DOCKER

For Beginner

By Chatri Ngambenchawong

Who am I

Name: Chatri Ngambenchawong (ping)
Role : Dev (dotnet)
System Analyst
DevOps
Blogger

More [About Me | naiwaen@DebuggingSoft]

Find me on:

Agenda

Prerequisite: Install Docker Desktop / Account Docker Hub

Day1

- Why Container ?
- What is a Container ?
- Docker Architecture and Components
- Magic of Container, Why Linux ?
- Basic Docker Command
- Container - Resource Monitoring & Limit
- Container - Networking
- Container - Store Data

Day2

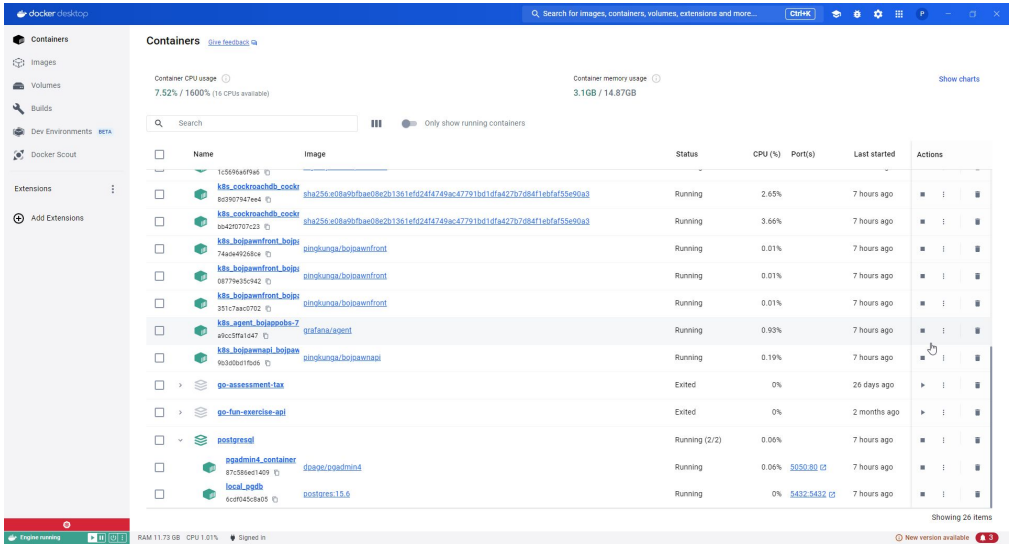
- Building Container (Dockerfile)
- Container Best Practice
- Docker Compose
- Portainer
- Wrap Up

Resource

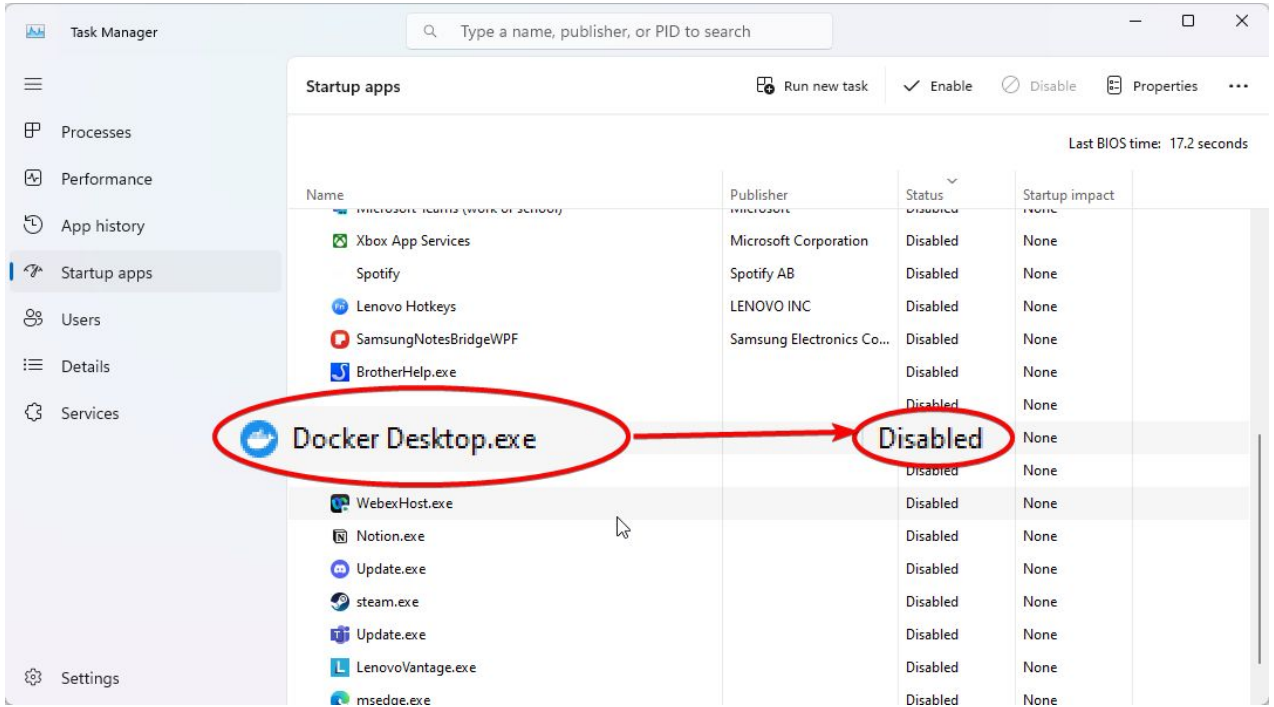
pingkunga/train-docker-at-BRSU (github.com)

Prerequisite

- docker hub account
- docker
  - Docker Desktop (Windows)
  - Install Docker Engine on Ubuntu | Docker Docs
- terminal (Optional)
  - **windows terminal**
  - command prompt

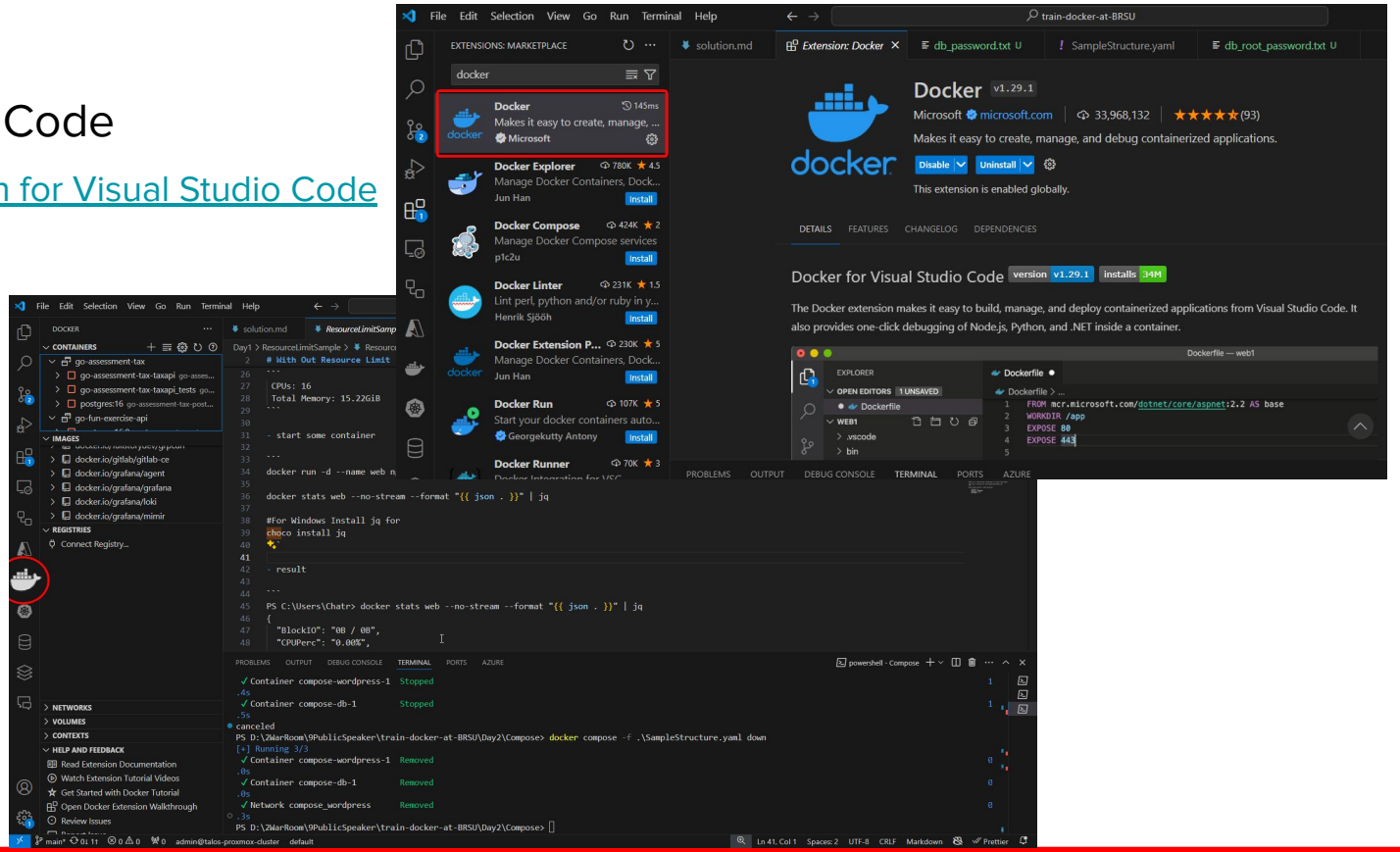


Prerequisite



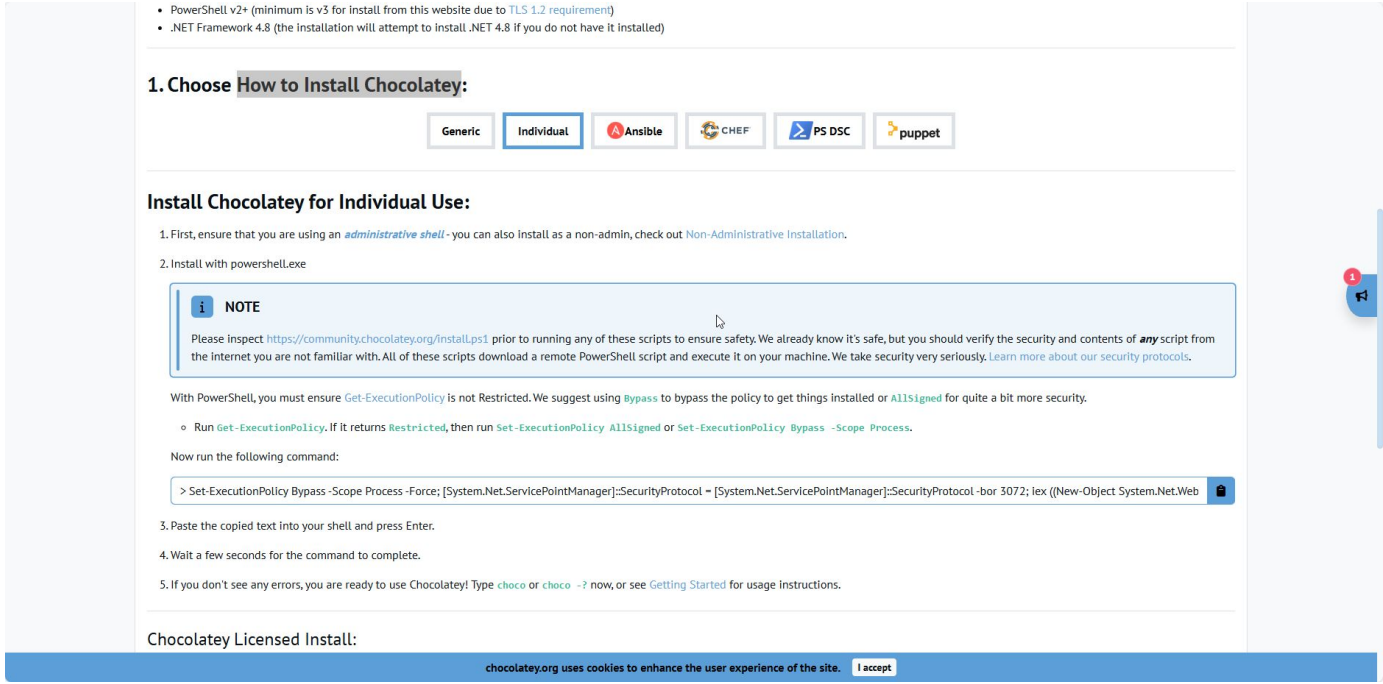
Prerequisite

- Text Editor VS Code
  - [Docker extension for Visual Studio Code]
- Git (Optional)



Prerequisite

- How to Install Chocolatey



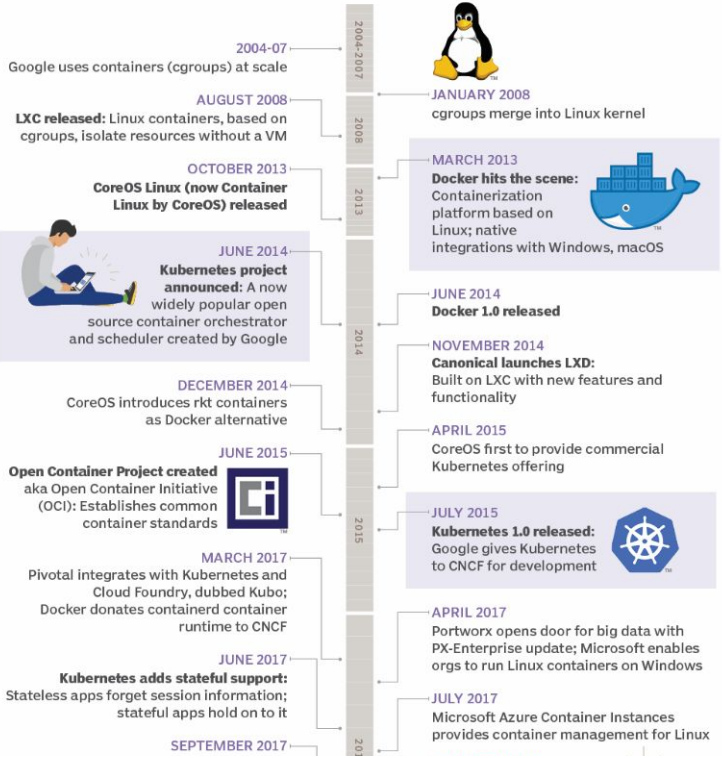
Prerequisite

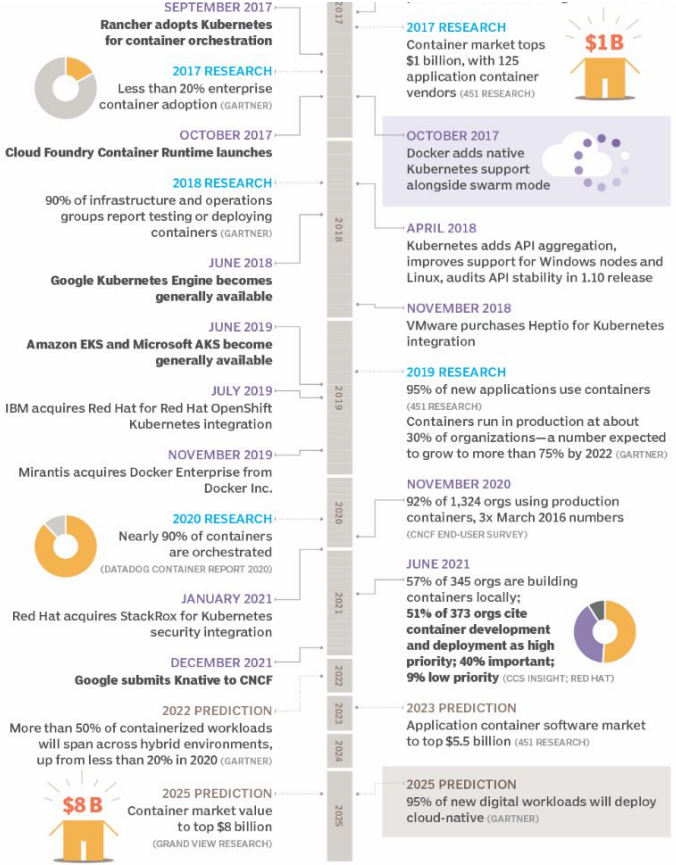
- docker
  - Docker Desktop (Windows)
  - Install Docker Engine on Ubuntu | Docker Docs
- docker hub account
- Test Editor VS Code
- Git

Malcolm McLean (Container for Shipping)

The evolution of containers

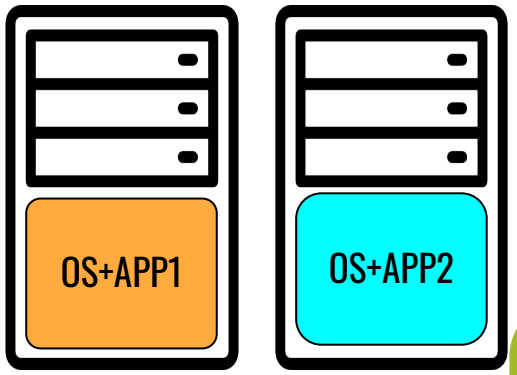
Container technology has come a long way from its chroots, starting with Google's exploration into cgroups and working up into widespread organizational adoption.





Ref: The evolution of containers: Docker, Kubernetes and the future | TechTarget / Something Missed? : History of Container Technology | by Ozan Eren | Medium

Why Container ?

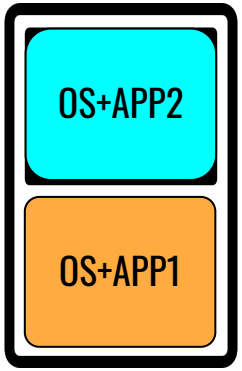


1

BARE MERAL

- **Full Control Everything**
- **Downside**
  - More App, More Hardware
  - Waste of Resource utilize
  - Hard to maintain - security/ backup / patch / Install App

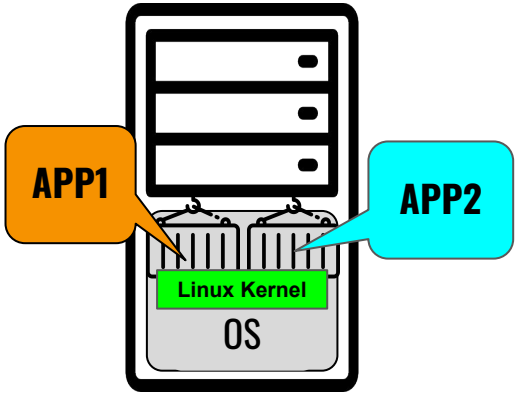
2



VIRTUAL MACHINE

- **Hardware Share: Full Control in VM**
  - Hardware Cost Savings
  - Application isolation
  - Improved scalability & availability
  - Centralized administration & security
- **Downside**
  - Resource utilize
  - Hard to maintain - Install App

3

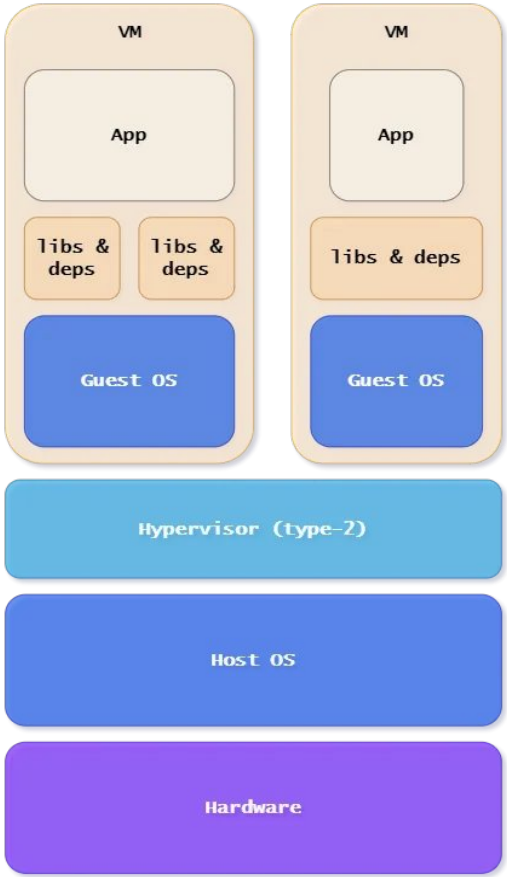


CONTAINER

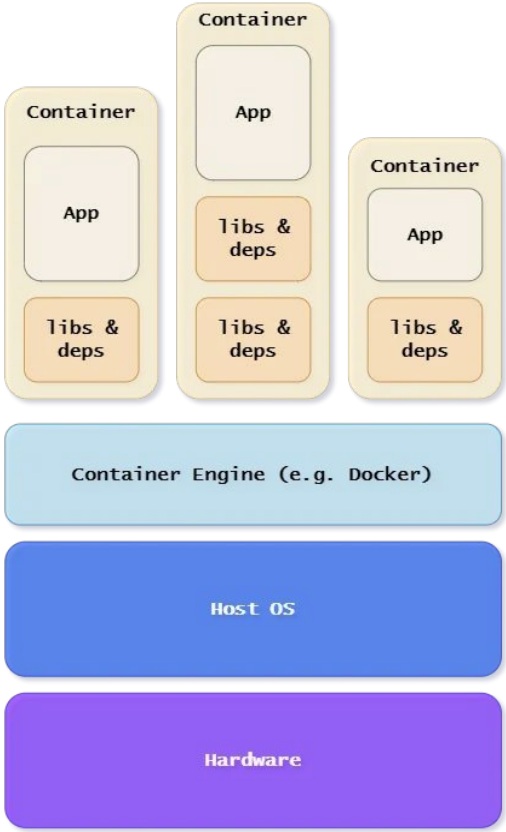
- **Hardware + Kernel Share**
  - **More Cost Savings**
  - Application isolation
  - Improved scalability & availability
  - Centralized administration & security
  - Install App Config Runtime
- **Downside**
  - Best for Linux Base
  - Complexity
  - Limited Functionality - Cloud Native

Why Container ?

BARE METAL



VIRTUAL MACHINE



CONTAINERS

Container Runtime

High Level /
Low Level

podman

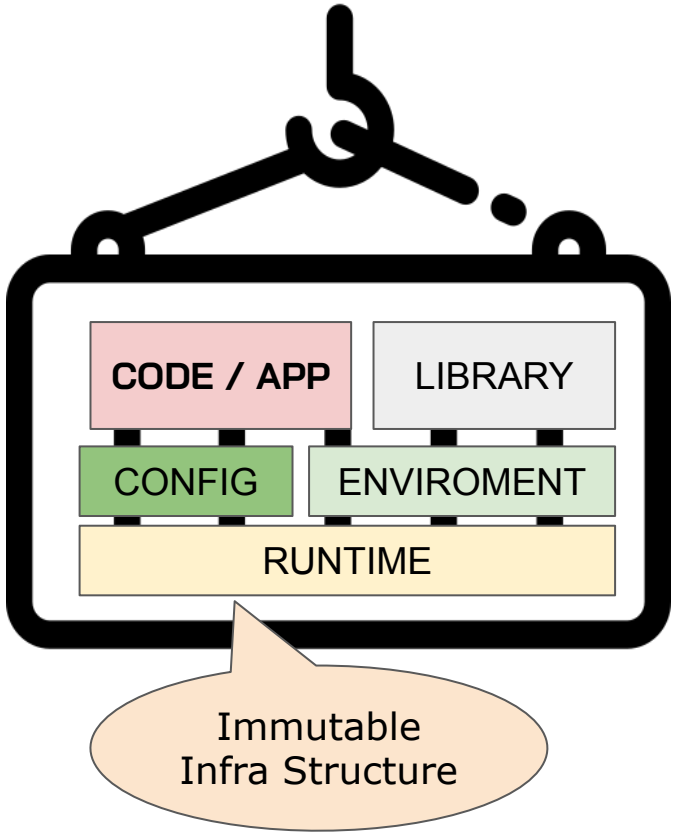
Linux Container (LXC)

What is a Container ?

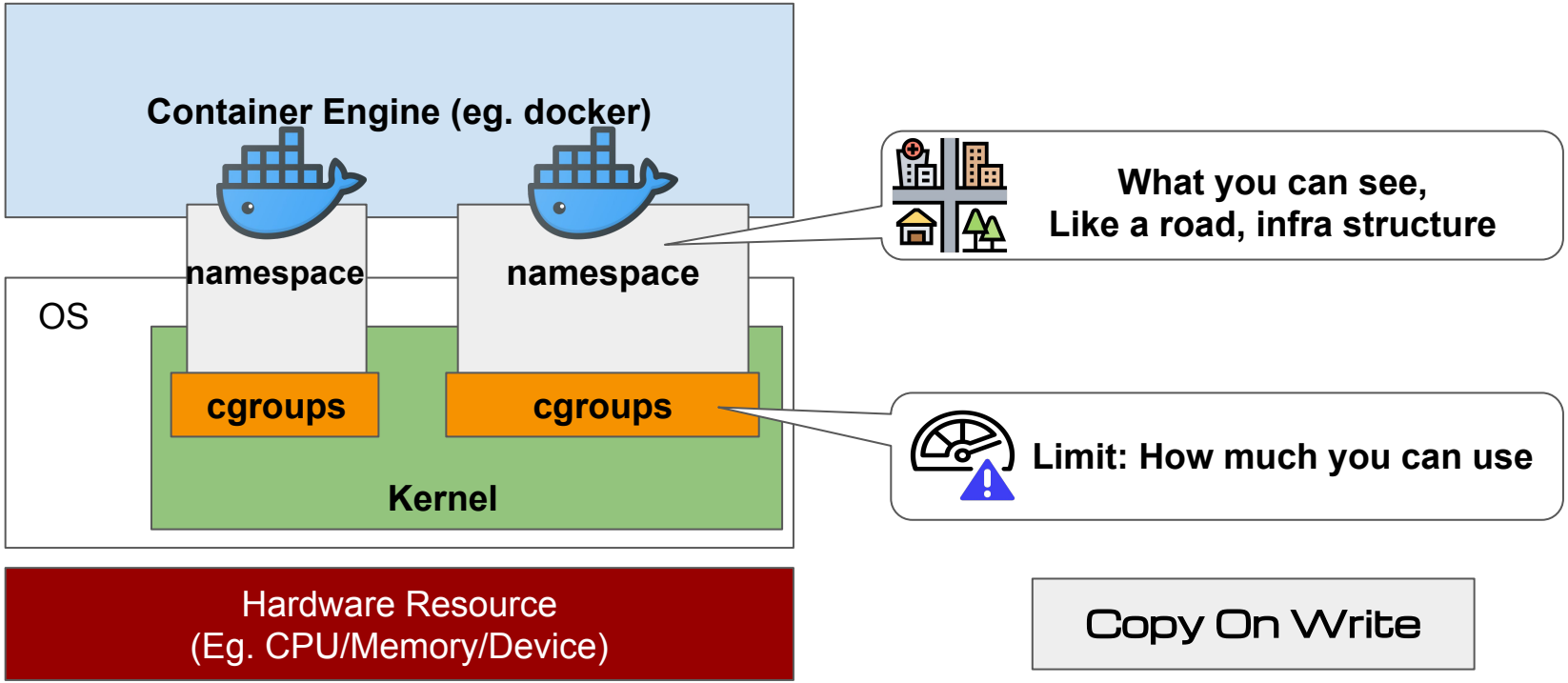
Containers are packages of software that contain all of the necessary elements to run in any environment

Benefits of using Containers

- Efficiency
- Application Isolation - and its dependencies from the host system
- Portability
- Separation of Responsibility - Dev + Ops / Create / Deploy
- Faster Application Development - CI / CD
- Easy Scaling

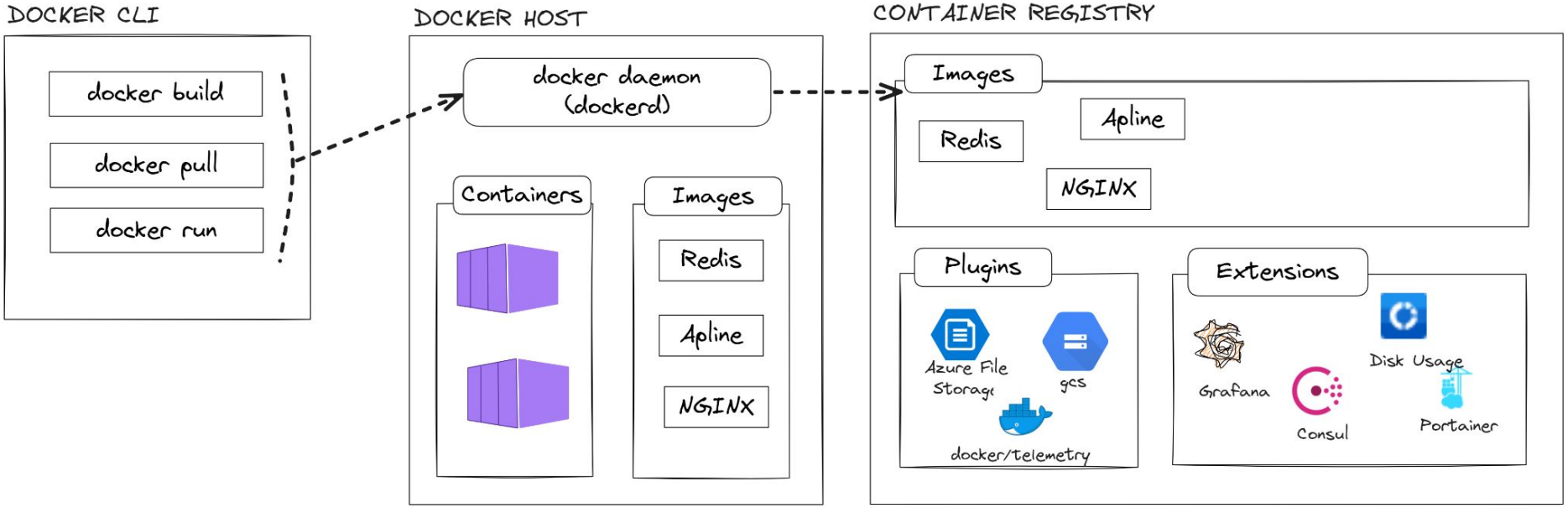


Magic of Container, Why Linux ?



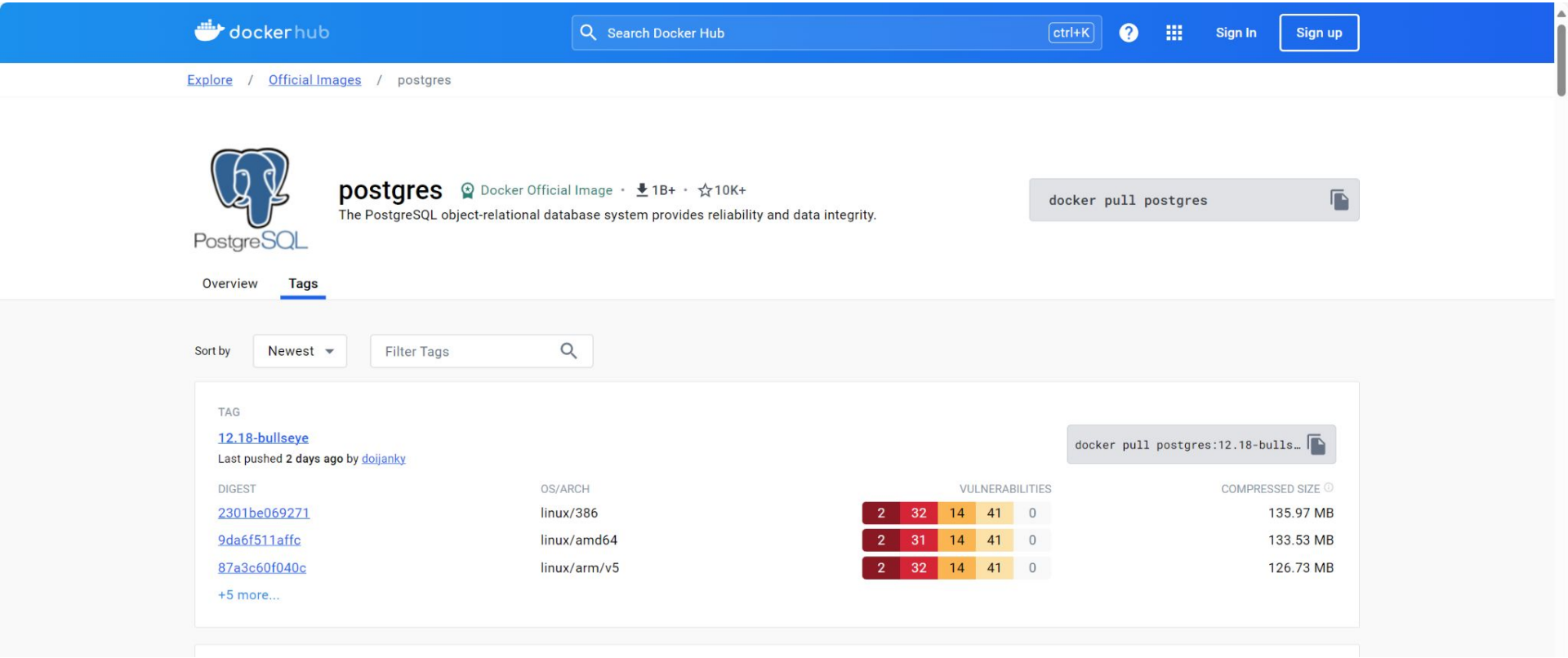
Ref: [linux - difference between cgroups and namespaces - Stack Overflow]

Docker Architecture and Components



Docker Architecture

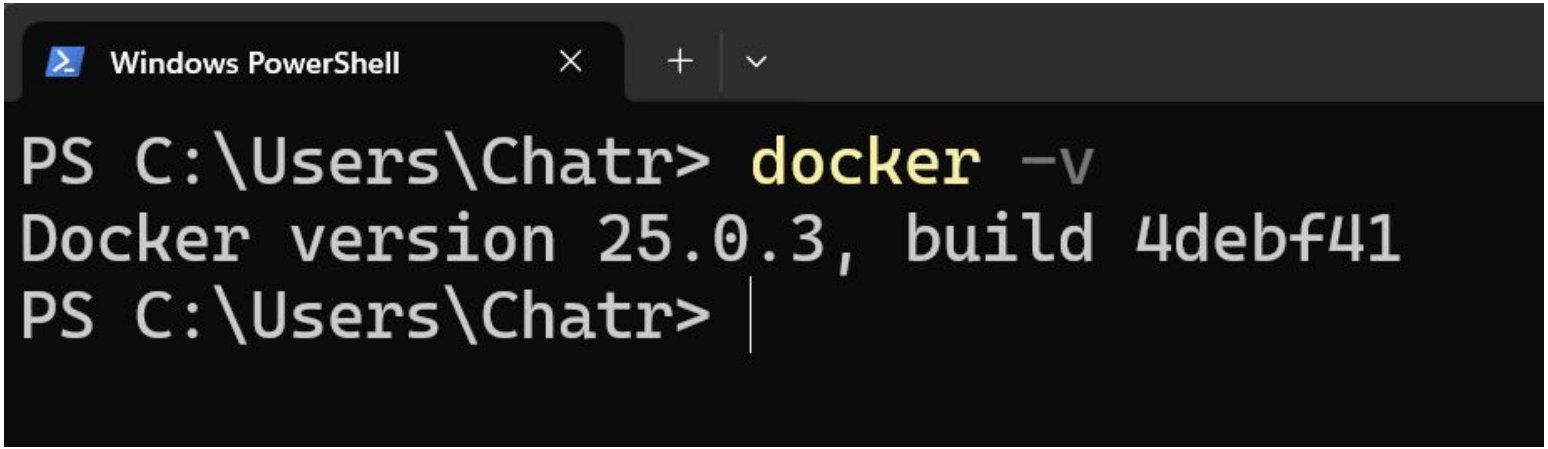
- **Docker Client (cli)**- Send Command via TCP (2375)
- **Docker Host** - Server Docker (เครื่องที่ลง docker)
  - Docker Deamon (dockerd) - process for handle request from docker cli eg.
  1. Pull Image from docker regiastry (If Notcvexist)
  2.Create Container
  ...
  - Image - Local Storage
- **Container Registry** (eg. docker hub / RedHat Quay)
  - Share Image / Free / Not Free (Private Repo)
  - Increase Automation
  - Other - Share Extension / Plugin


dockerhub
Search Docker Hub
ctrl+K
Sign In
Sign up
Explore / Official Images / postgres
PostgreSQL
postgres
Docker Official Image · 1B+ · 10K+
The PostgreSQL object-relational database system provides reliability and data integrity.
docker pull postgres
Overview
Tags
Sort by
Newest
Filter Tags
TAG
12.18-bullseye
Last pushed 2 days ago by doijanky
docker pull postgres:12.18-bulls…
DIGEST
OS/ARCH
VULNERABILITIES
COMPRESSED SIZE
2301be069271
linux/386
2 32 14 41 0
135.97 MB
9da6f511affc
linux/amd64
2 31 14 41 0
133.53 MB
87a3c60f040c
linux/arm/v5
2 32 14 41 0
126.73 MB
+5 more...

Basic Docker Command

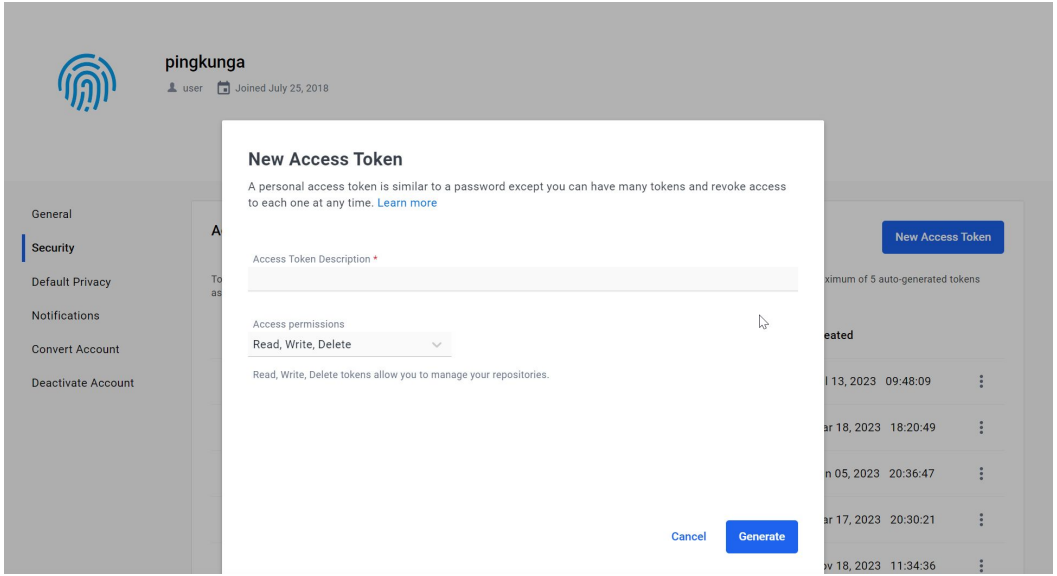
Check Version

```
docker -v
```

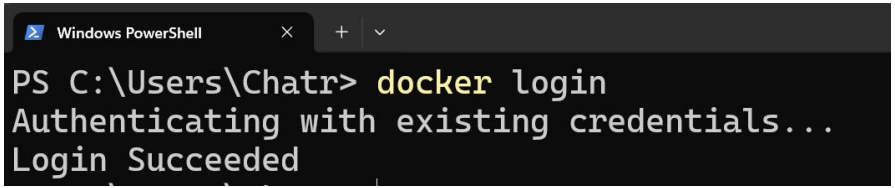


docker login

- Login docker hub
  - Generate token first : https://hub.docker.com/settings/security
  - Keep Token
  - docker login with user + token




```
docker login
username: <docker username>
password: <Token>
```



docker logout

- Logout from docker hub

```
docker logout
```

ปกติไม่ค่อยได้ใช้ รอ Session หมดมากกว่า หรือไม่ไป revoke token

docker login / logout
another registry eg. Nexus / gchr.io / ACR

```
docker login <your_registry_url>
username: <your_username>
password: <Token>
```

```
docker logout <registry_url>
```

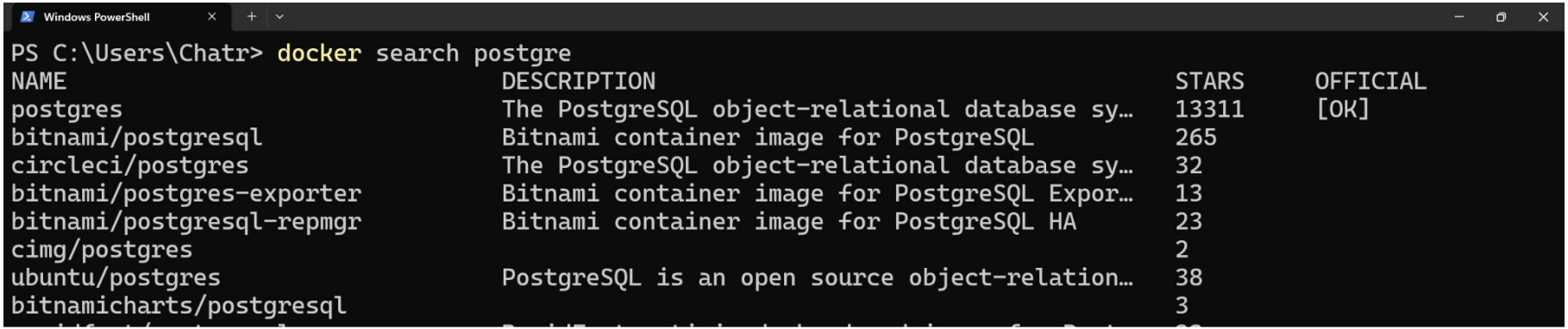
Nexus: [sonatype/nexus3 - Docker Image | Docker Hub]

docker search

- Search Image

ปกติไม่ค่อยได้ใช้เหมือนกัน
ดูจาก hub เอามากกว่า

```
docker search <KEYWORD>
```

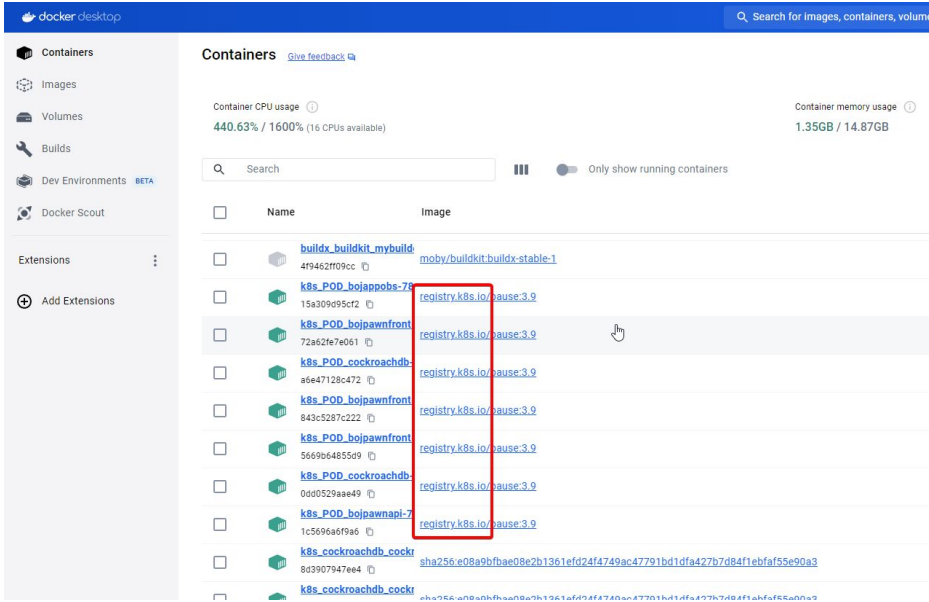


Container Image Name Pattern

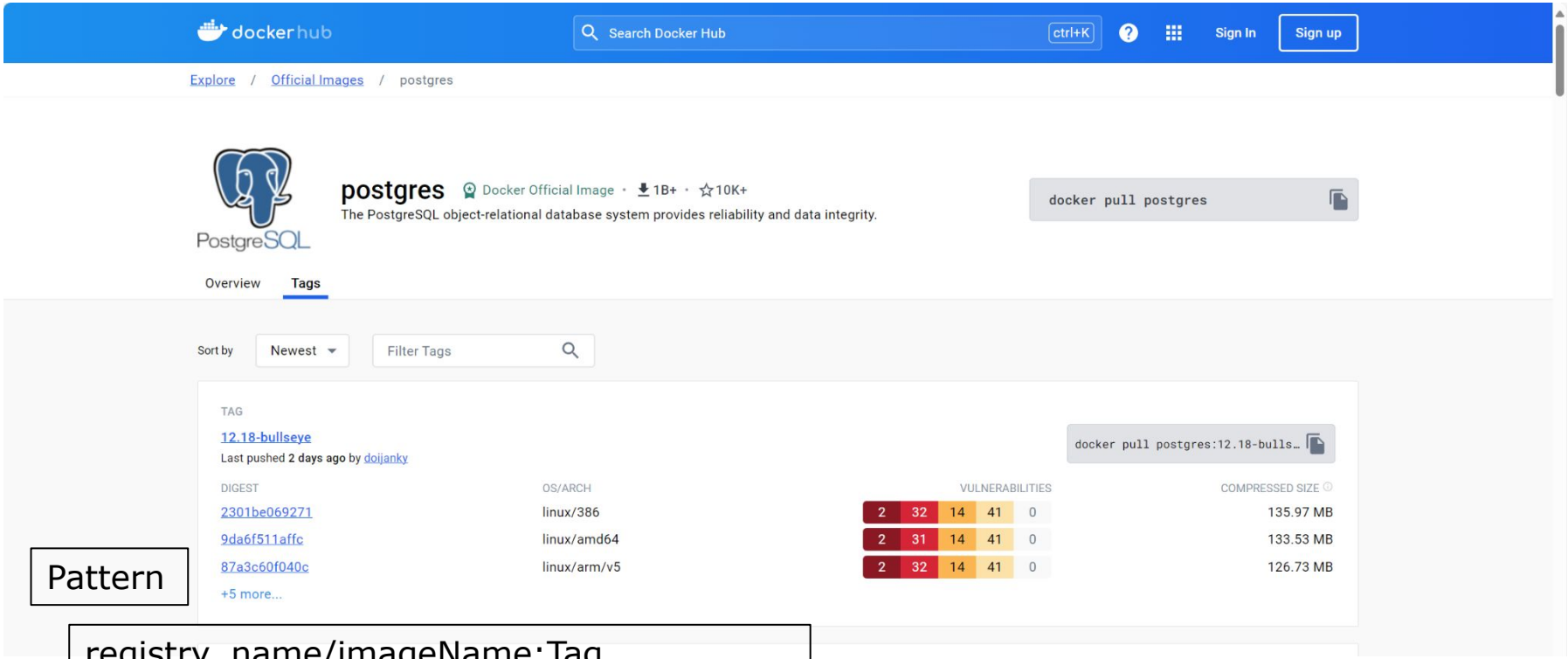
- Pattern

registry_name/ImageName:Tag

ImageName:Tag
hub.docker.com/ImageName:Tag



Explore Docker's Container Image Repository | Docker Hub



Pattern

registry_name/imageName:Tag

imageName:Tag

docker pull

- Download Image from Container Registry to Local

```
docker pull <registry_url><image_name>:<tag>
docker pull <image_name>:<tag>
```

```
docker pull postgres:16.2-bookworm
```

```
docker pull redis
```

```
docker pull registry.k8s.io/kube-apiserver:v1.29.2
```

```
docker pull yourcompany.com/invs-apiserver:8.23.1
```

```
docker pull ghcr.io/someorg/someimage:sometag
```

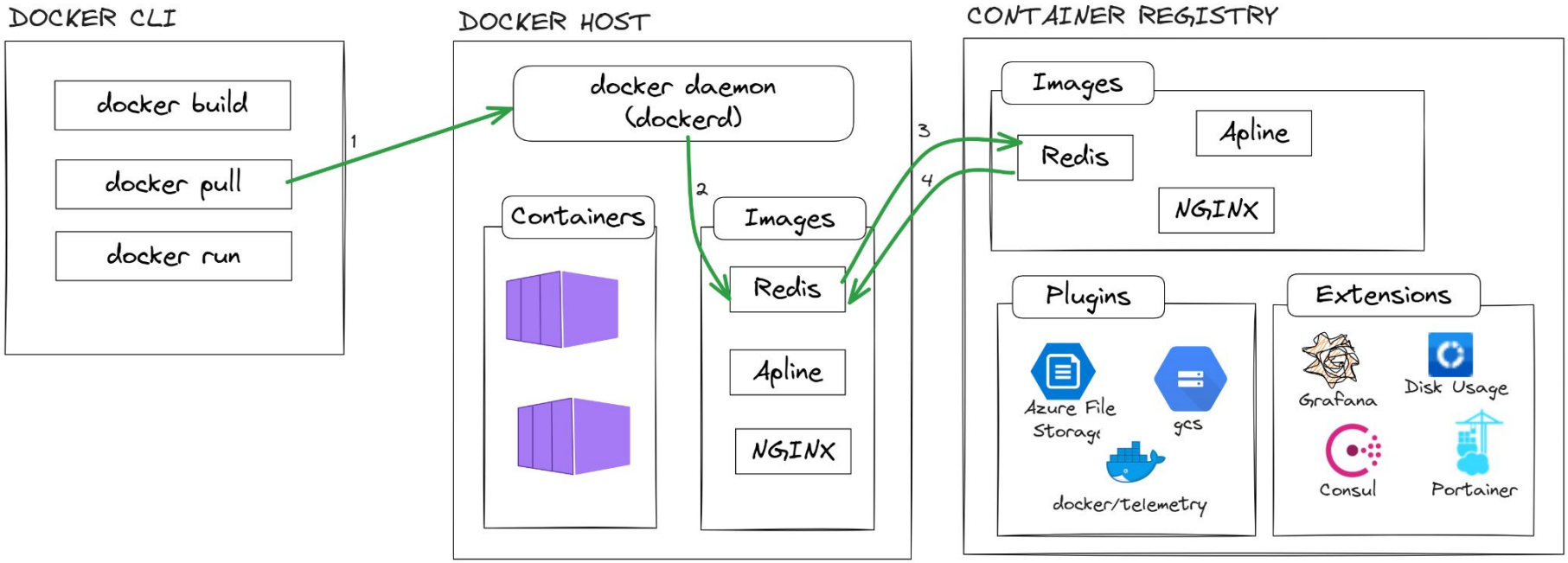
docker pull - image type

```
docker pull postgres:16.2-bookworm
```

Base Image:
- Alpine
- bookworm (ubuntu)
- Slim vs Minimal
- Secured by Design: distroless / chiselled / chainguard
- etc.

Docker Pull : How it works

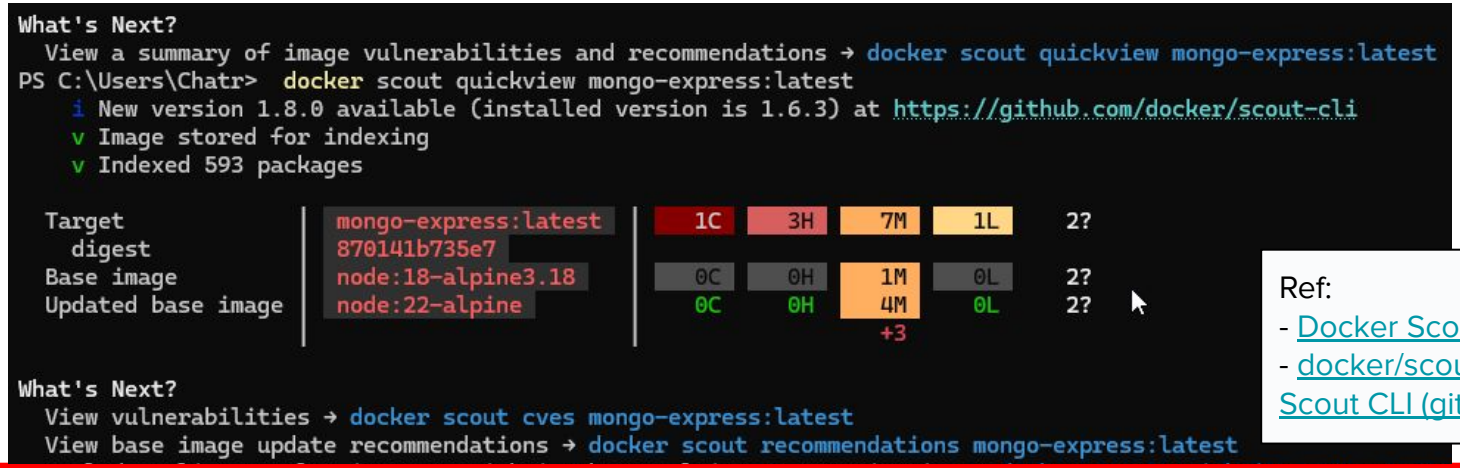
docker pull redis



docker scout

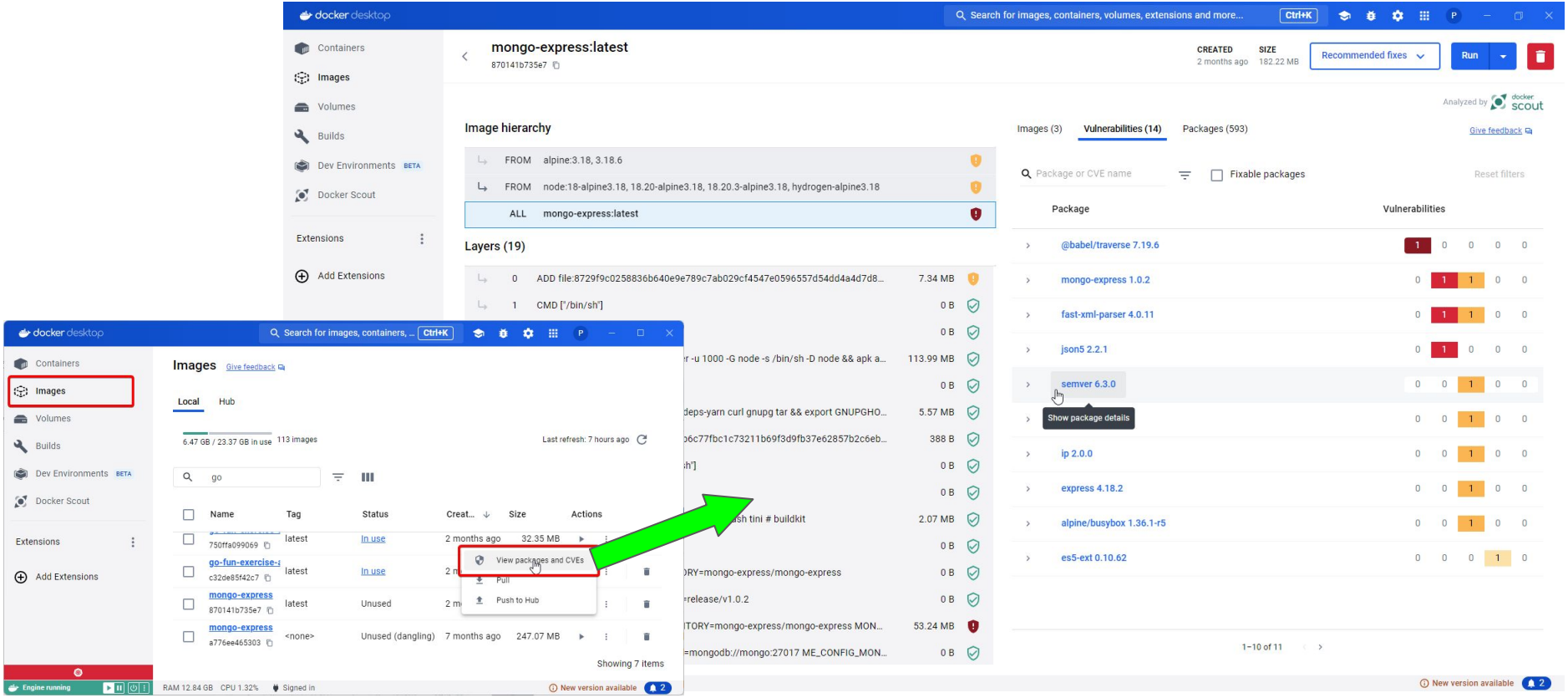
- A tools analyzes image contents and generates a detailed report of packages and vulnerabilities that it detects. It can provide you with suggestions for how to remediate issues discovered by image analysis.

```
docker scout quickview <image_name:tag>
docker scout quickview mongo-express:latest
```



Ref:
- Docker Scout | Docker Docs
- docker/scout-cli: Docker Scout CLI (github.com)

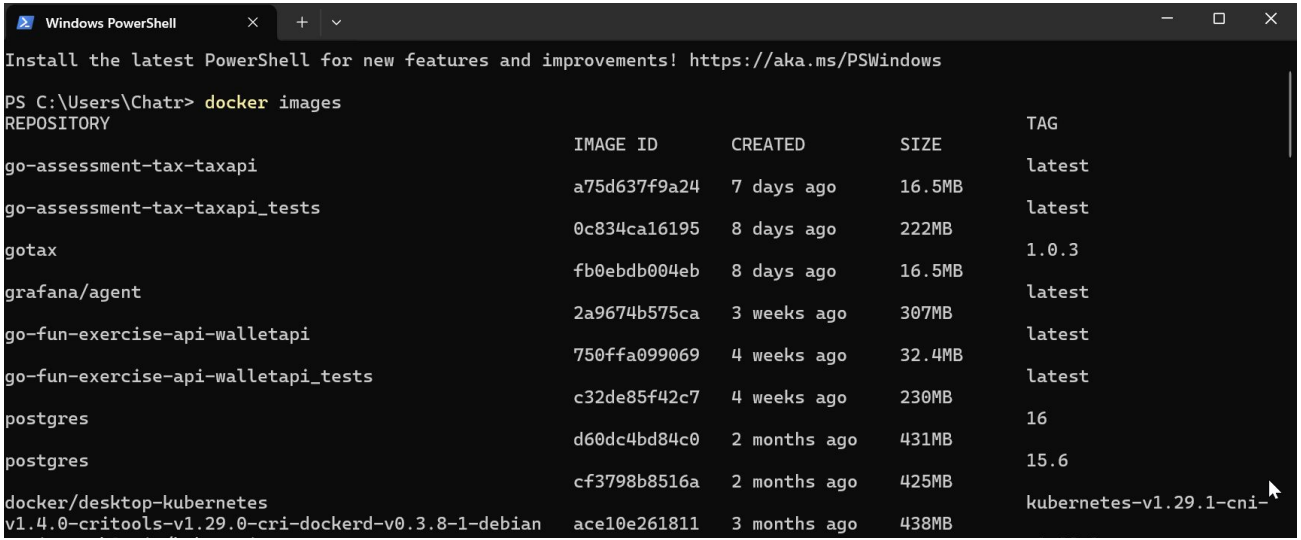
docker scout - Docker Desktop



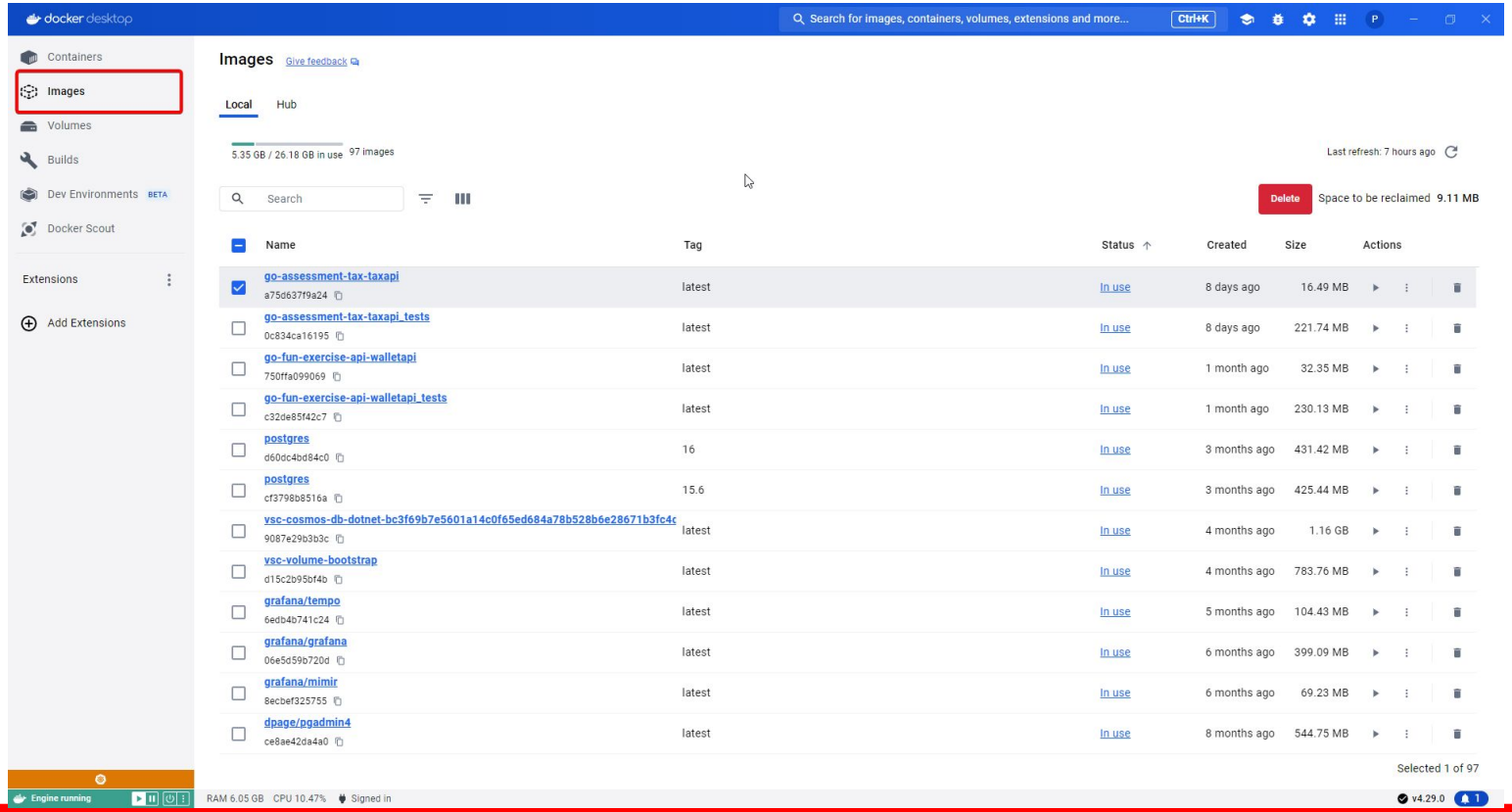
docker images

- List image at local

```
docker images
docker images --no-trunc // Show full image id
```



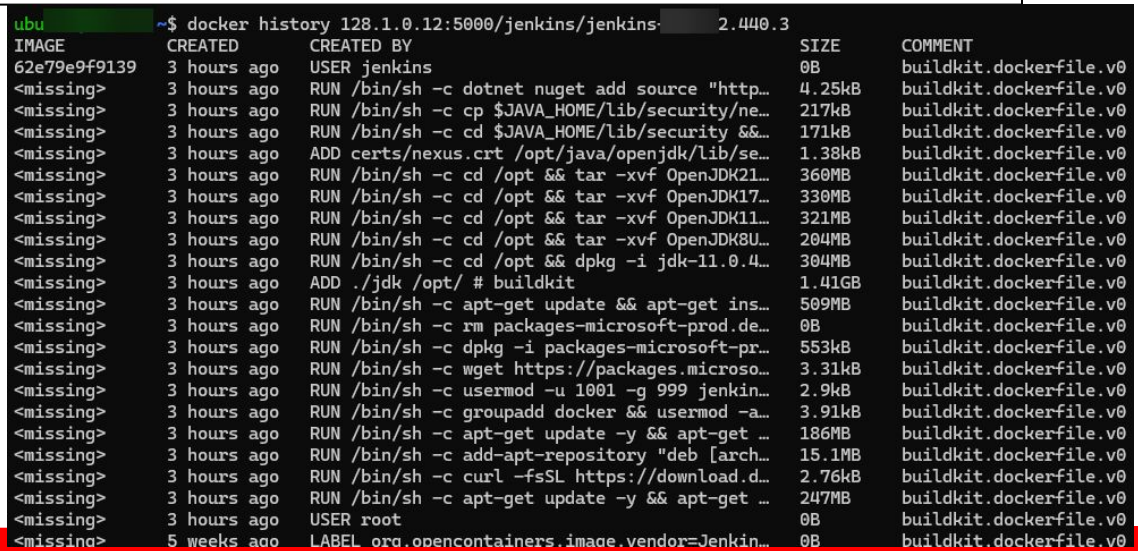
docker images - docker desktop



docker history

- Show the history of an image

```
docker history <repo_url><image_name>:<image_tag>
docker history jenkins:2.440.3
docker history --no-trunc 128.1.0.12:5000/jenkins/jenkins-ds:2.440.3
```



```
ubu          ~$ docker history 128.1.0.12:5000/jenkins/jenkins-    2.440.3
IMAGE          CREATED       CREATED BY                                      SIZE      COMMENT
62e79e9f9139   3 hours ago   USER jenkins                                    0B        buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c dotnet nuget add source "http…   4.25kB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cp $JAVA_HOME/lib/security/ne…   217kB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd $JAVA_HOME/lib/security &&…   171kB     buildkit.dockerfile.v0
<missing>      3 hours ago   ADD certs/nexus.crt /opt/java/openjdk/lib/se…   1.38kB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd /opt && tar -xvf OpenJDK21…   360MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd /opt && tar -xvf OpenJDK17…   330MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd /opt && tar -xvf OpenJDK11…   321MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd /opt && tar -xvf OpenJDK8U…   204MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c cd /opt && dpkg -i jdk-11.0.4…   304MB     buildkit.dockerfile.v0
<missing>      3 hours ago   ADD ./jdk /opt/ # buildkit                      1.41GB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c apt-get update && apt-get ins…   509MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c rm packages-microsoft-prod.de…   0B        buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c dpkg -i packages-microsoft-pr…   553kB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c wget https://packages.microso…   3.31kB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c usermod -u 1001 -g 999 jenkin…   2.9kB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c groupadd docker && usermod -a…   3.91kB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c apt-get update -y && apt-get …   186MB     buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c add-apt-repository "deb [arch…   15.1MB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c curl -fsSL https://download.d…   2.76kB    buildkit.dockerfile.v0
<missing>      3 hours ago   RUN /bin/sh -c apt-get update -y && apt-get …   247MB     buildkit.dockerfile.v0
<missing>      3 hours ago   USER root                                       0B        buildkit.dockerfile.v0
<missing>      5 weeks ago   LABEL org.opencontainers.image.vendor=Jenkin…   0B        buildkit.dockerfile.v0
```

docker run - start container / docker create

- Create Container + Run

```
docker run [options] <image_name>:<tag>
```

- Create Container Only !!

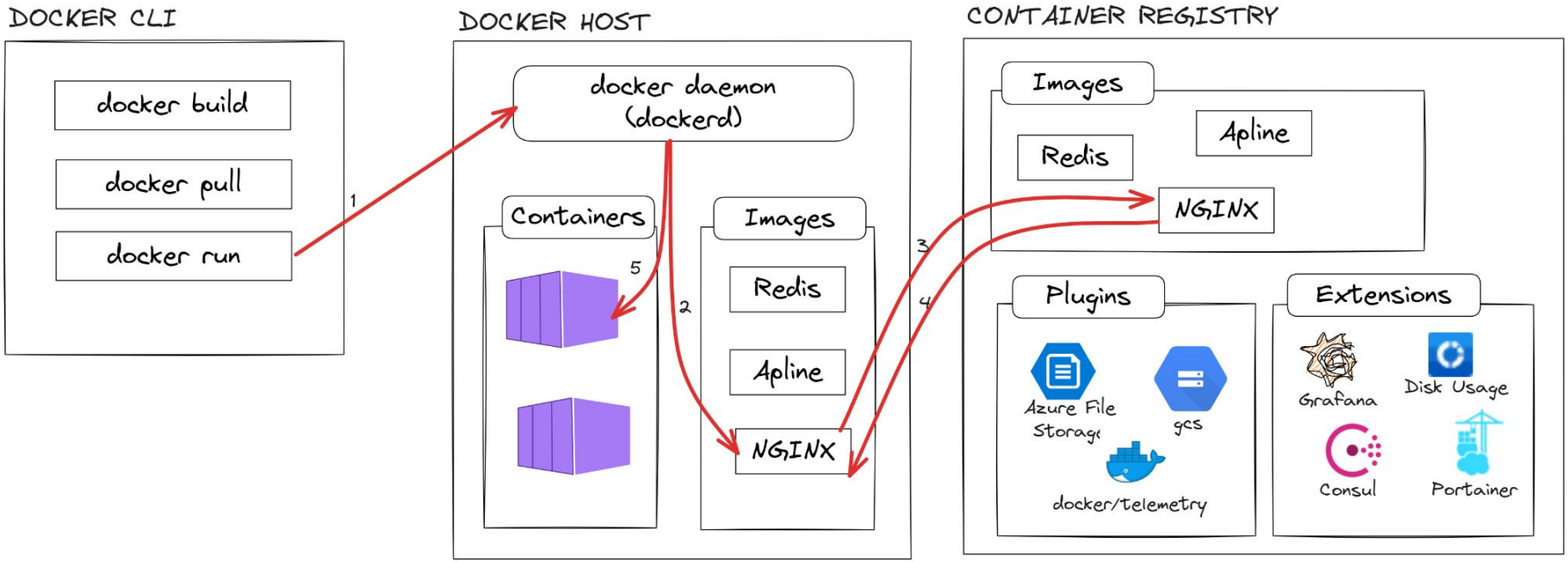
```
docker create [options] <image_name>:<tag>
```

[options]
- p <host>:<container port> mapping port
- d background mode (docker run only)
- v mount data between Host / Docker
- n set container name / default random
- e <key:value> set environment varaible
--network <network_name> set container network
--it interactive mode
--rm auto remove when stop
--label your.key=value
--restart containers retart policy

เราจะรู้ได้ยังไงว่าต้องใส่อะไรบ้าง ?

docker run: How it works

```
docker run --rm -d -p 8080:80 --name web nginx
```



docker run: How it works

```
docker run --rm -d -p 8080:80 --name web nginx
```

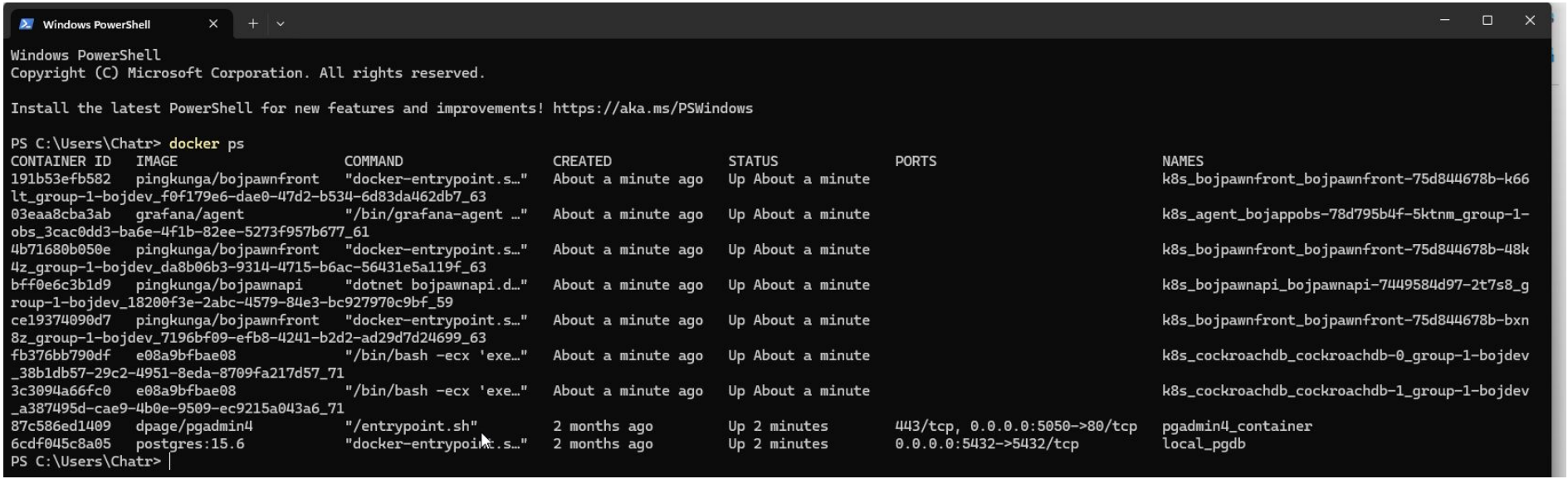
docker ps - list container

- List Running Container

```
docker ps
```

- List All Container (Run and Stop)

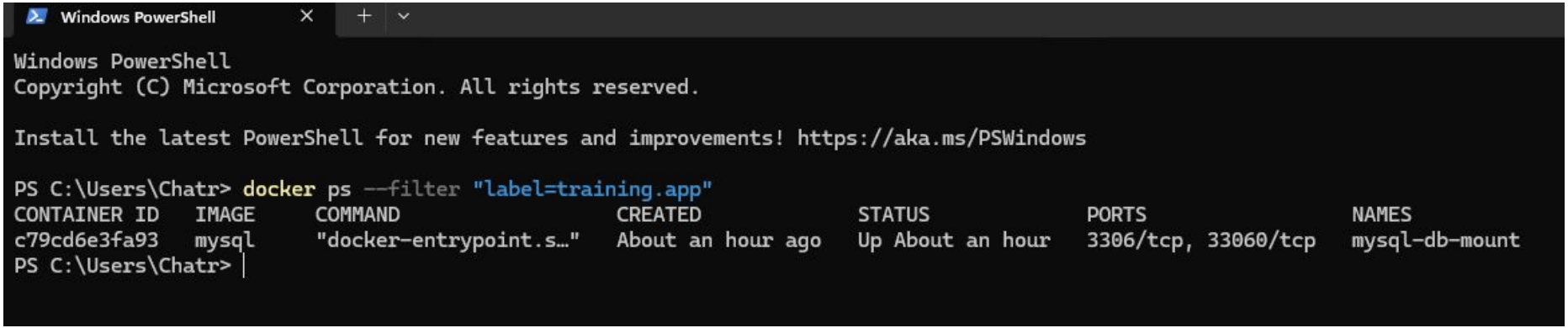
```
docker ps -a
```



docker ps - list container

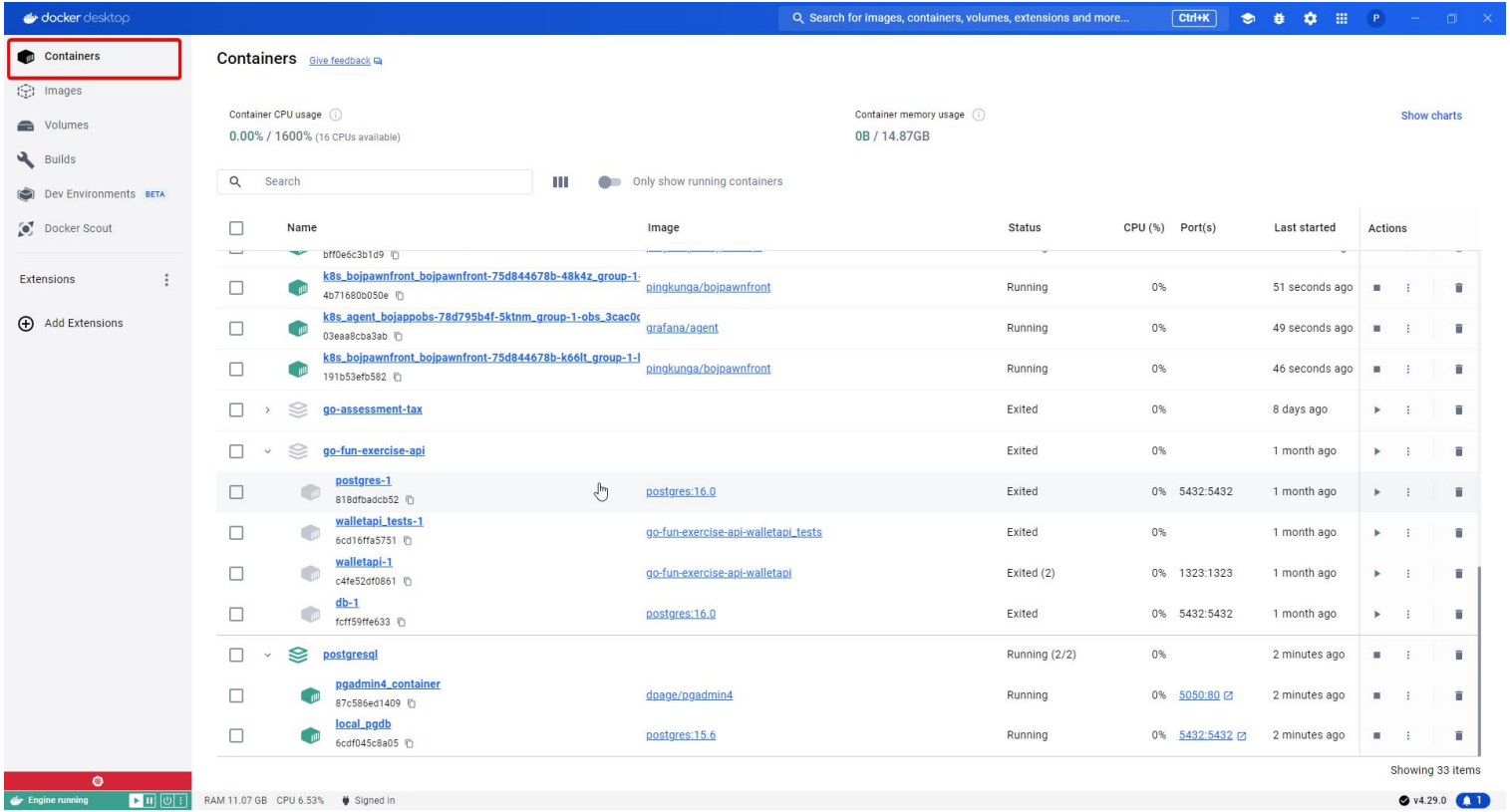
- List Container by Label

```
docker ps --filter "label=training.app"
```



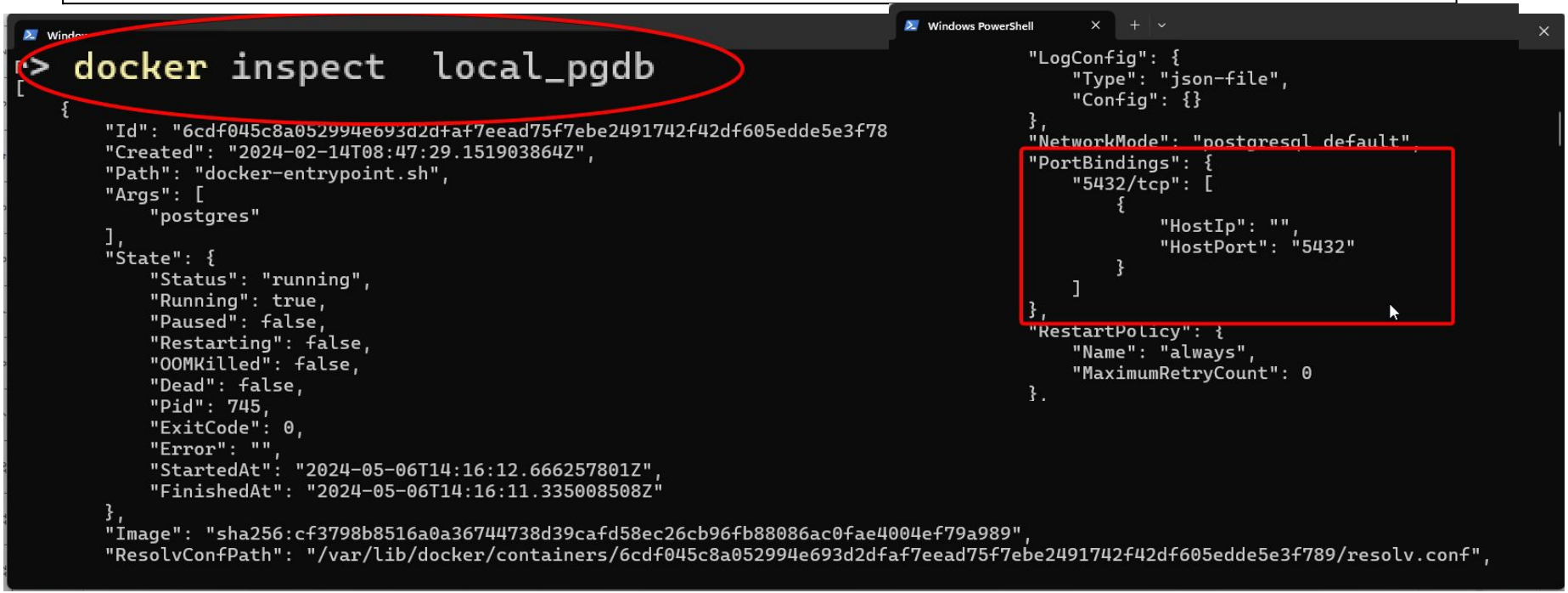
Ref: [docker ps --filter | Docker Docs]

docker ps - docker desktop



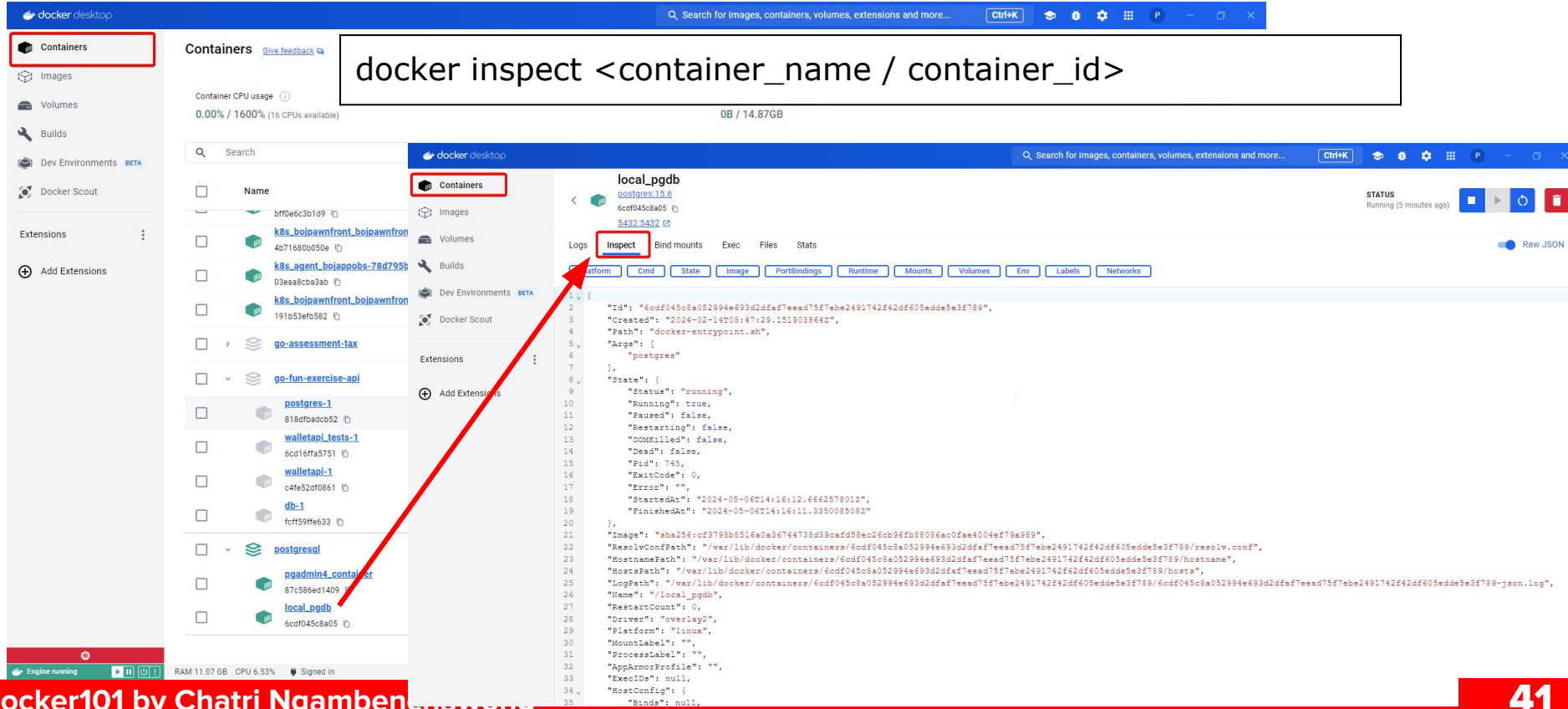
docker inspect - Get Container Info

docker inspect <container_name / container_id>



docker inspect - Docker Desktop

docker inspect <container_name / container_id>



Hand-on 1

Hand-On1: Get your first Container

- Run PostgreSQL Container with requirement
  - version: 15.6 + small image
  - username: docker101
  - password: dockerPwd
  - expose: 5432
  - container name: pg_dev_test
  - run background mode

- Create NGINX contain with requirement
  - expose: 18080
  - container name: webapp

- Check Container Status & Inspect

```
PS C:\Users\Chatr> docker ps -a
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS          PORTS
               NAMES
c761f0d189fe   nginx                  "/docker-entrypoint.…"   22 seconds ago   Created
               webapp
7fb7ec1847f1   postgres:15.6-alpine3.19  "docker-entrypoint.s…"   48 minutes ago   Up 47 minutes   0.0.0.0:5432->5432/tcp
               pg_dev_test
```

Hand-on 1 ....

Solution: GetYourFirstContainer.md at main · pingkunga(github.com)

Hand-On1: Get your first Container (Solution)

- Run PostgreSQL Container with requirement
  - version: 15.6 + small image
  - username: docker101
  - password: dockerPwd
  - expose: 5432
  - container name: pg_dev_test
  - run background mode

```
docker run -d -e POSTGRES_USER=docker101 -e POSTGRES_PASSWORD=dockerPwd
-p 5432:5432 --name pg_dev_test postgres:15.6-alpine3.19
```

Hand-On1: Get you first Container (Solution)

- Create NGINX container with requirement
  - expose: 18080
  - container name: webapp
  - auto remove when container stop

```
docker create --rm -p 18080:80 --name webapp nginx
```

- Check Container Status & Inspect

```
docker ps -a

#inspect
docker inspect webapp
docker inspect pg_dev_test
```

Basic Docker Command #02

docker exec - Access Container

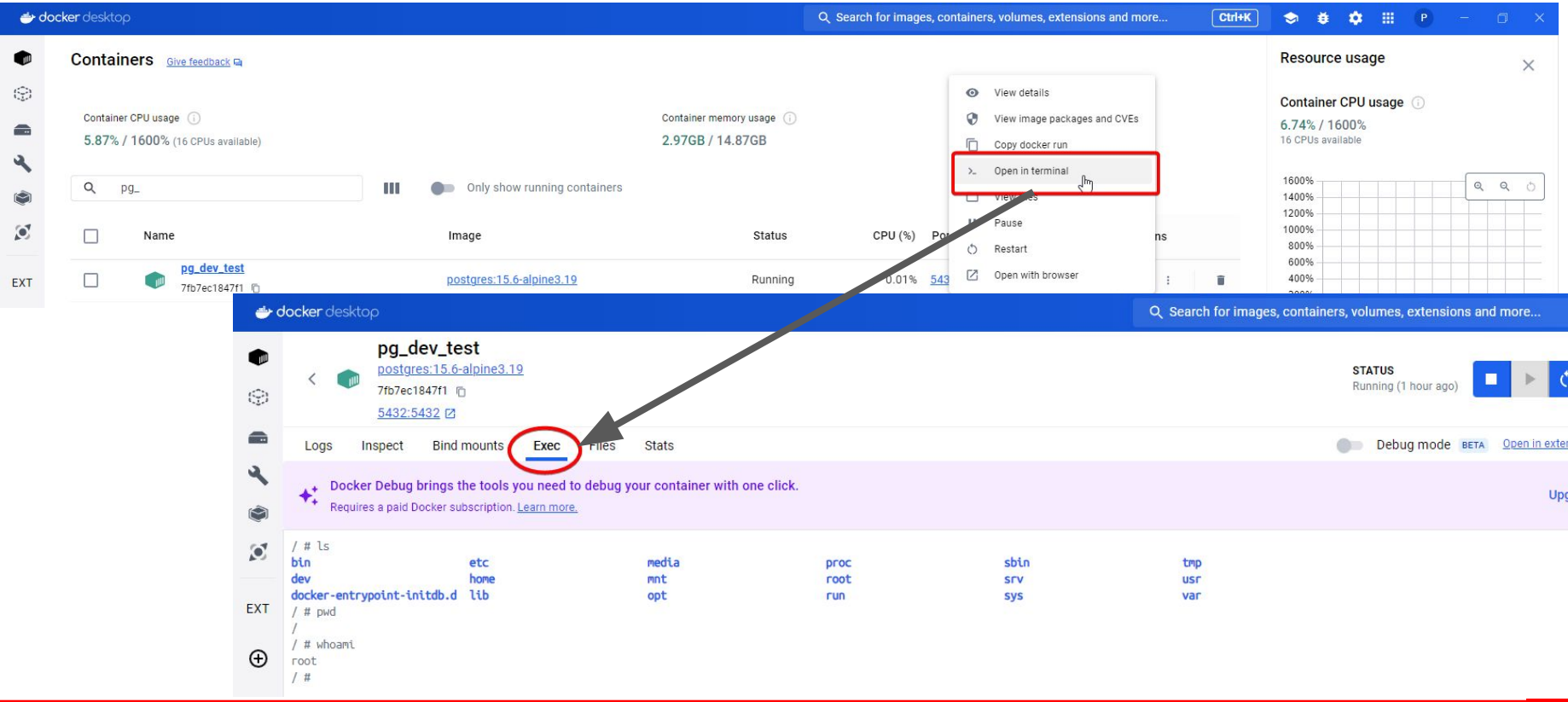
```
docker exec -it <container_name / container_id> <command>
```

Run Command in a Container - such as create file / manage app / other operation etc.

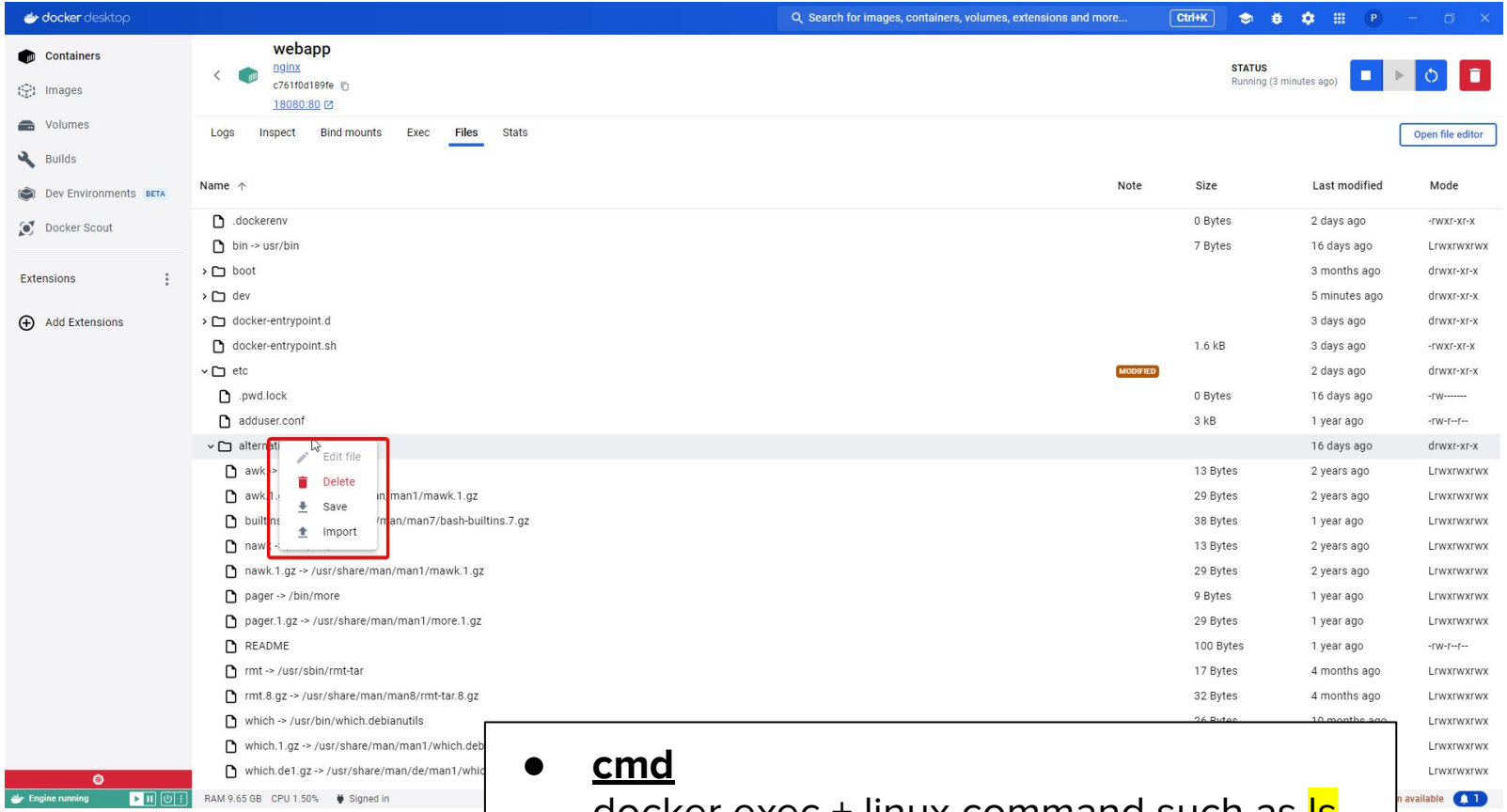
Access PostgreSQL (Hand-on1)

```
docker exec -it pg_dev_test /bin/bash
psql -U docker101
docker101-# \l
                                  List of databases
   Name    |   Owner   | Encoding |  Collate   |   Ctype    | ICU Locale | Locale Provider |    Access privileges
-----------+-----------+----------+------------+------------+------------+-----------------+-------------------------
 docker101 | docker101 | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
 postgres  | docker101 | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            |
 template0 | docker101 | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            | =c/docker101          +
           |           |          |            |            |            |                 | docker101=CTc/docker101
 template1 | docker101 | UTF8     | en_US.utf8 | en_US.utf8 |            | libc            | =c/docker101          +
           |           |          |            |            |            |                 | docker101=CTc/docker101
(4 rows)

docker101-# \q
7fb7ec1847f1:/# exit
```

docker exec - Docker Desktop



Inside docker



- **cmd**
  docker exec + linux command such as ls

docker cp - transfer file / folder between host ⇔ container

- Copy File

```
#FROM HOST TO CONTAINER
docker cp [OPTIONS] [SRC_PATH TO FILE] [CONTAINER_ID:DEST_PATH + FILE]

#FROM CONTAINER TO HOST
docker cp [OPTIONS] [CONTAINER_ID:SRC_PATH_TO FILE] [DEST_PATH OF HOST+ FILE]
```

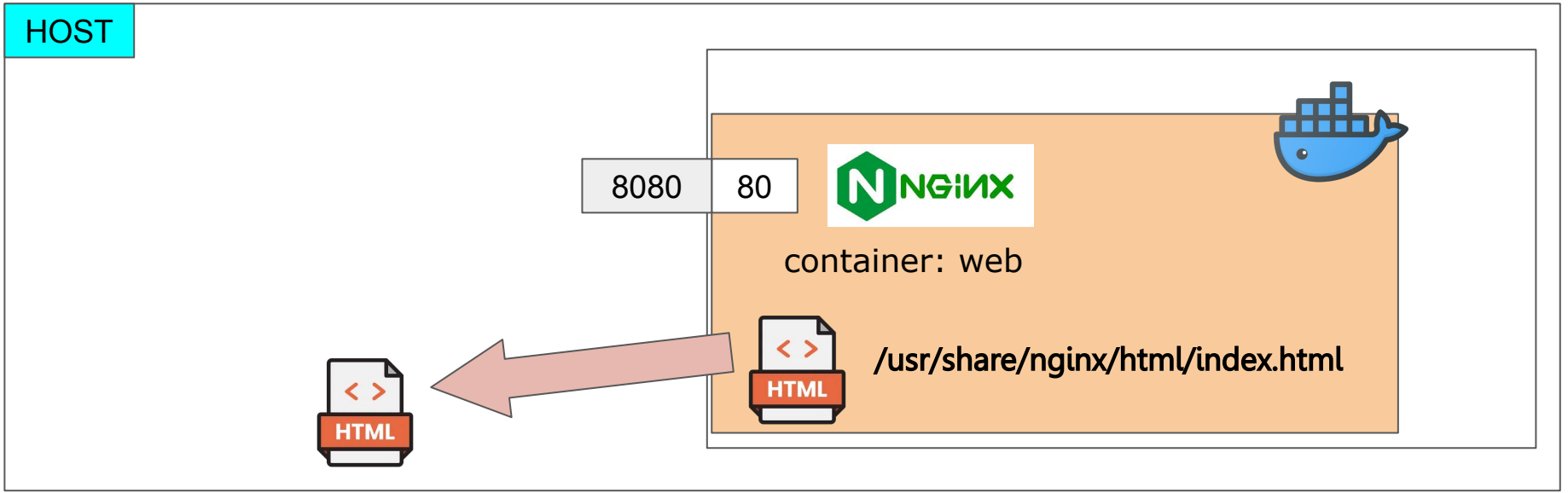
- Copy Folder

```
#FROM HOST TO CONTAINER
docker cp [OPTIONS] [SRC_PATH OF HOST] [CONTAINER_ID:DEST_PATH]

#FROM CONTAINER TO HOST
docker cp [OPTIONS] [CONTAINER_ID:SRC_PATH] [DEST_PATH OF HOST]
```

docker cp: Visualize

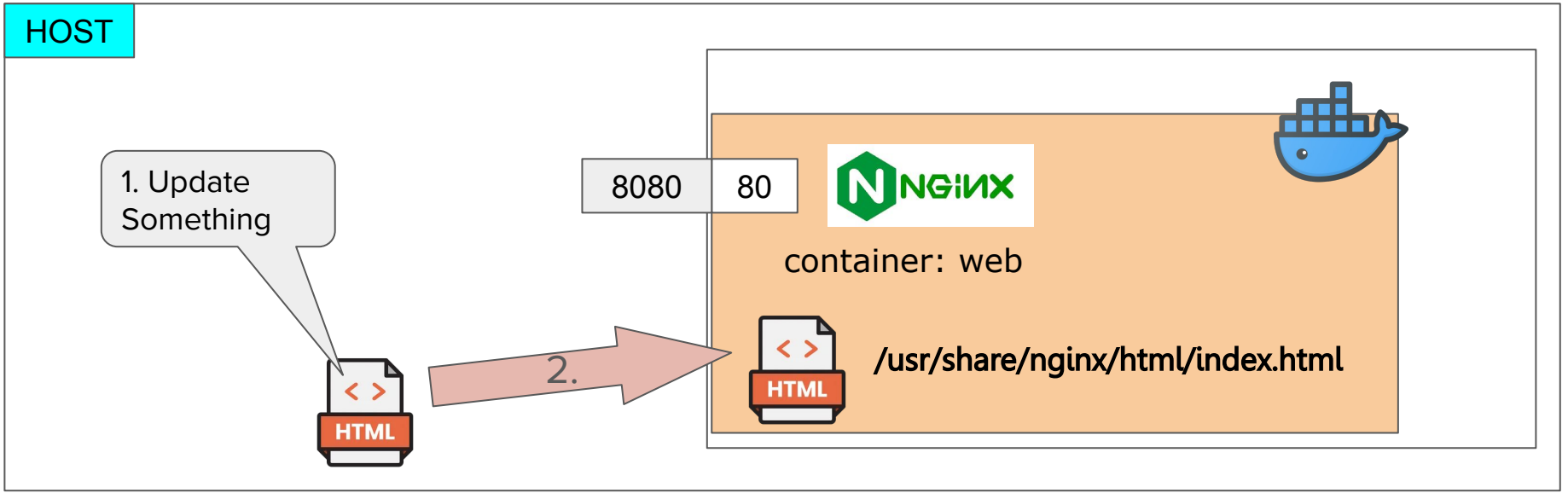
Current Dir

```
docker cp webapp:/usr/share/nginx/html/index.html .
```



docker cp: Visualize

```
docker cp index.html webapp:/usr/share/nginx/html/index.html
```



Control Container State - 1

- start container

```
docker start <container_name / container_id>
```

- stop container (graceful shutdown)

```
docker stop <container_name / container_id>
```

- restart container (stop + start)

```
docker restart <container_name / container_id>
```

- kill - kill hang process

```
docker kill<container_name / container_id>
```

Control Container State - 2

- pause - container process / keep resouce (Stop Clear Resource)

```
docker pause <container_name / container_id>
```

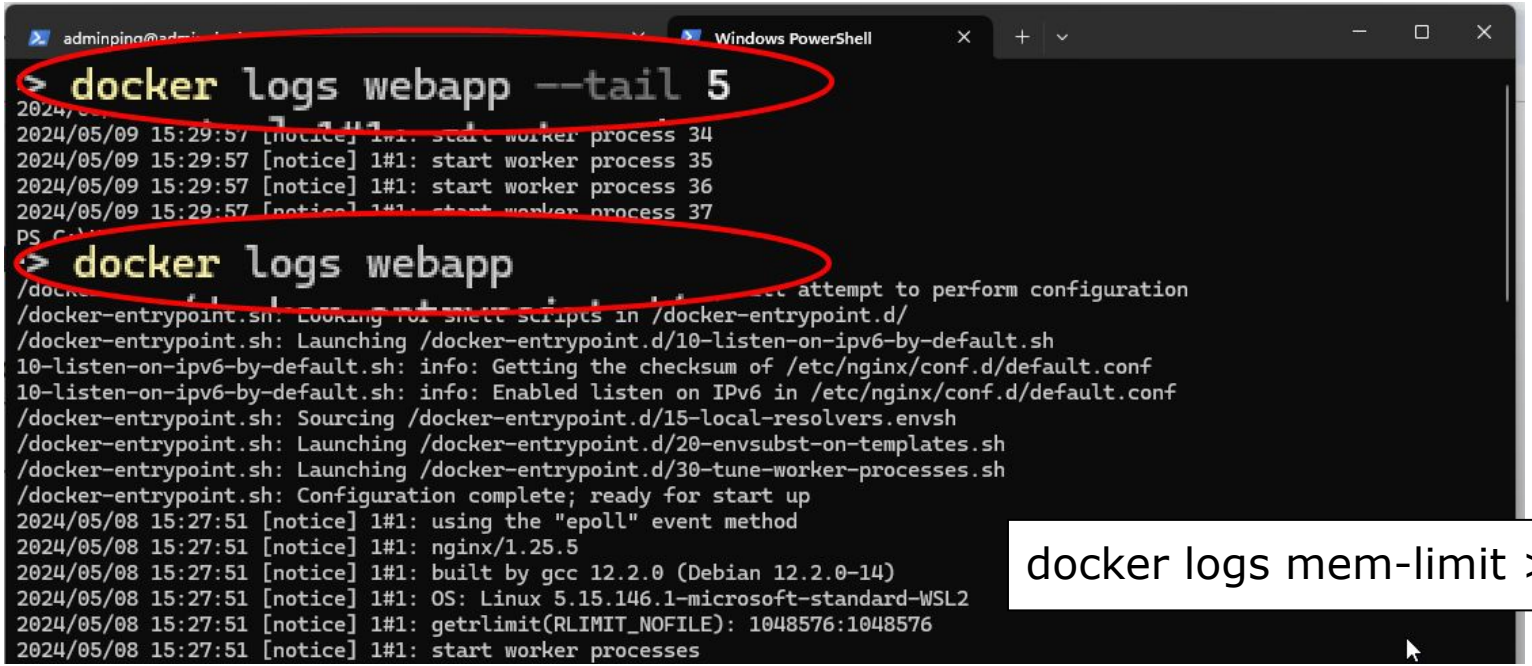
Why Pause
- **Resource Manamgement**
- **Trobuleshooting - Inspect Log**
- **Test & Development**

- unpause

```
docker unpause <container_name / container_id>
```

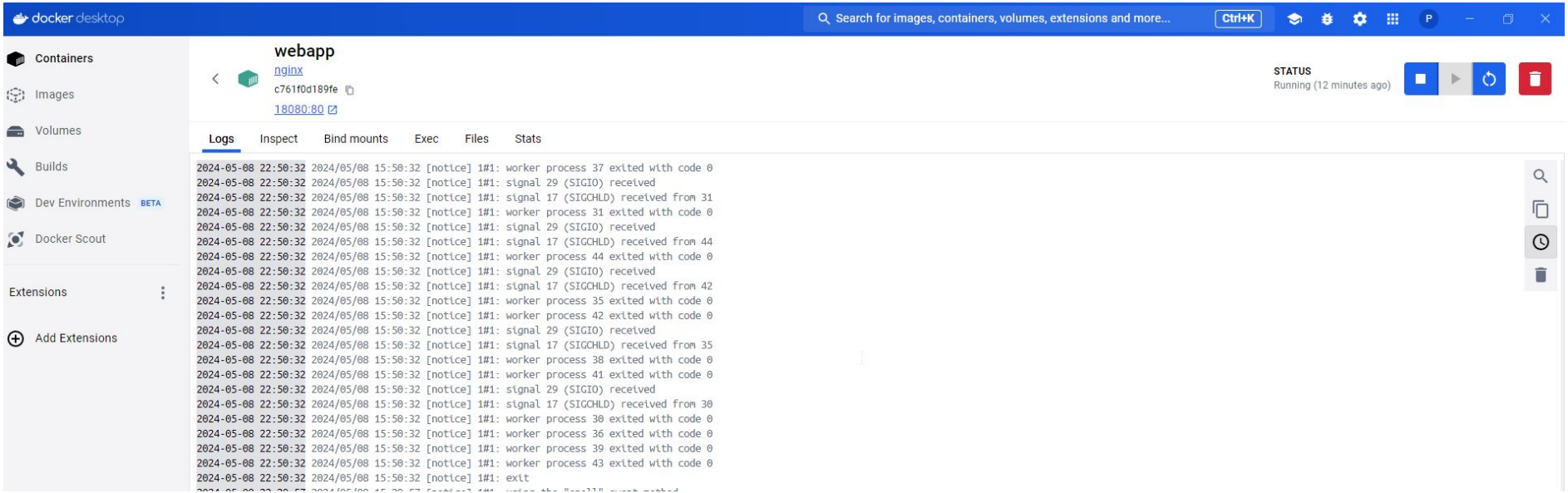
docker logs - Investigate Case a Crimescene

docker logs <container_name / container_id>



docker logs mem-limit > log.txt

docker logs - Docker Deskop



Basic Docker Command (Clean Up)

- Delete Containers

```
docker rm <container_name or container_id>
docker rm -f <container_name or container_id>      //Force Delete
docker rm $(docker ps -a -q)                       //Delete All
docker rm $( docker ps -q -f status=exited)        //Delete All Stop (Status Exited)
```

- Delete Images

  Note: <repo_url> ใส่ในกรณีที่มาจากแหล่งอื่นๆ ที่ไม่ใช่ docker hub เช่น gchr.io (github) หรือ repo กลางขององค์กร

```
docker rmi <repo_url>/<image_name>:<tag>
docker rmi -f <repo_url>/<image_name>:<tag>    //Force Delete
docker rmi -f $(docker images -a -q)           //Delete All
```

- Clear Everything > [docker system prune | Docker Docs]

15-20 minute

Try

docker exec -it
docker cp > CopySample at main · pingkunga(github.com)
docker start / stop / kill / pause / unpause
docker logs

Hand On - PostgresSQL

- Pull PostgresSQL
- Check Image
- Run PostgresSQL Container
- List Container Image
- Access VIA CLI
- Create Sample DB
- Inspect Container info
- Delete PostgresSQL Container
- Delete PostgresSQL Image

Hand On - PostgresSQL (Soln)

- Pull PostgresSQL

```
docker pull postgres:16.3-alpine3.18
```

- Check Image

```
docker images
```

- Run PostgresSQL Container

```
docker run --name invs_postgresdb -e
POSTGRES_USER=postgres -e
POSTGRES_PASSWORD=postgres -p 5432:5432 -d
postgres:16.3-alpine3.18
```

- List Container Image

```
docker ps -a
```

Hand On - PostgresSQL (Soln- Con.)

- Access VIA CLI

```
docker exec -it invs_postgresdb psql -U postgres
```

- Create Sample DB

```
CREATE DATABASE invs_db;
```

คำสั่งพื้นฐานกับ psql

\q ออกจาก psql
\! cls , \! clear เคลียร์สกรีน
\l ดูรายการฐานข้อมูล
\du ดู user
\c ดูว่าปัจจุบันเลือกฐานข้อมูลอะไรและใช้ user ไหนอยู่
\d ดูรายการข้อมูลทั้งหมดใน schema
\dt, \dt+ ดูเฉพาะตาราง
\d <table_name> ดูโครงสร้างตาราง
\dx ดูว่ามี Extensions อะไร

```
postgres=# CREATE DATABASE invs_db ENCODING unicode;
CREATE DATABASE

postgres-# \l
                                                    List of databases
   Name    |  Owner   | Encoding | Locale Provider |  Collate   |   Ctype    | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+------------+------------+------------+-----------+-----------------------
 invs_db   | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           |
 postgres  | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           |
 template0 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en_US.utf8 | en_US.utf8 |            |           | =c/postgres          +
           |          |          |                 |            |            |            |           | postgres=CTc/postgres
(4 rows)
```

Hand On - PostgresSQL (Soln- Con.)

- Inspect Container info

```
docker inspect invs_postgresdb
```

- Delete PostgresSQL Container
- Delete PostgresSQL Image

```
docker stop invs_postgresdb
docker rm invs_postgresdb
docker rmi postgres:16.3-alpine3.18
```

Container Resource Monitoring & Limit Resource

Ref: Runtime options with Memory, CPUs, and GPUs | Docker Docs

Resource Monitoring

Why
- Prevent OOME
- Resource Planing / Allocation

- get each container resouce

```
docker stats
```

- get per container

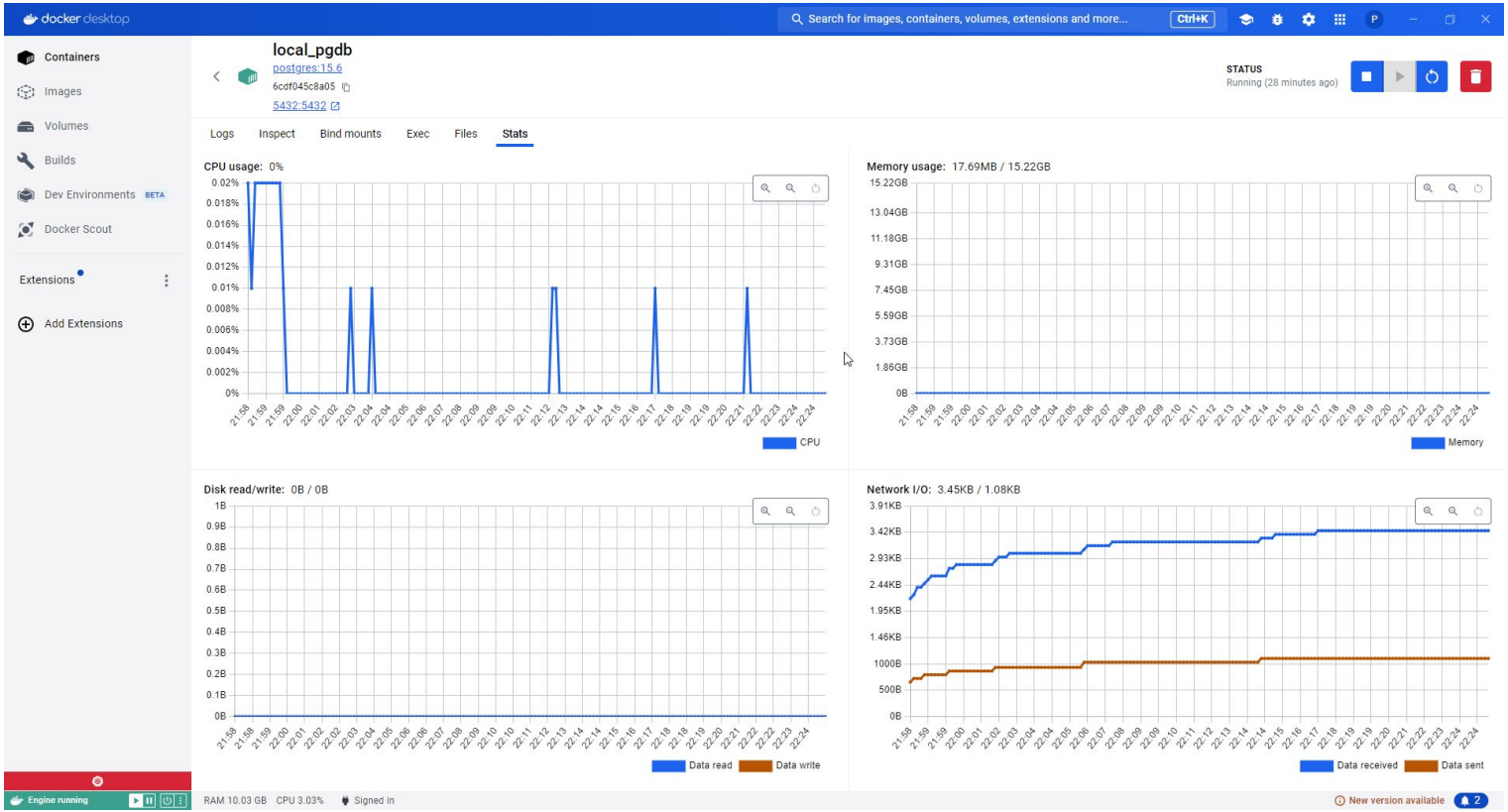
```
docker stats <<container_name/id>>
```

- filter

```
docker stats <<container_name/id>> --no-stream --format "{{ json . }}" | jq .
```

```
#For Windows Install jq first
choco install jq
```

Resource Monitoring - Docker Desktop

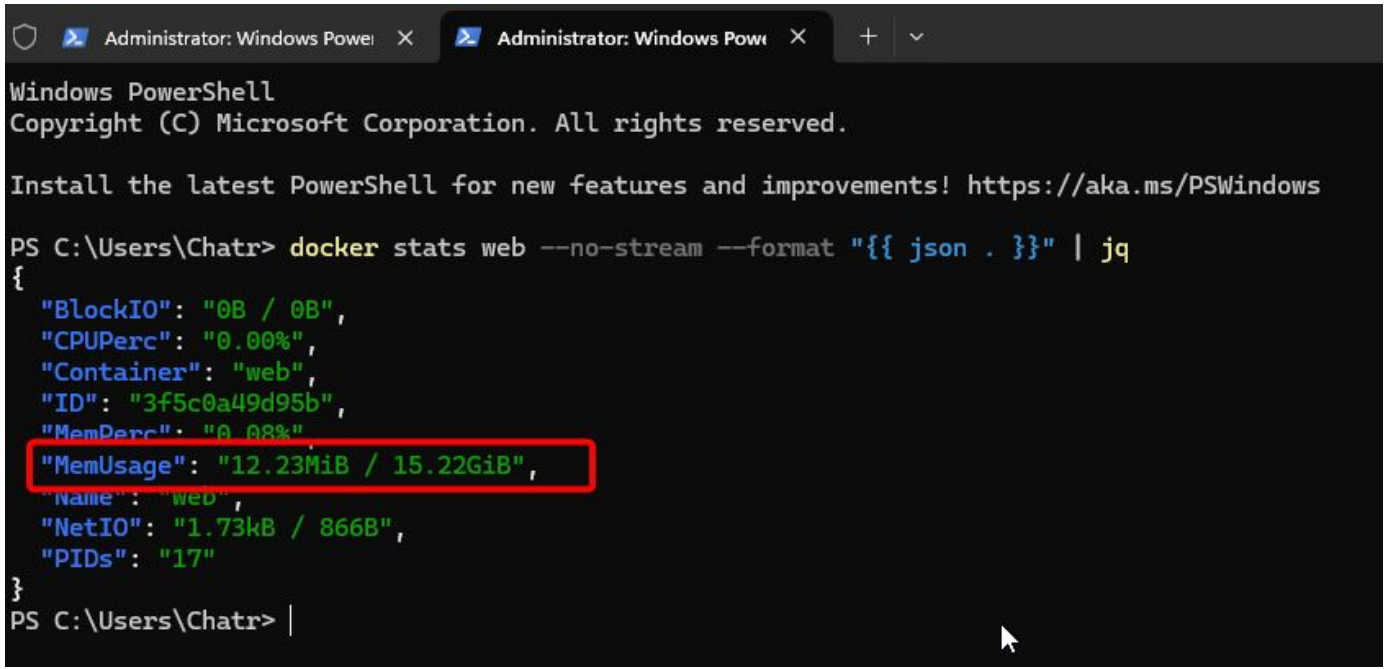


Resource Monitoring - Other Tools

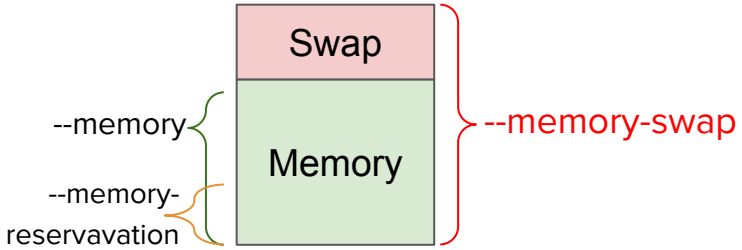
- Command
  - docker stats
  - Glances
- Web Base
  - uptime-kuma
  - Portainer
  - Zabbix - (Ref: Docker monitoring and integration with Zabbix)
  - Checkmk Raw
  - LGTM (More than monitoring > Observability)

Resource Monitoring - Limit Resource

Why - Default is Max of Host Resource



Resource Monitoring - Limit Resource (Memory)



- Memory (Hard / Soft / Swap)
  - --memory = Maximum Memory (Hard Limit)

```
docker run -d --name mem-limit --memory=256m nginx:alpine
```

  - --memory-reservavation = Minimum Memory (Soft Limit)

```
docker run -d --name mem-limit --memory=256m --memory-reservation=128m nginx:alpine
```

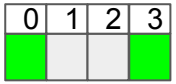
  - --memory-swap = Excess Memory (Critical Case use disk like linux swap)

```
docker run -d --name mem-limit --memory=256m --memory-reservation=128m
--memory-swap=384m nginx:alpine
```

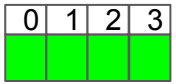
Resource Monitoring - Limit Resource (CPU)

- CPU (Req. No of vCPU)
  - --cpuset-cpus = Fixed CPU Core

```
docker run -d --name cpu-limit --cpuset-cpus 0,3 nginx:alpine
```

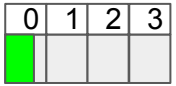



```
docker run -d --name cpu-limit --cpuset-cpus 0-3 nginx:alpine
```

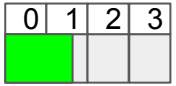


  - --cpus = Fixed by Resource Size (Percent)

```
docker run -d --name cpu-limit --cpus=0.8 nginx:alpine
```

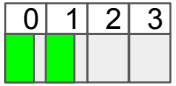



```
docker run -d --name cpu-limit --cpus=1.5 nginx:alpine
```



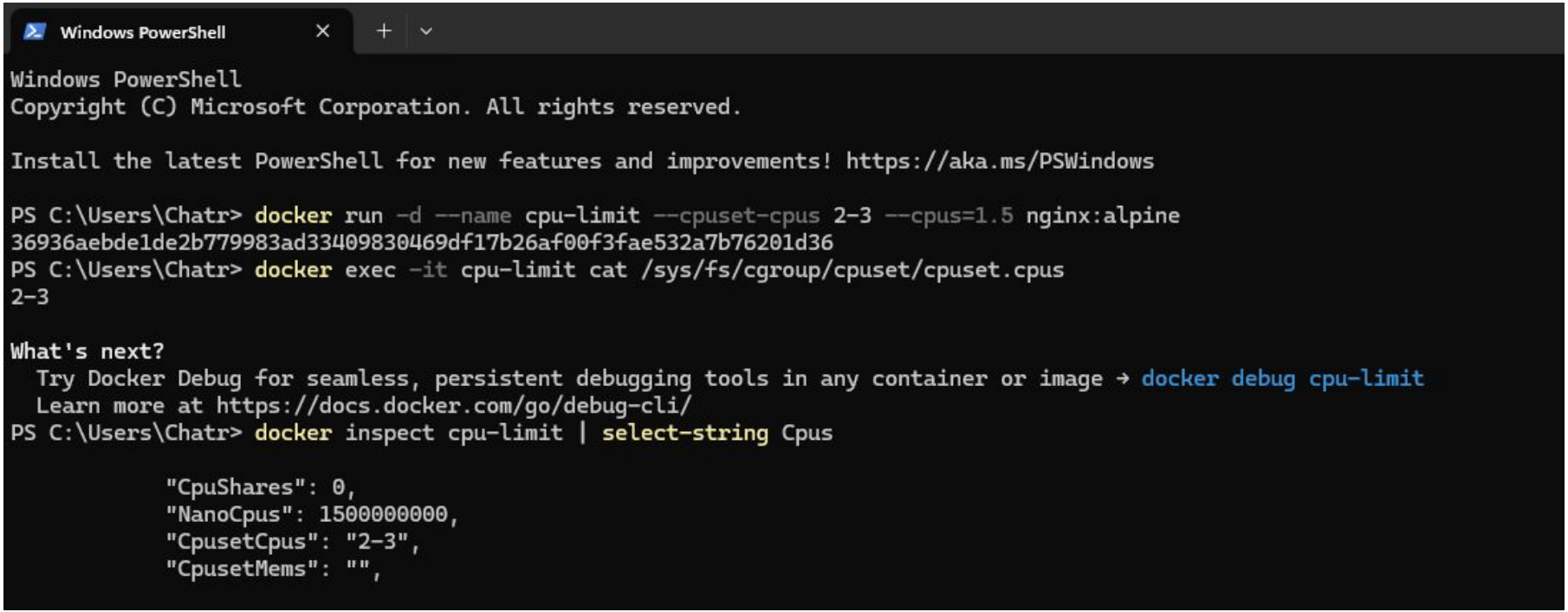

```
docker run -d --name cpu-limit --cpus=2.0 --cpuset-cpus=0,1 nginx:alpine
```


```
docker run -d --name cpu-limit --cpus=1.5 --cpuset-cpus=0,1 nginx:alpine
```



แบ่ง Core ไหนเท่าไหร่ Random

Resource Monitoring - Limit Resource (CPU)



Resource Monitoring - Hand-On

- Day1ResourceLimitSample.md at main · pingkunga (github.com)

Container Networking

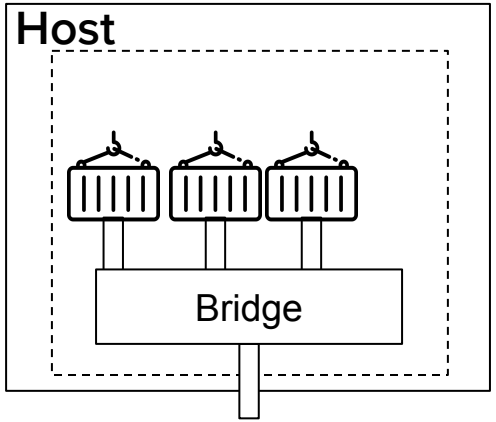
Networking

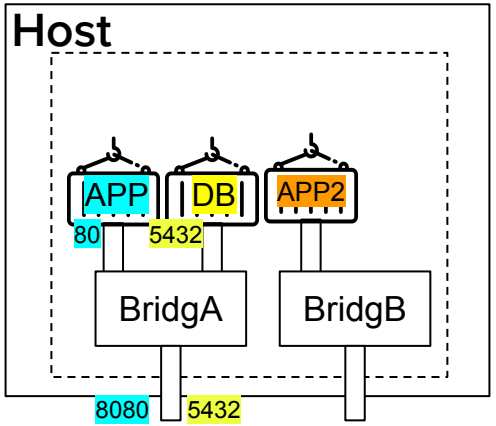
- Allow containers to communicate with each other / host / system / external etc.
- Access by Container Name, Docker Resolve IP automatically.
- Type
  - bridge (default) - private network
  - host - use host netwok / port (Linux Only)
  - none - no network
  - overlay - share network between docker hosts (Recommend K8S)
  - ipvlan
  - macvlan - assign mac address

Ref: Day in the Life: Docker Networking - YouTube

Networking

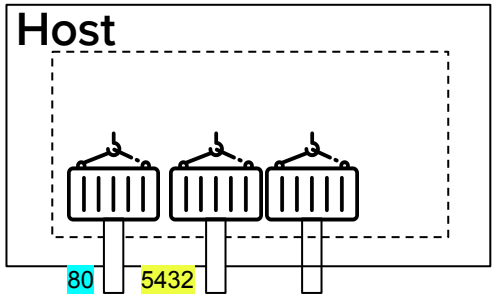
- Type
  - bridge (default)
  - host
  - none



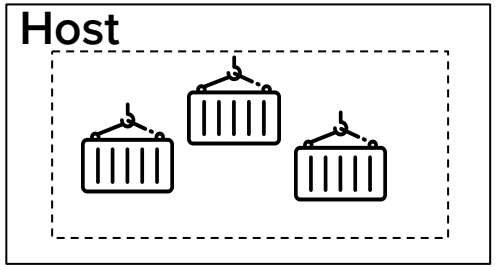


Networking

- Type
  - bridge (default)
  - host
  - none



- Type
  - bridge (default)
  - host
  - none



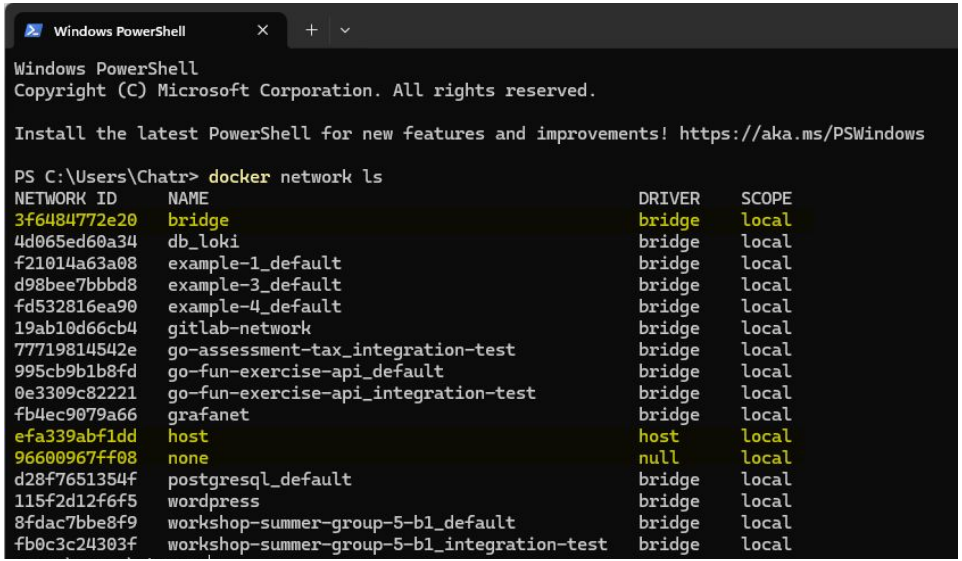
Networking

- Create Network

```
docker network create <<network_name>>
docker network create -d <<driver name>> <<network_name>>
```

- List Network

```
docker network ls
```



Networking

- Add Container to Network

```
docker network connect <<network_name>> <<container_name>>
```

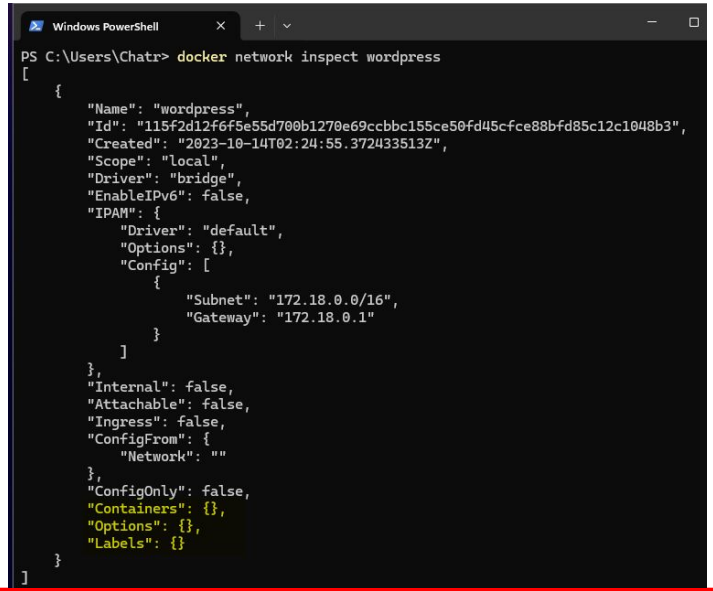
- Create Container with Network

```
docker run -d --name webapp --network <<network_name>> nginx:alpine
```

Networking

- Inspect Network

```
docker network inspect <<network_name>>
```



Networking

- Disconnect from network

```
docker network disconnect <<network_name>> <<container_name>>
```

- Remove Network

```
docker network rm <<network_name>>
```

Networking - Sample

- Day1/NetworkSample/NetworkSample.md at main · pingkunga (github.com)

Container - Store Data

Store Data

Container Stateless

- Way to Store Data
  - volume - docker manage
  - mount
- Basic Command docker manage volume

```
# Create a named volume
docker volume create mysql-files

# List all volumes
docker volume ls

# Connect a volume to a container
docker run --name mysql-db -d -e MYSQL_ROOT_PASSWORD=secret -v
mysql-files:/var/lib/mysql --label dev.app=persistent-db mysql

# Remove a volume (only works if it’s not in use)
docker volume rm mysql-files
```

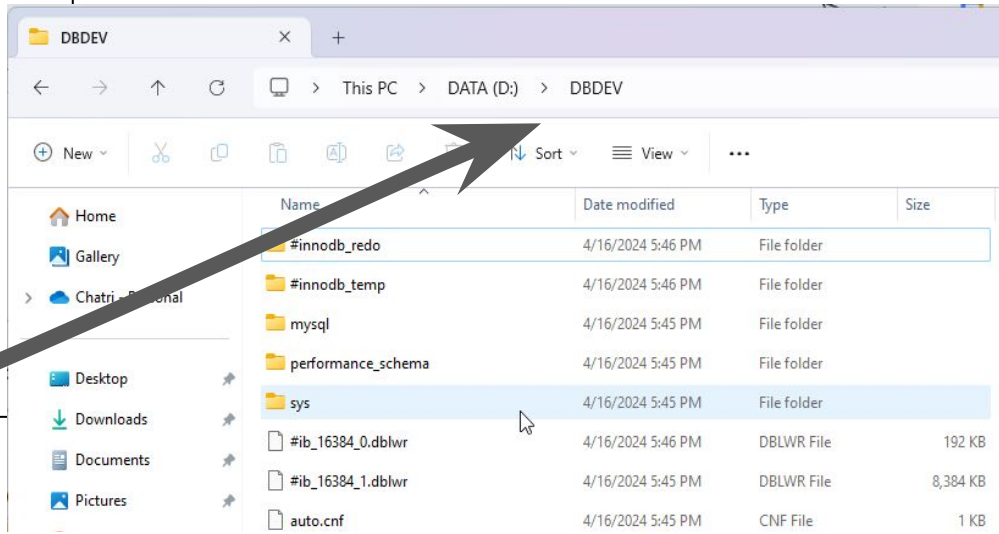
How to back up and restore your Docker Desktop data | Docker Docs

Back Up and Share Docker Volumes with This Extension | Docker

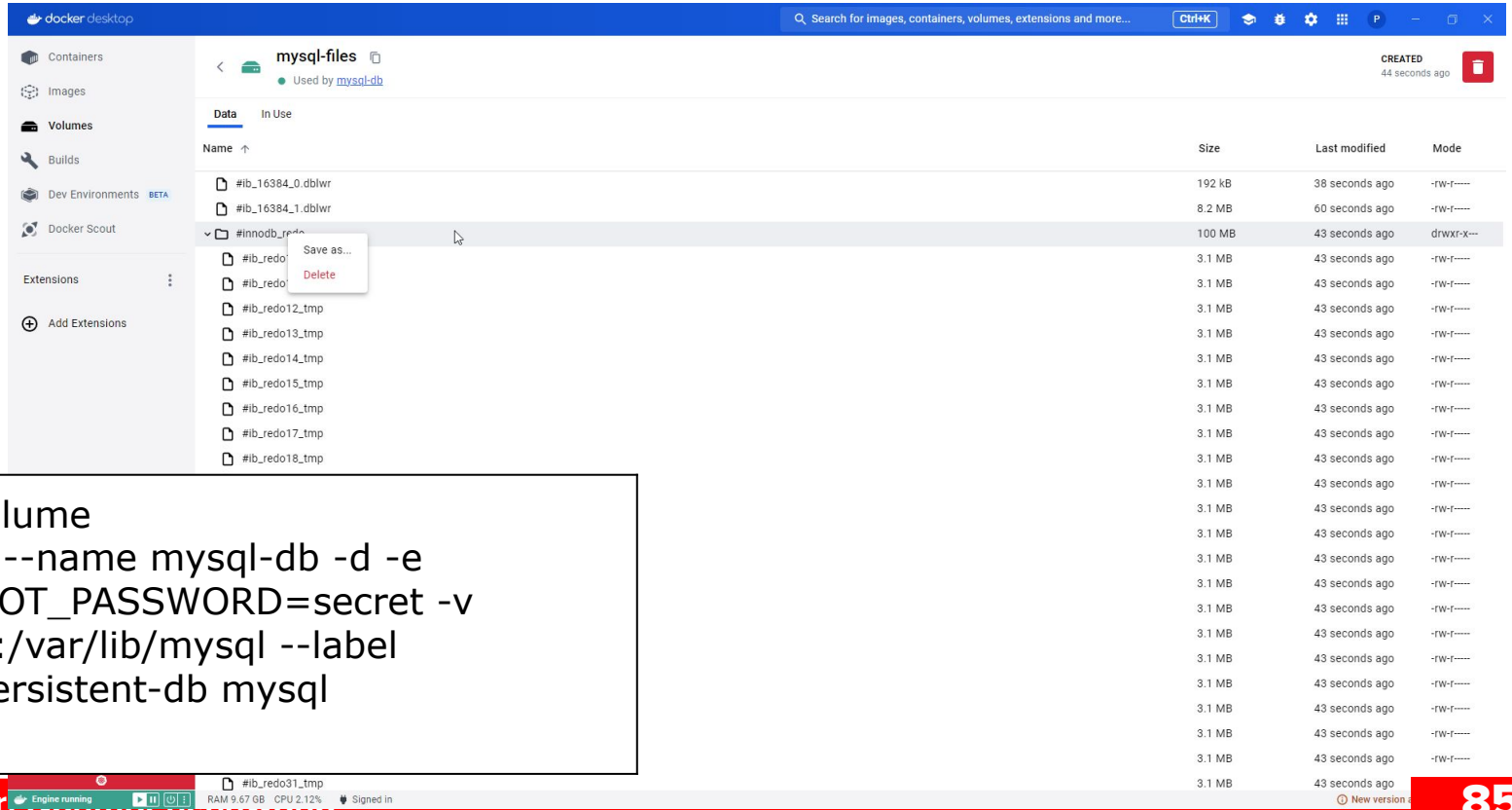
Store Data - docker volume vs mount

- Mount - Map HostPath / Network Path (NFS) with container path

```
#docker volume
docker run --name mysql-db -d -e
MYSQL_ROOT_PASSWORD=secret -v
mysql-files:/var/lib/mysql --label
dev.app=persistent-db mysql

# mount
docker run --name mysql-db-mount -d
-e MYSQL_ROOT_PASSWORD=secret
-v D:\DBDEV:/var/lib/mysql
--label training.app=persistent-db mysql
```



Store Data - docker volume in Docker Desktop

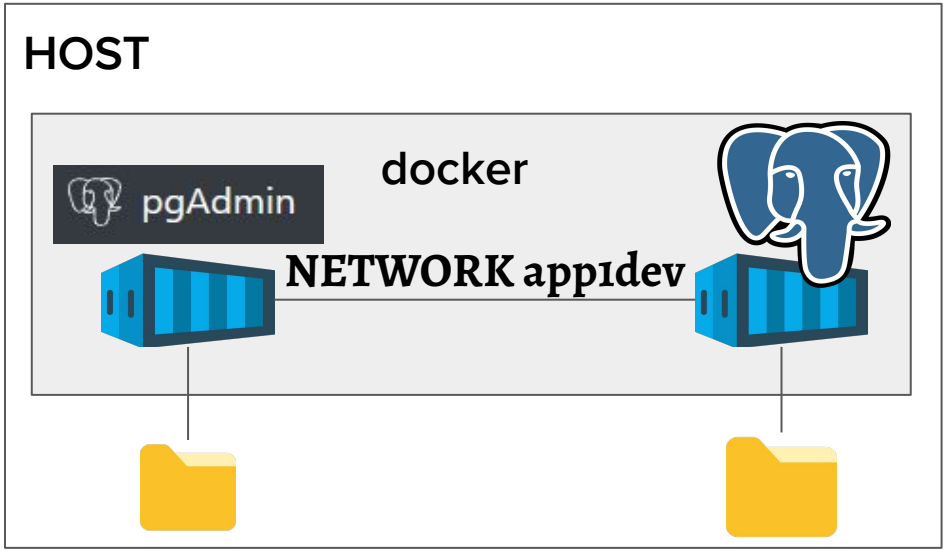



```
#docker volume
docker run --name mysql-db -d -e
MYSQL_ROOT_PASSWORD=secret -v
mysql-files:/var/lib/mysql --label
dev.app=persistent-db mysql
```

Hand On - Volume

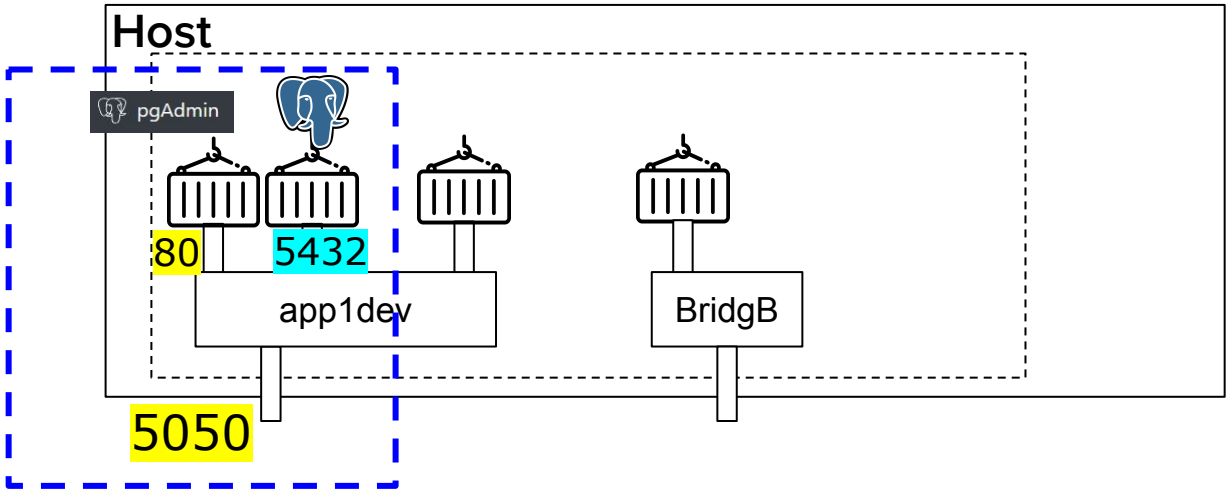
- Day1/VolumeSample/VolumeSample.md at main · pingkunga/ (github.com)

Hand On - Multiple Image



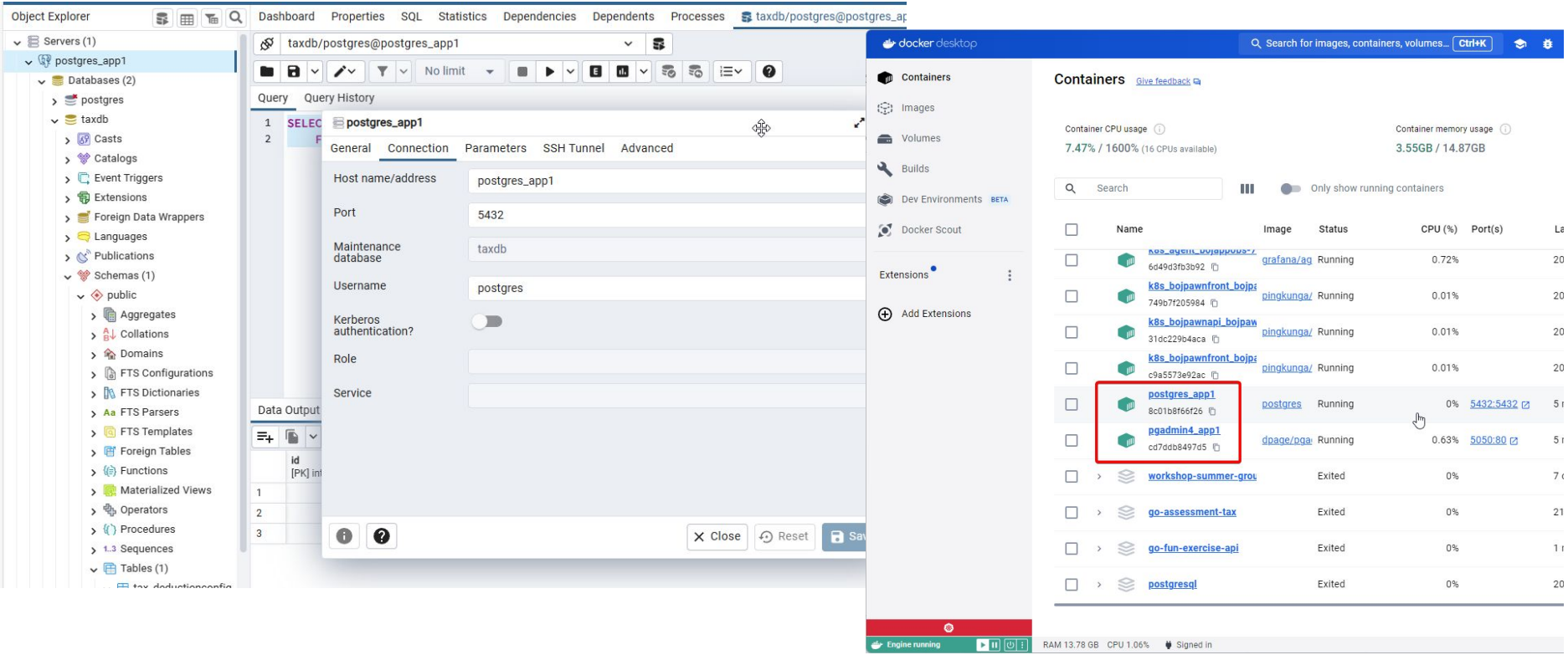
- Pull PostgresSQL
- Check Image
- Run PostgresSQL Container with
  - network app1dev+ Volume/Mount
  - Init Database with init.sql
- List Container Image
- Pull PGAdmin4
- Run PostgresSQL Container with network app1dev + Volume/Mount
- Create Connection
- Delete PostgresSQL/PGAdmin4 Container

Hand On - Multiple Image Solution

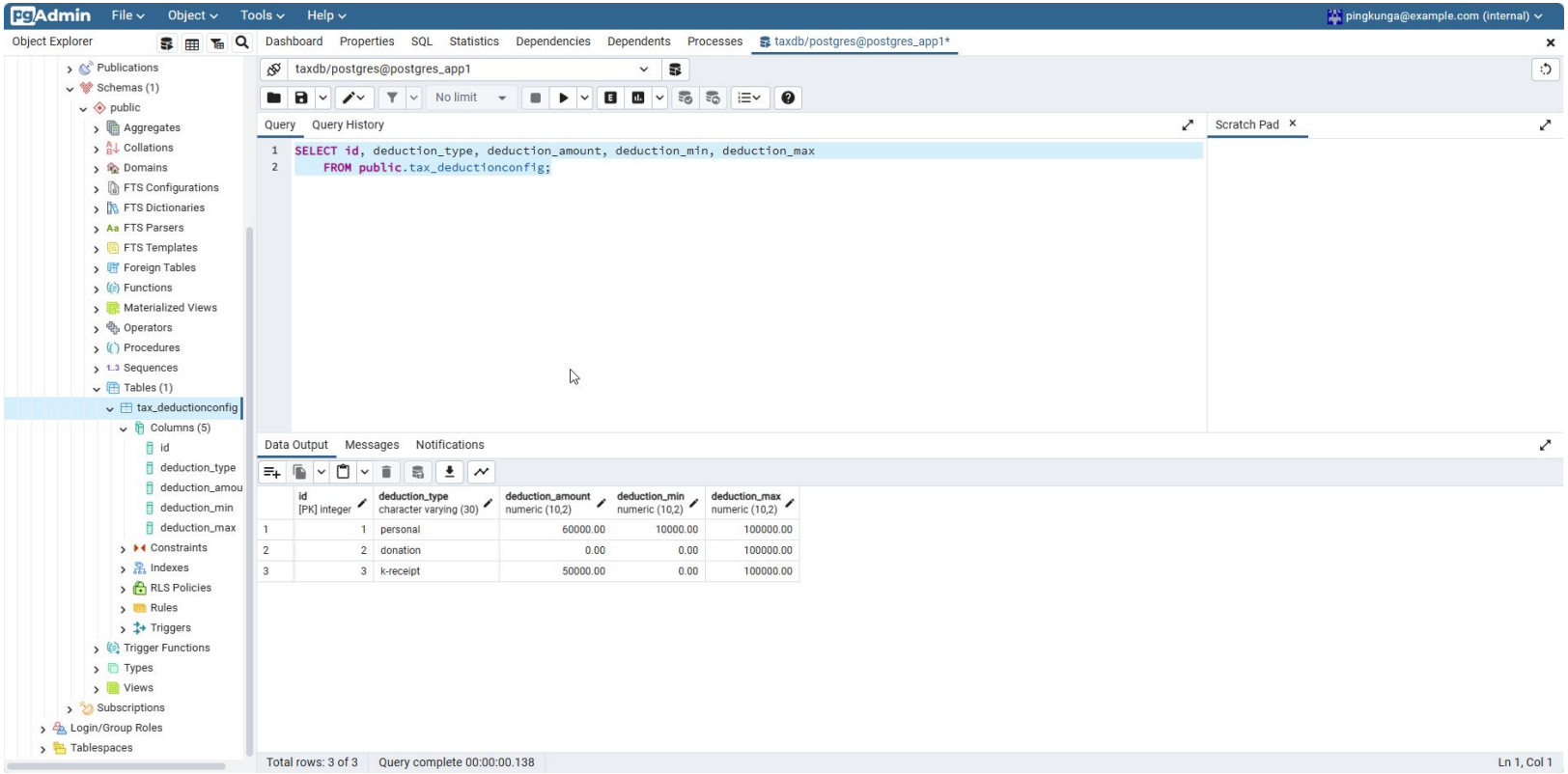


Solution: [Day1/MultipleImageWithVolumeNetwork/solution.md at main · pingkunga(github.com)]

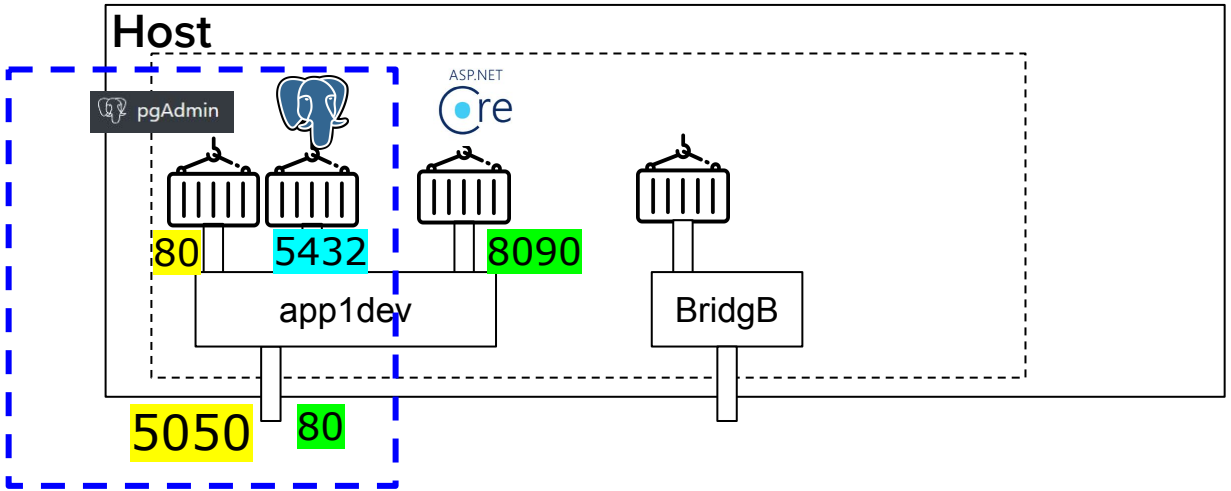
Hand On - Multiple Image Solution



Hand On - Multiple Image Solution



Hand On - Multiple Image Solution (Improve)



Day 2

More Example with Resouce Limit

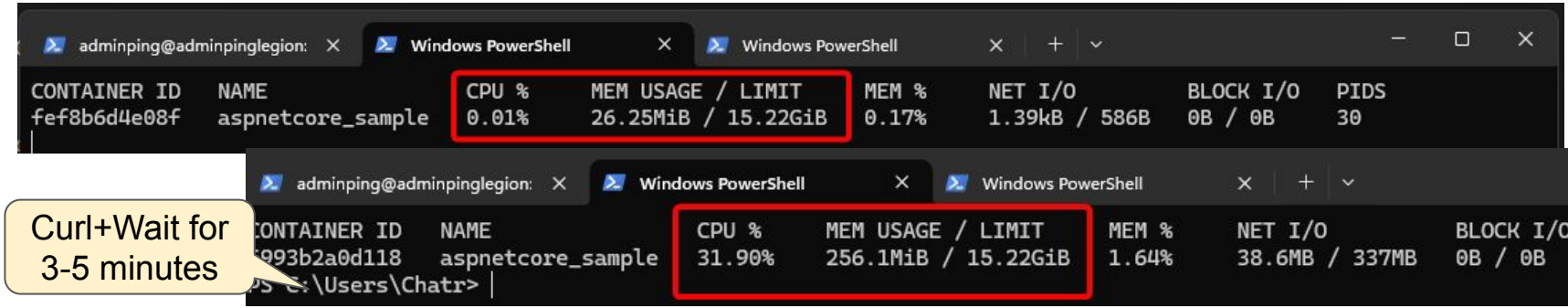
- ASP NET8 microsoft-dotnet-samples - Official Image
- Sample curl with powershell
- Hand-On: Day1/ResourceLimitWithASPNETCore.md at main · pingkunga/(github.com)

More Example with Resouce Limit

- ASP NET8 [microsoft-dotnet-samples - Official Image]
- Sample curl with powershell

Initial:

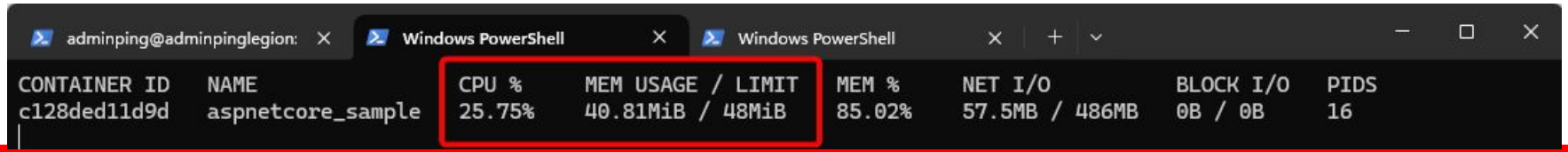
```
docker run -it --rm -p 8000:8080 --name aspnetcore_sample
mcr.microsoft.com/dotnet/samples:aspnetapp
```



Curl+Wait for 3-5 minutes

Limit Resource:

```
docker run -it --rm -p 8000:8080 --memory=48m --memory-reservation=32m
--memory-swap=64m --cpus=0.8 --name aspnetcore_sample
mcr.microsoft.com/dotnet/samples:aspnetapp
```



How can I add a volume / host directory (disk / NFS) to an existing Docker container?

Update configuration of running Containers

```
docker update <options> <container id/name>
```

- Example

```
docker update -m 300M aspnetcore_sample aspnetcore_api
```

Ref: docker container update | Docker Docs / docker update command sample (github.com)

Options

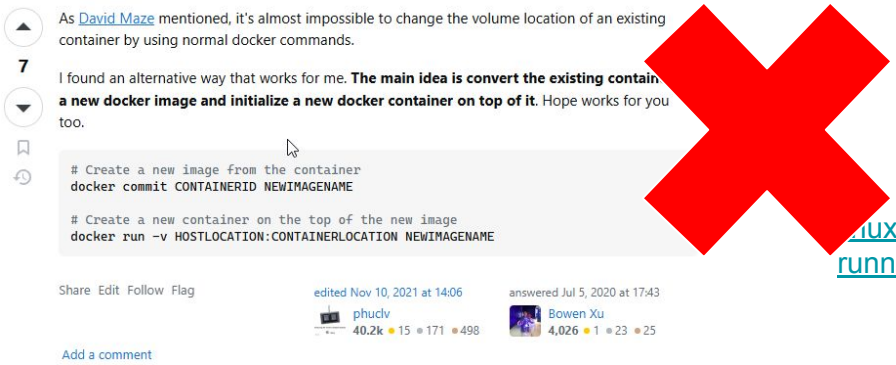
Option	Default	Description
`--blkio-weight`		Block IO (relative weight), between 10 and 1000, or 0 to disable (default 0)
`--cpu-period`		Limit CPU CFS (Completely Fair Scheduler) period
`--cpu-quota`		Limit CPU CFS (Completely Fair Scheduler) quota
`--cpu-rt-period`		API 1.25+ Limit the CPU real-time period in microseconds
`--cpu-rt-runtime`		API 1.25+ Limit the CPU real-time runtime in microseconds
`-c, --cpu-shares`		CPU shares (relative weight)
`--cpus`		API 1.29+ Number of CPUs
`--cpuset-cpus`		CPUs in which to allow execution (0-3, 0,1)
`--cpuset-mems`		MEMs in which to allow execution (0-3, 0,1)
`-m, --memory`		Memory limit
`--memory-reservation`		Memory soft limit
`--memory-swap`		Swap limit equal to memory plus swap: -1 to enable unlimited swap
`--pids-limit`		API 1.40+ Tune container pids limit (set -1 for unlimited)
`--restart`		Restart policy to apply when a container exits

How can I add a volume / host directory (disk / NFS) to an existing Docker container? (Work Around)

Solution 1 : docker cp

- docker cp [OPTIONS] OLD_CONTAINER:SRC_PATH DEST_PATH (disk / NFS)
- create new container + mount
- remove old container

How can I add a volume / host directory (disk / NFS) to an existing Docker container? (Work Around)



As David Maze mentioned, it's almost impossible to change the volume location of an existing container by using normal docker commands.

I found an alternative way that works for me. **The main idea is convert the existing contain a new docker image and initialize a new docker container on top of it**. Hope works for you too.

```
# Create a new image from the container
docker commit CONTAINERID NEWIMAGENAME

# Create a new container on the top of the new image
docker run -v HOSTLOCATION:CONTAINERLOCATION NEWIMAGENAME
```

Share Edit Follow Flag

edited Nov 10, 2021 at 14:06 — phuclv — 40.2k ● 15 ● 171 ● 498

answered Jul 5, 2020 at 17:43 — Bowen Xu — 4,026 ● 1 ● 23 ● 25

Add a comment

[…ux - How to mount a host directory into a running docker container - Stack Overflow]

Solution 2 : create a new image from old container

- docker commit <ContainerId> <NewImageName>
- create new container from NewImageName + mount
- remove/stop NewImageName Container
- start new container from old image with mount option

!!! Data Loss ถ้า Images นั้นถูกกำหนดว่า ถ้าถูก Deploy ให้ Initial ข้อมูลใหม่

Build Your Own Image

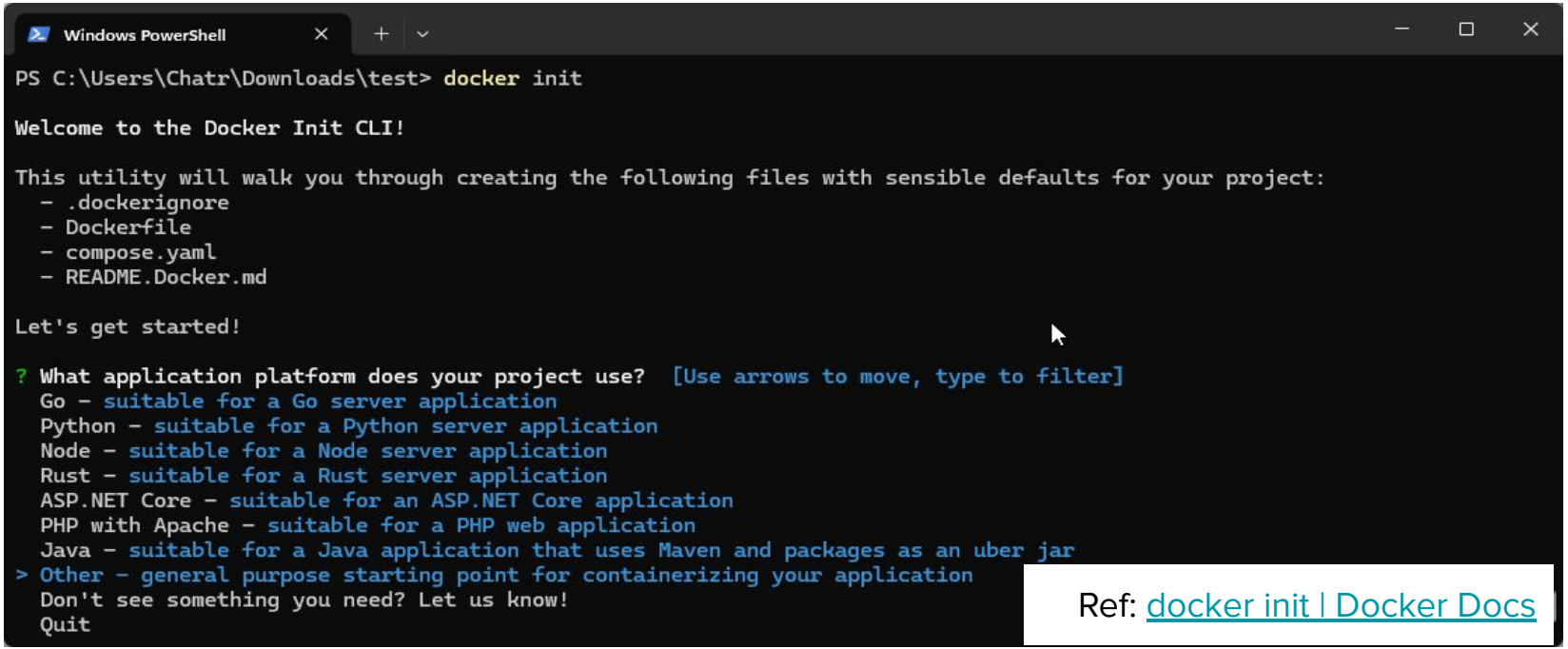
- Building Container (Dockerfile)
- Hand-On

Why Custom image

- Customer Requirement !!!
- Optimize Cost
- Security Concert (non-root)

docker init

- Initialize a project with the files necessary to run the project in a container.



Ref: [docker init | Docker Docs]

docker init - output

1. .dockerignore - specifies which files and directories should be excluded from the build such as source code / secret
2. Dockerfile - a script that contains instructions for building a customized docker image
3. compose.yaml - make container working together
4. README.Docker.mdv

Dockerfile - Way to defined Container

- Structure
  - FROM : select the base image
  - WORKDIR : defines a working directory for RUN CMD ENTRYPOINT COPY ADD
  - ADD/**COPY** : move file from host > container image
  - RUN : execute command (linux)
  - ENV : set environment variable
  - USER : defines the user to use to run the RUN CMD ENTRYPOINT command

FROM scratch

Security Issue

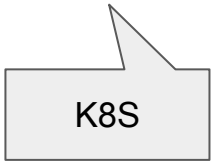
Ref: [Dockerfile reference | Docker Docs]

Dockerfile - Way to defined Container

- Structure (Cont.)
  - EXPOSE : specifies the container to wait for requests on the specified port. Use it with -p when using docker run.

  - CMD : execute command, but only once in file, if more than one use last one or use as default parameter to ENTRYPOINT

  - ENTRYPOINT : Command to run after starting the container



Ref: Dockerfile reference | Docker Docs

Sample Docker File

- donet6_db2_dockerfile (github.com)

donet6_db2_dockerfile

```
1  FROM mcr.microsoft.com/dotnet/aspnet:6.0 AS runtime
2
3  COPY bin/Release/net6.0/publish/ /app
4
5  RUN apt-get update && apt-get install -y libxml2-dev
6  ENV DB2_CLI_DRIVER_INSTALL_PATH="/app/clidriver"
7  ENV LD_LIBRARY_PATH="/app/clidriver/lib:/app/clidriver/lib/libdb2.so"
8  ENV LIBPATH="/app/clidriver/lib"
9
10 ENV PATH=$PATH:"/app/clidriver/bin"
11 ENV PATH=$PATH:"/app/clidriver/adm"
12 ENV PATH=$PATH:"/app/clidriver/lib"
13
14 RUN apk add --no-cache tzdata
15 ENV TZ=Asia/Bangkok
16 RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
17
18 RUN addgroup --system --gid 10001 invsopgp && adduser --system --uid 1001 --ingroup invsopgp --shell /bin/sh invsbch
19 RUN chown -Rf invsbch:invsopgp /app && chmod -Rf 760 /app
20
21 USER invsbch
22 WORKDIR /app
23 RUN mkdir /app/BNZStorage
24
25 ENV ASPNETCORE_URLS=http://+:8080
26 EXPOSE 8080
27
28 ENTRYPOINT ["dotnet", "ds.Invest.WebAPI.dll" , "-buc", "/app" , "-env", "Production"]
```

Build Image from Dockerfile

~~docker build~~ not support multi-platform

Context - Current Dir

```
docker build -t inv-api:1.0.0 .

docker build -t inv-api:1.0.0 -f abc.DockerFile

docker build -t inv-api:1.0.0 --no-cache -f abc.DockerFile
```

Build Image from Dockerfile

docker buildx for multiple platform

- create builder

```
#Create Builder (Container)
docker buildx create --name mybuilder
docker buildx use mybuilder
```

- build

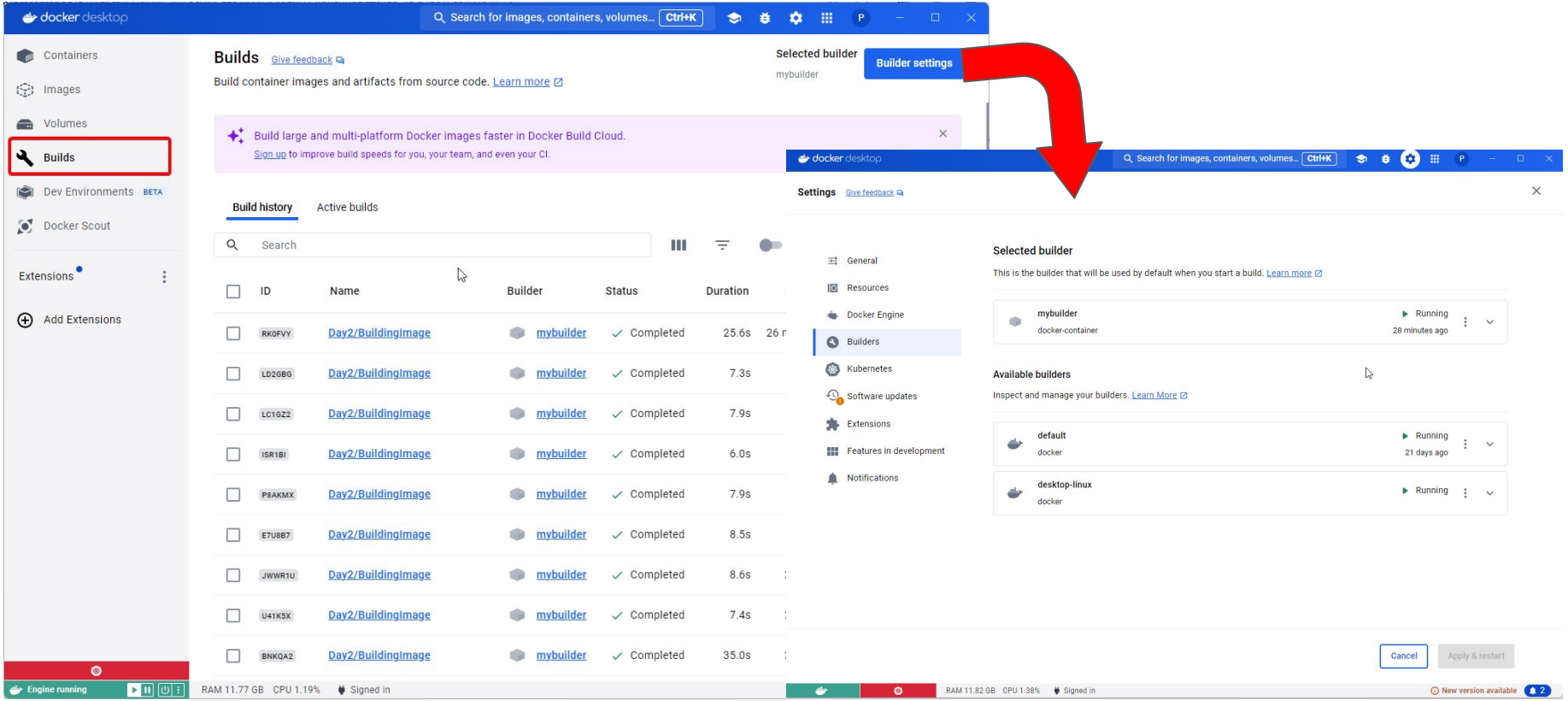
```
#build and save local
docker buildx build --platform linux/amd64,linux/arm64 -t ourapp:latest . --load
docker buildx build --platform linux/amd64 -t my_org_postgres:16.3 . -f .\DockkerfileNotOptimize
--load
```

```
#build and push
docker buildx build --platform linux/amd64,linux/arm64 -t ourapp:latest . --push
```

Build Image from Dockerfile

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE                                   powershell - BuildingImage

ERROR: unable to prepare context: path ".DockkerfileNotOptimize" not found
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage> docker buildx build --platform linux/amd64 -t my_org_postgres-l:16.3 DockkerfileNotOptimize --load
[+] Building 0.0s (0/0)                                                                                                  docker-container:mybuilder
ERROR: unable to prepare context: path "DockkerfileNotOptimize" not found
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage>  docker buildx build --platform linux/amd64 -t my_org_postgres-l:16.3 . -f .\DockkerfileNotOptimize --load
[+] Building 25.9s (13/13) FINISHED                                                                                      docker-container:mybuilder
 => [internal] load build definition from DockkerfileNotOptimize                                                                          0.0s
 => => transferring dockerfile: 763B                                                                                                      0.0s
 => [internal] load metadata for docker.io/library/postgres:16.3-bookworm                                                                 2.9s
 => [auth] library/postgres:pull token for registry-1.docker.io                                                                           0.0s
 => [internal] load .dockerignore                                                                                                         0.0s
 => => transferring context: 2B                                                                                                           0.0s
 => CACHED [1/7] FROM docker.io/library/postgres:16.3-bookworm@sha256:1bf73ccae25238fa555100080042f0b2f9be08eb757e200fe6afc1fc413a1b3c    0.0s
 => => resolve docker.io/library/postgres:16.3-bookworm@sha256:1bf73ccae25238fa555100080042f0b2f9be08eb757e200fe6afc1fc413a1b3c            0.0s
 => [2/7] RUN apt-get update                                                                                                              3.6s
 => [3/7] RUN apt-get install -y curl                                                                                                     4.0s
 => [4/7] RUN apt-get -y install postgresql-16-pgvector                                                                                   3.6s
 => [5/7] RUN apt-get -y install postgresql-16-cron                                                                                       4.5s
 => [6/7] RUN echo "shared_preload_libraries='pg_cron'" >> /usr/share/postgresql/postgresql.conf.sample                                   0.1s
 => [7/7] RUN echo "cron.database_name='postgres'" >> /usr/share/postgresql/postgresql.conf.sample                                        0.1s
 => exporting to docker image format                                                                                                      6.7s
 => => exporting layers                                                                                                                   1.0s
 => => exporting manifest sha256:1ac4765b500f5710ca5db65820a925087454688ffbd032a7921b3085b9541fea                                          0.0s
 => => exporting config sha256:0e7877a249961d01aadb7bbc31de57f806816bdf136ec4dd0637e2126971fdcc                                            0.0s
 => => sending tarball                                                                                                                    5.6s
 => importing to docker                                                                                                                   0.7s
 => => loading layer a6a7d9a4e0c9 163.84kB / 15.68MB                                                                                      0.7s
 => => loading layer a9b0a56913dc 32.77kB / 1.96MB                                                                                        0.5s
 => => loading layer eab3188bfd74 32.77kB / 936.60kB                                                                                      0.3s
 => => loading layer 168cdc8e04f4 32.77kB / 715.25kB                                                                                      0.2s
 => => loading layer 9278ee4ec784 9.14kB / 9.14kB                                                                                         0.1s
```

Build Image from Dockerfile - Docker Desktop

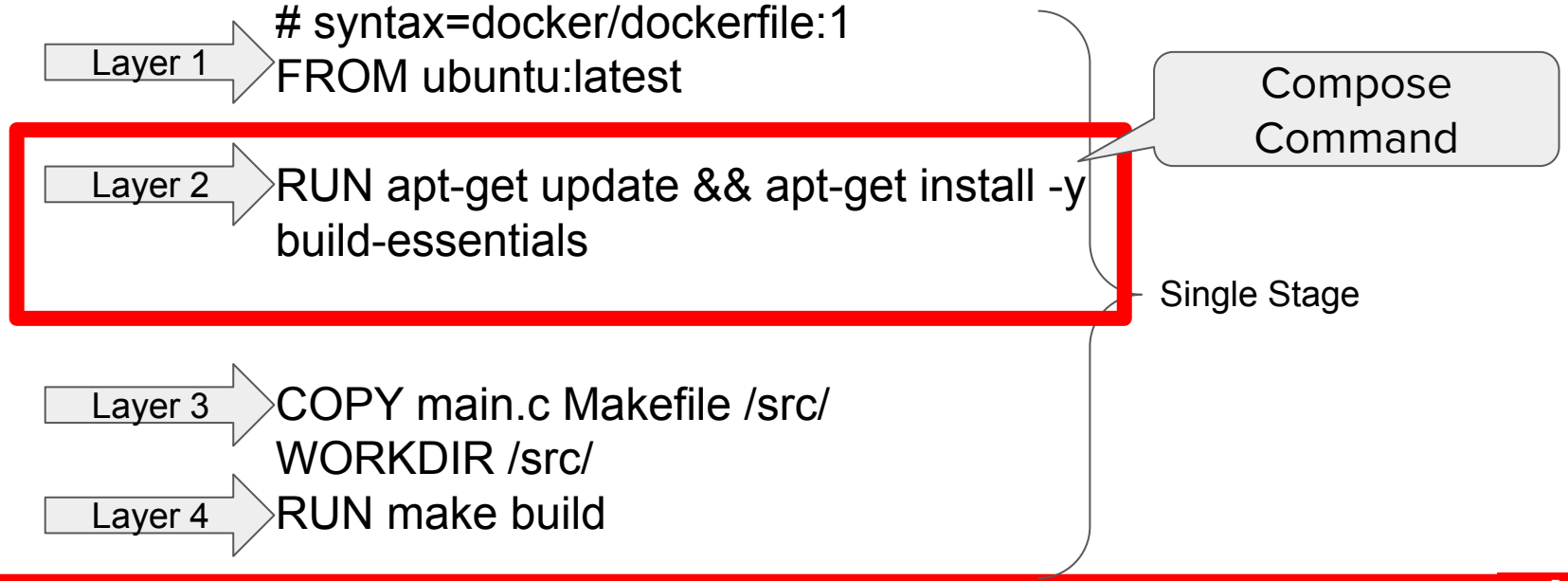


Docker Layer

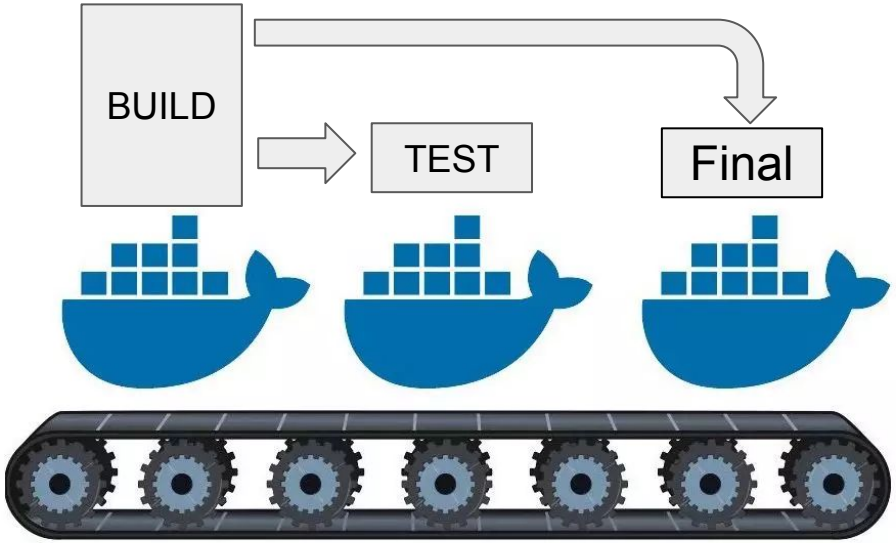
- COW (Copy On Write) Concept - Each Action on docker create layer (Like Code Commit)

Single Stage - Compose Command

- More Layer > Increase Image Size
  Hint: Compose Command into Single Step / or Using Multi-Stage



Multi-Stage



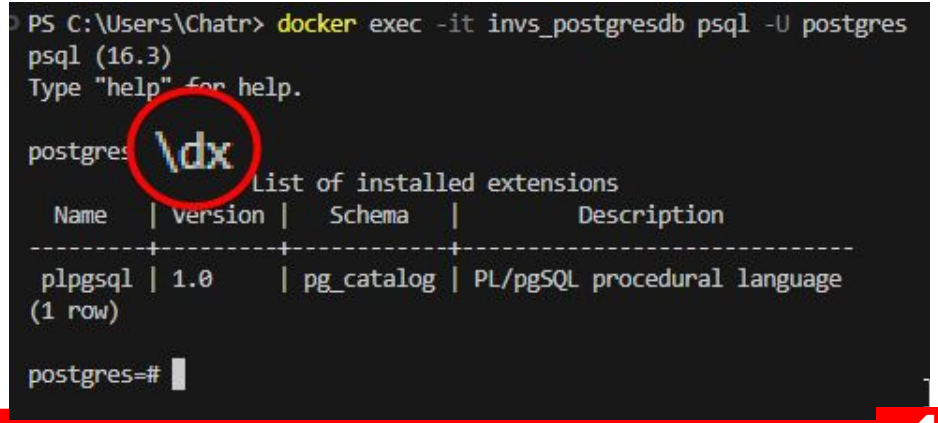
Sample: [go-assessment-tax/Dockerfile at main · pingkunga/go-assessment-tax (github.com)]

Blog: [Optimize Container image size | naiwaen@DebuggingSoft]

Create Dockerfile-Problem

Problem:

- Manager Ask me for create Postgresql with
  - AI Support (pgvector/pgvector: Open-source vector similarity search for Postgres (github.com))
  - Simple cron-based (citusdata/pg_cron: Run periodic jobs in PostgreSQL (github.com))
- But Postgresql Container Image does not fit with your requirement.



Create Dockerfile-Problem

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE

Error response from daemon: No such container: invs_postgresdb

What's next:
    Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug invs_postgresdb
    Learn more at https://docs.docker.com/go/debug-cli/
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage> docker exec -it invs_postgresdb_raw psql -U postgres
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

postgres=# \dx
                List of installed extensions
  Name   | Version |   Schema   |         Description
---------+---------+------------+------------------------------
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
(1 row)

postgres=# Create extension vector;
ERROR:  extension "vector" is not available
DETAIL:  Could not open extension control file "/usr/share/postgresql/16/extension/vector.control": No such file or directory.
HINT:  The extension must first be installed on the system where PostgreSQL is running.
postgres=# CREATE EXTENSION pg_cron;
ERROR:  extension "pg_cron" is not available
DETAIL:  Could not open extension control file "/usr/share/postgresql/16/extension/pg_cron.control": No such file or directory.
HINT:  The extension must first be installed on the system where PostgreSQL is running.
postgres=# exit
```

Create Dockerfile - Custom Docker

```
FROM postgres:16.3-bookworm

#https://packages.debian.org/trixie/amd64/database/postgresql-16-pgvector

#https://packages.debian.org/trixie/amd64/database/postgresql-16-cron

RUN apt-get update \
      && apt-get install -y curl \
      && apt-get -y install postgresql-16-pgvector \
      && apt-get -y install postgresql-16-cron
RUN echo "shared_preload_libraries='pg_cron'" >> /usr/share/postgresql/postgresql.conf.sample

RUN echo "cron.database_name='postgres'" >> /usr/share/postgresql/postgresql.conf.sample

#For your own database name

#RUN echo "shared_preload_libraries='pg_cron'" >> /usr/share/postgresql/postgresql.conf

#RUN echo "cron.database_name='your_db_name'" >> /usr/share/postgresql/postgresql.conf
```

Create Dockerfile - Custom Container Postgress Image

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  AZURE

be9b83e393ad169a382ad7512d0367f212019b5b905c75f3fdb66b21bc1f3a16
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage> docker exec -it invs_postgresdb /bin/sh
# psql -U postgres
psql (16.3 (Debian 16.3-1.pgdg120+1))
Type "help" for help.

postgres=# CREATE EXTENSION pg_cron;
CREATE EXTENSION
postgres=# CREATE EXTENSION vector;
CREATE EXTENSION
postgres=# \xn
postgres-# \dx
                              List of installed extensions
  Name   | Version |   Schema   |                     Description
---------+---------+------------+------------------------------------------------------------
 pg_cron | 1.6     | pg_catalog | Job scheduler for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
 vector  | 0.7.0   | public     | vector data type and ivfflat and hnsw access methods
(3 rows)

postgres-#
```

Top 8 PostgreSQL Extensions (timescale.com)

Create Dockerfile - Custom Container Postgress Image

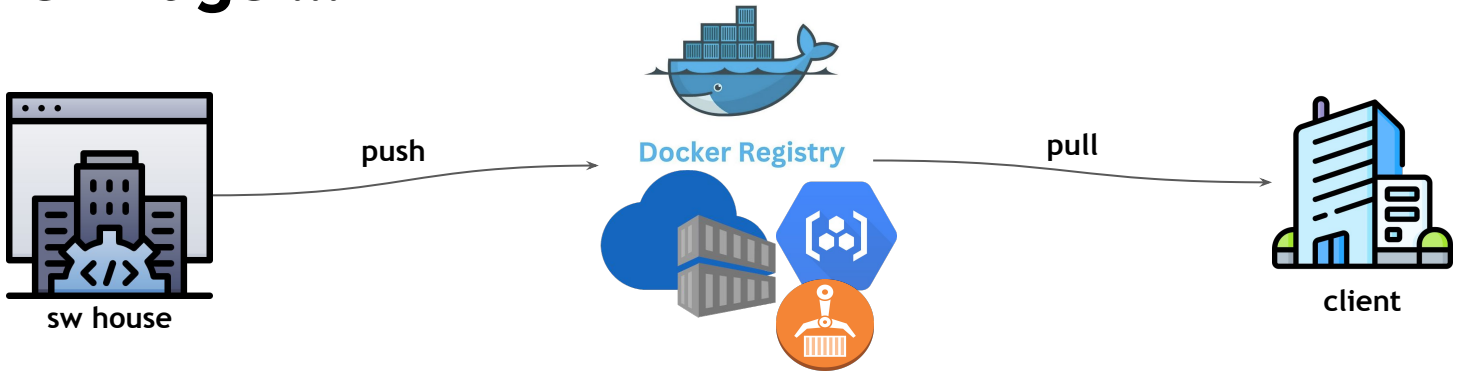
- resource: Day2/BuildingImage at main · pingkunga/ (github.com)

Hand On - Building Image

- Download 2 Docker File
  - A DockerfileA at main · pingkunga/train-docker-at-BRSU (github.com)
  - B DockerfileB at main · pingkunga/train-docker-at-BRSU (github.com)
- Build Images
- Use docker images, to compare size

```
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage> docker images "my_org_postgres*"
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
my_org_postgres     16.3      efdddf767550   3 hours ago   459MB
my_org_postgres-l   16.3      0e7877a24996   7 days ago    462MB
PS D:\2WarRoom\9PublicSpeaker\train-docker-at-BRSU\Day2\BuildingImage>
```

Share Image !!!



- docker build & push

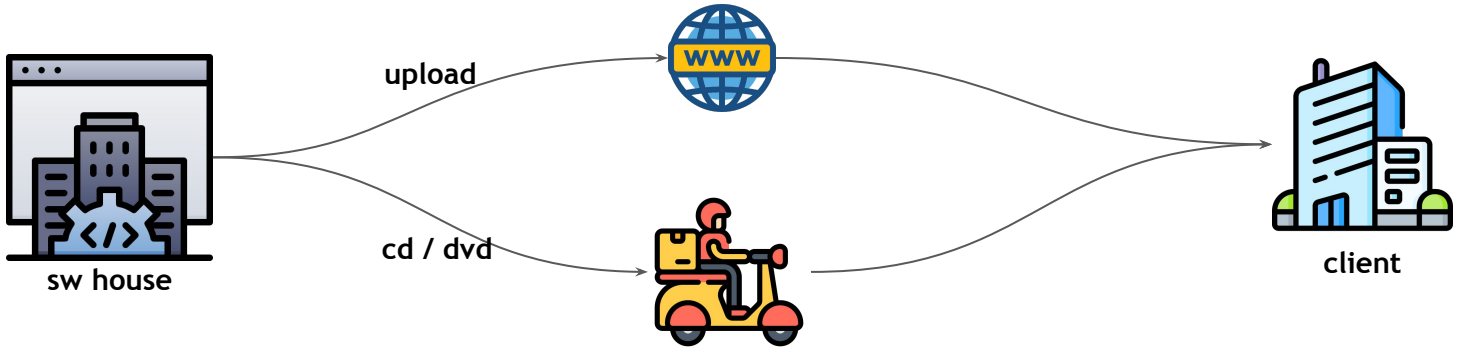
```
#build and push
docker buildx build --platform linux/amd64,linux/arm64 -t ourapp:latest . --push
```

- some case re-tag & Push

```
docker tag app1:1.0.0-dev some_hub/app1:1.0.0
docker push some_hub/app1:1.0.0
```

No Internet !!!

The customer said you cannot use the Internet to pull docker images, and there is no container registry.



- docker save

```
docker save -o sampleapp:25.05.24
```

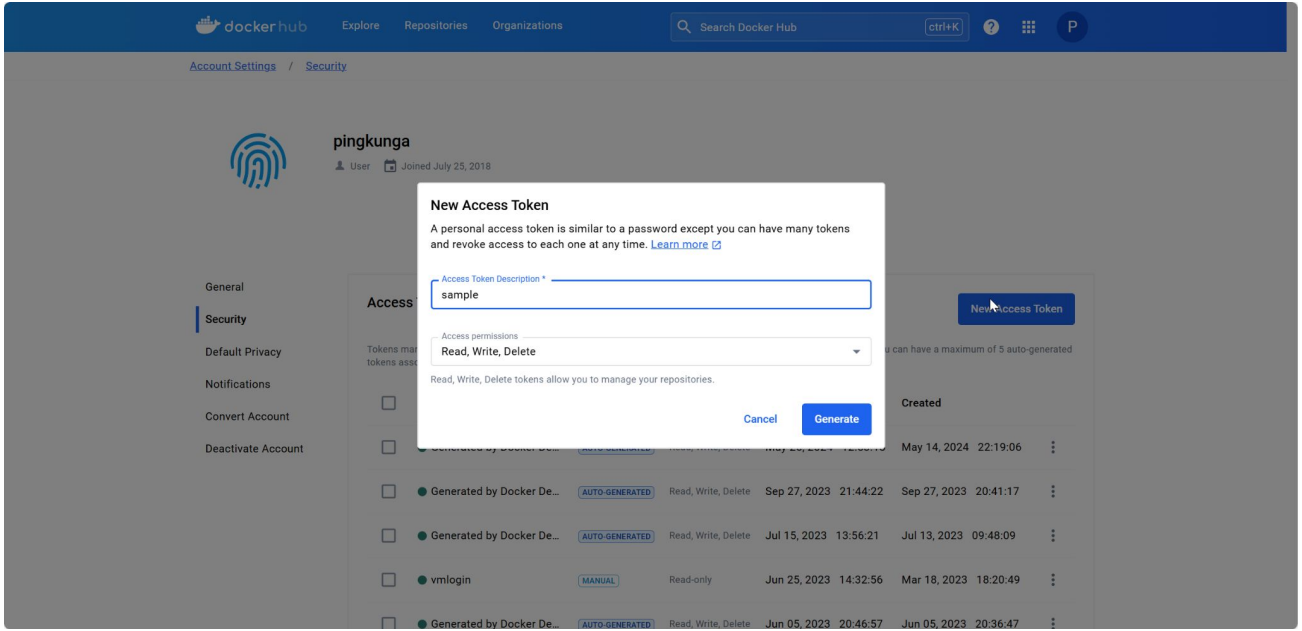
- docker image load

```
docker load --input sampleapp:25.05.24
```

- Note: some case re-tag & Push

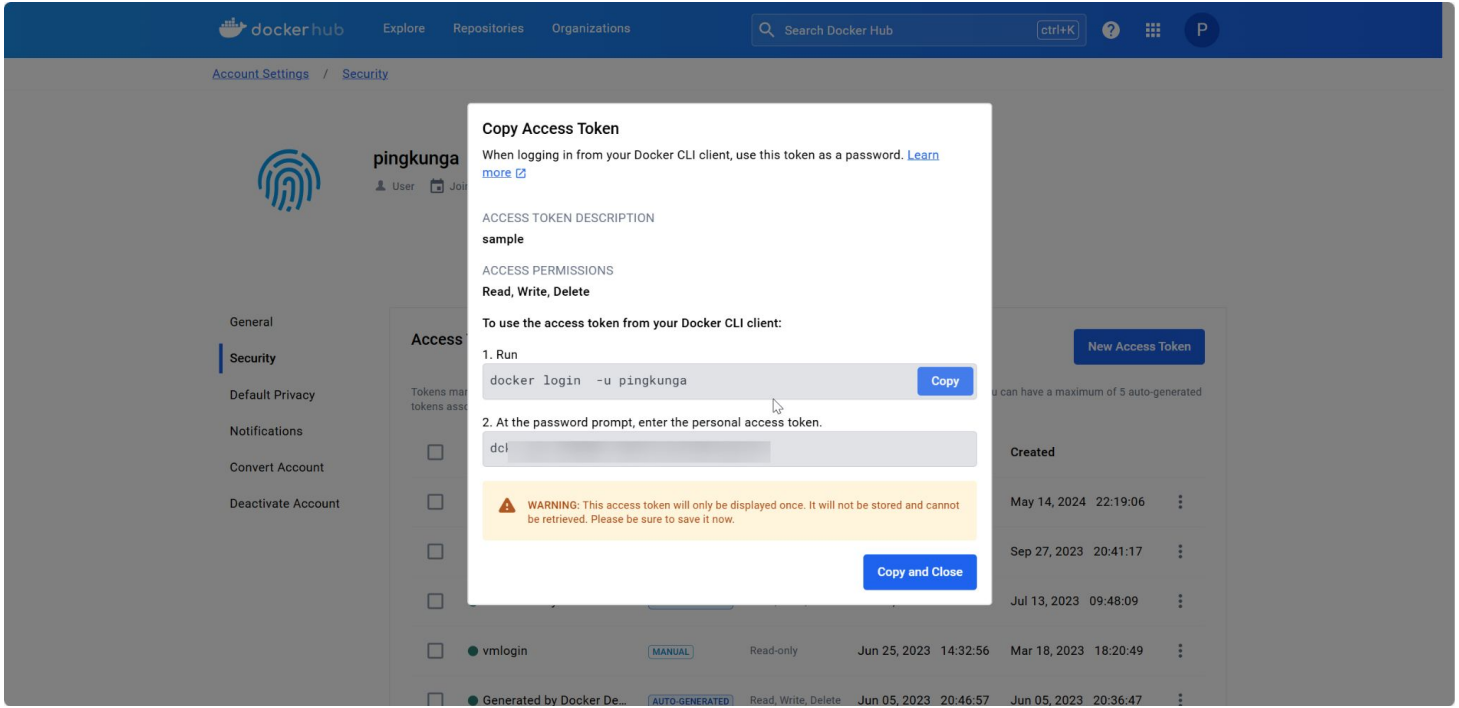
Hand-On Push Image to docker hub

- Generate Token & docker login



Hand-On Push Image to docker hub

- Run Command to connect a docker hub



Hand-On Push Image to docker hub

Case: Build & Push Command

```
docker buildx build --platform linux/amd64 -t
<docker_username>/my_org_postgres:16.3 . --push
```

Case: Retag & Push Command

```
docker tag my_org_postgres:16.3 <docker_username>/my_org_postgres:16.3
```

```
docker push <docker_username>/my_org_postgres:16.3
```

Container Best Practice

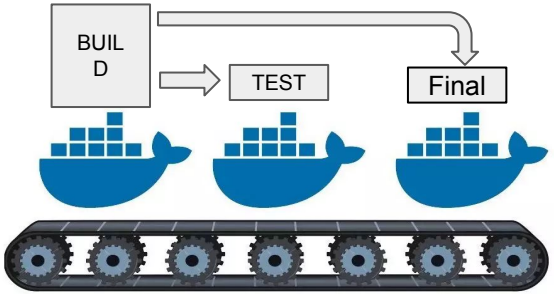
- Train your Team !!!
- Use trusted base images
- Keep images up to date
  - Fixed Version
  - With Schedule Update
- Reduce the attack surface (Alphine/ Chiselled / Chainguard Image / Distroless)
- Limit container privileges (Non root user in dockerfile **USER**)
- Limit container resource
- Seperate Data from Container (Design Data Path for Mount / Volume)

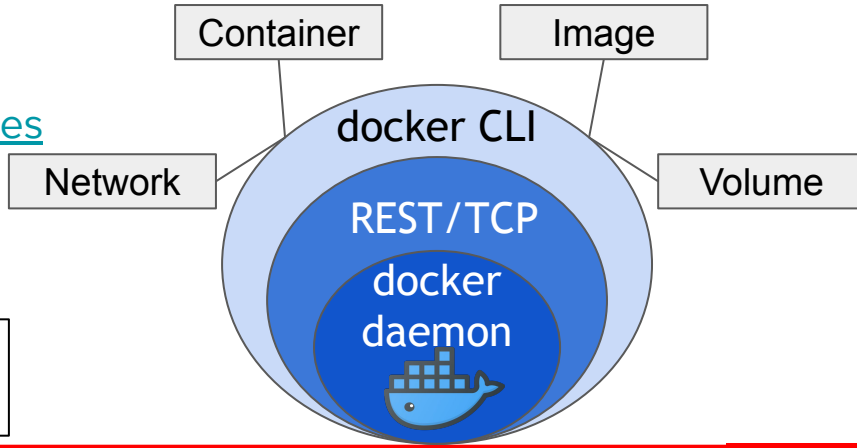
Container Best Practice

- Reduce Layer
  - Merge Layer
  - Multi-Stage

  Blog: Optimize Container image size
- Implement Access Controls
  - Host VM
  - Docker
    - Docker Security - OWASP Cheat Sheet Series
    - CIS Docker Benchmarks (cisecurity.org)
    - docker/docker-bench-security(github.com)
  - K8S

Rootless Container Enhancing application container security and compliance with Podman (redhat.com)





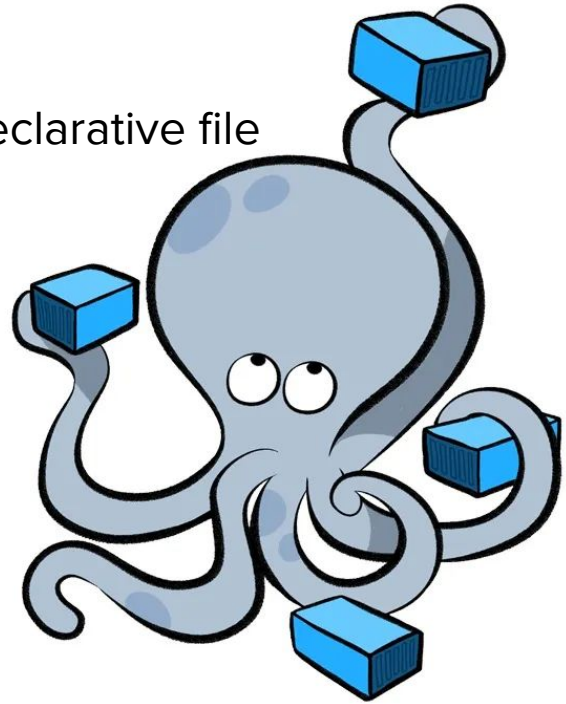
Container Best Practice

- Scan Images for vulnerabilities (Docker Scout / Trivy)
- Container Image Signing - (**Cosign** / docker trust) - supply chain attack
- Implement network security
- Monititor / Audit Activity (Observability)

Docker Compose

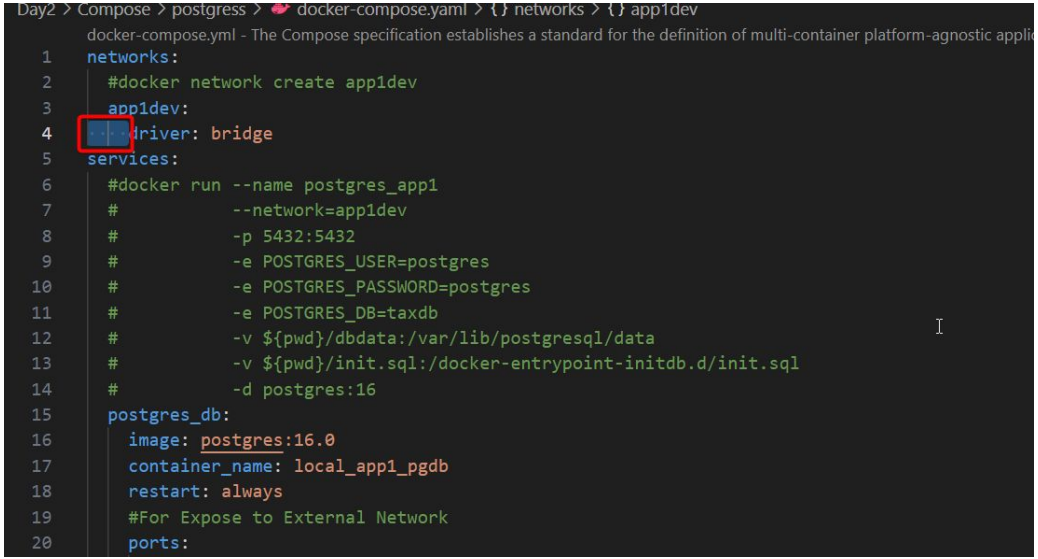
Docker Compose

- Old a lot of command
  - Solution: Script .sh / ps
  - Problem: Container Dependency / Human Error
- New way > Define final state of system in declarative file (docker-compose.yaml)



YAML

- YAML is a data-serialization language that is human-friendly format like json
- **Indentation -** recommended indentation for YAML files is two spaces per level



Ref: [A Gentle Introduction to the YAML format - DEV Community]

Docker Compose

- **Network**
- **Service**
- Config
- Secret
- **Volume**

Full Sample Compose File
[/Day2/Compose/SampleStructure.yaml at main · pingkunga(github.com)]

Day2 > Compose > ! SampleStructure.yaml > { } volumes > db_data

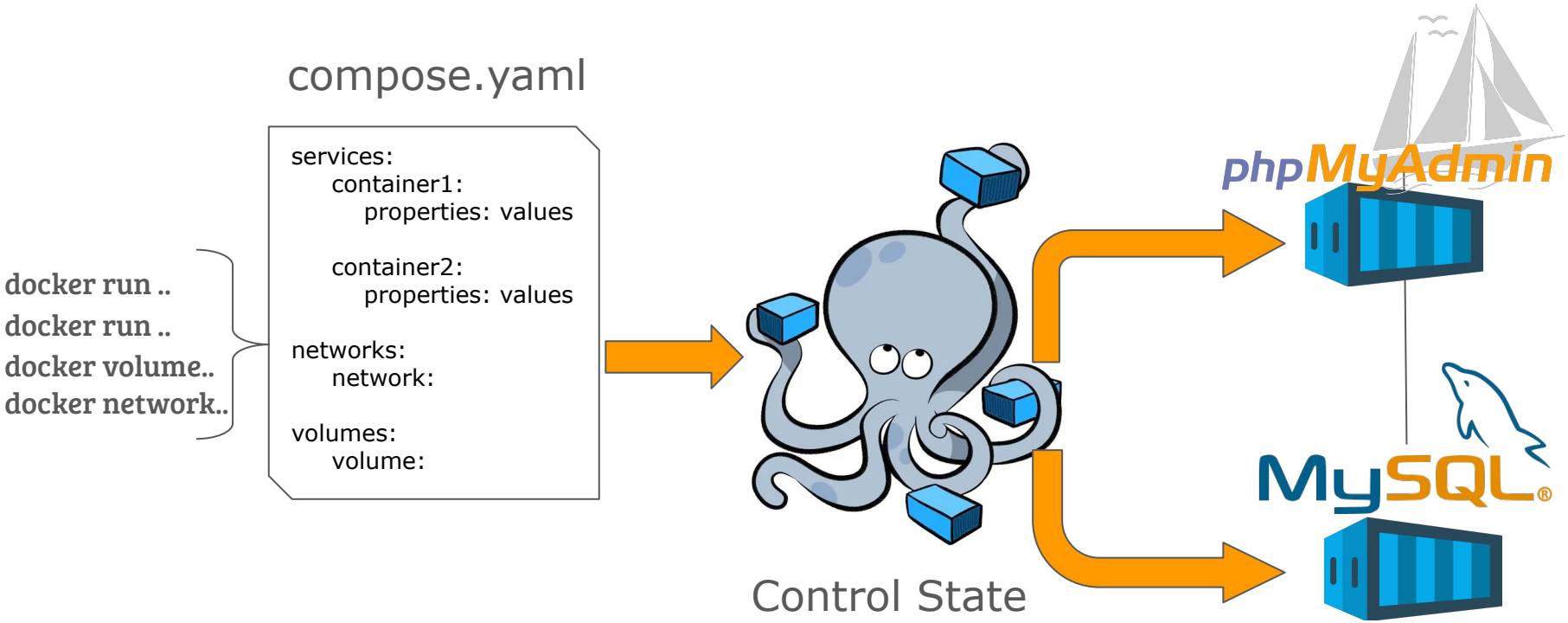
```yaml
networks:
  wordpress:
    driver: bridge

services:
  db: ···

  wordpress: ···

# Note Good Practice to use secrets for passwords
# config:
#     db_root_password:
#       file: db_root_password.txt
#     db_password:
#       file: db_password.txt
secrets:
    db_password:
      file: db_password.txt
    db_root_password:
      file: db_root_password.txt

volumes:
    db_data:
```

Convert Docker Command to Compose


```
docker run -d --name phpmyadmin \
  -e PMA_HOST= mysqldb \
  -e PMA_USER=root \
  -e PMA_PASSWORD=test \
  -p 8080:80 \
  --restart always \
  --link mysqldb\
  phpmyadmin/phpmyadmin:5.2
```

```
docker run -d -p 3399:3306 \
          --name mysqldb
          -e MYSQL_ROOT_PASSWORD=test \
          -e MYSQL_DATABASE=dev \
          -v mysql-data:/var/lib/mysql \
          mysql:8.0
```

Convert Docker Command to Compose



Convert Docker Command to Compose

- Declare Service (Container name and Base Image) create docker-compose.yaml

```
docker run -d -p 3399:3306 \
        --name mysqldb
        -e MYSQL_ROOT_PASSWORD=test \
        -e MYSQL_DATABASE=dev \
        -v mysql-data:/var/lib/mysql \
        mysql:8.0
```

```
services:
  mysqldb:
    image: mysql:8.0
```

Convert Docker Command to Compose

- Next, we bring over the port mapping.

```
docker run -d -p 3399:3306 \
        --name mysqldb
        -e MYSQL_ROOT_PASSWORD=test \
        -e MYSQL_DATABASE=dev \
        -v mysql-data:/var/lib/mysql \
        mysql:8.0
```

```
services:
  mysqldb:
    image: mysql:8.0
    ports:
    - 3399:3306
```

Convert Docker Command to Compose

- Next, environment variables.

```
docker run -d -p 3399:3306 \
        --name mysqldb
        -e MYSQL_ROOT_PASSWORD=test \
        -e MYSQL_DATABASE=dev \
        -v mysql-data:/var/lib/mysql \
        mysql:8.0
```

```
services:
  mysqldb:
    image: mysql:8.0
    ports:
    - 3399:3306
    environment:
    - MYSQL_ROOT_PASSWORD=test
    - MYSQL_DATABASE=dev
```

Convert Docker Command to Compose

- Next, define the volume, ensuring our data is persisted across environment restart

```
docker run -d -p 3399:3306 \
        --name mysqldb
        -e MYSQL_ROOT_PASSWORD=test \
        -e MYSQL_DATABASE=dev \
        -v mysql-data:/var/lib/mysql \
        mysql:8.0
```

```
services:
  mysqldb:
    image: mysql:8.0
    ports:
    - 3399:3306
    environment:
    - MYSQL_ROOT_PASSWORD=test
    - MYSQL_DATABASE=dev
    volumes:
    - mysql-data:/var/lib/mysql
volumes:
  mysql-data:
```

Convert Docker Command to Compose

- check point: test docker compose file
  - Day2/Compose/mysql/Compose1.yaml at main · pingkunga (github.com)

Basic Compose commands

```
# Use Compose to reconcile and launch what’s defined
docker compose up

# Start Compose in background mode
docker compose up -d

# Monitor Docker Compose Status
docker compose ps

# Tear everything down (leaves volumes by default)
docker compose down

# Share logs from all of the application services
docker compose logs

# or specific compose file
docker compose -f <your-compose-file>.yaml up / down / logs
```

Convert Docker Command to Compose

- Next Convert docker command for phpmyadmin to compose

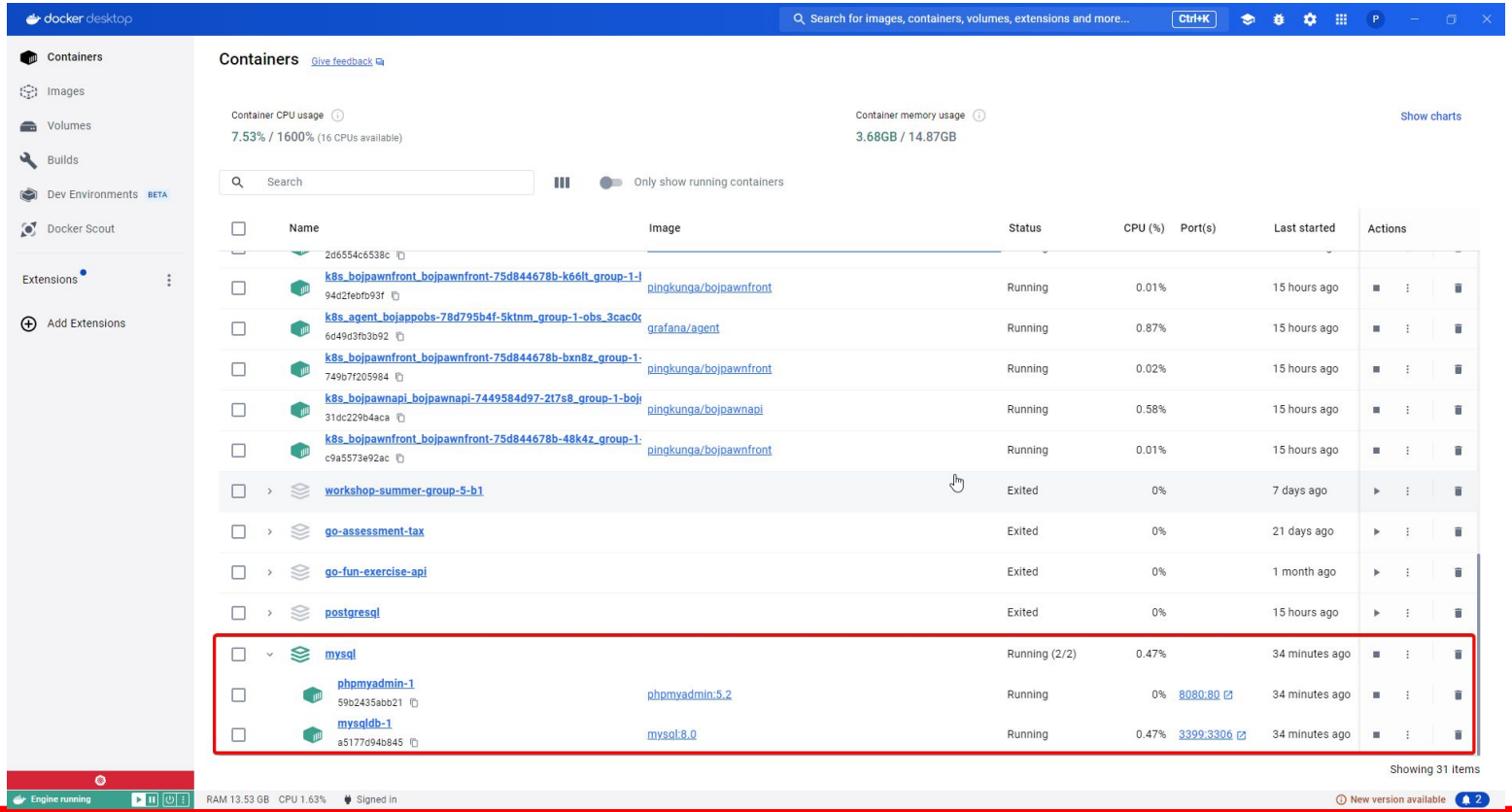
```
docker run -d --name phpmyadmin \
  -e PMA_HOST= mysqldb \
  -e PMA_USER=root \
  -e PMA_PASSWORD=test \
  -p 8080:80 \
  --restart always \
  --link mysqldb\
  phpmyadmin/phpmyadmin:5.2
```

```
services:
  mysqldb:
    …
  phpmyadmin:
    image: phpmyadmin:5.2
    ports:
    - 8080:80
   restart: always
   depends_on:
    - mysqldb
   links:
    - mysqldb
   environment:
      PMA_HOST: mysqldb
      PMA_USER: root
      PMA_PASSWORD: test
volumes:
  mysql-data:
```

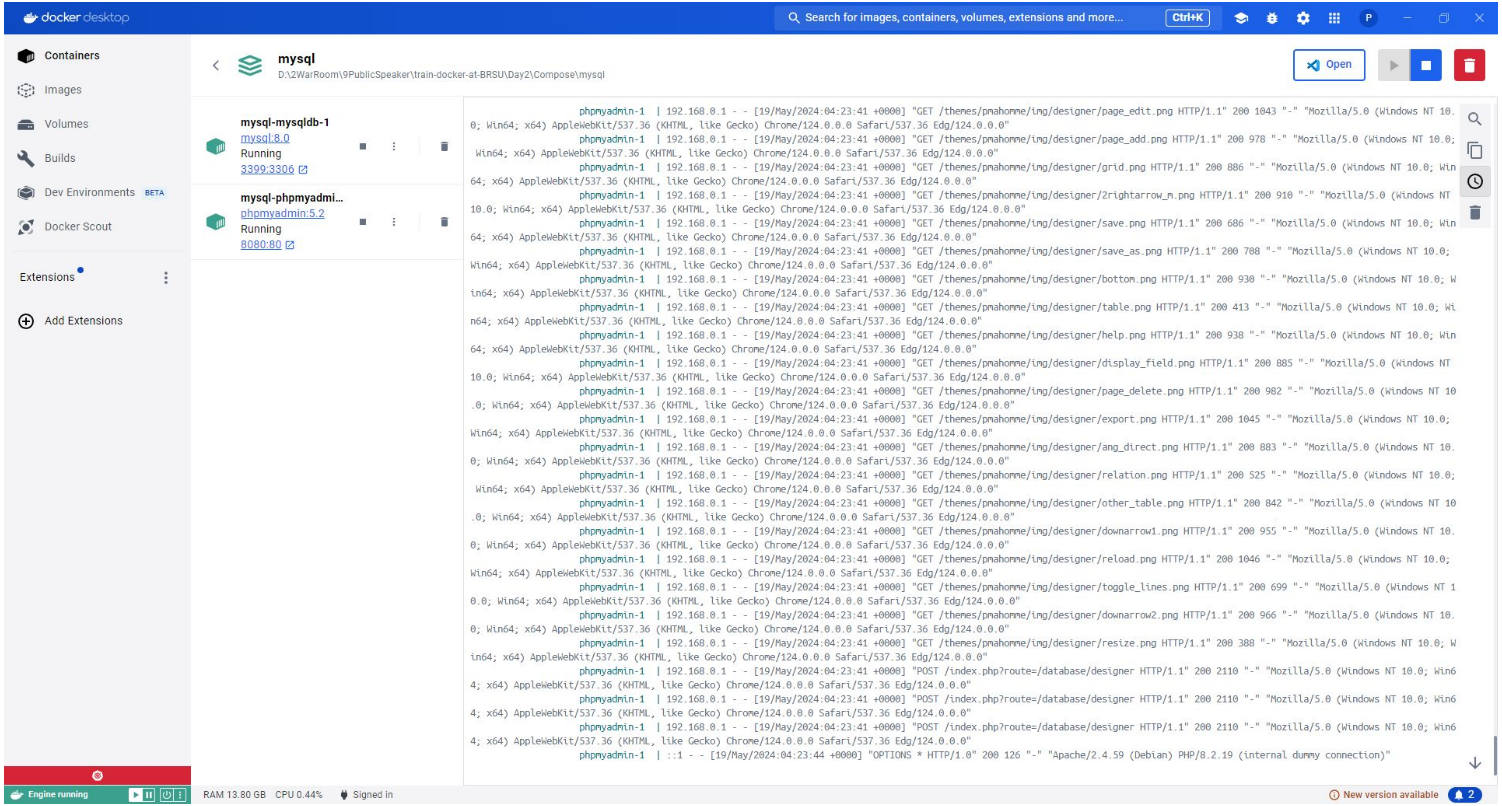
Convert Docker Command to Compose

- Full Compose File: mysql/Compose.yaml at main · pingkunga (github.com)
- Test

docker compose - Docker Desktop



docker compose - Docker Desktop



DOCKER COMPOSE CHEAT SHEET

File

structure

```
services:
   container1:
      properties: values

   container2:
      properties: values

networks:
   network:

volumes:
   volume:
```

Types

value

```
key: value
```

array

```
key:
   - value
   - value
```

dictionary

```
master:
   key: value
   key: value
```

Properties

build

build image from dockerfile in specified directory

```
container:
   build: ./path
   image: image-name
```

image

use specified image

```
image: image-name
```

container_name

define container name to access it later

```
container_name: name
```

volumes

define container volumes to persist data

```
volumes:
   - /path:/path
```

command

override start command for the container

```
command: execute
```

environment

define env variables for the container

```
environment:
   KEY: VALUE
---
environment:
   - KEY=VALUE
```

env_file

define a env file for the container to set and override env variables

```
env_file: .env
---
env_file:
 - .env
```

restart

define restart rule (no, always, on-failure, unless-stopped)

```
expose:
    - "9999"
```

networks

define all networks for the container

```
networks:
   - network-name
```

ports

define ports to expose to other containers and host

```
ports:
    - "9999:9999"
```

expose

define ports to expose only to other containers

```
expose:
    - "9999"
```

network_mode

define network driver (bridge, host, none, etc.)

```
network_mode: host
```

depends_on

define build, start and stop order of container

```
depends_on:
    - container-name
```

Other

idle container

send container to idle state > container will not stop

```
command: tail -f /dev/null
```

named volumes

create volumes that can be used in the volumes property

```
services:
  container:
    image: image-name
    volumes:
      - data-
volume:/path/to/dir

volumes:
  data-volume:
```

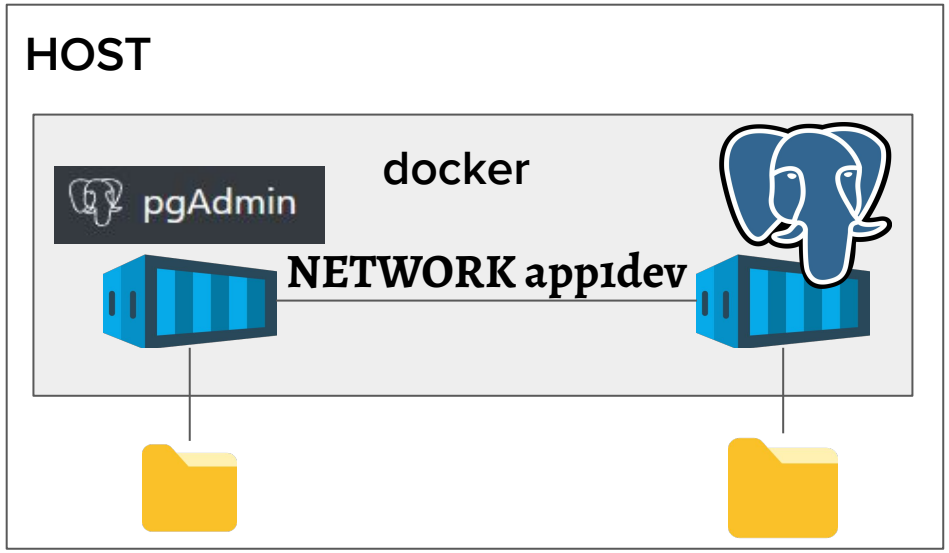
networks

create networks that can be used in the networks property

```
networks:
   frontend:
      driver: bridge
```

- Copy & Paste (Template) [docker/awesome-compose: Awesome Docker Compose samples (github.com)]

Hand On + Exercise



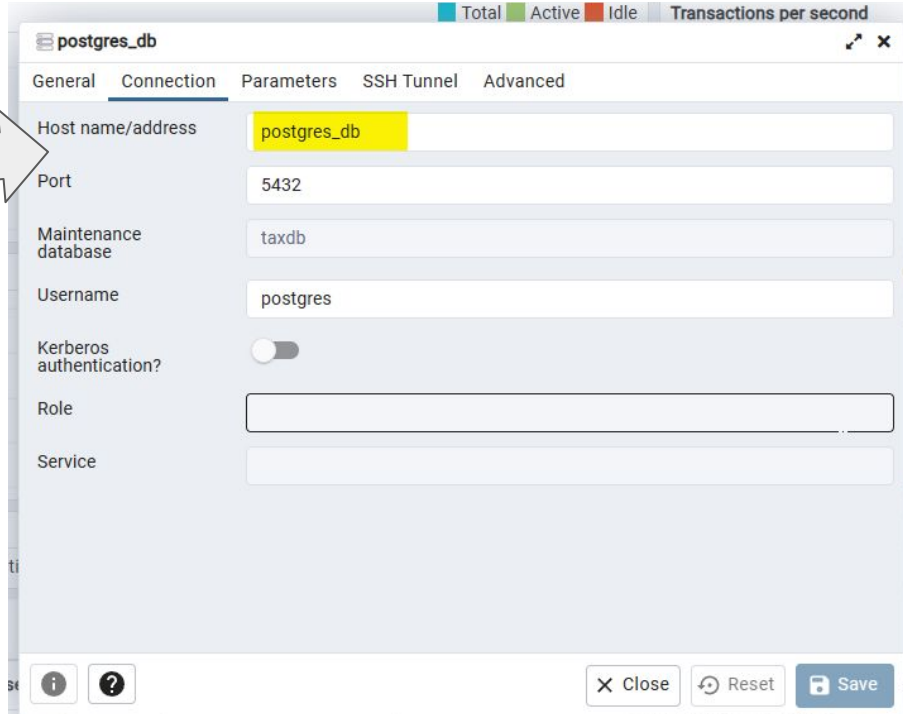
Hand On + Exercise Solution

- Full docker compose file postgress/docker-compose.yaml at main · pingkunga (github.com)

Hand On + Exercise Solution

- Test

```yaml
networks:
  app1dev:
    driver: bridge
services:
  postgres_db:
    image: postgres:16.0
    container_name: local_app1_pgdb
    restart: always
    ports:
      - "5432:5432"
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: taxdb
    volumes:
      #- local_app1_pgdata:/var/lib/postgresql/data
      - ./dbdata:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - app1dev
    healthcheck:
      test: ["CMD-SHELL", "pg_isready"]
```



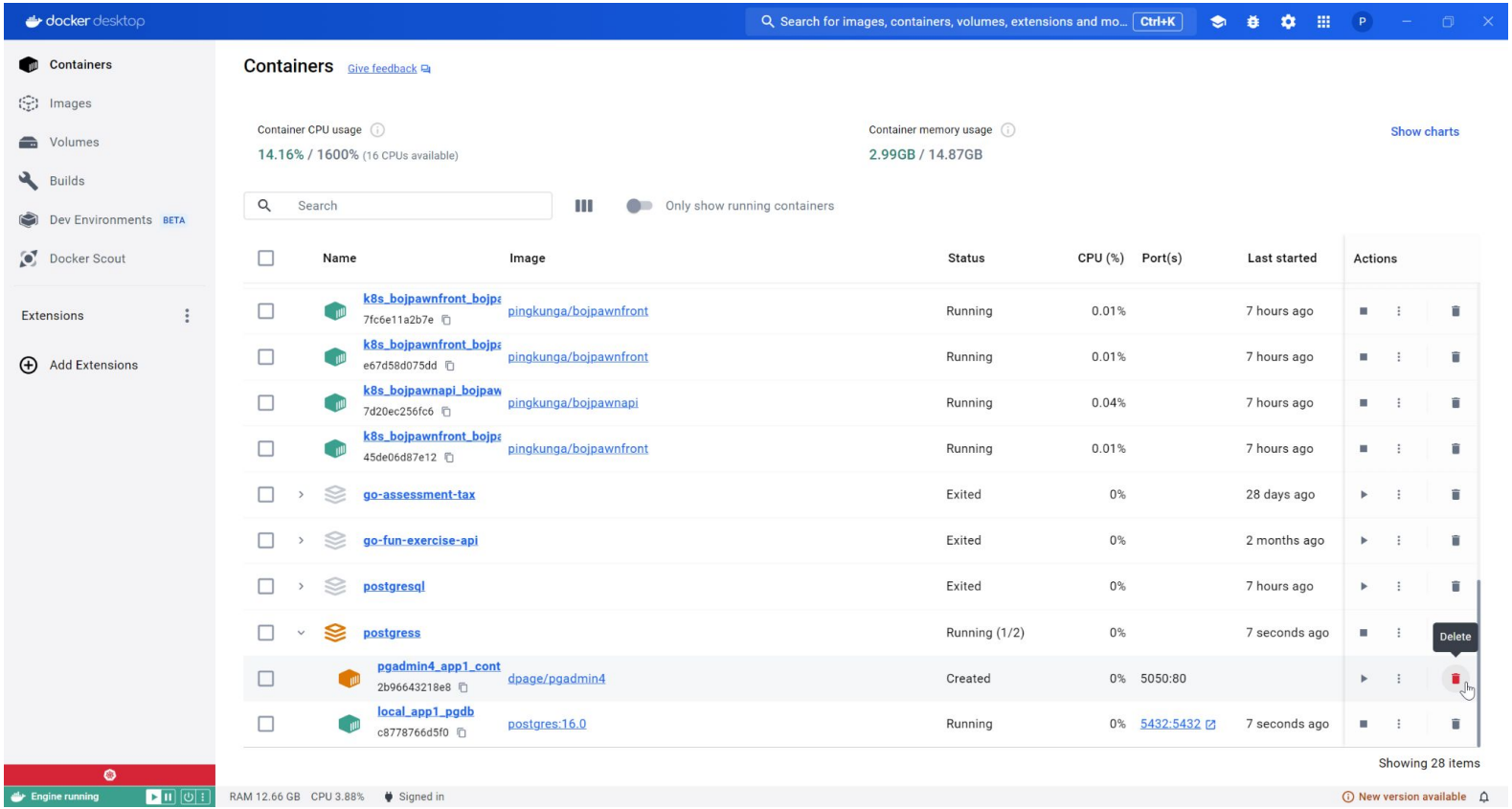
Hand On + Exercise Solution

- health check

```yaml
  pgadmin:
    image: dpage/pgadmin4
    container_name: pgadmin4_app1_container
    restart: always
    ports:
      - "5050:80"
    environment:
      PGADMIN_DEFAULT_EMAIL: pingkunga@example.com
      PGADMIN_DEFAULT_PASSWORD: VeryStrongP@ssw0rd
    volumes:
      - pgadmin-app1-data:/var/lib/pgadmin
    networks:
      - app1dev
    depends_on:
      postgres_db:
        condition: service_healthy
volumes:
  local_app1_pgdata:
  pgadmin-app1-data:
```

```yaml
# docker-compose.yml - The Compose specification establishes a standard for the definition of
networks:
  app1dev:
    driver: bridge
services:
  postgres_db:
    image: postgres:16.0
    container_name: local_app1_pgdb
    restart: always
    ports:
      - "5432:5432"
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: taxdb
    volumes:
      #- local_app1_pgdata:/var/lib/postgresql/data
      - ./dbdata:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - app1dev
    healthcheck:
      test: ["CMD-SHELL", "pg_isready"]
```

Ref: [Services elements - Health Check | Docker Docs]


docker desktop
Search for images, containers, volumes, extensions and mo... Ctrl+K
Containers
Images
Volumes
Builds
Dev Environments BETA
Docker Scout
Extensions
Add Extensions
Containers
Give feedback
Container CPU usage
14.16% / 1600% (16 CPUs available)
Container memory usage
2.99GB / 14.87GB
Show charts
Search
Only show running containers
Name
Image
Status
CPU (%)
Port(s)
Last started
Actions
k8s_bojpawnfront_bojpa 7fc6e11a2b7e
pingkunga/bojpawnfront
Running
0.01%
7 hours ago
k8s_bojpawnfront_bojpa e67d58d075dd
pingkunga/bojpawnfront
Running
0.01%
7 hours ago
k8s_bojpawnapi_bojpaw 7d20ec256fc6
pingkunga/bojpawnapi
Running
0.04%
7 hours ago
k8s_bojpawnfront_bojpa 45de06d87e12
pingkunga/bojpawnfront
Running
0.01%
7 hours ago
go-assessment-tax
Exited
0%
28 days ago
go-fun-exercise-api
Exited
0%
2 months ago
postgresql
Exited
0%
7 hours ago
postgress
Running (1/2)
0%
7 seconds ago
Delete
pgadmin4_app1_cont 2b96643218e8
dpage/pgadmin4
Created
0%
5050:80
local_app1_pgdb c8778766d5f0
postgres:16.0
Running
0%
5432:5432
7 seconds ago
Showing 28 items
Engine running
RAM 12.66 GB
CPU 3.88%
Signed in
New version available

Hand On + Exercise Solution

- Resource Limit

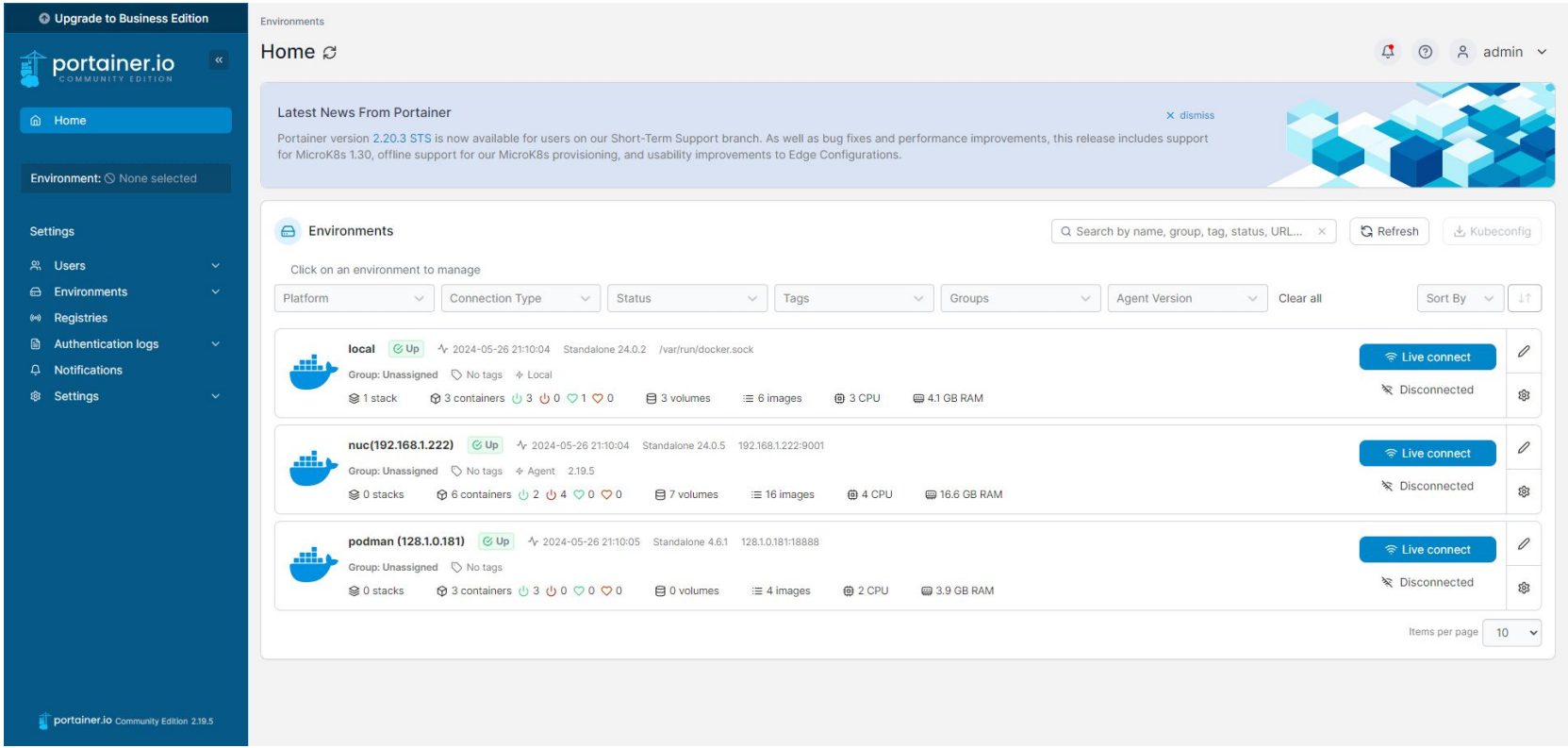
docker-compose.yml - The Compose specification establishes a standard for the definition of multi-container platform-agnostic applications (compose-spec.json)

```yaml
networks:
  app1dev:
    driver: bridge
services:
  postgres_db:
    image: postgres:16.0
    container_name: local_app1_pgdb
    restart: always
    #For Expose to External Network
    ports:
      - "5432:5432"
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: taxdb
    volumes:
      #- local_app1_pgdata:/var/lib/postgresql/data
      - ./dbdata:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
    networks:
      - app1dev
    healthcheck:
      test: ["CMD-SHELL", "pg_isready"]
    #Sample recommendation not fixed cpuset
    #cpuset: "0-1"
    deploy:
      resources:
        limits:
          cpus: '0.50'
          memory: 512M
        reservations:
          cpus: '0.25'
          memory: 256M
  pgadmin: ···
volumes:
  local_app1_pgdata:
  pgadmin-app1-data:
```

Ref: [Compose Deploy+ Resource Specification | Docker Docs]

Portainer

Portainer

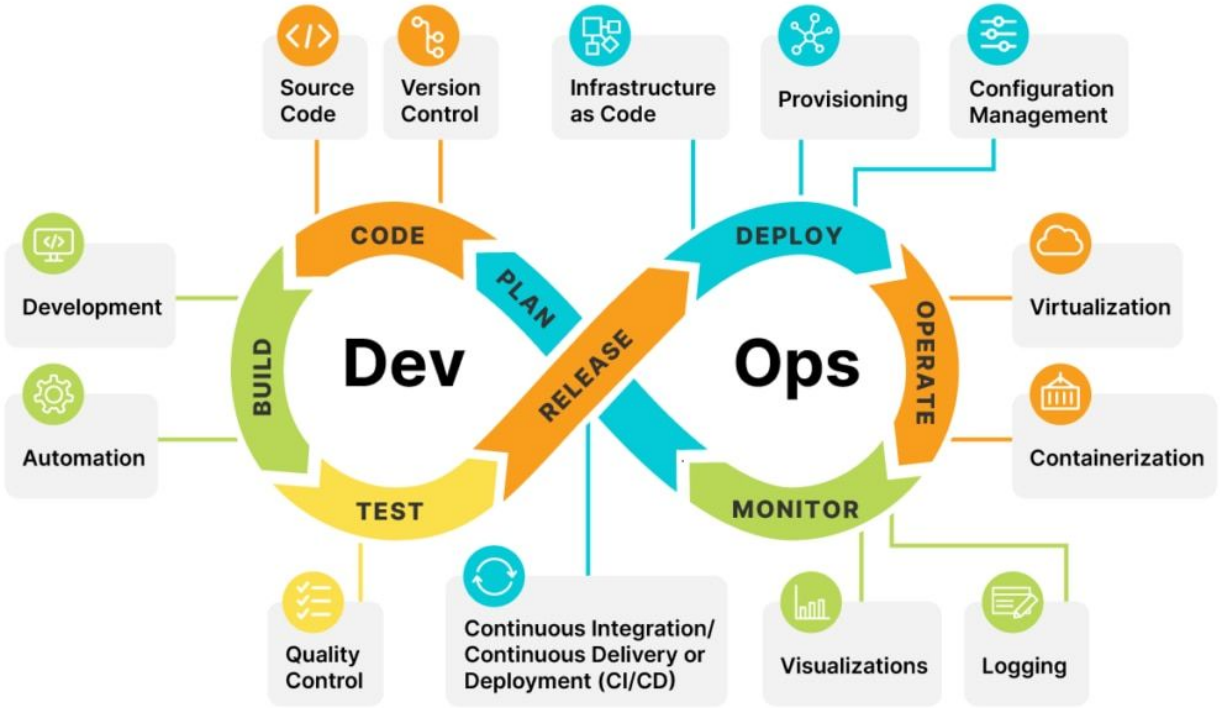

Upgrade to Business Edition
portainer.io
COMMUNITY EDITION
Home
Environment: None selected
Settings
Users
Environments
Registries
Authentication logs
Notifications
Settings
portainer.io Community Edition 2.19.5
Environments
Home
admin
Latest News From Portainer
dismiss
Portainer version 2.20.3 STS is now available for users on our Short-Term Support branch. As well as bug fixes and performance improvements, this release includes support for MicroK8s 1.30, offline support for our MicroK8s provisioning, and usability improvements to Edge Configurations.
Environments
Search by name, group, tag, status, URL...
Refresh
Kubeconfig
Click on an environment to manage
Platform
Connection Type
Status
Tags
Groups
Agent Version
Clear all
Sort By
local Up 2024-05-26 21:10:04 Standalone 24.0.2 /var/run/docker.sock
Group: Unassigned No tags Local
1 stack 3 containers 3 0 1 0 3 volumes 6 images 3 CPU 4.1 GB RAM
Live connect
Disconnected
nuc(192.168.1.222) Up 2024-05-26 21:10:04 Standalone 24.0.5 192.168.1.222:9001
Group: Unassigned No tags Agent 2.19.5
0 stacks 6 containers 2 4 0 0 7 volumes 16 images 4 CPU 16.6 GB RAM
Live connect
Disconnected
podman (128.1.0.181) Up 2024-05-26 21:10:05 Standalone 4.6.1 128.1.0.181:18888
Group: Unassigned No tags
0 stacks 3 containers 3 0 0 0 0 volumes 4 images 2 CPU 3.9 GB RAM
Live connect
Disconnected
Items per page 10

Introduction to Portainer

- Setup Day2/Portainer/docker-compose.yaml at main · pingkunga (github.com)
- Add Agent Install Portainer Agent on Docker Standalone | 2.19 | Portainer Documentation
  - Secure **“-e AGENT_SECRET=yoursecret”**
- Manage & Deploy Stack (Docker Compose)

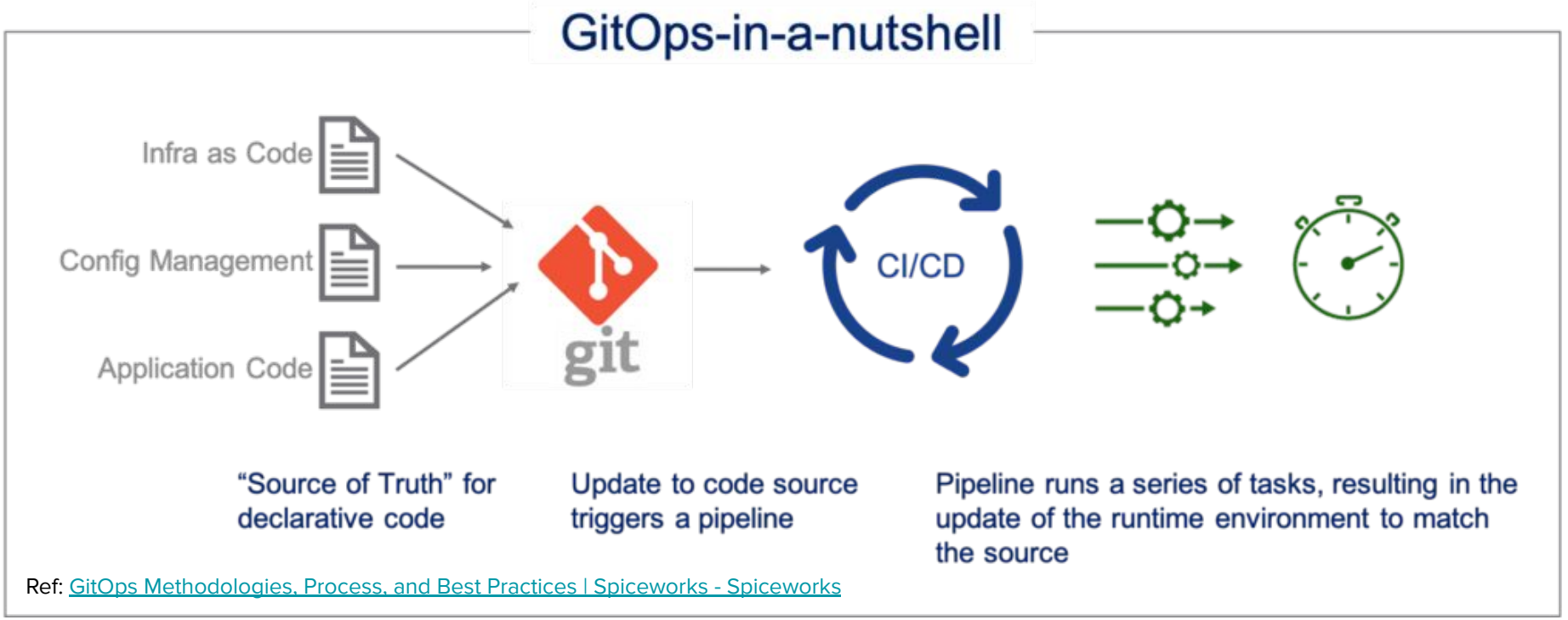
Ref: Getting Started - Portainer Knowledge Base

Next:Dev[XXX]Ops



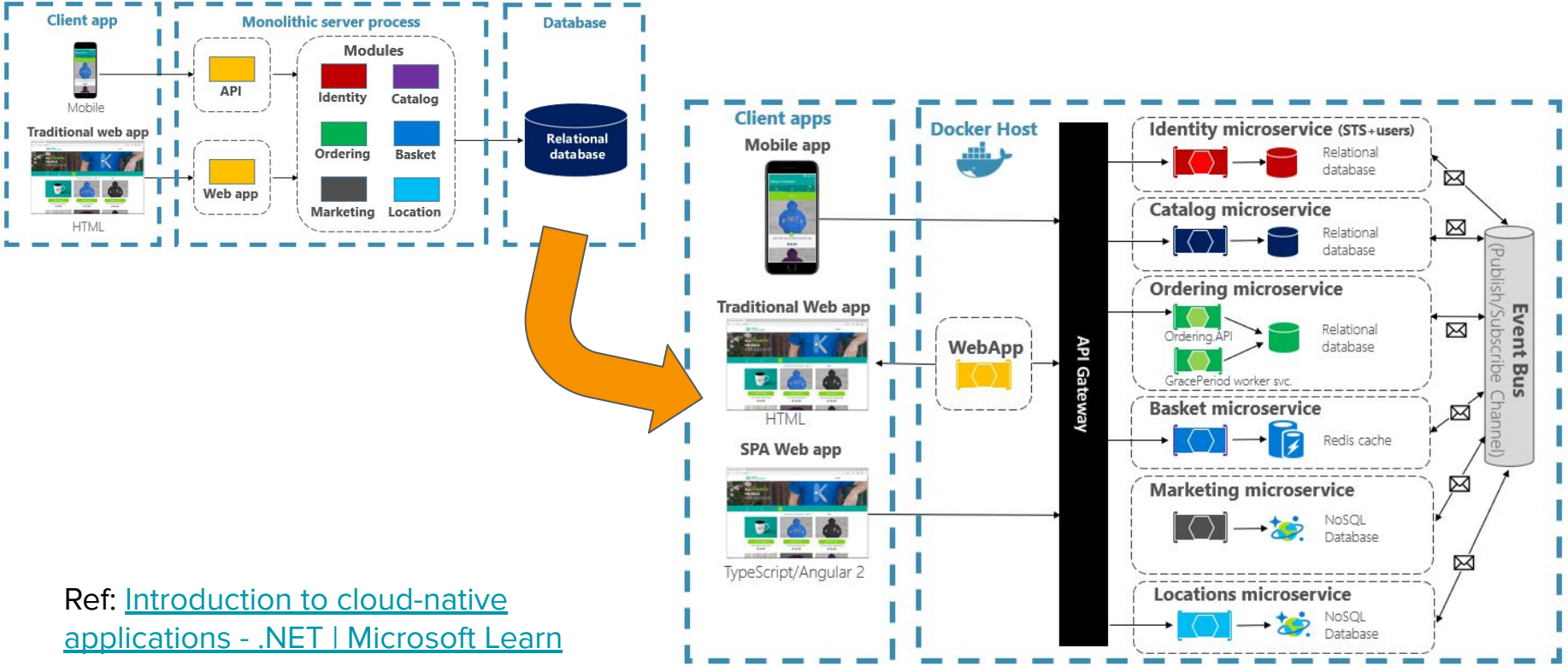
Ref: [What Is DevOps? Complete Guide to Best Practices - Orange Matter (solarwinds.com)]

Next: GitOps



Ref: GitOps Methodologies, Process, and Best Practices | Spiceworks - Spiceworks

Next: Cloud Native



Ref: [Introduction to cloud-native applications - .NET | Microsoft Learn]

Wrap-up

- Container
- Docker docker cli | Docker Docs
- Docker Compose docker compose speccification | Docker Docs
- Use-Case
  - DevOps
  - GitOps
  - Cloud Native App

Thank you

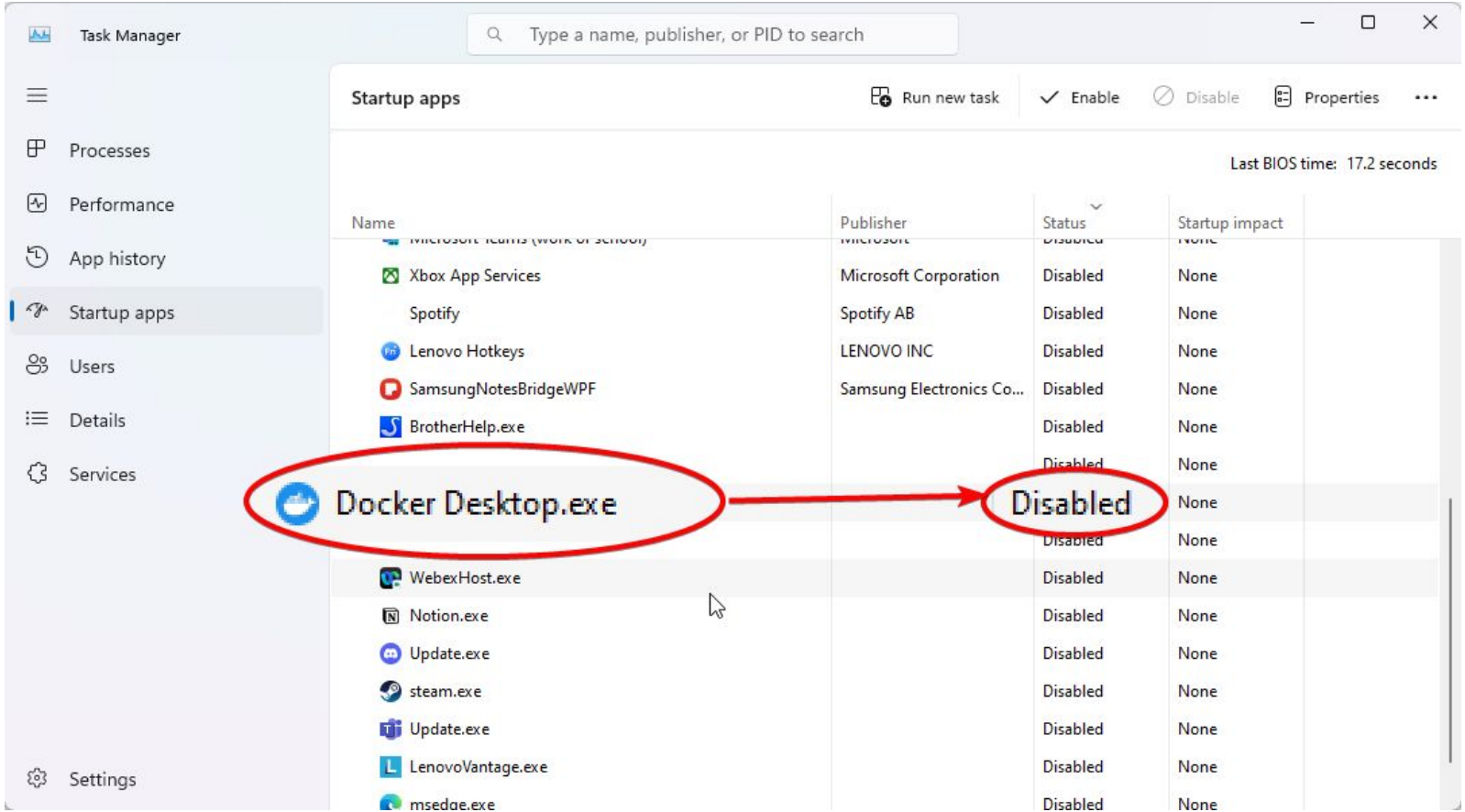
Resource pingkunga/train-docker-at-BRSU (github.com)

Feedback https://forms.gle/ySmFcNZawv3MDyQDA

Find me on:

CONTACT ME

Others


Task Manager
Type a name, publisher, or PID to search
Processes
Performance
App history
Startup apps
Users
Details
Services
Settings
Startup apps
Run new task
Enable
Disable
Properties
Last BIOS time: 17.2 seconds
Name
Publisher
Status
Startup impact
Xbox App Services
Microsoft Corporation
Disabled
None
Spotify
Spotify AB
Disabled
None
Lenovo Hotkeys
LENOVO INC
Disabled
None
SamsungNotesBridgeWPF
Samsung Electronics Co...
Disabled
None
BrotherHelp.exe
Disabled
None
Docker Desktop.exe
Disabled
None
WebexHost.exe
Disabled
None
Notion.exe
Disabled
None
Update.exe
Disabled
None
steam.exe
Disabled
None
Update.exe
Disabled
None
LenovoVantage.exe
Disabled
None
msedge.exe
Disabled
None

docker : invalid reference format - Stack Overflow

28 Answers

Sorted by: Highest score (default)

In powershell you should use `${pwd}` instead of `$(pwd)`

278

Share Edit Follow Flag

edited Mar 5, 2021 at 2:35
SiddAjmera
39k • 5 • 74 • 111

answered Jul 17, 2018 at 10:59
Levon Petrosyan
9,345 • 8 • 56 • 68

37 THIS is what caused me to bang my head into the wall (sometimes I hate M$FT) – beerwin Jan 15, 2019 at 10:25

1 `docker run --rm -ti --name zalenium -p 4444:4444 -p 5555:5555 \ -e SAUCE_USERNAME -e SAUCE_ACCESS_KEY \ -v /tmp/videos:/home/seluser/videos \ -v /var/run/docker.sock:/var/run/docker.sock \ dosel/zalenium start --sauceLabsEnabled true` what's wrong with my command? It is also giving same error. – paul Feb 4, 2019 at 5:45

2 For anyone having problem like @paul for a perfectly correct command, Please refer to this answer stackoverflow.com/a/65690853/807104 – Mohd Abdul Mujib Jan 12, 2021 at 19:37

docker & syslog

- จัดการ Log จาก Docker container ด้วย Fluentd (somkiat.cc)
- Fluentd logging driver | Docker Docs
- An Overview of Syslog Parsing with Fluentd | Tanmay Bhat (tanmay-bhat.github.io)

ทด Postgress Note for Legacy DB

- PostgreSQL : คำสั่ง SQL การ Create Database (mindphp.com)
- สร้าง Database เพื่อให้ใช้การเรียงลำดับ ภาษาไทยได้ | Simple has no boundaries... (wordpress.com)
- แก้ปัญหา update ubuntu server แล้วแจ้ง error ว่า perl: warning: Setting locale failed. – CoP PSU IT Blog
- How to change DATE command output language locales in Alpine-Linux? - Linux - nixCraft Linux/Unix Forum