Deployment Infrastructure

http://d3s.mff.cuni.cz

Pavel Parízek
parizek@d3s.mff.cuni.cz
Motivation

- Presenting your software in expected settings
  - To whom: labs tutor, lecturer (supervisor), reviewer (for bachelor thesis)

- Running your application somewhere in cloud
  - Where just the image has to be uploaded

- Shipping the software (package) for someone to try it without difficulties
  - Evaluation committees at various competitions
Docker

- Platform for deploying (sharing) and running applications in a controlled environment
- Runs on Windows, Linux, MacOS

- Key features
  - Isolating applications from the underlying host system and each other

- Web: https://www.docker.com/
Docker – resources

• Installing
  - https://docs.docker.com/desktop/install/windows-install/
  - https://docs.docker.com/desktop/install/linux-install/

• Basic usage
  - https://docs.docker.com/guides/get-started/
  - https://docs.docker.com/guides/walkthroughs/what-is-a-container/
  - https://docs.docker.com/guides/walkthroughs/run-a-container/
  - https://docs.docker.com/build/building/packaging/

• Documentation
  - https://docs.docker.com/manuals/
  - https://docs.docker.com/get-started/overview/
Docker – important concepts

- **Image**
  - Static description (template)

- **Container**
  - Runtime instance (process)
Docker – architecture

- Daemon dockerd

- Available clients
  - Command-line: docker
  - GUI: Docker Desktop
Docker – usage

- Requirements
  - WSL on Windows 10

- Main steps
  1. Prepare or get some image (template)
  2. Run the container (with application)

- Inspect the containers (files, ...)
Docker – custom images

- **Image**
  - Template for creating Docker containers
  - Based on another existing image

- **Dockerfile**
  - Specifies how to create the image
    - List of all necessary packages to be installed
    - Building steps (macros in a special notation)

- **Relevant commands**
  - `docker build -t <name_of_image> <directory with the Dockerfile>`
    - Example: `docker build -t mydb:pp`.
  - `docker run <image> [<command>]`
Docker – custom images

- Useful macros to be used in Dockerfile
  - `FROM <base image>`
  - `ENV <variable>=<value>`
  - `WORKDIR <work dir inside the image>`
  - `COPY <work dir on host> <work dir in the image>`
  - `RUN <command that helps to build your app>`
  - `CMD ["<command to run your app within the container>", "<arg 1>", ..., "<arg N>"]`
  - `EXPOSE <network port number>`
Docker – registry

- Purpose: repository (storage) for images

- Docker Hub
  - Global public registry (free)
  - [https://hub.docker.com/](https://hub.docker.com/)

- Relevant commands
  - `docker pull`
  - `docker push`
Docker – advanced topics

• Docker Compose
  ▪ Starting multiple containers together
  ▪ Use case: one container for each application
    • Example: database + auth service + web frontend
  ▪ Resources
    • https://docs.docker.com/compose/
    • https://docs.docker.com/compose/compose-application-model/
    • https://docs.docker.com/compose/gettingstarted/

• How it works
  ▪ Virtualization at the OS level
Exercise

- Create the Docker image for your software
  - Essential part: Dockerfile

- This represents also the homework assignment for Docker
  - Start now (in the lab) and continue later

- Hint: use the available resources
  - See links to guides on the web
Other tools

- **Package managers (Linux distributions)**

- **Building from sources (Linux/Unix, Java)**
  - GNU Autotools: `./configure; make; make install`

- **Installers (Windows)**

- **Virtualization**
  - Oracle VM VirtualBox ([https://www.virtualbox.org/](https://www.virtualbox.org/))
Homework

• Assignment
  - [Link](http://d3s.mff.cuni.cz/files/teaching/nswi154/ukoly)

• Deadline
  - 31.3.2024