Web Development Advanced Concepts: Docker Primer

Virtualization

Using containers with Docker

Just Overview, we go step-by-step

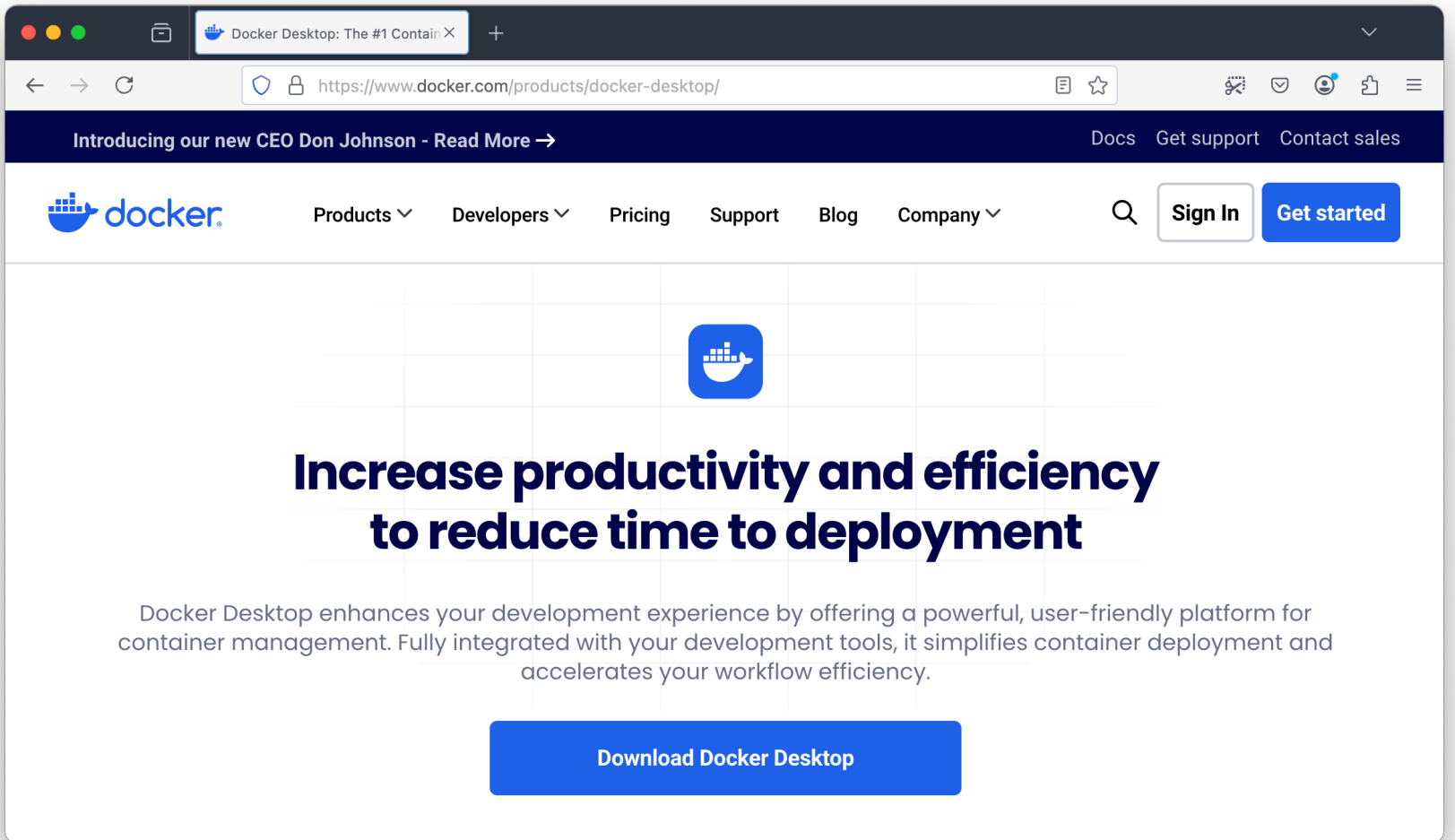
- Download and install Docker
- Pull the latest version of the nginx image from Docker Hub:
  `docker pull nginx:latest`
- Run a container and map port 80 from the container to port 8080 on your host:
  `docker run -d -p 8080:80 --name my-nginx nginx:latest`
  - -d runs the container in detached mode (in the background).
  - -p 8080:80 maps port 80 inside the container to port 8080 on your host.
  - --name my-nginx assigns a name to the container for easier management.
  - nginx:latest specifies the image and tag to use.

- Check that it runs: `docker ps / docker ps -a`
- Open browser: `http://localhost:8080`
- View logs: `docker logs my-nginx`
- Access container: `docker exec -it my-nginx /bin/bash`
- Stop container: `docker stop my-nginx`
- Remove container: `docker rm my-nginx`

Virtualization

Using containers with Docker

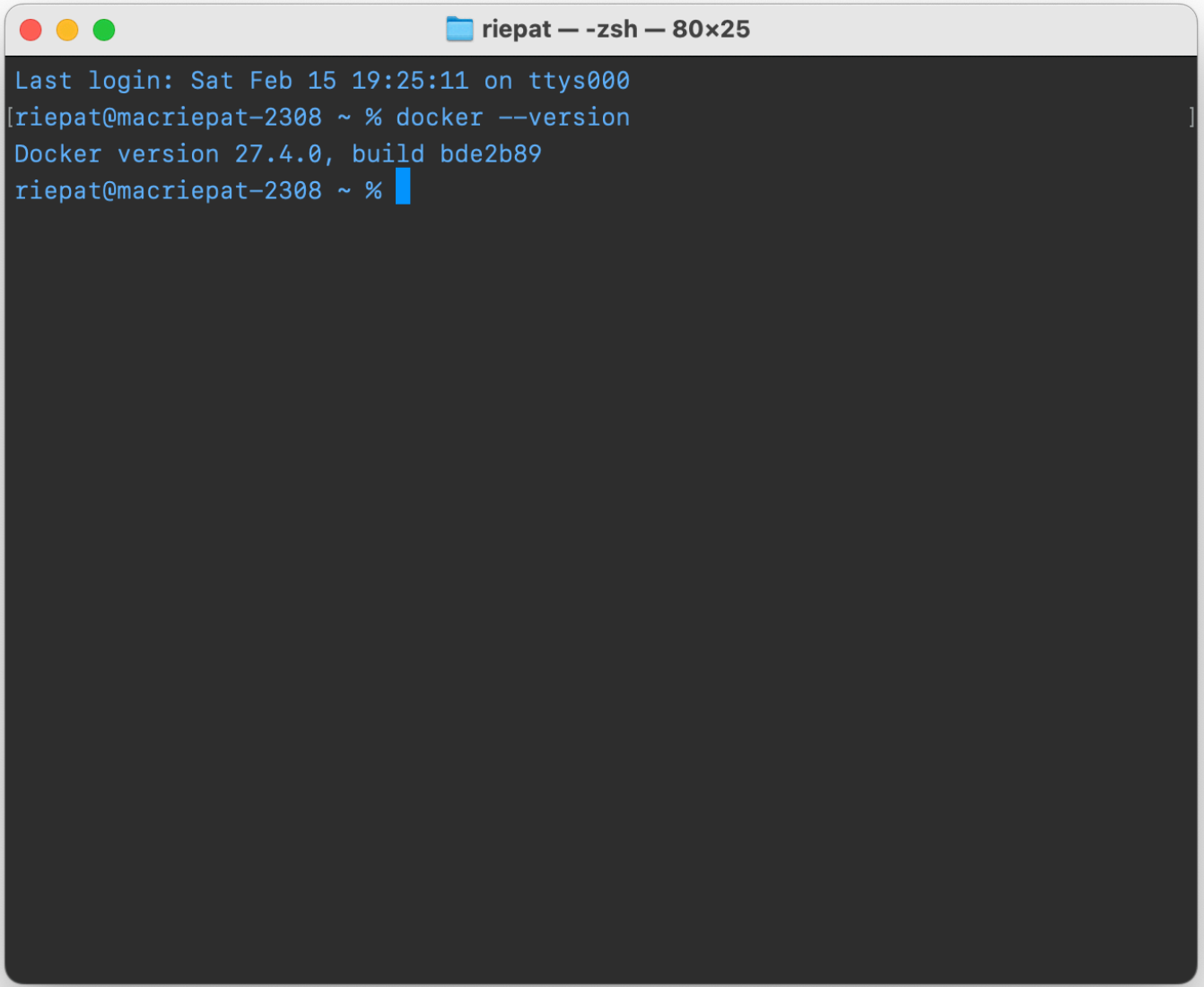
- ❑ Download and install Docker and macOS and Windows



- Install Docker in Linux

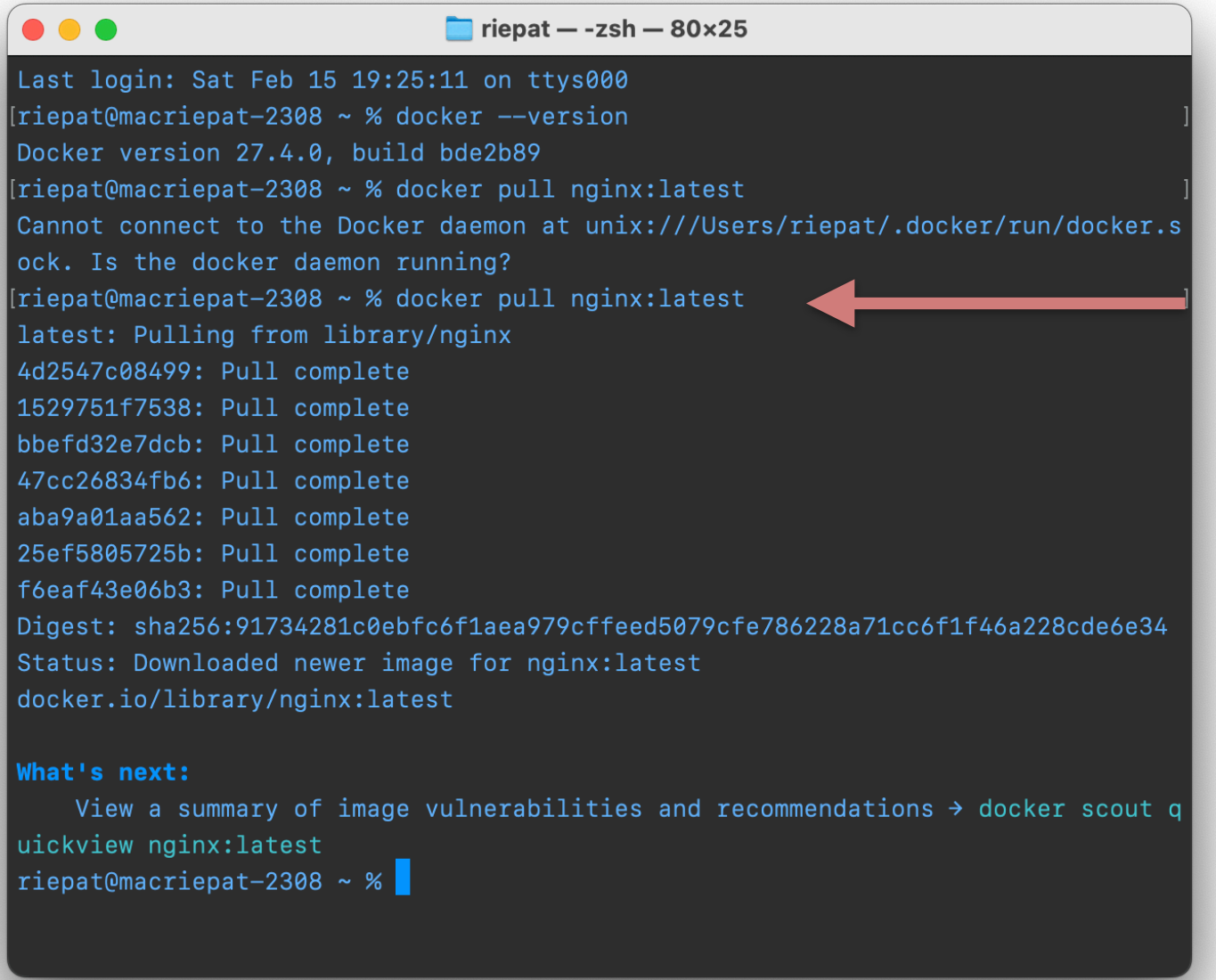
```
sudo apt-get update
sudo apt-get install -y docker.io
sudo systemctl start docker
sudo systemctl enable docker
```

❑ Check installed version



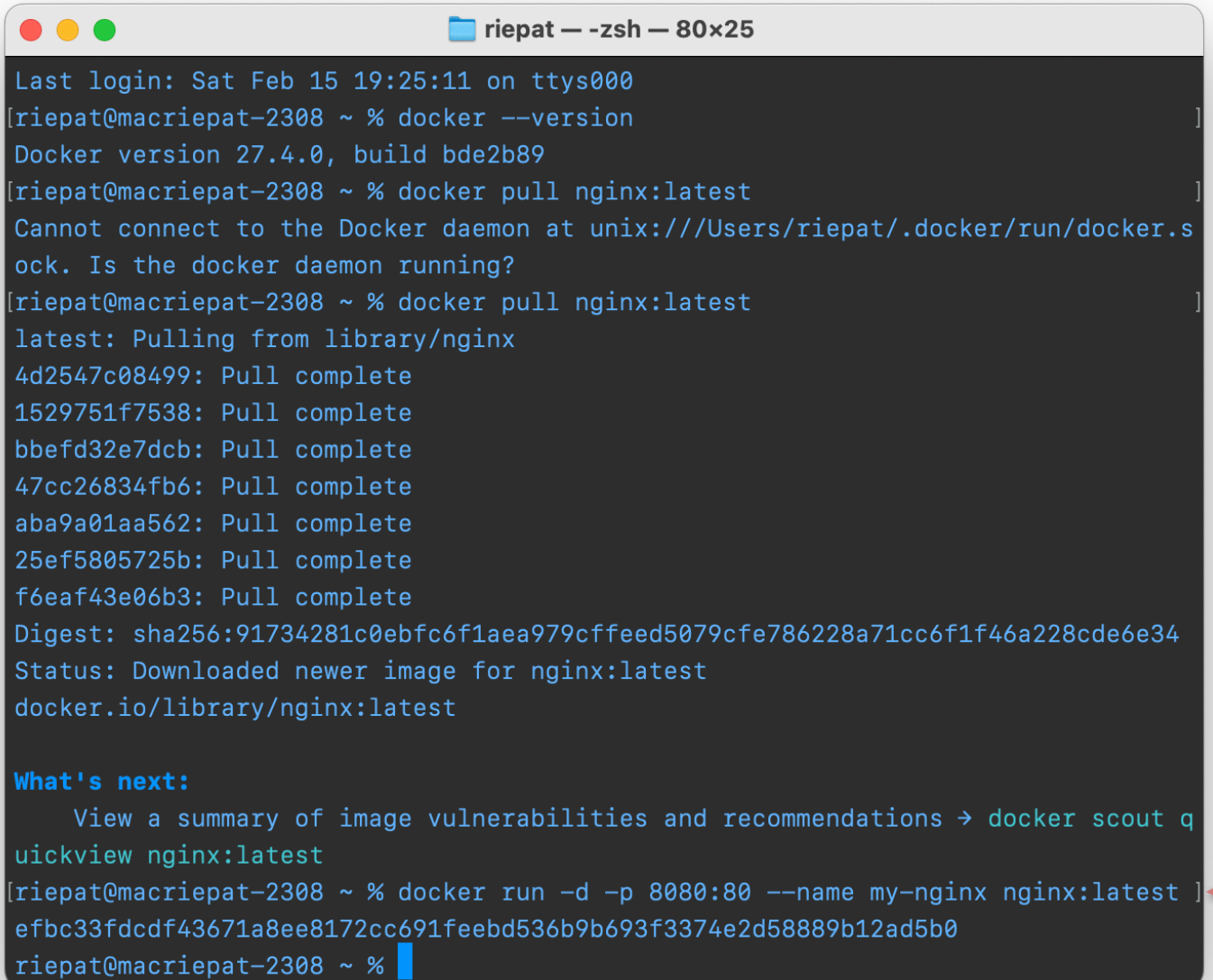

```
docker pull nginx:latest
```

- ❑ Pull an image



`docker pull nginx:latest`

- Create and run container from image

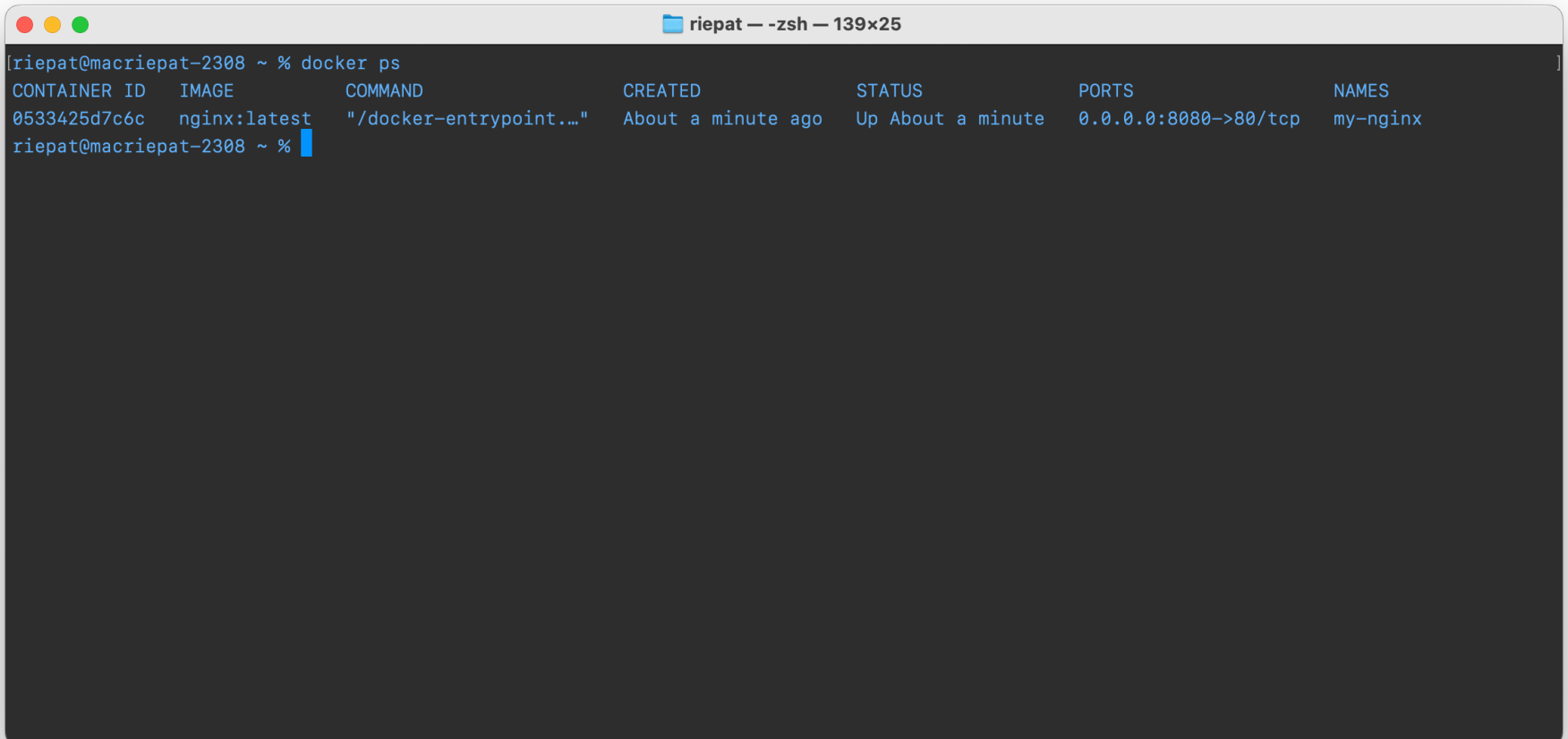



```
docker run -d -p 8080:80 --name my-nginx nginx:latest
```

Virtualization

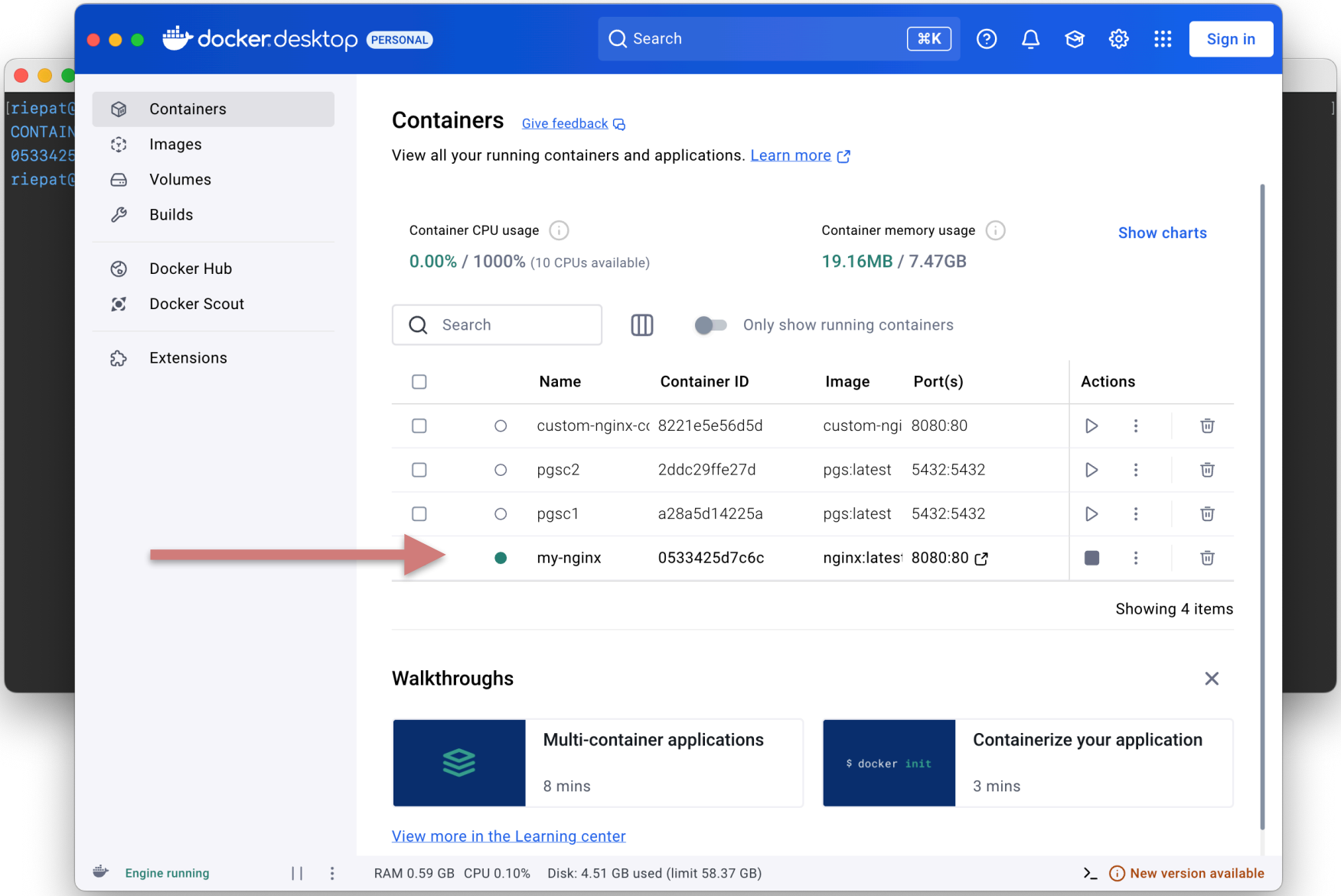
Using containers with Docker

❑ Optional

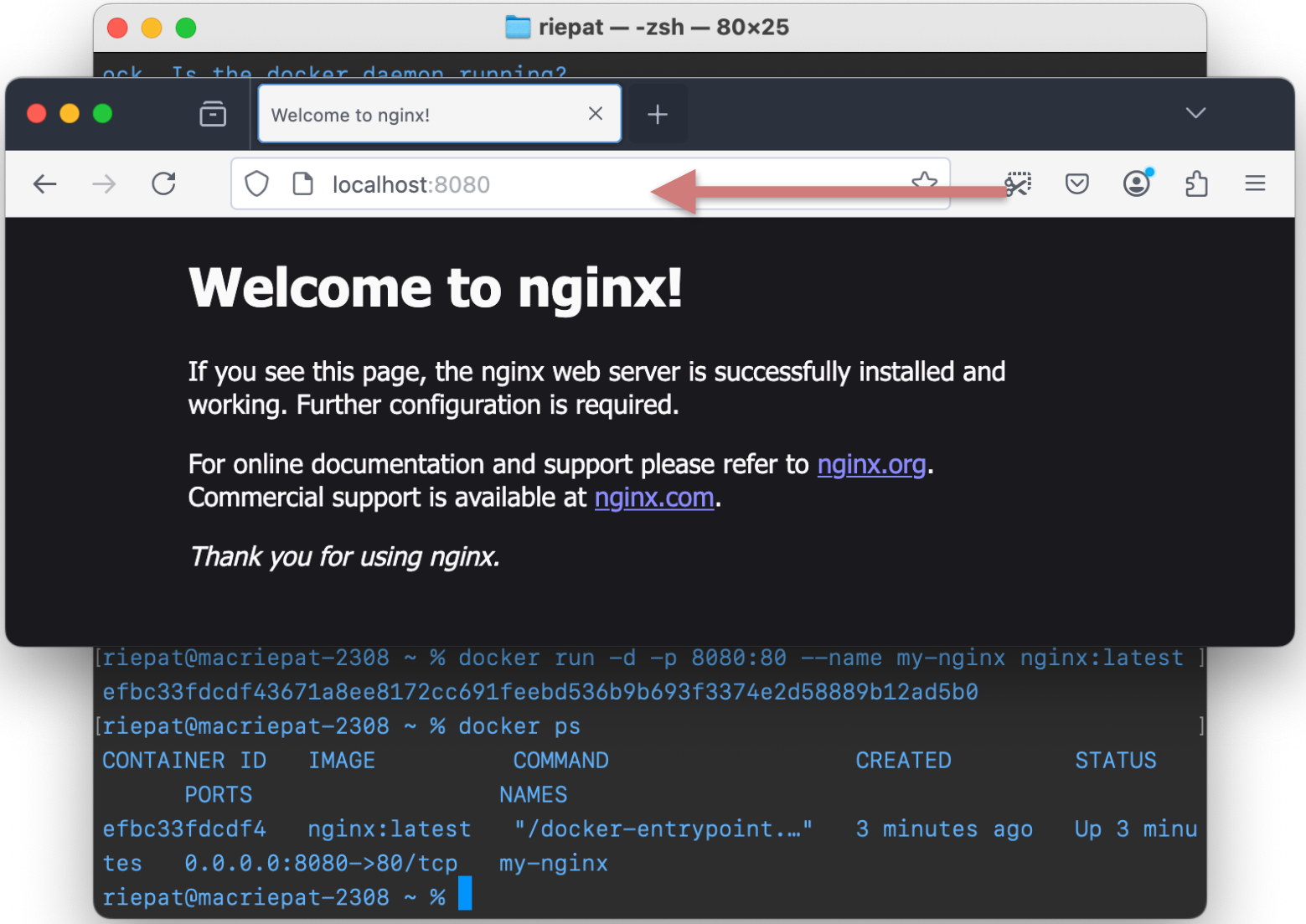


`docker ps`

❑ Alternative check in Docker Desktop



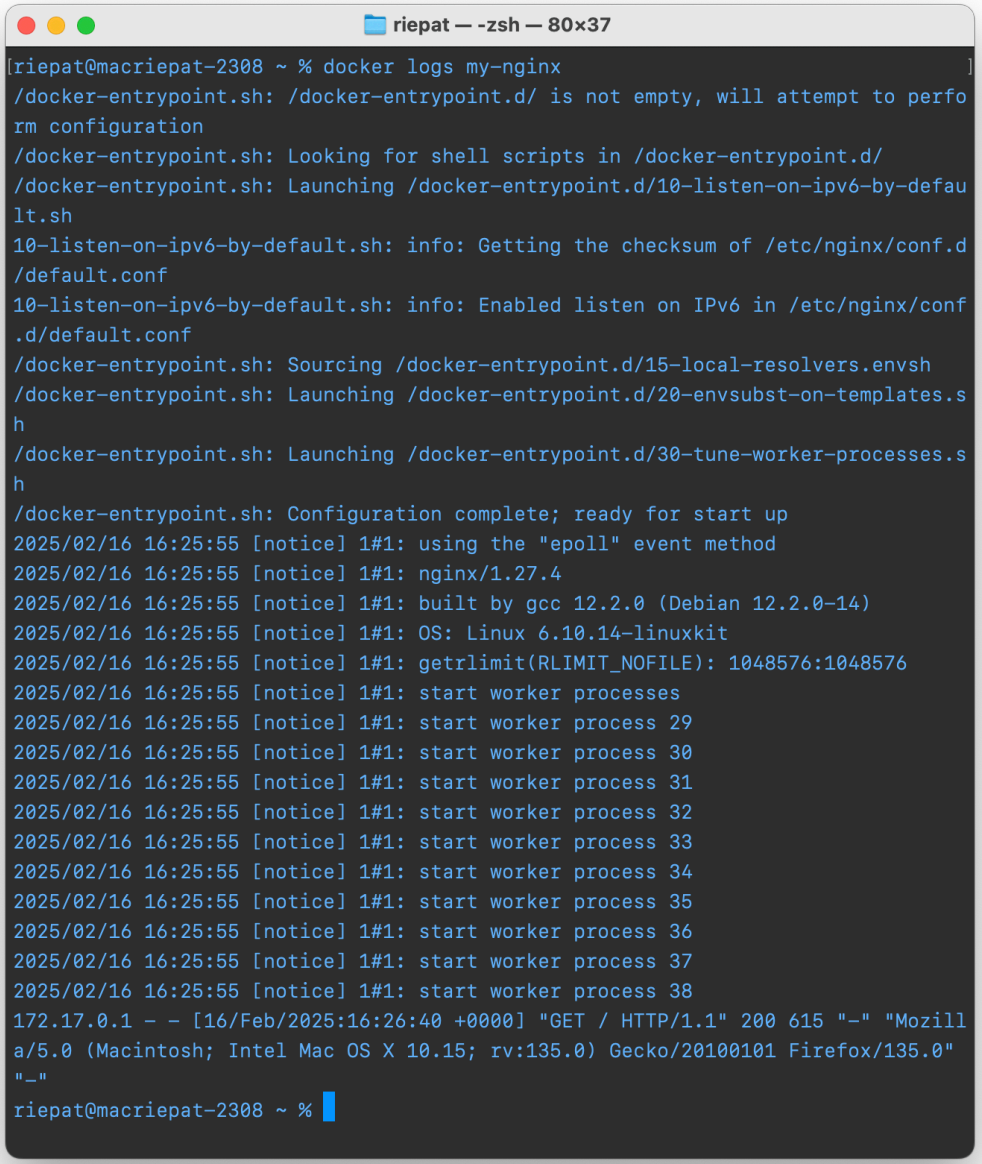
❑ Open browser



Virtualization

Using containers with Docker

- View logs

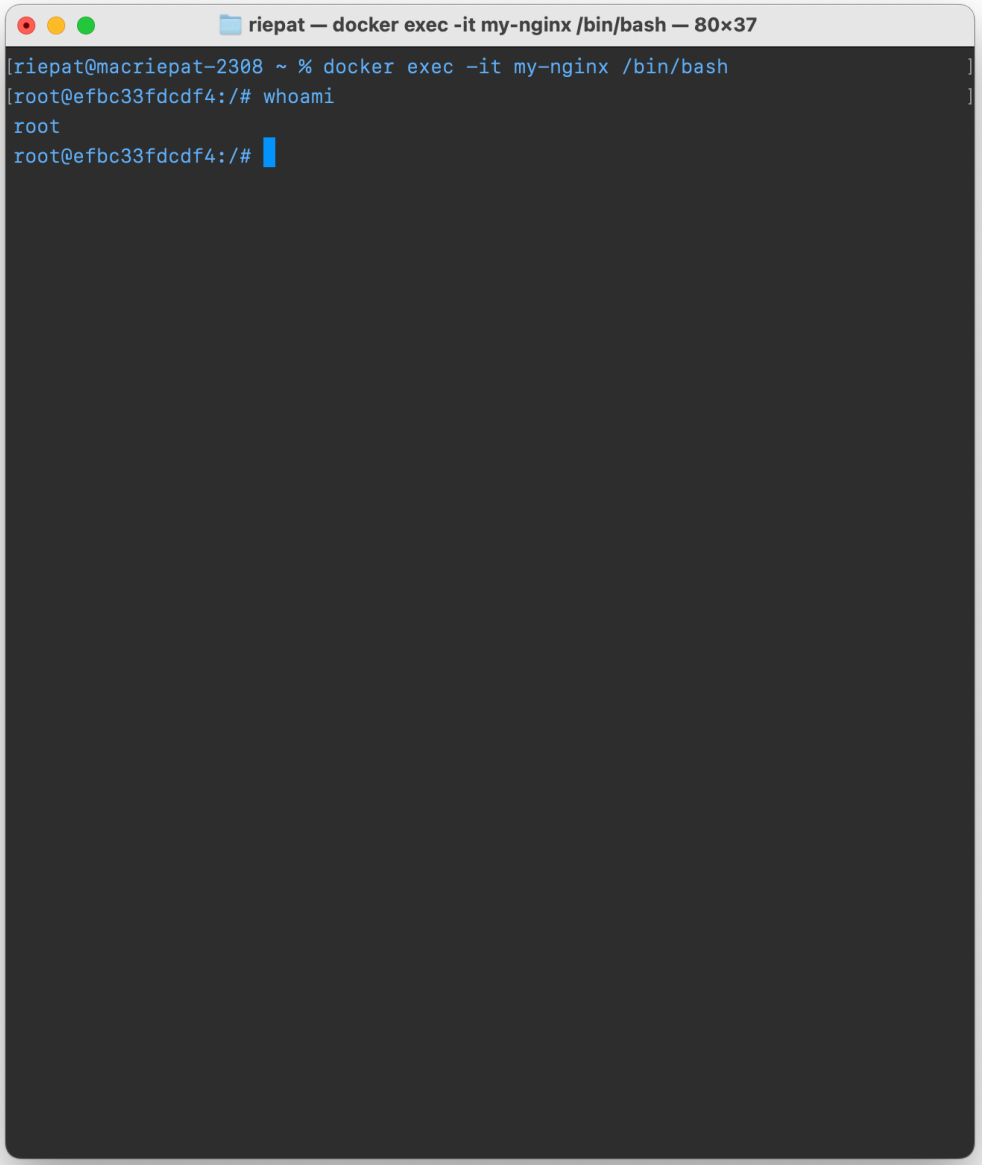


docker logs my-nginx

Virtualization

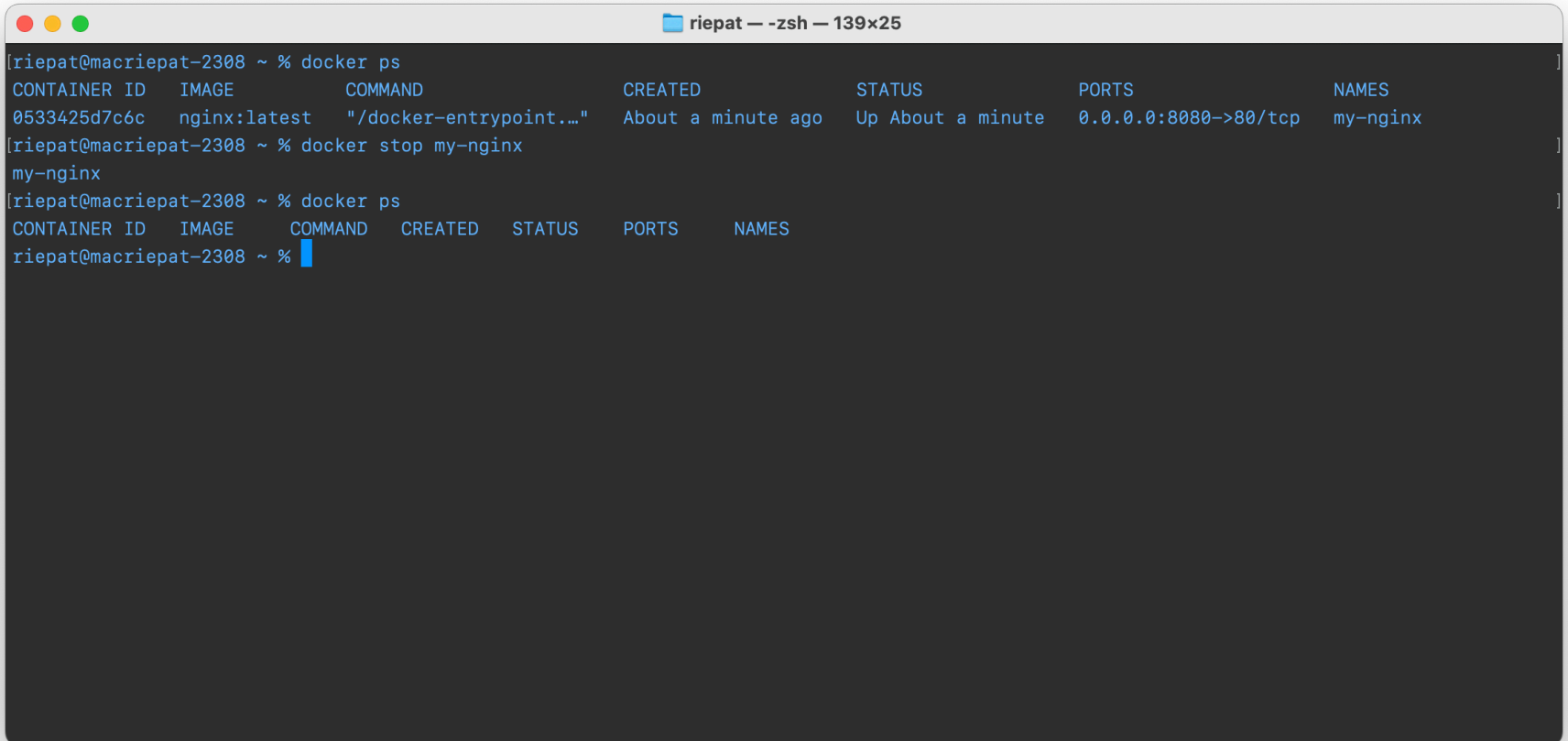
Using containers with Docker

- Access container




```
docker exec -it my-nginx /bin/bash
```

❑ Stop container:

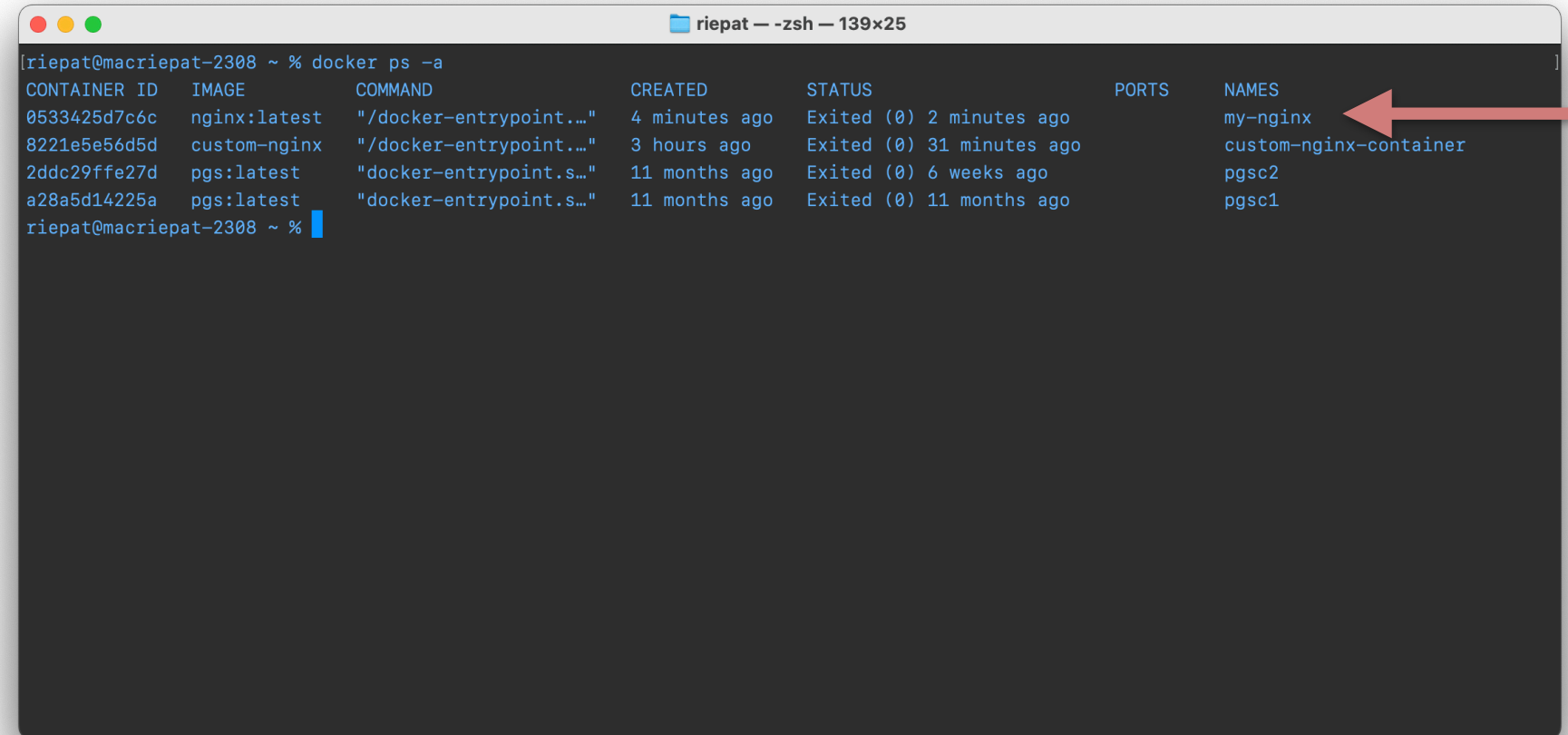


❑ `docker stop my-nginx`

Virtualization

Using containers with Docker

❑ Show stopped container:

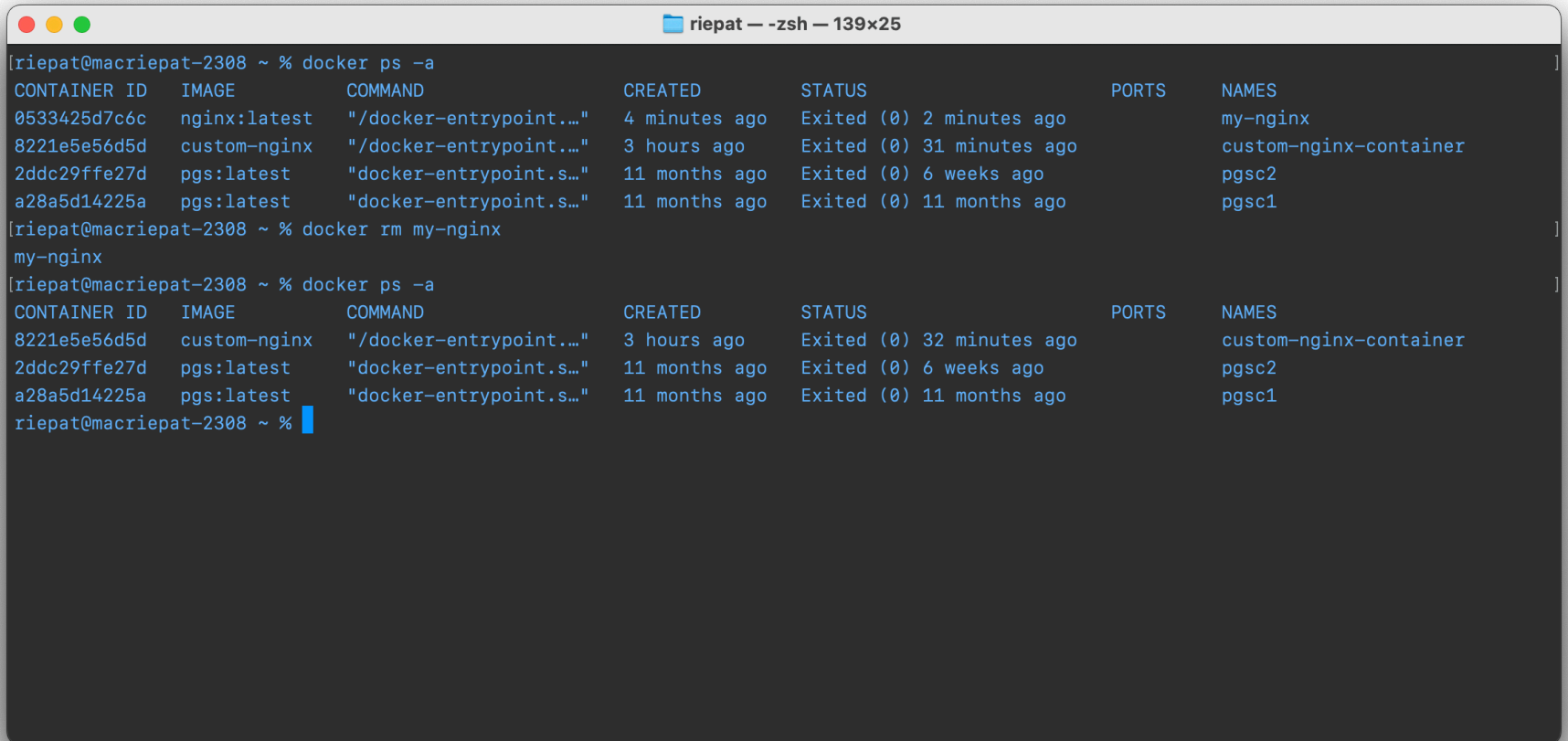


❑ `docker stop my-nginx`

Virtualization

Using containers with Docker

- Show stopped container:



- `docker rm my-nginx`

Virtualization

Using containers with Docker

- Container is gone, but the pulled image is still there

riepat — -zsh — 163×34

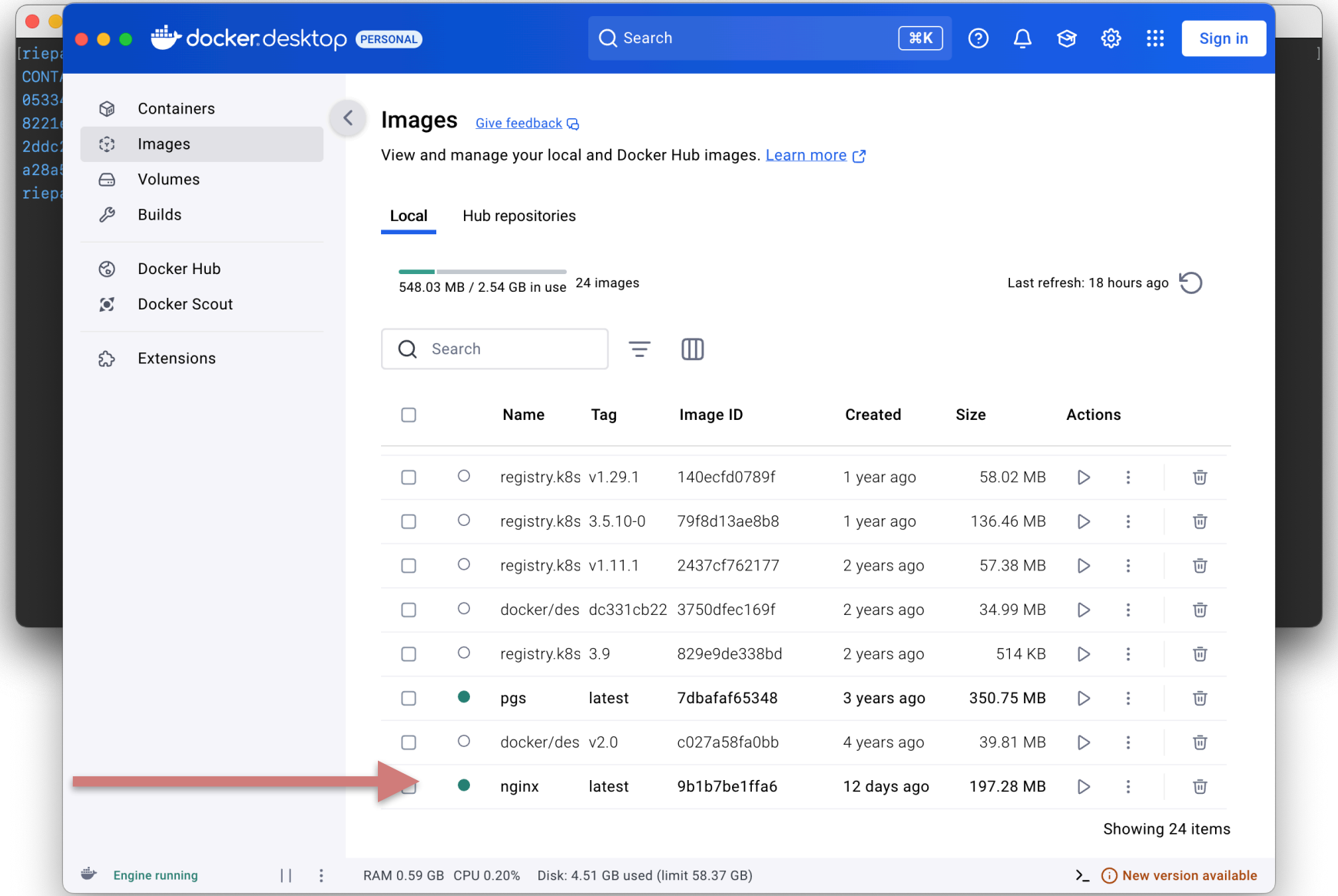
```
[riepat@macriepat-2308 ~ % docker rm my-nginx
my-nginx
[riepat@macriepat-2308 ~ % docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                     PORTS     NAMES
8221e5e56d5d   custom-nginx   "/docker-entrypoint.…"   3 hours ago     Exited (0) 32 minutes ago            custom-nginx-container
2ddc29ffe27d   pgs:latest     "docker-entrypoint.s…"   11 months ago   Exited (0) 6 weeks ago               pgsc2
a28a5d14225a   pgs:latest     "docker-entrypoint.…"    11 months ago   Exited (0) 11 months ago             pgsc1
[riepat@macriepat-2308 ~ % docker image ls
REPOSITORY                                TAG                                                                IMAGE ID       CREATED         SIZE
custom-nginx                              latest                                                             13db8e1d9f5a   3 hours ago     197MB
nginx                                     latest                                                             9b1b7be1ffa6   11 days ago     197MB
docker/desktop-kubernetes                 kubernetes-v1.30.5-cni-v1.4.0-critools-v1.29.0-cri-dockerd-v0.3.11-1-debian   a7b8c38c701e   4 months ago    419MB
registry.k8s.io/kube-apiserver            v1.30.5                                                            2bf7f63bc5e4   5 months ago    112MB
registry.k8s.io/kube-controller-manager   v1.30.5                                                            e1be44cf89df   5 months ago    107MB
registry.k8s.io/kube-scheduler            v1.30.5                                                            b6db73bf7694   5 months ago    60.5MB
registry.k8s.io/kube-proxy                v1.30.5                                                            57f247cd1b56   5 months ago    87.9MB
registry.k8s.io/coredns/coredns           v1.11.3                                                            2f6c962e7b83   6 months ago    60.2MB
jkpgcity                                  latest                                                             ecef03278336   11 months ago   140MB
stores-img2                               latest                                                             6e92c89749a5   11 months ago   140MB
stores-img                                latest                                                             d58b9f1af73d   11 months ago   140MB
registry.k8s.io/etcd                      3.5.12-0                                                           014faa467e29   12 months ago   139MB
alpine                                    latest                                                             ace17d5d883e   12 months ago   7.73MB
docker/desktop-kubernetes                 kubernetes-v1.29.1-cni-v1.4.0-critools-v1.29.0-cri-dockerd-v0.3.8-1-debian   e953cb83dd7e   13 months ago   422MB
registry.k8s.io/kube-apiserver            v1.29.1                                                            f3b81ff188c6   13 months ago   123MB
registry.k8s.io/kube-controller-manager   v1.29.1                                                            8715bb0e3bc2   13 months ago   118MB
registry.k8s.io/kube-proxy                v1.29.1                                                            b125ba1d1878   13 months ago   85.4MB
registry.k8s.io/kube-scheduler            v1.29.1                                                            140ecfd0789f   13 months ago   58MB
registry.k8s.io/etcd                      3.5.10-0                                                           79f8d13ae8b8   15 months ago   136MB
registry.k8s.io/coredns/coredns           v1.11.1                                                            2437cf762177   18 months ago   57.4MB
docker/desktop-vpnkit-controller          dc331cb22850be0cdd97c84a9cfecaf44a1afb6e                           3750dfec169f   21 months ago   35MB
registry.k8s.io/pause                     3.9                                                                829e9de338bd   2 years ago     514kB
pgs                                       latest                                                             7dbafaf65348   3 years ago     351MB
docker/desktop-storage-provisioner        v2.0                                                               c027a58fa0bb   3 years ago     39.8MB
riepat@macriepat-2308 ~ %
```

`docker image ls`

Virtualization

Using containers with Docker

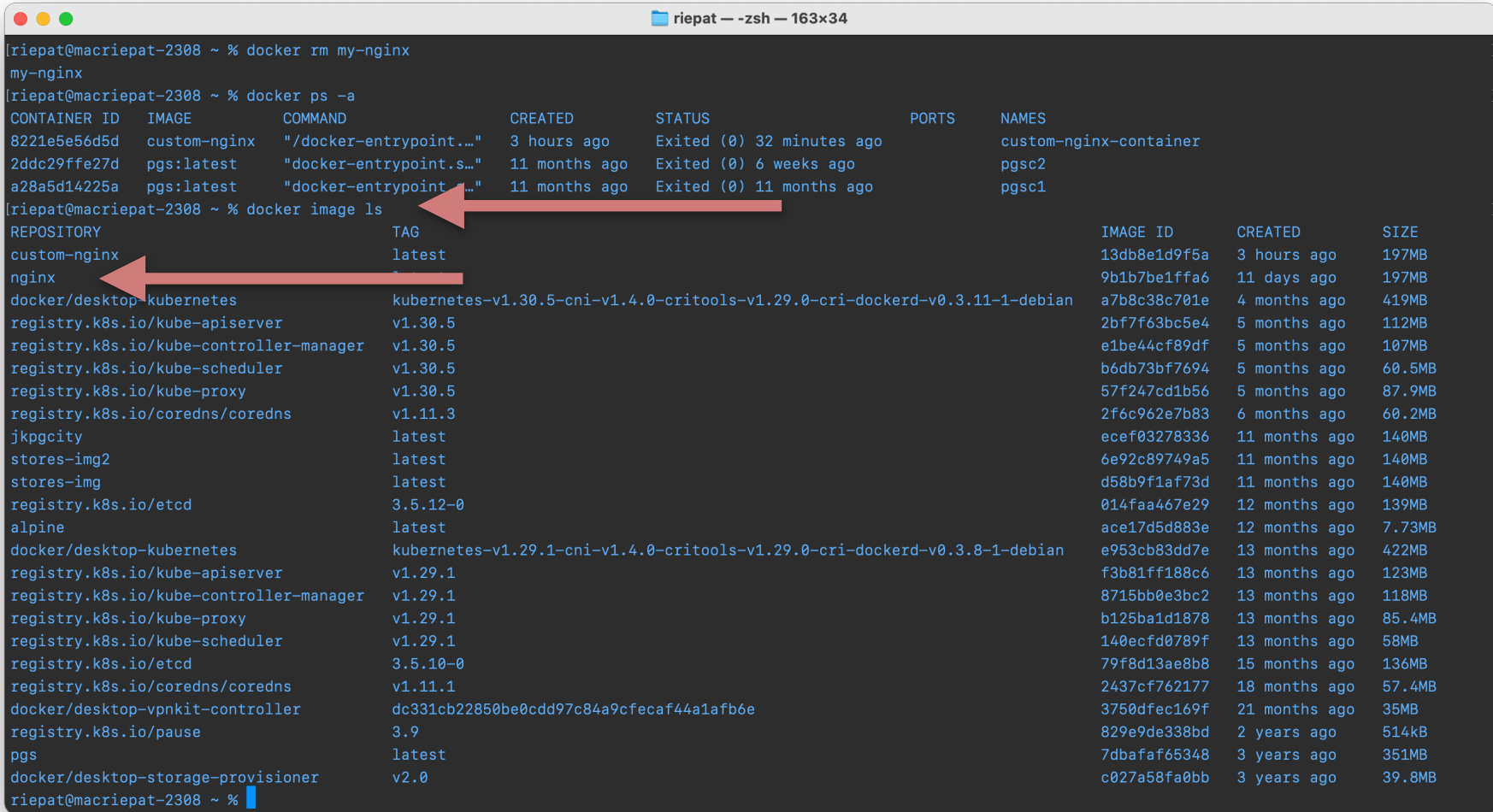
❑ Show stopped container:



Virtualization

Using containers with Docker

- Container is gone, but the pulled image is still there



`docker image ls`

- Image can be deleted by `docker image rm nginx`

Virtualization

Container setup with Dockerfile

Just Overview,
we go step-by-step

```
# Dockerfile
# Use the official Nginx image as the base image
FROM nginx:latest

# Copy the index.html file into Nginx's default directory
COPY index.html /usr/share/nginx/html/index.html

# Expose port 80 for the web server
EXPOSE 80
```

```
<!-- index.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>My Nginx Site</title>
  </head>
  <body>
    <h1>Hello, Nginx from Docker!</h1>
    <p>This page is served by a custom Docker image.</p>
  </body>
</html>
```

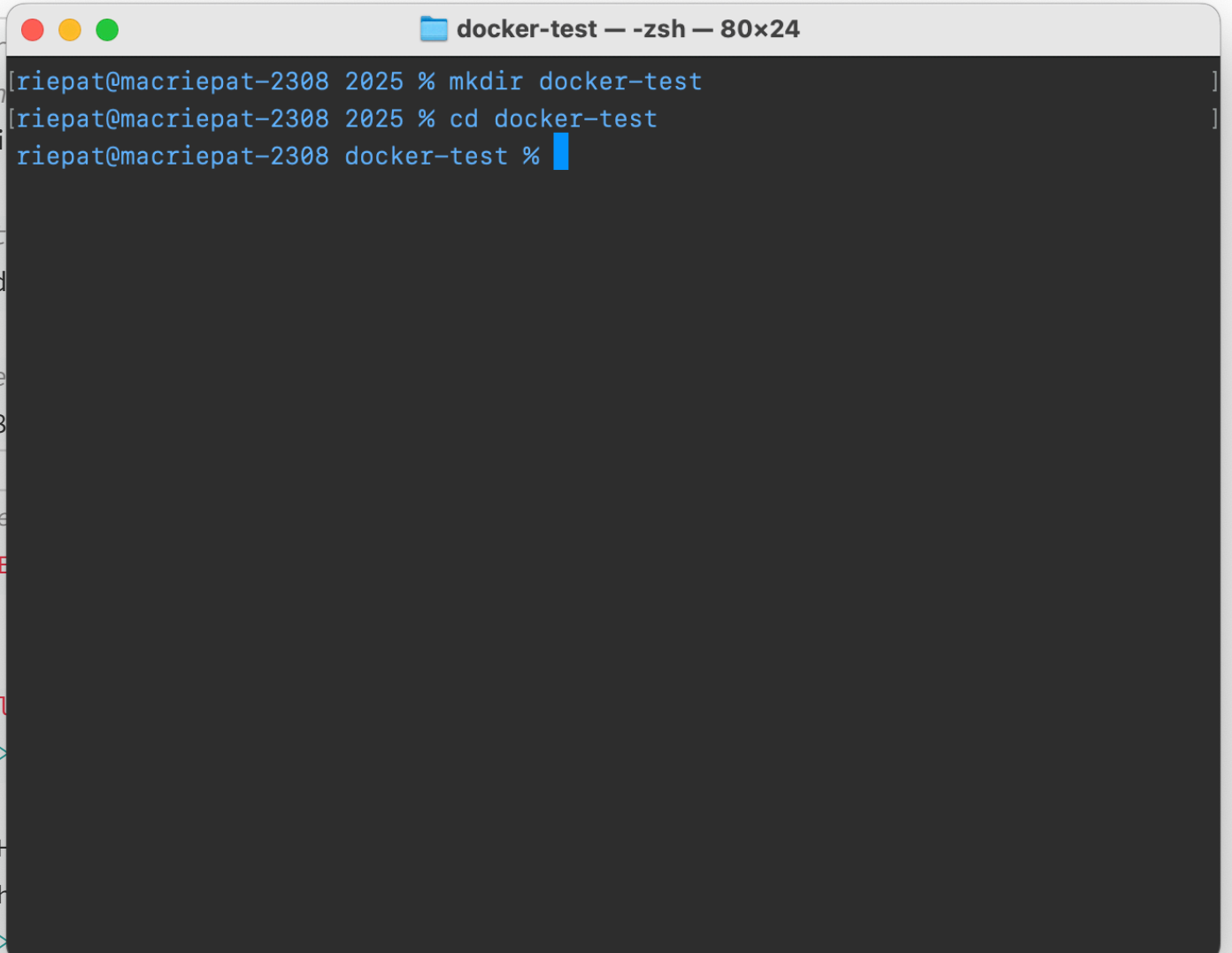
https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

Container setup with Dockerfile

```
# Docker
# Use th
FROM ngi

# Copy t
COPY ind

# Expose
EXPOSE 8
```

```
<!-- inde
<!DOCTYPE
<html>
  <head>
    <titl
  </head>
  <body>
    <h1>H
    <p>Th
  </body>
</html>
```



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

Container setup with Dockerfile

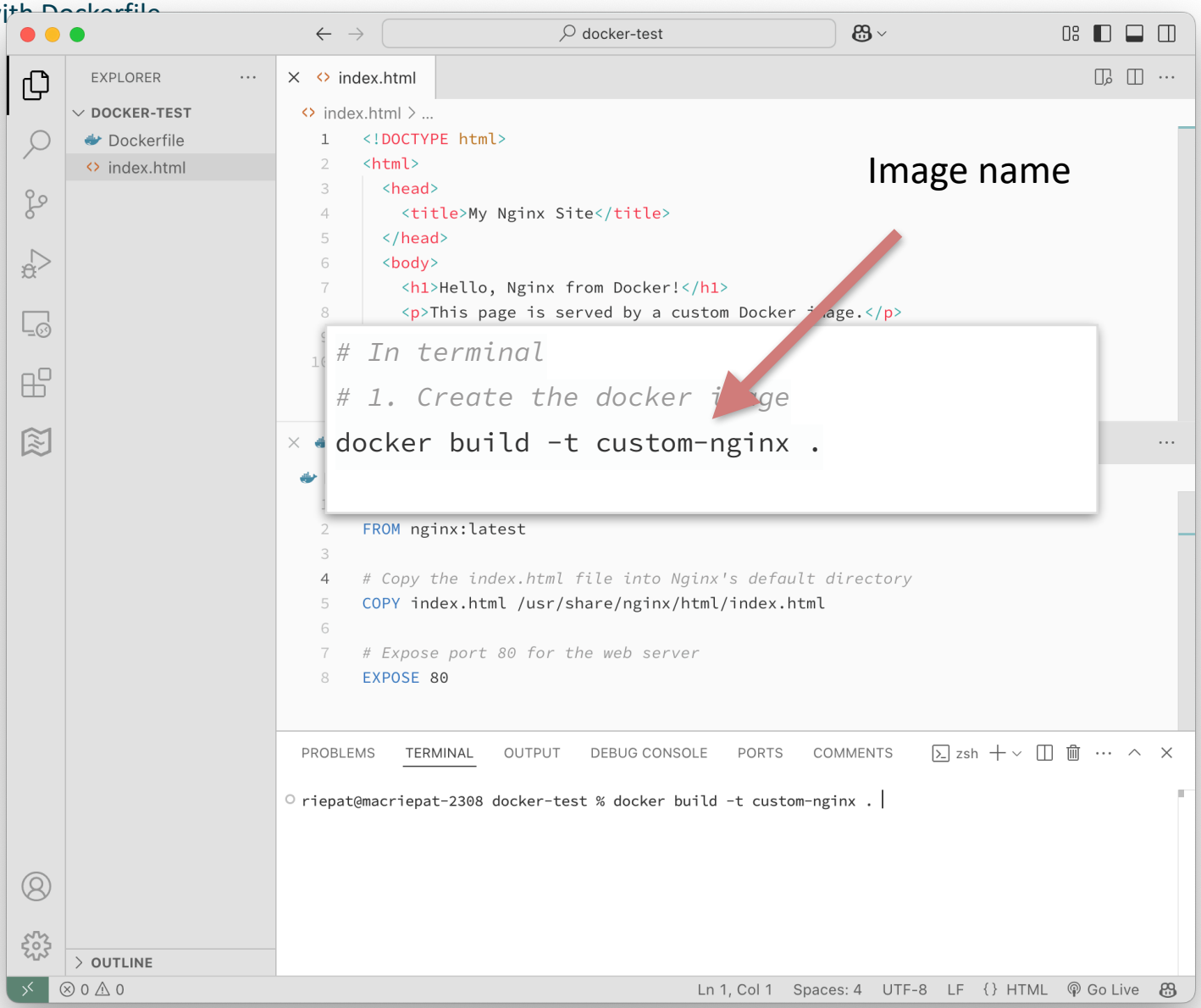
```
# Dockerfile
# Use the official Nginx image as the base image
FROM nginx:latest

# Copy the index.html file into Nginx's default directory
COPY index.html /usr/share/nginx/html/index.html

# Expose port 80 for the web server
EXPOSE 80
```

```
<!-- index.html -->
<!DOCTYPE html>
<html>
  <head>
    <title>My Nginx Site</title>
  </head>
  <body>
    <h1>Hello, Nginx from Docker!</h1>
    <p>This page is served by a custom Docker image.</p>
  </body>
</html>
```

https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

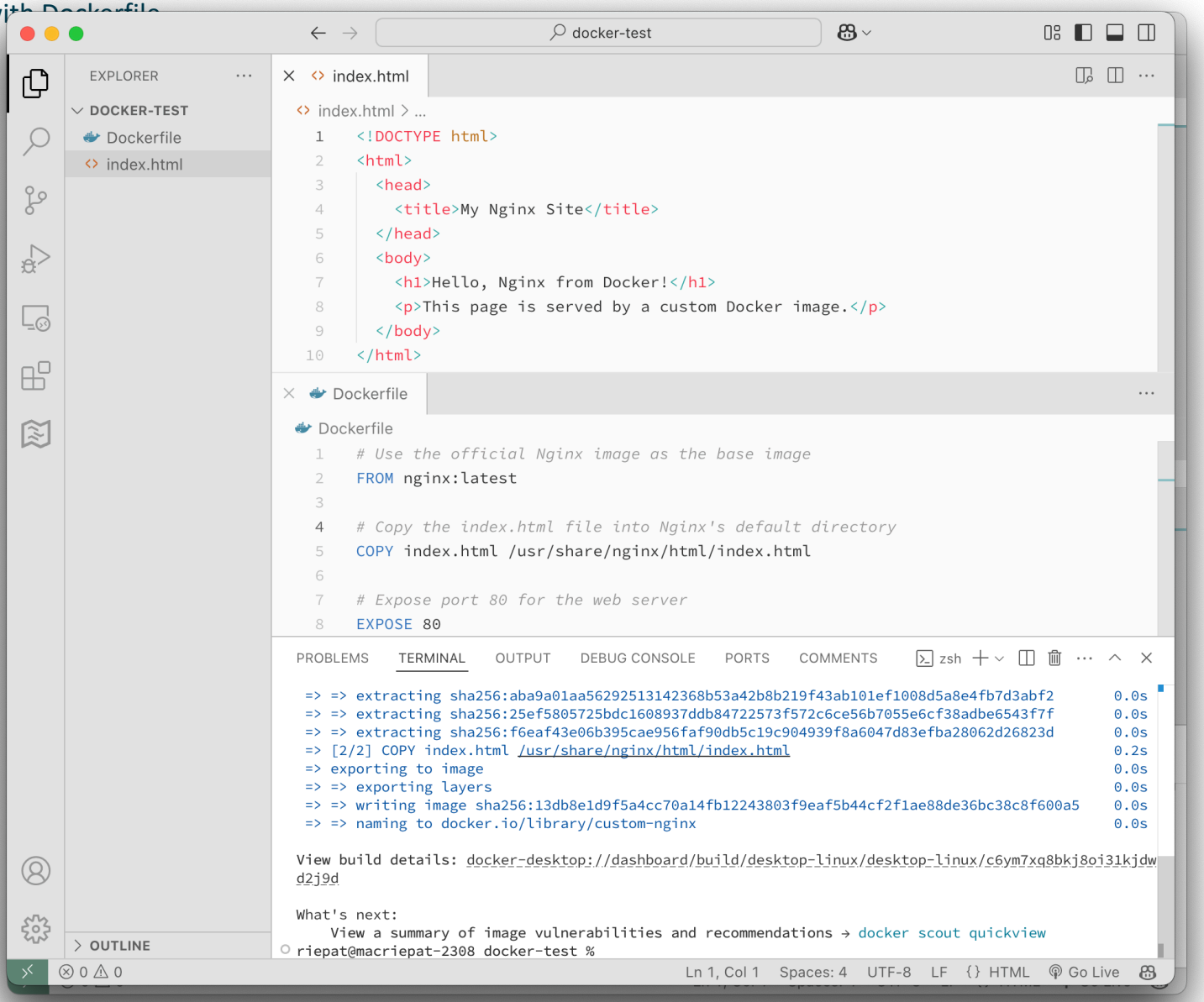
Container setup with Dockerfile



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

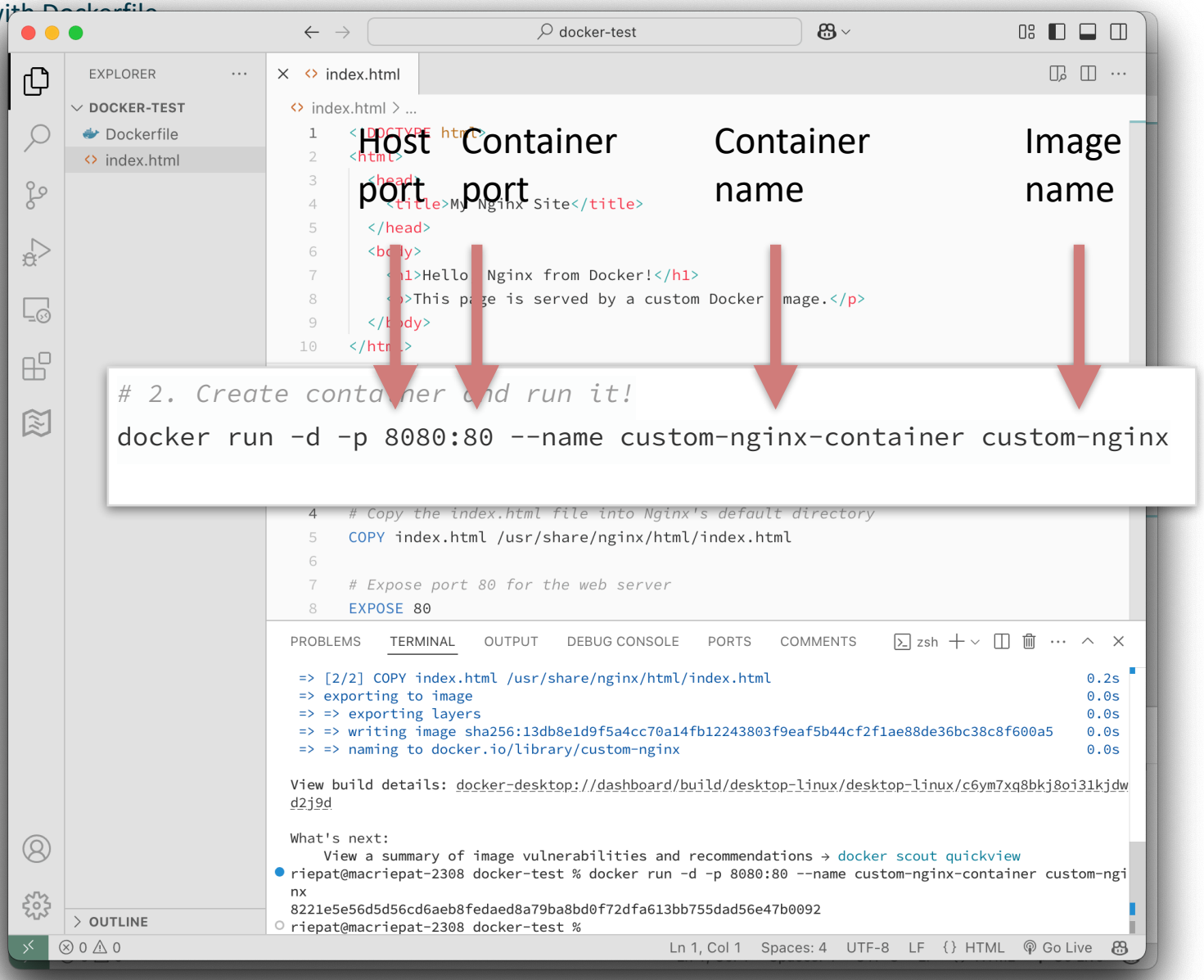
Container setup with Dockerfile



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

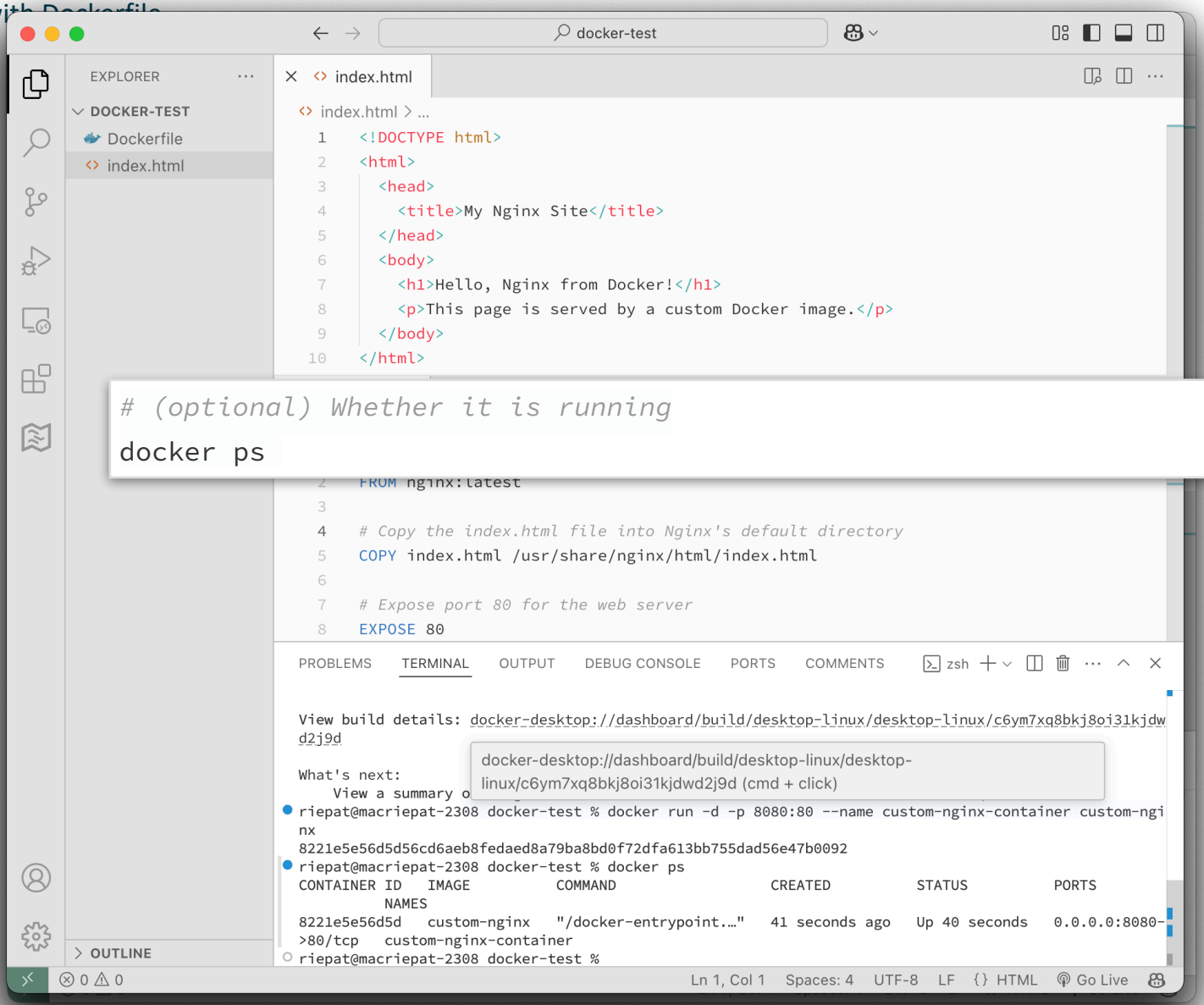
Container setup with Dockerfile



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

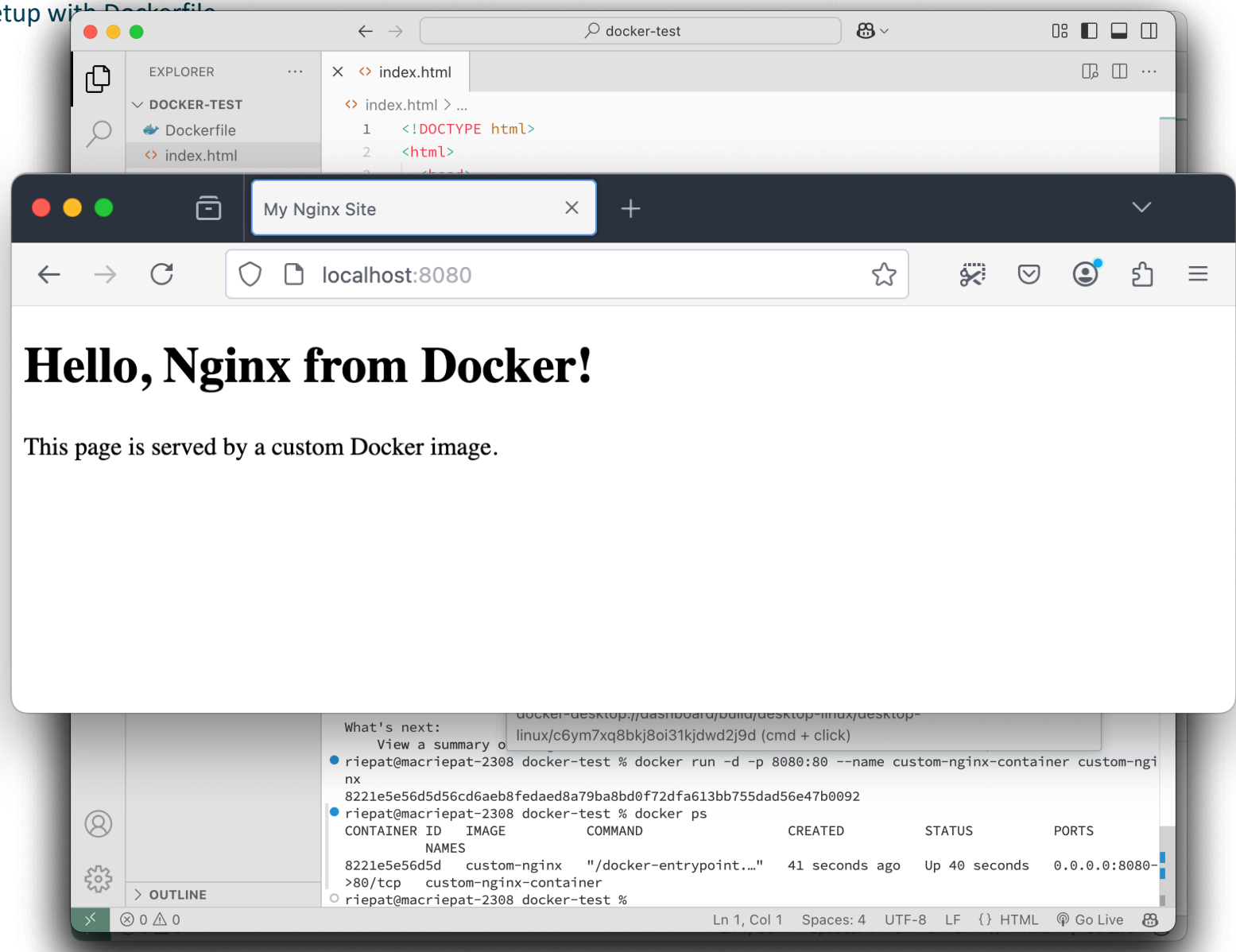
Container setup with Dockerfile



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

Container setup with Dockerfile



https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Virtualization

Container setup with Dockerfile

```
# In terminal

# Container can be removed with
docker rm container-name



# Check images
docker image ls

# Images can be removed with
docker image rm container-name
```

https://docs.docker.com/get-started/docker-concepts/building-images/writing-a-dockerfile/

Project

Hint

Hint for your project: Dockerfile for Node needs more setup!

```
# Pull a image with node already integrated
FROM node:20-alpine

# Define a directory to use inside the container
WORKDIR /usr/src/app

# Copy the content of our working directory into
# the container's working directory
COPY . .

# Install necessary requirements
RUN npm install

# Expose port 3000 for the web server
EXPOSE 3000

# Define the command for starting our web application
# inside the container (often app.js)
CMD [ "node", "server.js"]
```