Docker Up Your Development Environment

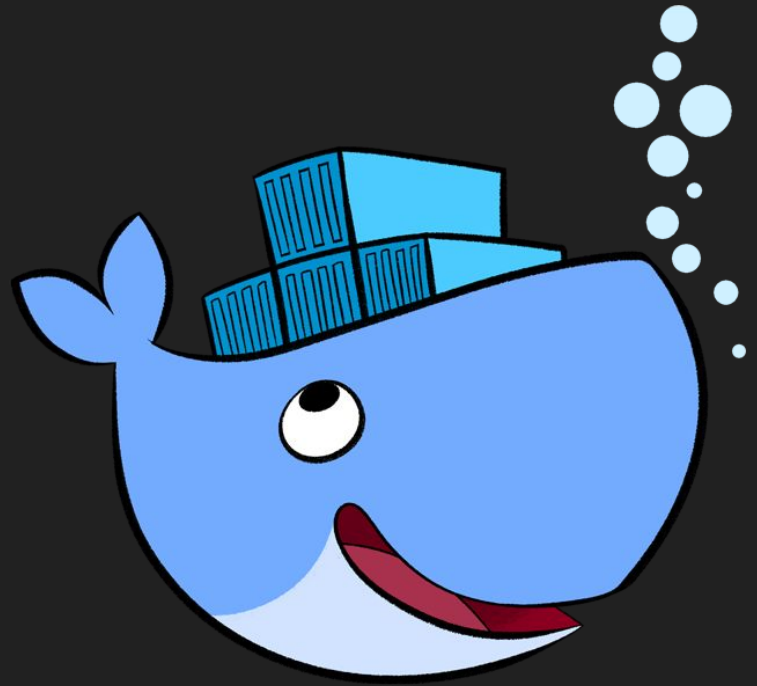
Daniel Schröder
skriptfabrik GmbH

Overview

1. Definitions of Docker, Containers and Images
2. Getting started with Docker Desktop
3. “Hello Docker!”
4. Going further with Docker Compose
5. Multi-Container Application
6. Dev Environment
7. Advanced Usages
8. Best Practices

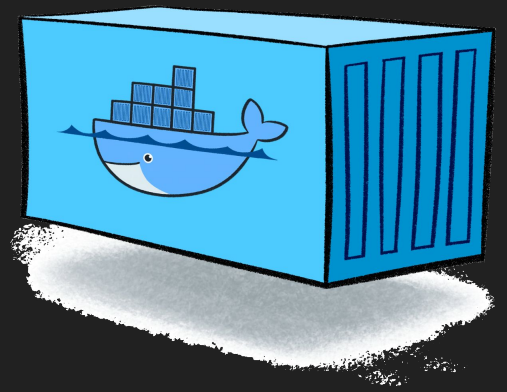
Docker

- free software for isolating applications using container virtualization
- written in Go
- simplifies application provisioning:
  - containers can be easily transported
  - containers contain necessary packages
- ensures the separation and management of resources used on a computer
- includes: code, runtime engine, system tools, system libraries, etc.



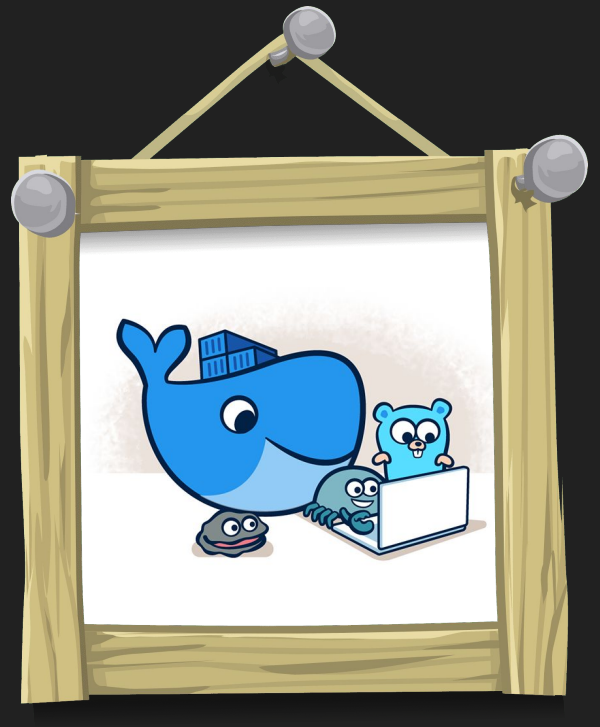
Containers

- sandboxed process on your machine
- isolated from all other processes on the host machine and from other containers
- runs its own software, binaries, and configurations
- is a runnable instance of an image
- can be created, started, stopped, moved, or deleted using the API or CLI
- can be run on local machines, virtual machines or deployed to the cloud
- portable (can be run on any OS)



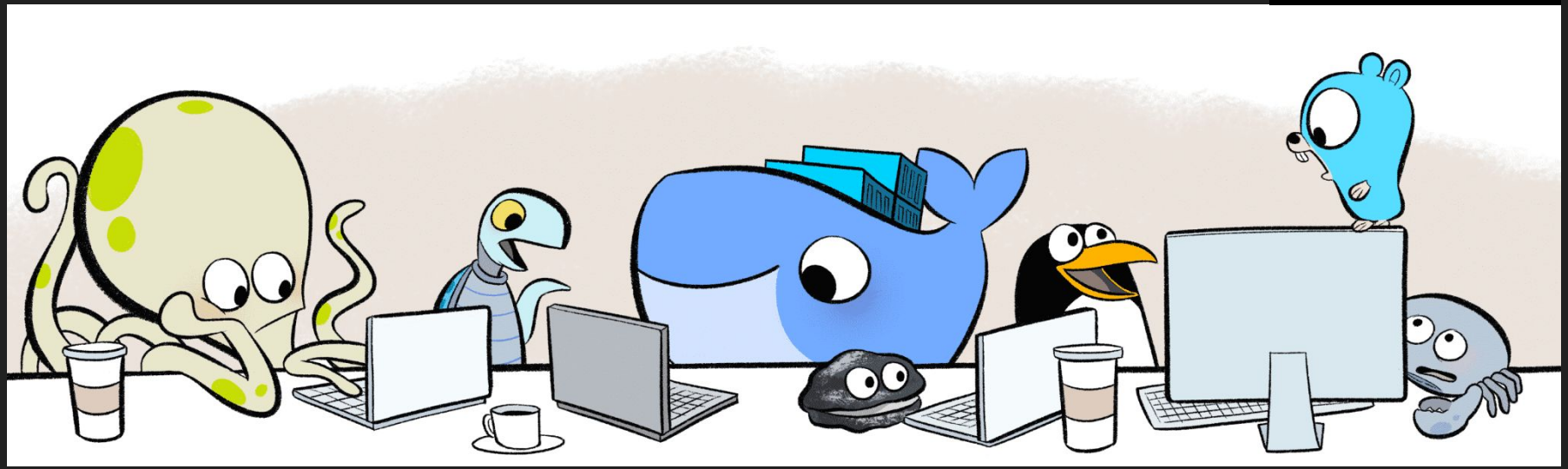
Images

- isolated filesystem for container is provided by an image
- must contain everything needed to run an application - all dependencies, configurations, scripts, binaries, etc.
- contains other configuration for the container: environment variables, default command, other metadata



Docker Desktop

- available for macOS, Windows and Linux: https://www.docker.com/products/docker-desktop/



Hello Docker!

My first Docker image

```
$ mkdir -p hello-docker/app
$ code hello-docker
```

Hello Docker!

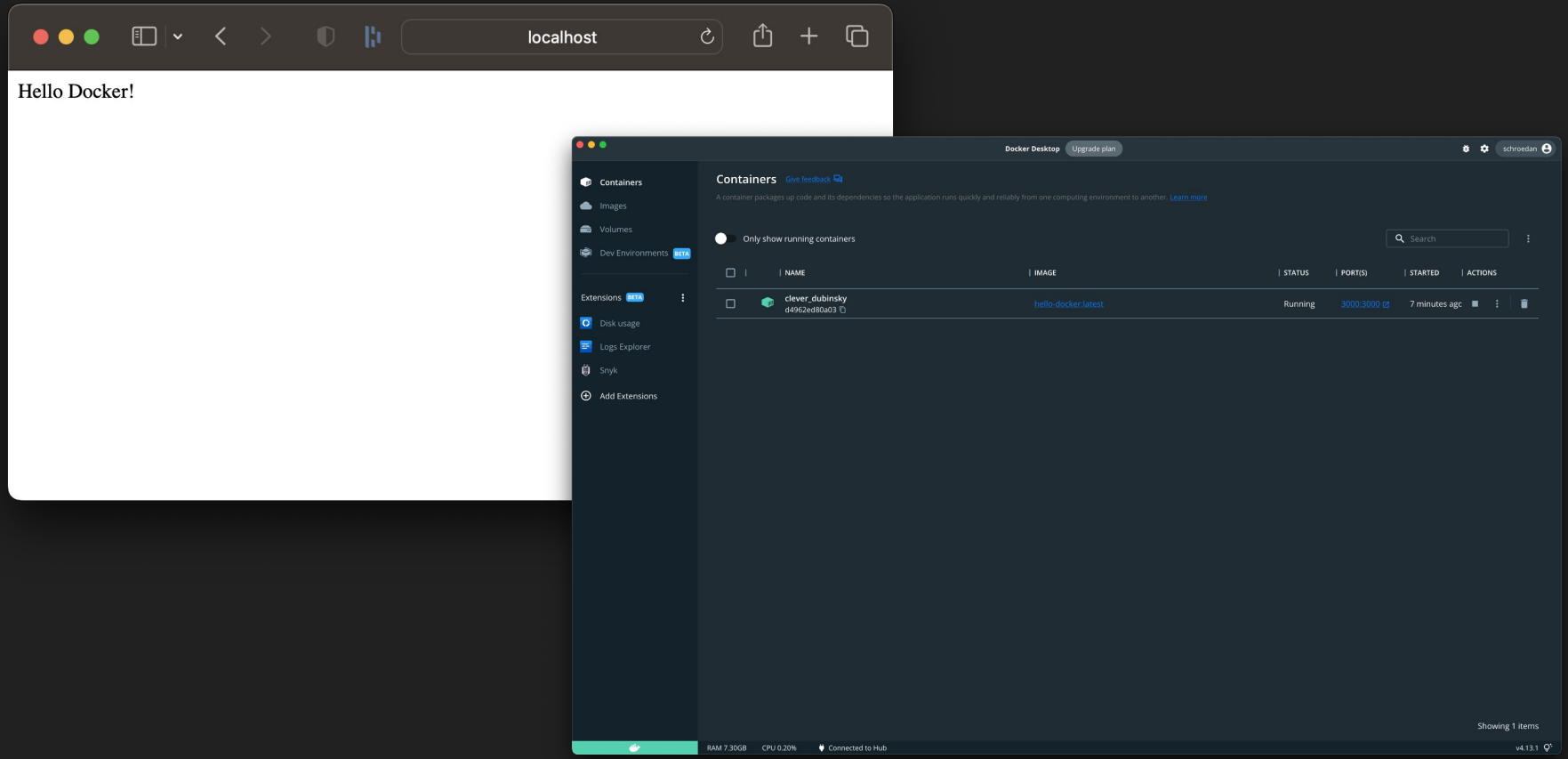
My first Docker image

```
# syntax=docker/dockerfile:1
FROM node:18-alpine
WORKDIR /app
COPY --link . .
RUN npm ci --prod
CMD ["node", "src/index.js"]
EXPOSE 3000
```

Hello Docker!

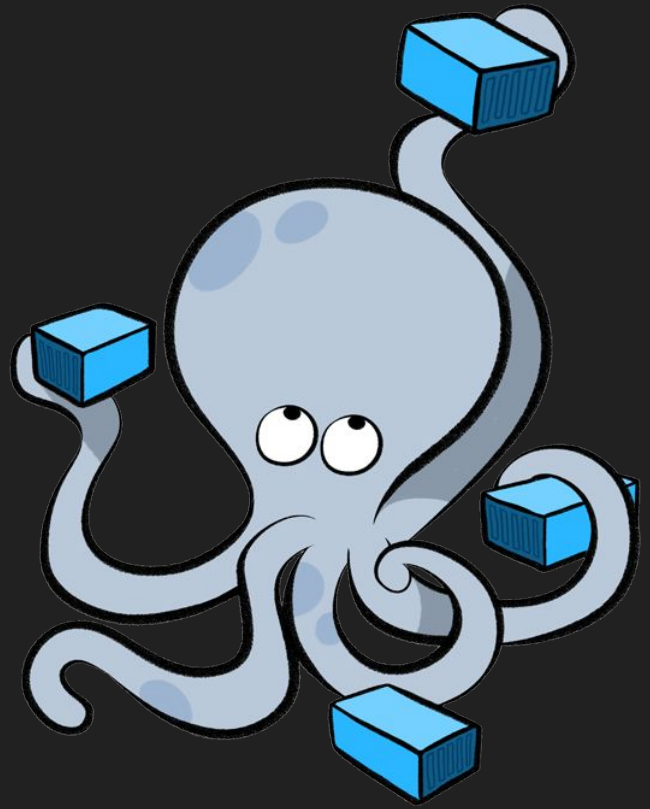
My first Docker image

```
$ docker build -t hello-docker
app
$ docker run --rm -p 3000:3000 \
    hello-docker
```


localhost
Hello Docker!
Docker Desktop
Upgrade plan
schroedan
Containers
Images
Volumes
Dev Environments
BETA
Extensions
BETA
Disk usage
Logs Explorer
Snyk
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
clever_dubinsky
d4962ed80a03
hello-docker:latest
Running
3000:3000
7 minutes ago
Showing 1 items
RAM 7.30GB
CPU 0.20%
Connected to Hub
v4.13.1

Docker Compose

- tool to help define and share multi-container applications
- define all the services as YAML file
- spin everything up or tear it all down with a single command
- advantage: define application stack in a single file so someone else would only need to clone your repo and start the compose app



Docker Compose

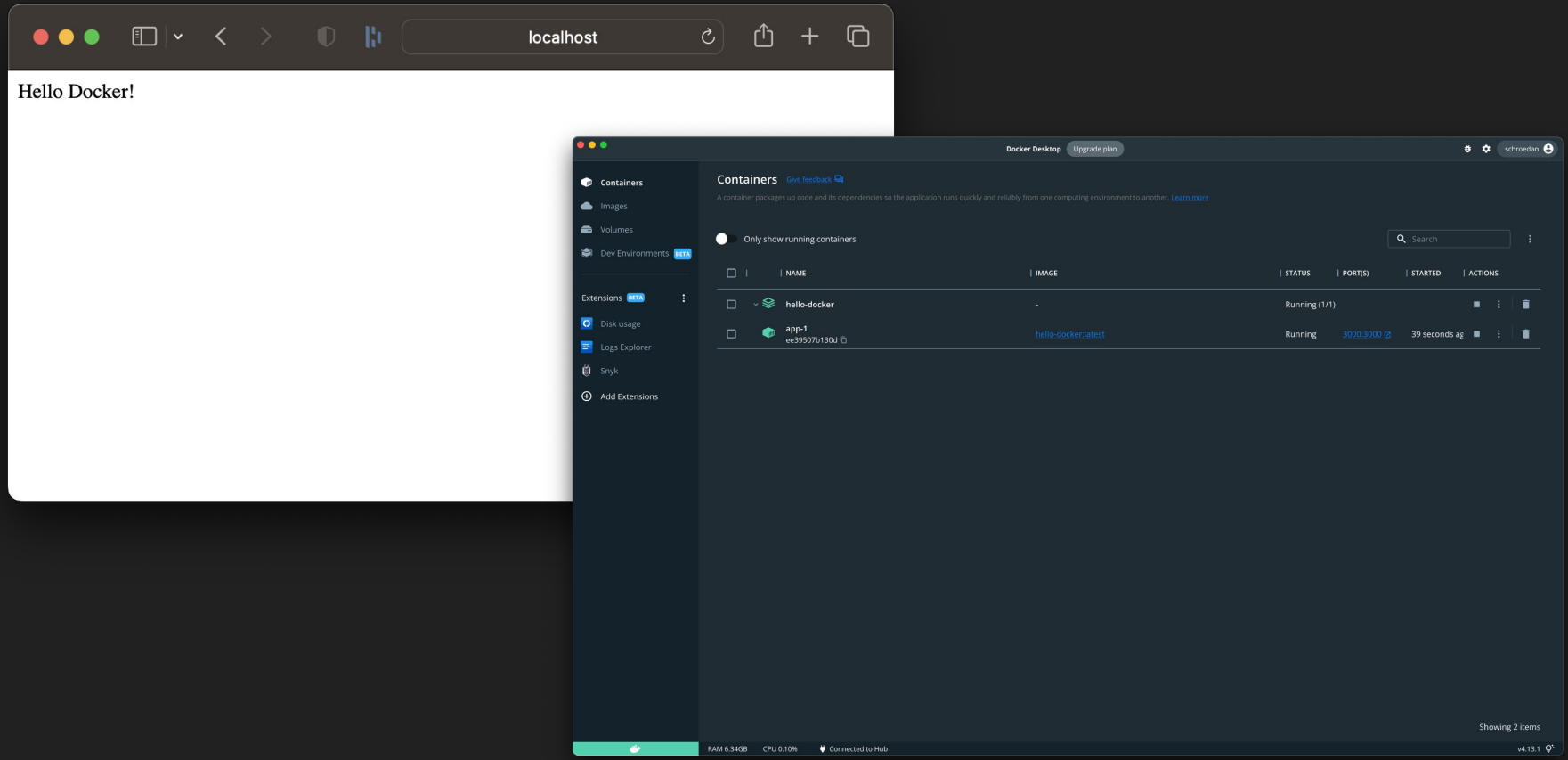
My first Compose file

```yaml
version: '3.8'
services:
  app:
    image: hello-docker
    build: app
    command: |
      sh -c 'npm ci && npm run dev'
    ports:
      - 3000:3000
    volumes:
      - ./app:/app
    working_dir: /app
```

Docker Compose

My first Compose file

```
$ docker compose build
$ docker compose up
```


localhost
Hello Docker!
Docker Desktop
Upgrade plan
schroedan
Containers
Images
Volumes
Dev Environments
BETA
Extensions
BETA
Disk usage
Logs Explorer
Snyk
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
hello-docker
-
Running (1/1)
app-1
ee39507b130d
hello-docker:latest
Running
3000:3000
39 seconds ag
Showing 2 items
RAM 6.34GB
CPU 0.10%
Connected to Hub
v4.13.1

Multi-Container Application

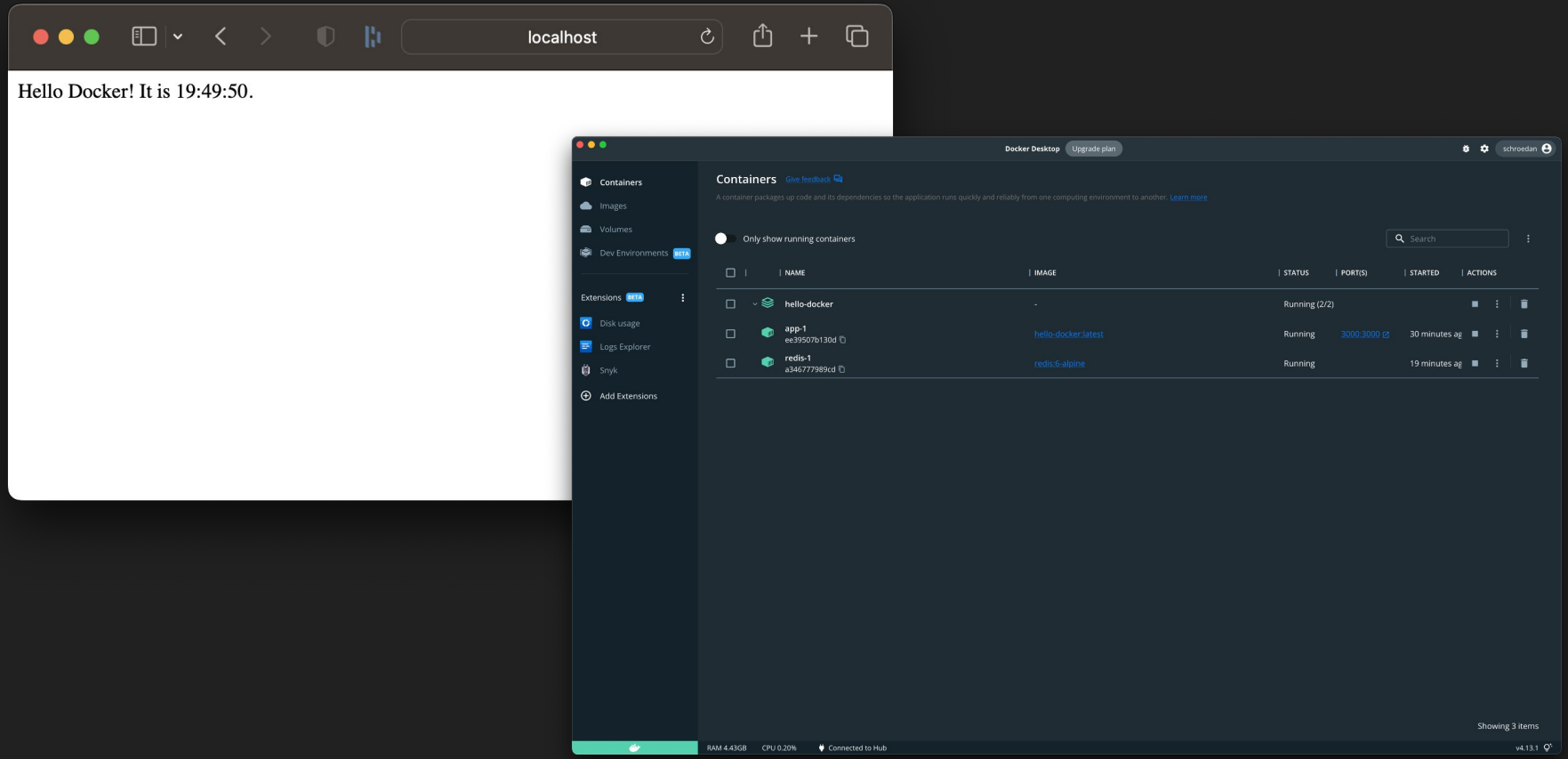
Additional Redis service

```yaml
version: '3.8'
services:
  app:
    ...
  redis:
    image: redis:6-alpine
```

Multi-Container Application

Additional Redis service

```
$ docker compose up -build
```


localhost
Hello Docker! It is 19:49:50.
Docker Desktop
Upgrade plan
schroedan
Containers
Images
Volumes
Dev Environments
BETA
Extensions
BETA
Disk usage
Logs Explorer
Snyk
Add Extensions
Containers
Give feedback
A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. Learn more
Only show running containers
Search
NAME
IMAGE
STATUS
PORT(S)
STARTED
ACTIONS
hello-docker
Running (2/2)
app-1
ee39507b130d
hello-docker:latest
Running
3000:3000
30 minutes ag
redis-1
a346777989cd
redis:6-alpine
Running
19 minutes ag
Showing 3 items
RAM 4.43GB
CPU 0.20%
Connected to Hub
v4.13.1

Dev Environment

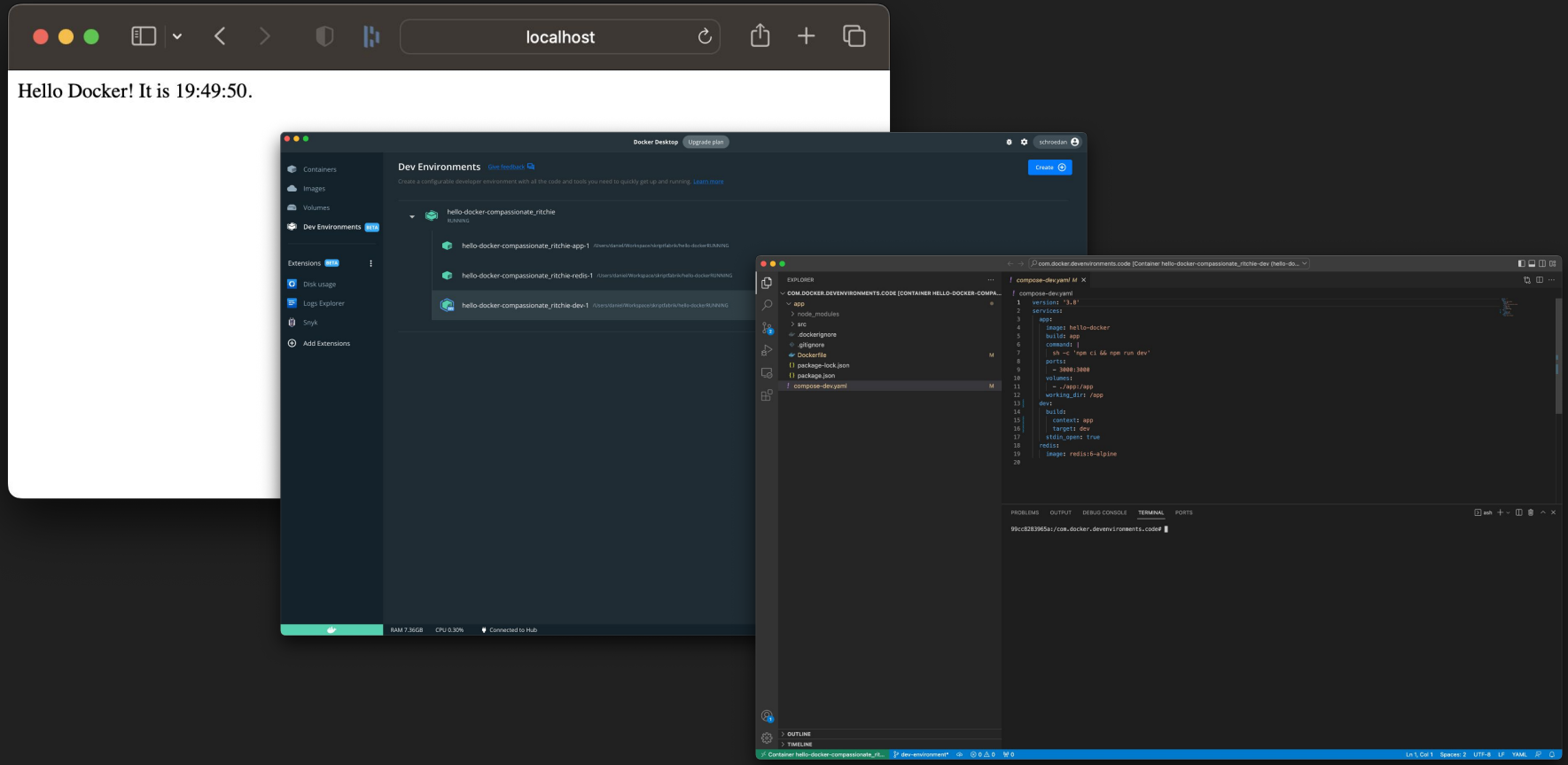
Development in a container

```
# syntax=docker/dockerfile:1
FROM node:18-alpine as node
FROM node as dev
RUN apk add --no-cache git
FROM node
WORKDIR /app
COPY --link . .
RUN npm ci --prod
CMD ["node", "src/index.js"]
EXPOSE 3000
```

Dev Environment

Development in a container

```
version: '3.8'
services:
  app:
    ...
  dev:
    build:
      context: .
      target: dev
    stdin_open: true
  redis:
    ...
```


localhost
Hello Docker! It is 19:49:50.
Docker Desktop
Upgrade plan
schroedas
Containers
Images
Volumes
Dev Environments
Extensions
Disk usage
Logs Explorer
Snyk
Add Extensions
Dev Environments
Give feedback
Create
Create a configurable developer environment with all the code and tools you need to quickly get up and running. Learn more
hello-docker-compassionate_ritchie
RUNNING
hello-docker-compassionate_ritchie-app-1
hello-docker-compassionate_ritchie-redis-1
hello-docker-compassionate_ritchie-dev-1
RAM 7.36GB
CPU 0.30%
Connected to Hub
EXPLORER
COM.DOCKER.DEVENVIRONMENTS.CODE [CONTAINER HELLO-DOCKER-COMPA...
app
node_modules
src
.dockerignore
.gitignore
Dockerfile
package-lock.json
package.json
compose-dev.yaml
compose-dev.yaml M
version: '3.8'
services:
app:
image: hello-docker
build: app
command: |
sh -c 'npm ci && npm run dev'
ports:
- 3000:3000
volumes:
- ./app:/app
working_dir: /app
dev:
build:
context: app
target: dev
stdin_open: true
redis:
image: redis:6-alpine
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
99cc828396a:/com.docker.devenvironments.code#
OUTLINE
TIMELINE
Container hello-docker-compassionate_rit...
dev-environment*
Ln 1, Col 1
Spaces: 2
UTF-8
LF
YAML

Advanced Usages

Pull latest images

```
$ docker compose pull \
    --ignore-pull-failures

[+] Running 1/2
 ⠿ redis Pulled
 ⠿ app Warning
```

Advanced Usages

Start all containers in detached mode

```
$ docker compose up -d

[+] Running 3/3
 ⠿ Network hello-docker_default
   Created
 ⠿ Container hello-docker-redis-1
   Started
 ⠿ Container hello-docker-app-1
   Started
```

Advanced Usages

Stop all containers

```
$ docker compose up -d
$ docker compose down

[+] Running 3/3
 ⠿ Container hello-docker-redis-1
   Removed
 ⠿ Container hello-docker-app-1
   Removed
 ⠿ Network hello-docker_default
   Removed
```

Advanced Usages

Follow output of app container logs

```
$ docker compose up -d
$ docker compose logs -f app

...
hello-docker-app-1  | [nodemon]
watching path(s): *.*
hello-docker-app-1  | [nodemon]
watching extensions: js,mjs,json
hello-docker-app-1  | [nodemon]
starting `node src/index.js`
hello-docker-app-1  | App
listening on port 3000
```

Advanced Usages

List status of all containers

```
$ docker compose up -d
$ docker compose ps -a

NAME
COMMAND
SERVICE   STATUS   PORTS

hello-docker-app-1
"docker-entrypoint.s…"
app running
0.0.0.0:3000->3000/tcp

hello-docker-redis-1
"docker-entrypoint.s…"
redis running 6379/tcp
```

Advanced Usages

Execute interactive shell in app container

```
$ docker compose up -d
$ docker compose exec app sh

/app # yarn --version
3.3.0
/app #
```

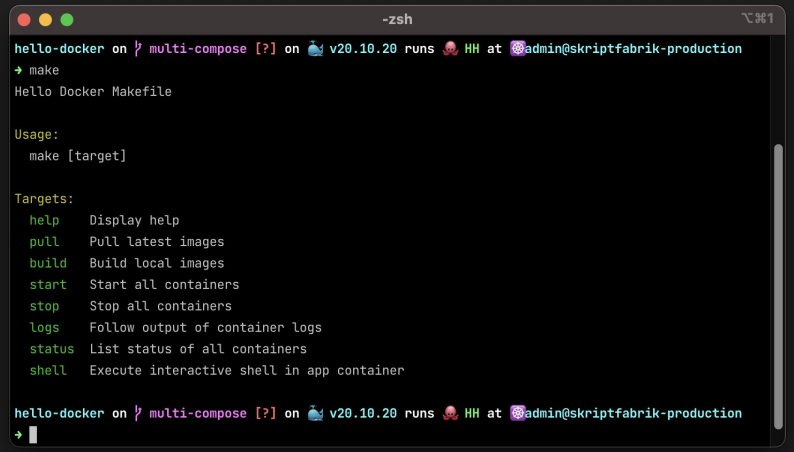
Advanced Usages

Run one-off command on a service

```
$ docker compose run --rm redis \
    redis-cli -h redis

redis:6379>
```

Best Practices



- create ephemeral containers
- understand build context
- exclude with .dockerignore
- don’t install unnecessary packages
- decouple applications
- minimize the number of layers: use multi-stage builds
- sort multi-line arguments
- leverage build cache

Questions?

Thank You!