

$ minikube docker-env

```
❯ minikube docker-env
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://172.17.0.2:2376"
export DOCKER_CERT_PATH="/home/chani/.minikube/certs"
export MINIKUBE_ACTIVE_DOCKERD="minikube"

# To point your shell to minikube's docker-daemon, run:
# eval $(minikube -p minikube docker-env)
```

- 나오는 가이드에 따라 실행

$ eval $(minikube -p minikube docker-env)

```
❯ docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
hostname                    v1                  c64a791a11e8        5 days ago          943MB
node                        14.13.1             f47907840247        7 days ago          943MB
gcr.io/k8s-minikube/kicbase v0.0.12-snapshot3   25ac91b9c8d7        7 weeks ago         952MB
hello-world                 latest              bf756fb1ae65        9 months ago        13.3kB
```

- minikube를 위한 환경임을 확인해볼 수 있다

$ docker images

```
❯ docker images
REPOSITORY                                  TAG                 IMAGE ID            CREATED             SIZE
k8s.gcr.io/kube-proxy                       v1.19.2             d373dd5a8593        4 weeks ago         118MB
k8s.gcr.io/kube-apiserver                   v1.19.2             607331163122        4 weeks ago         119MB
k8s.gcr.io/kube-controller-manager          v1.19.2             8603821e1a7a        4 weeks ago         111MB
k8s.gcr.io/kube-scheduler                   v1.19.2             2f32d66b884f        4 weeks ago         45.7MB
gcr.io/k8s-minikube/storage-provisioner     v3                  bad58561c4be        6 weeks ago         29.7MB
k8s.gcr.io/etcd                             3.4.13-0            0369cf4303ff        7 weeks ago         253MB
kubernetesui/dashboard                      v2.0.3              503bc4b7440b        3 months ago        225MB
k8s.gcr.io/coredns                          1.7.0               bfe3a36ebd25        3 months ago        45.2MB
kubernetesui/metrics-scraper                v1.0.4              86262685d9ab        6 months ago        36.9MB
k8s.gcr.io/pause                            3.2                 80d28bedfe5d        8 months ago        683kB
```

Docker in Minikube #2

- 앞에서 만들었던 Dockerfile을 이용해서 똑같이 build 수행

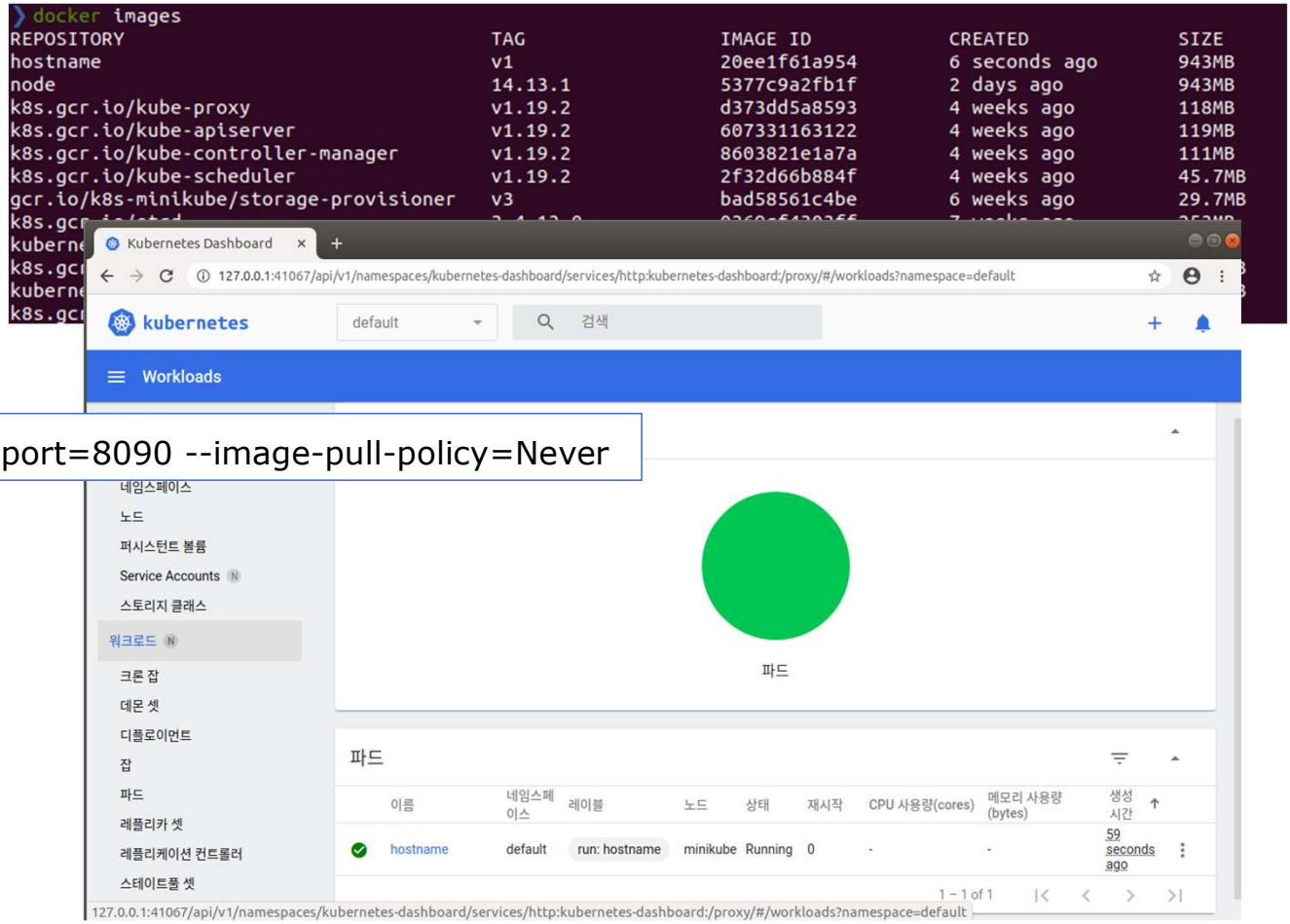
```
$ docker build -t hostname:v1 -f ./dockerfile1 .
```

- 결과 확인

```
$ docker images
```

- Kubernetes에서 실행을 해보자.

```
$ kubectl run hostname --image=hostname:v1 --port=8090 --image-pull-policy=Never
```



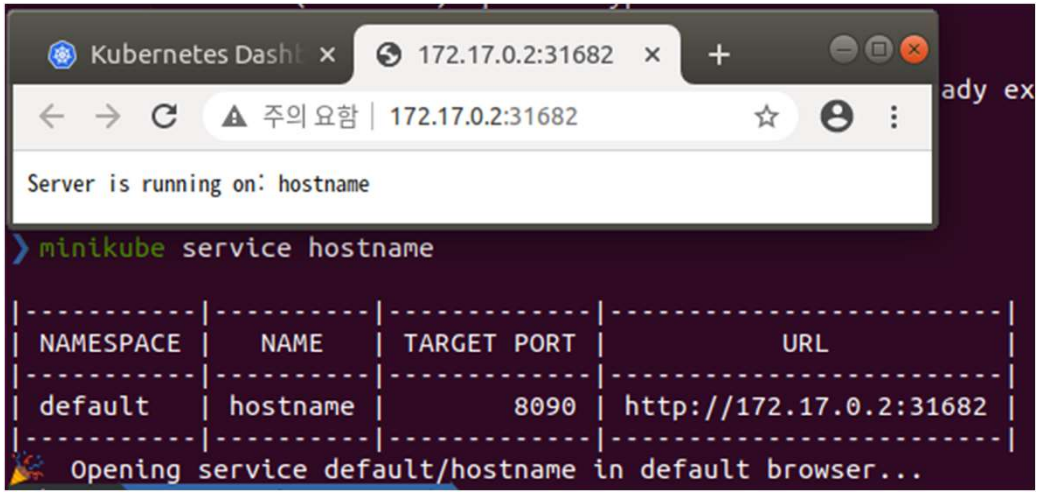
Kubernetes 맛보기 #1

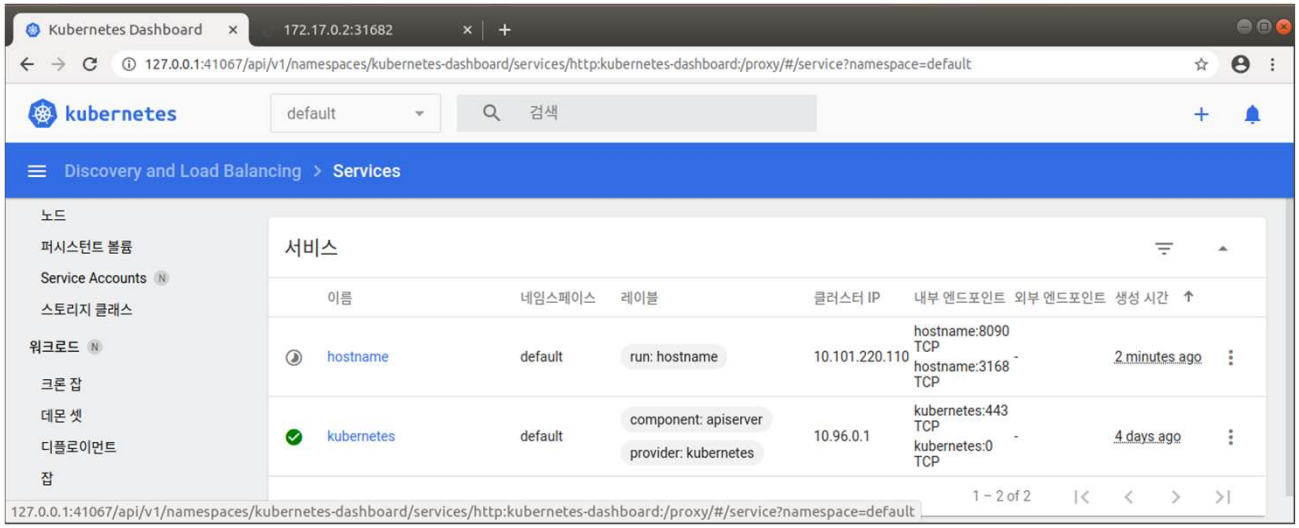
- 앞에서 생성한 Pod를 노출하기

```
$ kubectl expose pod hostname --type=LoadBalancer
```

- 확인

```
$ minikube service hostname
```





Kubernetes 맛보기 #2

- Deployment 구성하기

```
$ kubectl create deployment hostname --image=hostname:v1
```

- Scale 3으로 변경

- 기존 Service 삭제

- Service 신규로 생성

```
$ kubectl expose deployment hostname --port=8090 --type=LoadBalancer
```

- 확인

```
$ minikube service hostname
```