# December 6th Rotterdam

## Docker Workshop:

Containerize your python applications
(with a RESTful interface)

*Bahadir Kasap - KPN ICT Consulting*

https://github.com/bkasap/dypa-workshop

# Introduction round

- Who are we?
- What we do?
- Why are we here?

# Containerization process


*In April 1956, the first container ship set out from New York City to Houston. Fifty-eight boxes were brought ashore in mere hours, and a day later the vessel was making its way back with another full load of cargo.Before the invention of the steel box, ships might spend four to six days* ***at port, fully 50% of their time. A couple years later, just 10%.*** (Rutger Bregman - Utopia for realists)

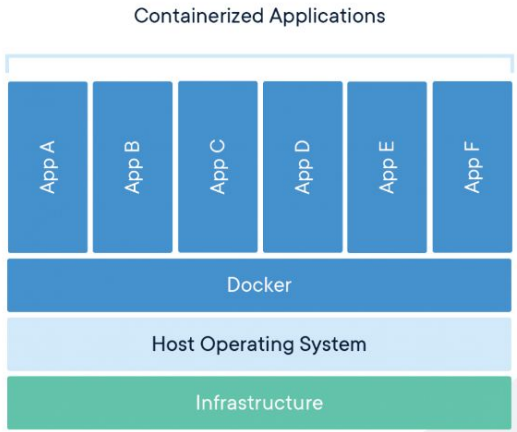
docker

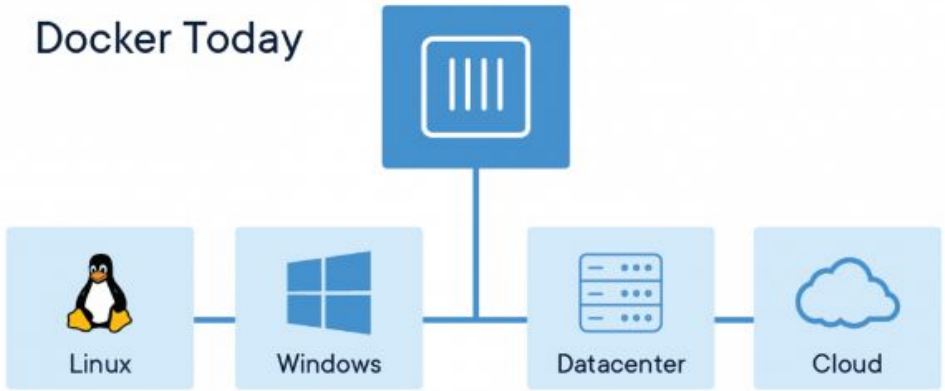
Why Docker? Products Solutions Customers Resources

Sign In Get Started

# What is a Container?

A standardized unit of software

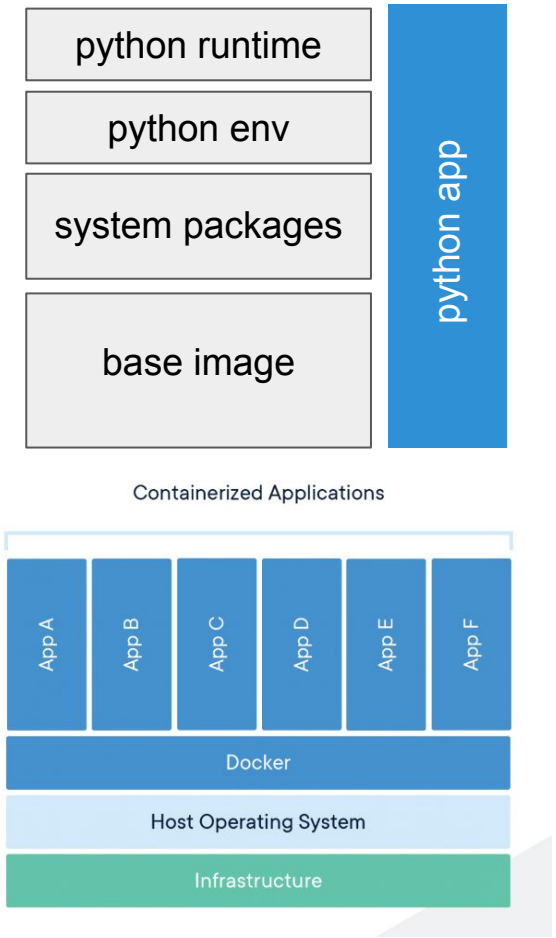


# What's inside a python container?

- python script and files
- python environment (`import this`)
  - `pip install package`
  - `conda install package`
  - `...`
- system dependencies ( `sdk, gcc, setuptools, etc…`)
  - `apk add package` (alpine package manager)
  - `apt install package` (deb-ubuntu package manager)
  - `...`
- base image (`alpine, deb, arch, ubuntu, etc...`)

All declared in `Dockerfile` to build container image.



```
import this
import math
```

*Pip, conda, venv etc…*

List installed packages in a python environment:

```
pip freeze > requirements.txt
```

Setup the same python environment on another system:

```
pip install -r requirements.txt
```

# pip vs. conda

https://www.anaconda.com/understanding-conda-and-pip/

**Comparison of conda and pip**

| | conda | pip |
|---|---|---|
| manages | binaries | wheel or source |
| can require compilers | no | yes |
| package types | any | Python-only |
| create environment | yes, built-in | no, requires virtualenv or venv |
| dependency checks | yes | no |
| package sources | Anaconda repo and cloud | PyPI |

# `import responder`

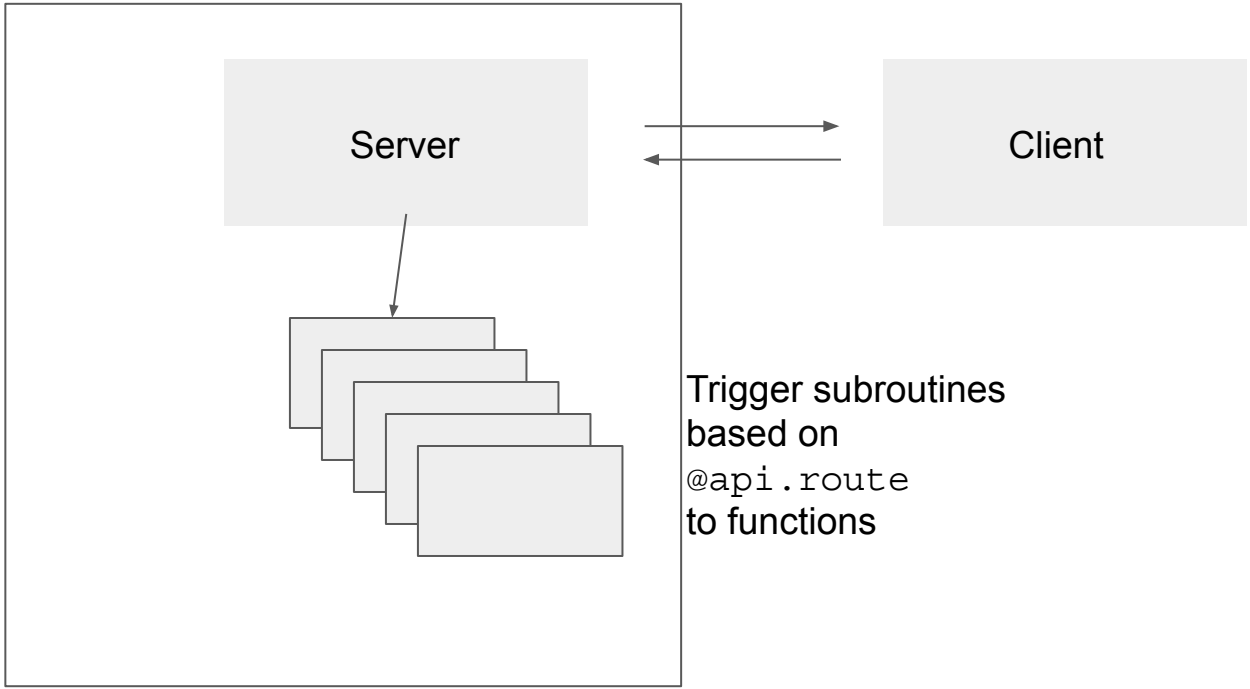
A familiar HTTP Service Framework for Python. https://responder.readthedocs.io

```
import responder

api = responder.API()

@api.route("/")
def hello_world(req, resp):
      resp.text = "hello, world!"


api.run()
```

Also see:

```
import tornado
import sanic
import vibora
import quart
import fastapi
```

# What will we stitch together?



# Hands on!

Questions, remarks, wishes, expectations?

https://github.com/bkasap/dypa-workshop