

```
    if _operation == "MIRROR_X":  
        mirror_mod.use_x = True  
        mirror_mod.use_y = False  
        mirror_mod.use_z = False  
    elif _operation == "MIRROR_Y":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = True  
        mirror_mod.use_z = False  
    elif _operation == "MIRROR_Z":  
        mirror_mod.use_x = False  
        mirror_mod.use_y = False  
        mirror_mod.use_z = True  
  
    #selection at the end -add back the deselected mirror modifier  
    mirror_ob.select= 1  
    modifier_ob.select=1  
    bpy.context.scene.objects.active = modifier_ob  
    print("Selected" + str(modifier_ob)) # modifier ob is the active object  
    #mirror_ob.select = 0  
    bone = bpy.context.selected_objects[0]  
    bpy.ops.object.select_all(action='DESELECT'); bone.select = 1  
    bpy.ops.object.select_all(action='SELECT')
```

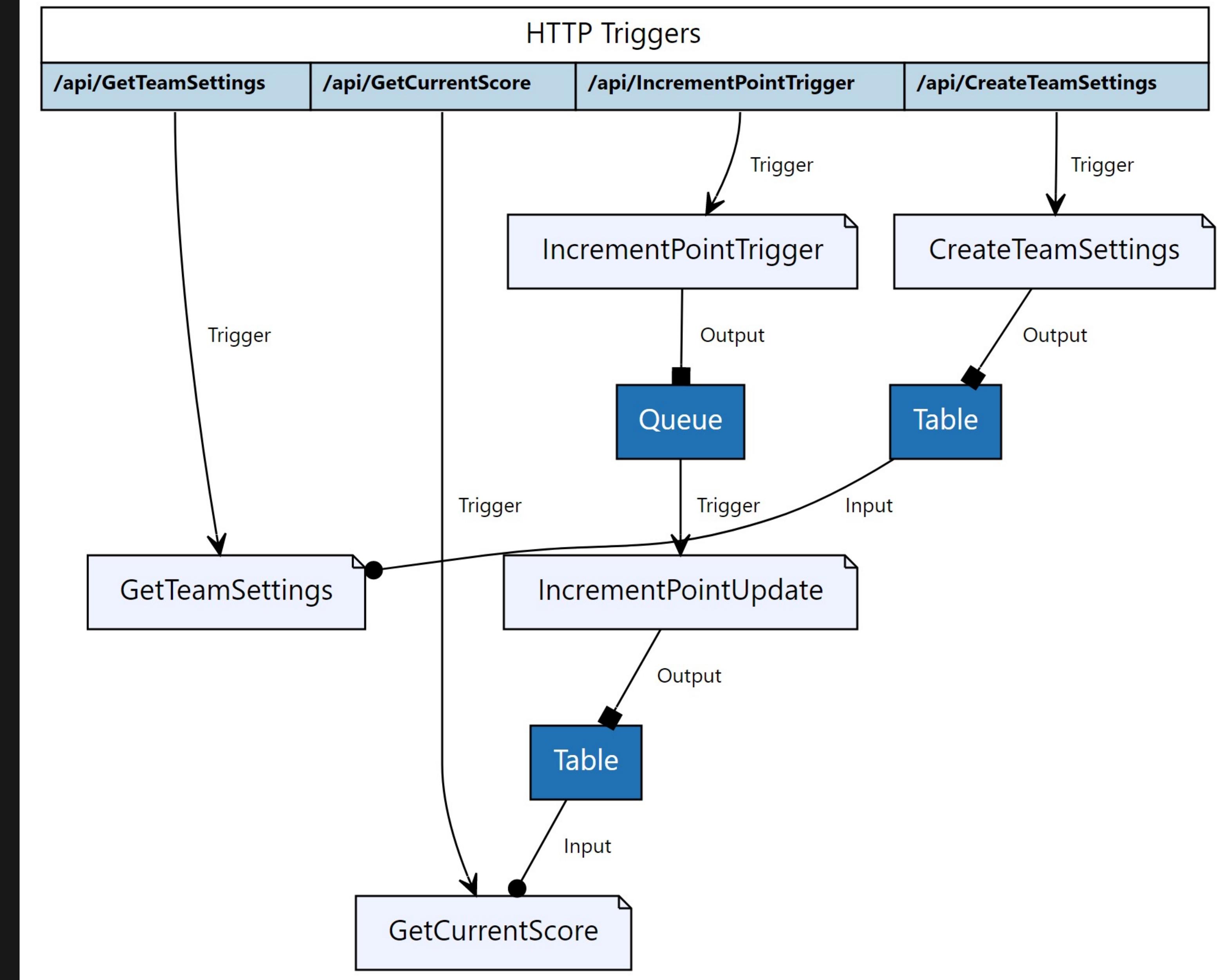


GitHub Codespaces



Let's play a game of
<https://aka.ms/tug-of-war>

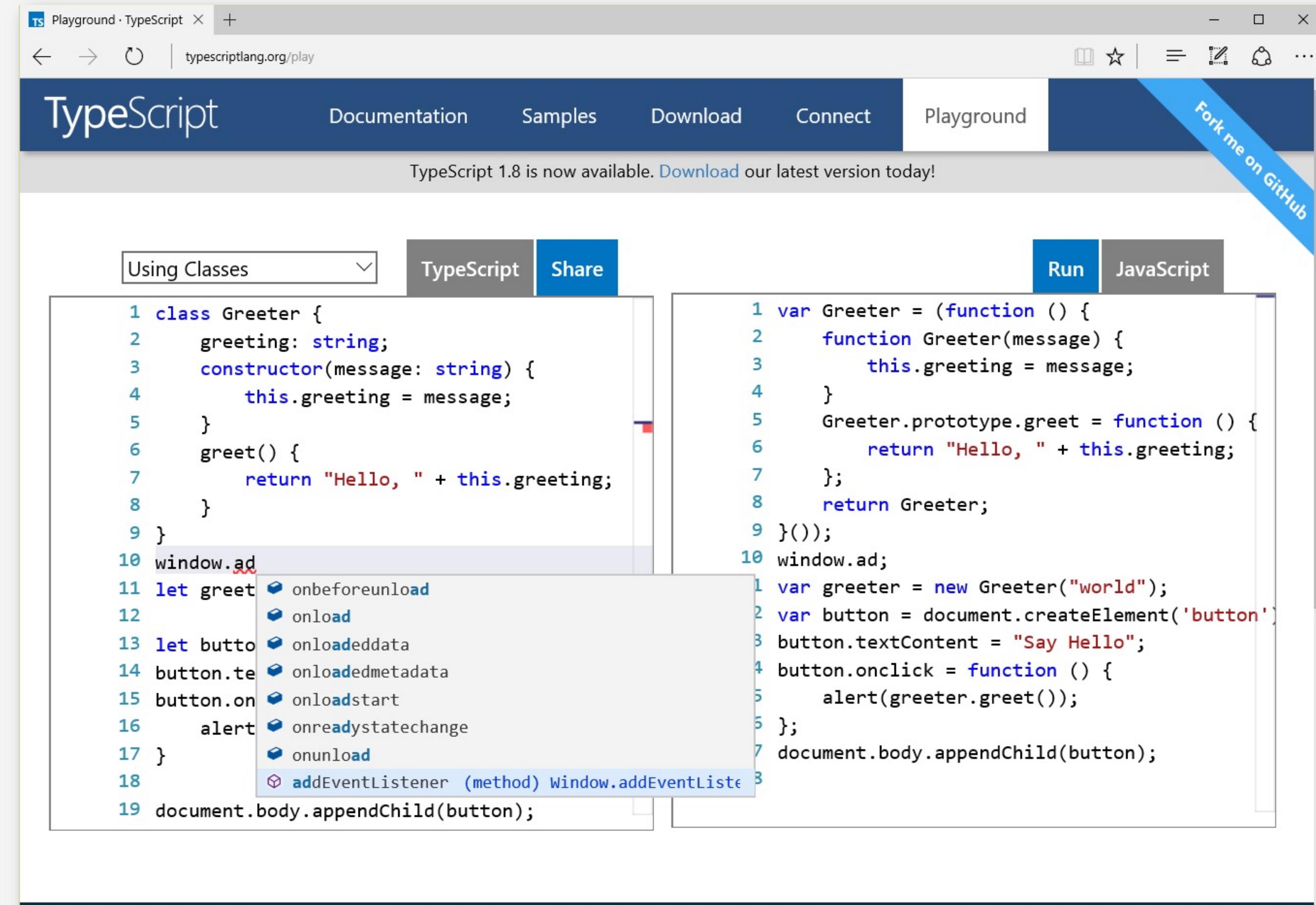
DEMO



<https://github.com/joescars/TugOfWar-FunctionsDemo>

2011

"MONACO" HIGH PERFORMANCE WEB BASED EDITOR



2013

VISUAL STUDIO ONLINE "MONACO" (V1)

The screenshot shows the Visual Studio Online Monaco editor interface. The title bar indicates it's an App Service Editor session for a project named 'app.js - wwwroot - App S'. The address bar shows the URL <https://vsomonaco.scm.azurewebsites.net/dev/wwwroot/app.js>. The ribbon menu includes 'App Service Editor' and 'Chris'. The left sidebar has icons for 'Apps', 'personal', 'email', 'news', 'sharepoint', 'Standup', 'New Issue', and 'Azure'. Below that is a navigation bar with 'App Service Editor | vsomonaco | master* | Chris Dias | ? | ☺'. The main area has a dark theme with a light border. On the left is an 'EXPLORE' sidebar showing the project structure:

- WORKING FILES:
 - index.js
 - .gitignore
- WWWROOT:
 - .git
 - bin
 - node_modules
 - public
 - routes
 - views
 - .gitignore
 - app.js
 - hostingstart.html
 - index.js
 - package.json

The 'app.js' file is selected in the sidebar. The right pane displays the code content:

```
1 var express = require('express');
2 var path = require('path');
3 var favicon = require('serve-favicon');
4 var logger = require('morgan');
5 var cookieParser = require('cookie-parser');
6 var bodyParser = require('body-parser');
7
8 var index = require('./routes/index');
9 var users = require('./routes/users');
10
11 var app = express();
12
13 // view engine setup
14 app.set('views', path.join(__dirname, 'views'));
15 app.set('view engine', 'jade');
16
17 // uncomment after placing your favicon in /public
18 //app.use(favicon(path.join(__dirname, 'public', 'favicon.ico')));
19 app.use(logger('dev'));
20 app.use(bodyParser.json());
21 app.use(bodyParser.urlencoded({ extended: false }));
22 app.use(cookieParser());
23 app.use(express.static(path.join(__dirname, 'public')));
24
25 app.use('/', index);
26 app.use('/users', users);
27
28 // catch 404 and forward to error handler
29 app.use(function(req, res, next) {
30   var err = new Error('Not Found');
31   err.status = 404;
32   next(err);
33 });
34
35 // if err.code === 404, just log
36 //err.message = 'Page Not Found';
37 //res.status(err.status).send(err.message);
```

A NEW CLASS OF TOOLS

Editors

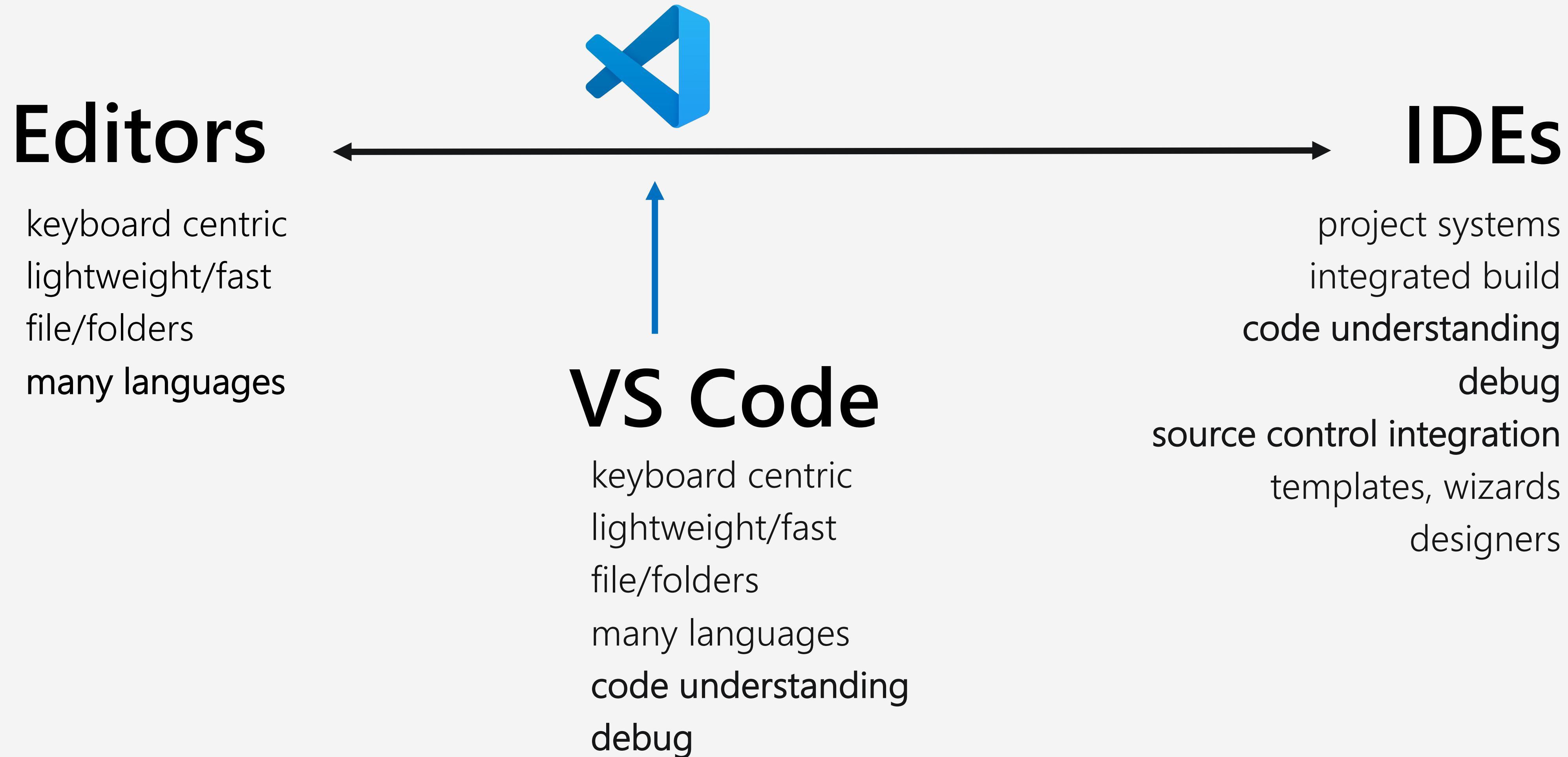
keyboard centric
lightweight/fast
file/folders
many languages

IDEs

project systems
integrated build
code understanding
debug
source control integration
templates, wizards
designers



A NEW CLASS OF TOOLS



2015

VISUAL STUDIO CODE

The image shows a screenshot of the Visual Studio Code website and the VS Code application window side-by-side.

Visual Studio Code Website:

- Header:** Visual Studio Code - Code Editor, https://code.visualstudio.com
- Navigation:** Visual Studio Code, Docs, Updates, Blog, API, Extensions, FAQ, Search Docs, Download
- Message:** Version 1.42 is now available! Read about the new features and fixes from January.
- Section:** Code editing. Redefined. Free. Built on open source. Runs everywhere.
- Download Buttons:** Download for Windows (Stable Build), Other platforms and Insiders Edition.
- Text:** By using VS Code, you agree to its license and privacy statement.

Visual Studio Code Application:

- File Explorer:** EXTENSIONS: MARKETPLACE
- Code Editors:** JS App.js, JS index.js, JS serviceWorker.js
- Terminal:** 1: node
- Status Bar:** master, 0 ▲ 0, Ln 43, Col 19, Spaces: 2, UTF-8, LF, JavaScript

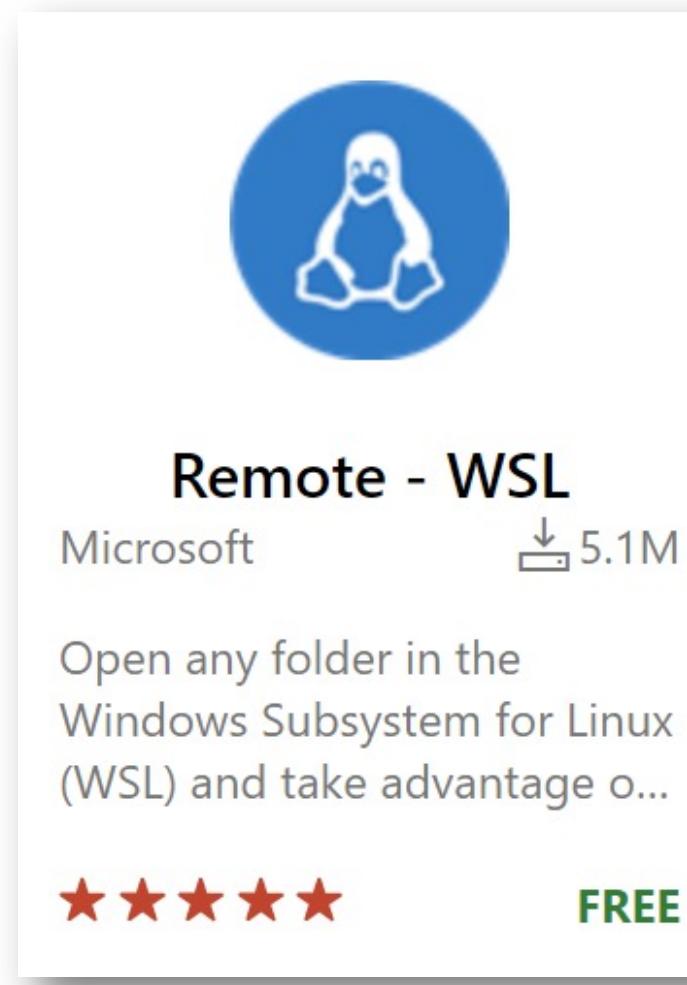
Bottom Features:

- IntelliSense (Lightbulb icon)
- Debugging (Bug icon)
- Built-in Git (Git icon)
- Extensions (Extensions icon)

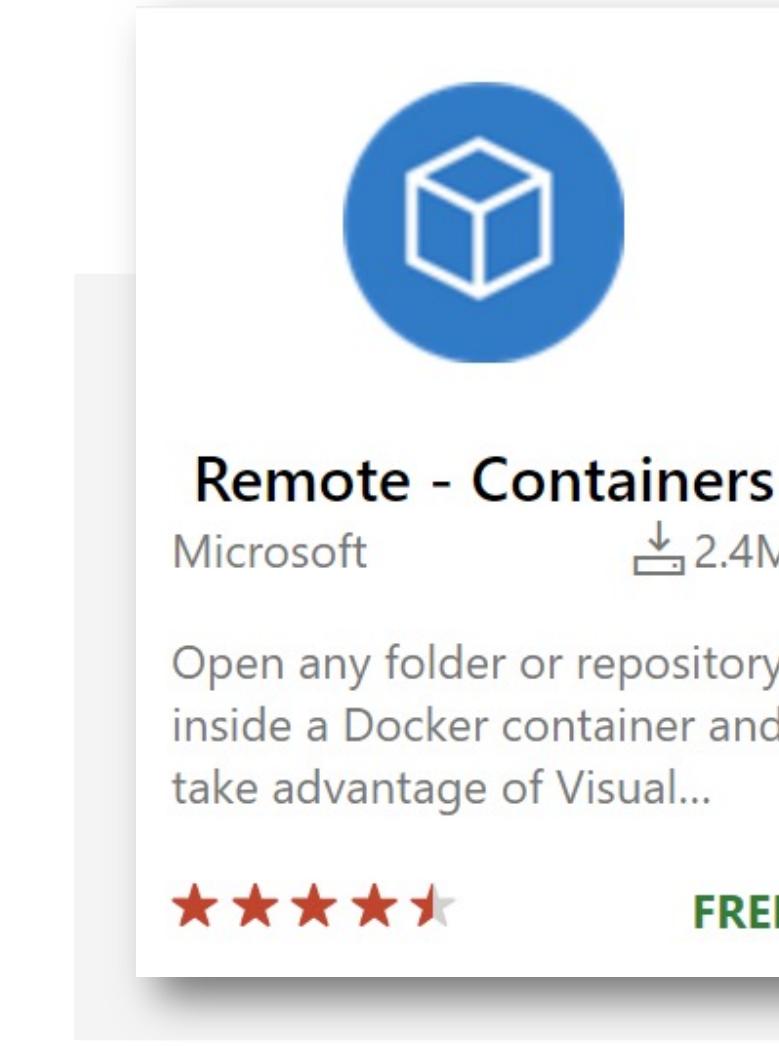
Microsoft Logo: Microsoft

2021

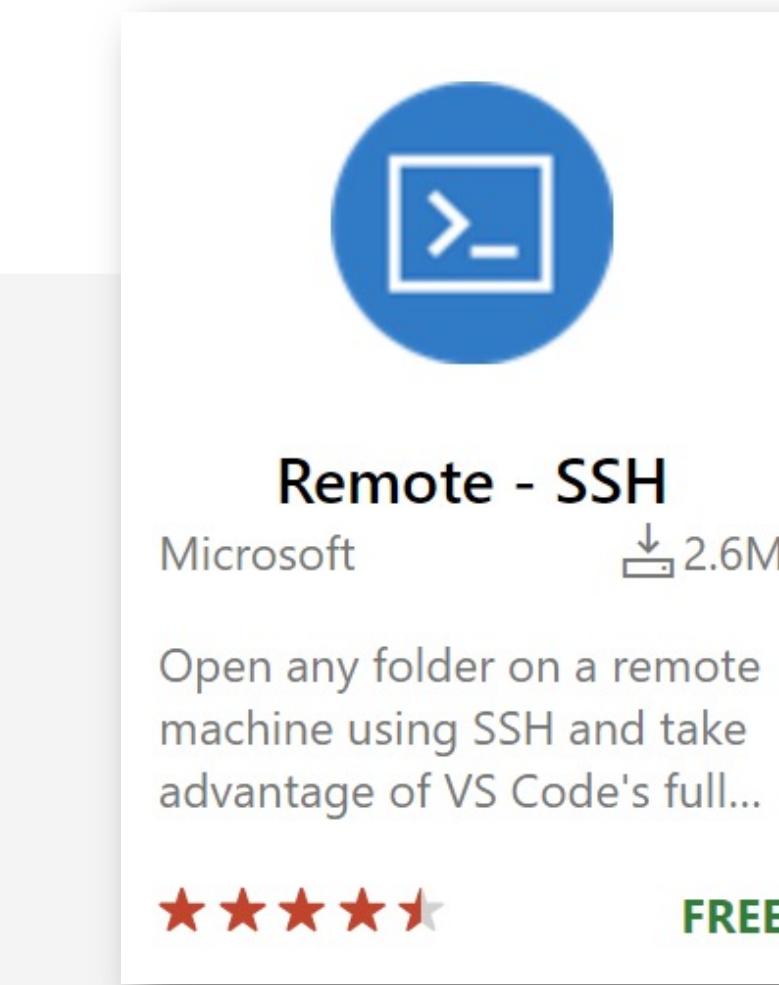
REMOTE DEVELOPMENT EXTENSIONS



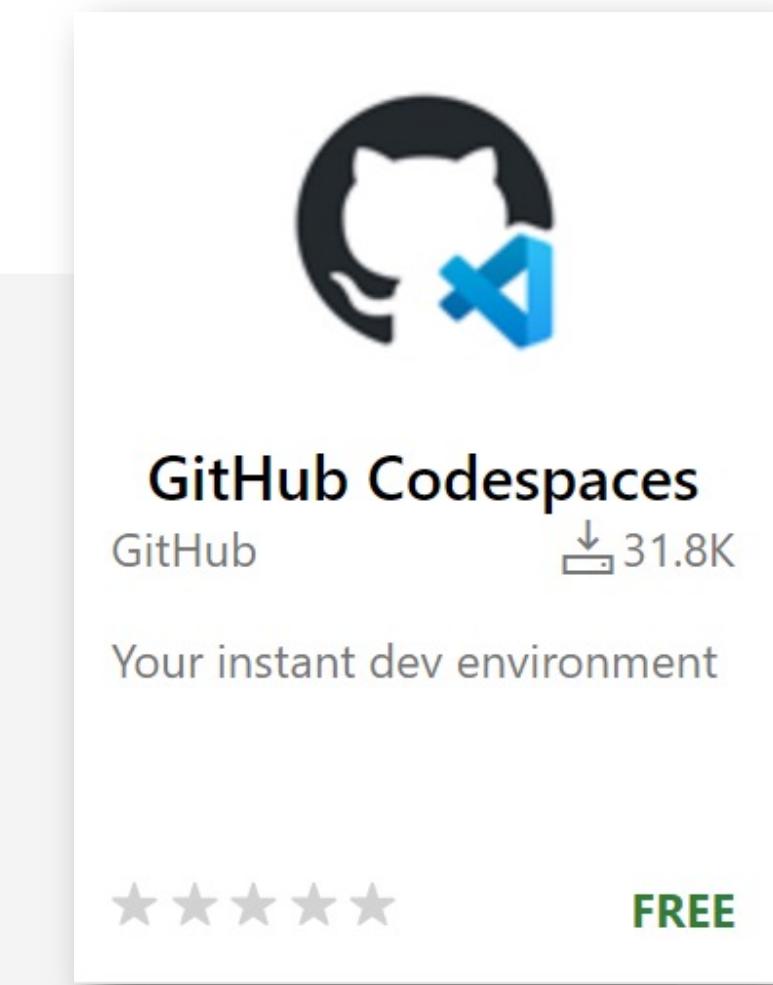
Remote-WSL



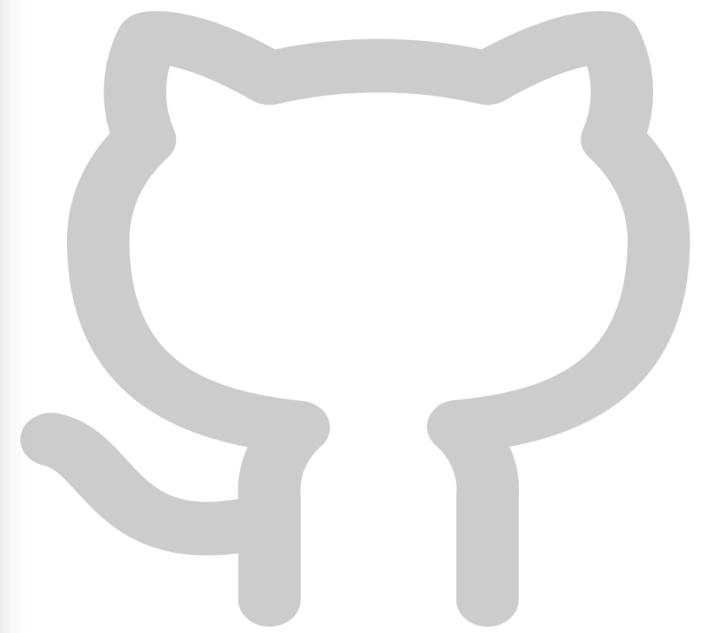
Remote-Containers



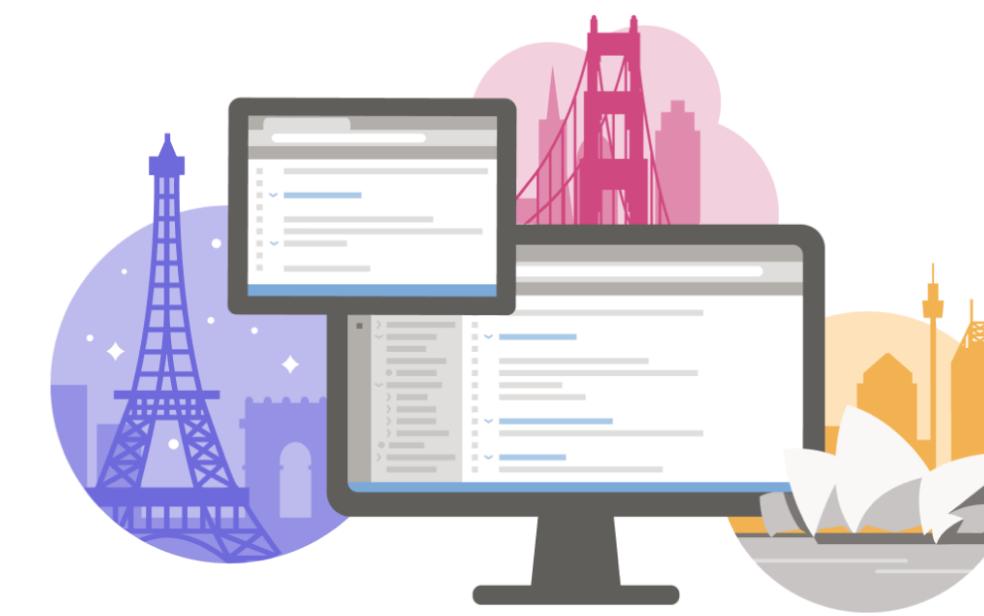
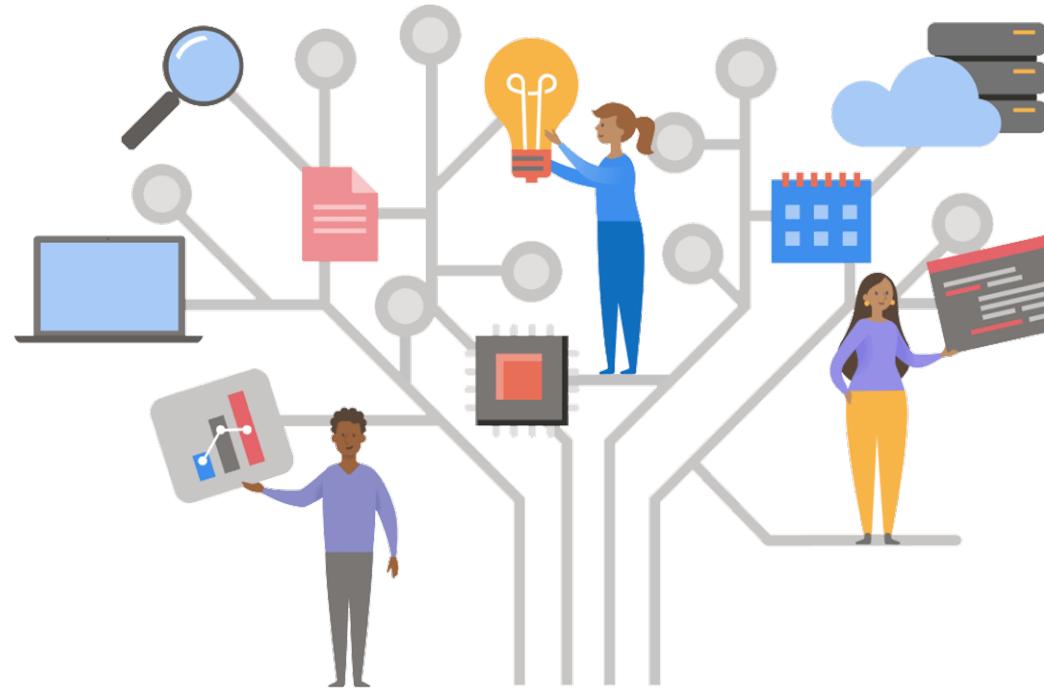
Remote-SSH



GitHub Codespaces



WHAT IS CODESPACES?



MANAGED ENVIRONMENTS FOR ANY PROJECT

Fully-managed, on-demand dev environments that dramatically reduce setup and maintenance

FEEL AT HOME EVEN WHEN AWAY

Environment requirements are auto-detected, but can be configured, and your settings, themes, Git identity, and dotfiles are roamed so the experience looks and feels like local

DEVELOP FROM ANYWHERE

Connect from anywhere: Visual Studio Code, browser-based companion editor

SUPERCHARGED COLLABORATIVE DEVELOPMENT

Fully collaborative environment, complete with AI-assisted development support

MACHINES

Create codespace for github/octocat x

Machine type

<input type="radio"/> 2 core Recommended for most projects	2 core • 4GB RAM • 32GB
<input checked="" type="radio"/> 4 core Recommended for complex projects	4 core • 8GB RAM • 32GB
<input type="radio"/> 8 core Recommended for top performance	8 core • 16GB RAM • 64GB
<input type="radio"/> 16 core Recommended for top performance	16 core • 32GB RAM • 128GB
<input type="radio"/> 32 core Recommended for top performance	32 core • 64GB RAM • 64GB

Create codespace

Product	SKU	Unit of measure	Price
Codespaces Compute	2 core	1 hour	\$0.18
	4 core	1 hour	\$0.36
	8 core	1 hour	\$0.72
	16 core	1 hour	\$1.44
	32 core	1 hour	\$2.88
Codespaces Storage	Storage	1 GB-month	\$0.07

GET STARTED

- Quickstarts

The screenshot shows a GitHub repository page for 'microsoft/vscode-remote-try-java'. The repository is described as a 'Public template' for Java development in a remote container. The 'Code' tab is selected, indicated by an orange underline. The main content area displays a list of files with update notifications:

- main: Chuxel Update to 0.195.0
- ..
- Dockerfile: Update to 0.195.0
- devcontainer.json: Update to 0.195.0

At the bottom of the page, there are links for GitHub's footer: © 2021 GitHub, Inc., Terms, Contact GitHub, and Pricing.

GET STARTED

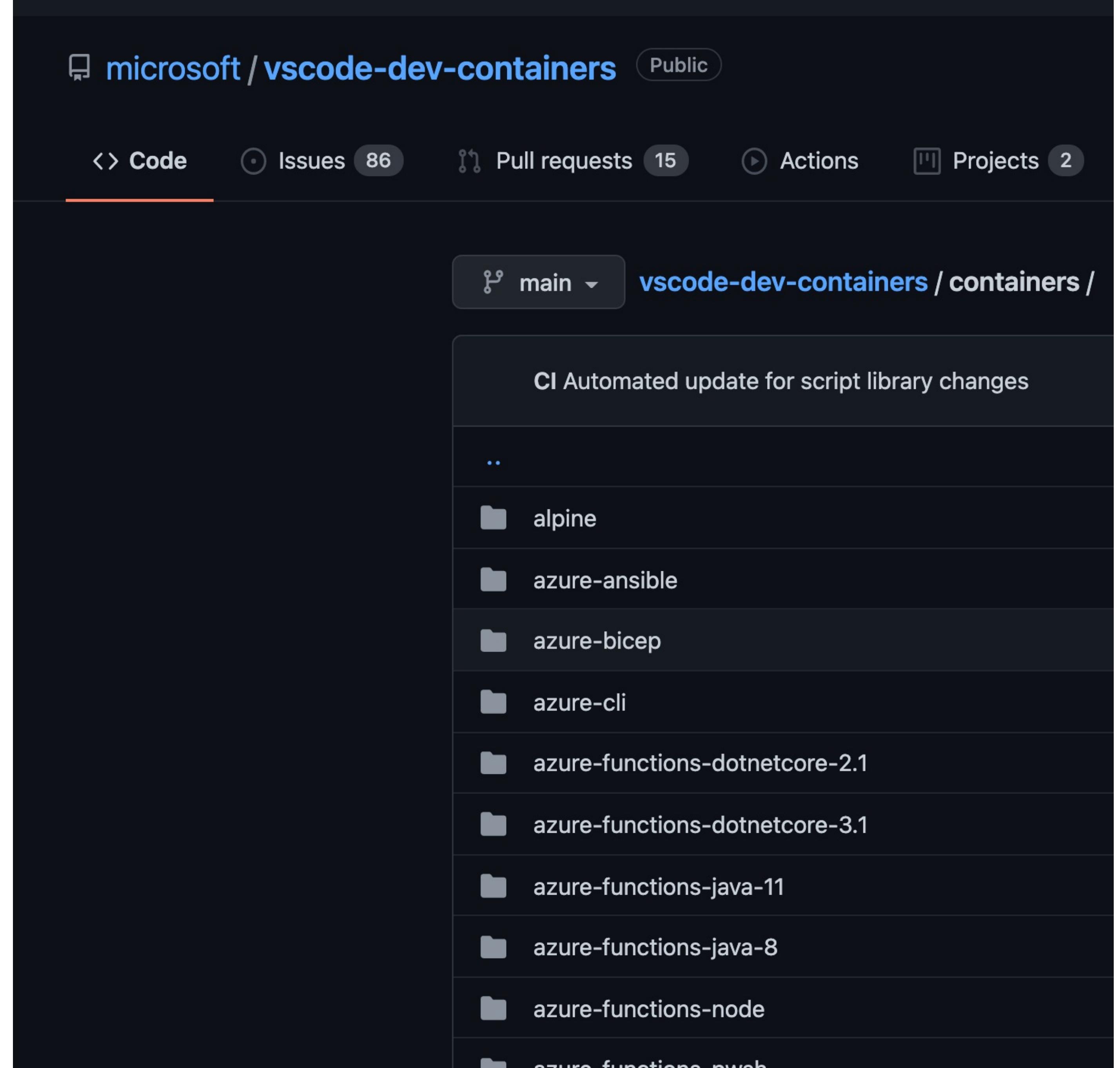
- Templates

The screenshot shows a user interface for adding development container configuration files. At the top, there's a header bar with the text '>dev container' on the left and 'Codespaces: Add **Development Container** Configuration Files...' on the right, accompanied by a gear icon. Below the header is a back arrow and the title 'Add Development Container Configuration Files'. A search bar says 'Select a container configuration definition'. A list of templates follows:

- Java**
Develop Java applications.
- Java 8**
Develop Java applications.
- Azure Functions & Java 11**
Develop Azure Functions in Java.
- Azure Functions & Java 8**
Develop Azure Functions in Java.
- GitHub Codespaces - Stretch (Legacy)**
Legacy Debian 9 / Stretch based Codespaces universal default image.
- Salesforce Project (Community)**
Salesforce Extension for VS Code supports remote development and allows you to use a d...
- Alpine**
Simple Alpine container with Git installed.
- Bazel (Community)**
Develop and compile efficiently on any language with the Bazel compilation tool.

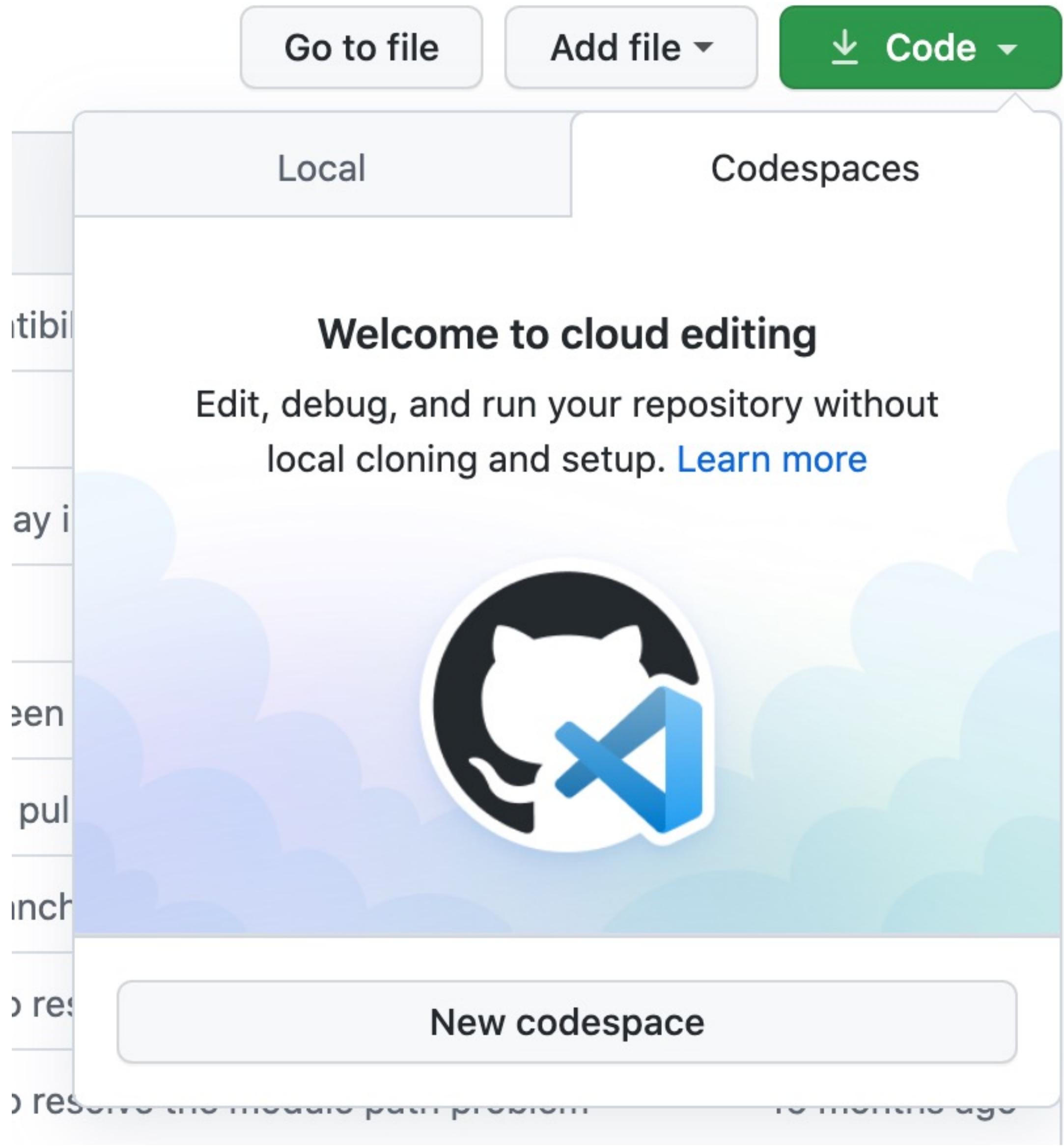
GET STARTED

- Advanced Templates

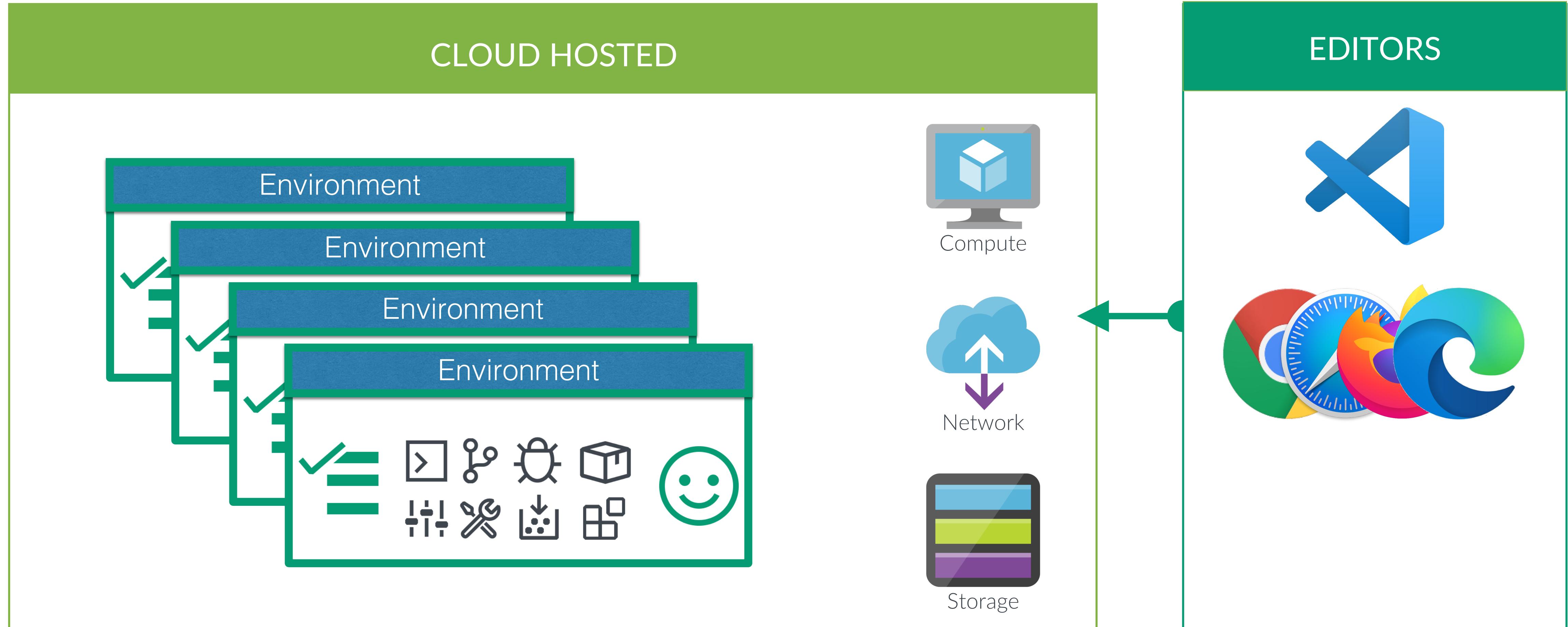


GET STARTED

- “Roll your own”

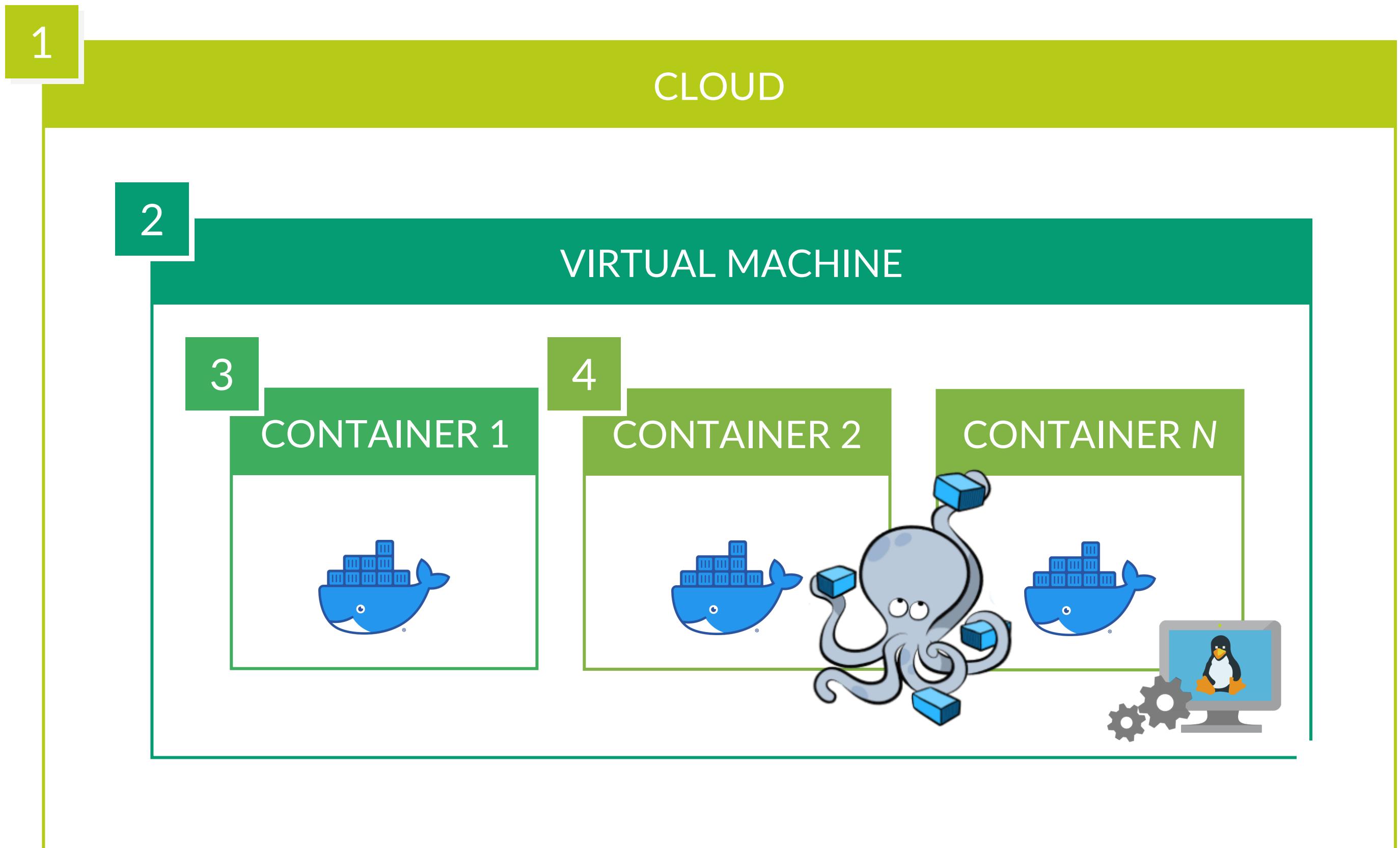


CODESPACES



COMPUTE

- 1 Cloud HOSTED**
Environments are hosted in Cloud
- 2 VIRTUAL MACHINE ISOLATION**
Each environment runs on a dedicated VM which is automatically suspended and deleted when not in use to reduce costs.
- 3 ENVIRONMENT CUSTOMIZATION**
Environments run in containers based on our [default image](#), or [your own custom Dockerfile or image](#).
- 4 COMPOSED ENVIRONMENTS**
Compose together complex development environments with [Docker Compose](#) support



NETWORK

1 NO PUBLIC IP ADDRESS

Environments are not addressable via a public IP.

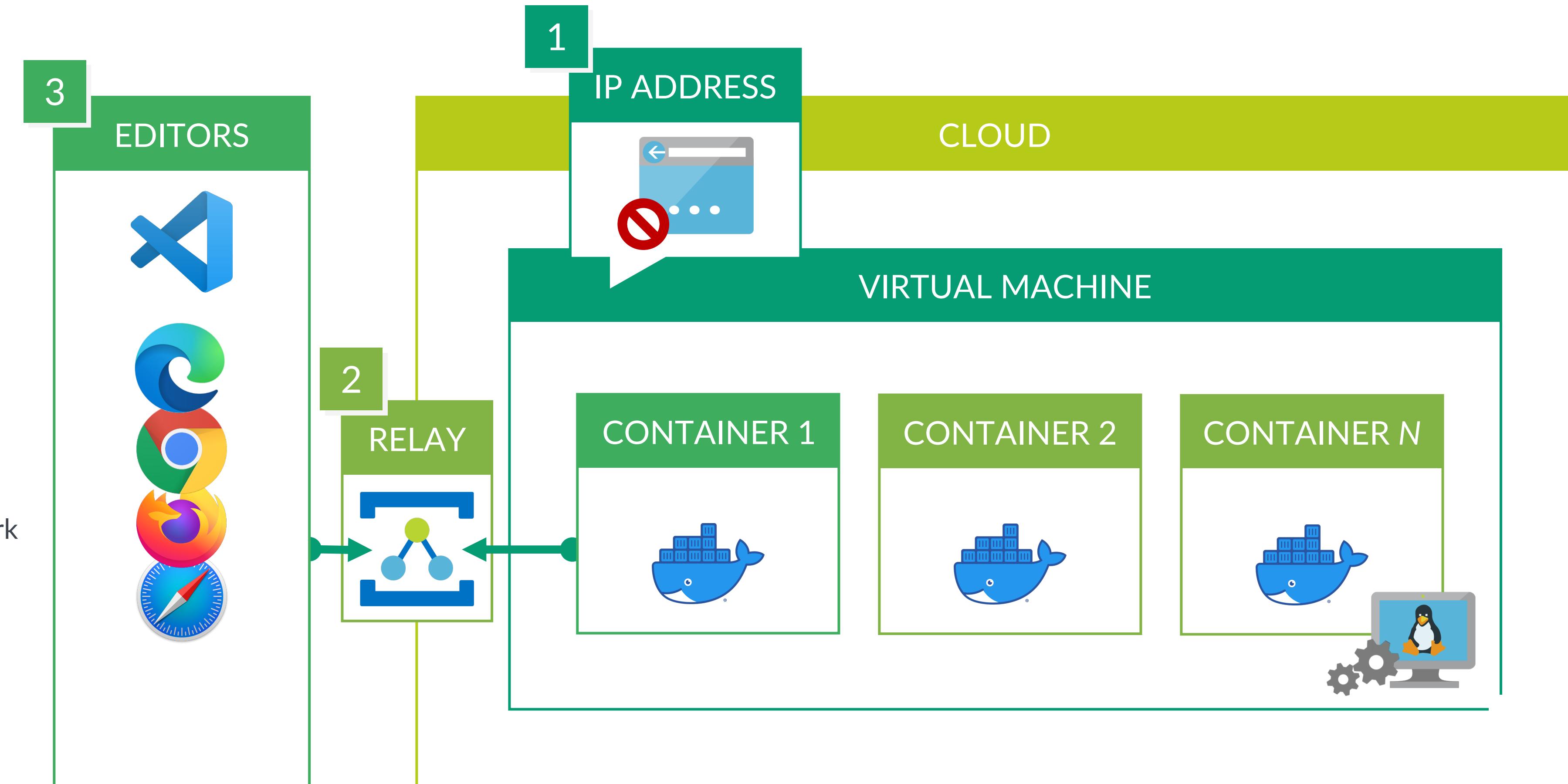
2 NO INBOUND TRAFFIC

Environments make outbound-only connections to relay and are:

- ✓ End-to-end Encrypted
- ✓ Authenticated
- ✓ Authorized

3 CONNECTED CLIENTS

Clients also leverage Relay to eliminate intrusive changes to firewall and network infrastructure.



STORAGE

1 SOURCE OF TRUTH

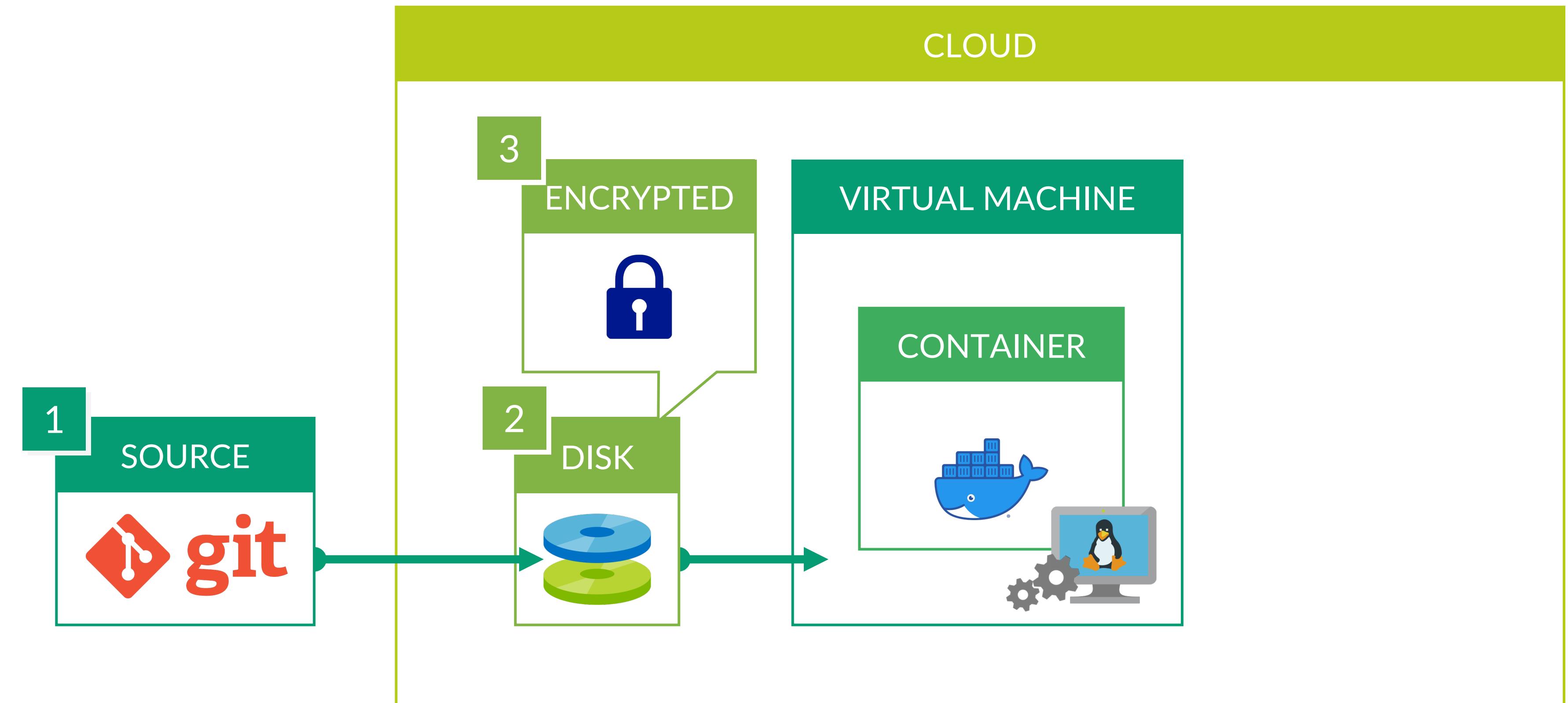
Codespaces leverages existing [Git](#) repositories as the source of truth.

2 DEDICATED STORAGE

Each environment runs in its own dedicated and isolated storage account, which is mounted to the VM when active.

3 ENCRYPTED AT REST

Environment [storage is encrypted](#) with managed keys to protect data.



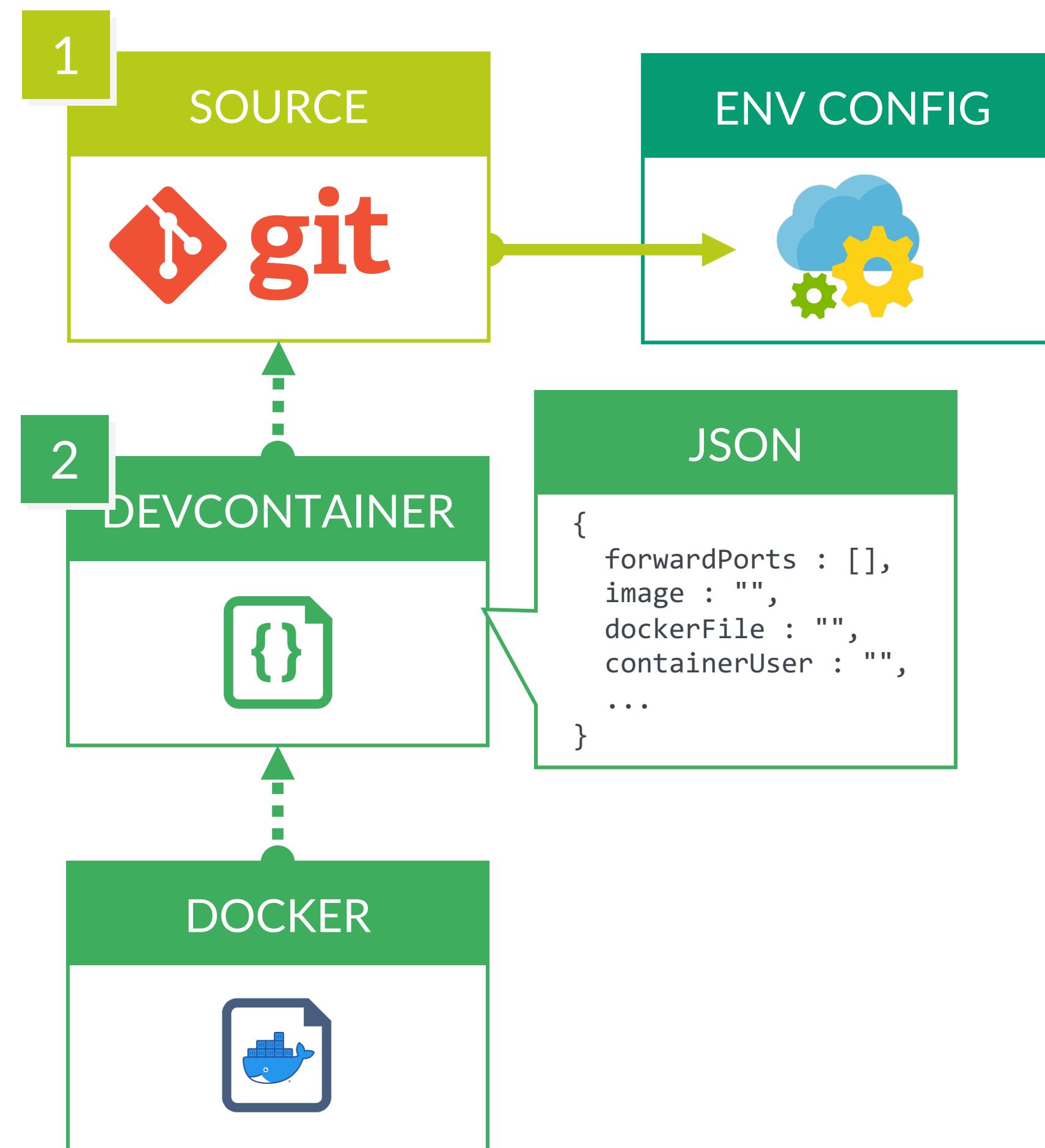
CONFIG

1 “GIT-OPS”

Configuration is versioned in Git. Git credentials are proxied from the client machine, not stored in the environment.

2 SECURE BY DEFAULT

[Configure devcontainer.json](#) to expose ports to client machine and customize environment container/user.



Devcontainer

Image



```
1   "name": "Java",
2   "build": {
3     "dockerfile": "Dockerfile",
4     "args": {
5       "VARIANT": "11-bullseye",
6       // Options
7       "INSTALL_MAVEN": "true",
8       "MAVEN_VERSION": "3.6.3",
9       "INSTALL_GRADLE": "false",
10      "NODE_VERSION": "lts/*"
11    }
12  },
13 },
14 "settings": {
15   "java.home": "/docker-java-home",
16   "maven.executable.path": "/usr/local/sdkman/candidates",
17 },
18 },
19 "extensions": [
20   "vscjava.vscode-java-pack",
21   "ms-vscode.vscode-node-azure-pack"
22 ],
23 },
24 "forwardPorts": [8080],
25 "remoteUser": "vscode"
26 }
27 }
28 }
```

Settings



Extensions



Ports



User



Docker

Image

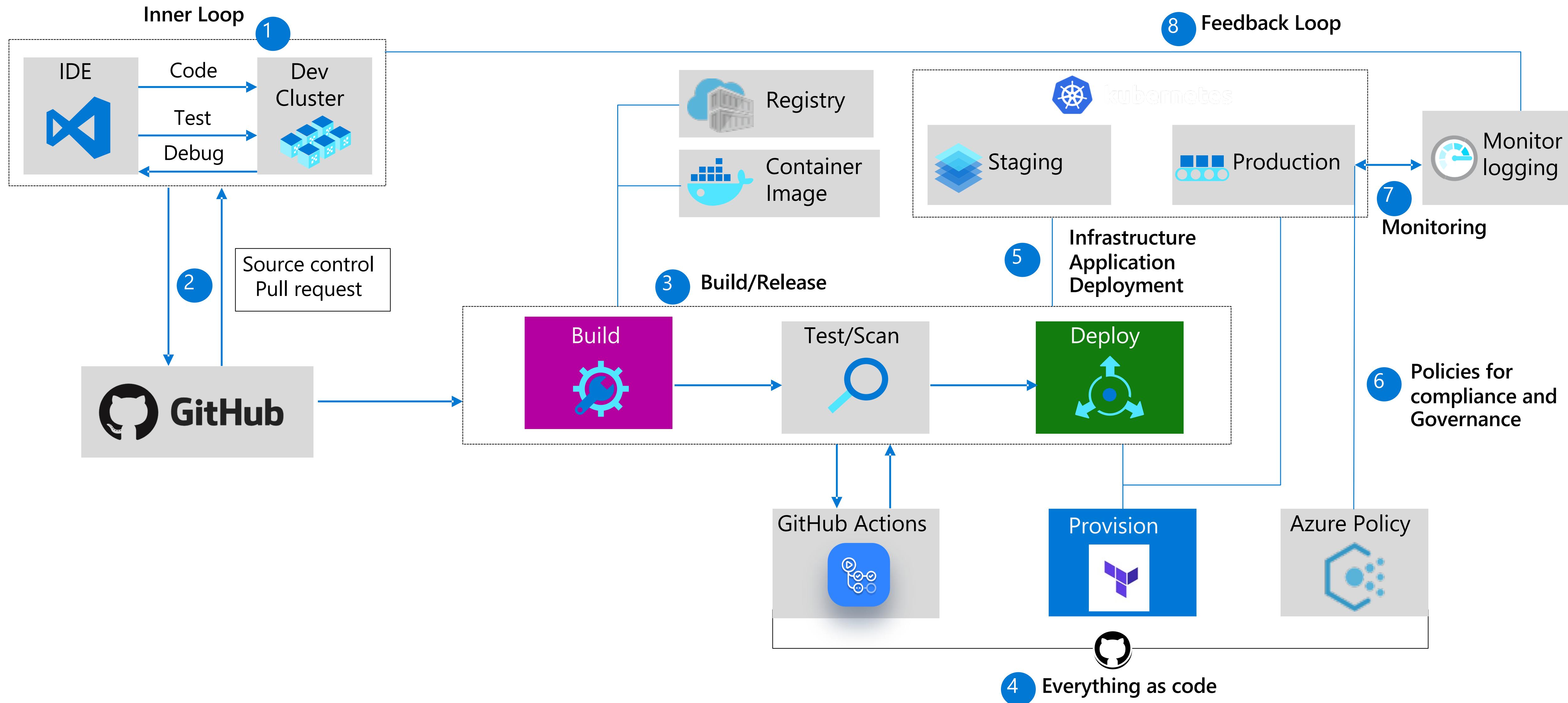


Installs



```
1 # See here for image contents: https://github.com/microsoft/vscode-dev-containe
2 # [Choice] Java version (use -bullseye variants on local arm64/Apple Silicon):
3 ARG VARIANT=11-bullseye
4 FROM mcr.microsoft.com/vscode/devcontainers/java:0-${VARIANT}
5
6 # [Option] Install Maven
7 ARG INSTALL MAVEN="false"
8 ARG MAVEN VERSION=""
9 # [Option] Install Gradle
10 ARG INSTALL GRADLE="false"
11 ARG GRADLE VERSION=""
12 RUN if [ "${INSTALL MAVEN}" = "true" ]; then su vscode -c "umask 0002 && . /usr
13             && if [ "${INSTALL GRADLE}" = "true" ]; then su vscode -c "umask 0002 && . /
14
15 # [Choice] Node.js version: none, lts/*, 16, 14, 12, 10
16 ARG NODE VERSION="none"
17 RUN if [ "${NODE VERSION}" != "none" ]; then su vscode -c "umask 0002 && . /usr
18
```

EVERYTHING AS CODE



PORTS

F
C
m

PROBLEMS OUTPUT DEBU

Port

- 3000
- Staging (4000) ⚡ × https://recour... ↴ ↴ /opt/nodejs/14.15.1/bin/node

Add Port

Open in Browser

Set Label and Update devcontainer.json

Copy Local Address ⌘C

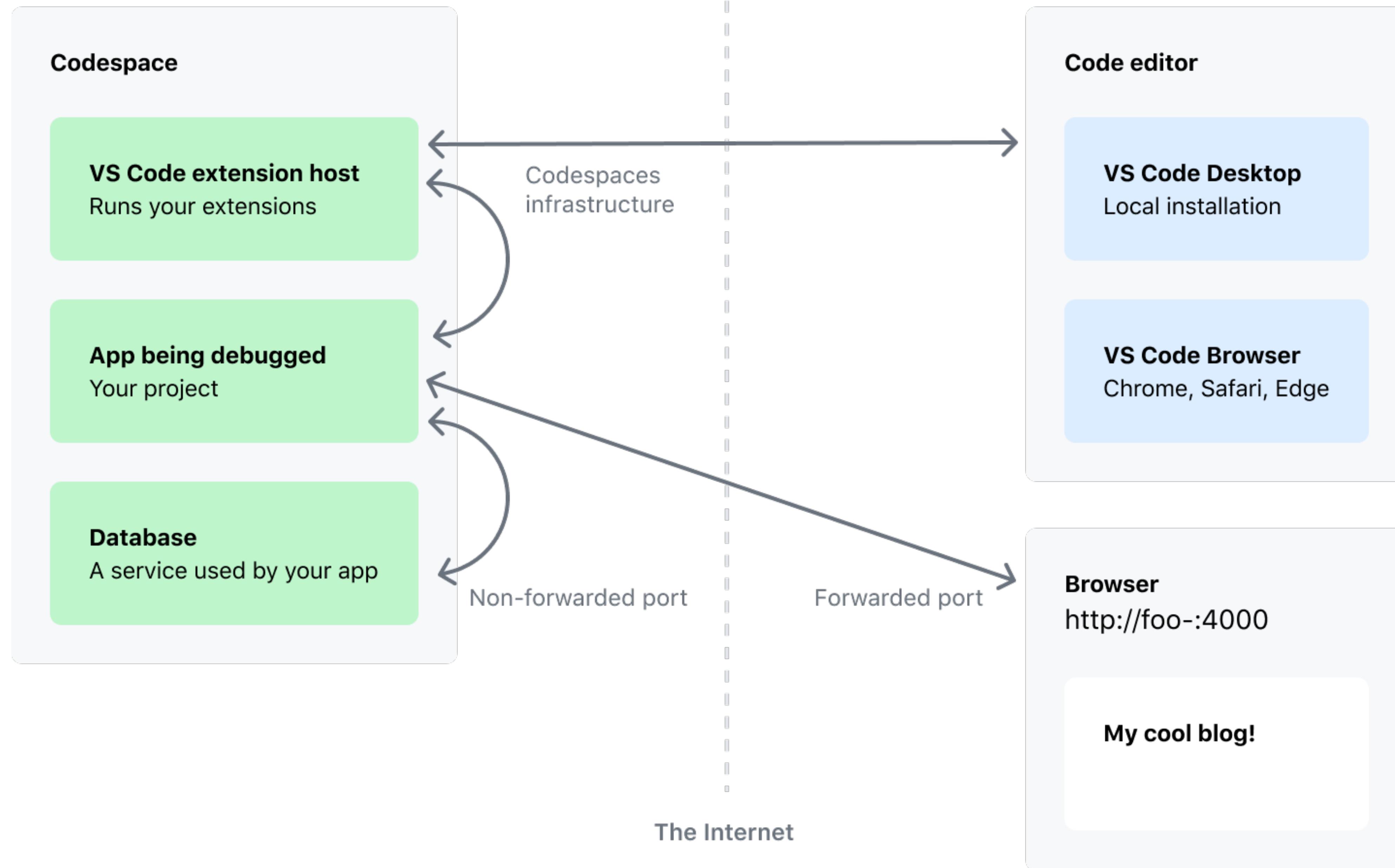
Make Public

Stop Forwarding Port ⌘Backspace

Forward a Port

The Cloud

Your device



DEMO

```
    .on("openClosed", gameClosed))
    .on("offScreen")
    .on("openClosed", seasonClosed))

    if (homeScore = homeTeamScoreQuestion,
        awayScore = awayTeamScoreQuestion,
        questionNumber = questionNumbers.next())
    {
        let model = ScorePredictionCell.Model(
            isHomeTeam: shouldGreyOutControls,
            initialHomeScore: scoreAnswer?.home ?? homeScore.defaultValue,
            initialAwayScore: scoreAnswer?.away ?? awayScore.defaultValue,
            maxHomeScore: max(homeScore.maximumValue, awayScore.defaultValue),
            minHomeScore: min(homeScore.minimumValue, awayScore.maximumValue),
            maxAwayScore: max(homeScore.minimumValue, awayScore.minimumValue),
            currentHomeScore: currentScore,
            currentAwayScore: currentScore,
            questionNumber: questionNumber
        )
        teamScoreView(model, {home: homeScore.id, away: awayScore.id}))
    }

    items --> questions.map { question -> MatchdayPredictorNextGameItem? in
        when (question) {
            is HomeTeamScoreQuestion, is AwayTeamScoreQuestion -
                return nil
            is OtherIndividualScorerQuestion<let goalscorerQuestion>:
                let selectorItems = goalscorerQuestion.players.map {
                    FirstScorerSelectorItem(
                        title: $0.name,
                        image: $0.photo,
                        playerId: $0.playerId,
                        backgroundColor: $0.roster.backgroundColor
                    )
                }
                let model = FirstScorerPredictionCell.Model(
                    isHomeTeam: shouldGreyOutControls,
                    isGoalscorerAnswer: goalscorerAnswer ?? goalscorerQuestion.preselectedPlayerOPTAId,
                    selectorItems: selectorItems,
                    ...
                )
                teamScoreView(model, {home: homeScore.id, away: awayScore.id})
            ...
        }
    }
}
```

NEXT STEPS

<https://aka.ms/codespaces-quickstart>



@RoryPreddy 

