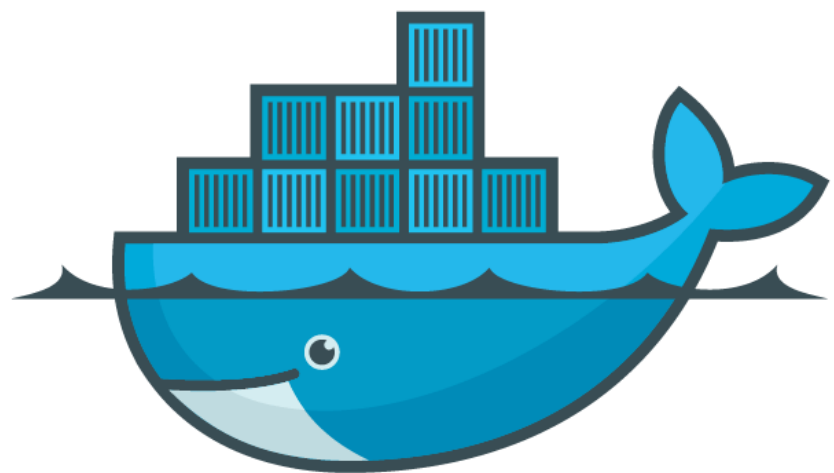


Chapter 10

Docker Container



docker

for beginners

Introduction to Docker

- Containerization technology to package and run applications in isolated environments.
- Lightweight alternative to virtual machines.
- Ensures consistent runtime environments across development, staging, and production.
- Simplifies orchestration for development, testing, and production environments.

Setting Up Docker

- Install Docker on your local PC via official documentation.
- Test your installation:
 - `docker run hello-world`
- Confirm “Hello from Docker” output for a successful setup

```
$ docker run hello-world
```

```
Hello from Docker.
```

```
This message shows that your installation appears to be working correctly.
```

```
...
```

Terminology

- Images: Blueprints of applications (read-only)
- Containers: Running instances of images (ephemeral)
- Docker Daemon: Background service managing containers & images
- Docker Client: Command-line tool (CLI) to interact with daemon
- Docker Hub: Cloud-based registry for images



Containers

Images

Volumes

Builds

Docker Scout

Extensions

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

Local

Hub repositories

206.29 MB / 2.26 GB in use 3 images

Last refresh: 2 hours ago ↻

Search



<input type="checkbox"/>	Name	Tag	Image ID	Created	Size	Actions
<input type="checkbox"/>	docker/welcome-to-docl	latest	eedaff45e3c7	1 year ago	29.46 MB	
<input type="checkbox"/>	hello-world	latest	5b3cc85e16e3	2 years ago	24.38 KB	
<input type="checkbox"/>	drz00l/static-site	latest	57f62a58e5b3	1 hour ago	278.38 MB	

Showing 3 items

Walkthroughs



Run Docker Hub images

Publish your image

Running Your First Container

- Example: docker pull busybox (fetch an image)
 - docker run busybox echo "hello from busybox"
- Inspecting images"
 - docker images
- Inspecting containers:
 - shows running containers
 - docker ps
 - shows all containers, even exited ones
 - docker ps -a

```
C:\Users\kokus>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
064a194a1e44	drz00l/static-site	"/docker-entrypoint..."	16 minutes ago	Exited (0) 10 minutes ago		flamboyant_chebyshe
v						
92b16188d92f	drz00l/static-site	"/docker-entrypoint..."	17 minutes ago	Exited (0) 12 minutes ago		static-site
1e8a88f21f31	zool/catnip	"python ./app.py"	2 hours ago	Exited (0) 2 hours ago		festive_satoshi
35a6ded3d2af	0b7b12f0bb96	"python ./app.py"	2 hours ago	Exited (1) 2 hours ago		hungry_goldstine
f10e35542678	217c37971621	"python ./app.py"	2 hours ago	Exited (1) 2 hours ago		strange_liskov
6bc2c77d19a9	217c37971621	"python ./app.py"	2 hours ago	Exited (1) 2 hours ago		angry_wing
5b761e9d4439	217c37971621	"python ./app.py"	2 hours ago	Exited (1) 2 hours ago		confident_lamarr
0e29389f90b6	hello-world	"/hello"	4 hours ago	Exited (0) 4 hours ago		festive_gates
49bead189d39	docker/welcome-to-docker:latest	"/docker-entrypoint..."	2 months ago	Exited (0) 23 hours ago		welcome-to-docker

Cleaning Up Containers and Images

- Removing stopped containers:
 - `docker rm <container_id>`
 - `docker rm $(docker ps -a -q -f status=exited)`

```
C:\Users\kokus>docker rm 36ef1081f96b 493fce40ad54
36ef1081f96b
493fce40ad54
```

- Removing images:
 - `docker rmi <image_id>`
- Pruning unused containers:
 - `docker container prune`

Running Web Applications in Containers

- Quickly run a static web server:
 - `docker run -d -P --name static-site drz00l/static-site`

```
C:\Users\kokus>docker run -d -P --name static-site drz00l/static-site
Unable to find image 'drz00l/static-site:latest' locally
latest: Pulling from drz00l/static-site
808baf0fcf92: Already exists
2bc7b46196db: Already exists
Digest: sha256:57f62a58e5b3707c71f7c0976f91fd3741f8cbc61b4388d716cf6139d9d39734
Status: Downloaded newer image for drz00l/static-site:latest
7169a9f8ee3eaf0a36ae4e3b50bc4376f74ec12e5898b8258a44a49f2f961513
```

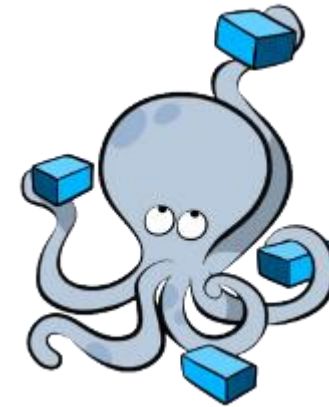
- Map container port to host:
 - `docker run -p 5002:80 drz00l/static-site`
- Access via `http://localhost:<port_number>`
 - `docker port static-site`
- To Stop
 - `docker stop static-site`

Building Our Own Docker Images

- Example (Flask app)
 - `$ git clone https://github.com/dr-zool/docker-curriculum`
 - `$ cd docker-curriculum/flask-app`
- Use a Dockerfile to automate image creation
- Build & run:
 - `docker build -t drz00l/cat .`
 - `docker run -p 5002:5000 drz00l/cat`

Congratulations! You have successfully created your first docker image.

Pushing Images to Docker Hub



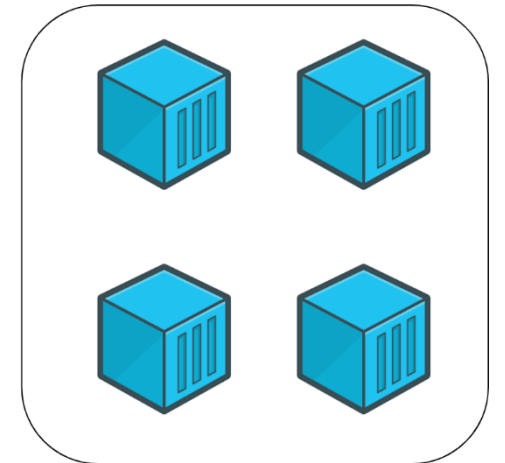
- Create a Docker Hub account
- Login:
 - `docker login`
- Push image:
 - `docker push username/cat`
- Share with others:
 - `docker run -p 8888:5000 username/cat`

Multi-Container Applications

- Real-world apps often need multiple services
 - Web + Database/search
 - AI Model Classification + Image Preprocessing + Machine Vision
- Docker approach:
 - One container per service
 - Use networks for communication
- Example
 - Run Elasticsearch & Flask app in same network



Docker



Docker-Compose

Docker Compose for Multi-Container Apps

- **docker-compose.yml** defines services, networks, volumes
- Run **docker-compose up** to bring entire environment online
- Example **docker-compose.yml**:
 - Defines es and web services
 - Automatically creates a dedicated network
 - Simplifies startup, teardown, scaling