



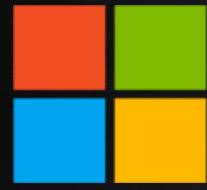
2021/09/17 イベント配布版

GitHub Codespaces 概要

Microsoft Japan
Uiko Yokoi

今日お話ししたこと

- Codespaces って、何？
- 気になるお値段
- Codespaces の何が嬉しいの？



Microsoft ❤ GitHub

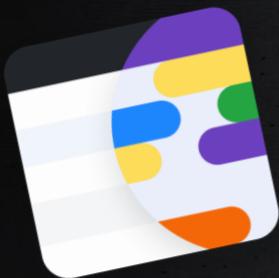
Visual Studio Codespaces → GitHub Codespaces へ統合



Codespaces って、何？



GitHub Codespaces



Codespaces とは、クラウドで
ホストされている開発環境です。
必要な開発ツールやランタイムの
バージョンが揃った、信頼性と
再現性のある環境を提供します。

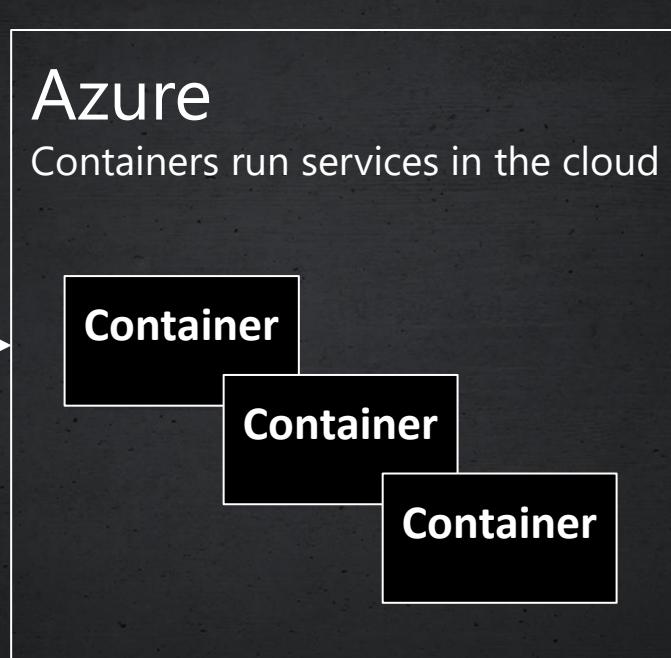
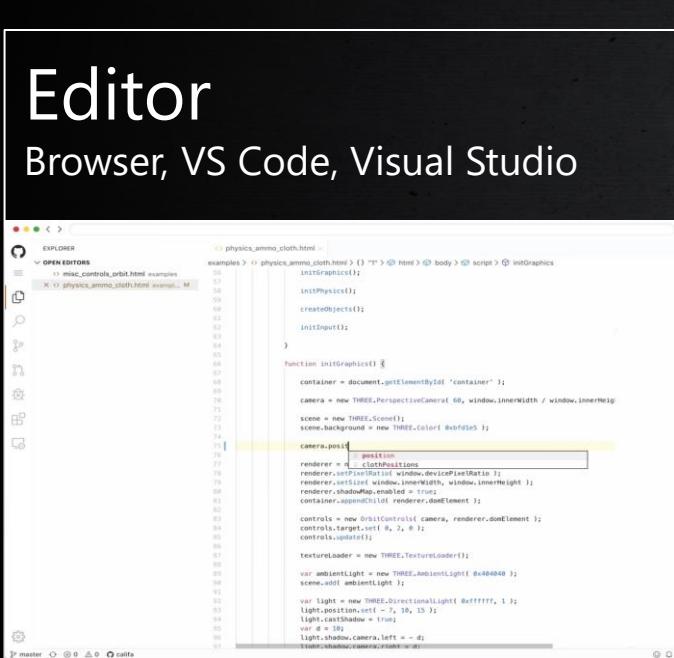
The screenshot shows the GitHub Codespaces interface running in a web browser. The left side features a sidebar with icons for file navigation, search, and other development tools. The main area is a code editor with two tabs: 'index.js' (JavaScript) and 'App.ts' (TypeScript). The code in 'index.js' is as follows:

```
1  const express = require('express')
2  const path = require('path');
3  const app = express()
4  const bodyParser = require('body-parser')
5  const Pool = require('pg').Pool
6
7  const http = require('http').createServer(app);
8
9  app.use(express.static(path.join(__dirname, 'build')));
10 app.use(bodyParser.urlencoded({ extended: false }))
11 app.use(bodyParser.json())
12
13 app.get('/', function (req, res) {
14   res.sendFile(path.join(__dirname, 'build', 'index.html'));
15 })
16
17 app.get('/visits/day', (request, response) => {
18   pool.query(`SELECT * FROM visits WHERE created_at > current_date - interval '24 hours' ORDER BY seconds ASC`)
19   if (error) {
20     throw error
21   }
22   response.status(200).json(results.rows)
23 })
24
25
26 app.get('/visits/week', (request, response) => {
27   pool.query(`SELECT * FROM visits WHERE created_at > current_date - interval '7 days' ORDER BY seconds ASC`)
```

Below the code editor is a terminal window showing the command `npm run dev` being executed, with the output "Starting server on port 3000...". The bottom of the interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL, with the TERMINAL tab currently selected.

Codespaces :

“ほんの数秒で起動するハイパフォーマンスなクラウド上のVirtual Machine 群に支えられた Visual Studio Code”



Codespaces はレイテンシを低減するために
グローバルなデータセンターで運用されています

迅速な作成、使い捨てが可能

フルマネージド環境なので
メンテナンスの必要性低減

クラウドなので個人のマシン
スペック制限なく開発可能

ブラウザがあればどこでも
即時開発開始可能



VS Code でできることは Codespaces でもできる

- エディタ、ターミナル、デバッガ
- GitHub Copilot
(9/13 現在 Technical Preview)
- バージョン管理
- 設定同期
- 拡張機能
- ブラウザ作業 ←→ デスクトップ作業へのスイッチ など

The screenshot shows the VS Code interface running in a Codespace. The Explorer sidebar on the left displays a GitHub repository structure with files like .devcontainer, build-devcontainer.sh, devcontainer.json, on-create-command.sh, post-attach-command.sh, setup-devcontainer.sh, sshr, and welcome_message.txt. The main code editor area shows a file named emotion.rb with the following content:

```
attr_reader :label
attr_reader :pronounceable_label
# Public: Get the Emoji that this reaction's content represents.
#
# Returns an Emoji.
attr_reader :emoji_character

def initialize(content:, label: nil, pronounceable_label: nil, emoji_character: nil)
  @content = content
  @label = label || @content
  @pronounceable_label = pronounceable_label || @label
  @emoji_character = emoji_character || Emoji.find_by_alias(@content)
  @platform_enum = @pronounceable_label.gsub(" ", "_").upcase
end
```

The terminal tab at the bottom shows log output from various language server processes:

```
[09:43:36] Starting 'watch-extension:vscode-api-tests'...
[09:43:36] Finished 'clean-extension:typescript-language-features' after 248 ms
[09:43:36] Starting 'watch-extension:typescript-language-features'...
[09:43:36] Finished 'clean-extension:php-language-features' after 384 ms
[09:43:36] Starting 'watch-extension:php-language-features'...
[09:43:40] Finished 'clean-extension:html-language-features-server' after 4.66 s
[09:43:40] Starting 'watch-extension:html-language-features-server'...
[09:43:43] Finished 'clean-client' after 7.33 s
[09:43:43] Starting 'watch-client'...
[09:44:50] [monaco.d.ts] Starting monaco.d.ts generation
[09:44:56] [monaco.d.ts] Finished monaco.d.ts generation
```

At the bottom right, status indicators show "Ln 29, Col 7", "Tab Size: 4", "UTF-8", "LF", "Ruby", and a few other icons.

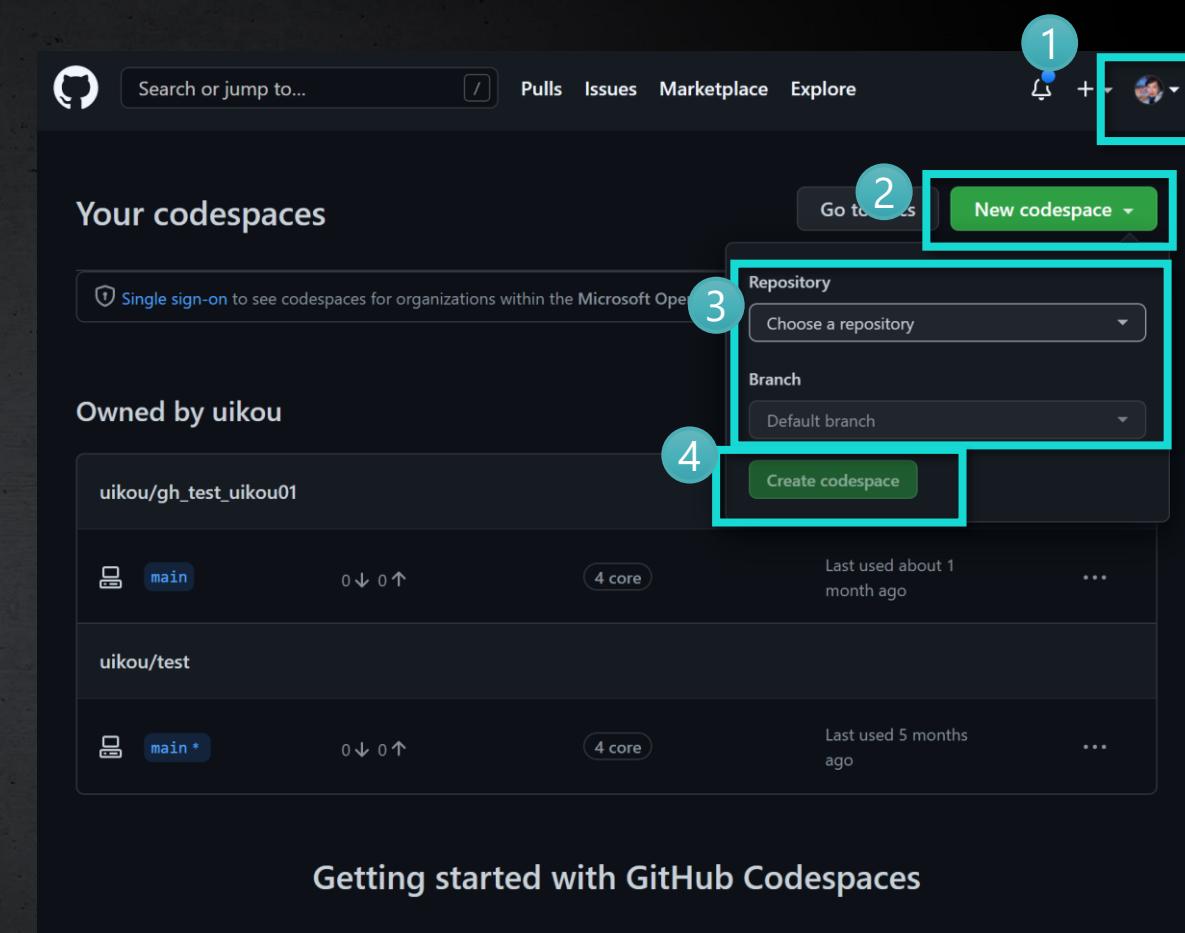
はじめかた

- リポジトリ内の任意のブランチやコミットから Codespaces を作成し、クラウドベースの Compute リソースを使用して開発を開始
 - Codecpaces 内のランタイムやツールをカスタマイズするには、カスタム構成を作成し、リポジトリに特化した環境(または dev コンテナ)を定義します。

カスタム設定を行わない場合

Codespaces は対象のリポジトリを、ツール、言語、ランタイム環境を含む既定の Codespaces イメージを持つ環境にクローニングします。

[Configuring Codespaces for your project - GitHub Docs](#) ← 詳細はこちら



- ユーザーアイコンをプルダウン
- [New codespaces] をクリック
- 対象リポジトリとブランチを選択
- [Create codespaces] で作成

環境を自分好みに

Public な dotfiles リポジトリと Settings Sync を使用することで、Codespaces 環境をユーザーそれぞれ個人にあわせたお好みの設定にすることができます

シェルの
環境設定

追加ツール

エディタ
設定

VS Code
拡張機能

パーソナライズ対象

[Customizing your codespace - GitHub Docs](#)

↑ 詳細はこちら



サポートしている ブラウザ

ブラウザさえあればどこでも開発可能



GitHub.dev

リポジトリで “.” を入力すると、
Web エディタが立ち上がり、
素早く編集ができるようになります。
ちなみに、簡易エディタのようなも
のであり、ターミナルや
コンテナなどはついてきません。

Wonder what would happen if you went on a @GitHub repo and changed the URL from "github.com" to "github.dev"? 🙄

microsoft/playwright: Node.js | X +

https://github.com/microsoft/playwright

Search or jump to... / Pulls Issues

crosoft / playwright Watch ▾

0:03 125.9K views 274 Pull requests 21

2:03 AM · Aug 12, 2021 · TweetDeck

1,128 Retweets 289 Quote Tweets 3,975 Likes

実績：GitHub、Codespaces 使っています

GitHub の開発チーム自身も、開発環境を Codespaces に移行済み

クローン作製

20 分



90 秒

開発環境準備

設定済み
Codespaces
利用で



10 秒

コアリポジトリ : 100 万超のコミット
超巨大プロジェクト
(13 GB 以上、14 年以上かけて作ってきたでっかいプロジェクト)

GitHub も Codespaces に移行したら
こんなに短縮できたんだよ！



[GitHub 開発チーム 苦難の歴史](#)
[GitHub's Engineering Team has moved to Codespaces | The GitHub Blog](#)

気になるお値段



気になるお値段

- 環境に使用する CPU の性能と、利用時間（1 時間単位）の従量課金となります。また、データストレージは 1GB / 月単位で米ドル単位で月額請求されます。
- Organization で Codespaces の課金を管理するには、Organization の owner もしくは billing Manager である必要があります。

2021/9 現在、Team Plan および Enterprise プランのアカウントが課金対象。個人アカウント（ベータ）には課金されていません。

Product	SKU	Unit of measure	Price
Codespaces Compute	2 コア	1 時間	\$0.18
	4 コア	1 時間	\$0.36
	8 コア	1 時間	\$0.72
	16 コア	1 時間	\$1.44
	32 コア	1 時間	\$2.88
Codespaces Storage	Storage	1GB / 月	\$0.07

気になるお値段：Compute 利用料

Compute 利用料の
計算はどうするの？



- Codespaces インスタンスがアクティブになっている稼働時間の合計分数です。
- Compute 使用料は、すべての Codespaces で実際に使用された分数で毎日合計値が課金サービスにレポートされ、毎月単位で請求されます。

気になるお値段：Storage 利用料

ストレージ利用料の
計算はどうなるの？



- Codespaces の課金の対象は、アカウント内の全ての Codespaces で使用されるすべてのストレージとなります。
 - これには、クローンされたリポジトリ、構成ファイル、拡張機能など、Codespaces で使われるすべてのファイルが対象になります。これらの合計値は毎日課金サービスにレポートされ、月末にストレージについて最も近い MB 単位に丸められます。

気になるお値段： フォークされたリポジトリ

フォークされたリポジトリでの課金は？



- Codespaces は、課金対象となる Owner が定義されている Organization でのみ使用することができます。Organization に課金するには、ユーザーが member または collaborator である必要があります。
(これ以外のユーザーではコードスペースを作成できません)
- これは、Organization が親リポジトリの Owner であり、ユーザーのアクセス、フォークされたリポジトリ、およびコードスペースを削除できるためです。

気になるお値段： リポジトリ移管時の課金

リポジトリ移管
した場合の課金は？



- 使用量は 1 時間ごとに請求され
レポートされます。
 - リポジトリがお客様の Organization 内にある場合は、
使用量に応じてお支払いいただきます。
リポジトリがお客様の Organization 外に移管された
場合、そのリポジトリの Codespaces は移管
プロセスの一部として削除されます。

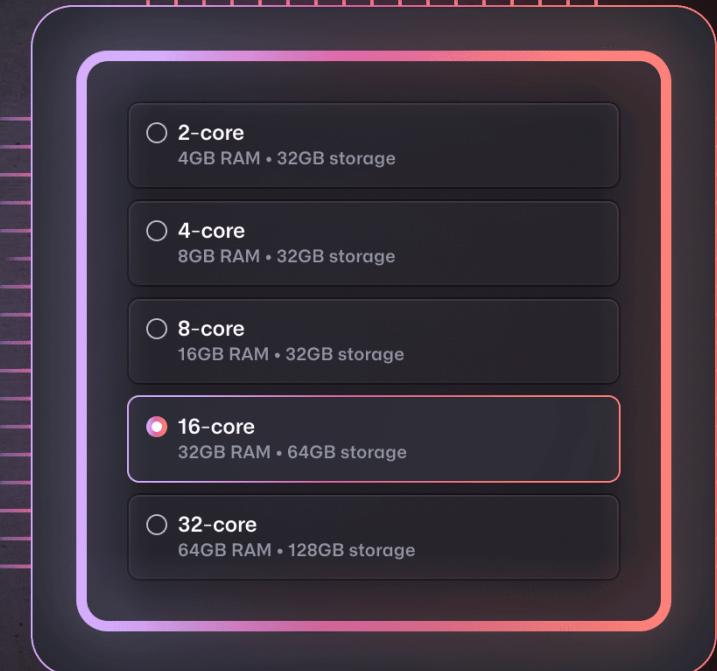
Codespaces の 何がうれしいの？



うれしい①：数秒で新規開発環境作成

- とにかく高速。開発が捲ります
 - あらかじめ用意されたイメージを使って、どんな規模のプロジェクトでも数秒で新しい開発環境を立ち上げることができます

35GB の開発イメージは、10 秒以内に起動。
クラウド VM は、最大で 32 コア、
64GB の RAM まで拡張可能



うれしい②：新しい人も即時コーディング可能

- 開発環境を標準化。今日 Join した人も即コーディング可能
 - 設定を devcontainer.json 設定ファイルで標準化。コンテナと docker-compose を使って、プロジェクト間の依存関係を分離します

どのようなプロジェクトでも
コードからコミットまでの時間を短縮！



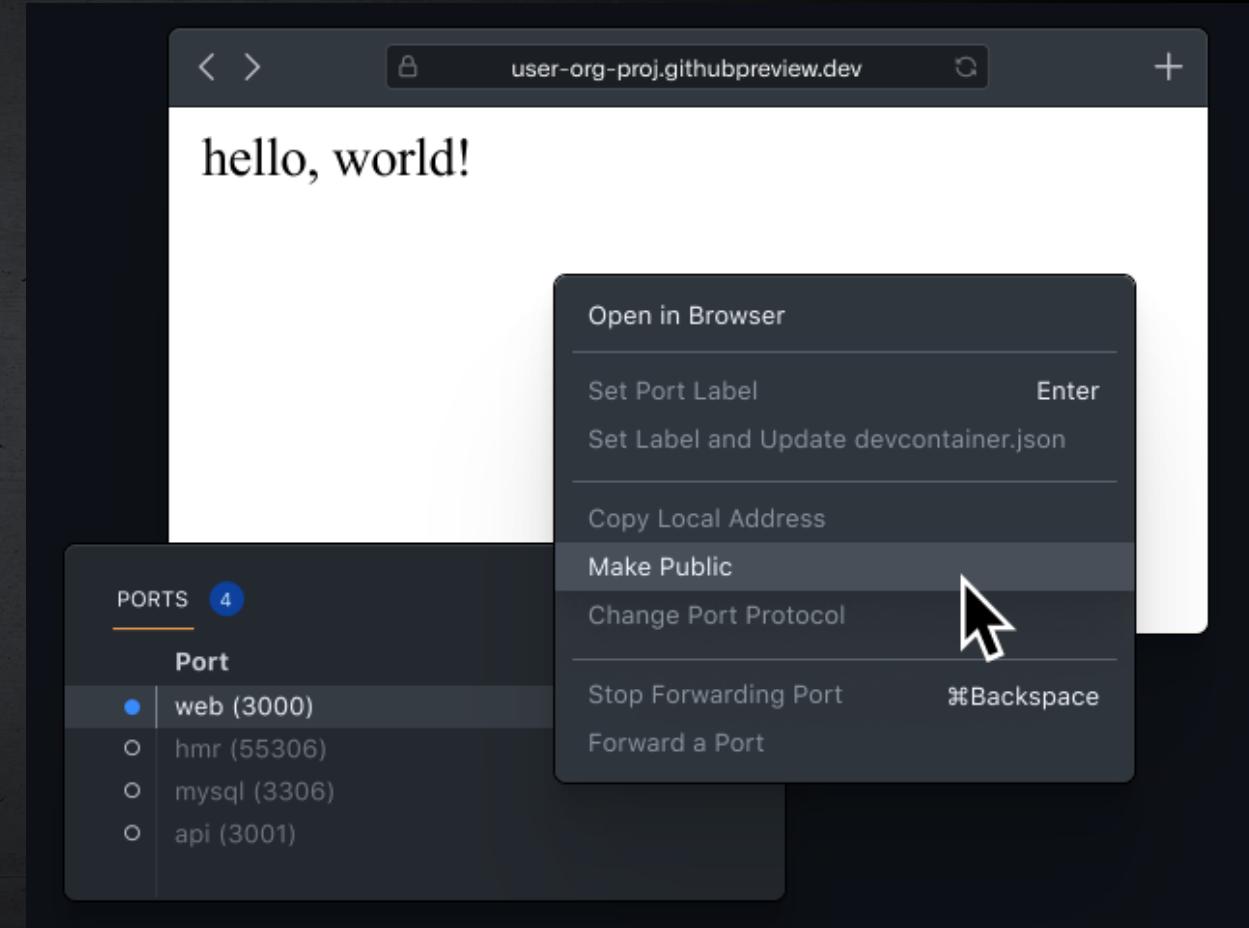
The screenshot shows the Visual Studio Code interface with the following details:

- EXPLORER** pane: Shows a tree view of the project structure under the GITHUB folder. The .devcontainer folder contains build-devcontainer.sh, devcontainer.json, on-create-command.sh, post-attach-command.sh, setup-devcontainer.sh, ssh, and welcome_message.txt. The app folder contains api, assets, components, controllers, helpers, jobs, mailers, and models. The models folder contains archived.rb, emotion.rb (which is currently selected), and interaction.rb.
- Editor** pane: Displays a file named emotion.rb with code related to emoji handling. The code includes methods for reading attributes, returning pronounceable labels, and initializing content with labels.
- Terminal** pane: Shows logs of background tasks, including starting and finishing 'watch-extension' and 'clean-extension' services, and a 'monaco.d.ts' generation process.

うれしい③：ブラウザプレビュー、ポートフォワーディングが簡単

- インスタントリロードで変更をブラウザでプレビューし、チームの関係者とプライベートおよびパブリックポートを共有可能

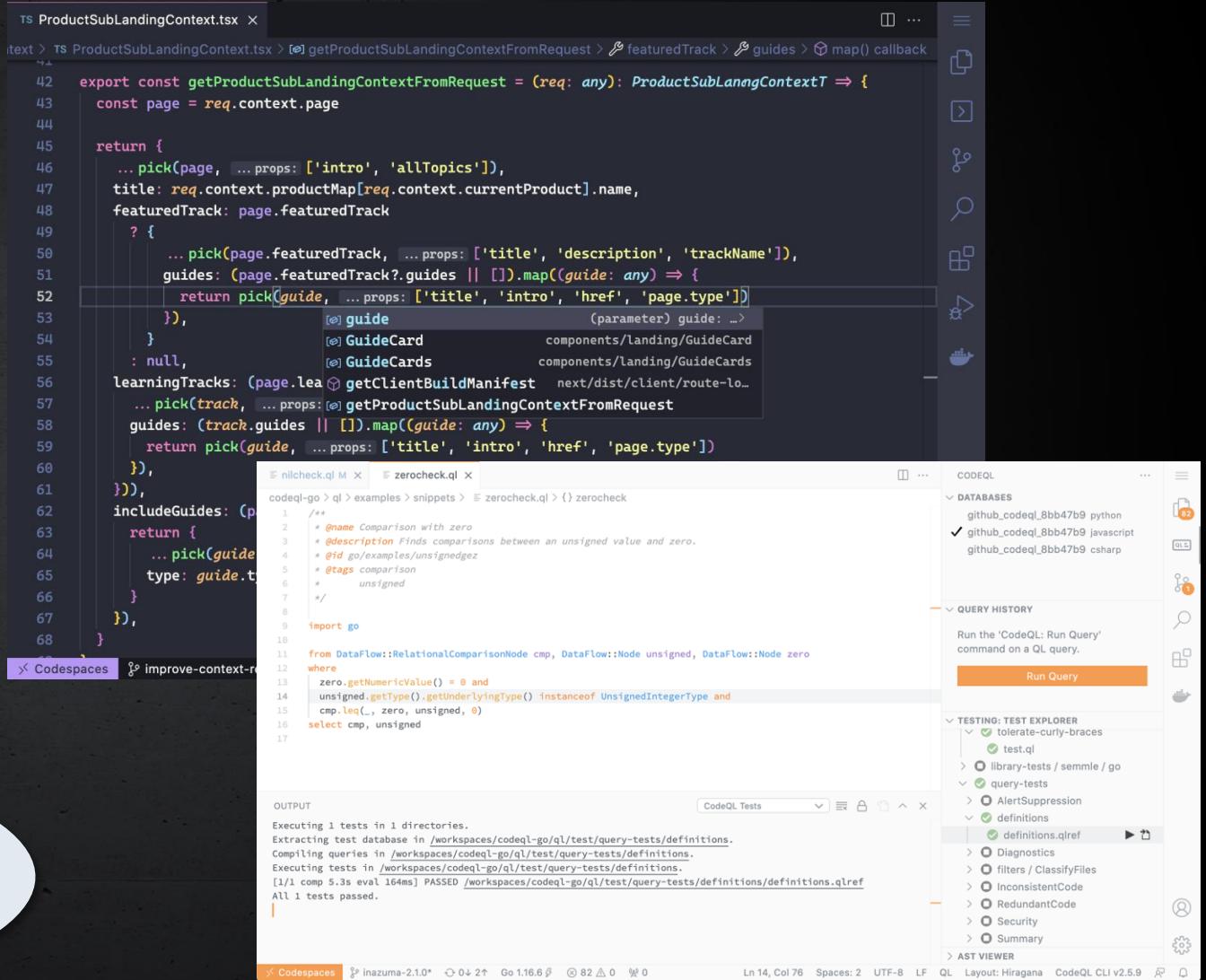
インスタントリロードはwebsocketとHMRをサポート



うれしい④：自分好みの環境設定もOK

- 自分自身の環境もちゃんと設定可能
 - Tab、Space、ライト / ダークなどのテーマなど、自分の好みの開発環境を設定

開発環境設定は
dotfiles リポジトリにバックアップ



The screenshot shows the VS Code interface with a dark theme. A code completion dropdown is open over the word 'guide' in a TypeScript file named 'ProductSubLandingContext.tsx'. The dropdown lists several suggestions: 'guide' (parameter), 'GuideCard', and 'GuideCards'. The file contains logic for handling product sub-landing contexts based on requests and returning specific page details like titles, descriptions, and featured tracks.

Below the editor, the 'OUTPUT' panel shows the results of a 'CodeQL' test run:

```
Executing 1 tests in 1 directories.
Extracting test database in /workspaces/codeql-go/ql/test/query-tests/definitions.
Compiling queries in /workspaces/codeql-go/ql/test/query-tests/definitions.
Executing tests in /workspaces/codeql-go/ql/test/query-tests/definitions.
[1/1 comp 5.3s eval 164ms] PASSED /workspaces/codeql-go/ql/test/query-tests/definitions/definitions.qlref
All 1 tests passed.
```

The sidebar on the right shows various open files, databases, and a query history section.



まとめ

- とにかく高速。開発が捲ります
 - あらかじめ用意されたイメージを使って、どんな規模のプロジェクトでも数秒で新しい開発環境を立ち上げることができます
- 開発環境を標準化。今日 Join した人も即コーディング可能
 - 設定を `.devcontainer.json` 設定ファイルで標準化。コンテナと `docker-compose` を使って、プロジェクト間の依存関係を分離します
- ブラウザプレビューとポートフォワーディングが簡単
 - インスタントリロード（websocket と HMR サポート）で変更をブラウザでプレビューし、チームの関係者とプライベートおよびパブリックポートを共有することができます
- 自分自身の環境もちゃんと設定できます
 - Tab、Space、ライト / ダークなどのテーマなど、自分の好みの開発環境設定をすれば、`dotfiles` リポジトリにバックアップされます。



- 本書に記載した情報は、本書各項目に関する発行日現在の **Microsoft** の見解を表明するものです。Microsoftは絶えず変化する市場に対応しなければならないため、ここに記載した情報に對していかなる責務を負うものではなく、提示された情報の信憑性については保証できません。
- 本書は情報提供のみを目的としています。 Microsoft は、明示的または暗示的を問わず、本書にいかなる保証も与えるものではありません。
- すべての当該著作権法を遵守することはお客様の責務です。 Microsoftの書面による明確な許可なく、本書の如何なる部分についても、転載や検索システムへの格納または挿入を行うことは、どのような形式または手段（電子的、機械的、複写、レコードティング、その他）、および目的であっても禁じられています。これらは著作権保護された権利を制限するものではありません。
- Microsoftは、本書の内容を保護する特許、特許出願書、商標、著作権、またはその他の知的財産権を保有する場合があります。 Microsoftから書面によるライセンス契約が明確に供給される場合を除いて、本書の提供はこれらの特許、商標、著作権、またはその他の知的財産へのライセンスを与えるものではありません。
- Microsoft, Windows, その他本文中に登場した各製品名は、Microsoft Corporation の米国およびその他の国における登録商標または商標です。その他、記載されている会社名および製品名は、一般に各社の商標です。