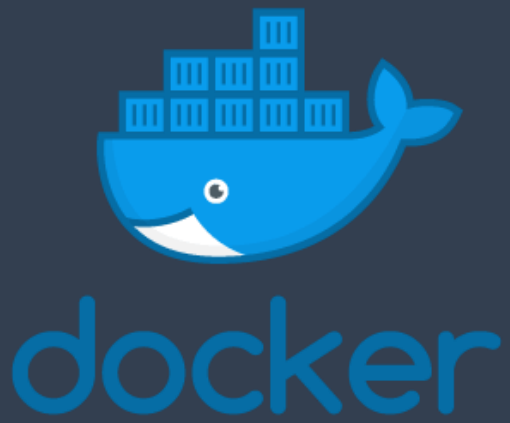


Basic usage of Docker containers in scientific research

Michał Drzał
Sybilla Technologies

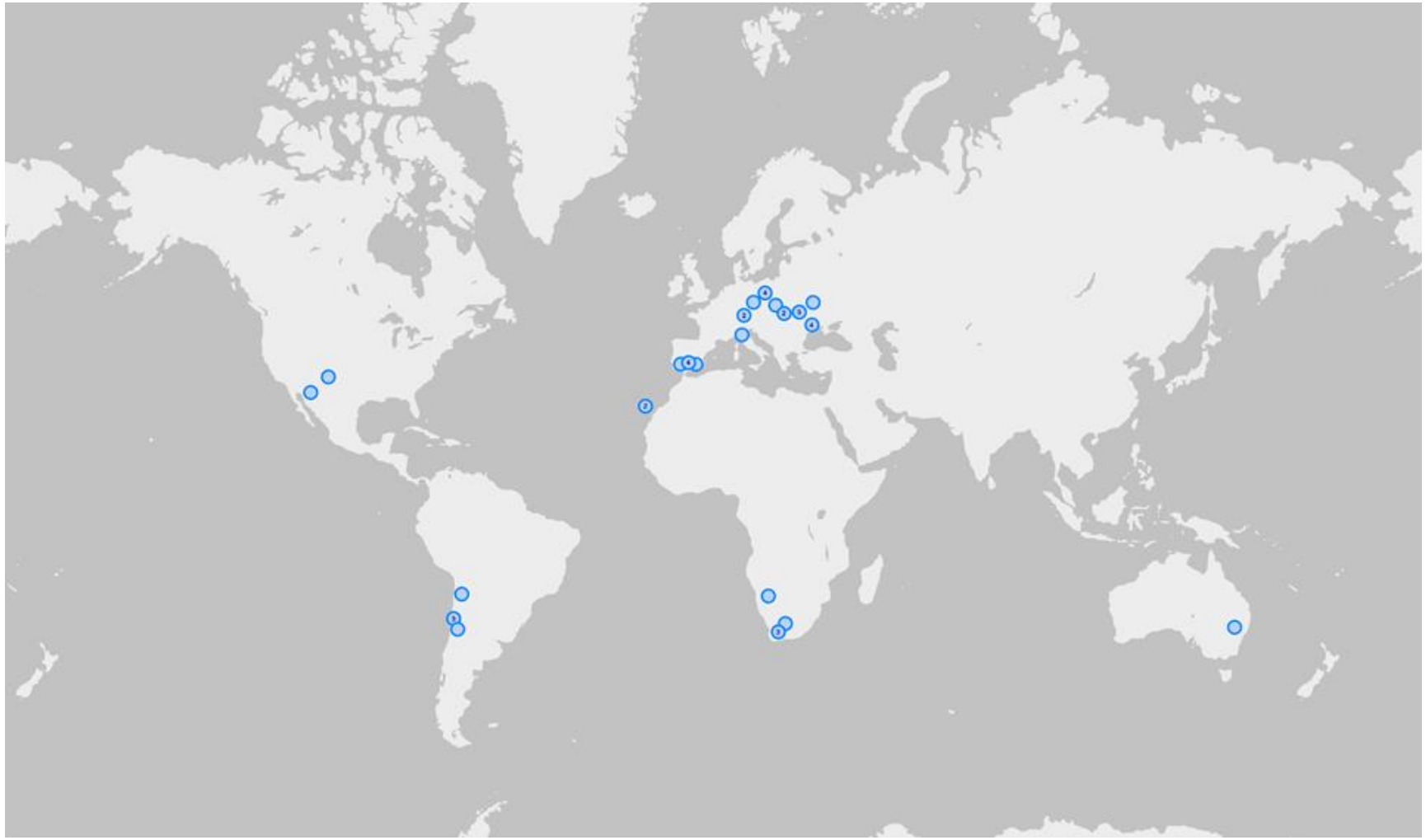
SYBILLA
TECHNOLOGIES

About me

- **MSc in Computer Science (AGH UST Kraków)**
- **Works in Sybilla Technologies**
  - **Robotization of astronomical observatories**
  - **Creating networks of astronomical sensors and scheduling them**
  - **Providing data products and services in SST for ESA and PAK (Polish Space Agency)**

SYBILLA
TECHNOLOGIES

Sensor network



40 sensors running under control of company's scheduling system, largest European network used for ESA and POLSA projects

SYBILLA TECHNOLOGIES

What is Docker?

What is Docker container?

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably **from one computing environment to another**. A Docker container image is a lightweight, standalone, executable package of software **that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.**

Source: https://www.docker.com/resources/what-container

SYBILLA
TECHNOLOGIES

How is Docker different than VirtualBox?

- Virtual machines are running a guest OSes with their own kernels and this virtualization requires preallocating resources, bootstraping kernels
- Docker containers use host kernel and separate between the containers via kernel-level feature called namespaces
- No need to boot the system and kernel – just run the code
- Docker provide also mechanisms for managing networking and sharing directories between containers
- How then Windows and MacOS can use container using Linux Kernel?
  - Both systems use just enough virtualization to have a Linux kernel available for Docker (HyperV/WSL on Windows and Hypervisor on MacOS)

Source: https://stackoverflow.com/questions/16047306/how-is-docker-different-from-a-virtual-machine

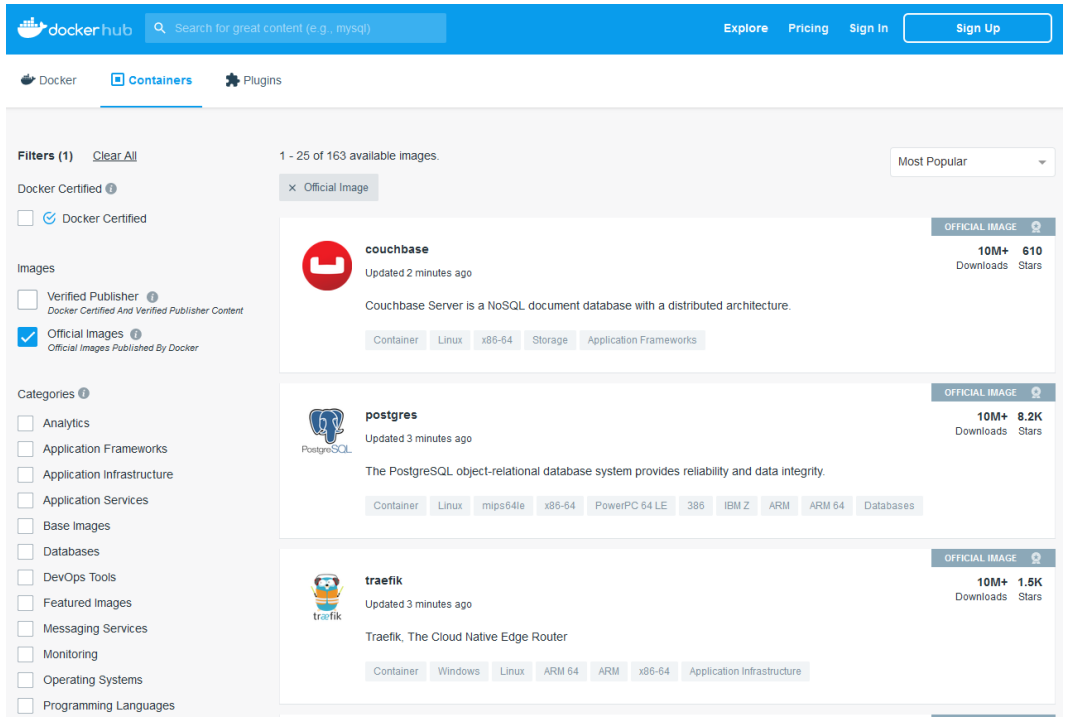
SYBILLA TECHNOLOGIES

Building and running containers

Downloading images from Docker Hub

- Wide selection of ready to use databases, application servers and other ready to download and run container images
- We will focus on jupyter/scipy-notebook

Source: https://hub.docker.com/search?q=&type=image



SYBILLA TECHNOLOGIES

Running containers from Docker Hub

```
docker run -p 8888:8888 jupyter/scipy-notebook:latest
```

- Interaction with docker is done via docker command with different subcommands
- -p publishes container's port (on the right) to host port (one the left)
- Jupyter is the name of the image publisher
- scipy-notebook is the name the image
- latest is the tag which is mostly used for versioning
- Right now container has no persistence – after restart it will lose any files saved internally
  - Fix that with `-v D:/test:/home/jovyan/work` now everything saved in work directory will also be preserved on host system

Source: https://docs.docker.com/engine/reference/commandline/docker/

SYBILLA TECHNOLOGIES

Creating and building Dockerfile

- Dockerfile is recipe for building container image
- Basically a bash build script with extra syntax
  - https://hub.docker.com/r/jupyter/scipy-notebook/dockerfile
- Designed to be reused

```
FROM jupyter/scipy-notebook
RUN pip install --upgrade pip
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
WORKDIR $HOME
```

- To build and run:
  - docker build -t custom_notebook .
  - docker run -p 8888:8888 custom_notebook

SYBILLA TECHNOLOGIES

Creating Dockerfile from scratch

- Select base distribution closest to your needs
  - https://hub.docker.com/search?q=&type=image&operating_system=linux&category=base
  - Debian/Ubuntu are a good first choice - you can select older versions (helpful with legacy) and you get apt-get package manager
  - Alpine is a distro designed with being used within Docker container – good choice when optimizing for smaller container footprint
- Follow exactly same procedure as previously but this time you have to remember about setting entrypoint command
  - CMD specifies the process that is being run inside container
- Simplest Ubuntu-based Dockerfile

```
FROM ubuntu:bionic
CMD ["/bin/bash"]
```

  - Good for wrapping commandline utilities – containers don't have to be long-lived

SYBILLA TECHNOLOGIES

Composing multiple containers

docker-compose.yml

Docker Compose allows for creating a single YAML file with all configuration that can be done via Docker CLI

- Convenient for storage in GIT repository
- Allows for simultaneous building and running of multiple containers at once
- After docker-compose.yml is created starting containers is simply:

```
docker-compose up
```

- And stopping them is:

```
docker-compose down
```

Source: https://docs.docker.com/compose/

SYBILLA TECHNOLOGIES

docker-compose.yml

- Services section defines Docker containers to run
- Volumes section allows for creation of named volumes
- Configuration of containers (mounting configuration files, injecting environment variables)
- Services names are also their hostnames and they by default are put in the same bridge network
  - Access from the host machine is done thanks to port publishing – otherwise network is separated

```
from sqlalchemy import create_engine
db  = create_engine('postgresql://postgres:changeme@db:5432/postgres')
result_set = db.execute("SHOW ALL")
for r in result_set:
    print(r)
```

SYBILLA TECHNOLOGIES

Summary of benefits of using Docker in science and education

- repeatable environment
- easily resetable environment
- reduction in work required for distribution of student assignments
  - Difficult configuration can be done once and everyone else can focus on assignment
- ability to wrap legacy code or code available only for specific linux distribution
- separation from environment
- composing small service ecosystems

SYBILLA
TECHNOLOGIES

Thank you

SYBILLA
TECHNOLOGIES