

Getting rid of:
“It works on my
machine”

Florian Humblot
March 2023

About me

Embedded software
engineer at Blue Current.

AC chargers for electric
vehicles

Tooling / CI

So many vendors

- Code generation
- Build roots
- Specific library dependencies



How it goes wrong

Accidental updates

Version mismatches

System changes messing with your setup

Difficult deployment

Multiple projects mixing on a machine

How do we fix this?

Virtual Machines

Pros:

- Stable
- Works on all platforms

Cons:

- Resource hog
- Maintainability
- Deploy ability

Docker

Pros:

- Configuration as code
- Versioned with your project
- Fast switching per project
- Reusable

Cons:

- Learning curve
- Not supported on all platforms

NixOS / DirEnv

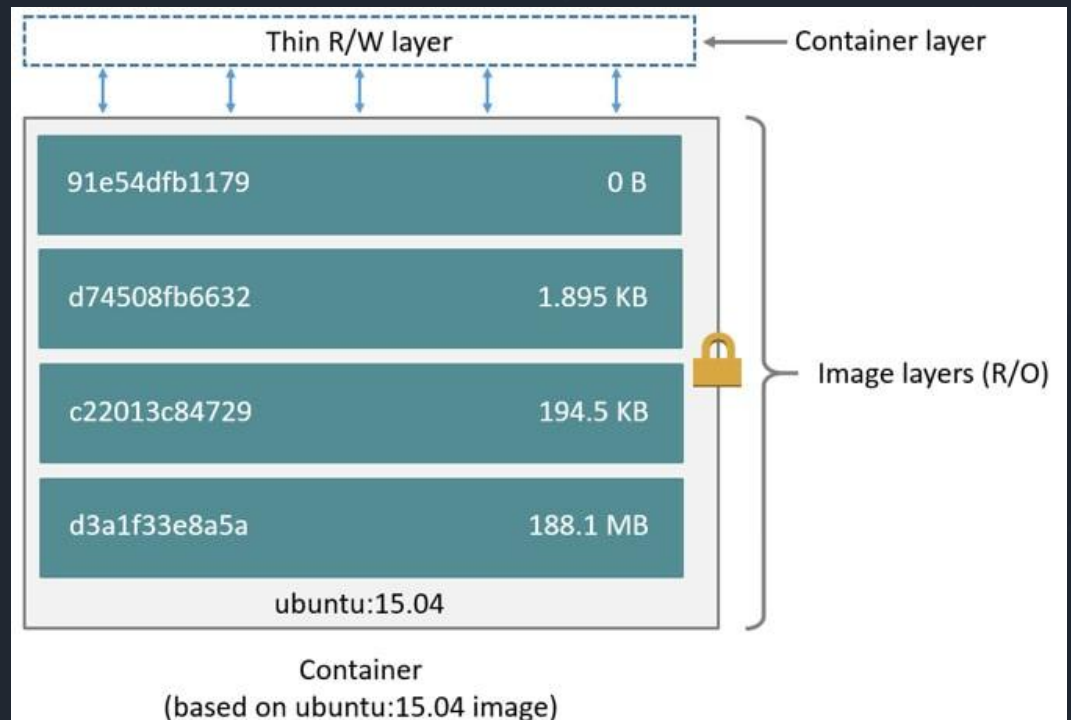
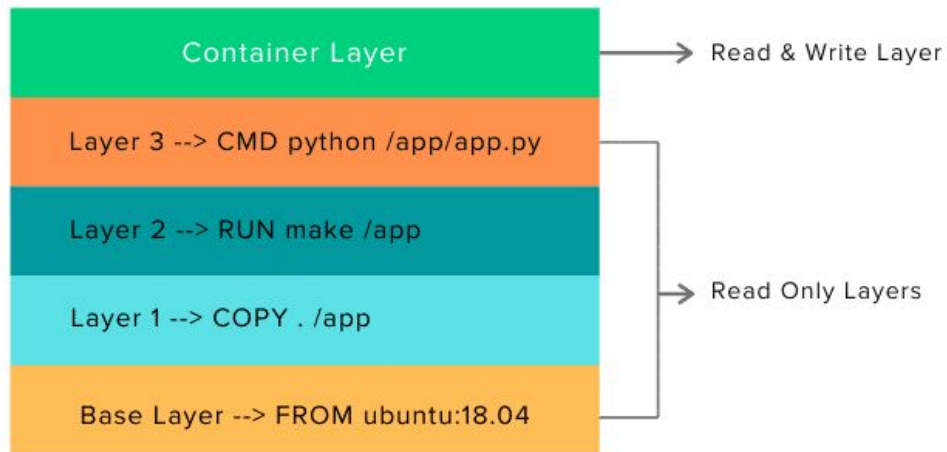
Pros:

- Made for reproducibility
- Great community
- Configuration as code
- Versioned with your project

Cons:

- Learning curve
- Integration with tools is lacking

What is docker



Sample Dockerfile

```
FROM espressif/idf:v4.4.1
```

```
RUN export DEBIAN_FRONTEND=noninteractive && \
    echo "force-unsafe-io" > /etc/dpkg/dpkg.cfg.d/02apt-speedup && \
    echo "Acquire::http {No-Cache=True;;}" > /etc/apt/apt.conf.d/no-cache && \
    apt -y update && apt -y upgrade && \
    apt -y install --no-install-recommends \
        libusb-1.0-0 \
        vim \
        curl \
        git \
        make \
        cmake \
        sudo \
        udev \
        wget \
        ninja-build \
        fd-find \
        ripgrep \
        gdb \
        python3.8-venv \
        doxygen \
    && rm -rf /var/lib/apt/lists/*
```

```
RUN ln -s /usr/bin/fdfind /usr/bin/fd
```

```
RUN echo "source /opt/esp/idf/export.sh" >> /root/.bashrc
```

```
RUN echo "source /opt/esp/idf/export.sh" >> /home/vscode/.bashrc
```

```
ENV _PIPELINE_CMAKE_PATH="/usr/local/lib/python3.6/dist-packages/cmake/data/bin"
```

```
RUN groupadd --gid 1000 vscode && \
    useradd --uid 1000 --gid 1000 -G plugdev,dialout --shell /bin/bash -m vscode && \
    echo "vscode ALL=(ALL:ALL) NOPASSWD:ALL" > /etc/sudoers.d/vscode
```

```
COPY entrypoint.sh /opt/entrypoint.sh
```

```
RUN chmod +x /opt/entrypoint.sh
```

```
ENTRYPOINT ["/opt/entrypoint.sh"]
```

```
CMD ["/bin/bash", "-c", "sleep infinity"]
```

Adding STM32 to the image

```
● ● ●  
RUN mkdir -p /opt/st/  
COPY st-stm32cubeide_1.10.1_12716_20220707_0928_amd64.sh /opt/st/install.sh  
RUN chmod +x /opt/st/install.sh \  
    && /opt/st/install.sh --target /opt/st/cubeide --noexec \  
    && rm -f /opt/st/install.sh \  
    && sed -i "s/egrep -q '^segger-'/false/" /opt/st/cubeide/install_as_root.sh \  
    && sed -i "s/sh \$installer/sh \$installer -- -f/" /opt/st/cubeide/install_as_root.sh \  
    && cd /opt/st/cubeide \  
    && LICENSE_ALREADY_ACCEPTED=1 ./setup.sh /opt/st/stm32cubeide_1.10.1  
  
ENV  
PATH="${PATH}:/opt/st/stm32cubeide_1.10.1:/opt/st/stm32cubeide_1.10.1/plugins/com.st.stm32cube.ide.mcu.  
externaltools.gnu-tools-for-stm32.10.3-2021.10.linux64_1.0.0.202111181127/tools/bin/"
```


Adding clang(-tools)



```
RUN wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key | apt-key add -  
RUN echo 'deb http://apt.llvm.org/focal/ llvm-toolchain-focal-15 main\ndeb-src  
http://apt.llvm.org/focal/ llvm-toolchain-focal-15 main' | tee -a /etc/apt/sources.list.d/llvm-15.list  
  
RUN apt-get update \  
  && apt-get clean \  
  && apt-get install -y --no-install-recommends \  
    clang-format-15 \  
    clang-tidy-15 \  
    clangd-15 \  
  && apt-get autoremove -y \  
  && rm -rf /var/lib/apt/lists/* \  
  && ln -s /usr/bin/clang-format-15 /usr/bin/clang-format \  
  && ln -s /usr/bin/clang-tidy-15 /usr/bin/clang-tidy \  
  && ln -s /usr/bin/clangd-15 /usr/bin/clangd
```

Choose wisely

Dockerfile per project

Dockerfile per
microcontroller/platform

Dockerfile per team

Editor integration

Can still use GUI

VSCode

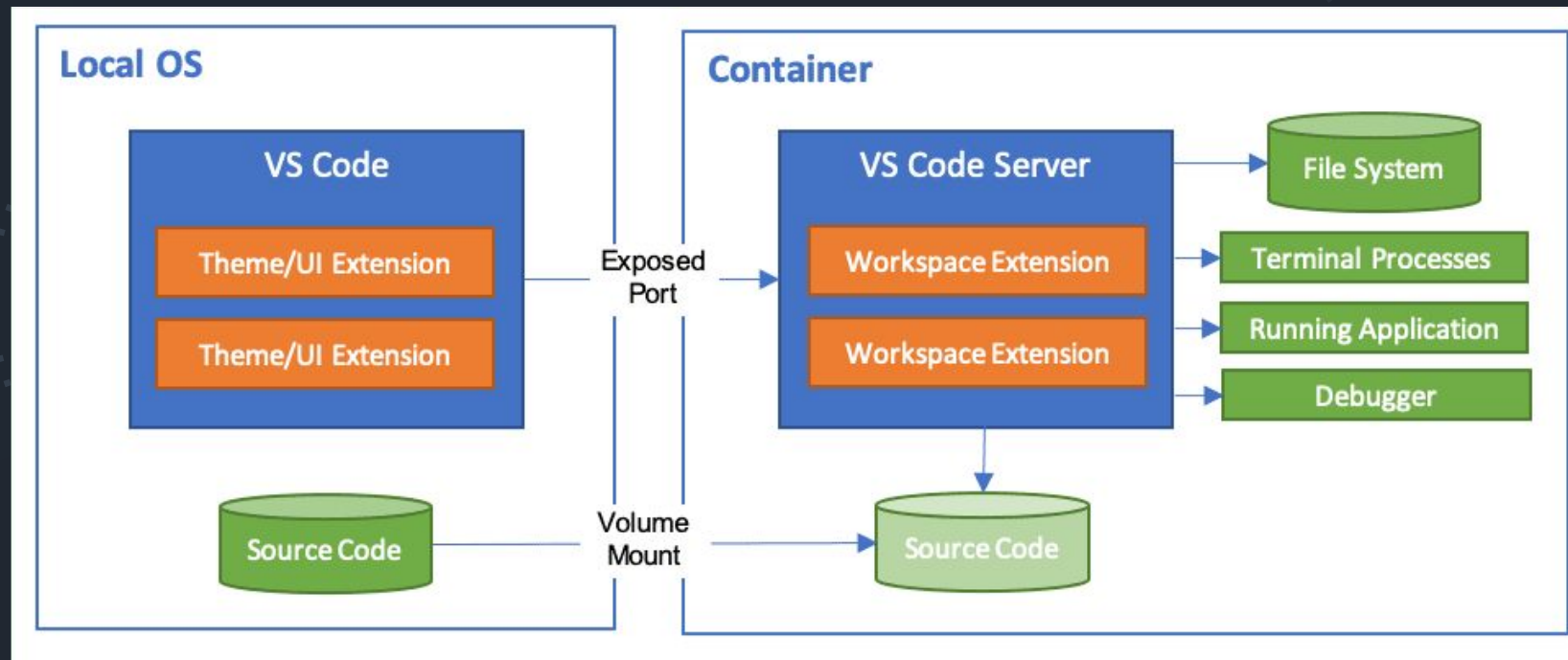
CLion

Something else?

Questions



VS Code integration



Sample devcontainer.json

```
// For format details, see https://aka.ms/devcontainer.json. For config options, see the README at:
// https://github.com/microsoft/vscode-dev-containers/tree/v0.183.0/containers/ubuntu
{
  "name": "MyProject",
  "image": "florianhumblotbc/devcontainer:v1.8",
  // Add the IDs of extensions you want installed when the container is created.
  "extensions": [
    "espressif.esp-idf-extension",
    "marus25.cortex-debug",
    "swellaby.workspace-config-plus",
    "xaver.clang-format",
    "llvm-vs-code-extensions.vscode-clangd"
  ],
  "mounts": [
    "source=/tmp/.X11-unix,target=/tmp/.X11-unix,type=bind",
    "source=${localEnv:HOME}${localEnv:USERPROFILE}/.ssh,target=/home/developer/.ssh,type=bind"
  ],
  "containerEnv": {
    "DISPLAY": ":0"
  },
  "settings": {
    "terminal.integrated.defaultProfile.linux": "bash",
    "idf.espIdfPath": "/opt/esp/idf",
    "idf.customExtraPaths": "",
    "idf.pythonBinPath": "/opt/esp/python_env/idf4.4_py3.8_env/bin/python",
    "idf.toolsPath": "/opt/esp/tools",
    "idf.gitPath": "/usr/bin/git"
  },
  "remoteUser": "vscode",
  "overrideCommand": false,
  "runArgs": [
    "--privileged"
  ]
}
```