Managing Kubernetes

2021-04-23
written by whatwant

Agenda

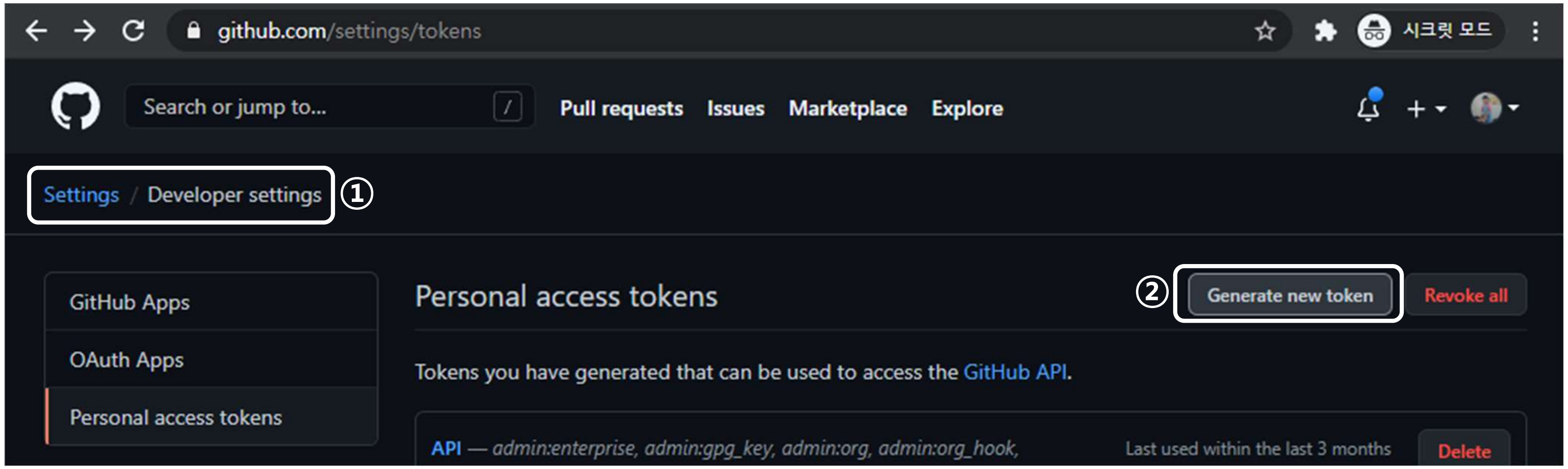
※ 참고 : https://home.modulabs.co.kr/product/managing-kubernetes/

2 week
Environment & POD

Homework

Container Registry in GitHub – 1/5

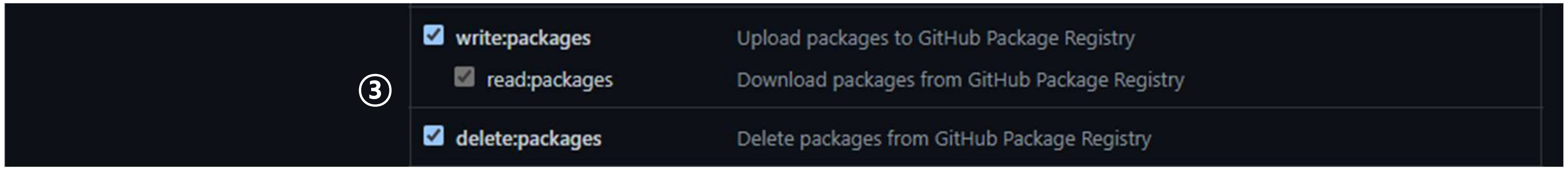
- GitHub Registry Server에 image를 업로드 하기 위해서, access token을 발급받자
- 권한은 'packages' 관련된 것들만 있으면 된다.



Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

API — *admin:enterprise, admin:gpg_key, admin:org, admin:org_hook,* Last used within the last 3 months

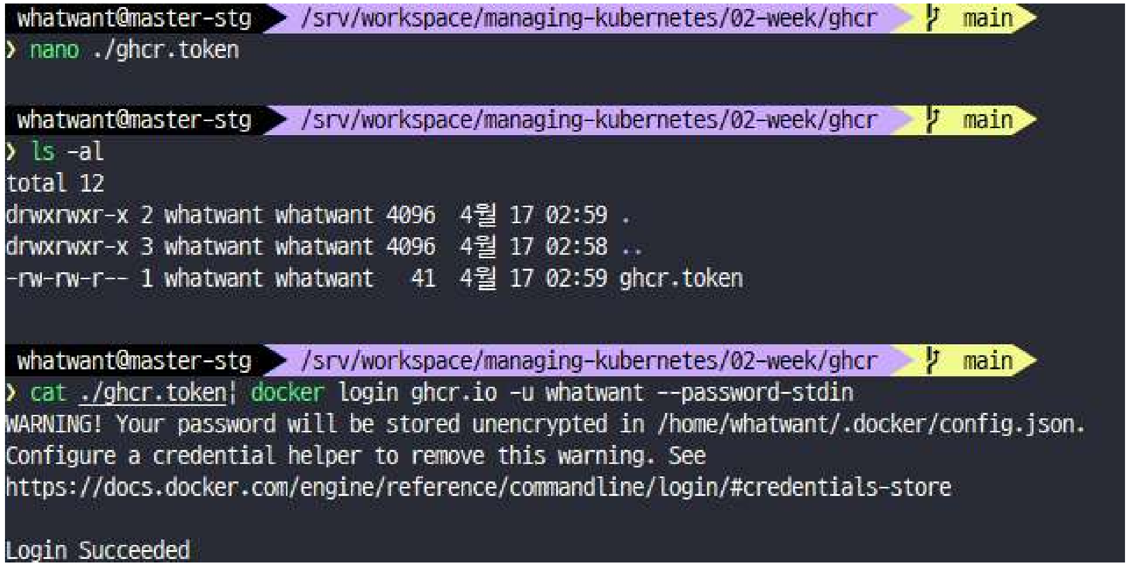


✓ write:packages	Upload packages to GitHub Package Registry
✓ read:packages	Download packages from GitHub Package Registry
✓ delete:packages	Delete packages from GitHub Package Registry

※ 참고 : https://docs.github.com/en/packages/guides/about-github-container-registry

Container Registry in GitHub – 2/5

- 발급 받은 token을 text로 저장 (반드시 이렇게 할 필요는 없지만, 사용 편의를 위해서…)
- GitHub Container Registry(ghcr.io)에 로그인



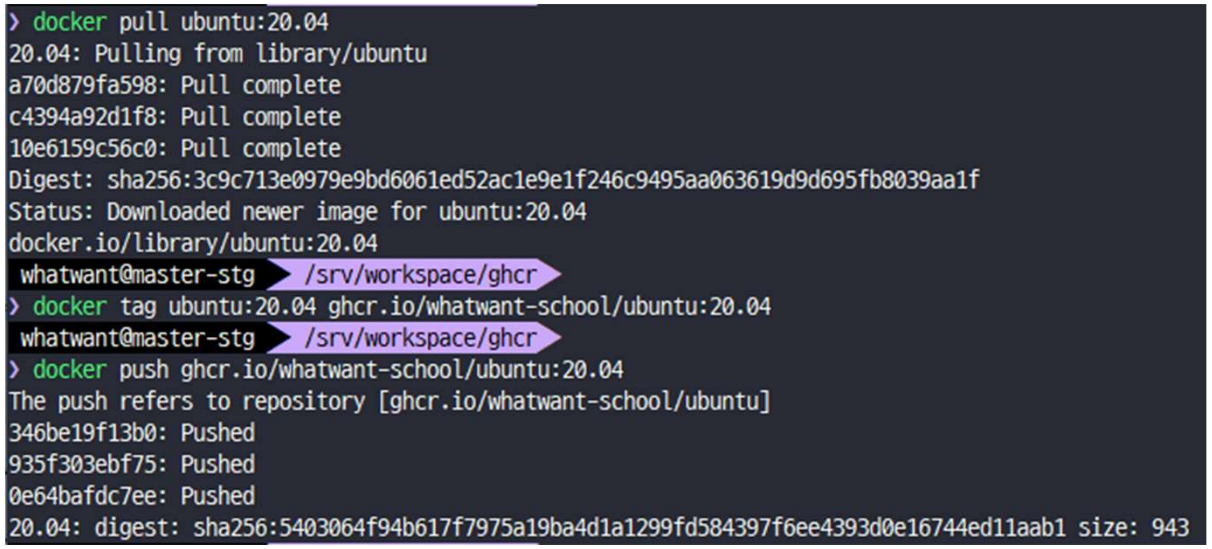
```
whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/ghcr  main
❯ nano ./ghcr.token

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/ghcr  main
❯ ls -al
total 12
drwxrwxr-x 2 whatwant whatwant 4096  4월 17 02:59 .
drwxrwxr-x 3 whatwant whatwant 4096  4월 17 02:58 ..
-rw-rw-r-- 1 whatwant whatwant   41  4월 17 02:59 ghcr.token

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/ghcr  main
❯ cat ./ghcr.token| docker login ghcr.io -u whatwant --password-stdin
WARNING! Your password will be stored unencrypted in /home/whatwant/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
> cat ./ghcr.token | docker login ghcr.io -u <USERNAME> --password-stdin
```

※ 참고 : https://www.44bits.io/ko/post/news--github-container-registry-beta-release

Container Registry in GitHub – 3/5

- 이미지 확보 → tagging → push
- 사용자 계정 또는 organization 기준으로 package upload



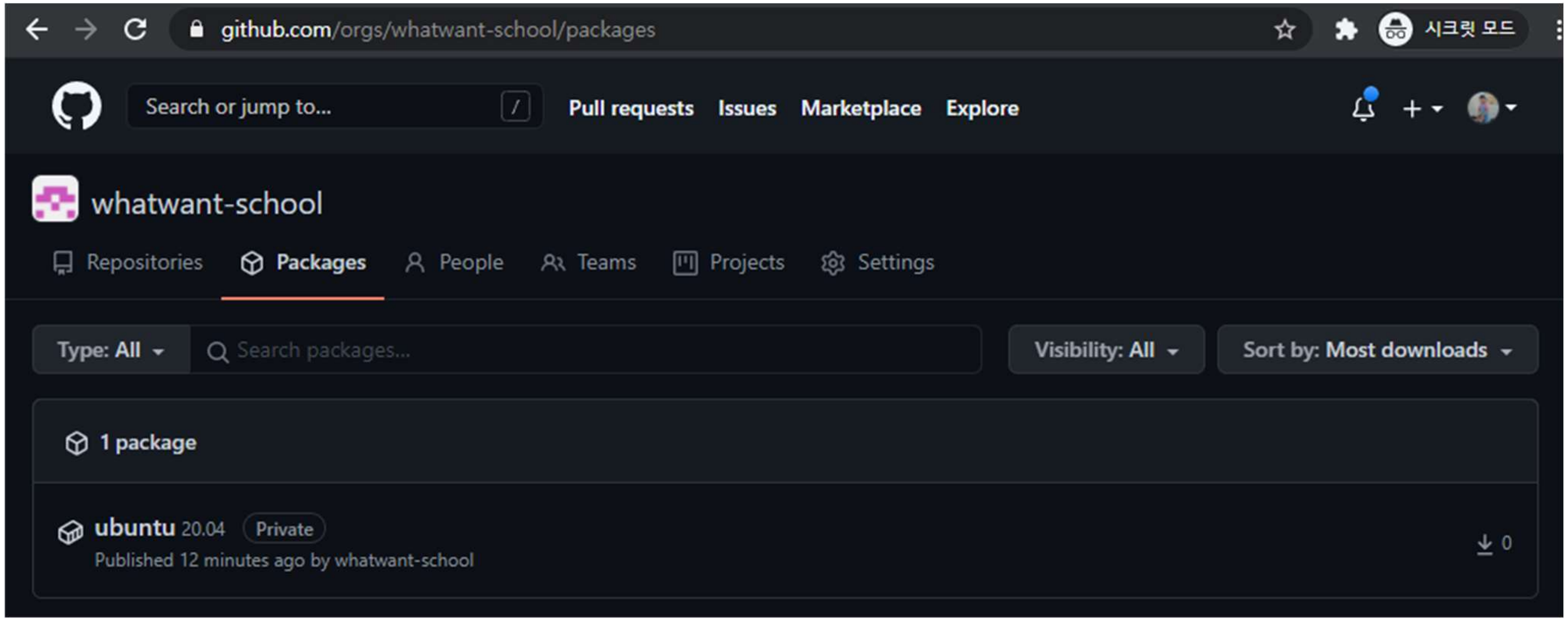
```
❯ docker pull ubuntu:20.04
20.04: Pulling from library/ubuntu
a70d879fa598: Pull complete
c4394a92d1f8: Pull complete
10e6159c56c0: Pull complete
Digest: sha256:3c9c713e0979e9bd6061ed52ac1e9e1f246c9495aa063619d9d695fb8039aa1f
Status: Downloaded newer image for ubuntu:20.04
docker.io/library/ubuntu:20.04
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker tag ubuntu:20.04 ghcr.io/whatwant-school/ubuntu:20.04
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker push ghcr.io/whatwant-school/ubuntu:20.04
The push refers to repository [ghcr.io/whatwant-school/ubuntu]
346be19f13b0: Pushed
935f303ebf75: Pushed
0e64bafdc7ee: Pushed
20.04: digest: sha256:5403064f94b617f7975a19ba4d1a1299fd584397f6ee4393d0e16744ed11aab1 size: 943
```

```
> docker tag ubuntu:20.04 ghcr.io/<user | org>/<image name:tag>
> docker push ghcr.io/<user | org>/<image name:tag>
```

※ 참고 : https://www.44bits.io/ko/post/news--github-container-registry-beta-release

Container Registry in GitHub – 4/5

- user/organization 메뉴를 보면 `Packages` 확인 가능



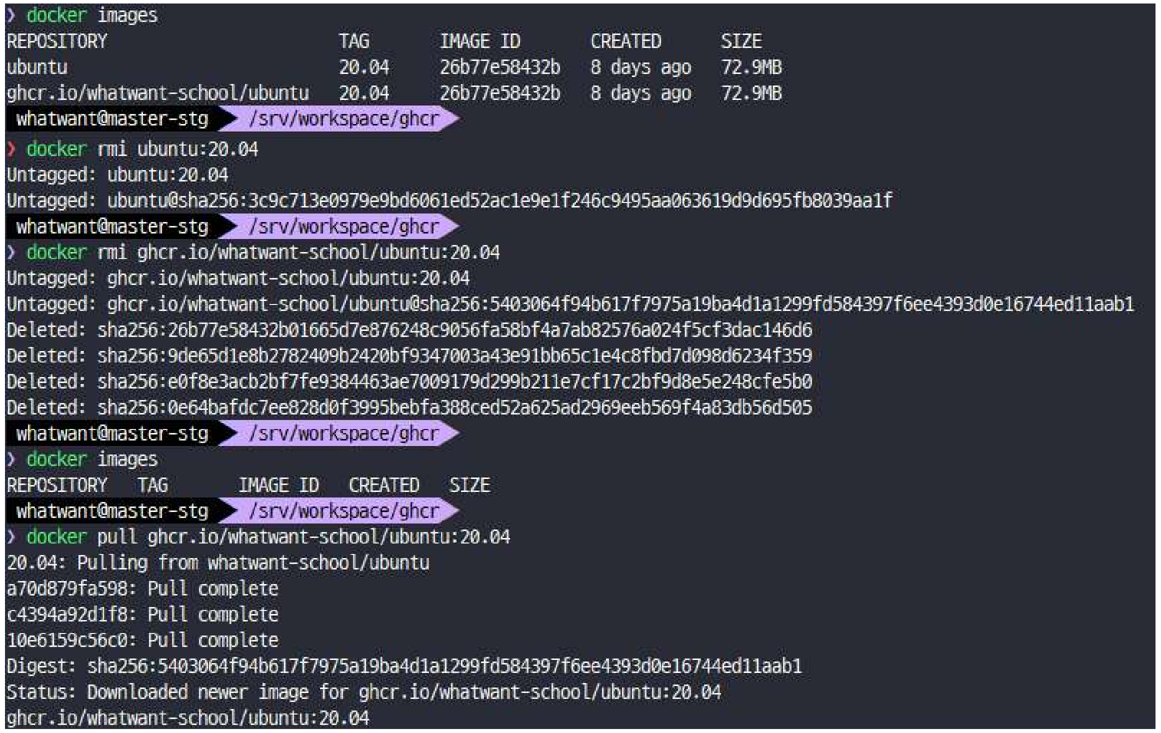
1 package

ubuntu 20.04 Private
Published 12 minutes ago by whatwant-school
0

※ 참고 : https://www.44bits.io/ko/post/news--github-container-registry-beta-release

Container Registry in GitHub – 5/5

- `docker pull` 해보기
- local에 있는 image 삭제하고 ghcr.io에서 내려 받기



```
❯ docker images
REPOSITORY                     TAG       IMAGE ID       CREATED      SIZE
ubuntu                         20.04     26b77e58432b   8 days ago   72.9MB
ghcr.io/whatwant-school/ubuntu 20.04     26b77e58432b   8 days ago   72.9MB
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker rmi ubuntu:20.04
Untagged: ubuntu:20.04
Untagged: ubuntu@sha256:3c9c713e0979e9bd6061ed52ac1e9e1f246c9495aa063619d9d695fb8039aa1f
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker rmi ghcr.io/whatwant-school/ubuntu:20.04
Untagged: ghcr.io/whatwant-school/ubuntu:20.04
Untagged: ghcr.io/whatwant-school/ubuntu@sha256:5403064f94b617f7975a19ba4d1a1299fd584397f6ee4393d0e16744ed11aab1
Deleted: sha256:26b77e58432b01665d7e876248c9056fa58bf4a7ab82576a024f5cf3dac146d6
Deleted: sha256:9de65d1e8b2782409b2420bf9347003a43e91bb65c1e4c8fbd7d098d6234f359
Deleted: sha256:e0f8e3acb2bf7fe9384463ae7009179d299b211e7cf17c2bf9d8e5e248cfe5b0
Deleted: sha256:0e64bafdc7ee828d0f3995bebfa388ced52a625ad2969eeb569f4a83db56d505
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
 whatwant@master-stg  /srv/workspace/ghcr
❯ docker pull ghcr.io/whatwant-school/ubuntu:20.04
20.04: Pulling from whatwant-school/ubuntu
a70d879fa598: Pull complete
c4394a92d1f8: Pull complete
10e6159c56c0: Pull complete
Digest: sha256:5403064f94b617f7975a19ba4d1a1299fd584397f6ee4393d0e16744ed11aab1
Status: Downloaded newer image for ghcr.io/whatwant-school/ubuntu:20.04
ghcr.io/whatwant-school/ubuntu:20.04
```

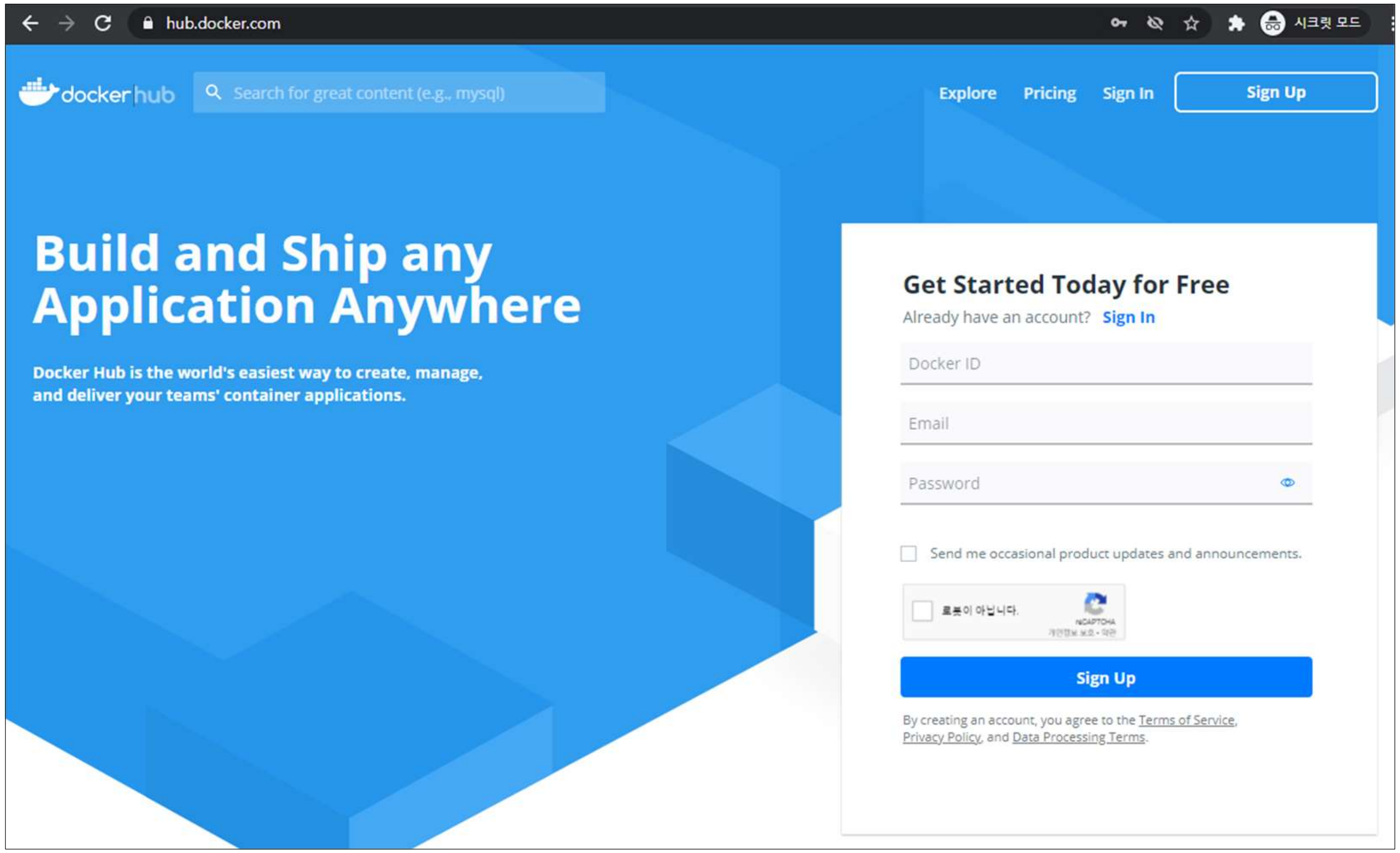
```
> docker rmi ghcr.io/<user | org>/<image name:tag>
> docker pull ghcr.io/<user | org>/<image name:tag>
```

Supplementary Lessons

Docker Hub – 1/6

- `docker hub`에 image를 업로드하기 위해서는 일단 사이트에 가입 필수

  . https://hub.docker.com/



Build and Ship any Application Anywhere

Docker Hub is the world's easiest way to create, manage, and deliver your teams' container applications.

Get Started Today for Free

Already have an account? Sign In

Docker ID

Email

Password

Send me occasional product updates and announcements.

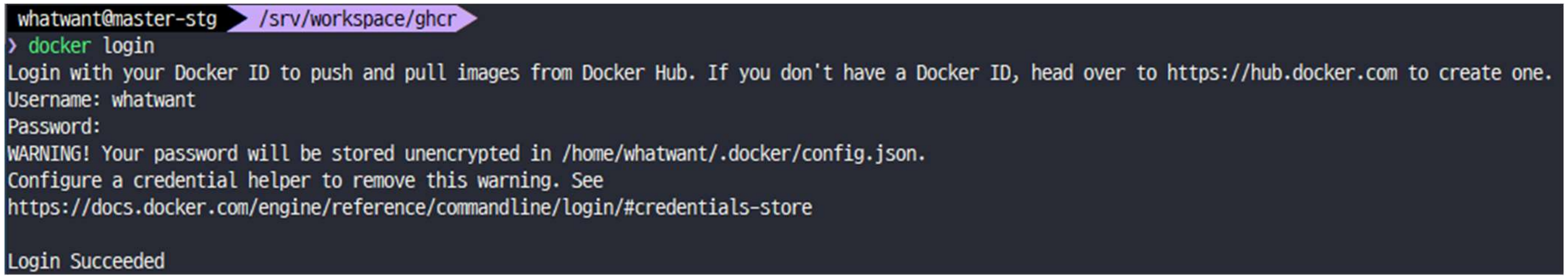
로봇이 아닙니다.

Sign Up

By creating an account, you agree to the Terms of Service, Privacy Policy, and Data Processing Terms.

Docker Hub – 2/6

- 업로드를 위한 권한을 얻기 위해서는 `docker login` 필요



```
whatwant@master-stg  /srv/workspace/ghcr
❯ 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: whatwant
Password:
WARNING! Your password will be stored unencrypted in /home/whatwant/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

```
> docker login
```

Docker Hub – 3/6

- 업로드 할 image를 하나 만들자

  . 파일 2개 만들고 build 까지 해보자

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>WHATWANT</title>
</head>
<body>
  <h2>Hello</h2>
  <h4>This is WHATWANT SCHOOL</h4>
</body>
</html>
```

Dockerfile-whatwant

※ `Dockerfile`이 아닌 이름으로 저장된 경우에 대해서도 보여주기 위해 파일 이름을 이렇게 작성해보았다.

```
FROM nginx:latest
COPY ./index.html  /usr/share/nginx/html/index.html
```

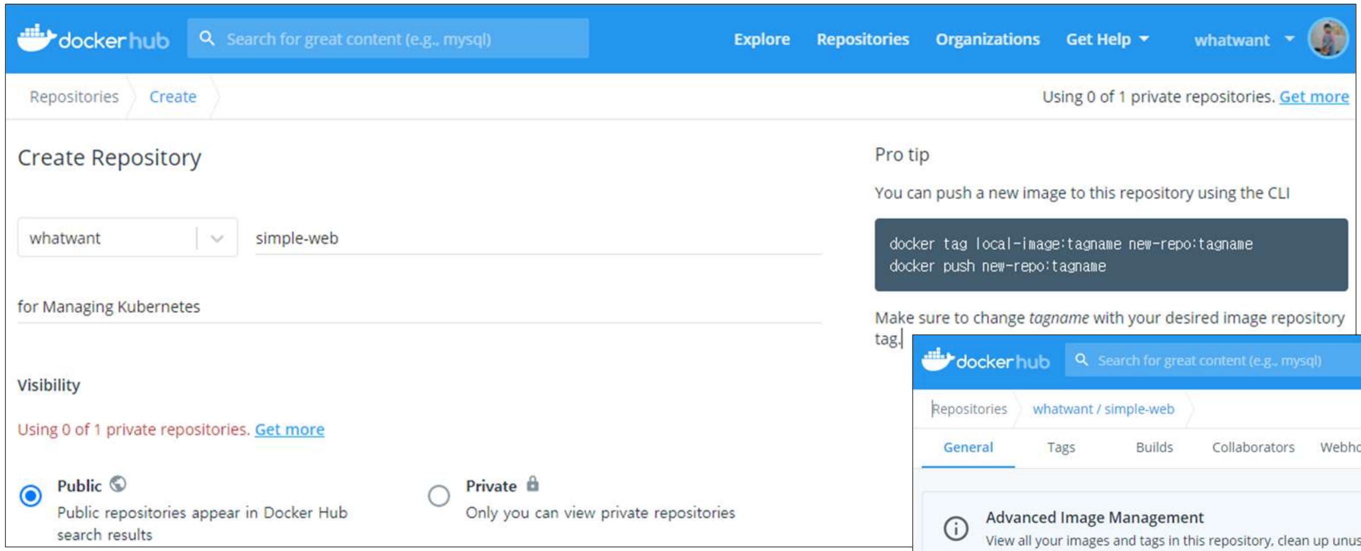
```
> docker build -t simple-web -f Dockerfile-whatwant .
```

※ 항상 제일 뒤의 `.`을 주의해야 한다.

Docker Hub – 4/6

- image를 업로드 할 Docker Hub Repository를 생성하자

. image 이름으로 생성해주면 된다 (선택 사항)



Using 0 of 1 private repositories. Get more

Create Repository

whatwant / simple-web

for Managing Kubernetes

Visibility

Using 0 of 1 private repositories. Get more

[✓ Public] Public repositories appear in Docker Hub search results

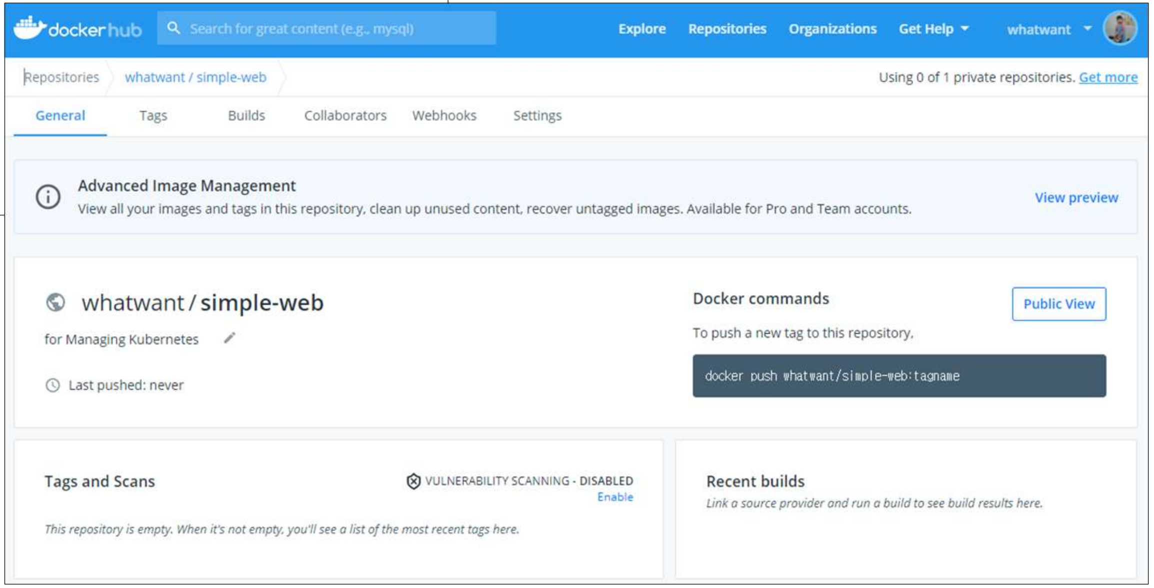
Private: Only you can view private repositories

Pro tip

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.



Using 0 of 1 private repositories. Get more

Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

whatwant / simple-web

for Managing Kubernetes

Last pushed: never

Docker commands

To push a new tag to this repository,

```
docker push whatwant/simple-web:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

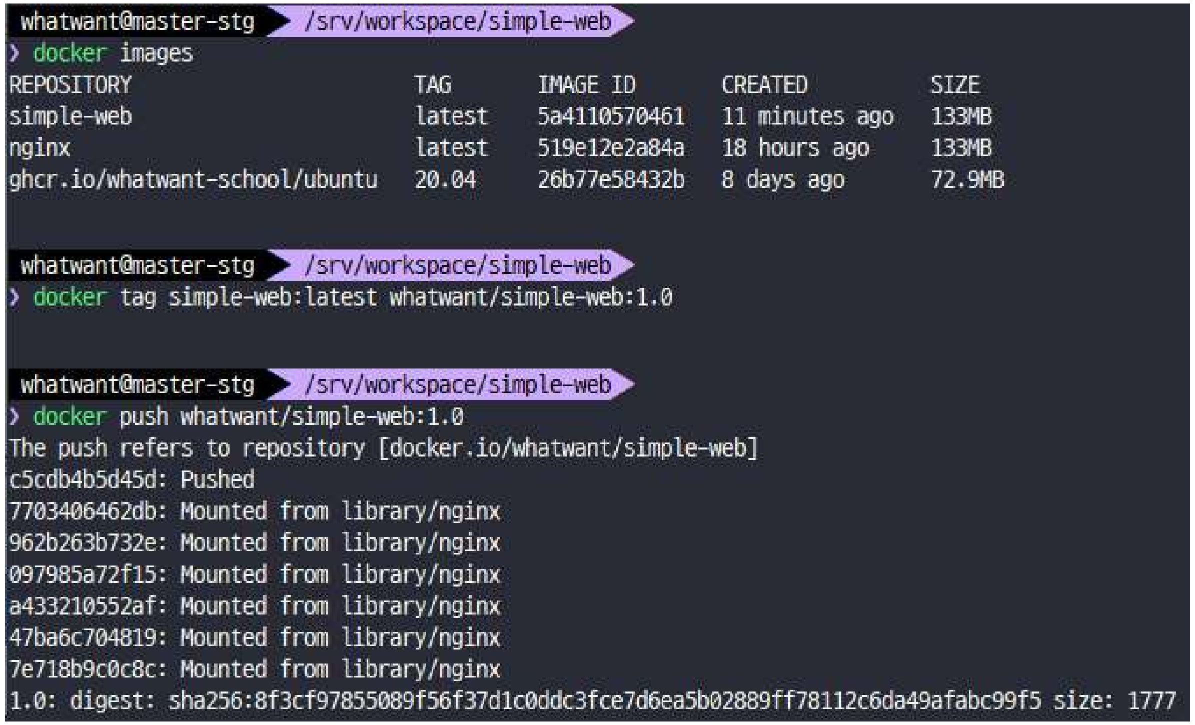
This repository is empty. When it's not empty, you'll see a list of the most recent tags here.

Recent builds

Link a source provider and run a build to see build results here.

Docker Hub – 5/6

- 업로드 하기 위한 tagging 작업 후 push 하자.

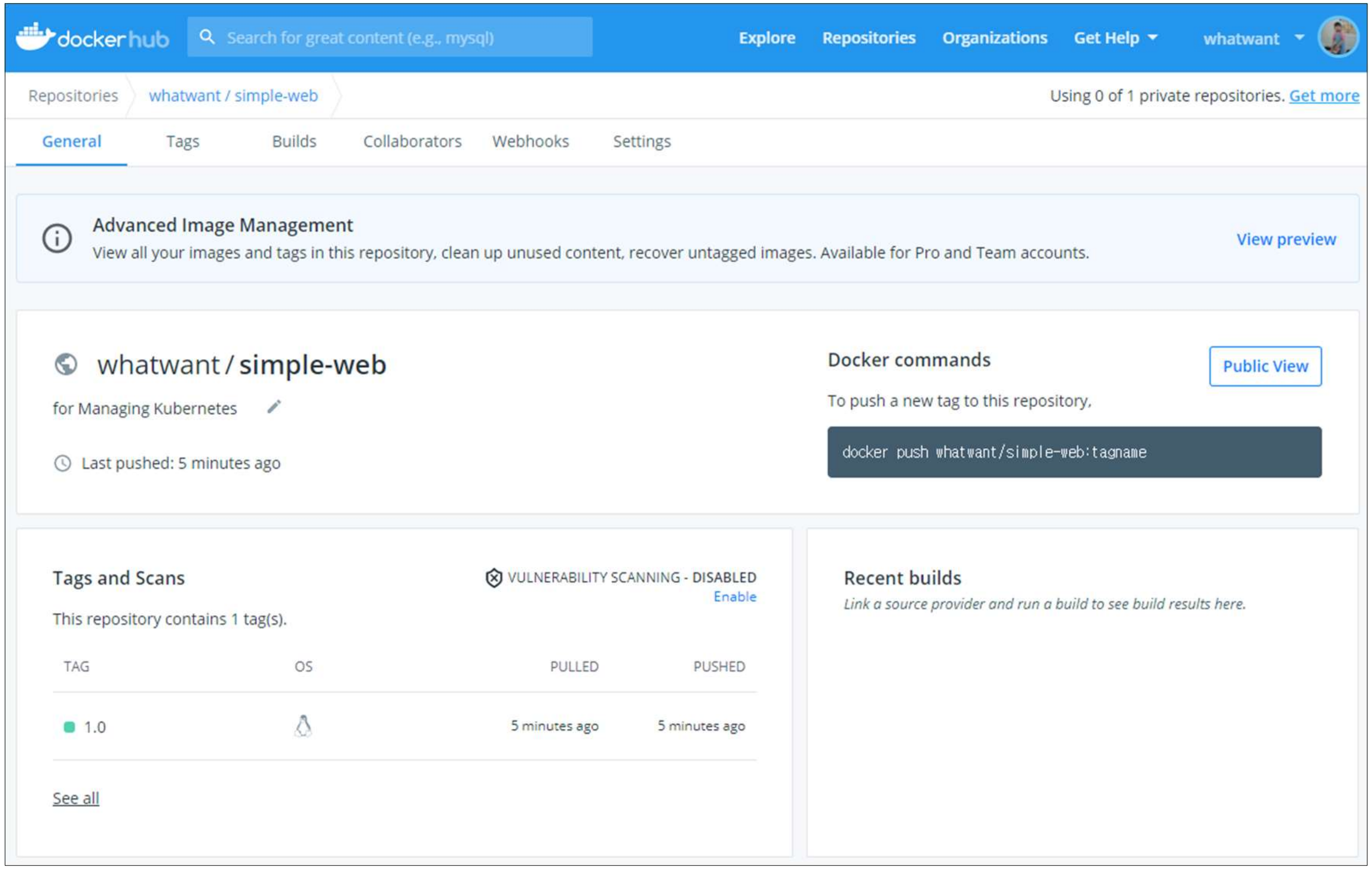


```
whatwant@master-stg  /srv/workspace/simple-web
❯ docker images
REPOSITORY                        TAG       IMAGE ID       CREATED          SIZE
simple-web                        latest    5a4110570461   11 minutes ago   133MB
nginx                             latest    519e12e2a84a   18 hours ago     133MB
ghcr.io/whatwant-school/ubuntu    20.04     26b77e58432b   8 days ago       72.9MB

whatwant@master-stg  /srv/workspace/simple-web
❯ docker tag simple-web:latest whatwant/simple-web:1.0

whatwant@master-stg  /srv/workspace/simple-web
❯ docker push whatwant/simple-web:1.0
The push refers to repository [docker.io/whatwant/simple-web]
c5cdb4b5d45d: Pushed
7703406462db: Mounted from library/nginx
962b263b732e: Mounted from library/nginx
097985a72f15: Mounted from library/nginx
a433210552af: Mounted from library/nginx
47ba6c704819: Mounted from library/nginx
7e718b9c0c8c: Mounted from library/nginx
1.0: digest: sha256:8f3cf97855089f56f37d1c0ddc3fce7d6ea5b02889ff78112c6da49afabc99f5 size: 1777
```

```
> docker tag <local-image>:<tagname>  <hub-image>:<tagname>
> docker push <hub-image>:<tagname>
```

Docker Hub – 6/6

- 업로드 된 image 확인



Advanced Image Management

View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

whatwant / simple-web

for Managing Kubernetes

Last pushed: 5 minutes ago

Docker commands

To push a new tag to this repository,

```
docker push whatwant/simple-web:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
🟩 1.0	🐧	5 minutes ago	5 minutes ago

Recent builds

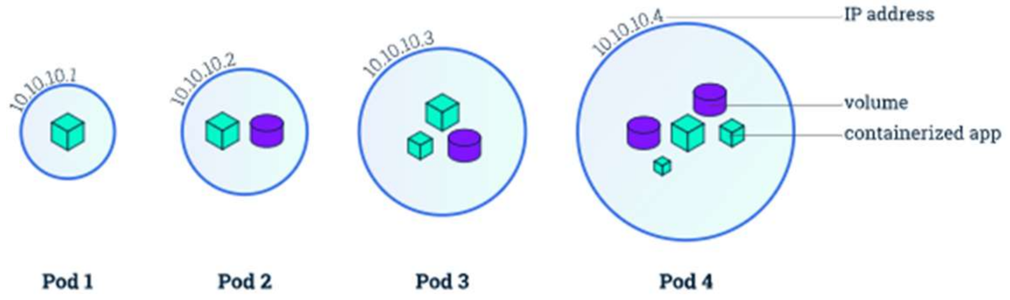
Link a source provider and run a build to see build results here.

Kubernetes – Pod

Pod is …

Pod는 Kubernetes에서 생성하고 관리할 수 있는 **배포 가능한 가장 작은 컴퓨팅 단위**이다.

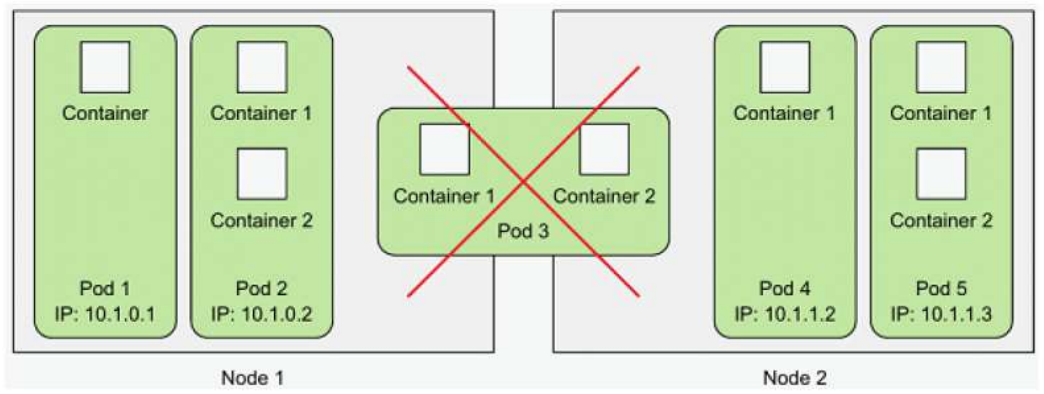
Pod는 **하나 이상의 컨테이너 그룹**이다.



※ 참고 : https://kubernetes.io/ko/docs/tutorials/kubernetes-basics/explore/explore-intro/

Pod is …

- Pod는 함께 배치된 Container 그룹을 의미
- Container는 단일 프로세스를 실행하는 것을 목적으로 설계
- 따라서, 여러 Container를 묶고 하나의 단위로 관리할 수 있는 상위 구조가 필요 → Pod
- Kubernetes는 Pod 단위로 배포하고 운영



▲ 그림 3.1 파드 안에 있는 모든 컨테이너는 같은 노드에서 실행된다. 절대로 두 노드에 걸쳐 배포되지 않는다.

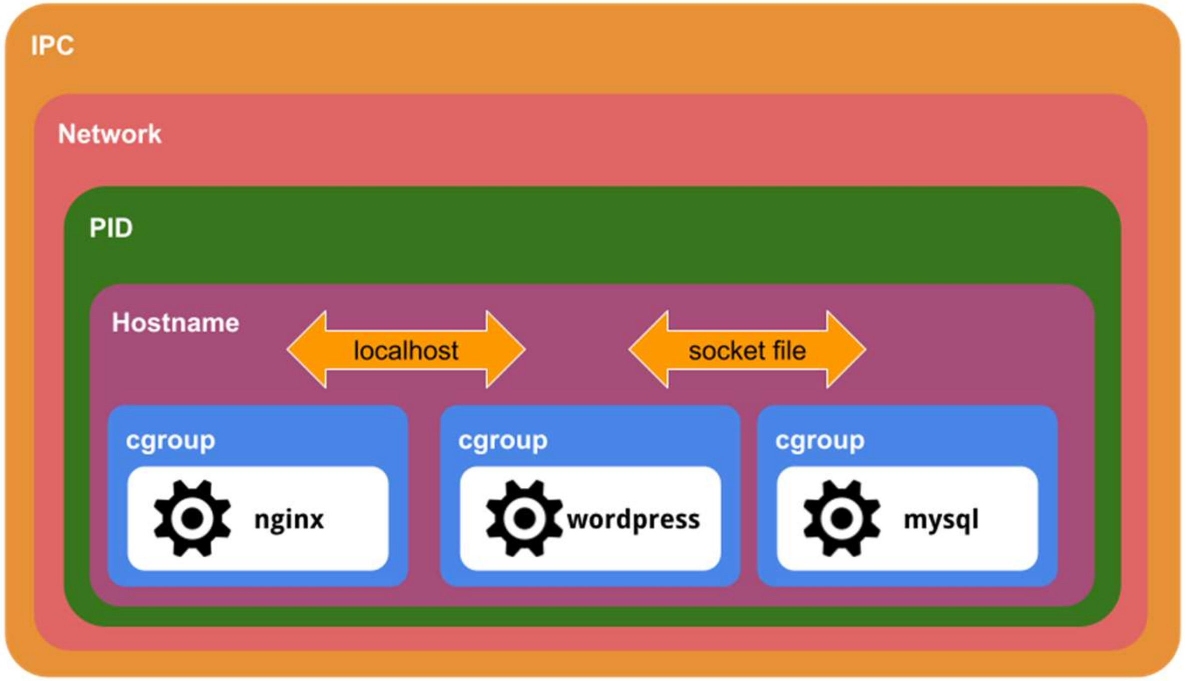
※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-3/10

Pod is …

Pod 1개 안에서 여러 개의 Container가 실행되는 것은 단순하게 각 프로세스가 동일한 머신 위에서 실행한다고 생각하면 된다.

이들 프로세스는 localhost(127.0.0.1)로 네트워크 통신을 할 수 있으며, 볼륨에 있는 파일을 공유 할 수 있다.

또는 IPC(Inter Process Communication)를 이용하거나 HUP, TERM 시그널을 보낼 수도 있다.

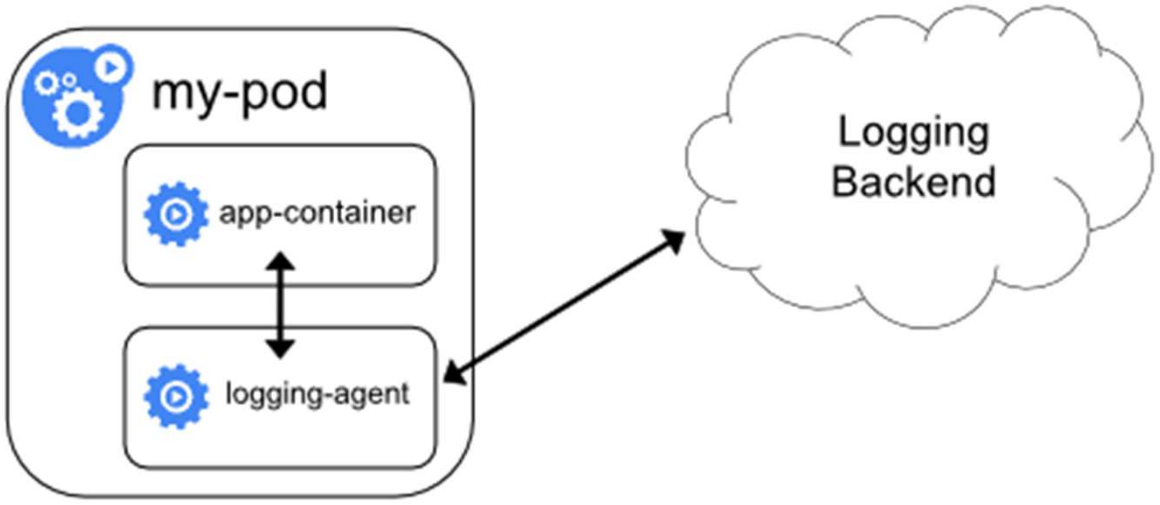


※ 참고 : https://www.joinc.co.kr/w/man/12/kubernetes/Pod

Sidecar pattern

사이드카 패턴(Sidecar Pattern)

- 기본 컨테이너의 기능을 확장하거나 강화하는 용도의 컨테이너를 추가하는 패턴
- 기본 컨테이너에는 원래 목적의 기능에만 충실하고 나머지 부가적인 공통 기능들은 사이드카 컨테이너를 추가해서 사용



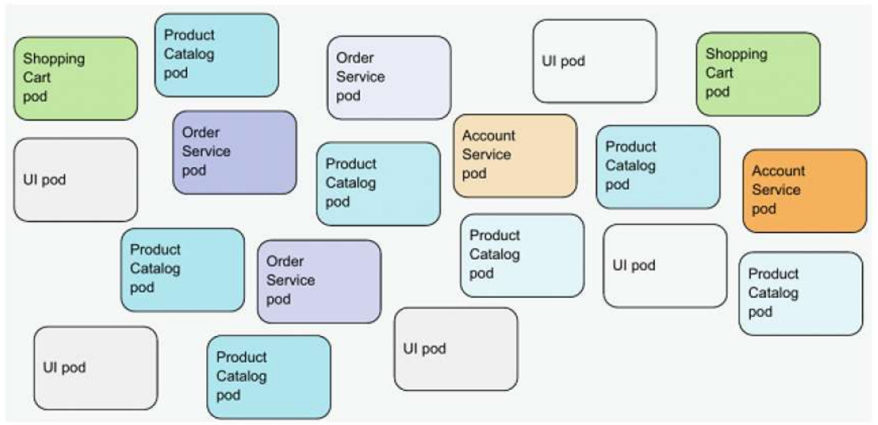
※ 참고 : https://arisu1000.tistory.com/27863

label

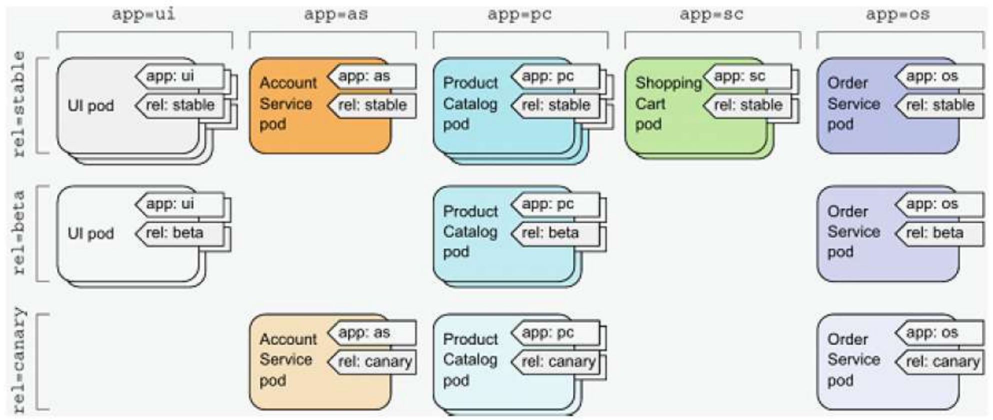
- Label은 Kubernetes 리소스를 분류할 수 있는 기능

- 각 오브젝트는 하나 이상의 레이블을 가질 수 있으며 label은 Key-Value Pair로 이루어짐

- Kubernetes 명령어에서 동일한 label을 가진 오브젝트를 선택할 수 있음



▲ 그림 3.6 마이크로서비스 아키텍처 안에 있는 분류되지 않는 파드



▲ 그림 3.7. 파드 레이블을 이용해 마이크로서비스 아키텍처 안에 파드를 조직화했다.

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-3/176

Kubernetes - Pod 실습

Create Image

- Pod 실습을 위한 image 하나 만들어서 Docker Hub에 업로드 해보자

※ 이 부분은 k8s 없이, docker만 설치되어 있는 곳이면 어디서든 가능하다.　※ 그냥 이미 올려놓은 것을 사용할 것이라면 생략 가능하다.

app.js

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

console.log("node-web 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);
```

Dockerfile

```
FROM node:latest
ADD app.js /app.js
ENTRYPOINT ["node", "app.js"]
```

```
> docker build -t node-web:1.0 .
> docker tag node-web:1.0 <user-id>/node-web:1.0
> docker push <user-id>/node-web:1.0
```

※ Docker Hub에 Repository 미리 만드는 것 잊지 말자!

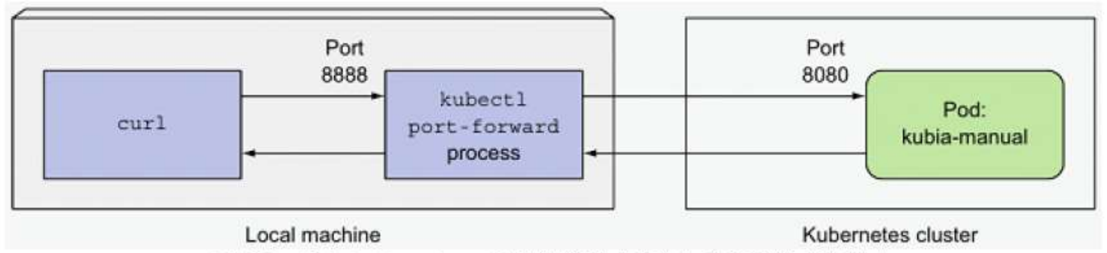
※ `docker login`도 미리 해놓는 것 잊지 말자!

※ 참고 : https://github.com/luksa/kubernetes-in-action/tree/master/Chapter02/kubia

Create Pod

node-web.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: node-web
  labels:
    creation_method: manual
    env: stage
spec:
  containers:
  - image: whatwant/node-web:1.0
    name: node-web
    ports:
    - containerPort: 8080
      protocol: TCP
```



▲ 그림 3.5 curl을 kubectl port-forward와 함께 사용할 때 일어나는 일을 간략하게 설명한다.

```
> kubectl create -f node-web.yaml
> kubectl get pods
> kubectl port-forward node-web 8080:8080 &
> curl http://localhost:8080
```

※ Pod에 접근할 수 있도록...

※ 참고 : https://livebook.manning.com/book/kubernetes-in-action/chapter-3/10

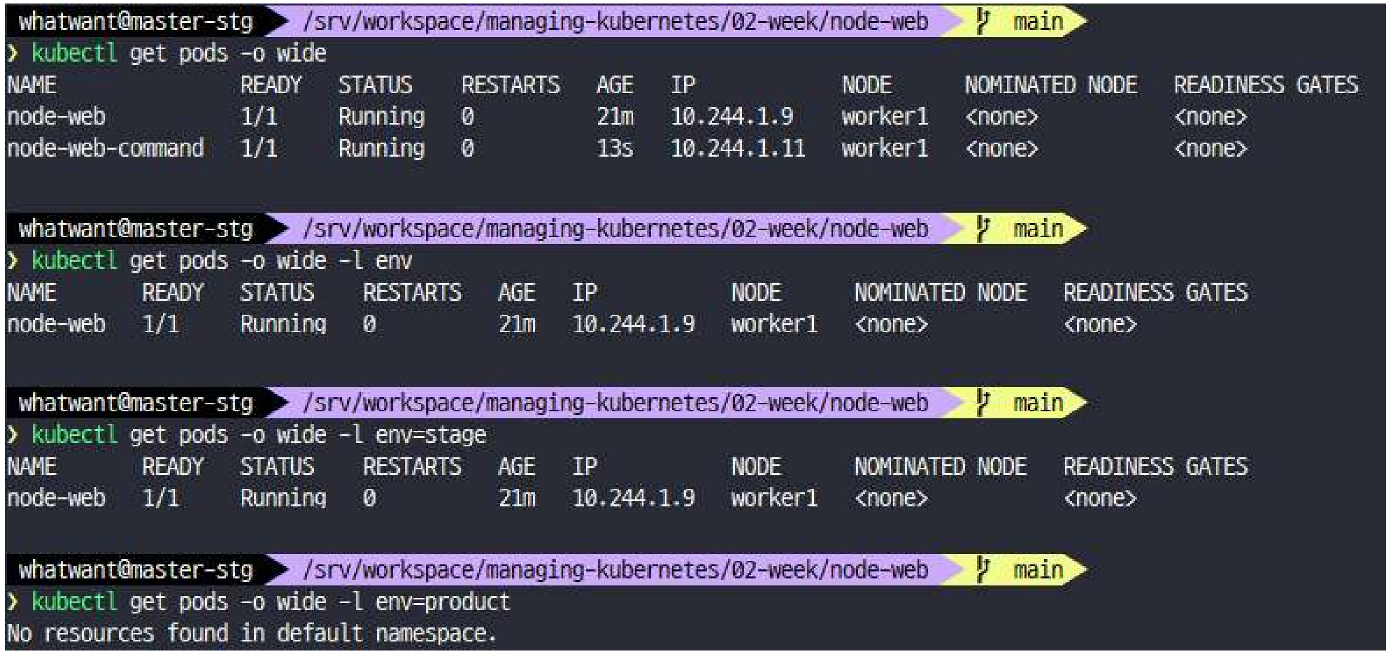
Create Pod (kubectl run)

```
> kubectl run node-web-command --image whatwant/node-web:1.0 --port=8080
```

```
> kubectl get pod node-web-command -o yaml
```

get Pods (kubectl get pods)

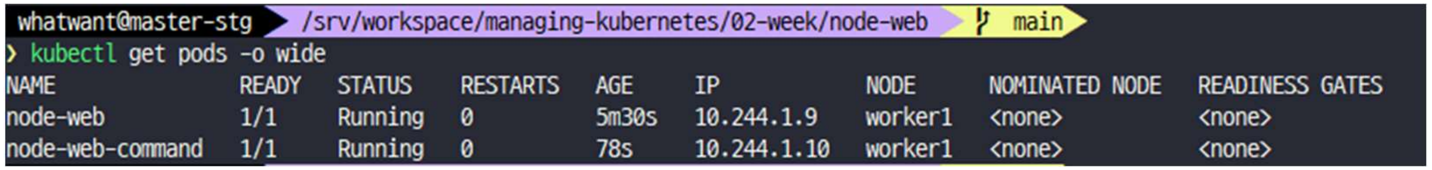
```
> kubectl get pods
> kubectl get pods -o wide
> kubectl get pods -l <label>
> kubectl get pods -l <label-key>=<label-value>
```



```
whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web  main
> kubectl get pods -o wide
NAME               READY   STATUS    RESTARTS   AGE   IP            NODE      NOMINATED NODE   READINESS GATES
node-web           1/1     Running   0          21m   10.244.1.9    worker1   <none>           <none>
node-web-command   1/1     Running   0          13s   10.244.1.11   worker1   <none>           <none>

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web  main
> kubectl get pods -o wide -l env
NAME       READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
node-web   1/1     Running   0          21m   10.244.1.9   worker1   <none>           <none>

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web  main
> kubectl get pods -o wide -l env=stage
NAME       READY   STATUS    RESTARTS   AGE   IP           NODE      NOMINATED NODE   READINESS GATES
node-web   1/1     Running   0          21m   10.244.1.9   worker1   <none>           <none>

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web  main
> kubectl get pods -o wide -l env=product
No resources found in default namespace.
```

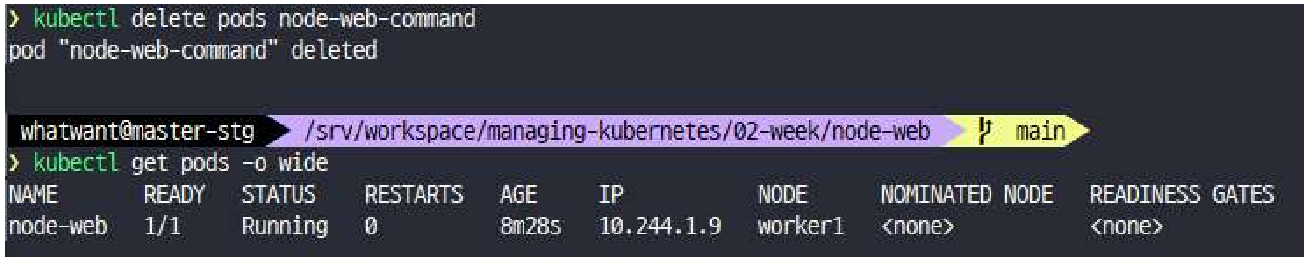
Delete Pod (kubectl delete)

```
> kubectl get pods -o wide
```



```
whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web   main
❯ kubectl get pods -o wide
NAME               READY   STATUS    RESTARTS   AGE     IP            NODE      NOMINATED NODE   READINESS GATES
node-web           1/1     Running   0          5m30s   10.244.1.9    worker1   <none>           <none>
node-web-command   1/1     Running   0          78s     10.244.1.10   worker1   <none>           <none>
```

```
> kubectl delete pods node-web-command
> kubectl get pods -o wide
```



```
❯ kubectl delete pods node-web-command
pod "node-web-command" deleted

whatwant@master-stg  /srv/workspace/managing-kubernetes/02-week/node-web   main
❯ kubectl get pods -o wide
NAME       READY   STATUS    RESTARTS   AGE     IP           NODE      NOMINATED NODE   READINESS GATES
node-web   1/1     Running   0          8m28s   10.244.1.9   worker1   <none>           <none>
```

namespace

```
> kubectl get namespaces
```

```
NAME              STATUS   AGE
default           Active   22m
kube-node-lease   Active   22m
kube-public       Active   22m
kube-system       Active   22m
```

```
> kubectl get po --namespace kube-system
```

```
NAME                                      READY   STATUS             RESTARTS   AGE
coredns-66bff467f8-462lt                  1/1     Running            0          18m
coredns-66bff467f8-gvs5f                  1/1     Running            0          18m
etcd-controlplane                         1/1     Running            0          18m
katacoda-cloud-provider-58f89f7d9-s5z9z   0/1     CrashLoopBackOff   7          18m
kube-apiserver-controlplane               1/1     Running            0          18m
kube-controller-manager-controlplane      1/1     Running            0          18m
kube-flannel-ds-amd64-cdjf8               1/1     Running            0          18m
kube-flannel-ds-amd64-s42wz               1/1     Running            0          18m
kube-keepalived-vip-sx8r7                 1/1     Running            0          17m
kube-proxy-64mbq                          1/1     Running            0          18m
kube-proxy-j2fqd                          1/1     Running            0          18m
kube-scheduler-controlplane               1/1     Running            0          18m
```

cheat sheet

```
> kubectl cluster-info
```

```
> kubectl get nodes
```

```
> kubectl get namespaces
```

```
> kubectl create -f <yaml file>
```

```
> kubectl apply -f <yaml file>
```

```
> kubectl run <name> --image <image>
```

```
> kubectl get pods
```

```
> kubectl get pods -w
```

```
> kubectl get pods -o wide
```

```
> kubectl get pods -n <namespace>
```

```
> kubectl get pods -l <label>
```

```
> kubectl logs <pod name>
```

```
> kubectl describe pod <pod name>
```

```
> kubectl delete pod <pod name>
```

GCP - GKE (Google Kubernetes Engine)

```
> gcloud container clusters create whatwant-school --num-nodes 3 --machine-type g1-small --no-enable-autoupgrade --zone us-central1-a
```

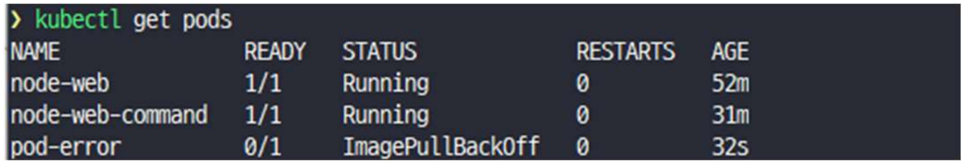
```
> gcloud container clusters delete whatwant-school --zone us-central1-a
```

trouble shooting

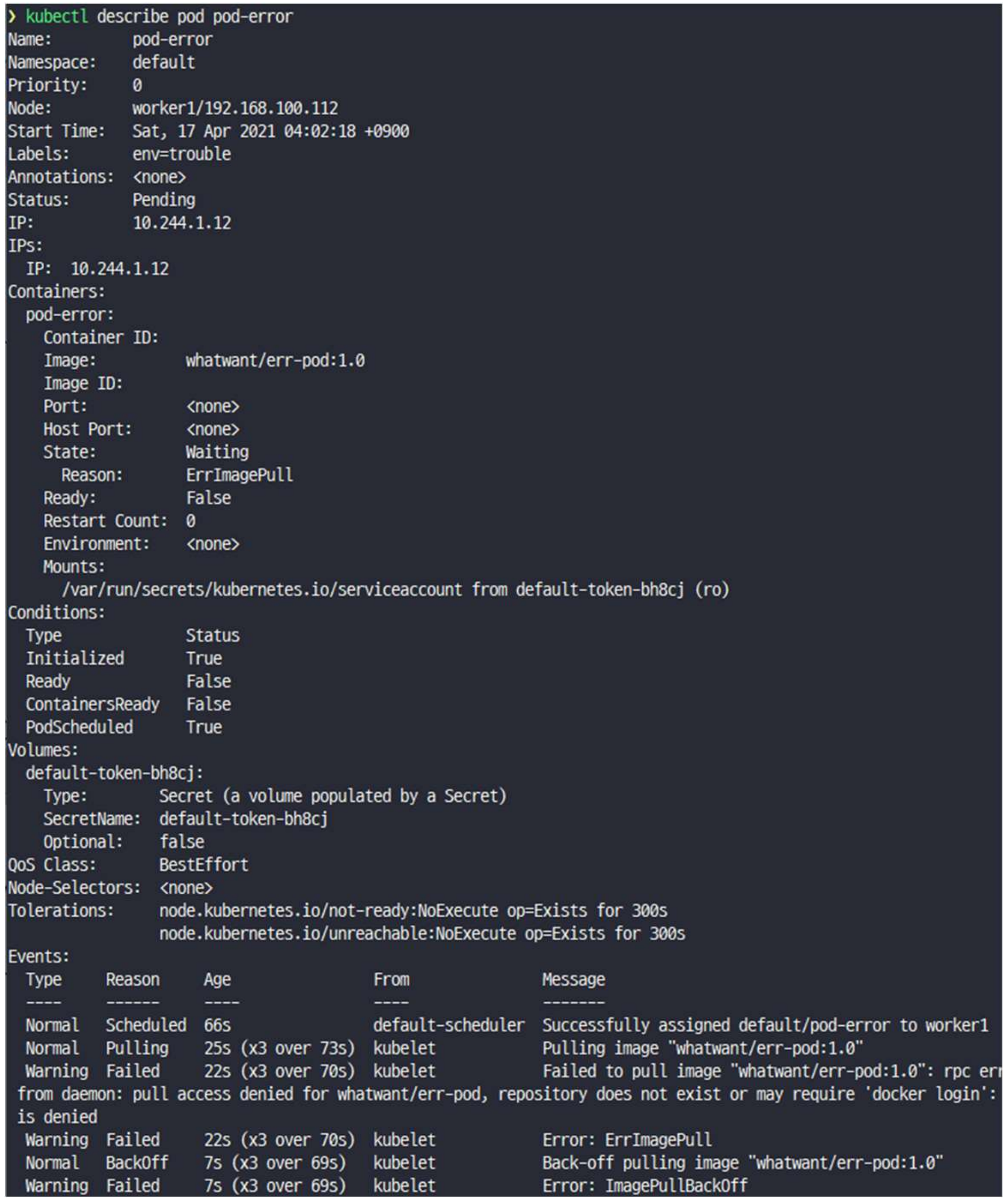
pod-error.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: pod-error
  labels:
    env: trouble
spec:
  containers:
  - image: whatwant/err-pod:1.0
    name: pod-error
```

```
> kubectl apply -f pod-error.yaml
> kubectl get pods
> kubectl describe pod pod-error
```



```
❯ kubectl get pods
NAME               READY   STATUS             RESTARTS   AGE
node-web           1/1     Running            0          52m
node-web-command   1/1     Running            0          31m
pod-error          0/1     ImagePullBackOff   0          32s
```

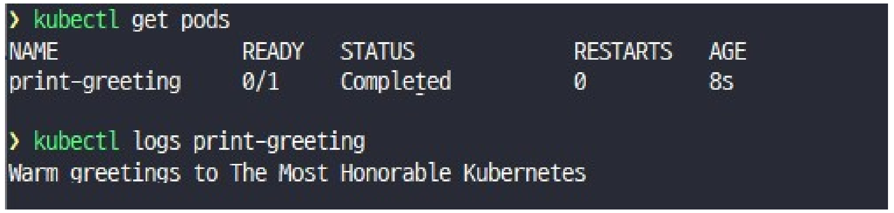


```
❯ kubectl describe pod pod-error
Name:         pod-error
Namespace:    default
Priority:     0
Node:         worker1/192.168.100.112
Start Time:   Sat, 17 Apr 2021 04:02:18 +0900
Labels:       env=trouble
Annotations:  <none>
Status:       Pending
IP:           10.244.1.12
IPs:
  IP:  10.244.1.12
Containers:
  pod-error:
    Container ID:
    Image:          whatwant/err-pod:1.0
    Image ID:
    Port:           <none>
    Host Port:      <none>
    State:          Waiting
      Reason:       ErrImagePull
    Ready:          False
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-bh8cj (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             False
  ContainersReady   False
  PodScheduled      True
Volumes:
  default-token-bh8cj:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-bh8cj
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                 node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type     Reason     Age                From               Message
  ----     ------     ----               ----               -------
  Normal   Scheduled  66s                default-scheduler  Successfully assigned default/pod-error to worker1
  Normal   Pulling    25s (x3 over 73s)  kubelet            Pulling image "whatwant/err-pod:1.0"
  Warning  Failed     22s (x3 over 70s)  kubelet            Failed to pull image "whatwant/err-pod:1.0": rpc err
 from daemon: pull access denied for whatwant/err-pod, repository does not exist or may require 'docker login':
 is denied
  Warning  Failed     22s (x3 over 70s)  kubelet            Error: ErrImagePull
  Normal   BackOff    7s (x3 over 69s)   kubelet            Back-off pulling image "whatwant/err-pod:1.0"
  Warning  Failed     7s (x3 over 69s)   kubelet            Error: ImagePullBackOff
```

Environment

pod-env.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: print-greeting
spec:
  containers:
  - name: env-print-demo
    image: bash
    env:
    - name: GREETING
      value: "Warm greetings to"
    - name: HONORIFIC
      value: "The Most Honorable"
    - name: NAME
      value: "Kubernetes"
    command: ["echo"]
    args: ["$(GREETING) $(HONORIFIC) $(NAME)"]
```



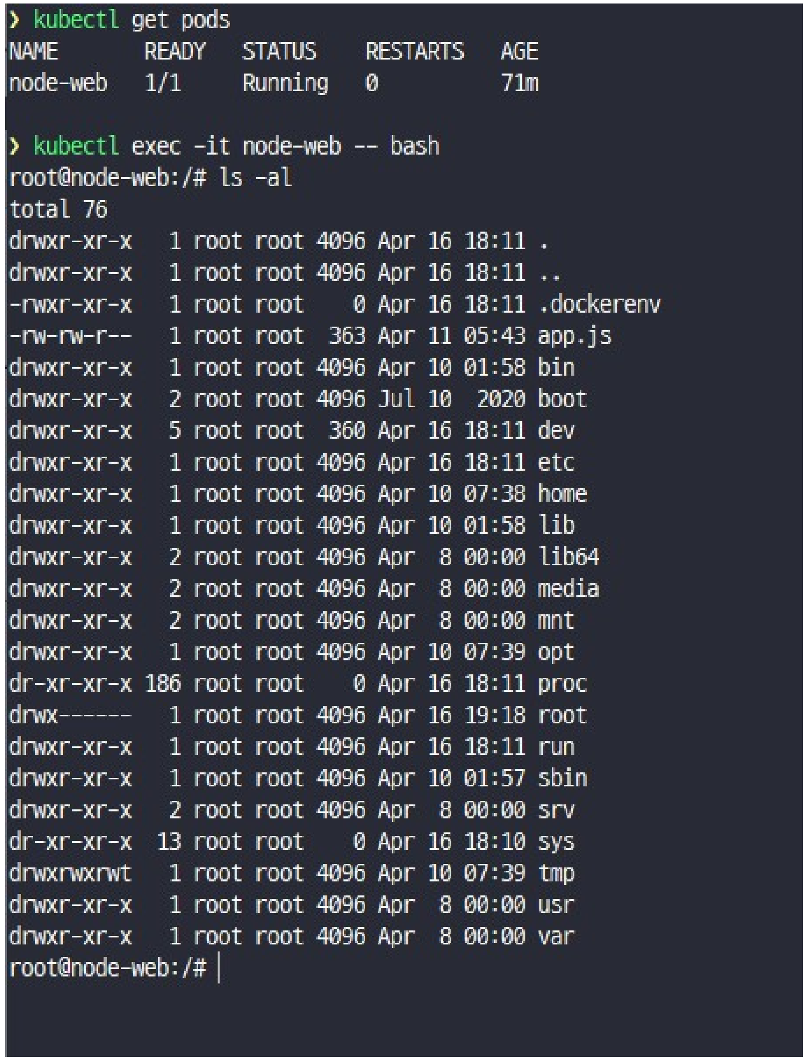
```
❯ kubectl get pods
NAME             READY   STATUS      RESTARTS   AGE
print-greeting   0/1     Completed   0          8s

❯ kubectl logs print-greeting
Warm greetings to The Most Honorable Kubernetes
```

```
> kubectl create -f pod-env.yaml
> kubectl get pods
> kubectl logs print-greeting
```

※ 참고 : https://kubernetes.io/ko/docs/tasks/inject-data-application/define-environment-variable-container/

exec

```
> kubectl get pods
> kubectl exec -it node-web -- bash
```



```
❯ kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
node-web   1/1     Running   0          71m

❯ kubectl exec -it node-web -- bash
root@node-web:/# ls -al
total 76
drwxr-xr-x   1 root root 4096 Apr 16 18:11 .
drwxr-xr-x   1 root root 4096 Apr 16 18:11 ..
-rwxr-xr-x   1 root root    0 Apr 16 18:11 .dockerenv
-rw-rw-r--   1 root root  363 Apr 11 05:43 app.js
drwxr-xr-x   1 root root 4096 Apr 10 01:58 bin
drwxr-xr-x   2 root root 4096 Jul 10  2020 boot
drwxr-xr-x   5 root root  360 Apr 16 18:11 dev
drwxr-xr-x   1 root root 4096 Apr 16 18:11 etc
drwxr-xr-x   1 root root 4096 Apr 10 07:38 home
drwxr-xr-x   1 root root 4096 Apr 10 01:58 lib
drwxr-xr-x   2 root root 4096 Apr  8 00:00 lib64
drwxr-xr-x   2 root root 4096 Apr  8 00:00 media
drwxr-xr-x   2 root root 4096 Apr  8 00:00 mnt
drwxr-xr-x   1 root root 4096 Apr 10 07:39 opt
dr-xr-xr-x 186 root root    0 Apr 16 18:11 proc
drwx------   1 root root 4096 Apr 16 19:18 root
drwxr-xr-x   1 root root 4096 Apr 16 18:11 run
drwxr-xr-x   1 root root 4096 Apr 10 01:57 sbin
drwxr-xr-x   2 root root 4096 Apr  8 00:00 srv
dr-xr-xr-x  13 root root    0 Apr 16 18:10 sys
drwxrwxrwt   1 root root 4096 Apr 10 07:39 tmp
drwxr-xr-x   1 root root 4096 Apr  8 00:00 usr
drwxr-xr-x   1 root root 4096 Apr  8 00:00 var
root@node-web:/# |
```

https://kahoot.it/