Build before installing

Zephyr Dev Environment using Codespaces

Mike Szczys
golioth.io

golioth

What if you could build Zephyr apps without any setup?

golioth

Mike Szczys

https://chaos.social/@szczys

- Firmware Engineer at Golioth
- 15 years of firmware experience
- Previously: Editor in Chief of Hackaday

golioth

Golioth is an IoT Cloud Company

- We make it easy to connect MCUs to the internet
- Device Management
  - OTA, fleet settings, RPC, remote logging
- Data Routing
  - Time-series and stateful data
  - Cloud integrations with numerous cloud platforms and database hosts
  - REST API, Webhook, Websockets

golioth

Challenge:

- Our device SDK functions as a Zephyr module
- For a potential user to validate Golioth, they need to be able to build Zephyr applications

golioth

How we can use Codespaces

- Free monthly Zephyr training
- Enable prospective customers to build code samples without waiting

Built on Development Containers

- Open standard easily added to repo
- Can be run locally (and easily) via VS Code

golioth

Codespaces Demo:

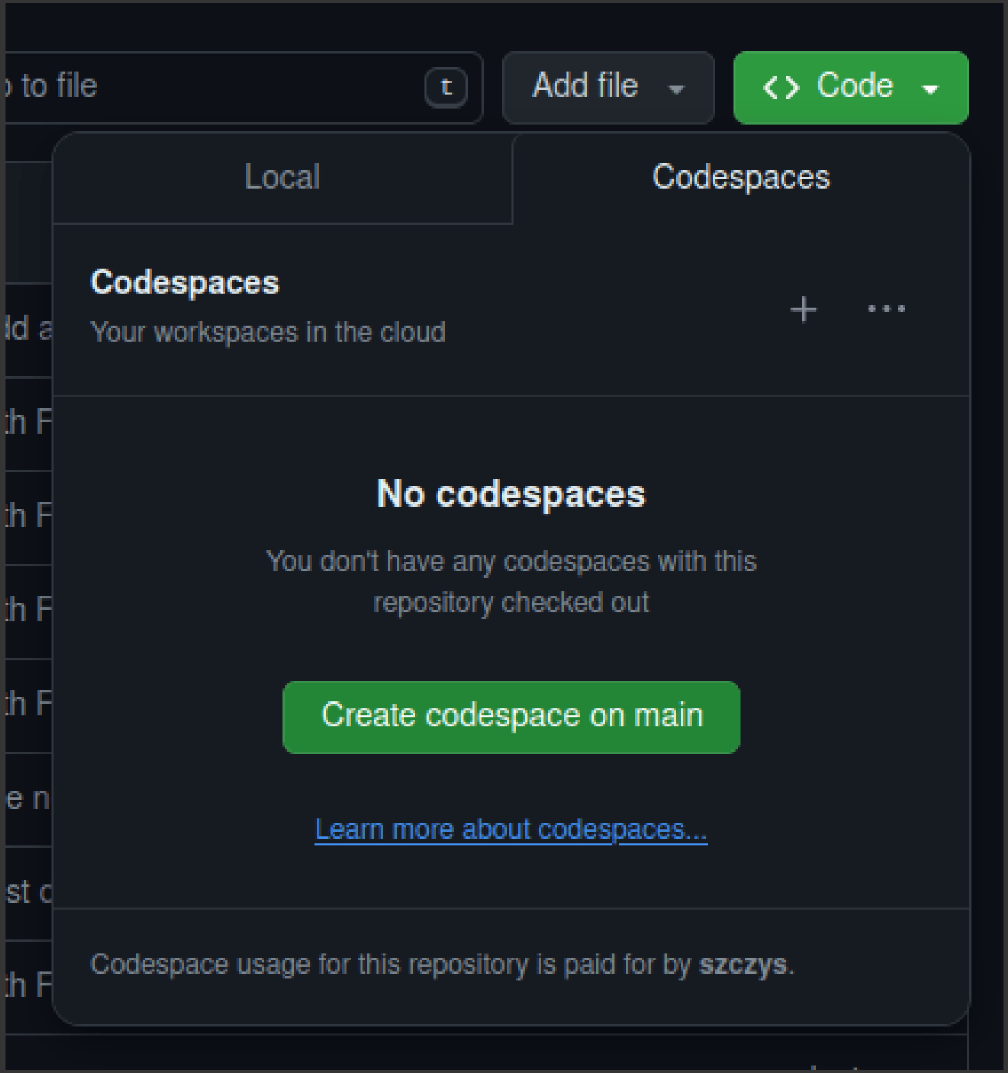
Building a Zephyr app within 90 seconds during free Golioth Zephyr trainning

(Try it yourself: https://github.com/golioth/zephyr-training)

golioth

Demo: Big Green 'Code'Button

GitHub repos that have devcontainers
have a Codespaces tab



Codespaces

Your workspaces in the cloud

No codespaces

You don't have any codespaces with this repository checked out

Create codespace on main

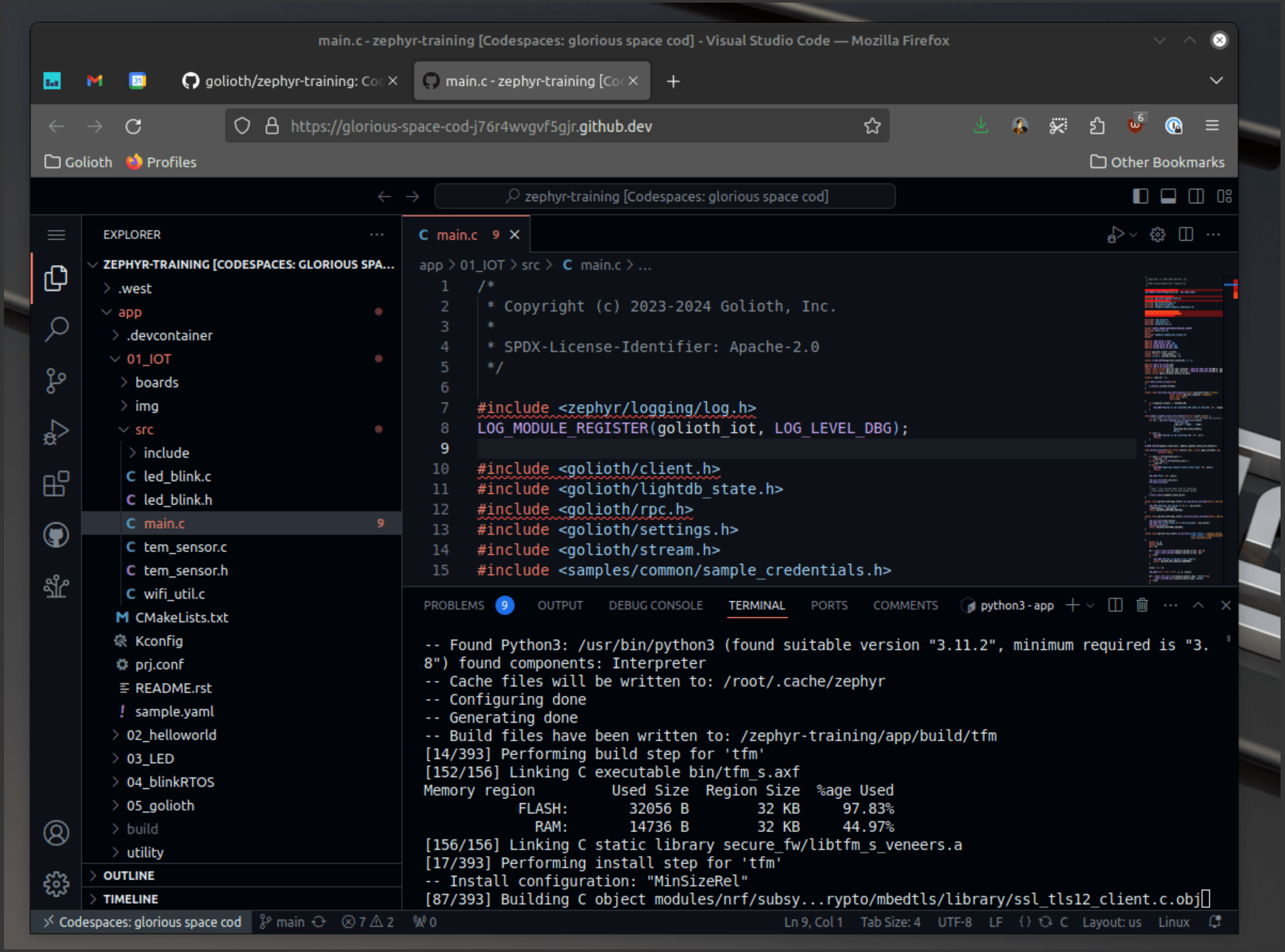
Learn more about codespaces...

Codespace usage for this repository is paid for by **szczys**.

golioth

Demo: VS Code in a Browser

You get the VS Code with a complete build environment



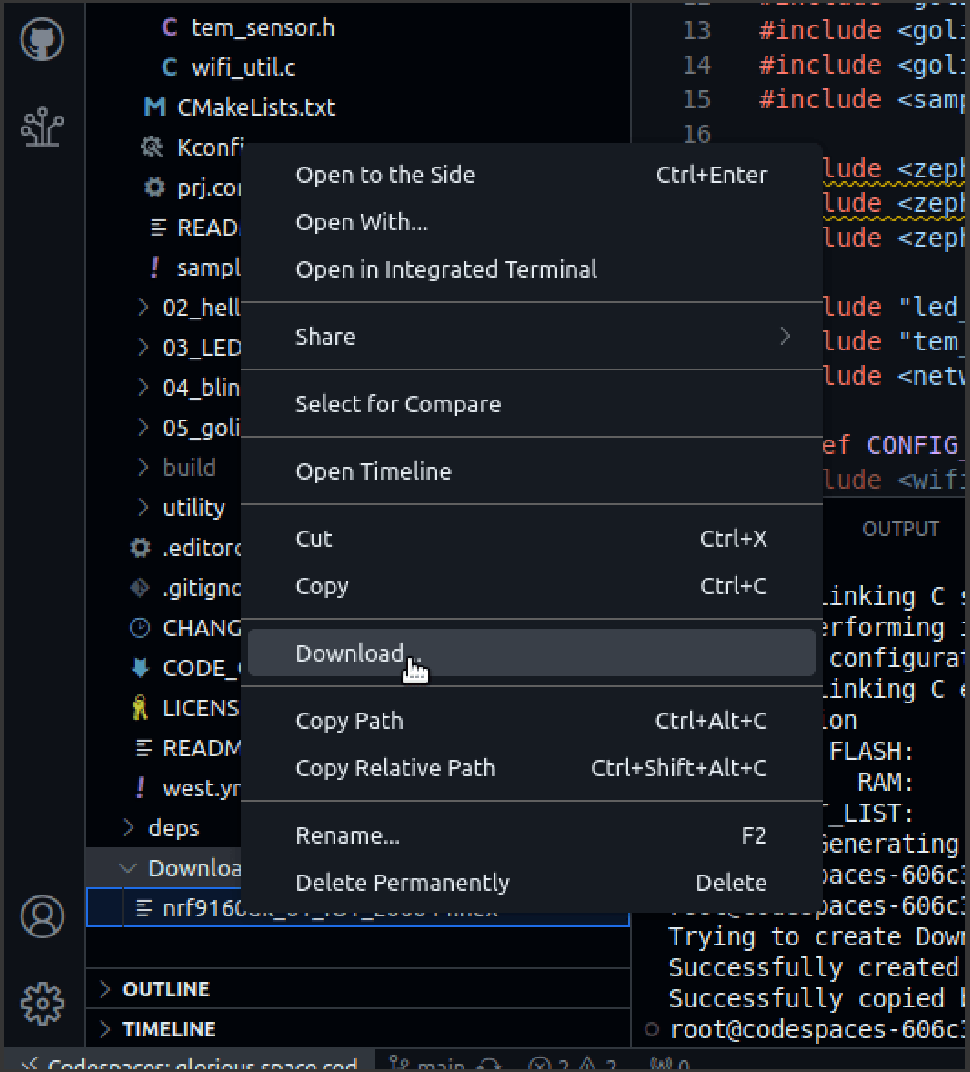
```c
/*
 * Copyright (c) 2023-2024 Golioth, Inc.
 *
 * SPDX-License-Identifier: Apache-2.0
 */

#include <zephyr/logging/log.h>
LOG_MODULE_REGISTER(golioth_iot, LOG_LEVEL_DBG);

#include <golioth/client.h>
#include <golioth/lightdb_state.h>
#include <golioth/rpc.h>
#include <golioth/settings.h>
#include <golioth/stream.h>
#include <samples/common/sample_credentials.h>
```

```
-- Found Python3: /usr/bin/python3 (found suitable version "3.11.2", minimum required is "3.
8") found components: Interpreter
-- Cache files will be written to: /root/.cache/zephyr
-- Configuring done
-- Generating done
-- Build files have been written to: /zephyr-training/app/build/tfm
[14/393] Performing build step for 'tfm'
[152/156] Linking C executable bin/tfm_s.axf
Memory region         Used Size  Region Size  %age Used
           FLASH:       32056 B        32 KB     97.83%
             RAM:       14736 B        32 KB     44.97%
[156/156] Linking C static library secure_fw/libtfm_s_veneers.a
[17/393] Performing install step for 'tfm'
-- Install configuration: "MinSizeRel"
[87/393] Building C object modules/nrf/subsy...rypto/mbedtls/library/ssl_tls12_client.c.obj
```

golioth

Demo: Cloud != USB

Binaries must be downloaded and flashed locally



```
#include <gol
#include <gol
#include <samp

lude <zeph
lude <zeph
lude <zeph

lude "led
lude "tem
lude <netv

ef CONFIG
lude <wif
```

```
inking C
erforming
configura
inking C
on
FLASH:
RAM:
_LIST:
enerating
aces-606c
aces-606c
Trying to create Dow
Successfully created
Successfully copied
root@codespaces-606c
```

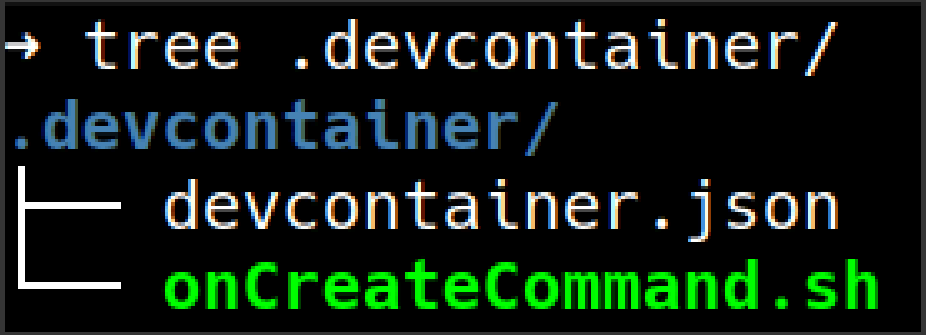
golioth

Dev Containers

Docker + JSON Config

golioth

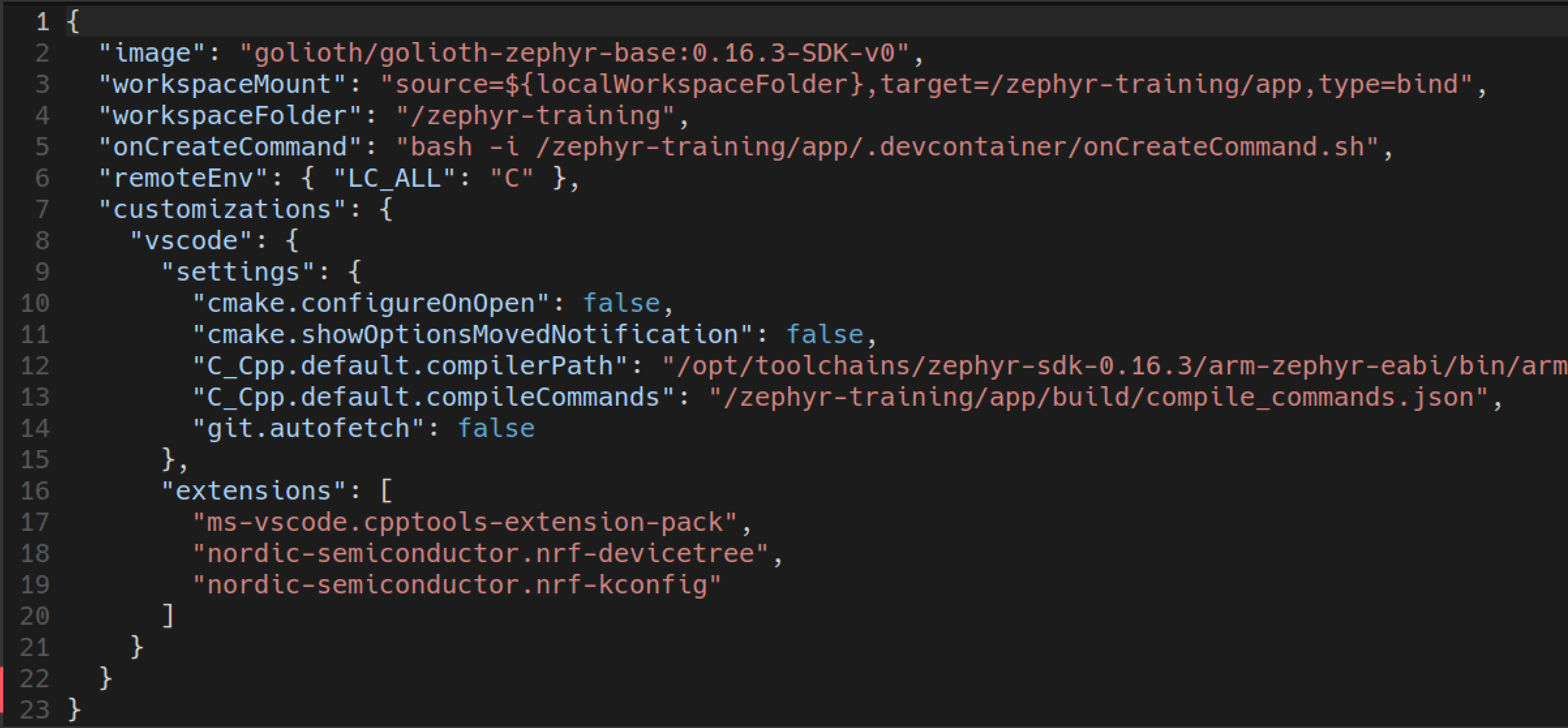
Adding a Dev Container


```
→ tree .devcontainer/
.devcontainer/
├── devcontainer.json
└── onCreateCommand.sh
```

- Required: .devcontainer subdirectory
- Required: devcontainer.json
- Optional: helper files (we'll get to that)

devcontainer.json

The entirety of what was shown in the demo



```json
{
  "image": "golioth/golioth-zephyr-base:0.16.3-SDK-v0",
  "workspaceMount": "source=${localWorkspaceFolder},target=/zephyr-training/app,type=bind",
  "workspaceFolder": "/zephyr-training",
  "onCreateCommand": "bash -i /zephyr-training/app/.devcontainer/onCreateCommand.sh",
  "remoteEnv": { "LC_ALL": "C" },
  "customizations": {
    "vscode": {
      "settings": {
        "cmake.configureOnOpen": false,
        "cmake.showOptionsMovedNotification": false,
        "C_Cpp.default.compilerPath": "/opt/toolchains/zephyr-sdk-0.16.3/arm-zephyr-eabi/bin/arm
        "C_Cpp.default.compileCommands": "/zephyr-training/app/build/compile_commands.json",
        "git.autofetch": false
      },
      "extensions": [
        "ms-vscode.cpptools-extension-pack",
        "nordic-semiconductor.nrf-devicetree",
        "nordic-semiconductor.nrf-kconfig"
      ]
    }
  }
}
```

Choose Docker image and mounts

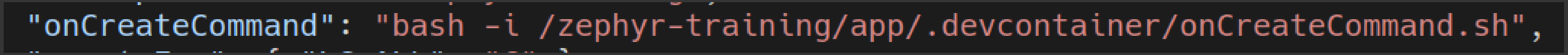
```
"image": "golioth/golioth-zephyr-base:0.16.3-SDK-v0",
"workspaceMount": "source=${localWorkspaceFolder},target=/zephyr-training/app,type=bind",
"workspaceFolder": "/zephyr-training",
```

- Debian-based image with bare minimum of Zephyr toolchain
- Optional:
  - Set a mount point for the parent repository
  - Set default path for the workspace

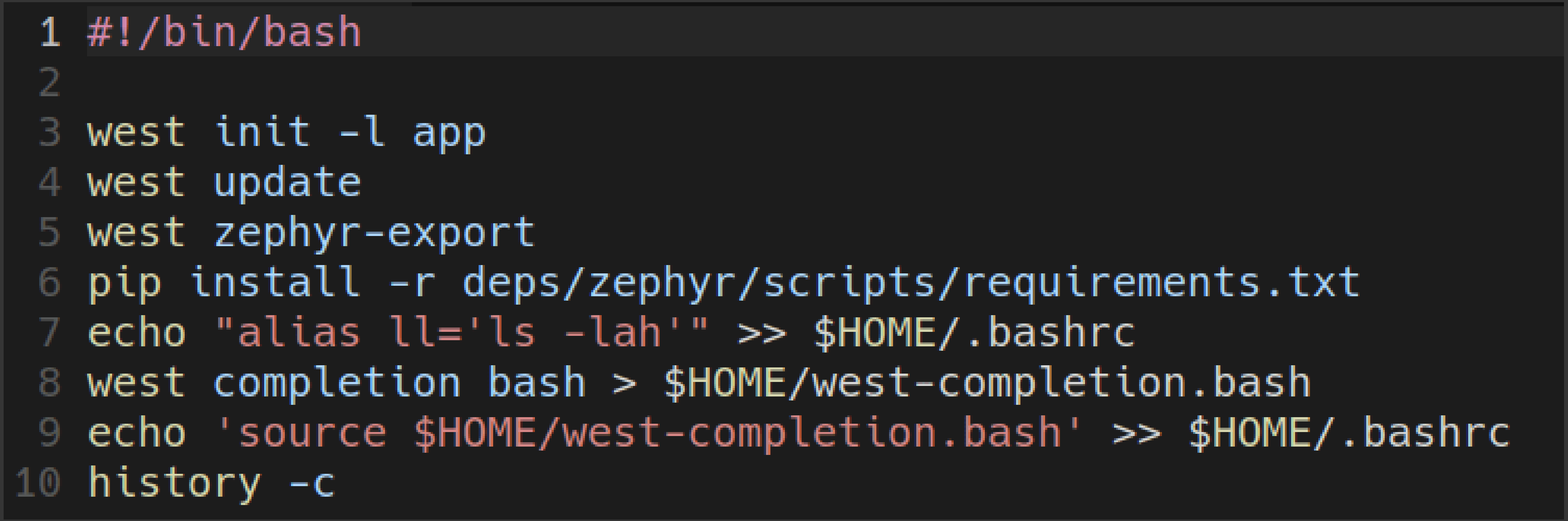
golioth

Call Script to Complete Build

Perform the repo-setup steps



```
"onCreateCommand": "bash -i /zephyr-training/app/.devcontainer/onCreateCommand.sh",
```



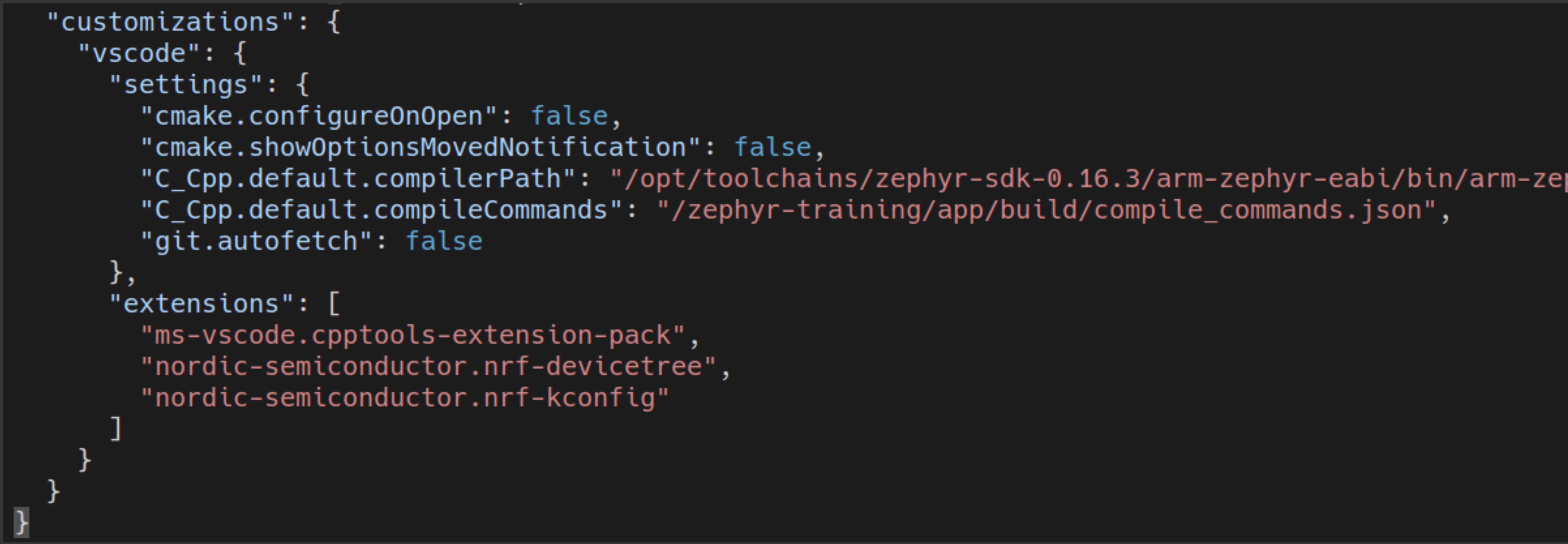
```bash
#!/bin/bash

west init -l app
west update
west zephyr-export
pip install -r deps/zephyr/scripts/requirements.txt
echo "alias ll='ls -lah'" >> $HOME/.bashrc
west completion bash > $HOME/west-completion.bash
echo 'source $HOME/west-completion.bash' >> $HOME/.bashrc
history -c
```

golioth

This is a manifest repository, so west update pulls in all dependencies (like the Zephyr tree).

Customize VS Code Itself

Install extensions and configure settings

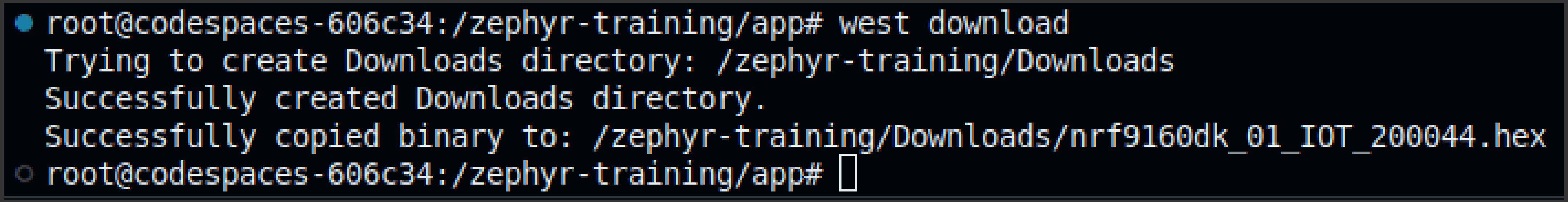


```
  "customizations": {
    "vscode": {
      "settings": {
        "cmake.configureOnOpen": false,
        "cmake.showOptionsMovedNotification": false,
        "C_Cpp.default.compilerPath": "/opt/toolchains/zephyr-sdk-0.16.3/arm-zephyr-eabi/bin/arm-ze
        "C_Cpp.default.compileCommands": "/zephyr-training/app/build/compile_commands.json",
        "git.autofetch": false
      },
      "extensions": [
        "ms-vscode.cpptools-extension-pack",
        "nordic-semiconductor.nrf-devicetree",
        "nordic-semiconductor.nrf-kconfig"
      ]
    }
  }
}
```

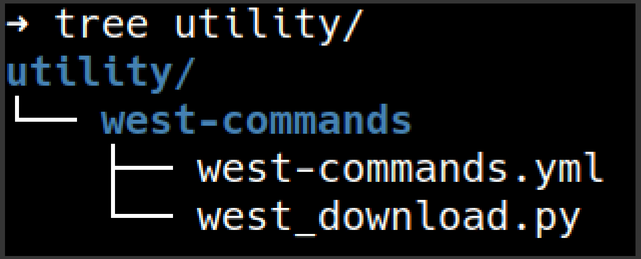
golioth

Solving for binary download

Custom west command used to rename binaries



```
● root@codespaces-606c34:/zephyr-training/app# west download
  Trying to create Downloads directory: /zephyr-training/Downloads
  Successfully created Downloads directory.
  Successfully copied binary to: /zephyr-training/Downloads/nrf9160dk_01_IOT_200044.hex
○ root@codespaces-606c34:/zephyr-training/app# 
```



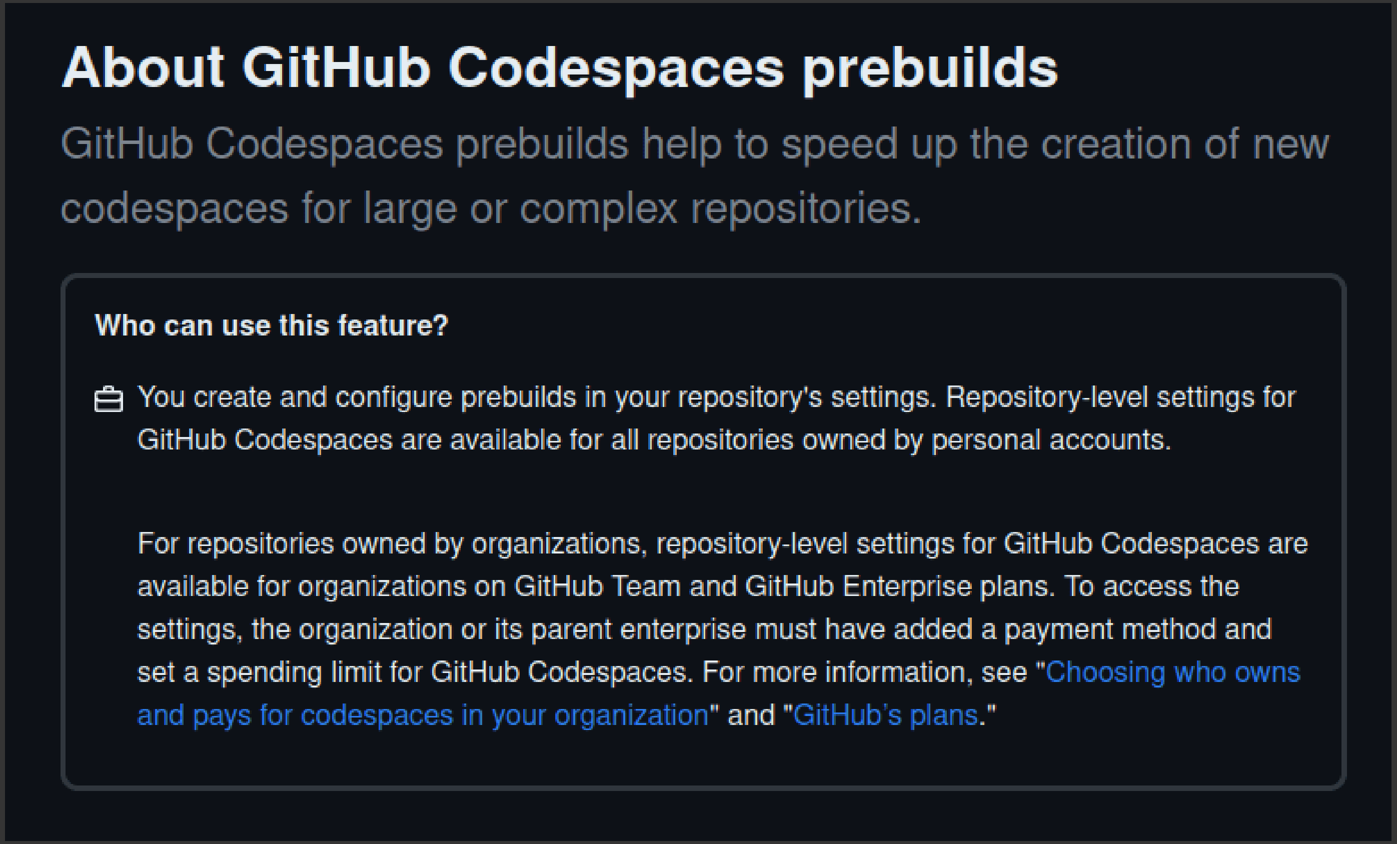
```
➜ tree utility/
utility/
└── west-commands
    ├── west-commands.yml
    └── west_download.py
```

- "West Extension" included in this repo
- west download finds correct binary, renames and timestamps, moves to a Download folder for easy location
- Local flashing tools used to program device after binary download

golioth

Pre-builds

Speed up the creation process



About GitHub Codespaces prebuilds

GitHub Codespaces prebuilds help to speed up the creation of new codespaces for large or complex repositories.

Who can use this feature?

You create and configure prebuilds in your repository's settings. Repository-level settings for GitHub Codespaces are available for all repositories owned by personal accounts.

For repositories owned by organizations, repository-level settings for GitHub Codespaces are available for organizations on GitHub Team and GitHub Enterprise plans. To access the settings, the organization or its parent enterprise must have added a payment method and set a spending limit for GitHub Codespaces. For more information, see "Choosing who owns and pays for codespaces in your organization" and "GitHub's plans."

A prebuild means the configuration step is already done.

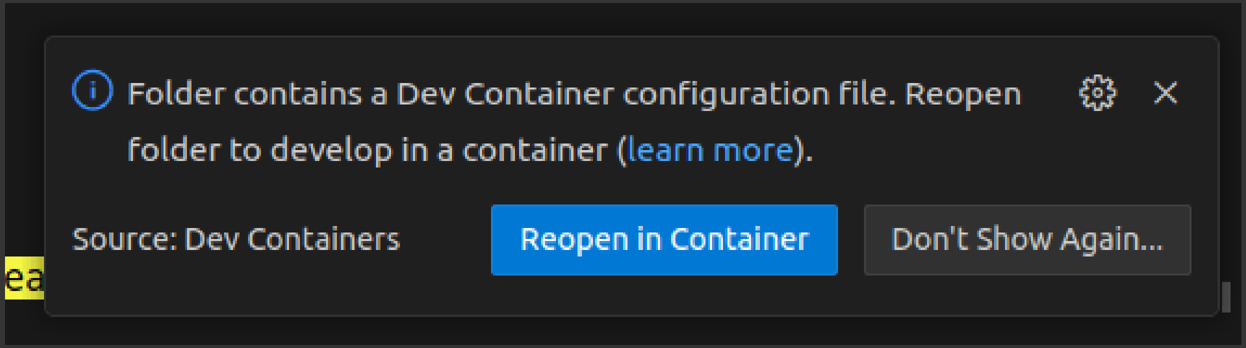
You won't have to wait for west update to clone all repos each time you start a new instance.

golioth

Running Dev Containers Locally

golioth

VS Code Knows!



Folder contains a Dev Container configuration file. Reopen folder to develop in a container (learn more).

Source: Dev Containers

VS Code recognizes the devcontainer and prompts:

- To install the dev container extension
- To build the container
- To reopen container on subsequent loads

golioth

Devcontainers in VS Code

What to know

- You must have Docker preinstalled
- Wait for download and build the first time
- Containers continue to run after you exit VS Code

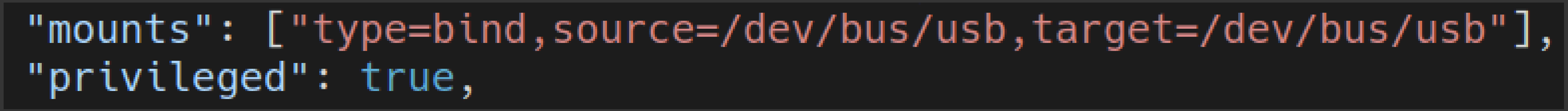
golioth

Hacking: Can I use west flash?

- Windows: No
- Mac: No
- Linux: Yes*

*if you don't mind running in priviledged mode

USB: devcontainer.json



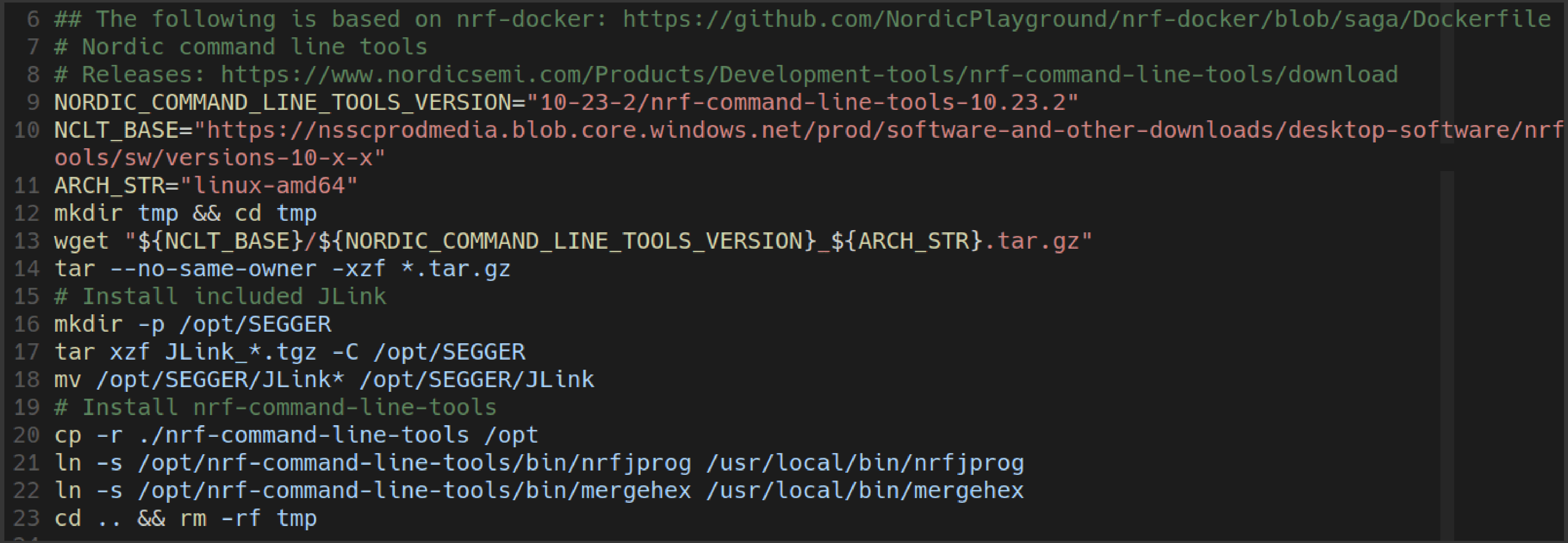
```
"mounts": ["type=bind,source=/dev/bus/usb,target=/dev/bus/usb"],
"privileged": true,
```

- Connect USB to the container
- Give the container privileges to access the ports

golioth

USB: add flashing tools

Adding SEGGER/Nordic tools to container



```
## The following is based on nrf-docker: https://github.com/NordicPlayground/nrf-docker/blob/saga/Dockerfile
# Nordic command line tools
# Releases: https://www.nordicsemi.com/Products/Development-tools/nrf-command-line-tools/download
NORDIC_COMMAND_LINE_TOOLS_VERSION="10-23-2/nrf-command-line-tools-10.23.2"
NCLT_BASE="https://nsscprodmedia.blob.core.windows.net/prod/software-and-other-downloads/desktop-software/nrf
ools/sw/versions-10-x-x"
ARCH_STR="linux-amd64"
mkdir tmp && cd tmp
wget "${NCLT_BASE}/${NORDIC_COMMAND_LINE_TOOLS_VERSION}_${ARCH_STR}.tar.gz"
tar --no-same-owner -xzf *.tar.gz
# Install included JLink
mkdir -p /opt/SEGGER
tar xzf JLink_*.tgz -C /opt/SEGGER
mv /opt/SEGGER/JLink* /opt/SEGGER/JLink
# Install nrf-command-line-tools
cp -r ./nrf-command-line-tools /opt
ln -s /opt/nrf-command-line-tools/bin/nrfjprog /usr/local/bin/nrfjprog
ln -s /opt/nrf-command-line-tools/bin/mergehex /usr/local/bin/mergehex
cd .. && rm -rf tmp
```

- Better option: add these tools to the container image

USB: not ready for Codespaces

What about WebUSB?

- WebUSB is a promising option, but...
  - Not currently supported in Codespaces
  - Only supported by Chrome Browser

golioth

Dev Containers

What are they good for?

- Online Zephyr training using CodeSpaces
- Allow protential customers to build sample code
- Deliver version controlled build environments to target specific sdk-ng toolchains

golioth

Thank you!

Try Golioth: golioth.io

References:

- Try Codespaces: https://github.com/golioth/zephyr-training/
- Free Zephyr training: https://golioth.io/training-signup
- Development Containers: https://containers.dev/
- Codespaces: https://github.com/features/codespaces

Mike Szczys
mike@golioth.io
chaos.social/@szczys

golioth