Dev Containers and Codespaces

CS 246

Objectives

- Students will be able to:
    - explain the relationship between Docker containers, Dev Containers, and Codespaces
    - develop in Codespaces

In Praise of Dev Containers and Codespaces

- Cloud-based development environments are common in the workforce
    - GitHub Codespaces, AWS Cloud9, Replit, etc.
- Solves the "it works on *my* machine" problem
- Won't break anything on your machine, either
- Great for testing - spin up a Codespace and play

Terminology: Docker, Dev Containers and Codespaces

- A **Docker container** is a general-purpose, isolated runtime environment that an run any application

- A Docker container is created from a **Docker image**, which specifies software, dependencies, and OS libraries

- A **dev container** is a specific Docker container customized to provide a full development environment

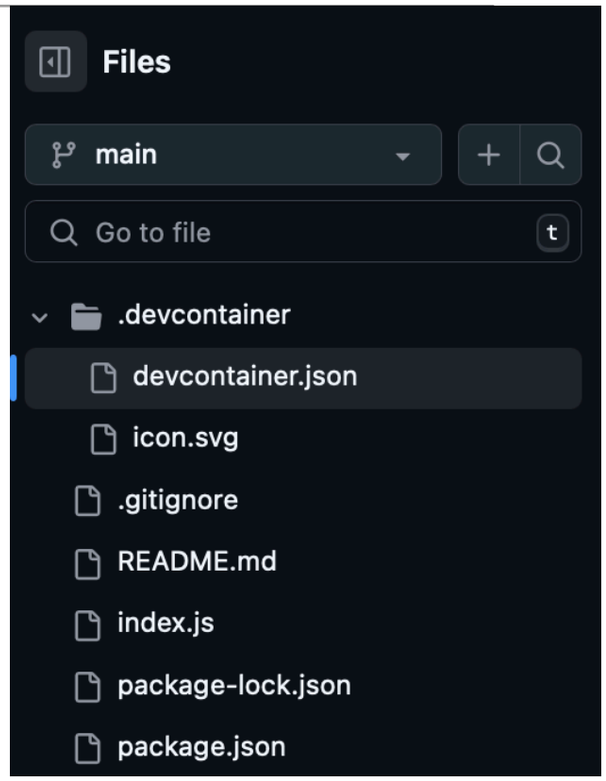
- Dev Container = Docker image + definition/config files*

- A **codespace** is a running instance of a Dev Container, hosted on GitHub.

*stored in .devcontainer/devcontainer.json

Configuring Dev Containers

- Before running a dev container, you need to make sure it has the right tools
- Configuration files are stored in devcontainer.json, in a .devcontainer directory in your repo
- GitHub provides predefined configurations, via templates, and those will meet our needs for this course



Codespace Templates

Choose a template

Start a codespace from a template and get to developing with the power of a virtual machine in the cloud.

Blank
By github

Start with a blank canvas or import any packages you need.

Use this template

React
By github

A popular JavaScript library for building user interfaces based on UI components.

Use this template

.NET
By github

A full-stack web application template written in C# leveraging the power of .NET 8.

Use this template

Ruby on Rails
By github

A full-stack web framework for building dynamic websites that deliver a rich user experience.

Use this template

Jupyter Notebook
By github

JupyterLab is the latest web-based interactive development environment for notebooks, code, and data.

Use this template

Express
By github

Express is a minimal and flexible Node.js web application framework.

Use this template

Next.js
By github

Next.js is a React framework that gives you building blocks to create web applications.

Use this template

Django
By github

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.

Use this template

Flask
By github

Flask is a lightweight web application framework.

Use this template

Preact
By github

A fast 3kB alternative to React with the same modern API.

Use this template

ISO the Dockerfile

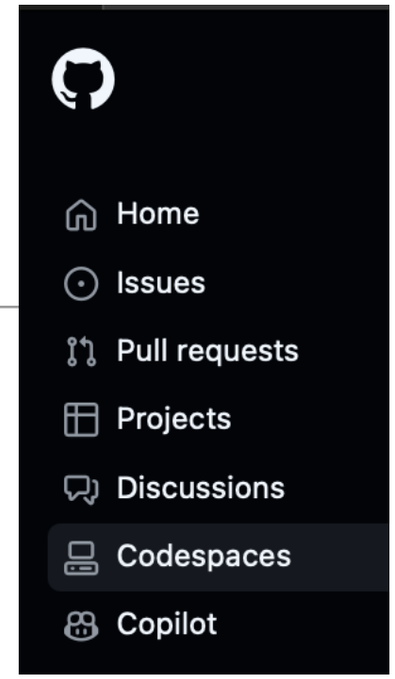
- Codespaces are running Docker images, so where's the Dockerfile, with the instructions on how to create a Docker container image?

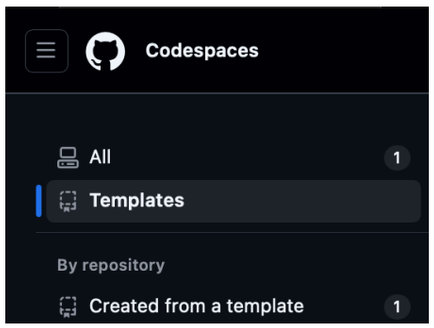
```
{
  "image": "mcr.microsoft.com/devcontainers/universal:2",
  "hostRequirements": {
    "cpus": 4
  },
  "waitFor": "onCreateCommand",
  "updateContentCommand": "npm install",
  "postCreateCommand": "",
  "postAttachCommand": {
    "server": "npm start"
  },
  "customizations": {
    "codespaces": {
      "openFiles": [
        "index.js"
      ]
    }
  },
  "portsAttributes": {
    "3000": {
      "label": "Application",
      "onAutoForward": "openPreview"
    }
  },
  "forwardPorts": [3000]
}
```

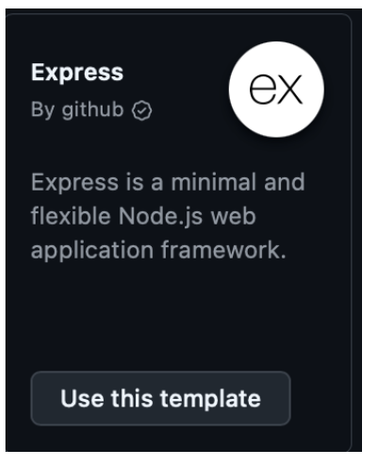
- If there is one, it will be in .devcontainer
- But devcontainer.json has an "image" property that can specify an existing image
- The universal image has pretty much everything you need

Getting Started with Codespaces

- Log in to GitHub
- From the hamburger menu on the top left, choose Codespaces
- Choose Templates
- Click on "Use this template" with Express
- Voilà!





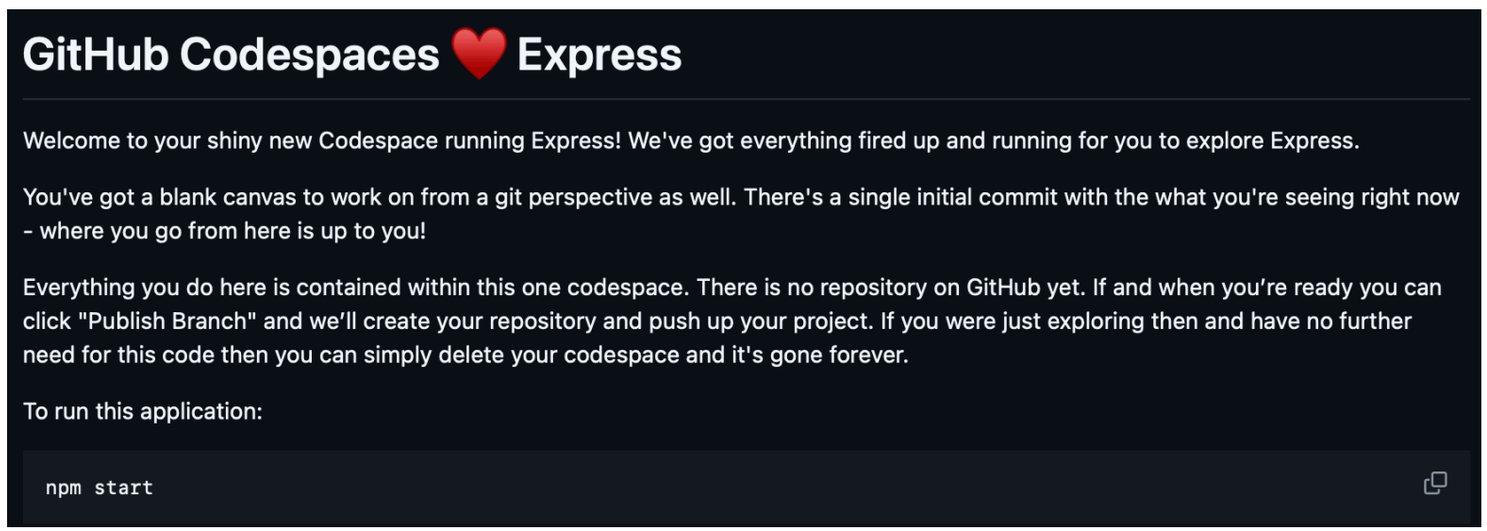


Codespace Lifecycle

- Create a codespace - from a template or an existing repo
- A codespace is created and becomes active
- You may then
  - develop in it - write code, test, etc.
  - stop it- it becomes inactive, billing stops
  - disconnect from it - it keeps running, as does the meter
  - delete it - billing stops, and your files are lost (!!)

Codespaces and Repos

- The fundamental unit on GitHub is, of course, a repo, not a codespace
- A codespace from a template has a single commit - you can then publish it to your own repo
- A repo that you own can be opened in a codespace and worked on
- Changes can be committed to the repo



In Class Demo

- In class we will:
    - Create a codespace from a template
    - Tweak it and publish it
    - Open a codespace on an existing repo
    - Tweak it and make a commit