

Woche 3

Lokales Entwickeln

Modul 324

Die Idee ist:

alle haben die gleiche
Entwicklungsumgebung, selbst
auf Produktion!



Verwendete Tools

- Docker Desktop
- Docker Compose
- VS Code
- Git oder Github Desktop



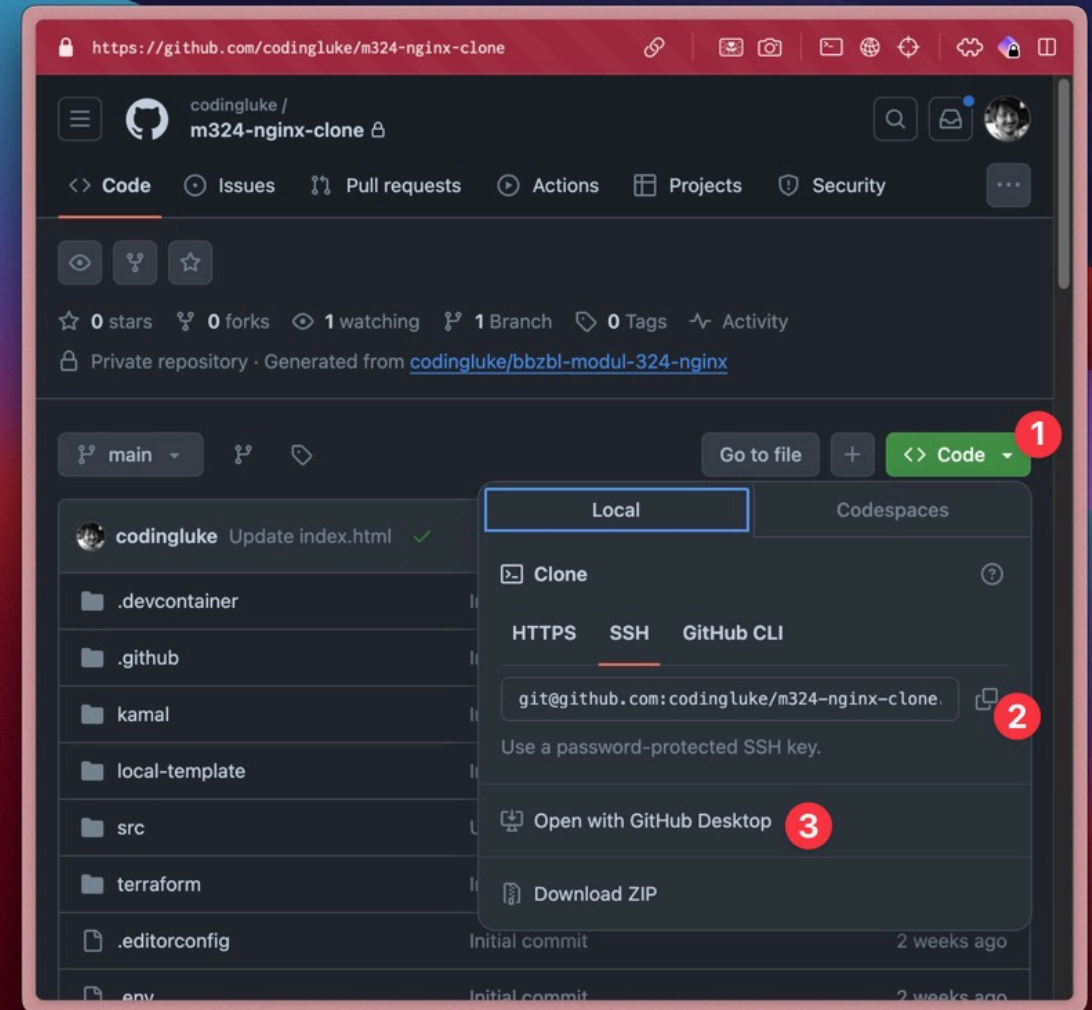
Repository klonen

1. Unter Code Das grüne Dropdown

<> Code öffnen

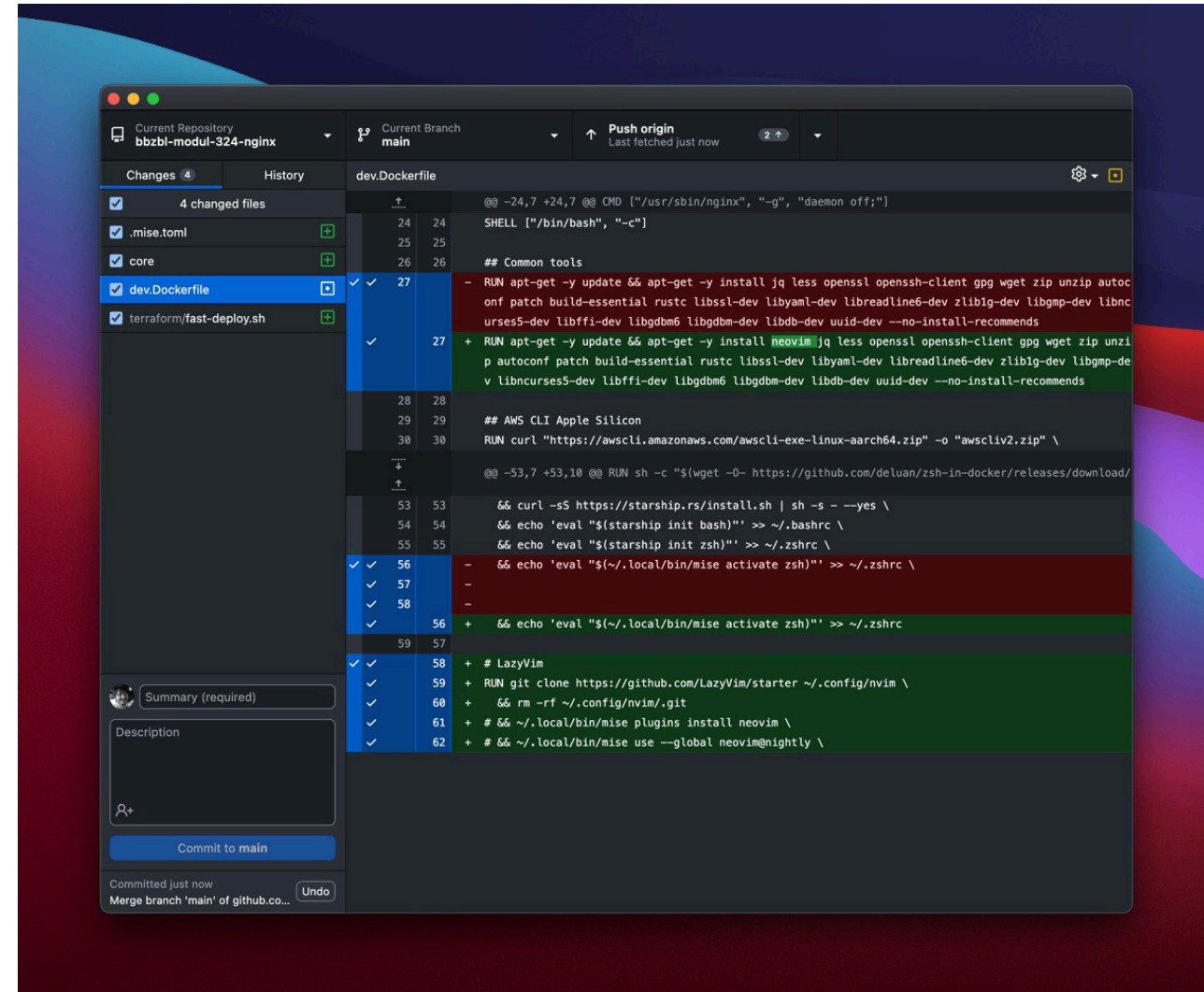
2. Die **SSH Url** kopieren

3. oder Sofern **GitHub Desktop** installiert wurde **Open with GitHub Desktop** wählen.



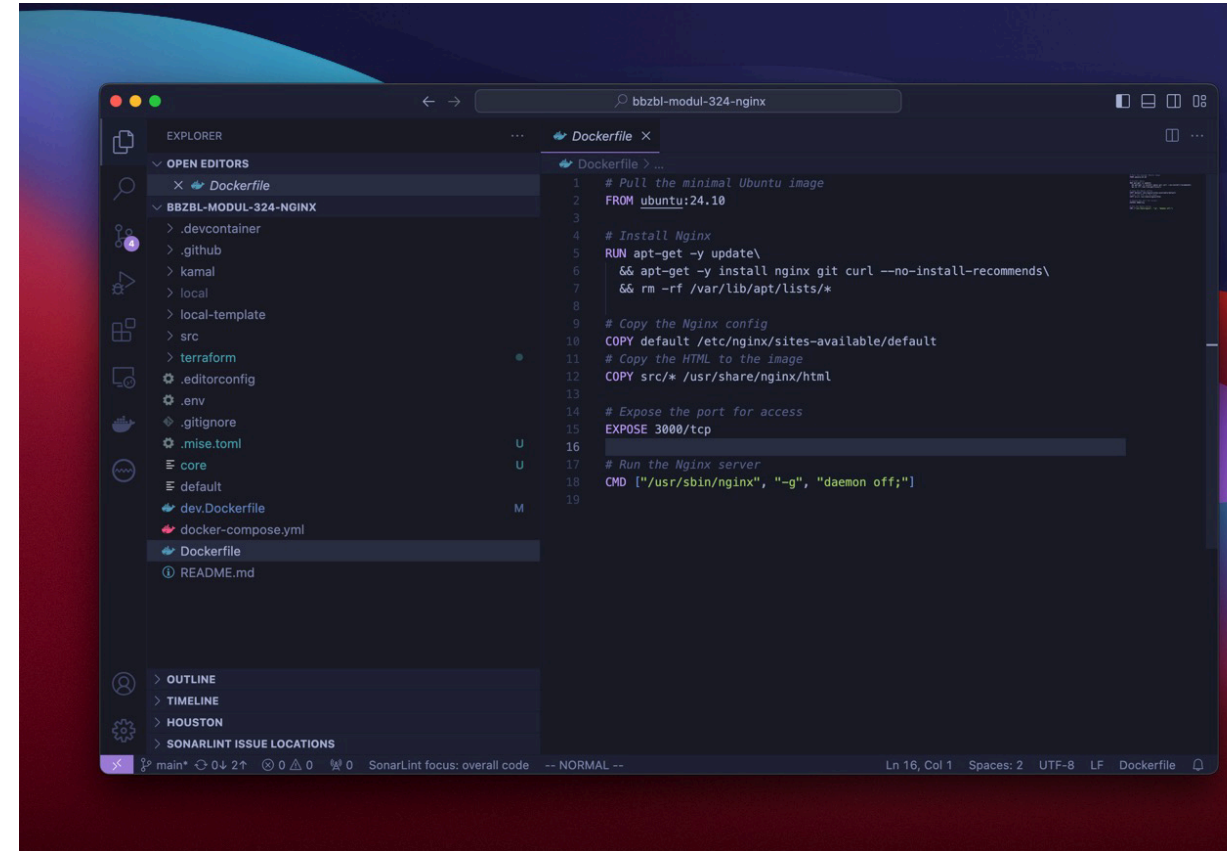
GitHub Desktop

GitHub Desktop ist ein Grafisches Tool um git repositories zu verwalten. In Github Desktop kann man mit einem GUI Comitten.



Öffnen in VS Code

1. File -> Open folder
2. Das frisch geklonte repository auswählen
3. Es sollte ungefähr wie rechts aussehen



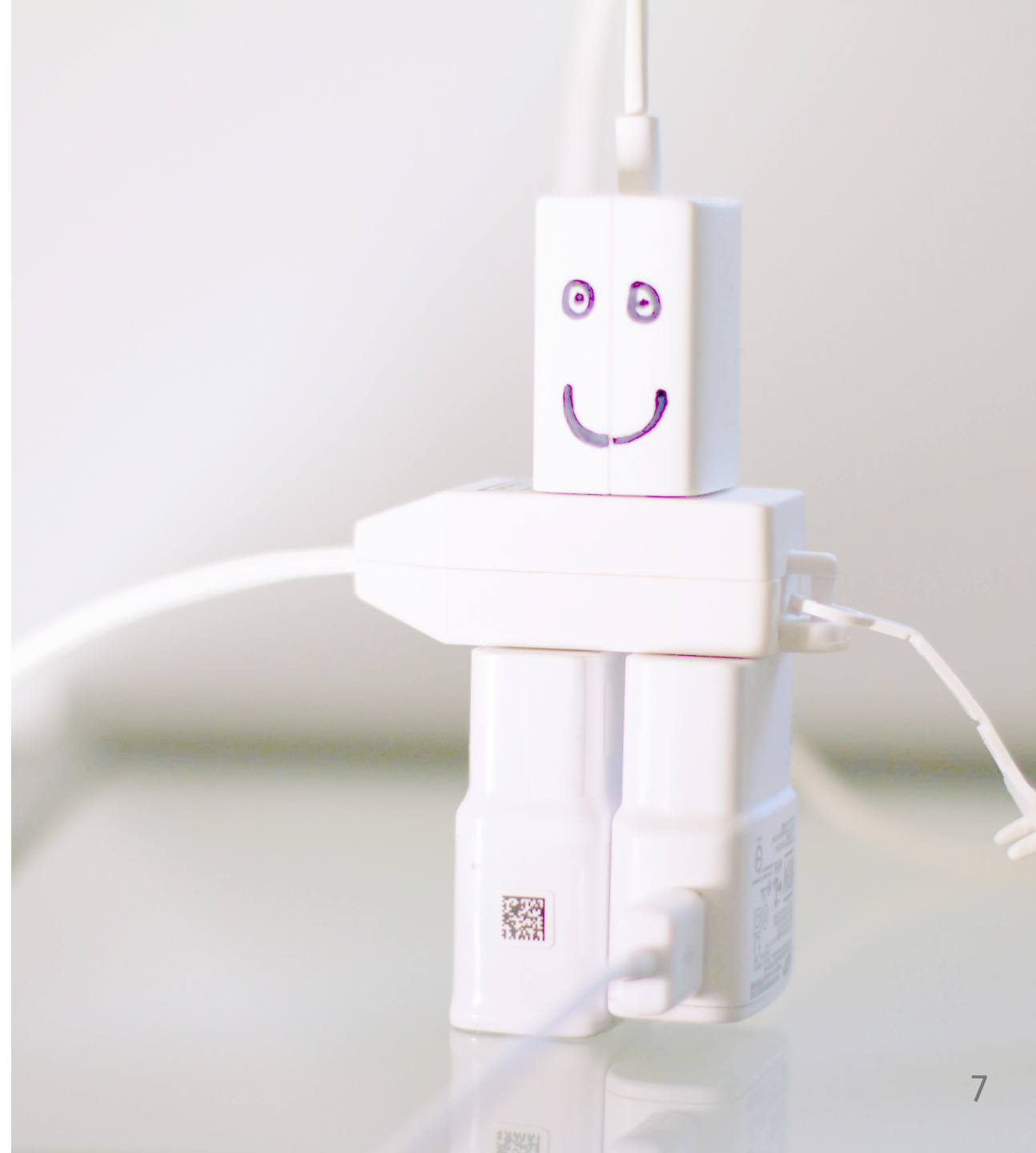
The screenshot shows the Visual Studio Code interface with a Dockerfile open in the editor. The Explorer view on the left shows a folder structure for 'bbzbl-modul-324-nginx'. The Dockerfile content is as follows:

```
1 # Pull the minimal Ubuntu image
2 FROM ubuntu:24.10
3
4 # Install Nginx
5 RUN apt-get -y update\
6   && apt-get -y install nginx git curl --no-install-recommends\
7   && rm -rf /var/lib/apt/lists/*
8
9 # Copy the Nginx config
10 COPY default /etc/nginx/sites-available/default
11 # Copy the HTML to the image
12 COPY src/* /usr/share/nginx/html
13
14 # Expose the port for access
15 EXPOSE 3000/tcp
16
17 # Run the Nginx server
18 CMD ["/usr/sbin/nginx", "-g", "daemon off;"]
19
```

VS Code Plugins

Core

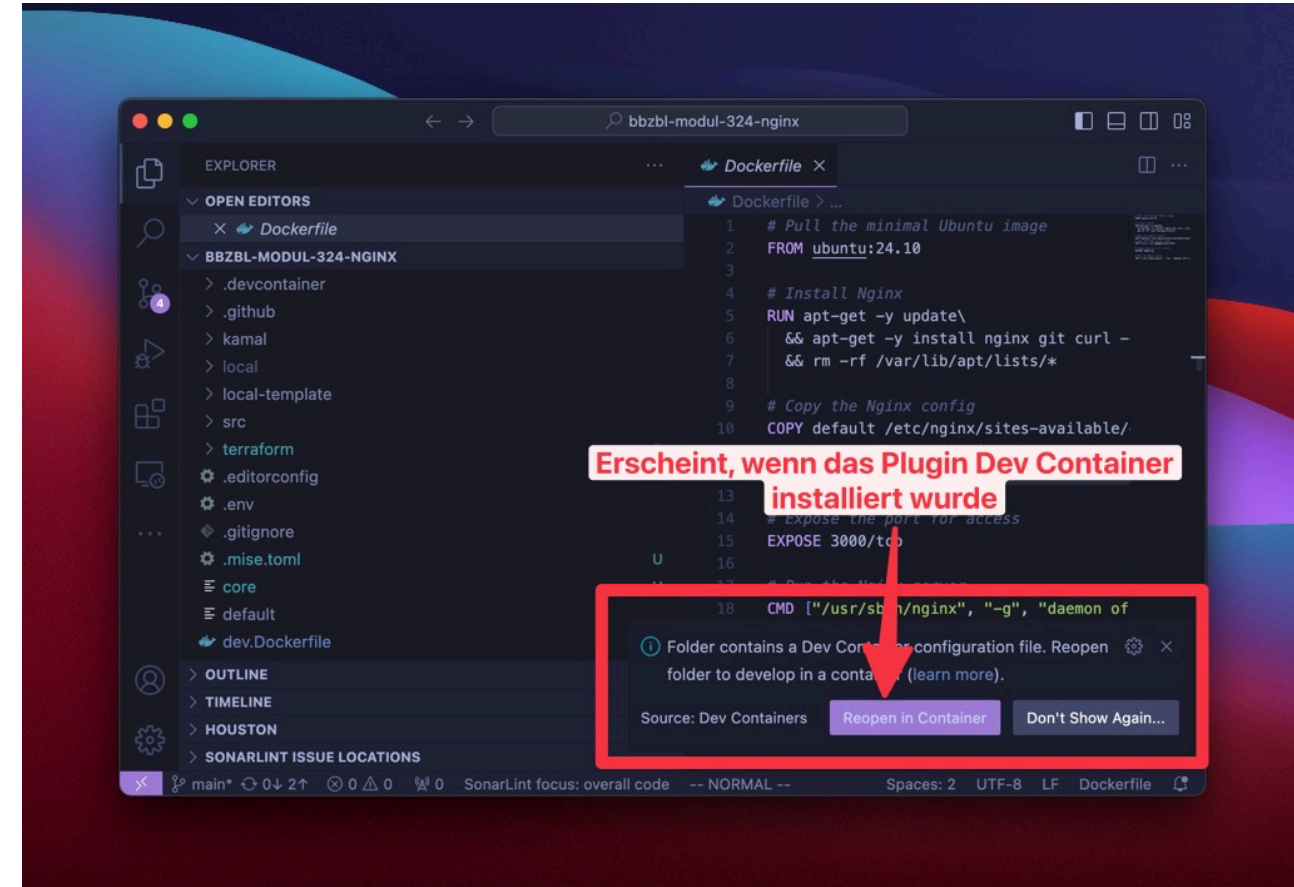
- Dev Containers
- Docker
- Remote - SSH
- EditorConfig



Öffnen in Devcontainer

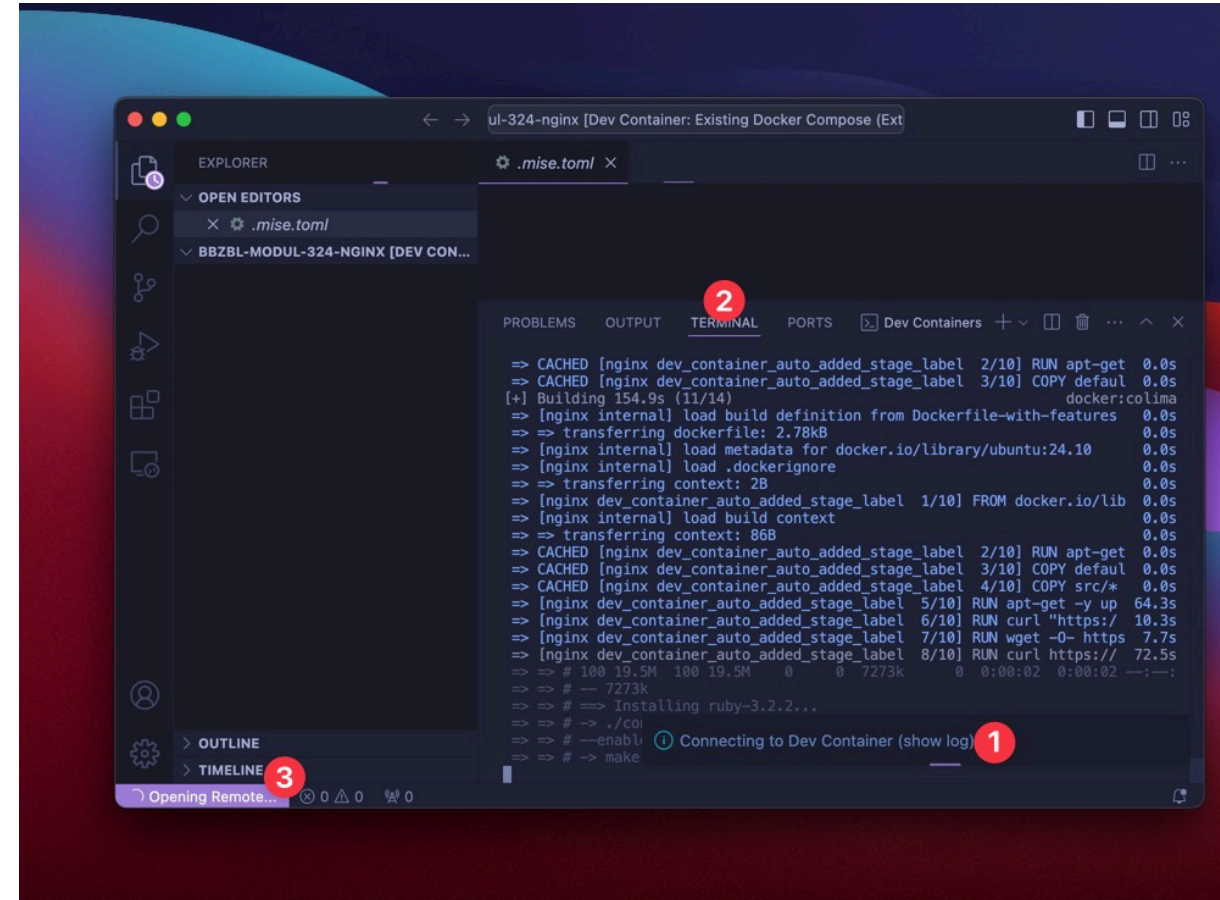
VS Code fragt automatisch nach, ob das Projekt im Container geöffnet werden soll sofern "Devcontainer" installiert wurde.

- "Reopen in Container" klicken und warten
- Nun wird der Container gebaut und gestartet. **Das kann einige Minuten dauern!**



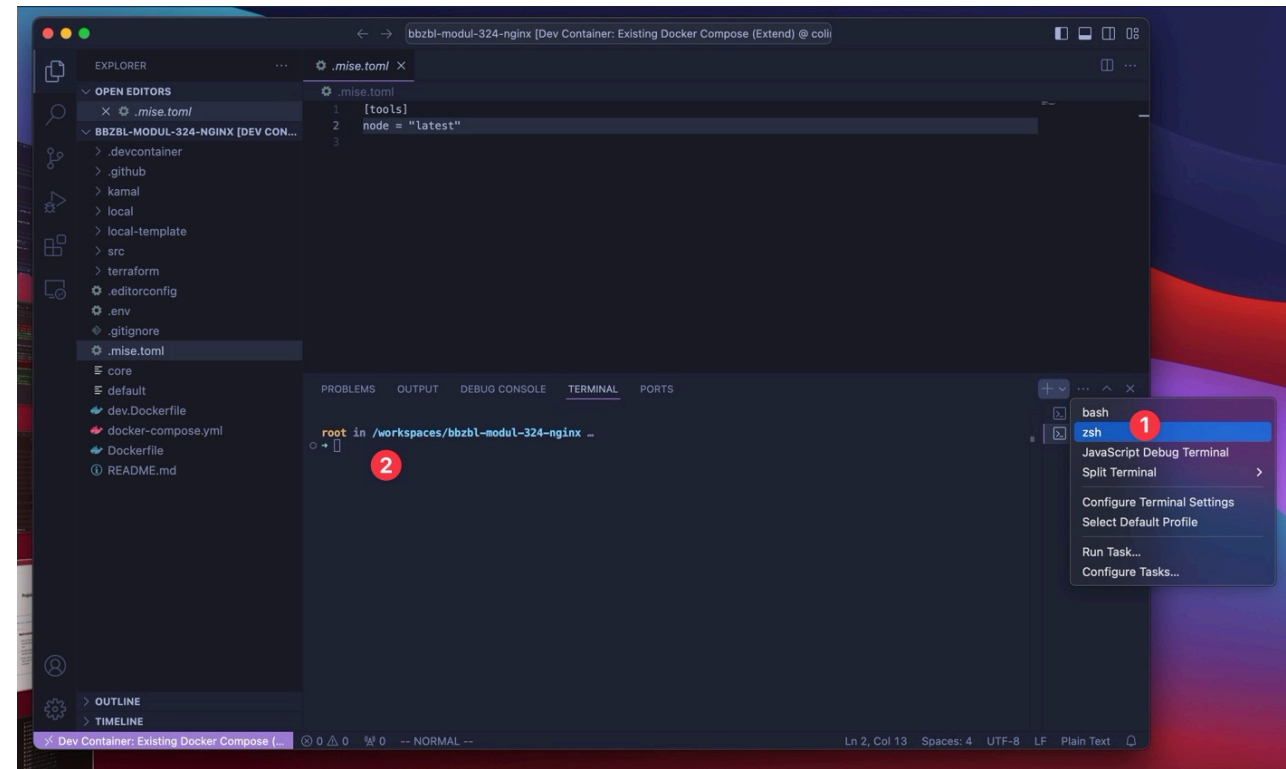
Devcontainer start

1. Wenn auf "Connecting to Dev Container (Show Logs)" geklickt wird
2. erscheint folgender Log. Es zeigt wie das "Image" gebaut wird
3. Unten rechts ist ersichtlich ob VS Code in einem Container geöffnet wird/wurde.



Devcontainer Terminal

1. Mit `+` kann ein neues Terminal geöffnet werden, (z.B. `zsh`)
2. Nun existiert ein ubuntu Terminal im Container.
(*selbst unter Windows*)
3. Die Dateien sind "gemountet" unter `/workspace`



Docker Compose

Infos unter Docker Compose

- Es verwendet das `.devcontainer/Dockerfile`
- Momentan wird `./src` zum Nginx gemountet
- `./local/.env` Environment Variablen werden auch im Container hinzugefügt
- Der Port 3000 wird exposed.

```
# docker-compose.yml
services:
  nginx:
    build:
      context: .
      dockerfile: .devcontainer/Dockerfile
    container_name: devcontainer
    ports:
      - "3000:3000"
    env_file: ./local/.env
    volumes:
      # sync workspace
      - ./workspace:cached
      ...
```

Installierte Tools im Container

Im Container sind folgendes Tool vorhanden

- `mise`

Optional (Auskommentiert):

- AWS Cli um nach AWS zu connecten
- Terraform um in AWS die Infrastruktur hochzufahren
- Kamal um ein Dockerfile zu deployen

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

root in /workspaces/bbzbl-modul-324-nginx ...
→ aws --version
aws-cli/2.17.37 Python/3.11.9 Linux/6.8.0-39-generic exe/aarch64.ubuntu.24

root in /workspaces/bbzbl-modul-324-nginx ...
→ terraform --version
Terraform v1.9.2
on linux_arm64

Your version of Terraform is out of date! The latest version
is 1.9.5. You can update by downloading from https://www.terraform.io/downloads.html

root in /workspaces/bbzbl-modul-324-nginx ...
→ kamal version
1.8.1

root in /workspaces/bbzbl-modul-324-nginx ...
→ mise --version
2024.8.12 linux-arm64 (778f806 2024-08-20)
```

Installiere Programmiersprachen mit `mise`

Java

```
mise use java@lts -g
openjdk version "24-loom" 2025-03-18
OpenJDK Runtime Environment (build 24-loom+4-42)
OpenJDK 64-Bit Server VM (build 24-loom+4-42, mixed mode, sharing)
mise java@24.0.0-loom+4-42 ✓ installed
mise /workspaces/bbzbl-modul-324-nginx/.mise.toml tools: java@24.0.0-loom+4-42
```

NodeJs

```
mise use node@lts -g
mise node@22.7.0 ✓ installed
mise /workspaces/bbzbl-modul-324-nginx/.mise.toml tools: node@22.7.0
```

Installiere Programmiersprachen im Dockerfile

Wir ein Container neu gestartet, müssen die Programmiersprachen neu installiert werden.

- Durch das `.devcontainer/Dockerfile` können Sprachen beim erstellen vom Image installiert werden.
- So wird ermöglicht, dass alle Teammitglieder die gleiche Umgebung besitzen.

- Hier ein Beispiel für nodejs

```
RUN mise use node@lts -g
```