Agenda

Outline of Todays Tutorial

- We encourage you to use modern libraries and tools
  - Docker for deployment
  - ir_datasets for data wrangling
  - PyTerrier for declarative retrieval pipelines

Agenda:

- Project Tutorial
- Docker Tutorial
- Outlook + We can create the first submissions together

Agenda

Outline of Todays Tutorial

- ❑ We encourage you to use modern libraries and tools
  - Docker for deployment
  - ir_datasets for data wrangling
  - PyTerrier for declarative retrieval pipelines

Agenda:

- ❑ Project Tutorial
- ❑ Docker Tutorial
- ❑ Outlook + We can create the first submissions together

Collect preferences:

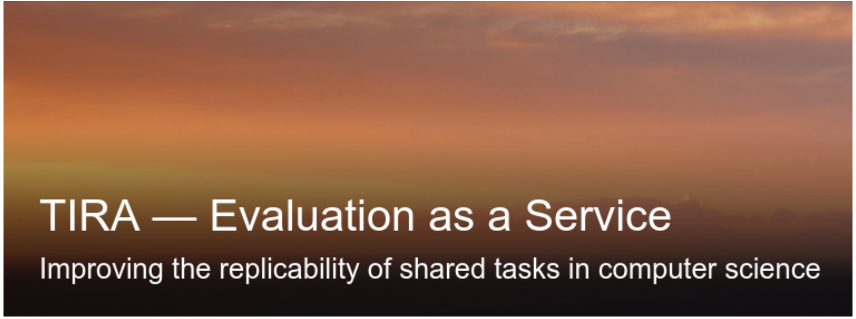
- ❑ How much focus on:
  - Docker (higher technical expertise needed)
  - Dev-Container (medium technical expertise needed)
  - Codespaces (low technical expertise needed)

Project Tutorial

Evaluation and Prototyping with TIRA

You will use TIRA.io for prototyping and the evaluation of the search engines.

`http://tira.io`



TIRA — Evaluation as a Service

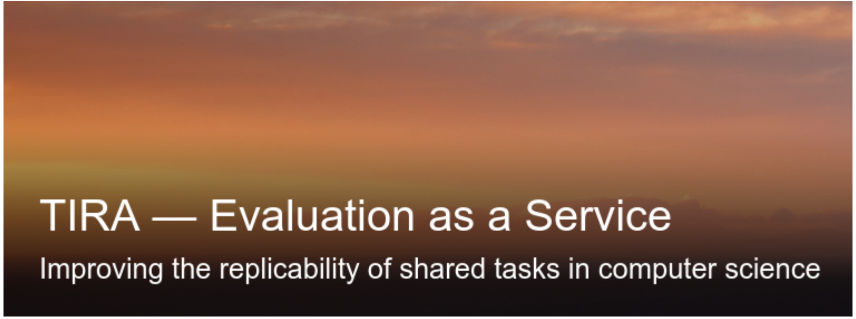
Improving the replicability of shared tasks in computer science

Project Tutorial

Evaluation and Prototyping with TIRA

You will use TIRA.io for prototyping and the evaluation of the search engines.

`http://tira.io`



TIRA — Evaluation as a Service

Improving the replicability of shared tasks in computer science

Procedure:

1. Implement Docker images to handle data access and retrieval
2. Upload image to dedicated image registry in TIRA
3. Everything is executed in a Kubernetes cluster

Project Tutorial

Tutorial

`https://www.tira.io/task-overview/ir-lab-sose-2024`

- Requirements: You need an Github and TIRA.io account
- Overview
  - You develop your system on the training and/or the validation data
  - The developed system(s) are then submitted to the leaderboard

Project Tutorial

Tutorial

https://www.tira.io/task-overview/ir-lab-sose-2024

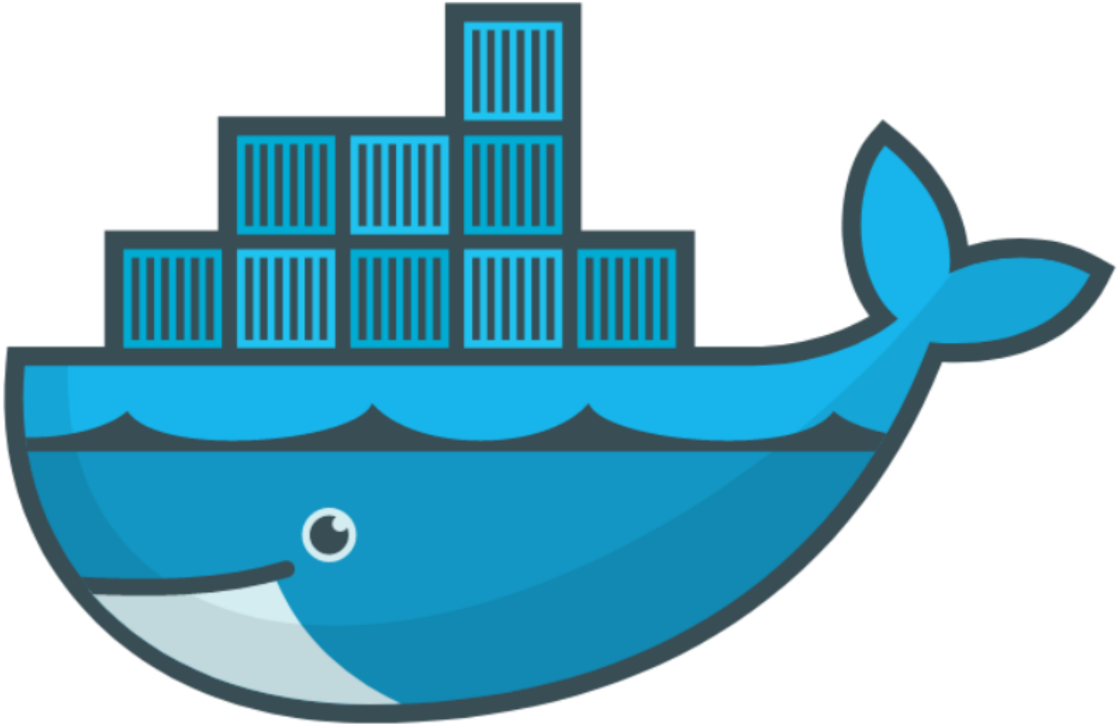
- Requirements: You need an Github and TIRA.io account
- Overview
  - You develop your system on the training and/or the validation data
  - The developed system(s) are then submitted to the leaderboard

Step-by-Step Guide

- Step 1: Develop your System(s) on the training data (we will use a BM25 baseline in the following)
- Step 2: Connect your TIRA account to your git repository
- Step 3: Upload your Code
- Step 4: Build your Docker image via Github Actions
- Step 5: Execute your Approach in TIRA

Docker Tutorial

Docker Basics



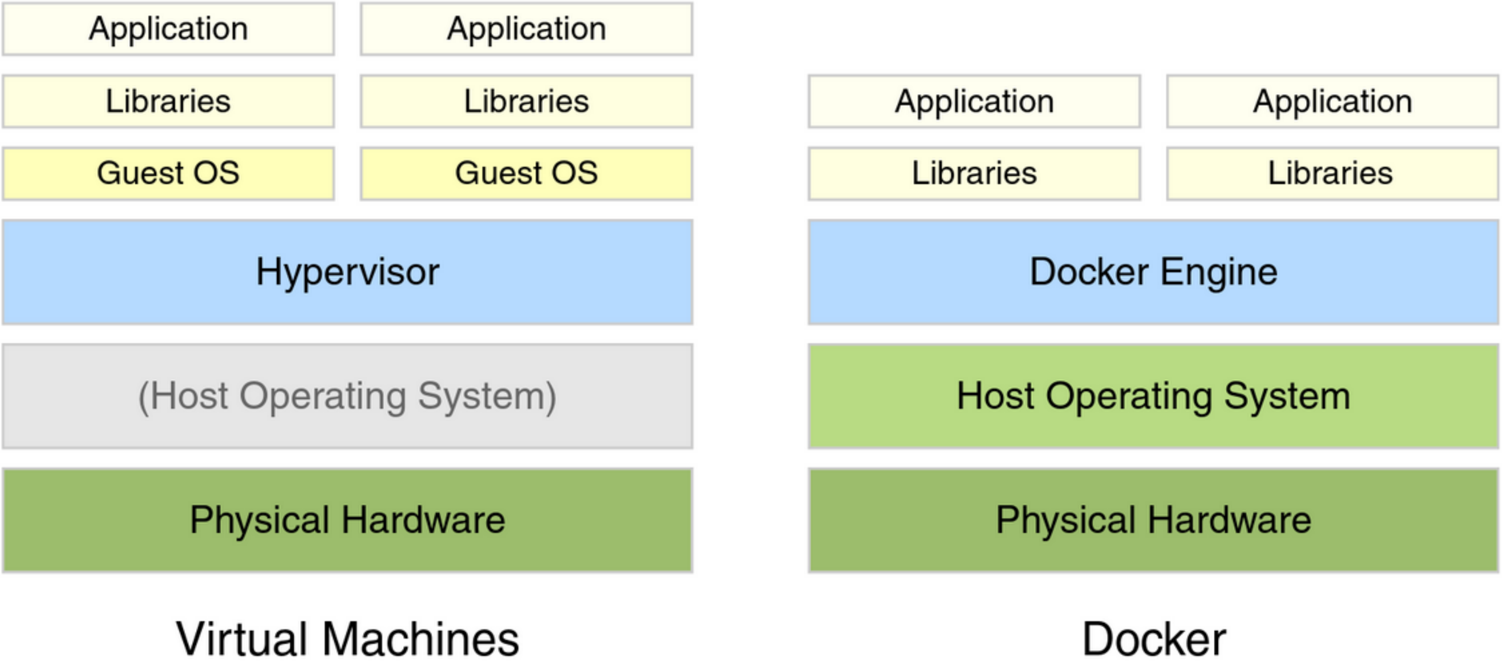
- Goal: If you can start/stop your jupyter notebook everything is fine
- https://docs.docker.com/get-docker/
- We will provide all required commands

Docker Tutorial

Use Cases for Docker

- Run guest systems as containers
- Shipping and running micro services as portable images
- Exploring and experimenting with new technologies
- Encapsulation mechanism to deploy applications in parallel without conflicts

Virtual Machines vs Docker



Docker Tutorial

Example Docker Commands

- Visit hub.docker.com
- We use the `bash` and `webis/tira-ir-starter-pyterrier:0.0.4-base` images

Docker Tutorial

Example Docker Commands

- Visit hub.docker.com
- We use the `bash` and `webis/tira-ir-starter-pyterrier:0.0.4-base` images

Bash Image

```
docker run --rm -ti bash
```

- `--rm`: Remove container after completion
- `-ti`: Attach stdin and stdout
- ToDo: Run above comand without `-ti`. What happens?
- ToDo: Write text to some file, restart the container. What happens?

Docker Tutorial

Example Docker Commands

- Visit hub.docker.com
- We use the `bash` and `webis/tira-ir-starter-pyterrier:0.0.4-base` images

Bash Image

```
docker run --rm -ti bash
```

- `--rm`: Remove container after completion
- `-ti`: Attach stdin and stdout
- ToDo: Run above comand without `-ti`. What happens?
- ToDo: Write text to some file, restart the container. What happens?

Bash Image With Volume Mounts

```
docker run --rm -ti -v $PWD:/bla bash
```

- `-v <HOST_PATH>:<CONTAINER_PATH>`: Mount the directory `<HOST_PATH>` on the system to the directory `<CONTAINER_PATH>` within the container
- ToDo: Write text to some file so that it is persistent.

Docker Tutorial

Jupyter Notebook and PyTerrier Pipelines with Docker

- We have preprared a Docker image with all reasonable libraries/frameworks preinstalled

```
docker run --rm -ti -p 8888:8888 \
  -v $PWD:/workspace/ \
  webis/tira-ir-starter-pyterrier:0.0.4-base \
  jupyter notebook --allow-root --ip 0.0.0.0
```

- `-p <HOST_PORT>:<CONTAINER_PORT>`: Map port `<HOST_PORT>` on the system to the port `<CONTAINER_PORT>` within the container
- `jupyter notebook --allow-root --ip 0.0.0.0`: The command executed in the container. This command starts a Jupyter notebook.
- ToDo: Play around with Python in the notebook for a few minutes

Docker Tutorial

Now We repeat this with Dev-Containers in VS Code

If we have time, we can see the same steps in a Dev-Container.

Outlook

We hopefully can make a set of IR-Components available in two weeks

See: `https://www.tira.io/tirex/components`