How To Dev - Workshop

Ben Sivan

Background

This workshop was created to give project students in Celleste-Bio the necessary background to get started with the project. It has been opened for all Trendlines startup hub as a knowledge share gesture.

Fill free to interrupt in any moment during the day, it is supposed to be an active session and most of the time it will be hands-on keyboard.

The workshop is focuses on giving a basic hand on confidence on using the tools and how to learn.

I will not get into advance topics and don't want to diverge to unrelated topics like history and such…

Intro

I called the workshop 'How To Dev' because it is mainly targeted for the software development audience, yet I do think that the ideas behind it is greater and can contribute to wider audience.

- dev short for developer.

This workshop revolves around one common theme and three major enablers to rich the ultimate goal.
The theme is: "Software is messy, especially when distributed"
No one wants to get a bug and hear "but it works on my machine" and the greatest compliment on software is "just works".

The three enablers are:
- Linux
- Git
- Docker

What brakes programs?
Code and Environment

Examples of programs can also be a word document that it's design doesn't look as intended.

To run the exact same program each time we will use version control system.
For reproducible environment we will use containerized environment for encapsulation.

First lets talk about the interactions between those three enablers, and then we will dive deep into each one of them.
- All open source tools.
- Docker containerize Linux environment.
- We define the environment specification of Docker in a build script called Dockerfile and we version the file in git.
- We are installing git within the Docker container to import our work.
- The developers of Linux are using Git to develop it.

- break?

Linux

- Fill at home in terminal.
- Most in use for computations -> most effective in resources.
- Linux's philosophy: do one thing and do it well.
- Open source, transparency.
- Common utilities.

Directory structure:
/ root directory for the file system
/bin all binary files
/home where users keeping there files
/opt optional software (packages)
/tmp temporary files
/usr users programs
/var environment variables
/etc configuration files

The operating directory structure is about the directories in the root directory.
I won't elaborate on that.

Utilities:
- ls
- cd
- cat / less
- cp / mv

- break?

Git

- Git playground.
- Local and Remote.
- Branches.
- Commands add, commit, push.
- Conflict resolution.
- break?

Docker

- Image -> Dockerfile.
- Container -> running instance.
- Commands build, run, exec.