SAP BTP, Kyma runtime

Managed Kubernetes(K8S)

Han, Jungwoo, SAP
Nov, 2023

PUBLIC

THE BEST RUN

Hands-on

“pure” Docker

Docker Hands-on

[Docker Install]

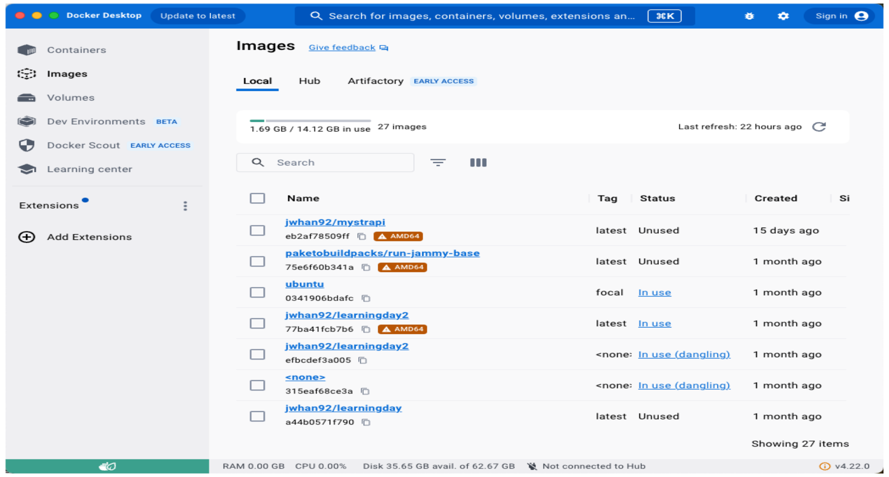
[Docker Hub 가입 및 Repository 생성]

Hands-on에 필요한 환경 점검

[local desktop]

```
$>docker –v
$>docker-compose –v
$>kubectl version
```

[docker host - desktop]



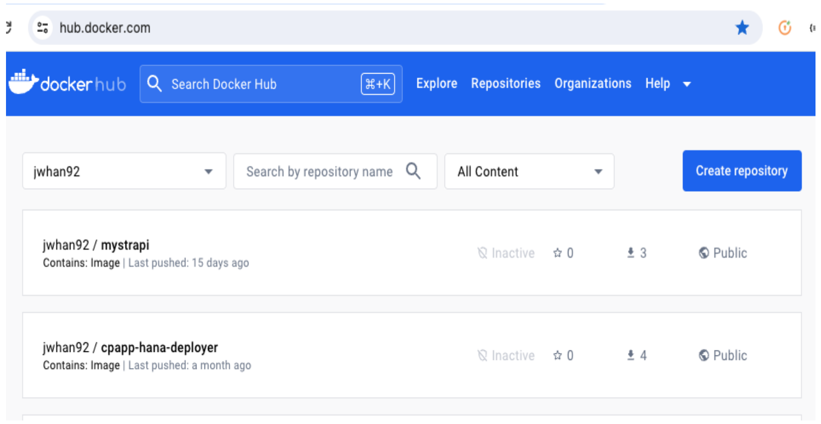
1.69 GB / 14.12 GB in use 27 images Last refresh: 22 hours ago

Name	Tag	Status	Created
jwhan92/mystrapi eb2af78509ff AMD64	latest	Unused	15 days ago
paketobuildpacks/run-jammy-base 75e6f60b341a AMD64	latest	Unused	1 month ago
ubuntu 0341906bdafc	focal	In use	1 month ago
jwhan92/learningday2 77ba41fcb7b6 AMD64	latest	In use	1 month ago
jwhan92/learningday2 efbcdef3a005	<none:	In use (dangling)	1 month ago
<none> 315eaf68ce3a	<none:	In use (dangling)	1 month ago
jwhan92/learningday a44b0571f790	latest	Unused	1 month ago

Showing 27 items

RAM 0.00 GB CPU 0.00% Disk 35.65 GB avail. of 62.67 GB Not connected to Hub v4.22.0

[docker registry] (hub.docker.com)



jwhan92 / mystrapi
Contains: Image | Last pushed: 15 days ago
Inactive ☆ 0 ⬇ 3 Public

jwhan92 / cpapp-hana-deployer
Contains: Image | Last pushed: a month ago
Inactive ☆ 0 ⬇ 4 Public

Hands-on

```
$>docker create nginx

$>docker ps –a

$>docker start container_id [ || names ]]

$>docker ps
```

```
$>docker run \
-i      \                        #호스트의 표준입력을 컨테이너와 연결(interactive
-t      \                        #TTY 할당
-rm  \                           #컨테이너 실행 종료 후 자동 삭제
-d    \                          #백그라운드 모드로 실행(detached) ==> docker ru
--name abcde \                   #컨테이너 이름 지정(default 자동지정)
-p 80:80 \                       #호스트 – 컨테이너 간 포트 바인딩
-v /opt/example:/example. \      #호스트 – 컨테이너간 볼륨 바인딩
Dockerhub/imagename:latest  \#실행할 이미지
my_command                       #컨테이너 내에서 실행할 명령어
```

Hands-on

$> docker run ubuntu:focal

$> docker ps

$> docker ps –a

```
i307487@N2CV4PQ344 ~ % docker ps -a
CONTAINER ID   IMAGE                              COMMAND                   CREATED            ST
3405a1e87fd8   nginx                              "/docker-entrypoint.…"   42 minutes ago   Cre
970e6c2ea3f0   ubuntu:focal                       "/bin/bash"               3 hours ago      Ex
11e903be35fa   ubuntu:focal                       "/bin/bash"               3 hours ago      Ex
```

$> docker run –i –t ubuntu:focal

root@ababcbdb:/# ls

종료시 exit

$>docker run ubuntu:focal id

$>docker ps -a

$> docker run nginx

,…foreground mode…

$> docker run –d nginx

$> docker run –d –name my-nginx

$> docker ps

$> docker run –p 80:80 –d nginx

$> docker run –p 80:80 –d nginx

$> docker ps

$> curl localhost:80

컨테이너 상태 확인

실행 중인 컨테이너 상태 확인

```
$ docker ps
```

컨테이너 상세 정보 확인

```
$ docker inspect [container]
```

컨테이너 일시중지 // 재개

```
$ docker pause [ container]
```

```
$ docker unpause [ container]
```

전체 컨테이너 상태 확인

```
$ docker ps -a
```

컨테이너 종료(SIGTERM시그널 전달)

```
$ docker stop [container]
```

컨테이너 종료(SIGKILL시그널 전달)

```
$ docker kill [container]
```

컨테이너 종료

```
$ docker stop $(docker ps –a –q)
```

컨테이너 삭제

컨테이너 삭제(실행중인 컨테이너 불가)

```
$ docker rm [container]
```

컨테이너 실행 종료 후 자동 삭제

```
$ docker run –rm
```

컨테이너 강제 종료 후 삭제(SIGKILL 시그널 전달)

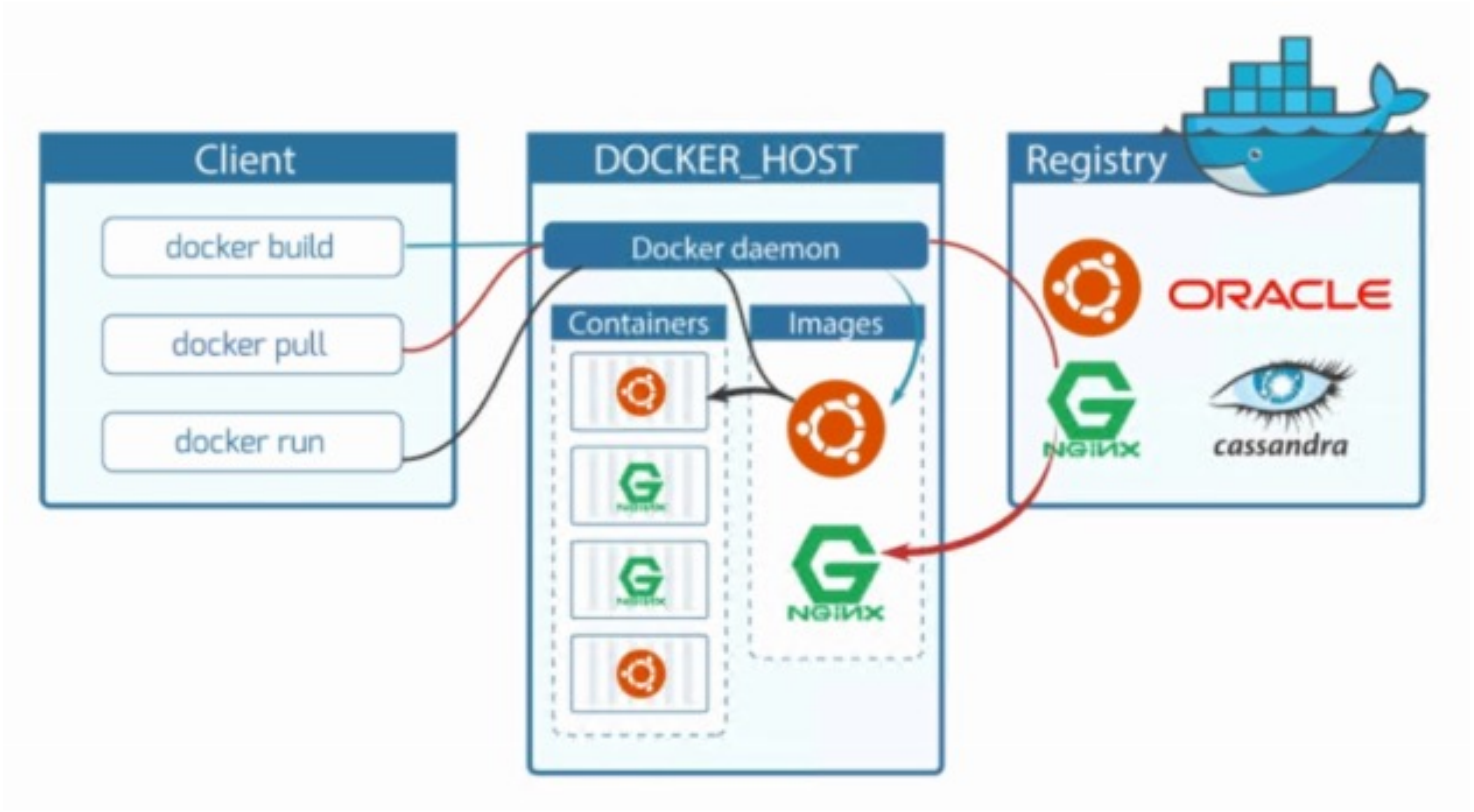
```
$ docker rm –rf [container]
```

중지된 모든 컨테이너 삭제

```
$ docker container prune
```

도커를 이용한 컨테이너 관리

도커 구성요소



도커 이미지 생성

도커 파일(Dockerfile) 없이 이미지 생성

$ docker commit –a *author* –m “*message*” *source_conainter target_image_name*

```
$> docker run -it --name my_unbuntu ubuntu:focal
root@72e3b6901108:/# ls
bin  boot  dev  etc  home  lib  media  mnt  opt  proc  r
oot  run  sbin  srv  sys  tmp  usr  var
root@72e3b6901108:/# cat > my_file
Hello Kyma!!!
^C
root@72e3b6901108:/# cat my_file
Hello Kyma!!!
```

Exit

```
i307487@N2CV4PQ344 ~ % docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED           STATUS
72e3b6901108   ubuntu:focal   "/bin/bash"   51 seconds ago    Up 51 se
i307487@N2CV4PQ344 ~ % docker commit -a me -m "Add_myfile" my_unbuntu
sha256:f7152a025e36f67f6a2ca65f529cb51e98ad18663a4471e3d457ec3035f327
i307487@N2CV4PQ344 ~ %
i307487@N2CV4PQ344 ~ % docker images
REPOSITORY                              TAG       IMAGE ID       CRE
my-ubuntu-image                         v1        f7152a025e36   11 
i307487@N2CV4PQ344 ~ % docker inspect my-ubuntu-image:v1
[
    {
        "Id": "sha256:f7152a025e36f67f6a2ca65f529cb51e98ad18663a4471e
        "RepoTags": [
            "my-ubuntu-image:v1"
        ],
….
```

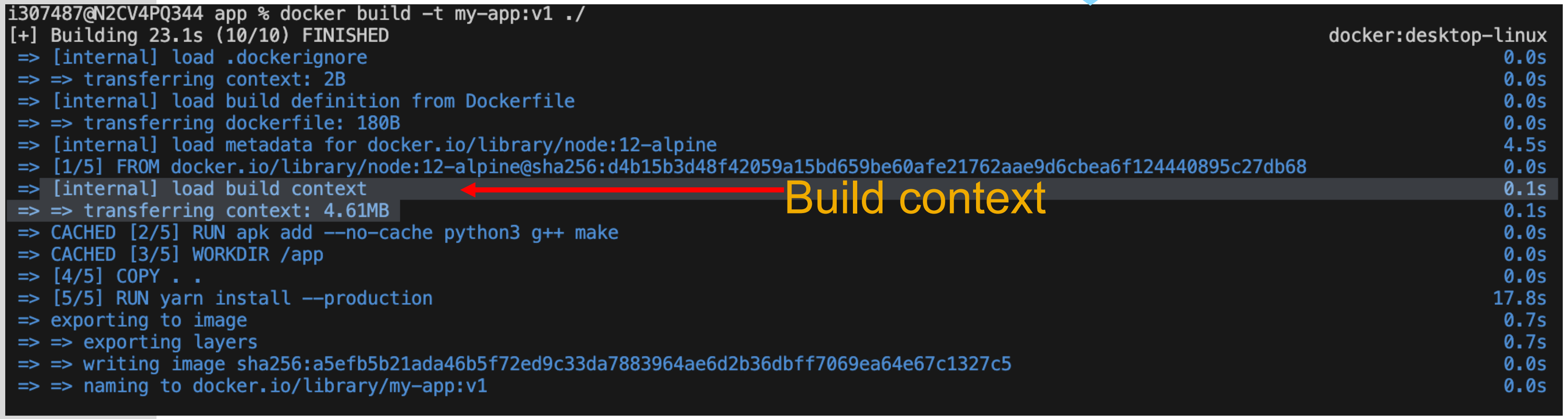
```
i307487@N2CV4PQ344 ~ % docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED          STATUS
72e3b6901108   ubuntu:focal   "/bin/bash"   5 minutes ago    Up 5 minu

i307487@N2CV4PQ344 ~ % docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAME

i307487@N2CV4PQ344 ~ % docker run -i -t my-ubuntu-image:v1
root@201b29008164:/# ls
bin  boot  dev  etc  home  lib  media  mnt  my_file  opt  proc  root
root@201b29008164:/# cat my_file
Hello Kyma!!!
root@201b29008164:/#
```

도커 이미지 생성

도커 파일(Dockerfile)을 이용한 이미지 생성

```
FROM node:16
LABEL description="Simple server with Node.js"

# Create app directory
WORKDIR /app

# Install app dependencies
# A wildcard is used to ensure both package.json AND
package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 8080
CMD [ "node", "server.js" ]
```

```
# docker build [OPTIONS] PATH
# ./ 디렉토리를 빌드 컨텍스트로 my_app:v1 이미지 빌드 (Dockerfile 이용)
$> docker build –t my_app:v1 ./

# ./ 디렉토리를 빌드 컨텍스트로 my_app:v1 이미지 빌드(example/MyDockerfile 이용)
$> docker build –t my_app:v1 –f example/MyDockerfile ./
```



```
i307487@N2CV4PQ344 app % docker build -t my-app:v1 ./
[+] Building 23.1s (10/10) FINISHED                                                                  docker:desktop-linux
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 180B                                                                                0.0s
 => [internal] load metadata for docker.io/library/node:12-alpine                                                   4.5s
 => [1/5] FROM docker.io/library/node:12-alpine@sha256:d4b15b3d48f42059a15bd659be60afe21762aae9d6cbea6f124440895c27db68  0.0s
 => [internal] load build context                                                                                   0.1s
 => => transferring context: 4.61MB                                                                                 0.1s
 => CACHED [2/5] RUN apk add --no-cache python3 g++ make                                                            0.0s
 => CACHED [3/5] WORKDIR /app                                                                                       0.0s
 => [4/5] COPY . .                                                                                                  0.0s
 => [5/5] RUN yarn install --production                                                                            17.8s
 => exporting to image                                                                                              0.7s
 => => exporting layers                                                                                             0.7s
 => => writing image sha256:a5efb5b21ada46b5f72ed9c33da7883964ae6d2b36dbff7069ea64e67c1327c5                        0.0s
 => => naming to docker.io/library/my-app:v1                                                                        0.0s
```

도커 파일(Dockerfile)을 이용한 이미지 생성

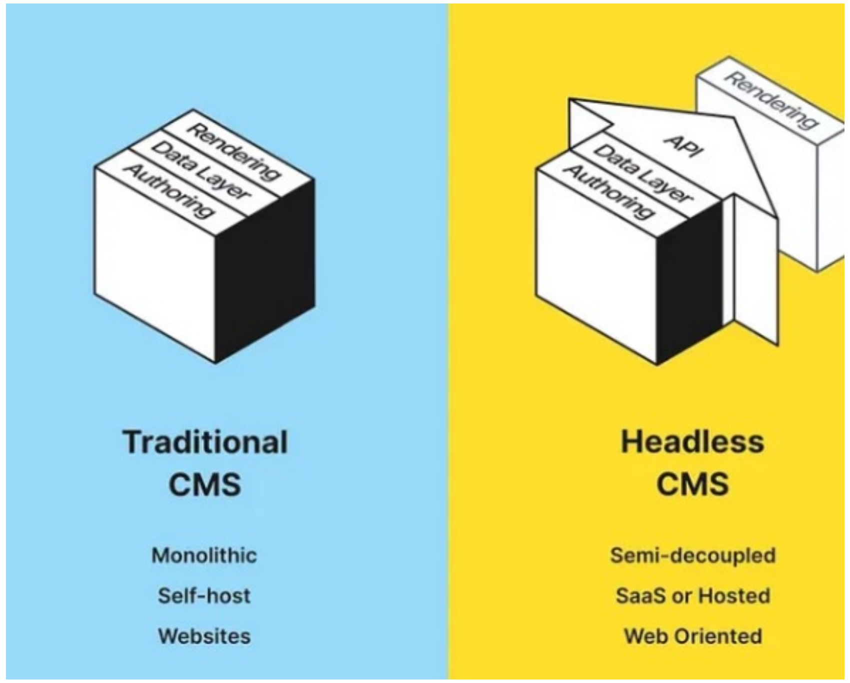
예제 : Node.js server 실행.

```
$> git clone https://github.com/micol92/Kyma01


$> cd Kyma01

$> docker build –force-rm –t nodejs-server .

……

$> docker images

$> docker run -d nodejs-server

$> docker ps
$> curl localhost:8080
curl: (7) Failed to connect to localhost port 8080 after

$> docker rm -f ae9405500507
Ae9405500507

$> docker run -d -p8080:8080 nodejs-server
3e589f78f3492b40b99b7337ec119b4769b210d82e1ef68068af22e1
$> curl localhost:8080
Hello World
```

```
FROM node:16
LABEL description="Simple server with Node.js"

# Create app directory
WORKDIR /app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --only=production

# Bundle app source
COPY . .

EXPOSE 8080
CMD [ "node", "server.js" ]
```

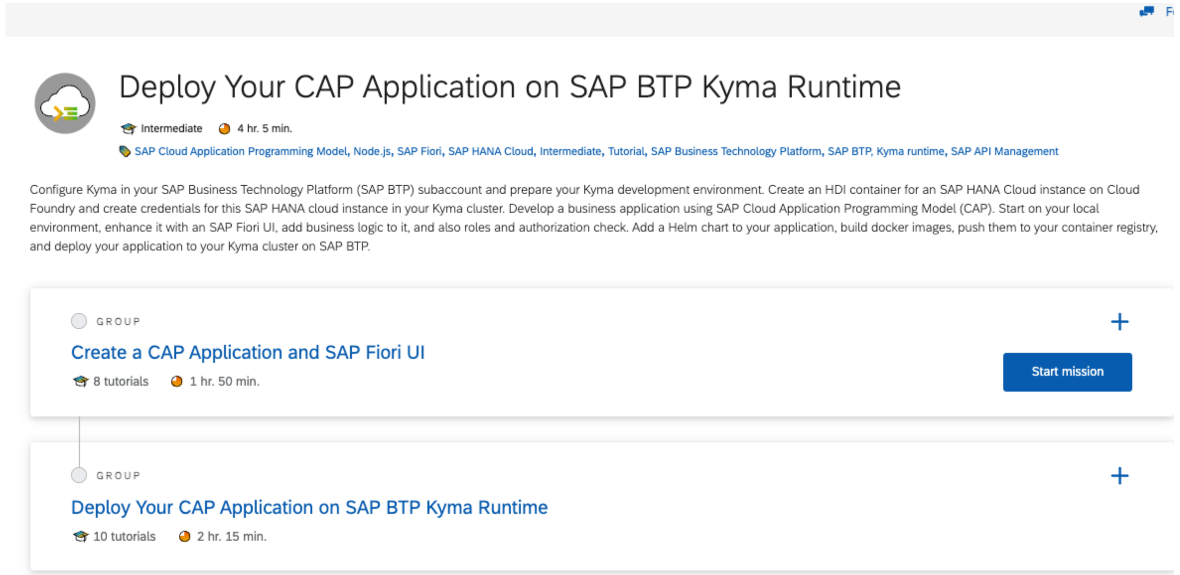
Hands-on

Strapi (Headless CMS)



Docker Hands-on

[Deploy your CAP application on SAP BTP Kyma Runtime]


Deploy Your CAP Application on SAP BTP Kyma Runtime

Intermediate 4 hr. 5 min.

SAP Cloud Application Programming Model, Node.js, SAP Fiori, SAP HANA Cloud, Intermediate, Tutorial, SAP Business Technology Platform, SAP BTP, Kyma runtime, SAP API Management

Configure Kyma in your SAP Business Technology Platform (SAP BTP) subaccount and prepare your Kyma development environment. Create an HDI container for an SAP HANA Cloud instance on Cloud Foundry and create credentials for this SAP HANA cloud instance in your Kyma cluster. Develop a business application using SAP Cloud Application Programming Model (CAP). Start on your local environment, enhance it with an SAP Fiori UI, add business logic to it, and also roles and authorization check. Add a Helm chart to your application, build docker images, push them to your container registry, and deploy your application to your Kyma cluster on SAP BTP.

GROUP
Create a CAP Application and SAP Fiori UI
8 tutorials 1 hr. 50 min.

GROUP
Deploy Your CAP Application on SAP BTP Kyma Runtime
10 tutorials 2 hr. 15 min.

[Docker Install]

[Docker Hub 가입 및 Repository 생성]

Hands-on에 필요한 환경 점검

BTP Trial Account 구성

Configure Kyma in the subaccount

Node.JS 설치 : https://nodejs.org/en/download

NPM Package 설치 :

```
$> npm install npm@latest –g
$> npm install @sap/cds-dk -g
```

VS Code 설치 : https://code.visualstudio.com/

Kubectl 설치

Kubelogin 설치

Kyma cluster에 로그인.

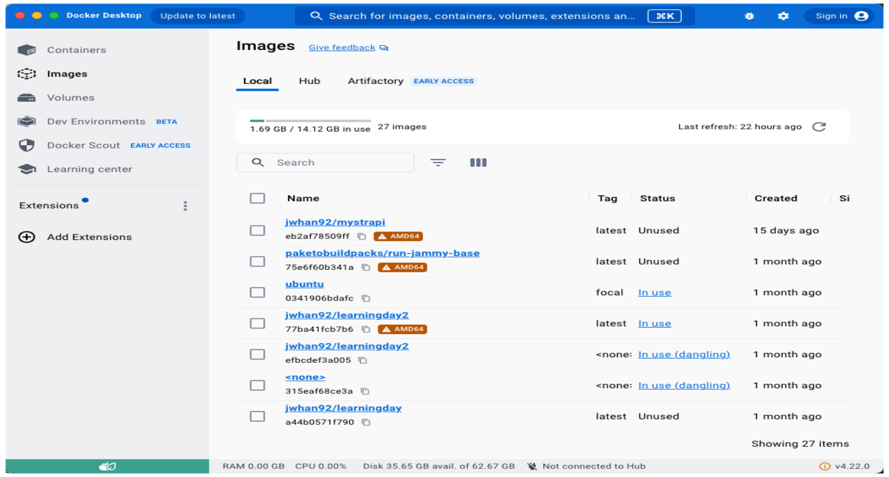
Tutorial 다운로드

Hands-on에 필요한 환경 점검

[local desktop]

```
$>docker –v
$>docker-compose –v
$>kubectl version
```

[docker host - desktop]


Docker Desktop — Update to latest — Search for images, containers, volumes, extensions an... ⌘K — Sign in

Containers · Images · Volumes · Dev Environments BETA · Docker Scout EARLY ACCESS · Learning center · Extensions · Add Extensions

Images — Give feedback

Local · Hub · Artifactory EARLY ACCESS

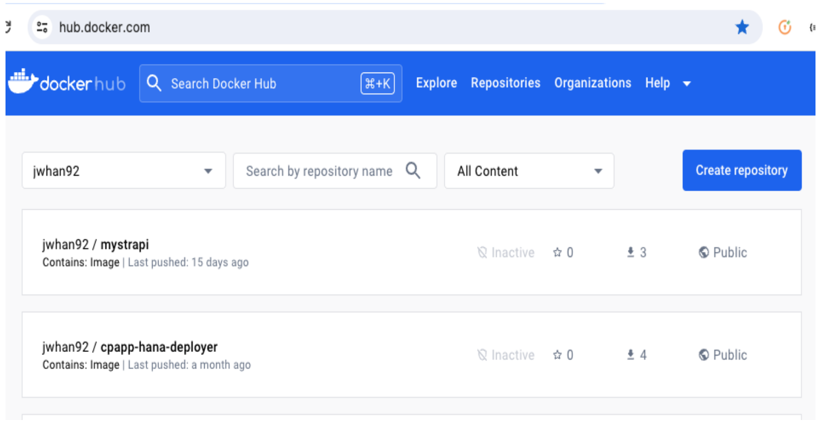
1.69 GB / 14.12 GB in use 27 images — Last refresh: 22 hours ago

Name	Tag	Status	Created
jwhan92/mystrapi eb2af78509ff AMD64	latest	Unused	15 days ago
paketobuildpacks/run-jammy-base 75e6f60b341a AMD64	latest	Unused	1 month ago
ubuntu 0341906bdafc	focal	In use	1 month ago
jwhan92/learningday2 77ba41fcb7b6 AMD64	latest	In use	1 month ago
jwhan92/learningday2 efbcdef3a005	<none:	In use (dangling)	1 month ago
<none> 315eaf68ce3a	<none:	In use (dangling)	1 month ago
jwhan92/learningday a44b0571f790	latest	Unused	1 month ago

Showing 27 items

RAM 0.00 GB CPU 0.00% Disk 35.65 GB avail. of 62.67 GB Not connected to Hub v4.22.0

[docker registry] (hub.docker.com)


hub.docker.com

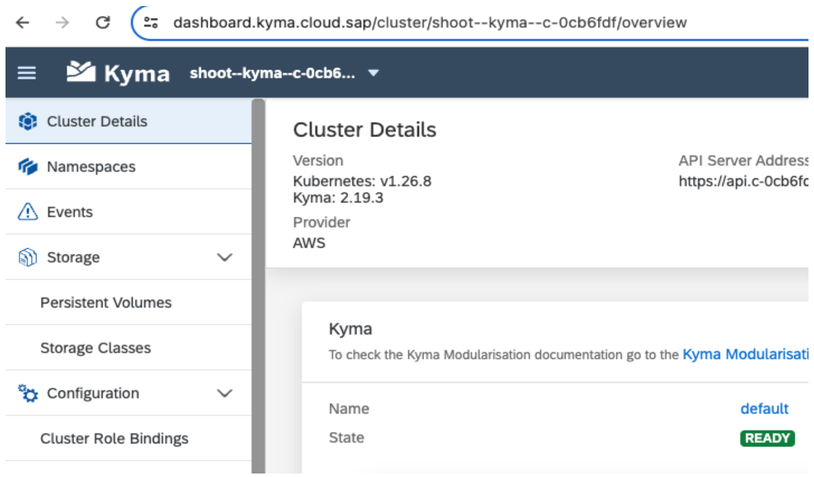
docker hub — Search Docker Hub ⌘+K — Explore · Repositories · Organizations · Help

jwhan92 — Search by repository name — All Content — Create repository

jwhan92 / mystrapi
Contains: Image | Last pushed: 15 days ago — Inactive ☆ 0 ⬇ 3 Public

jwhan92 / cpapp-hana-deployer
Contains: Image | Last pushed: a month ago — Inactive ☆ 0 ⬇ 4 Public

[Kyma]


dashboard.kyma.cloud.sap/cluster/shoot--kyma--c-0cb6fdf/overview

Kyma shoot--kyma--c-0cb6...

Cluster Details · Namespaces · Events · Storage · Persistent Volumes · Storage Classes · Configuration · Cluster Role Bindings

Cluster Details

Version
Kubernetes: v1.26.8
Kyma: 2.19.3
Provider
AWS

API Server Address
https://api.c-0cb6fc

Kyma
To check the Kyma Modularisation documentation go to the Kyma Modularisati

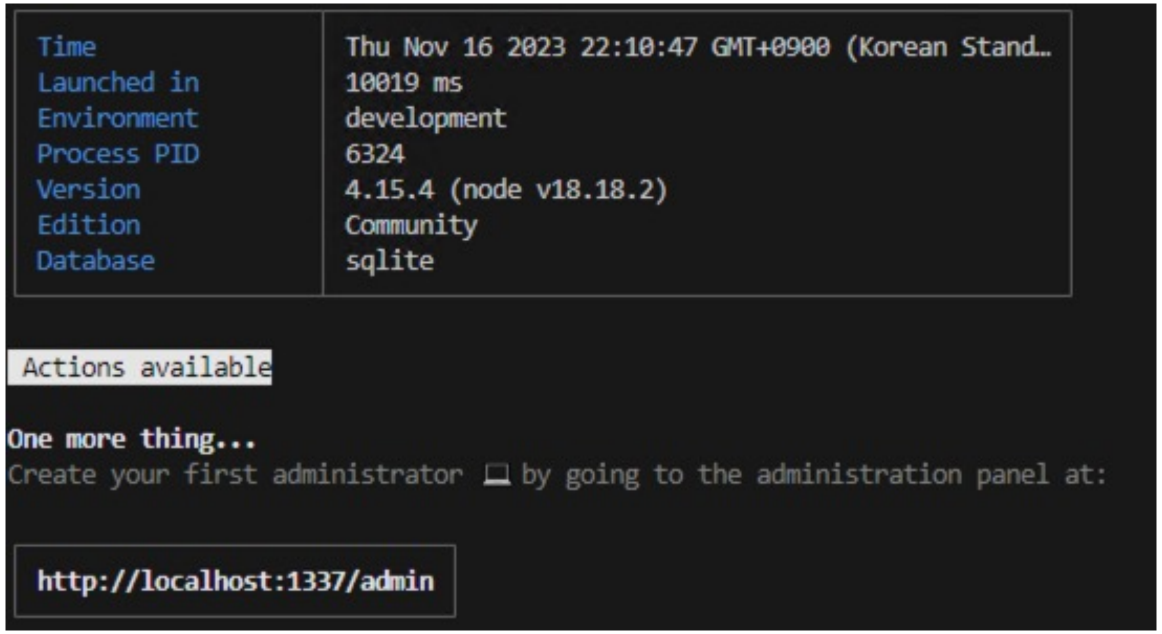
Name default
State READY

(https://dashboard.kyma.cloud.sap/cluster/shoot--kyma--c-0cb6fdf/overview)

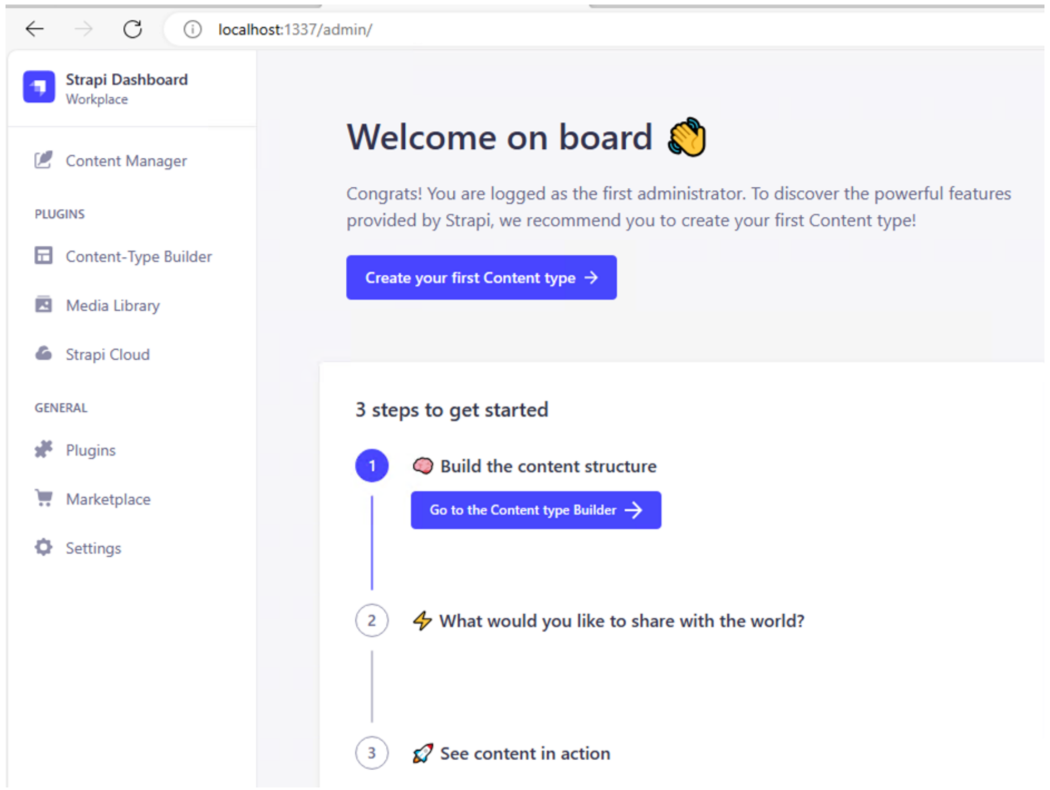
Setup Strapi

$ npx create-strapi-app@latest mystrapi20 --quickstart

Need to install the following packages:
create-strapi-app@4.15.4
Ok to proceed? (y) y
Creating a quickstart project.
Creating a new Strapi application at C:\Users\i307487\Kyma\mystrapi20.
Creating files.
Dependencies installed successfully.
Initialized a git repository.



```
Time            Thu Nov 16 2023 22:10:47 GMT+0900 (Korean Stand…
Launched in     10019 ms
Environment     development
Process PID     6324
Version         4.15.4 (node v18.18.2)
Edition         Community
Database        sqlite

 Actions available

One more thing...
Create your first administrator 💻 by going to the administration panel at:

http://localhost:1337/admin
```



localhost:1337/admin/

Welcome on board 👋

Congrats! You are logged as the first administrator. To discover the powerful features provided by Strapi, we recommend you to create your first Content type!

Create your first Content type →
3 steps to get started
1 Build the content structure
Go to the Content type Builder →
2 What would you like to share with the world?
3 See content in action

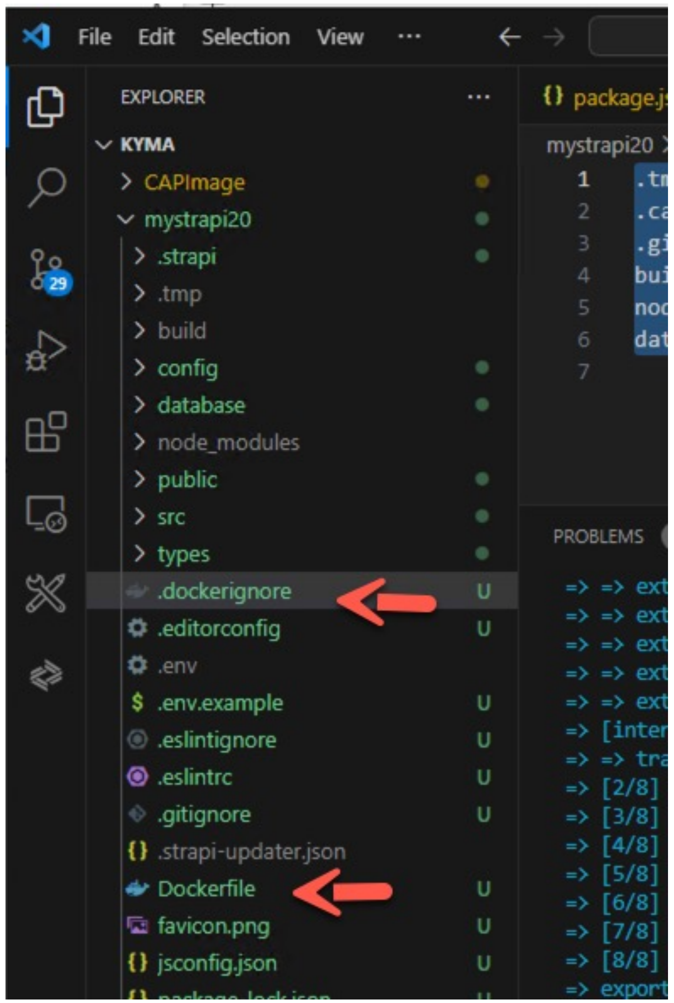
Docker Setup

Dockerfile

```
FROM node:16
# Installing libvips-dev for sharp Compatability
RUN apt-get update && apt-get install libvips-dev -y
ARG NODE_ENV=development
ENV NODE_ENV=${NODE_ENV}
WORKDIR /opt/
COPY ./package.json ./package-lock.json ./
ENV PATH /opt/node_modules/.bin:$PATH
RUN npm install
WORKDIR /opt/app
COPY ./ .
RUN npm run build
EXPOSE 1337
CMD ["npm", "run", "develop"]
```

.dockerignore

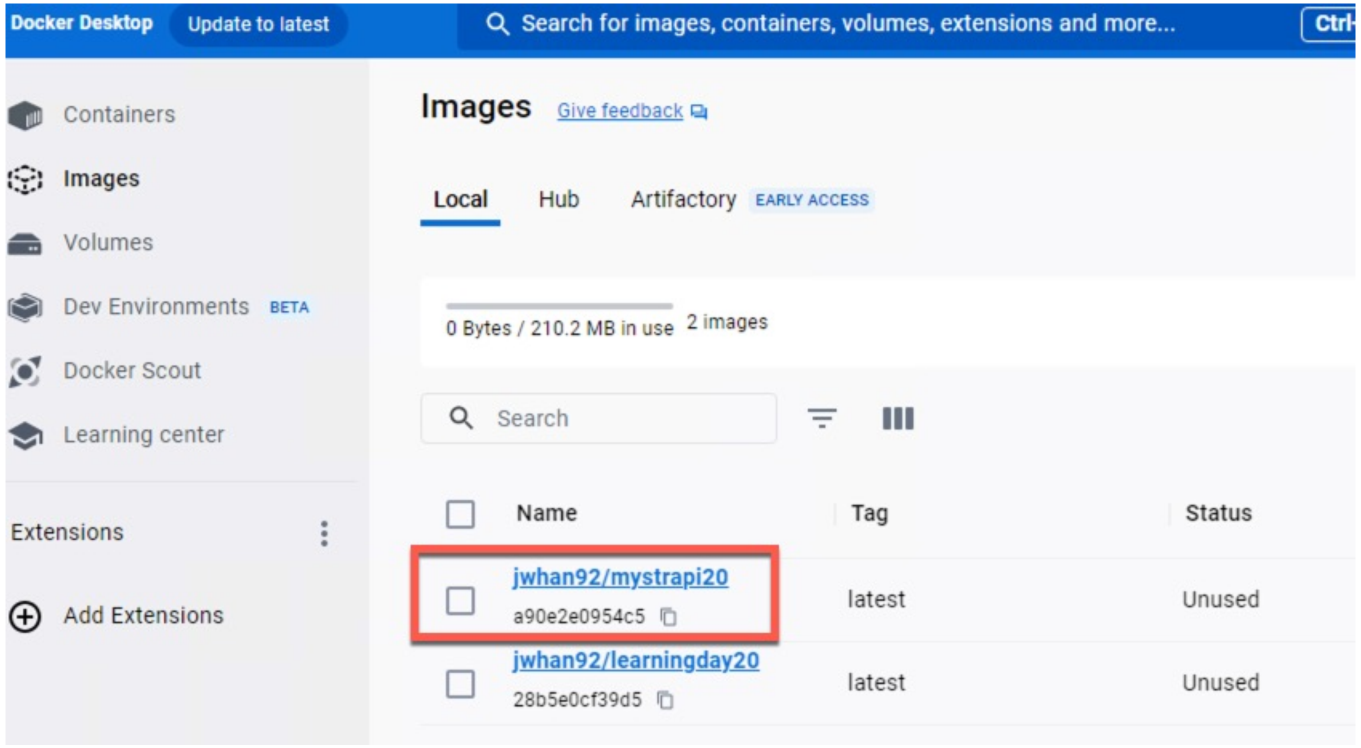
```
.tmp/
.cache/
.git/
build/
node_modules/
data/
```



```
KYMA
  CAPImage
  mystrapi20
    .strapi
    .tmp
    build
    config
    database
    node_modules
    public
    src
    types
    .dockerignore        U
    .editorconfig        U
    .env
    .env.example         U
    .eslintignore        U
    .eslintrc            U
    .gitignore           U
    .strapi-updater.json
    Dockerfile           U
    favicon.png          U
    jsconfig.json        U
```

Docker Build & Run

$> docker build -t jwhan92/mystrapi20:latest .



0 Bytes / 210.2 MB in use 2 images

Name	Tag	Status
jwhan92/mystrapi20 a90e2e0954c5	latest	Unused
jwhan92/learningday20 28b5e0cf39d5	latest	Unused

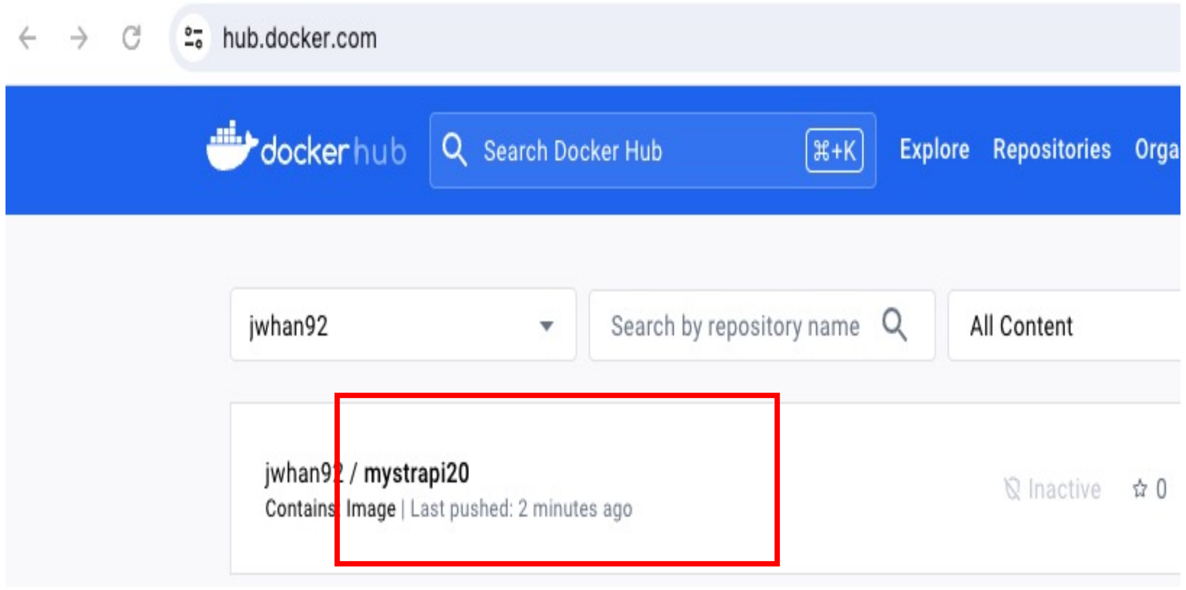
$> docker run –d -p 1337:1337 jwhan92/mystrapi20

```
$ docker ps
CONTAINER ID   IMAGE              COMMAND                  CREATED
STATUS         PORTS                  NAMES
00f0795e3f0e   jwhan92/mystrapi20   "docker-entrypoint.s…"   9 seconds ago
Up 7 seconds   0.0.0.0:1337->1337/tcp   sharp_colden
```

Verify ➔ http://localhost:1337

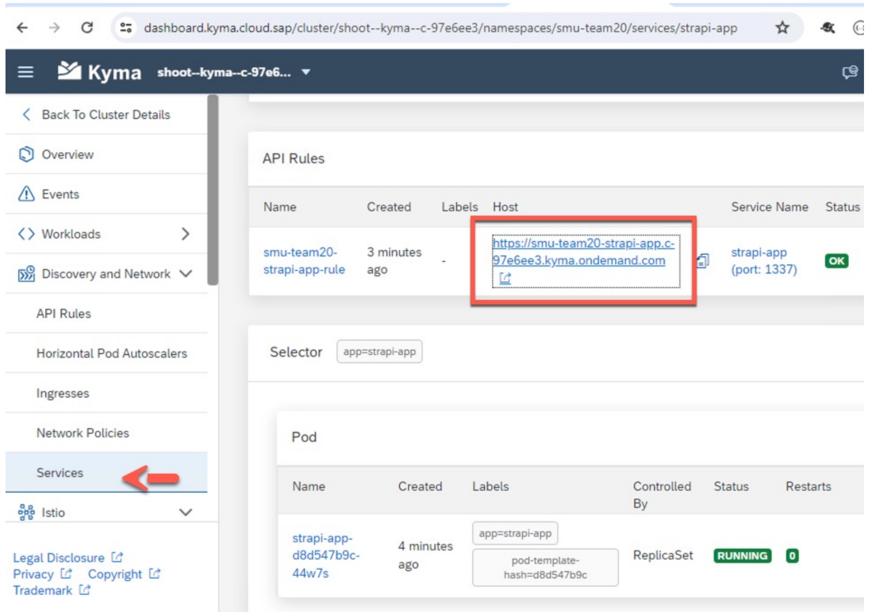
Docker push into Kyma from Docker hub

```
$ docker push jwhan92/mystrapi20
Using default tag: latest
The push refers to repository [docker.io/jwhan92/mystrapi20]
0293d0396bcd: Preparing
d1b6d97d553a: Preparing
0b3d58ecc00b: Preparing
a39ab6a99003: Pushed
……
794ce8b1b516: Mounted from library/node
3220beed9b06: Mounted from library/node
684f82921421: Mounted from library/node
9af5f53e8f62: Mounted from library/node
latest: digest: sha256:6ac351a9a0537e73fbfe8b3323861af435d7ff391a3795b9931112e01ebd655e size: 3474
```



jwhan92 / mystrapi20
Contains: Image | Last pushed: 2 minutes ago
Inactive ☆ 0

```
$ kubectl apply -f ./deployment.yaml
deployment.apps/strapi-app unchanged
service/strapi-app unchanged
apirule.gateway.kyma-project.io/smu-team20-strapi-app-rule created
```



API Rules

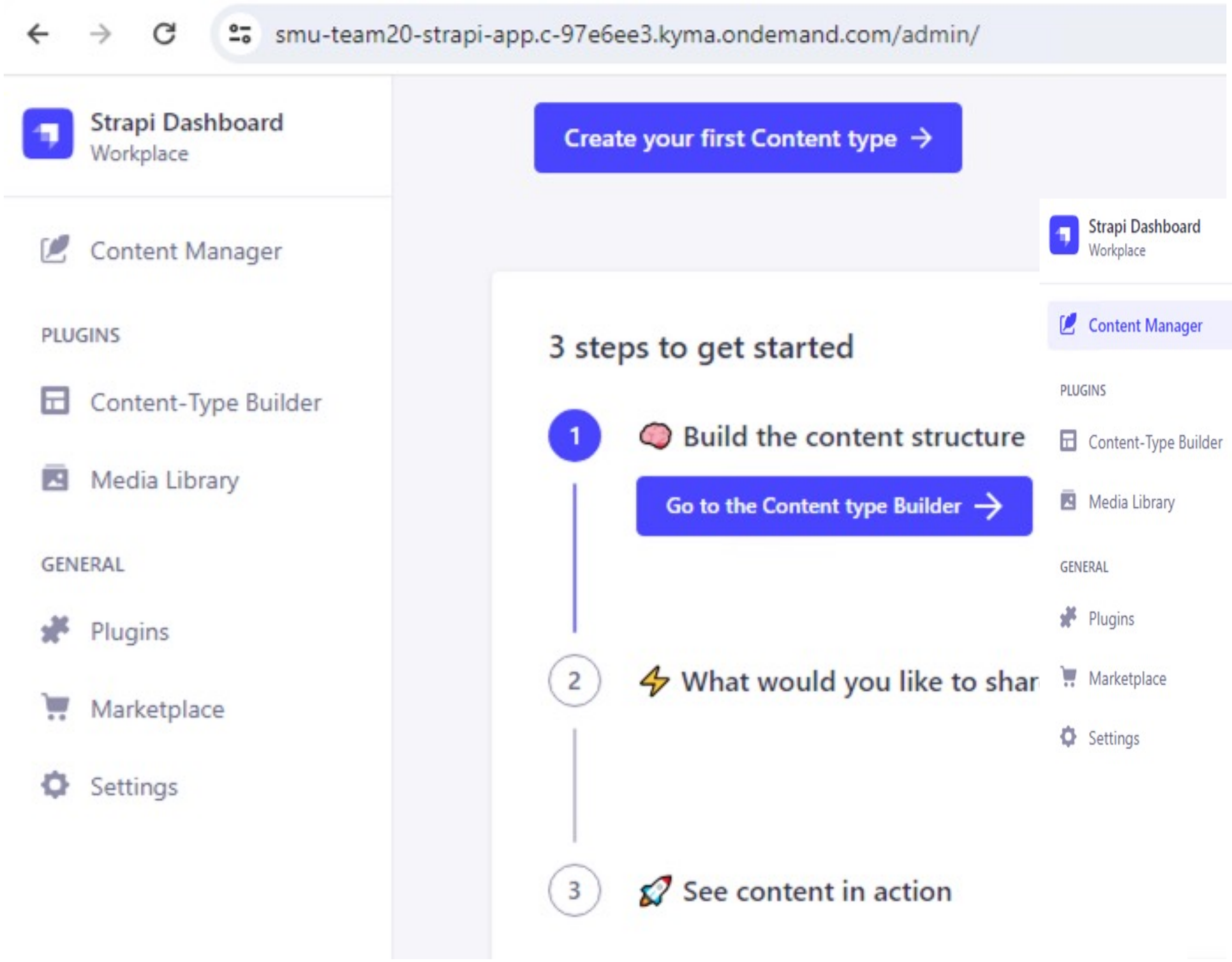
Name	Created	Labels	Host	Service Name	Status
smu-team20-strapi-app-rule	3 minutes ago	-	https://smu-team20-strapi-app.c-97e6ee3.kyma.ondemand.com	strapi-app (port: 1337)	OK

Selector app=strapi-app

Pod

Name	Created	Labels	Controlled By	Status	Restarts
strapi-app-d8d547b9c-44w7s	4 minutes ago	app=strapi-app pod-template-hash=d8d547b9c	ReplicaSet	RUNNING	0

Run Strapi @ Kyma runtime



smu-team20-strapi-app.c-97e6ee3.kyma.ondemand.com/admin/

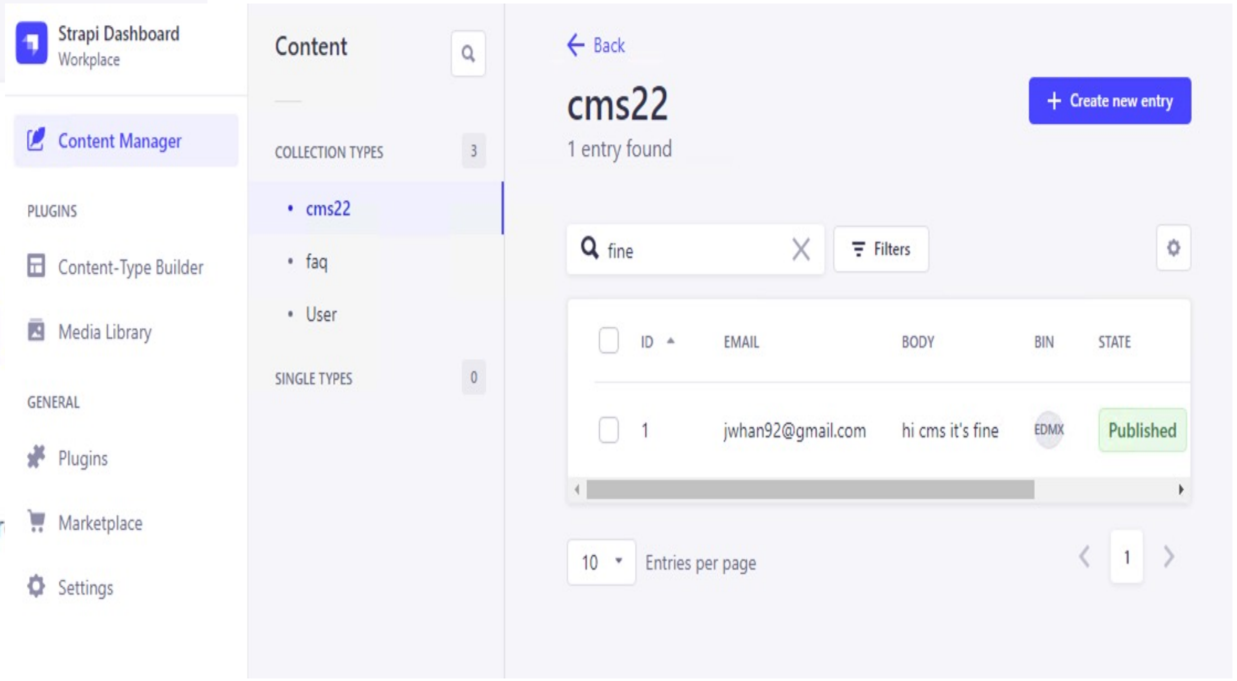
3 steps to get started

1 Build the content structure

Go to the Content type Builder

2 What would you like to shar

3 See content in action



cms22

1 entry found

fine

ID	EMAIL	BODY	BIN	STATE
1	jwhan92@gmail.com	hi cms it's fine	EDMX	Published

10 Entries per page 1

Hands-on

CAP on Kyma

Create CAP sample project

1. Install VS code : https://code.visualstudio.com/docs/setup/setup-overview
2. Install Node/NPM

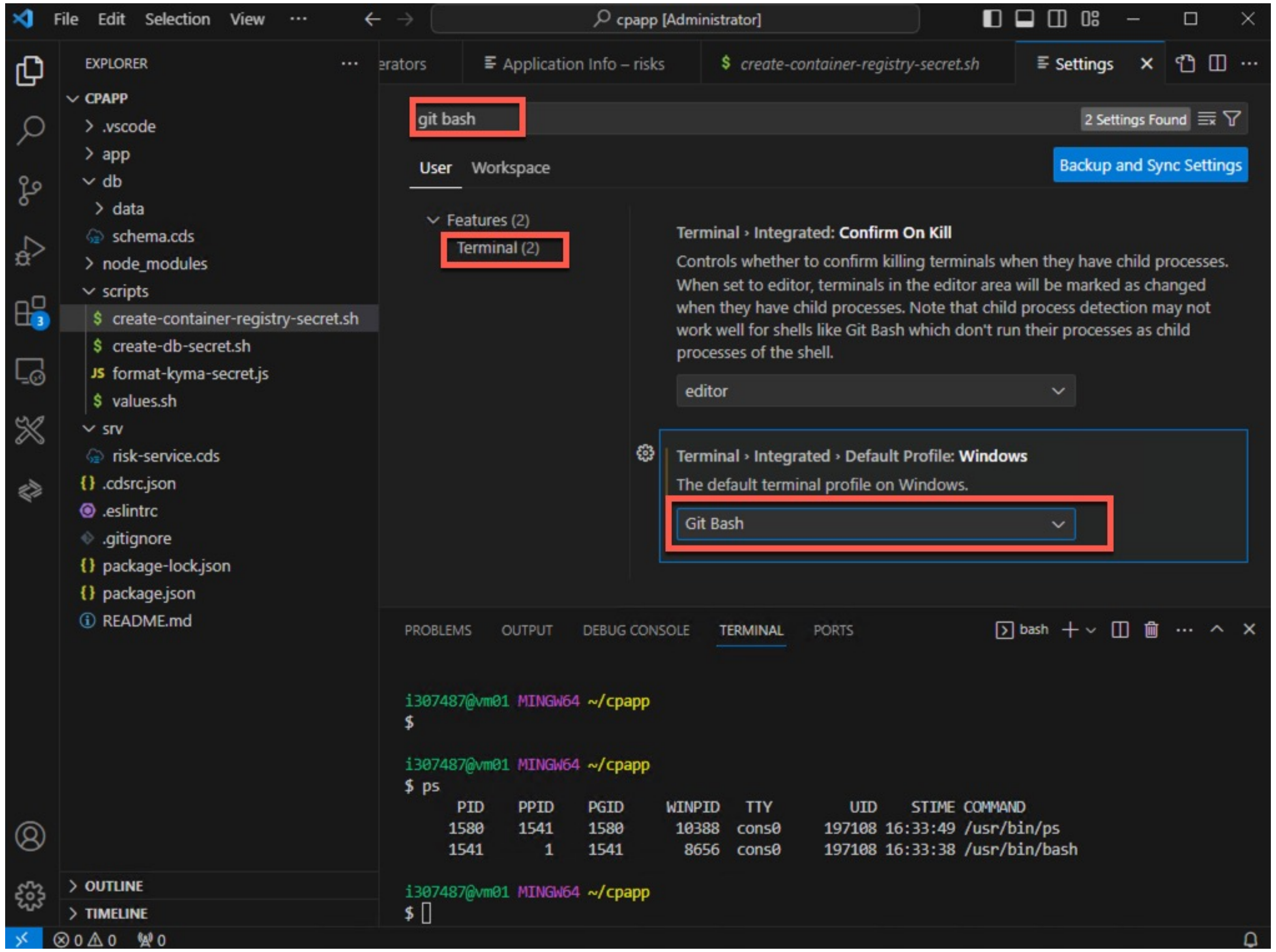
   $> npm install npm@latest –g
3. CAP developer kit 구성

   $> npm install @sap/cds-dk –g
4. Empty project 생성

   $> mkdir CAPImage
5. Windows OS : Terminal profile 조정 →
6. VS code Terminal open
7. CAP Project 생성

   $> cds init –add tiny-sample
8. CAP SQLite database 생성

   $> cds deploy –to sqlite:tutorial.db



git bash

Terminal › Integrated: Confirm On Kill

Controls whether to confirm killing terminals when they have child processes. When set to editor, terminals in the editor area will be marked as changed when they have child processes. Note that child process detection may not work well for shells like Git Bash which don't run their processes as child processes of the shell.

editor

Terminal › Integrated › Default Profile: Windows

The default terminal profile on Windows.

Git Bash

```
i307487@vm01 MINGW64 ~/cpapp
$

i307487@vm01 MINGW64 ~/cpapp
$ ps
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
     1580    1541    1580      10388  cons0     197108 16:33:49 /usr/bin/ps
     1541       1    1541       8656  cons0     197108 16:33:38 /usr/bin/bash

i307487@vm01 MINGW64 ~/cpapp
$ 
```

Create CAP sample project

9. Package.json 파일 수정 →

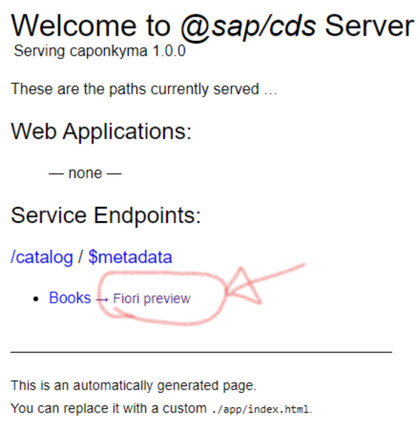
10. Project build

    $> cds build

11. Test →

    $> cds watch

```
"scripts": {
  "start": "cds-serve",
  "build": "cds build/all --clean"
}
```



Welcome to *@sap/cds* Server
Serving caponkyma 1.0.0

These are the paths currently served ...

Web Applications:

— none —

Service Endpoints:

/catalog / $metadata

- Books → Fiori preview

This is an automatically generated page.
You can replace it with a custom `./app/index.html`.

Create Docker image & Run container

12. Create Docker file. →

```
FROM node:lts-alpine3.16

# Create application directory
WORKDIR /usr/src/app

# Install application dependencies
COPY /package.json ./

RUN npm install

# Bundle app source
COPY /. .

EXPOSE 4004

CMD [ "npm", "start" ]
```

13. Create docker image

```
$> docker build -t jwhan92/capdemo20:latest -f Dockerfile .
```

맥북 M-series ARM 64인 경우 , ➔ 반드시 추가 “ --platform linux/amd64 ”

14. Run Docker image

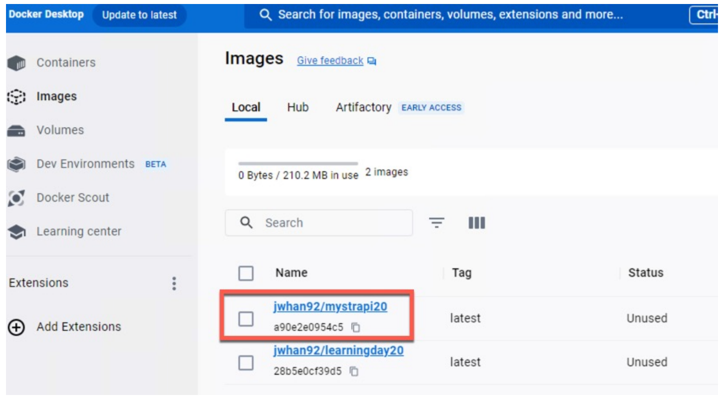
```
$> docker run –d –p 4004:4004 jwhan92/capdemo20:latest
```

15. Container 확인

```
$> docker ps
```

http://localhost:4004

```
$> docker images
```



0 Bytes / 210.2 MB in use 2 images

Name	Tag	Status
jwhan92/mystrapi20 a90e2e0954c5	latest	Unused
jwhan92/learningday20 28b5e0cf39d5	latest	Unused

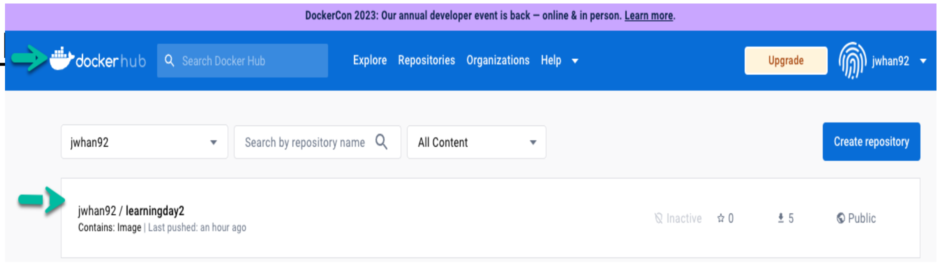
16. docker registry에 push 전 파일 생성 : .dockerignore →

```
npm-debug.log
default-env*.json
Dockerfile
README.md
```

Push the image to docker hub

17. Push the image into docker hub (image registry)

$> docker push jwhan92/capdemo20. → 결과는 왼쪽 docker hub에서 확인


DockerCon 2023: Our annual developer event is back – online & in person. Learn more.

jwhan92 / learningday2
Contains: Image | Last pushed: an hour ago
Inactive ☆ 0 ⬇ 5 Public

Deploy CAP app(image) into BTP Kyma

<< 사전 준비 사항 >>

BTP Trial Account 구성 →

Download kubeconfig.yaml (KubeconfigURL)

오른쪽과 같이 사용자 홈 디렉토리에 .kube 디렉토리 생성. kubeconfig.yaml 파일을 config로 이름 변경.

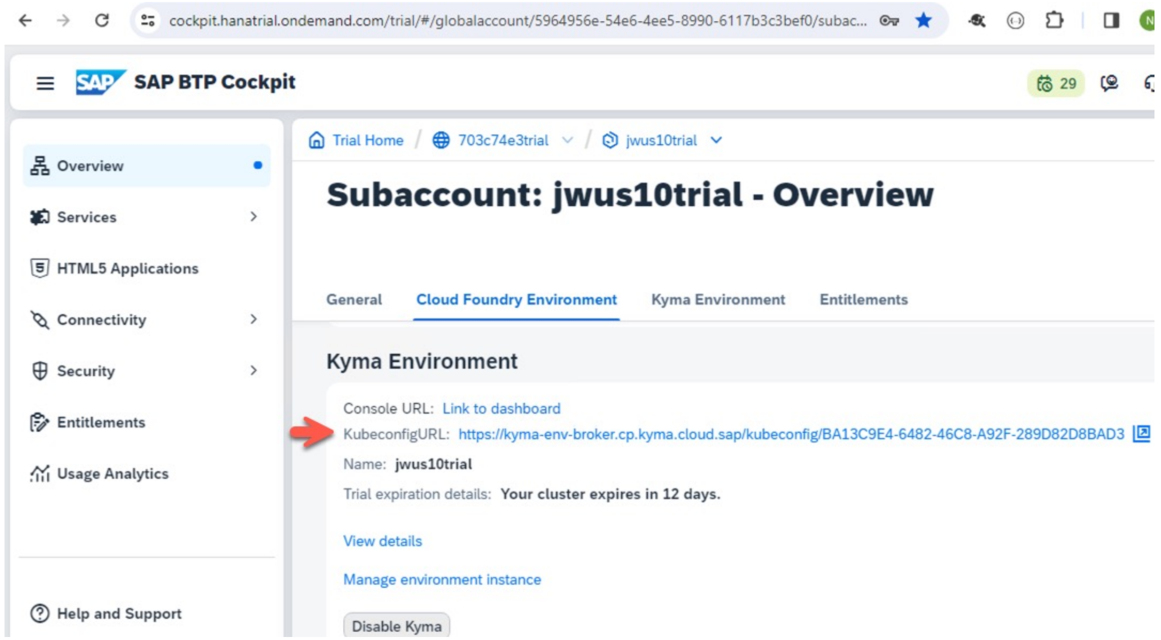
Krew 인스톨 링크 참조

Kubectl 에 oidc plugin 설치

```
$> kubectl krew install oidc-login
```

Windows 환경에서는 아래와 같이

```
$> kubectl krew install oidc-login -v=5
```

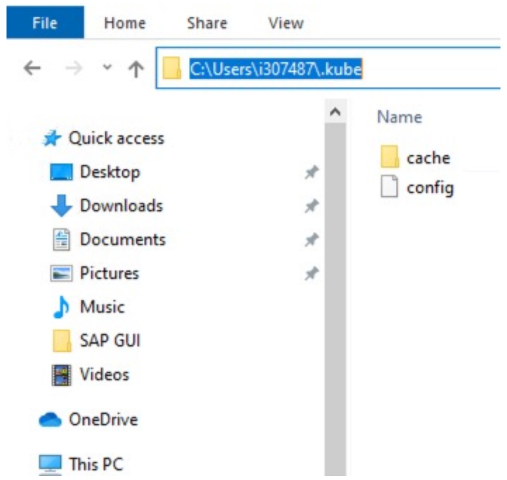


cockpit.hanatrial.ondemand.com/trial/#/globalaccount/5964956e-54e6-4ee5-8990-6117b3c3bef0/subac...
SAP BTP Cockpit
Overview
Services
HTML5 Applications
Connectivity
Security
Entitlements
Usage Analytics
Help and Support
Trial Home / 703c74e3trial / jwus10trial
General
Cloud Foundry Environment
Kyma Environment
Entitlements
View details
Manage environment instance
Disable Kyma

Subaccount: jwus10trial - Overview

Kyma Environment

Console URL: Link to dashboard
KubeconfigURL: https://kyma-env-broker.cp.kyma.cloud.sap/kubeconfig/BA13C9E4-6482-46C8-A92F-289D82D8BAD3
Name: jwus10trial
Trial expiration details: Your cluster expires in 12 days.



Thank you.

Contact information:

F name L name
Title
Address
Phone number

THE BEST RUN SAP