Develop in the cloud with
# Dev Containers
and
# GitHub Codespaces

NDC Porto 2023

@jakobehn

active
SOLUTION

Jakob Ehn
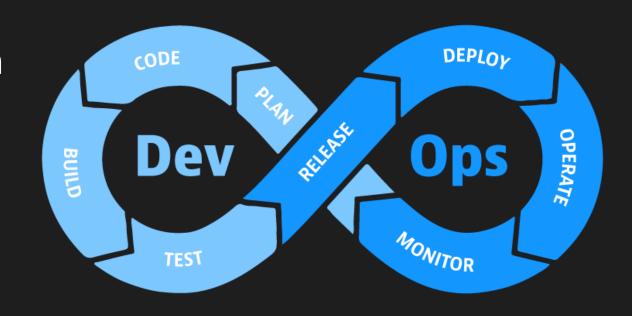
@jakobehn

Microsoft
MVP Most Valuable
Professional

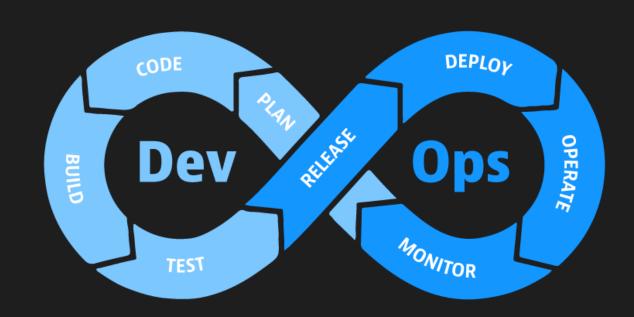# Common problems in software development

- Setting up a developer workstation

- Time to first commit

- The "works on my box" problem

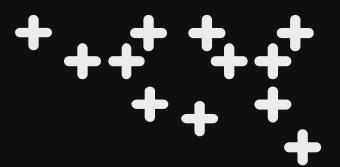- Hardware spec required for development keeps increasing

# Development Environment

- IDE
- Code
- Tools/Extensions
- SDK/Dependencies
- Personal/Team settings
- Setup scripts
- …

# Dev environment as code

- Put the development workstation in version control
  - Code, tools, extensions, settings, setup scripts, development dependencies...
  - Versioned!

- Project/Team-specific

- Allow developers to quickly be productive

- Optionally: Rent your development workstation

# GitHub Codespaces

# Blazing fast cloud developer environments

Visual Studio Code backed by high performance VMs that start in seconds.

# GitHub Codespaces

- Managed environments
- Develop from anywhere
- Customizable for teams and developers
- Multiple machine sizes
- Pay as you go
  - 60 hours/month for free

# GitHub Codespaces

**Code editor**

**VS Code Desktop**
Local installation

**VS Code Browser**
Chrome, Safari, Edge

Your editor is how you view and edit your codespace

Changes to your codespace are reflected back in your editor

**Azure**
Hosting

**Virtual Machine**
Linux hardware

**Container**
Docker development environment

**A clone of your repository**
Source code

**Languages**
Python, Ruby, etc

**Tooling**
Extensions, linting, etc

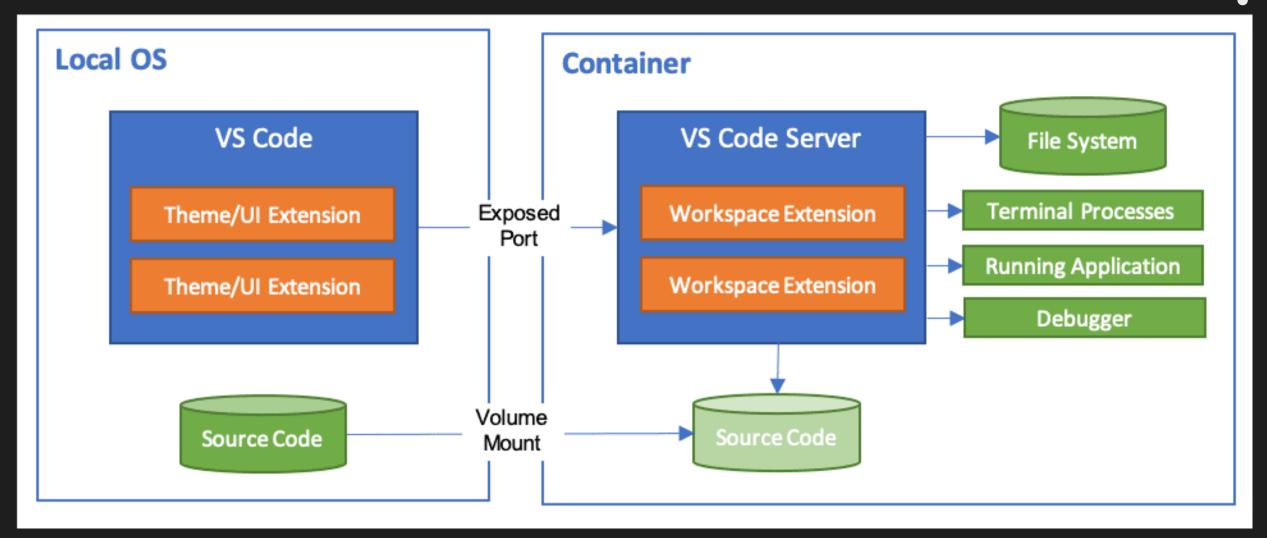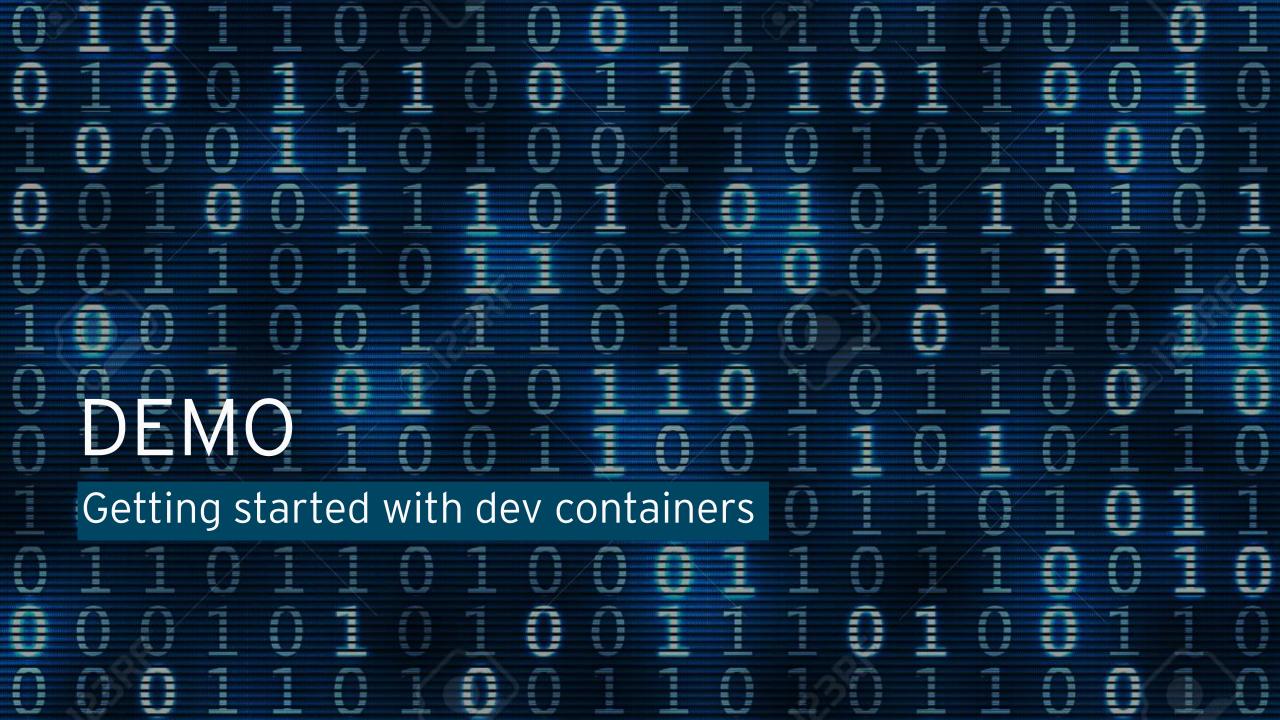# Visual Studio Code Remote Development

"Allows you to use a container, remote machine, or the Windows Subsystem for Linux (WSL) as a full-featured development environment"

# VS Code Remote Containers

# DEMO

Getting started with dev containers

# Customizing dev containers

A devcontainer.json file in your project tells VS Code how to access (or create) a **development container** with a well-defined tool and runtime stack.

# devcontainer.json
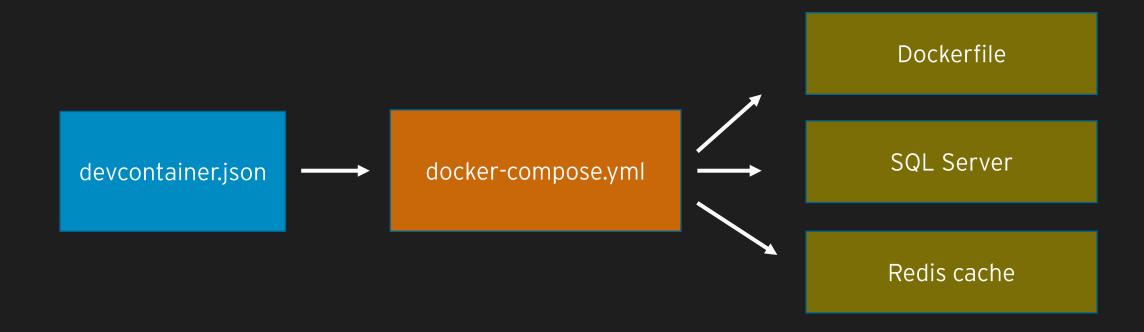
```json
{
    "name": "MyApp devcontainer",
    "image": "mcr.microsoft.com/devcontainers/dotnet:0-7.0",          Dev image

    "settings": {},

    "customizations": {
        "vscode": {
            "ms-azuretools.vscode-bicep"
        }
    ],                                                                  Extensions

    "forwardPorts": [5000, 5001],                                      Port mappings

    "postCreateCommand": "dotnet restore",                            Lifecycle commands

    // Comment out to connect as root instead.
    "remoteUser": "vscode"
}
```

# DEMO
## Customizing dev containers

# Managing dependent services

- Often dependency on other services
- SQL Server, Redis Cache, CosmosDB emulator...
- Dev containers support docker compose

DEMO
Running multiple containers

# GitHub Codespaces

**Code editor**

**VS Code Desktop**
Local installation

**VS Code Browser**
Chrome, Safari, Edge

Your editor is how you view and edit your codespace

Changes to your codespace are reflected back in your editor

**Azure**
Hosting

**Virtual Machine**
Linux hardware

**Container**
Docker development environment
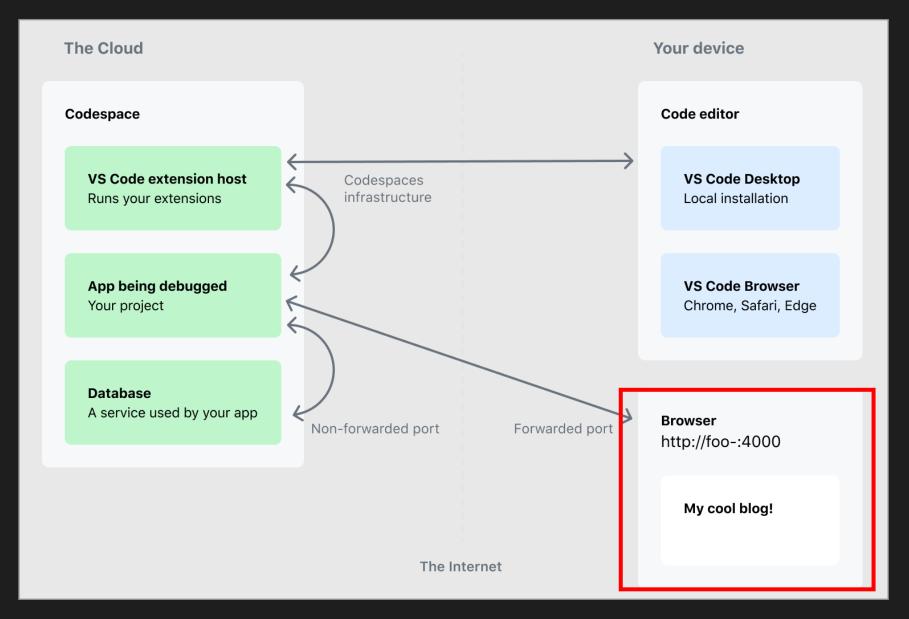
**A clone of your repository**
Source code

**Languages**
Python, Ruby, etc

**Tooling**
Extensions, linting, etc

# Runnings apps in a codespace

# Get up to 60 hours free each month

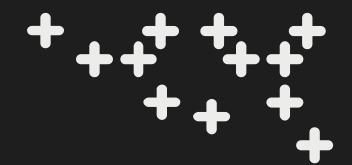Decide how many cores you need and go. Your free hours reset each month.

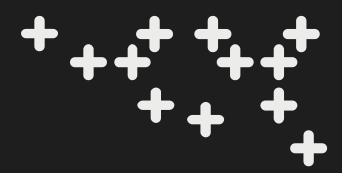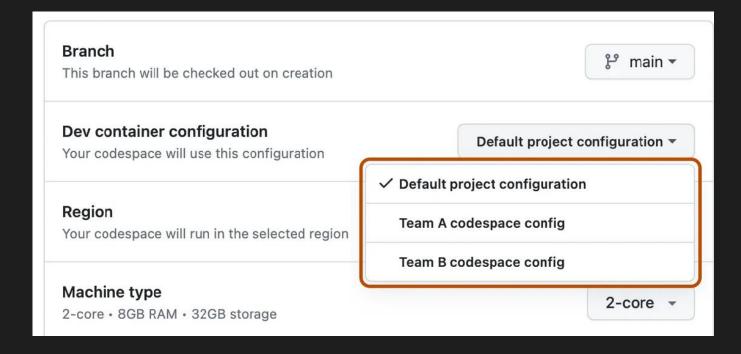| | | |
|---|---|---|
| 2 cores | 60 hours free /month | $0.18 /additional hour |
| 4 cores | 30 hours free /month | $0.36 /additional hour |
| 8 cores | 15 hours free /month | $0.72 /additional hour |
| 16+ cores | N/A | See pricing docs |
| Storage | 15 GB free /month | $0.07 /GB per month |

# DEMO
## GitHub Codespaces

But but.. Microservices?..
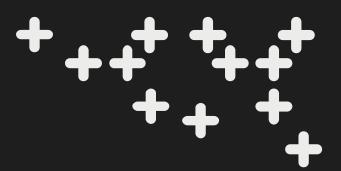
# MonoRepos
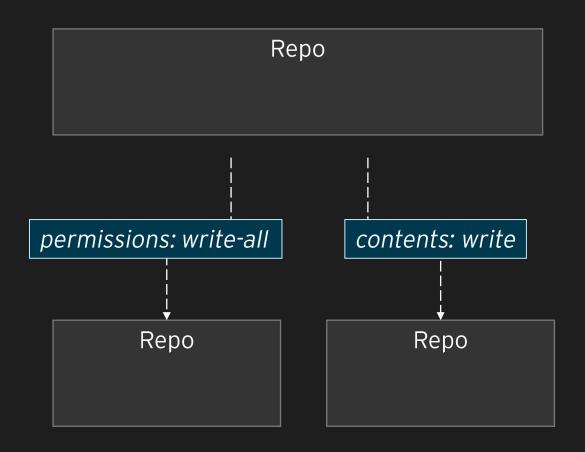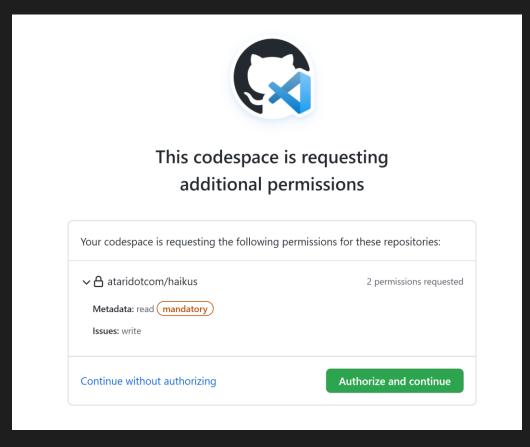
```
root
  ├─> service1
  │     └──> .devcontainer
  │              └─> devcontainer.json
  ├─> service2
  │     └──> .devcontainer
  │              └─> devcontainer.json
  └─> …
```

**Branch**
This branch will be checked out on creation

[ ⎇ main ▾ ]

**Dev container configuration**
Your codespace will use this configuration

[ Default project configuration ▾ ]

✓ Default project configuration

Team A codespace config

Team B codespace config

**Region**
Your codespace will run in the selected region

**Machine type**
2-core · 8GB RAM · 32GB storage

[ 2-core ▾ ]

# Multiple Repositories

Repo

*permissions: write-all*

*contents: write*

Repo

Repo

This codespace is requesting additional permissions

Your codespace is requesting the following permissions for these repositories:

∨ 🔒 ataridotcom/haikus                                    2 permissions requested

**Metadata:** read (mandatory)

**Issues:** write

Continue without authorizing                          Authorize and continue

DEMO
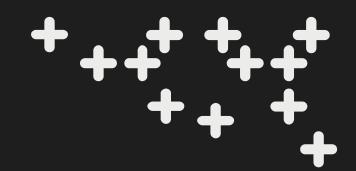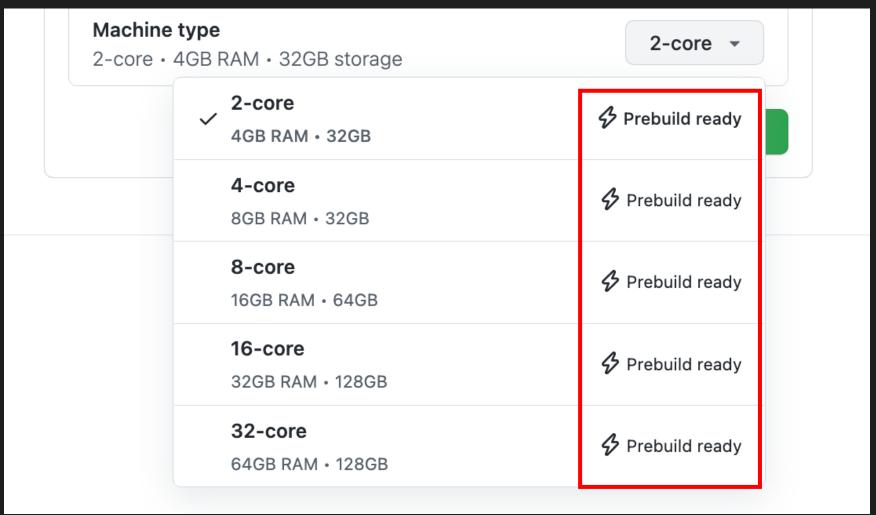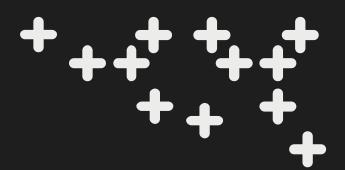GitHub Codespaces Multi-repo

# Prebuild codespaces

# Prebuild codespaces

**Prebuild Triggers**
You can specify how often to prebuild your codespace based on changes to your repository or a schedule. Read more about prebuild triggers.

○ **Every push** `Default`
Your codespace will prebuild on every push to this branch

○ **On configuration change**
Your codespace will prebuild when a change is detected on the devcontainer.json and associated configuration files

◉ **Scheduled**
Your codespace will prebuild on a schedule
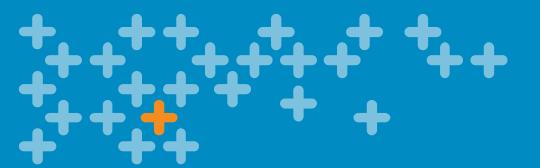
**Days**

Every day ▾

**Times**

1:00 PM, 3:30 AM ▾     (GMT -05:00) America/Chicago ▾

**Create**

https://github.blog/2021-08-11-githubs-engineering-team-moved-codespaces/

# Thank you!
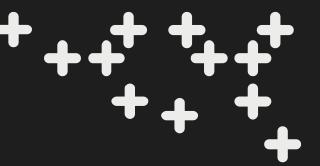
Jakob Ehn

@jakobehn

https://blog.ehn.nu

**active** SOLUTION
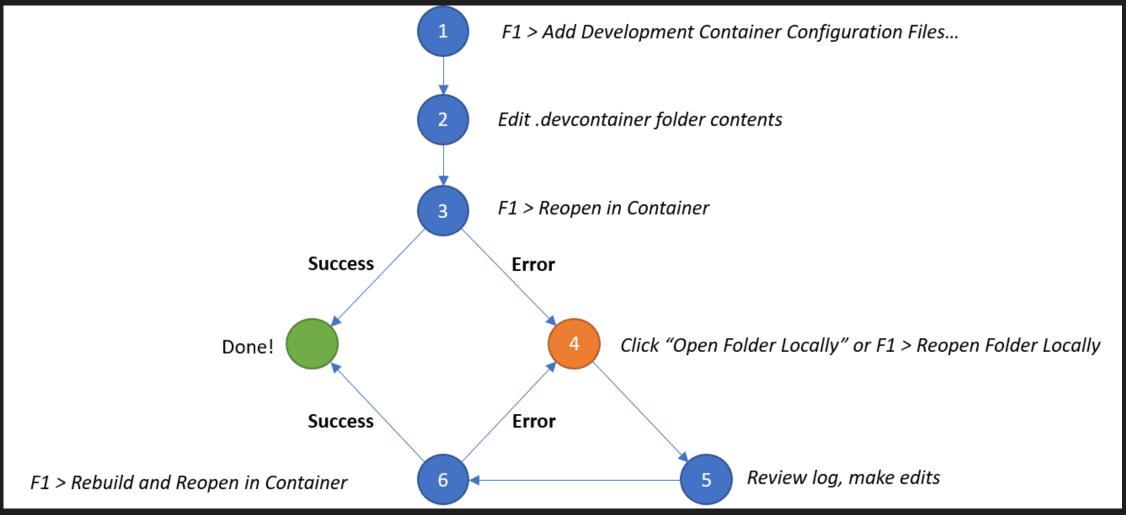
# Development container workflow

# Pre-building dev container images

```
devcontainer build --image-name ghcr.io/your-org/your-repo/img  ./path

docker push ghcr.io/your-org/your-repo/img
```

devcontainer.json
Dockerfile

→

Container Registry

Repo

devcontainer.json

Repo

devcontainer.json