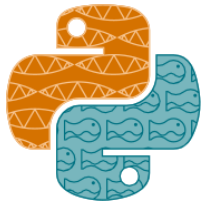


Docker meets Python – A look on the Docker SDK for Python

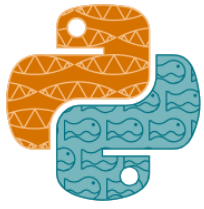
“pip install docker”

Jan Wagner
Data Science Consultant



Agenda

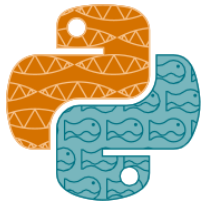
1. Who am I?
2. The Docker Daemon/Service and Ways to communicate with it
3. Docker SDK for Python!
 - a. Where and how to get it
 - b. Code Examples
4. Ideas for Usecases
 - a. Using Python as Container-Starting-Script
... end extend it
 - b. Get Container-Logs for further processing
 - c. pytest with different Python Versions
5. Wrap Up
6. Q&A



About me – Hi, my name is Jan!

- 32 Years old, Not married yet, but engaged
- Data Science Consultant @ accantec consulting AG
- We focus on Data
 - BI, Data Engineering & Data Science
- <https://accantec.de>





About me – Get in touch



j.wagner@accantec.com

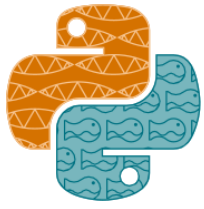


<https://twitter.com/wgnrjn>

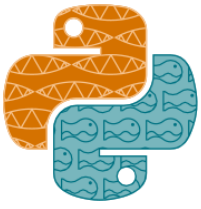
<https://www.instagram.com/wgnrjn>



<https://github.com/wgnrjn>



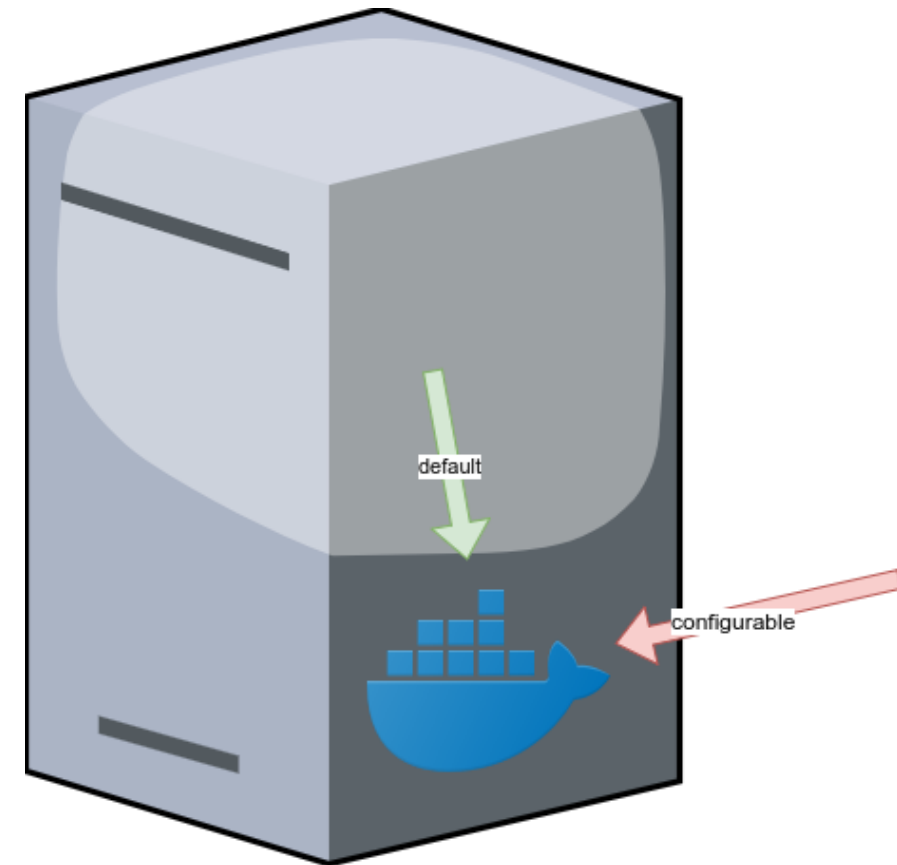
Docker

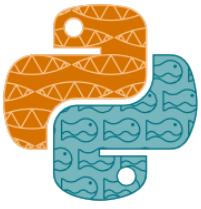


Your friendly house guest – The Whale!

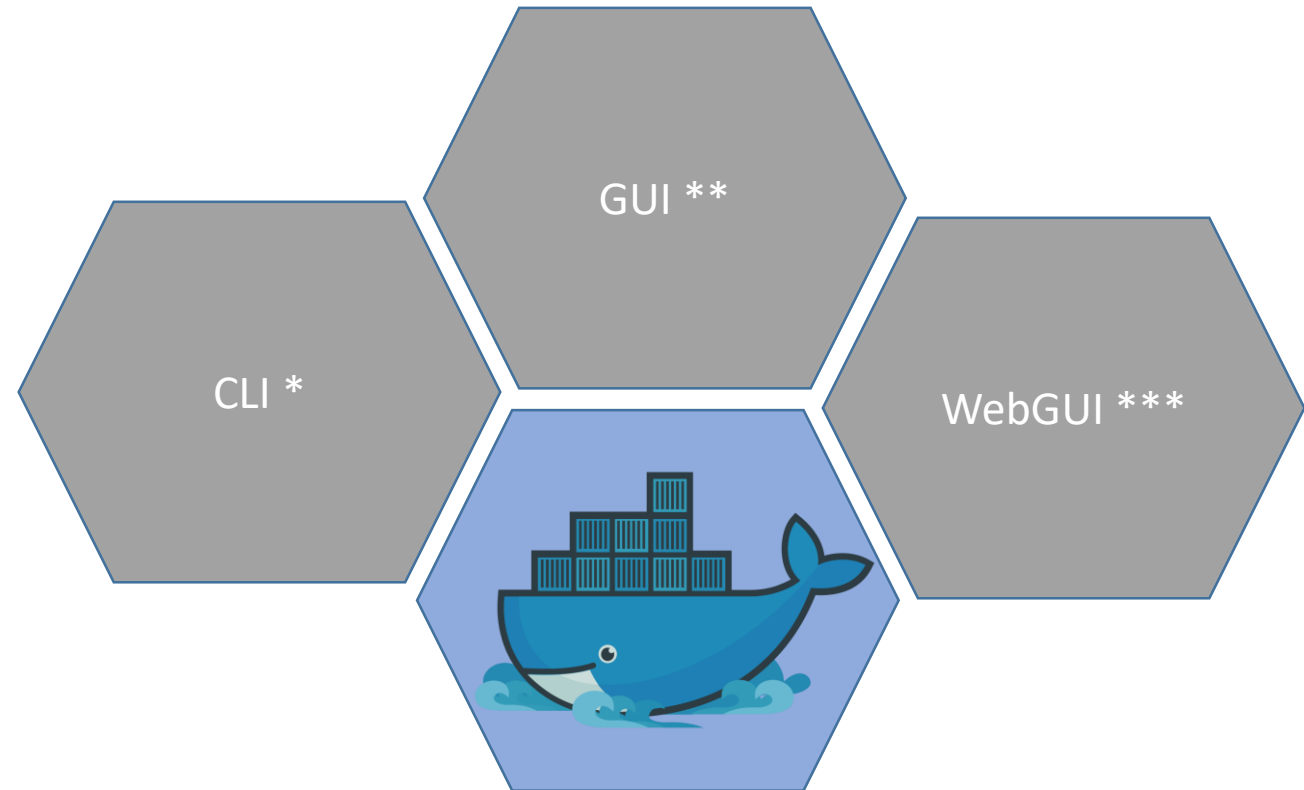
The Docker Daemon/Service:

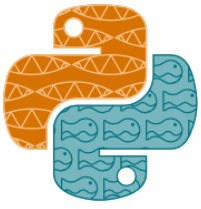
- Reachable per default from within your System (localhost)
- Reachable per configuration from the outside





Ways to communicate with it

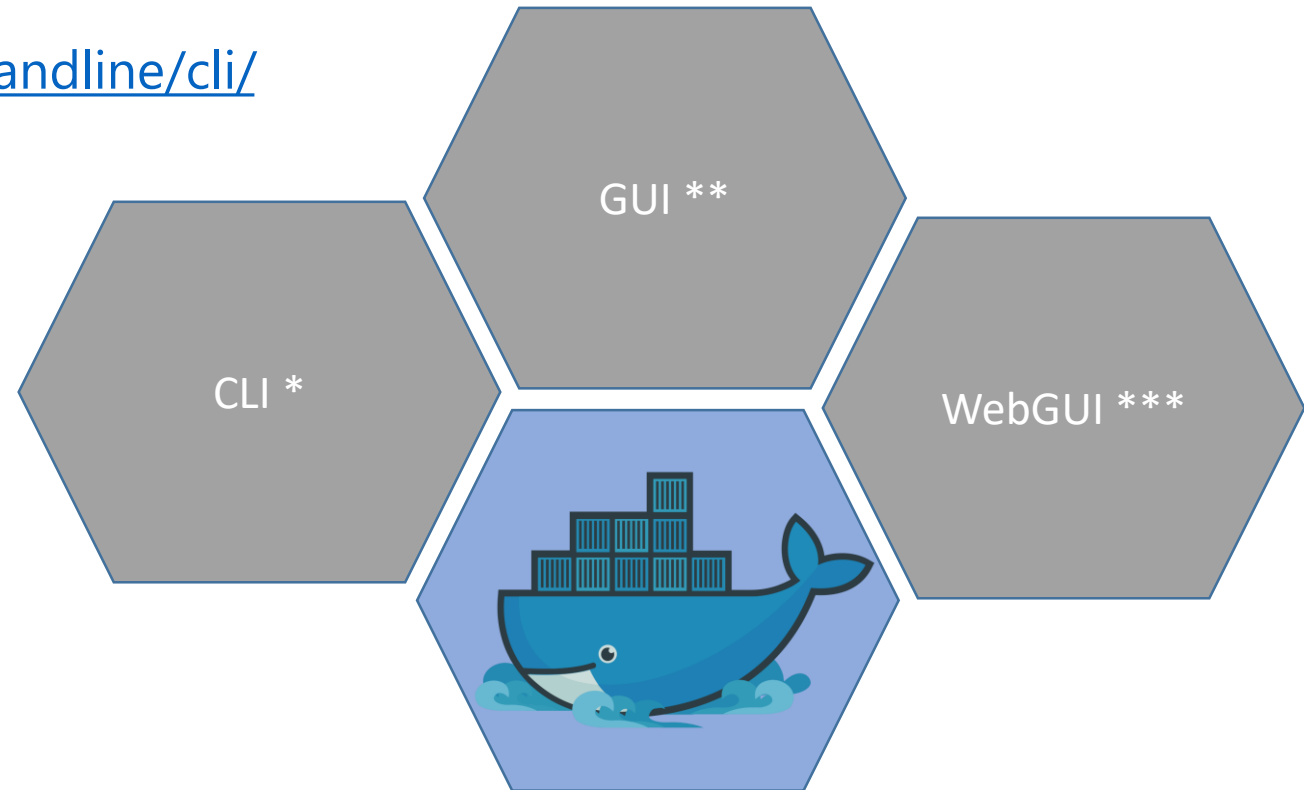


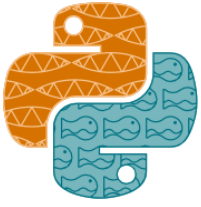


Ways to communicate with it

* Docker CLI (\$docker run hello-world)

<https://docs.docker.com/engine/reference/commandline/cli/>





Ways to communicate with it

* Docker CLI (\$docker run hello-world)

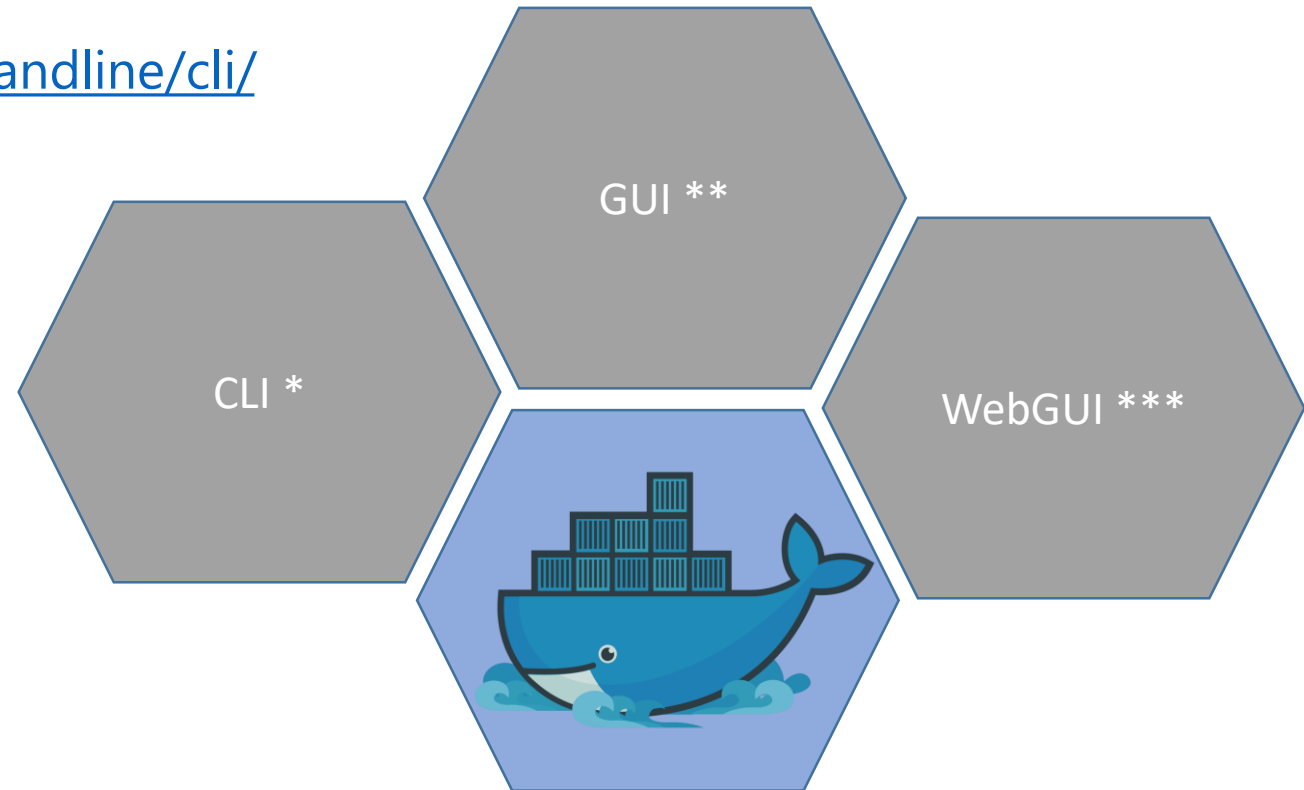
<https://docs.docker.com/engine/reference/commandline/cli/>

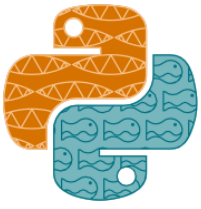
** Kitematic

<https://kitematic.com/>

** Docker Extension for VS Code

<https://github.com/microsoft/vscode-docker>





Ways to communicate with it

* Docker CLI (\$docker run hello-world)

<https://docs.docker.com/engine/reference/commandline/cli/>

** Kitematic

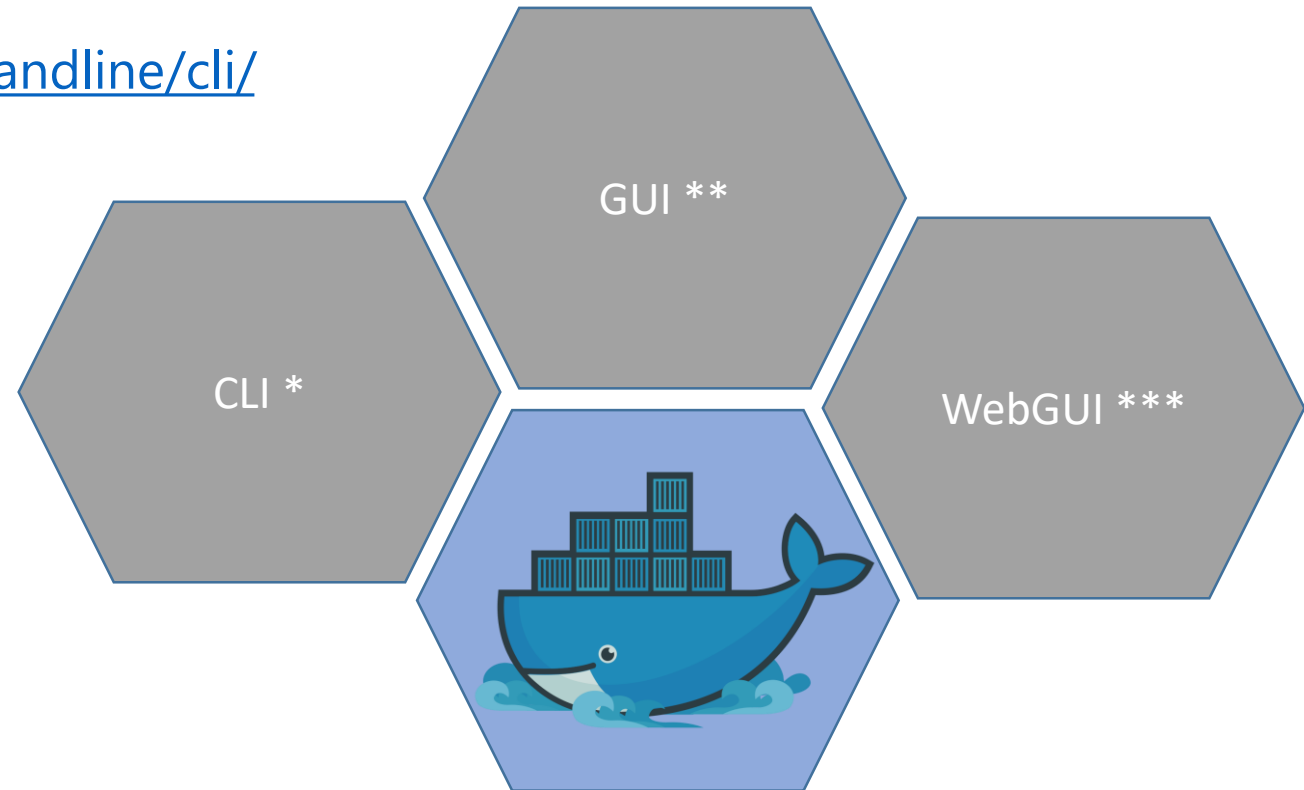
<https://kitematic.com/>

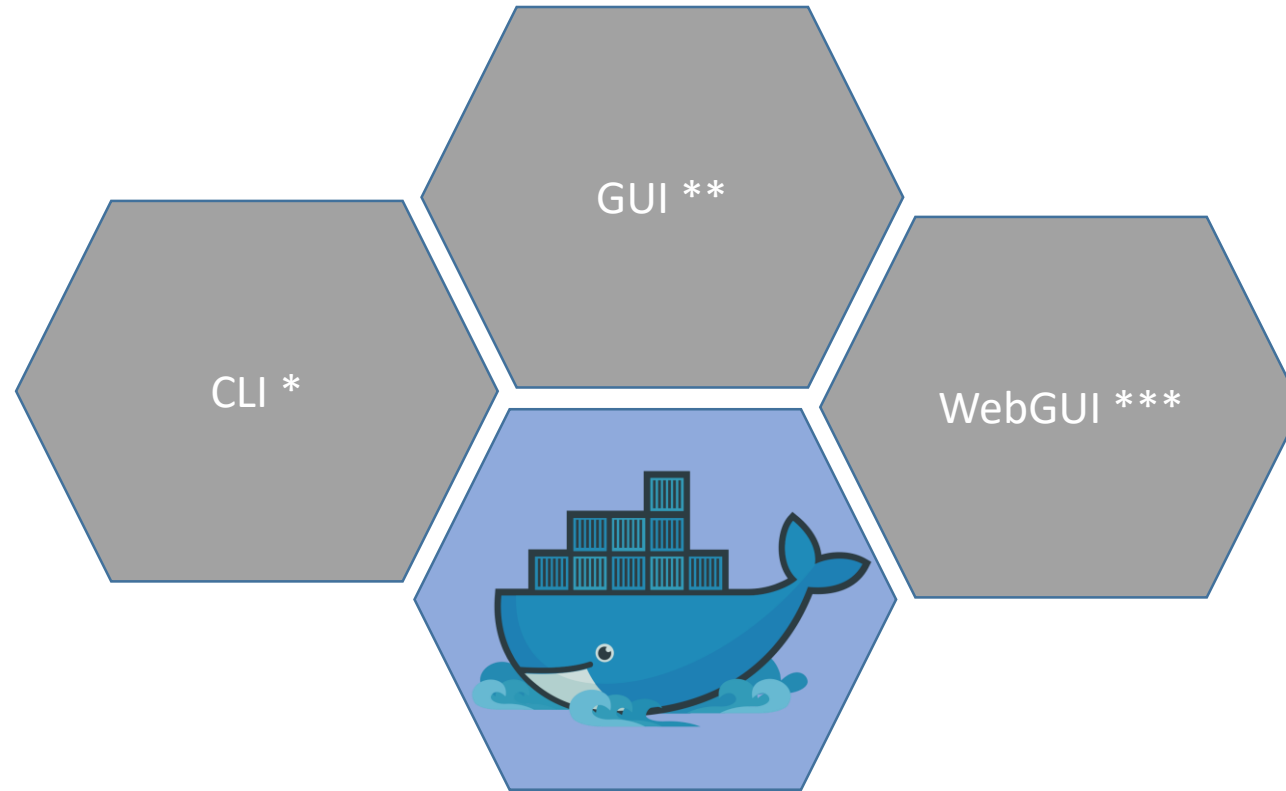
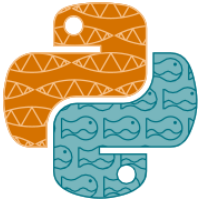
** Docker Extension for VS Code

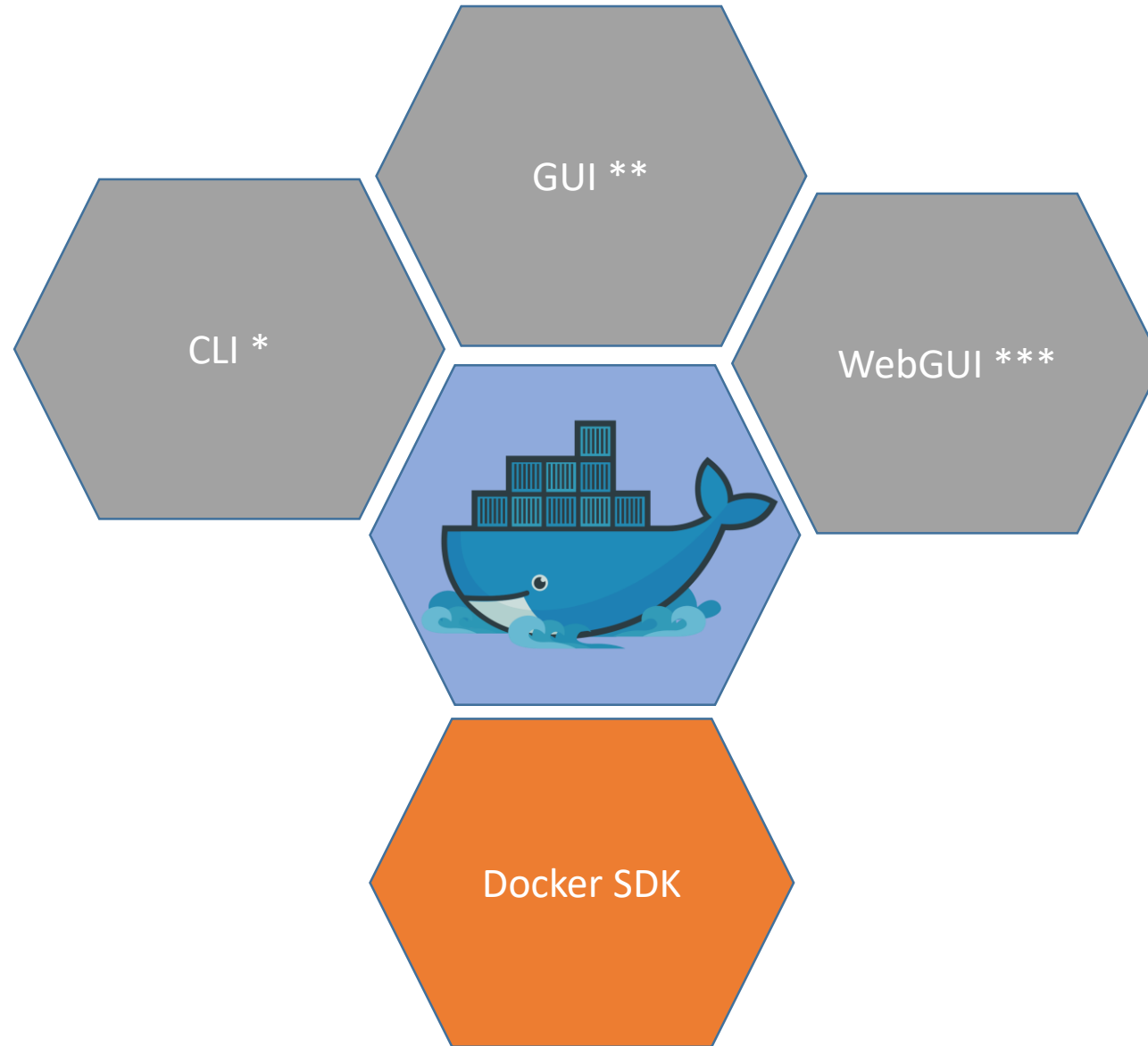
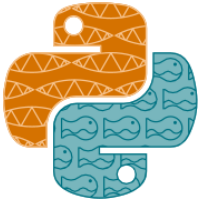
<https://github.com/microsoft/vscode-docker>

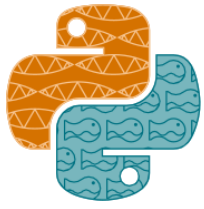
*** Portainer

<https://www.portainer.io/>





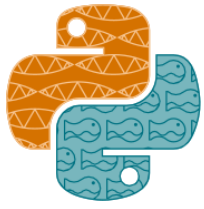




Get the Docker SDK for Python

- Officially available for Python and Golang

<https://docs.docker.com/develop/sdk/>



Get the Docker SDK for Python

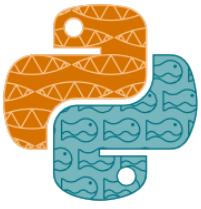
- Officially available for Python and Golang

<https://docs.docker.com/develop/sdk/>

- pip/pip3 install docker

- Package on conda-forge available too

<https://anaconda.org/conda-forge/docker-py>

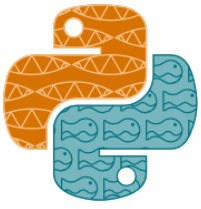


Code Examples

Taken from <https://docs.docker.com/develop/sdk/examples/> *

Docker CLI	Docker SDK – Python
<pre>\$ docker run hello-world</pre>	<pre>import docker client = docker.from_env() print (client.containers.run('hello-world'))</pre>
<pre>\$ docker pull alpine</pre>	<pre>import docker client = docker.from_env() image = client.images.pull("alpine") print (image.id)</pre>

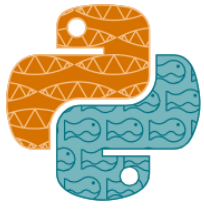
* Code Examples on Website are in Python 2 Syntax ... I don't know why ☺



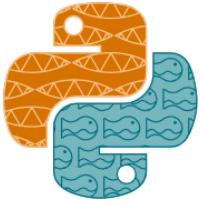
Hint!

Docker CLI	Docker SDK – Python
<pre>\$ docker image [...] \$ docker images</pre>	<pre>import docker client = docker.from_env() client.images.[...]</pre>
<pre>\$ docker container [...]</pre>	<pre>import docker client = docker.from_env() container = client.containers.[...]</pre>

<https://docker-py.readthedocs.io/en/stable/client.html>
<https://docs.docker.com/engine/reference/commandline/container/>
<https://docs.docker.com/engine/reference/commandline/image/>
<https://docs.docker.com/engine/reference/commandline/images/>



Ideas for Usecases

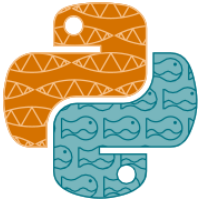


Using Python as Container-Starting-Script

https://hub.docker.com/_/postgres

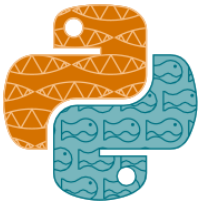
```
$ docker run --name test-db -e POSTGRES_PASSWORD=EuroPython -d postgres
```

```
$ docker run --name test-db -e POSTGRES_PASSWORD=EuroPython -v /some/where/data:/var/lib/postgresql/data [.....] postgres
```



Using Python as Container-Starting-Script

```
import os  
  
os.system ("Do-Stuff -parameterXYZ")
```

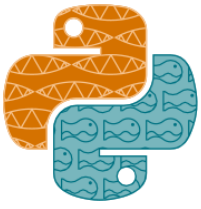


Using Python as Container-Starting-Script

```
import docker

var1 = "... "
var2 = ["...", "123"]
var3 = "... "

client = docker.from_env()
client.containers.run('postgres', name=var1, environment=var2,
mounts=var3, detach=True)
```

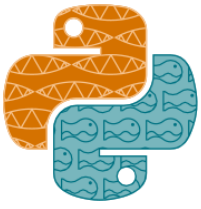


Extend it to ...

[...]

```
client = docker.from_env()  
myContainer = client.containers.get('myTestContainer')  
myContainer.stop()  
client.containers.prune()  
  
newImage = client.images.build('path/to/Dockerfile')  
newImageID = newImage.id()
```

[...]

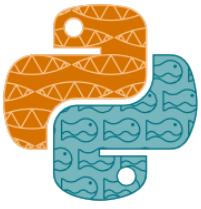


Get Container-Logs

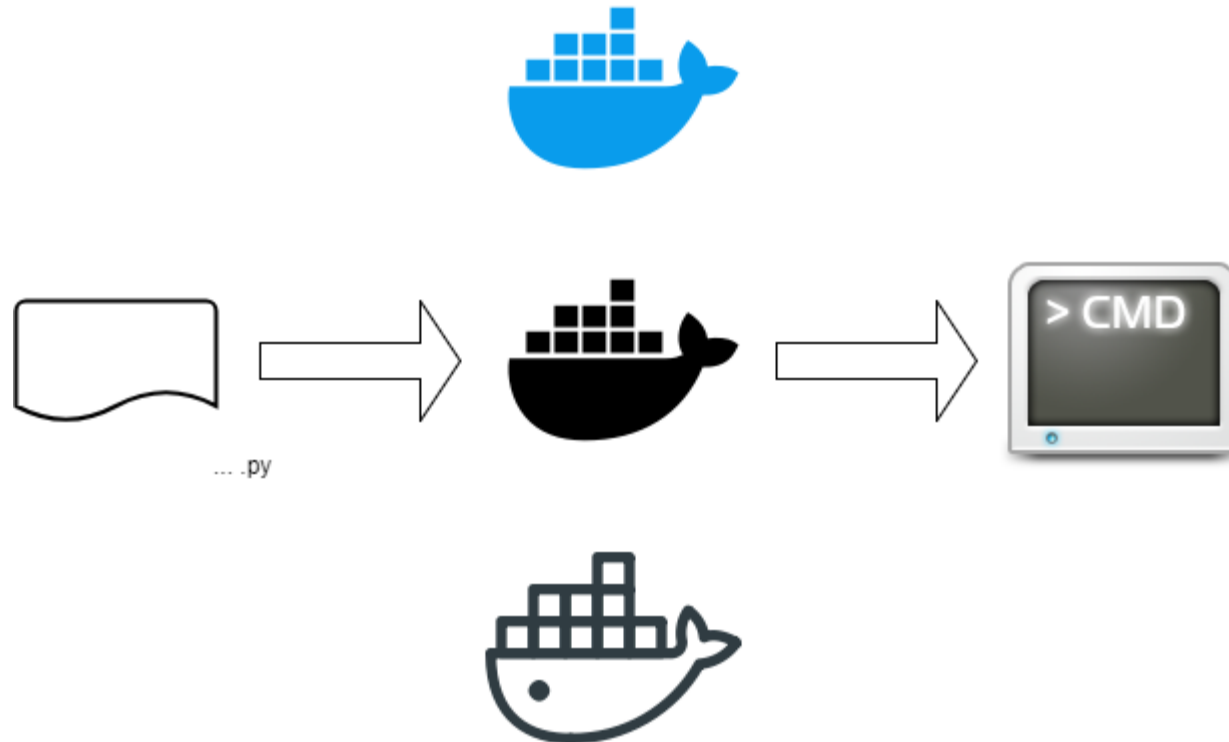
```
import docker
client = docker.from_env()

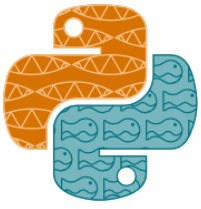
container = client.containers.run('ubuntu', detach=True)

f = open("myContainerLog", "w")
for line in container.logs(stream=True):
    f.write(line)
f.close()
```

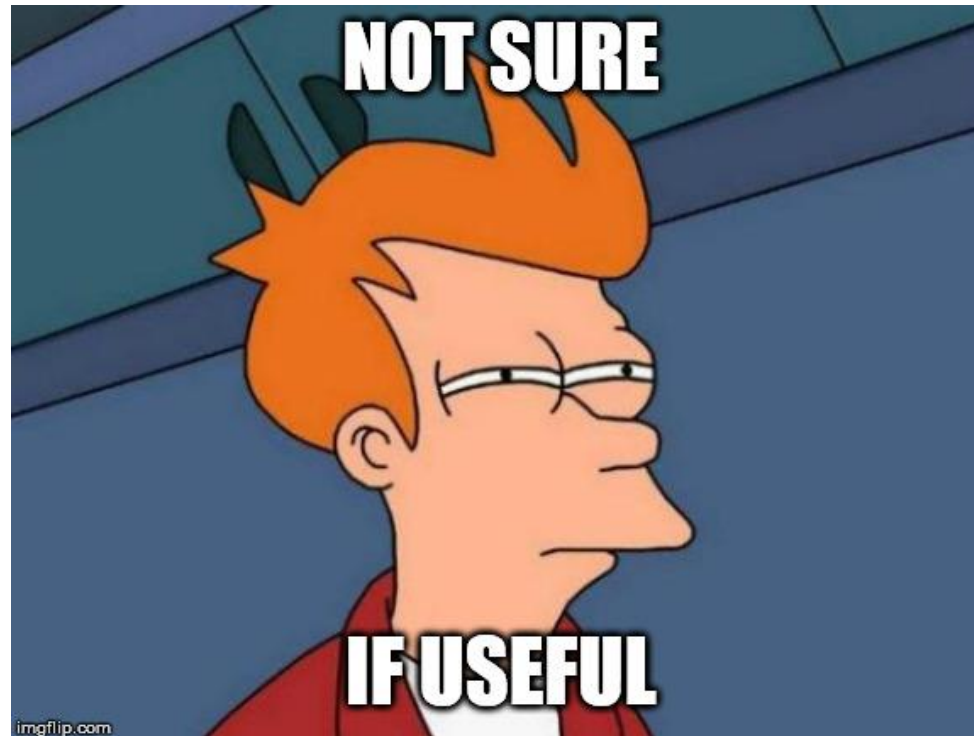


pyTest with different Python Versions

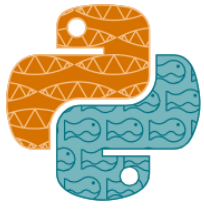




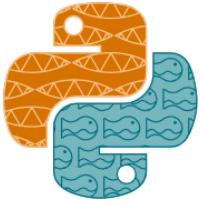
pytest with different Python Versions



.... maybe just build a CI/CD Pipeline in this case



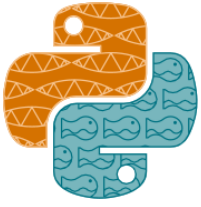
Wrap Up



Wrap Up

Depending on your Task at Hand:

- Python might be a better choice than e.g. bash Scripts
- Docker SDK might be a better choice than „import os“ or equivalents



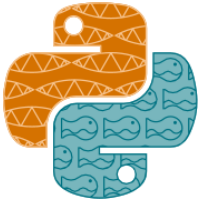
Wrap Up

Depending on your Task at Hand:

- Python might be a better choice than e.g. bash Scripts
- Docker SDK might be a better choice than „import os“ or equivalents

Hope you now have:

- An (high level) Idea about the Docker SDK for Python
- Some inspiration for your Day-to-Day work with Docker and Python



Wrap Up

Depending on your Task at Hand:

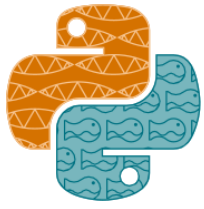
- Python might be a better choice than e.g. bash Scripts
- Docker SDK might be a better choice than „import os“ or equivalents

Hope you now have:

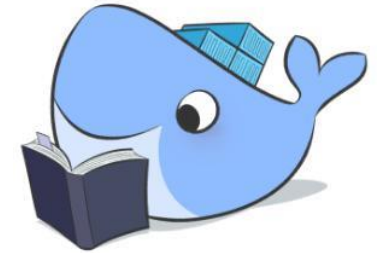
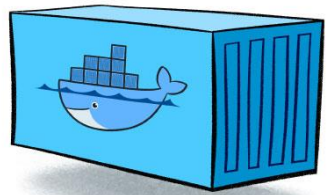
- An (high level) Idea about the Docker SDK for Python
- Some inspiration for your Day-to-Day work with Docker and Python

Last Hint:

- Read the Docker Docs



Thank you!



Questions?

