#03 practical class

Docker build & compose

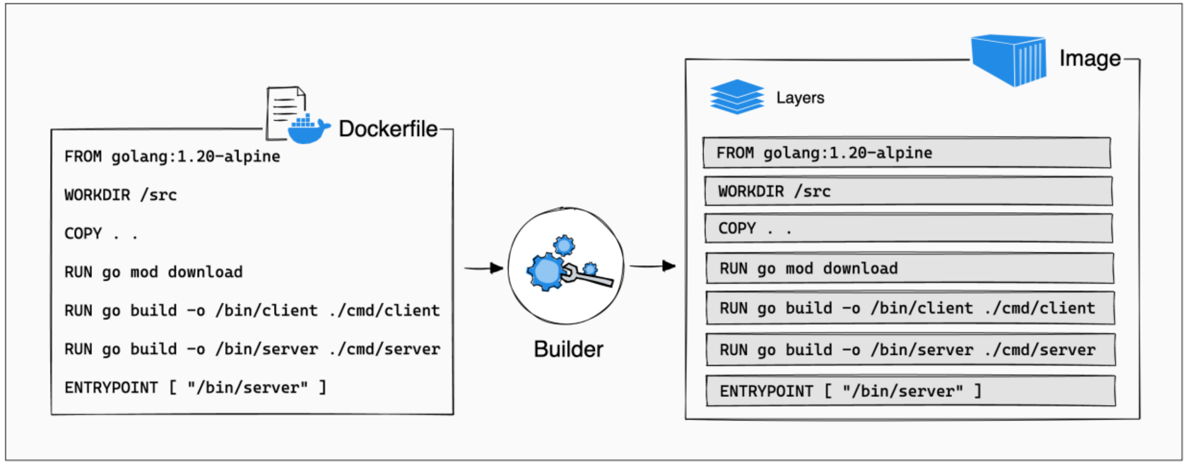
COMPUTING SYSTEMS AND INFRASTRUCTURES

(SISTEMAS E INFRAESTRUTURAS DE COMPUTAÇÃO)

Overview

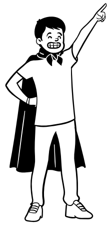
- Docker build
- Docker compose

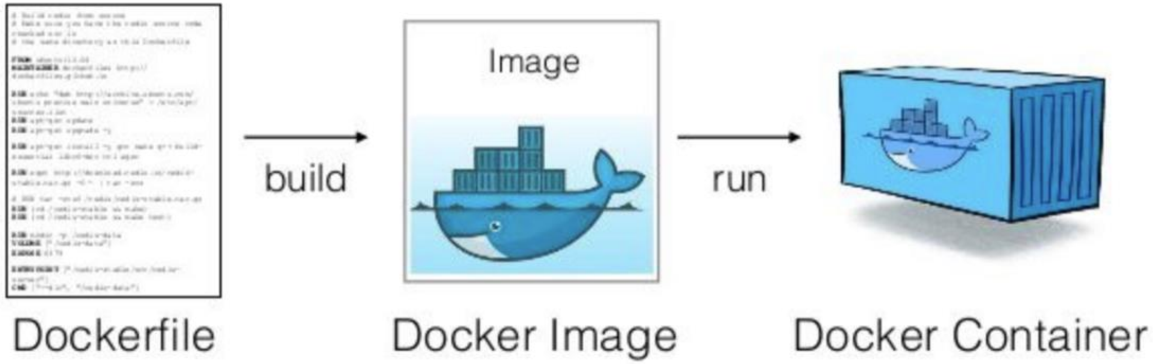
Docker build



```
FROM golang:1.20-alpine
WORKDIR /src
COPY . .
RUN go mod download
RUN go build -o /bin/client ./cmd/client
RUN go build -o /bin/server ./cmd/server
ENTRYPOINT [ "/bin/server" ]
```

- Build is a key part of your software development life cycle allowing you to package and bundle your code and ship it anywhere





Dockerfile example

- Objective: create image to run the app.py program

- Process

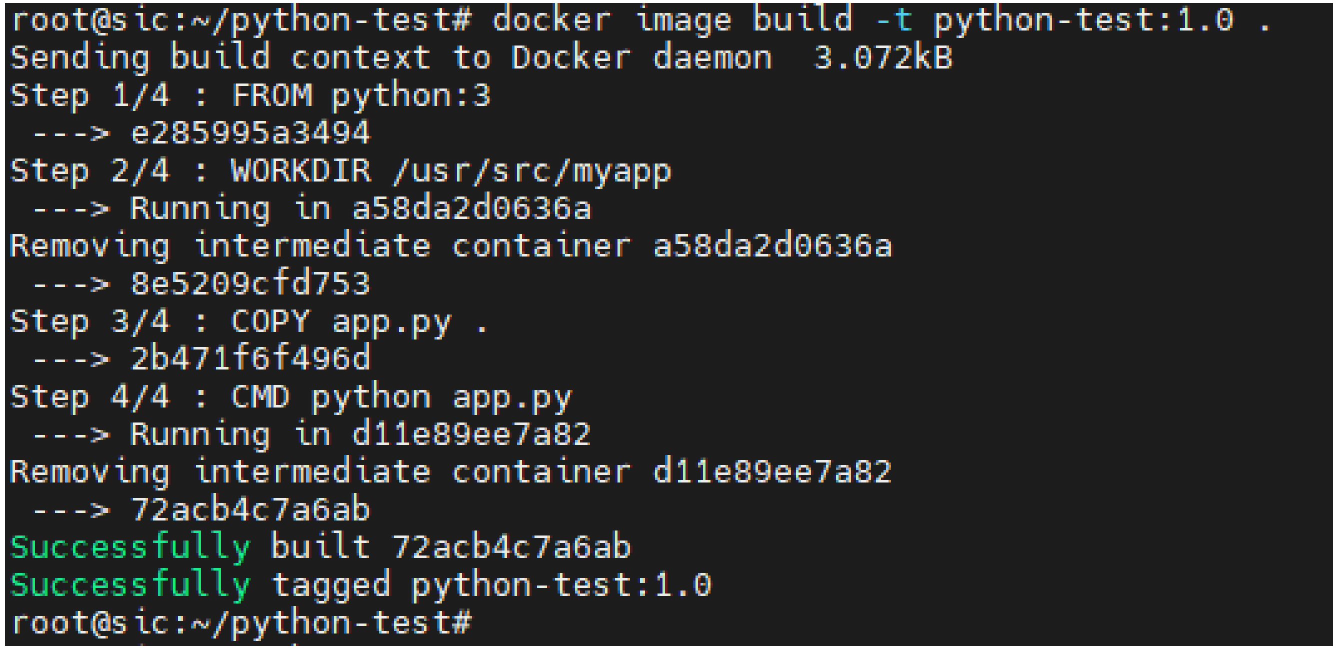
1. Create a Dockerfile (Pay attention to the syntax)
2. Use an official Python base image
3. Copy the app.py file from the VM to the container image
4. Run the python app.py program inside the container

Create Dockerfile

```
# Use an official Python base image
FROM python:3

# Sets the working directory for your app. This will influence the following COPY and CMD commands
WORKDIR /usr/src/myapp

# Copy the required files from the VM to the container (docker) image
COPY app.py .

# Run app.py inside the image filesystem
CMD python app.py
```

Build image

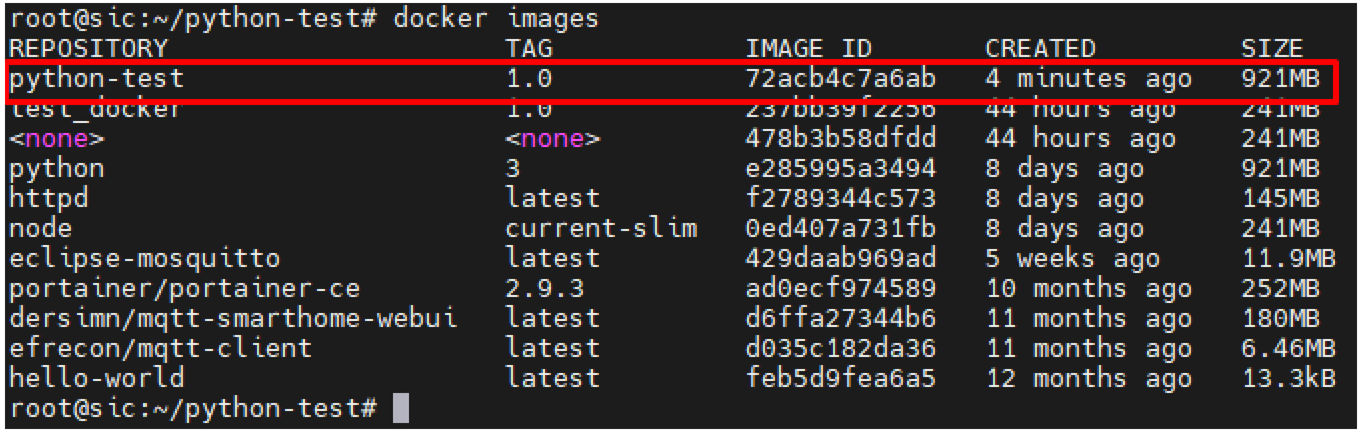
- docker image build -t python-test:1.0 .



```
root@sic:~/python-test# docker image build -t python-test:1.0 .
Sending build context to Docker daemon  3.072kB
Step 1/4 : FROM python:3
 ---> e285995a3494
Step 2/4 : WORKDIR /usr/src/myapp
 ---> Running in a58da2d0636a
Removing intermediate container a58da2d0636a
 ---> 8e5209cfd753
Step 3/4 : COPY app.py .
 ---> 2b471f6f496d
Step 4/4 : CMD python app.py
 ---> Running in d11e89ee7a82
Removing intermediate container d11e89ee7a82
 ---> 72acb4c7a6ab
Successfully built 72acb4c7a6ab
Successfully tagged python-test:1.0
root@sic:~/python-test#
```

Check created image

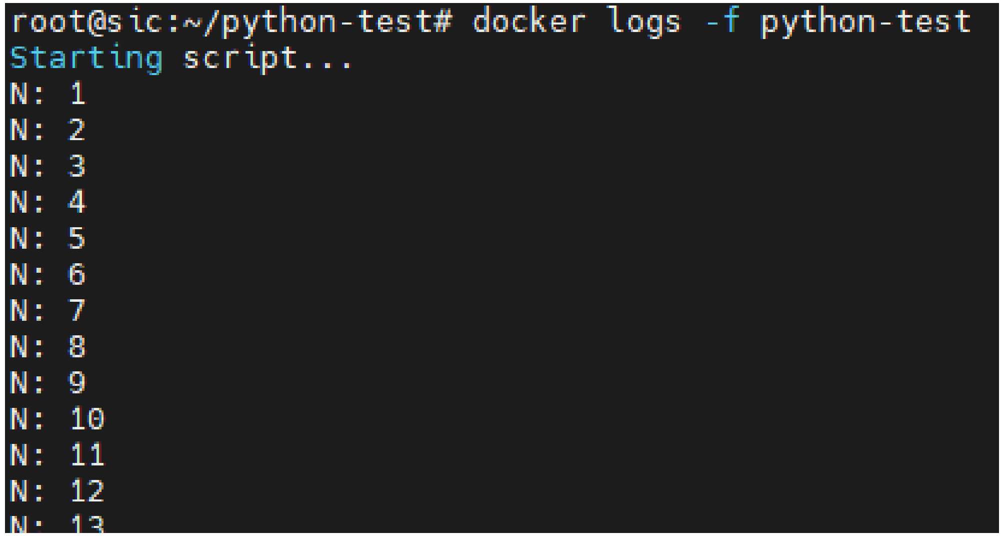
- docker images



```
root@sic:~/python-test# docker images
REPOSITORY                      TAG            IMAGE ID       CREATED          SIZE
python-test                     1.0            72acb4c7a6ab   4 minutes ago    921MB
test_docker                     1.0            237bb39f2256   44 hours ago     241MB
<none>                          <none>         478b3b58dfdd   44 hours ago     241MB
python                          3              e285995a3494   8 days ago       921MB
httpd                           latest         f2789344c573   8 days ago       145MB
node                            current-slim   0ed407a731fb   8 days ago       241MB
eclipse-mosquitto               latest         429daab969ad   5 weeks ago      11.9MB
portainer/portainer-ce          2.9.3          ad0ecf974589   10 months ago    252MB
dersimn/mqtt-smarthome-webui    latest         d6ffa27344b6   11 months ago    180MB
efrecon/mqtt-client             latest         d035c182da36   11 months ago    6.46MB
hello-world                     latest         feb5d9fea6a5   12 months ago    13.3kB
root@sic:~/python-test#
```

Run the new image

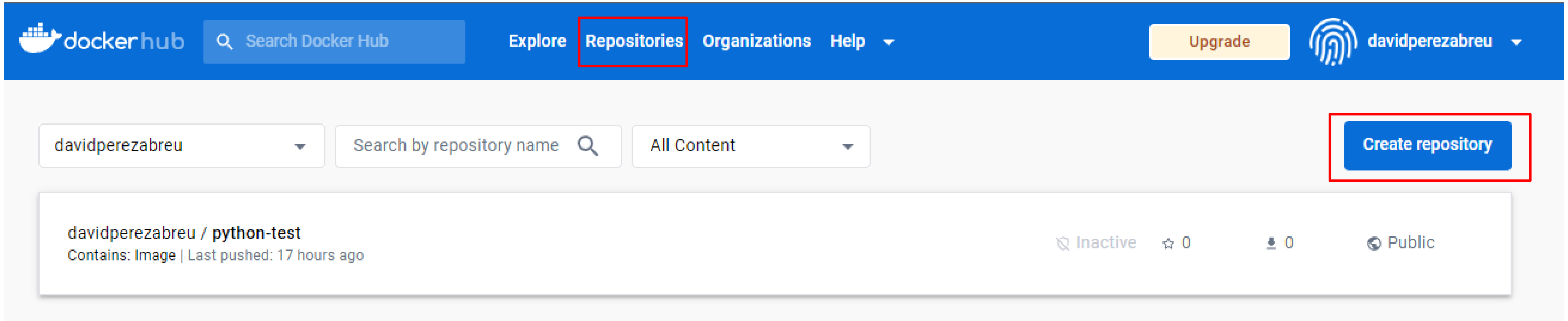
- docker run -dit --name python-test python-test:1.0
- docker logs -f python-test



```
root@sic:~/python-test# docker logs -f python-test
Starting script...
N:  1
N:  2
N:  3
N:  4
N:  5
N:  6
N:  7
N:  8
N:  9
N:  10
N:  11
N:  12
N:  13
```

Pushing your image to Docker Hub (1/3)

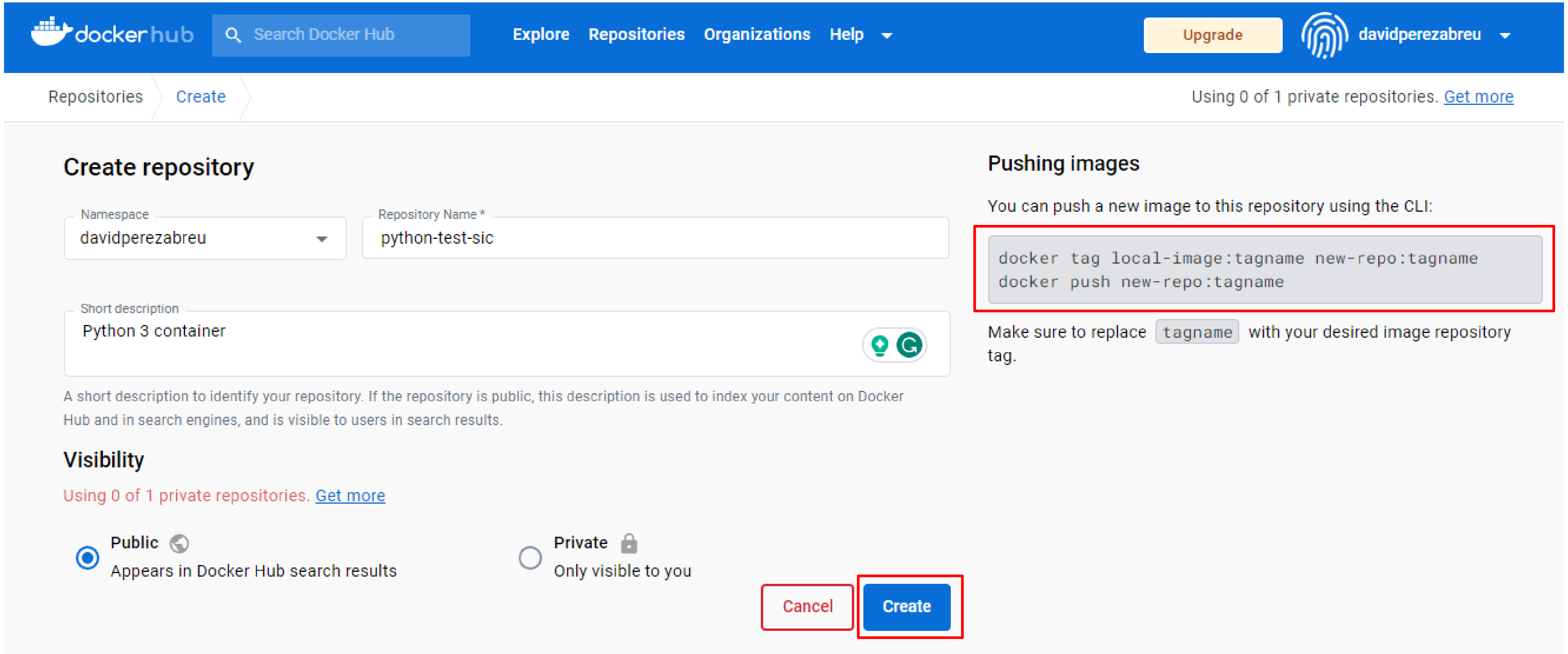
- https://hub.docker.com
  - Create an account or log in
- Create a repository



davidperezabreu / python-test
Contains: Image | Last pushed: 17 hours ago

Inactive ☆ 0 ⬇ 0 Public

Pushing your image to Docker Hub (2/3)



Using 0 of 1 private repositories. Get more

Create repository

Namespace: davidperezabreu

Repository Name *: python-test-sic

Short description: Python 3 container

A short description to identify your repository. If the repository is public, this description is used to index your content on Docker Hub and in search engines, and is visible to users in search results.

Visibility

Using 0 of 1 private repositories. Get more

● Public
Appears in Docker Hub search results

○ Private
Only visible to you

Pushing images

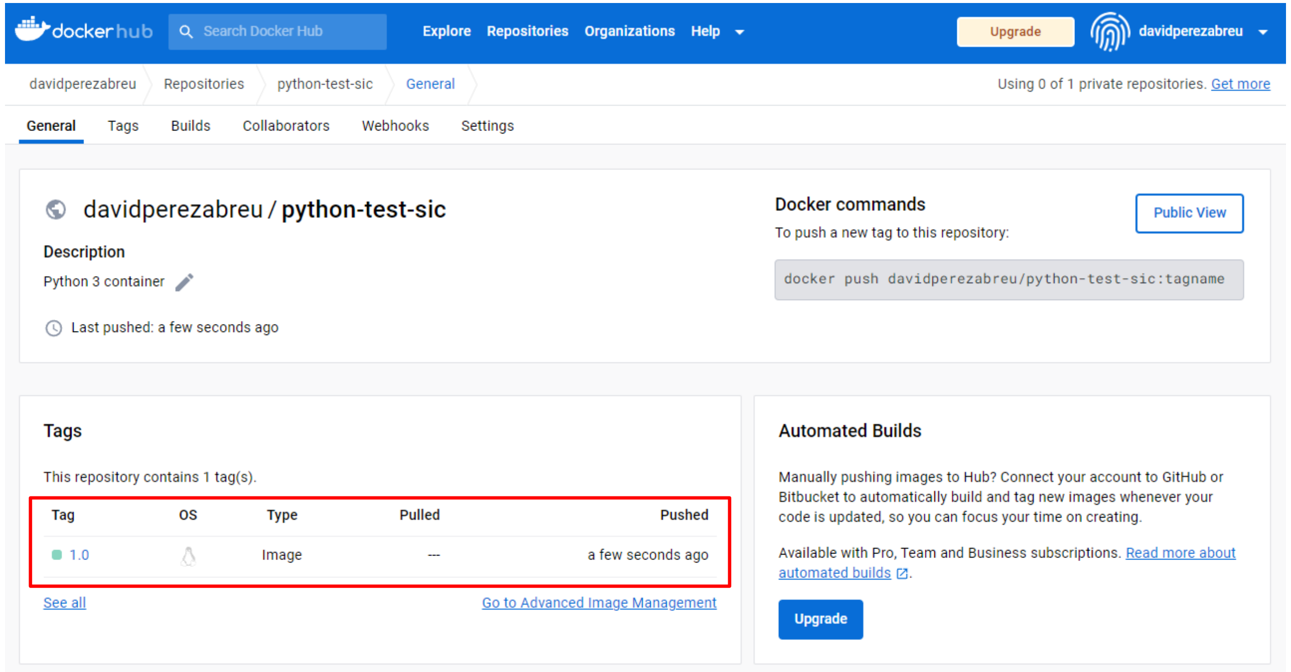
You can push a new image to this repository using the CLI:

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to replace `tagname` with your desired image repository tag.

Pushing your image to Docker Hub (3/3)

- Rename the Python image
  - #>docker tag python-test:1.0 \
    <your-DockerHub-user-name>/python-test-sic:1.0
- Login in the docker hub in your VM and provide the required information
  - #>docker login
- Push the image
  - docker push \
    <your-DockerHub-user-name>/python-test-sic:1.0



Using 0 of 1 private repositories. Get more

davidperezabreu / python-test-sic

Description

Python 3 container

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository:

```
docker push davidperezabreu/python-test-sic:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
1.0	🐧	Image	---	a few seconds ago

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Environment variables

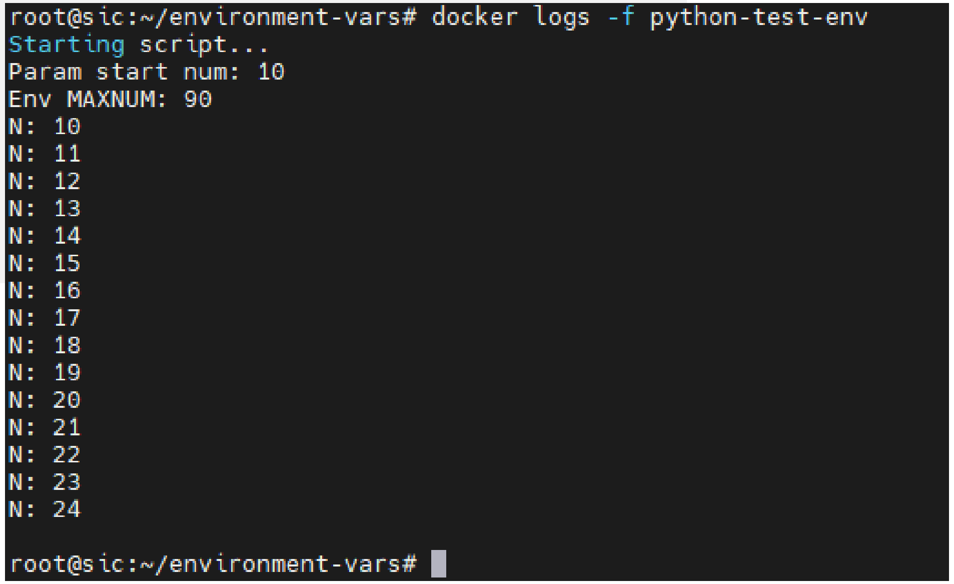
- Adaptation from the app.py application
- Use argument to receive start value
- Use environment variable to receive max value

```
#!/usr/bin/env python3

import time
import os
import sys

if __name__ == '__main__':
    print("Starting script...")

    start = int(sys.argv[1])
    max = int(os.environ.get("MAXNUM"))

    print("Param start num: "+str(start))
    print("Env MAXNUM: "+str(max))

    for n in range(start,max):
        print("N: "+str(n))
        time.sleep(1)

    print("Ending script!");
```

Dockerfile

```
# Use an official Python base image
FROM python:3

# Environment variables
ENV MAXNUM=90
ENV STARTNUM=10

# Sets the working directory for your app. This will influence the following COPY and CMD commands
WORKDIR /usr/src/myapp

# Copy the required files from the VM to the container (docker) image
COPY app-env-vars.py .

# Run app.py inside the image filesystem
CMD python app-env.py ${STARTNUM}
```

- How can we verify the start and max numbers inside the container?
- You have 5 min to figure out. Discuss with your colleagues how to do it

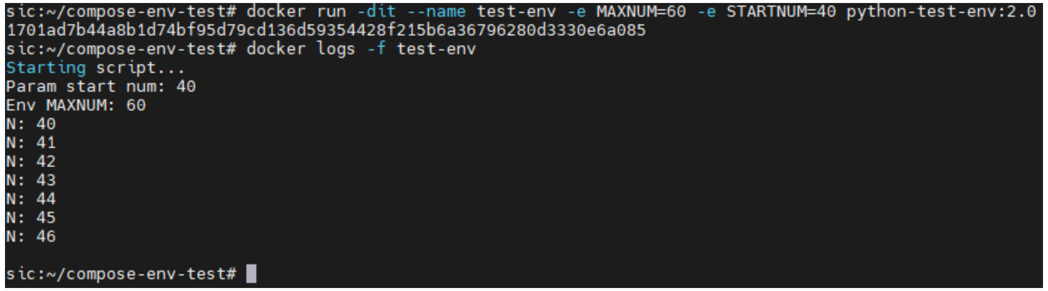


```
root@sic:~/environment-vars# docker logs -f python-test-env
Starting script...
Param start num: 10
Env MAXNUM: 90
N: 10
N: 11
N: 12
N: 13
N: 14
N: 15
N: 16
N: 17
N: 18
N: 19
N: 20
N: 21
N: 22
N: 23
N: 24

root@sic:~/environment-vars# 
```

Environment variables in Dockerfile

- Avoid setting environment variables in Dockerfile – restricts dynamicity
- Set environment variables when running an image

```
docker run -dit \
  --name test-env \
  -e MAXNUM=60 \
  -e STARTNUM=40 \
  python-test-env:2.0
```



```
sic:~/compose-env-test# docker run -dit --name test-env -e MAXNUM=60 -e STARTNUM=40 python-test-env:2.0
1701ad7b44a8b1d74bf95d79cd136d59354428f215b6a36796280d3330e6a085
sic:~/compose-env-test# docker logs -f test-env
Starting script...
Param start num: 40
Env MAXNUM: 60
N: 40
N: 41
N: 42
N: 43
N: 44
N: 45
N: 46

sic:~/compose-env-test# 
```

```
# Use an official Python base image
FROM python:3

# Environment variables
#ENV MAXNUM=90
#ENV STARTNUM=10

# Sets the working directory for your app. This will
influence the following COPY and CMD commands
WORKDIR /usr/src/myapp

# Copy the required files from the VM to the
container (docker) image
COPY app-env-vars.py .

# Run app.py inside the image filesystem
CMD python app-env.py ${STARTNUM}
```

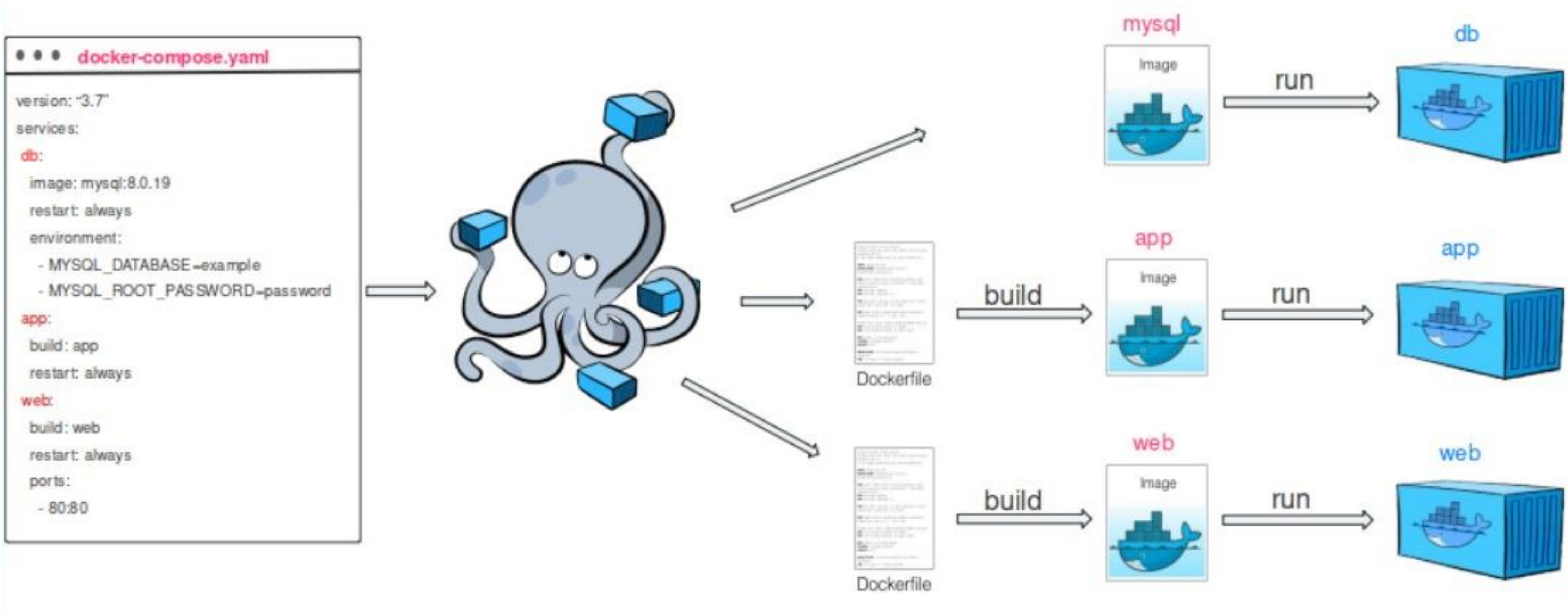
```
docker image build -t python-test-env:2.0 .
```

Docker compose (1/2)

- Tool for defining and running multi-container Docker applications
- With Compose, you use a YAML file to configure your application's services
- With a single command, you create and start all the services from your configuration
- Frequently used to bring more than one container on a single machine
- Organized by services (containers list)
- Services start based on the `depends_on` option
- Use environment variables

- File: compose.yaml

Docker compose (2/2)

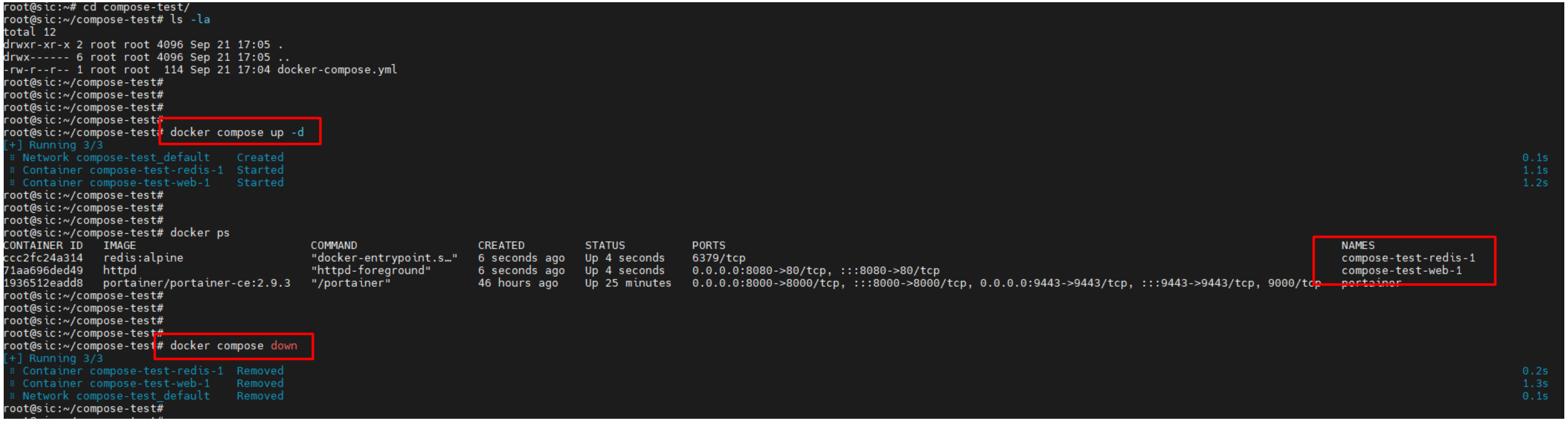


docker-compose.yaml

```
version: “3.7”
services:
 db:
  image: mysql:8.0.19
  restart: always
  environment:
   - MYSQL_DATABASE=example
   - MYSQL_ROOT_PASSWORD=password
 app:
  build: app
  restart: always
 web:
  build: web
  restart: always
  ports:
   - 80:80
```

Run a docker compose

- #>docker compose –f <config-file.yaml> up –d
- #>docker ps -a
- #>docker compose –f <config-file.yaml> down

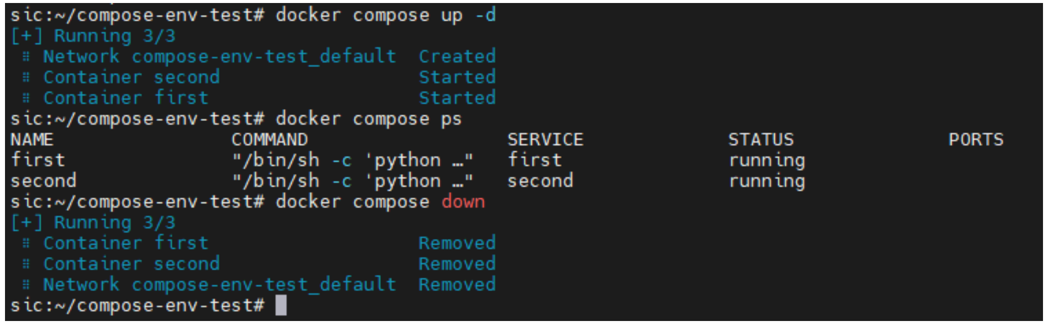


```
root@sic:~# cd compose-test/
root@sic:~/compose-test# ls -la
total 12
drwxr-xr-x 2 root root 4096 Sep 21 17:05 .
drwx------ 6 root root 4096 Sep 21 17:05 ..
-rw-r--r-- 1 root root  114 Sep 21 17:04 docker-compose.yml
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test# docker compose up -d
[+] Running 3/3
 ⠿ Network compose-test_default    Created                                                                                                                                                                                         0.1s
 ⠿ Container compose-test-redis-1  Started                                                                                                                                                                                         1.1s
 ⠿ Container compose-test-web-1    Started                                                                                                                                                                                         1.2s
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test# docker ps
CONTAINER ID   IMAGE                            COMMAND                  CREATED          STATUS          PORTS                                                                                            NAMES
ccc2fc24a314   redis:alpine                     "docker-entrypoint.s…"   6 seconds ago    Up 4 seconds    6379/tcp                                                                                         compose-test-redis-1
71aa696ded49   httpd                            "httpd-foreground"       6 seconds ago    Up 4 seconds    0.0.0.0:8080->80/tcp, :::8080->80/tcp                                                            compose-test-web-1
1936512eadd8   portainer/portainer-ce:2.9.3     "/portainer"             46 hours ago     Up 25 minutes   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp, 0.0.0.0:9443->9443/tcp, :::9443->9443/tcp, 9000/tcp   portainer
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test#
root@sic:~/compose-test# docker compose down
[+] Running 3/3
 ⠿ Container compose-test-redis-1  Removed                                                                                                                                                                                         0.2s
 ⠿ Container compose-test-web-1    Removed                                                                                                                                                                                         1.3s
 ⠿ Network compose-test_default    Removed                                                                                                                                                                                         0.1s
root@sic:~/compose-test#
```

Environment variables

- Set environment variables in the docker-compose file
- Env files can also be specified per compose file or per service

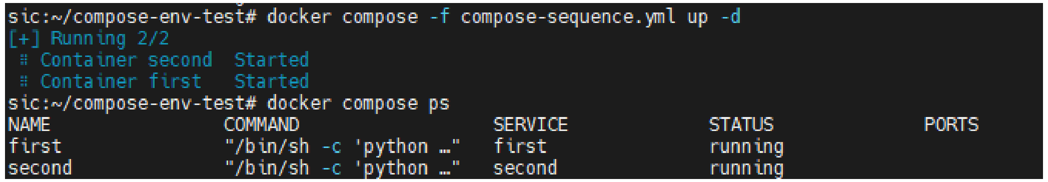
```
version: "3.8"
services:
  first:
    image: python-test-env:2.0
    container_name: first
    environment:
      MAXNUM: 80
      STARTNUM: 20
  second:
    image: python-test-env:2.0
    container_name: second
    environment:
      MAXNUM: 70
      STARTNUM: 30
```



```
sic:~/compose-env-test# docker compose up -d
[+] Running 3/3
 ⠿ Network compose-env-test_default  Created
 ⠿ Container second                  Started
 ⠿ Container first                   Started
sic:~/compose-env-test# docker compose ps
NAME                COMMAND                  SERVICE             STATUS              PORTS
first               "/bin/sh -c 'python …"   first               running
second              "/bin/sh -c 'python …"   second              running
sic:~/compose-env-test# docker compose down
[+] Running 3/3
 ⠿ Container first                   Removed
 ⠿ Container second                  Removed
 ⠿ Network compose-env-test_default  Removed
sic:~/compose-env-test# 
```

Sequence services

- Define startup and shutdown dependencies between services
- First start service “second” and only then service “first”
- Advanced sequence options are available

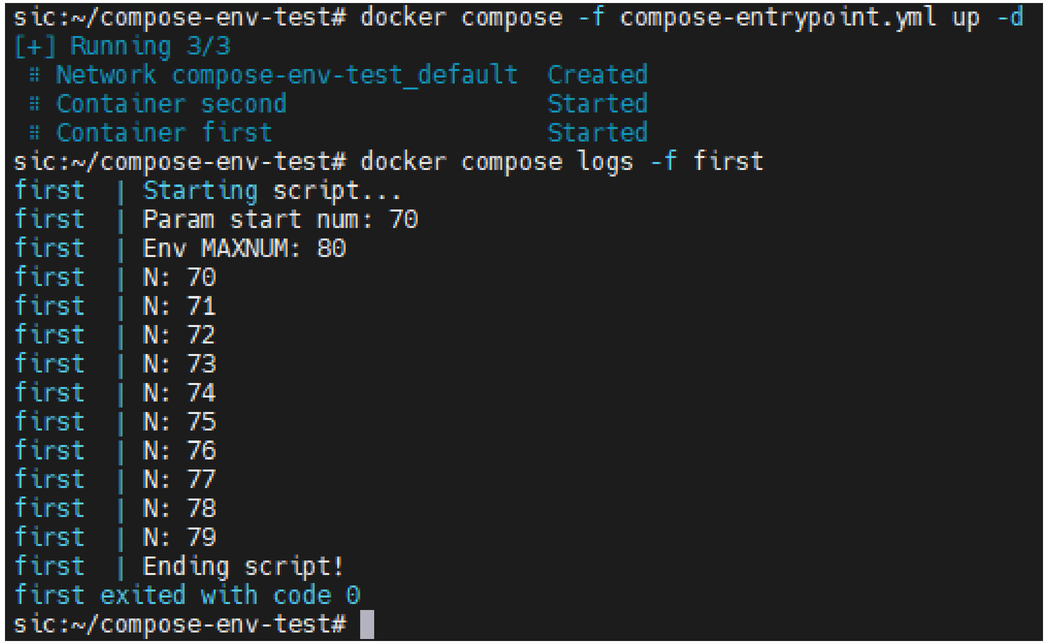


```
sic:~/compose-env-test# docker compose -f compose-sequence.yml up -d
[+] Running 2/2
 ⠿ Container second  Started
 ⠿ Container first   Started
sic:~/compose-env-test# docker compose ps
NAME                COMMAND                  SERVICE             STATUS              PORTS
first               "/bin/sh -c 'python …"   first               running
second              "/bin/sh -c 'python …"   second              running
```

```
version: "3.8"
services:
  first:
    image: python-test-env:2.0
    container_name: first
    environment:
      MAXNUM: 80
      STARTNUM: 20
    depends_on:
      - second
  second:
    image: python-test-env:2.0
    container_name: second
    environment:
      MAXNUM: 70
      STARTNUM: 30
```

Entrypoint

- Override image ENTRYPOINT or CMD



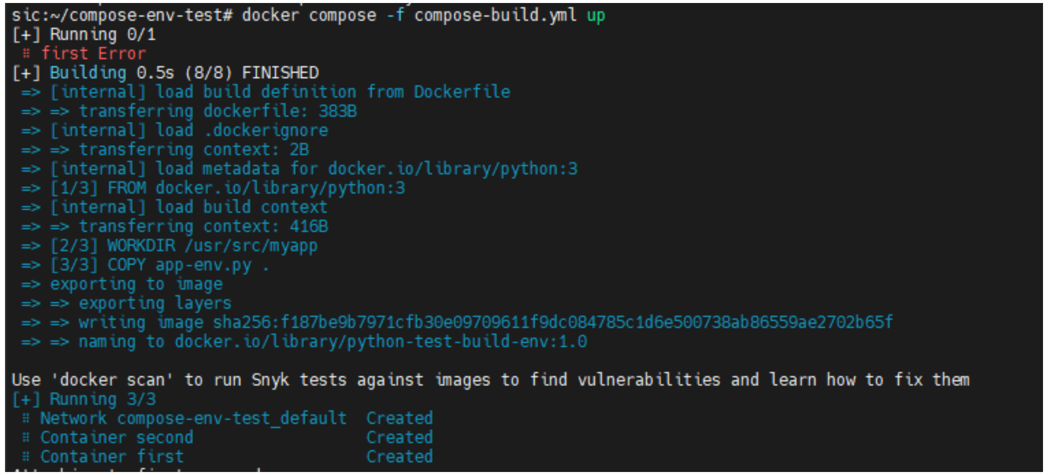
```
sic:~/compose-env-test# docker compose -f compose-entrypoint.yml up -d
[+] Running 3/3
 ⠿ Network compose-env-test_default  Created
 ⠿ Container second                  Started
 ⠿ Container first                   Started
sic:~/compose-env-test# docker compose logs -f first
first  | Starting script...
first  | Param start num: 70
first  | Env MAXNUM: 80
first  | N: 70
first  | N: 71
first  | N: 72
first  | N: 73
first  | N: 74
first  | N: 75
first  | N: 76
first  | N: 77
first  | N: 78
first  | N: 79
first  | Ending script!
first exited with code 0
sic:~/compose-env-test# 
```

```
version: "3.8"
services:
  first:
    image: python-test-env:2.0
    container_name: first
    environment:
      MAXNUM: 80
      STARTNUM: 20
        #depends_on:
        #- second
    entrypoint: python app-env-vars.py 70
  second:
    image: python-test-env:2.0
    container_name: second
    environment:
      MAXNUM: 70
      STARTNUM: 30
```

Build

- Build image using docker compose
- Check also:

```
#>docker inspect first
#>docker compose -f compose-build.yaml build
```



```
sic:~/compose-env-test# docker compose -f compose-build.yml up
[+] Running 0/1
 ⠿ first Error
[+] Building 0.5s (8/8) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 383B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/python:3
 => [1/3] FROM docker.io/library/python:3
 => [internal] load build context
 => => transferring context: 416B
 => [2/3] WORKDIR /usr/src/myapp
 => [3/3] COPY app-env.py .
 => exporting to image
 => => exporting layers
 => => writing image sha256:f187be9b7971cfb30e09709611f9dc084785c1d6e500738ab86559ae2702b65f
 => => naming to docker.io/library/python-test-build-env:1.0

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 3/3
 ⠿ Network compose-env-test_default  Created
 ⠿ Container second                  Created
 ⠿ Container first                   Created
```

```
version: "3.8"
services:
  first:
    image: python-test-env:1.0
    build: ./
    container_name: first
    environment:
      MAXNUM: 80
      STARTNUM: 20
  second:
    image: python-test-env:2.0
    container_name: second
    environment:
      MAXNUM: 70
      STARTNUM: 30
```