AUGUST 30, 2021 / #GITHUB

GitHub Codespaces vs Gitpod – Full Stack Development Moves to the Cloud

Nader Dabit

Gitpod and GitHub Codespaces are cloud based developer environments that allow you to spin-up high-performance, automated dev environments in seconds.

Over the past few months I have gone down the rabbit hole of cloud-based developer

This has been especially helpful for me, as a teacher and content creator helping developers learn how to code. Now these developers can deploy the example projects without having to set up their local environments so there's one less hurdle for them if they are just getting started.

It is also an overall productivity and efficiency boost as I don’t have to clone and set up projects anymore locally when making updates. Instead, I can just click a button and it’s ready to go.

In this article I'll share my view on the ecosystem together with a comparison of the two leading options in accomplishing this – GitHub Codespaces and Gitpod.

For transparency, Gitpod is also a sponsor of my YouTube channel. That being said, I only choose to accept sponsorships from projects that I already enjoy using myself.

Cloud-based Development – This Is the Way

As developers, we like to automate things. We speed up our own workflows, automate infrastructure and CI/CD pipelines, and even create tools that write code themselves.

If you look at modern software pipelines, there is one area we have not automated: our developer environments. They are still brittle, tied to local machines, and require nerve-wracking set-up and maintenance efforts that distract us from being creative and productive.

Dev environments are a constant source of friction during onboarding and ongoing development (remember your last “works on my machine” discussion).

As a teacher and content creator helping new developers learn how to code, one of the most common issues isn't the tutorial or content itself. Rather, it's the developer's local environment not being set up properly, and it's often not even their fault.

There are countless variances of operating systems and application versions that have to be taken into account.

The pain that comes with local dev environments will only get worse over time: larger

GitHub Codespaces and Gitpod Lead the Pack of Cloud-based Dev Environments

Some years ago, Cloud9 moved into the area as a first generation of browser IDEs. Though their ideas were going in the right direction, technology and community just were not ready (yet).

A lot has changed since then. In addition to the advent and meteoric rise of VS Code, there were big leaps in container and VM technology that made it faster, more secure, scalable, and practical to run developer environments in the cloud.

No surprise that we are seeing companies like Google, Facebook, Shopify and most recently GitHub fully moving software development to the cloud with internal solutions.

There are a few players in the area that have taken advantage of those developments – although some might argue that other existing solutions focus more on prototyping and playground work for specific languages and are not a full replacement for local development.

When it comes to cloud-based dev environments that work for day-to-day professional software development, there are two main options: Gitpod and GitHub Codespaces.

So what do they do? Well, essentially they let you spin up **task-based developer environments** in the cloud from any Git context. Once you're done, you just close them.

This is a huge boost for productivity. Think about multi-track development (no changing of dev environment settings when switching contexts to review pull-requests), onboarding, consistency or just working remotely. You can work from any computer, Chromebook, or tablet.

In this article we will take a look under the hood and assess and evaluate both products according to the following categories:

- Workflow and Collaboration
- IDE

- Open Source & Ecosystem
- Availability & Pricing

Let’s dive into it!

GitHub Codespaces vs Gitpod Workflow

GitHub Codespaces and Gitpod are both services that allow you to do exactly the same as what you could do locally on a Linux machine with one major addition: your developer environment can be configured as code and hence is version controlled, reproducible and automatable.

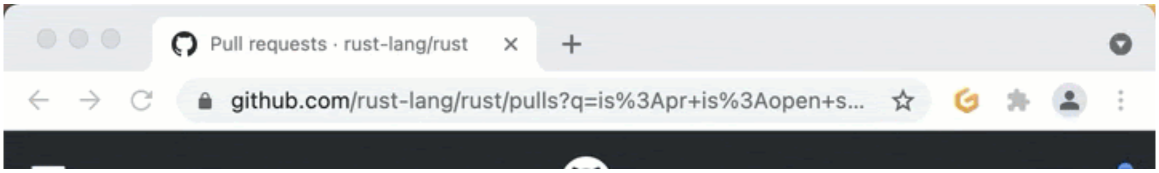
GitHub Codespaces allows you to do that with a devcontainer.json file and Gitpod with a .gitpod.yml file that you place into your repository (more on that later). Both do essentially the same thing, like define what Docker container to use, what scripts to run, and they control what extensions will be available in your Codespace (GitHub) / workspace (Gitpod).

Both products embrace a software development flow that enables developers to start a developer environment from any Git context (Issue, PR/MR, branches, and so on) with one-click for the specific task you are working on.

This task-based development flow is what ultimately sold me on switching from local dev for real world projects and, after using it, feels like it would be really hard to go back to not having it.

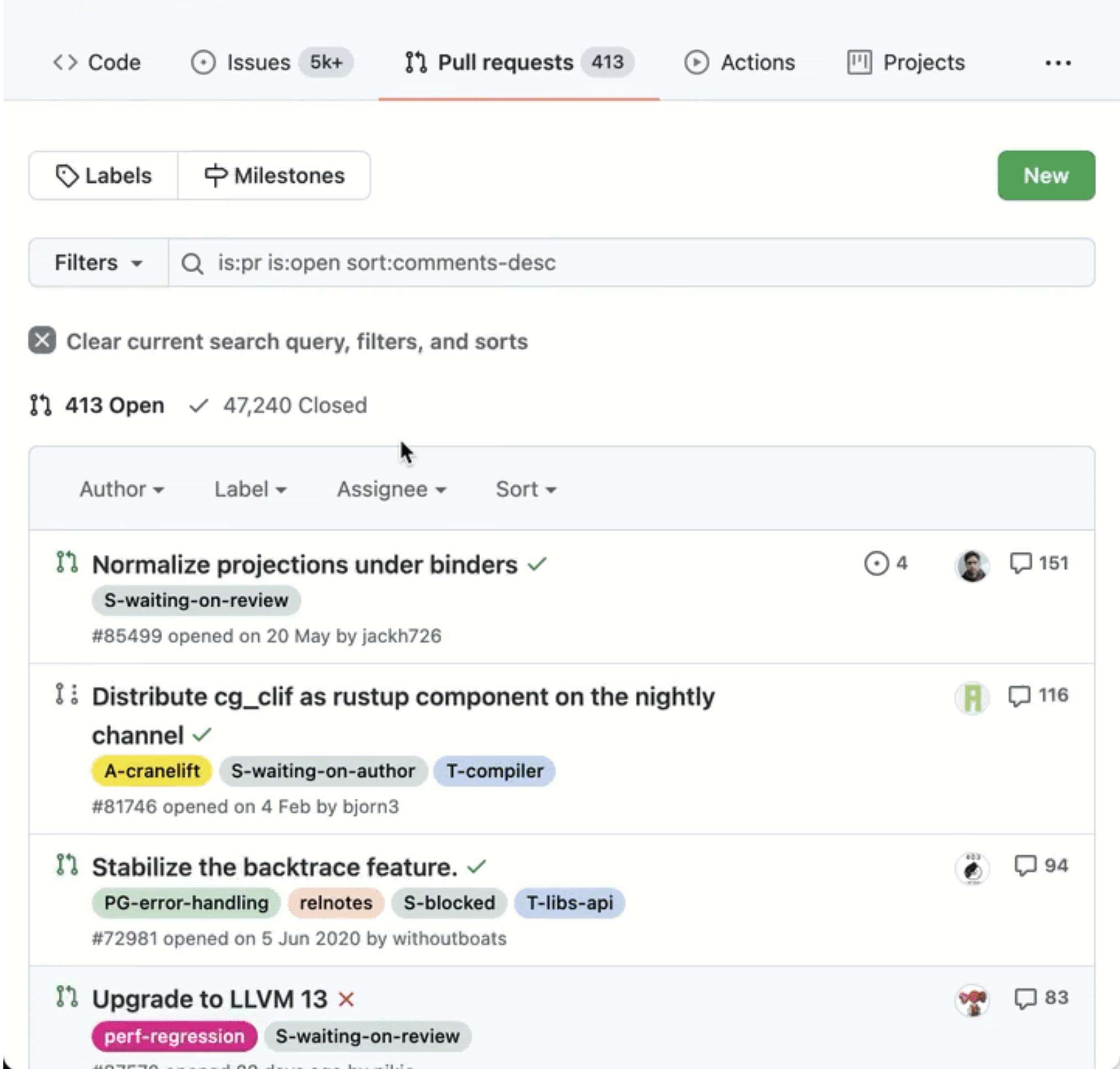
Developer environments turn just into some resource I can spin up on demand and close and forget about if I am done with my task. Gitpod is excelling at that workflow and built their product around it.

Reading the blog post from the engineering team at GitHub, it feels like they internally adapted a similar workflow, however this is not native with Codespaces right now.



Pull requests · rust-lang/rust

github.com/rust-lang/rust/pulls?q=is%3Apr+is%3Aopen+s...



is:pr is:open sort:comments-desc

Clear current search query, filters, and sorts

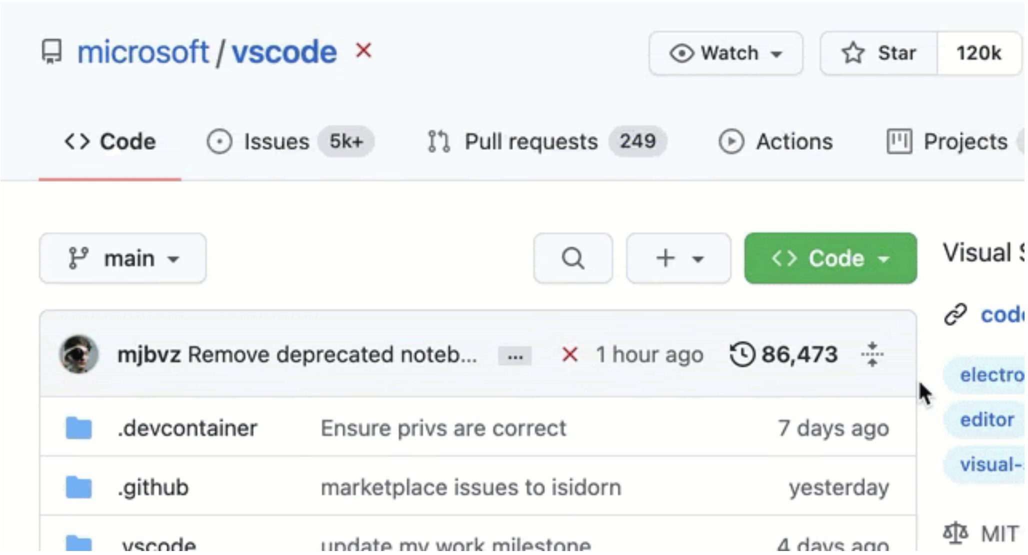
413 Open ✓ 47,240 Closed

Normalize projections under binders ✓ — 4 — 151
S-waiting-on-review
#85499 opened on 20 May by jackh726

Distribute cg_clif as rustup component on the nightly channel ✓ — 116
A-cranelift S-waiting-on-author T-compiler
#81746 opened on 4 Feb by bjorn3

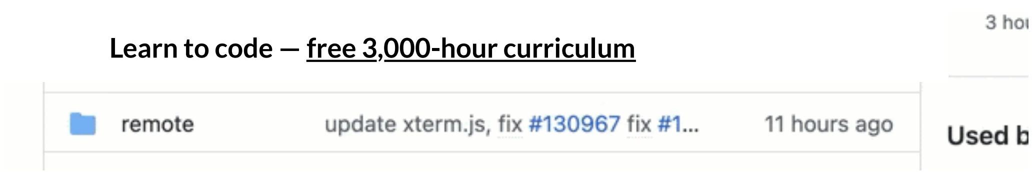
Stabilize the backtrace feature. ✓ — 94
PG-error-handling relnotes S-blocked T-libs-api
#72981 opened on 5 Jun 2020 by withoutboats

Upgrade to LLVM 13 ✕ — 83
perf-regression S-waiting-on-review



mjbvz Remove deprecated noteb... … ✕ 1 hour ago 86,473

```
.devcontainer   Ensure privs are correct        7 days ago
.github         marketplace issues to isidorn   yesterday
.vscode         update my work milestone        4 days ago
```



3 hou

remote update xterm.js, fix #130967 fix #1... 11 hours ago

Used b

Collaboration with Codespaces vs Gitpod

Both GitHub Codespaces and Gitpod offer ways to collaborate, but they differ in their approach.

On both platforms, when your development environment is running, you can expose any TCP port publicly or privately to the internet. This enables workflows where you can share links to a web-server or API server as standard URLs.

Gitpod offers functionality to share a snapshot of the workspace with a co-worker but GitHub Codespaces does not.

GitHub Codespaces offers the ability to interactively pair program when the LiveShare extension is installed while Gitpod offers the ability to share the workspace itself with collaborators.

Once Gitpod enables local VS Code support (1-2 weeks from this writing) you can also use LiveShare with Gitpod.

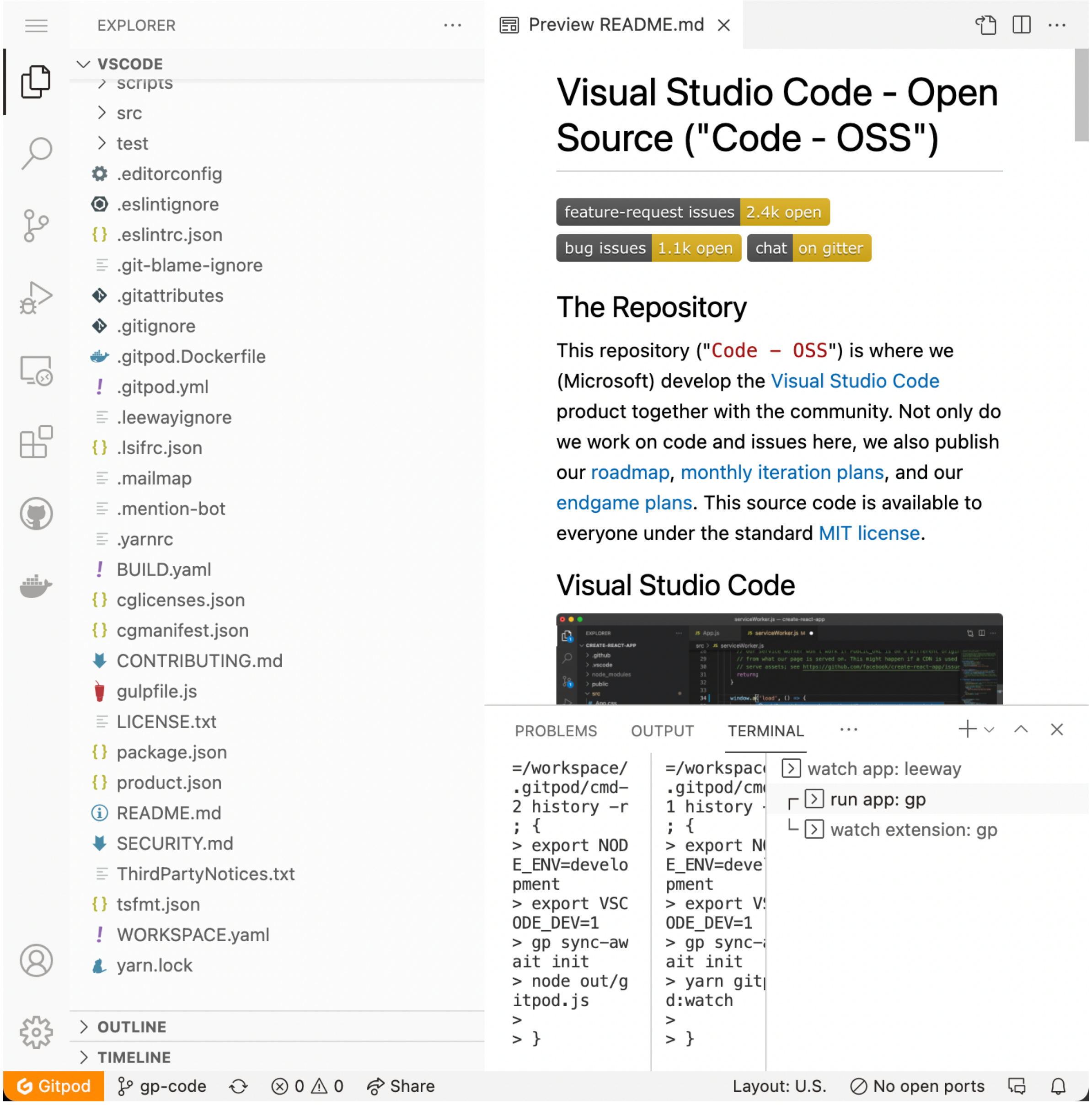
Here are some further resources where you can read more about this.

- https://www.gitpod.io/blog/i-said-goodbye-to-local-development-and-so-can-you#develop-a-new-feature
- https://github.blog/2021-08-11-githubs-engineering-team-moved-codespaces/
- https://visualstudio.microsoft.com/services/live-share/

Gitpod vs Codespaces IDE

Both Gitpod and Codespaces ship upstream, and stock VS Code as their default IDE running in the browser. It feels like, acts like, and is literally the same VS Code that you are used to from your desktop. Be aware of cmd+W however :).

The team behind Gitpod has a long background in open-source developer tooling, and they initially created Theia (which is also based on VS Code). |But recently, they switched to stock VS Code and maintain a very lightweight fork of VS Code that has also continued to gain adoption by other teams in the industry (https://github.com/gitpod-io/vscode/).



Visual Studio Code - Open Source ("Code - OSS")

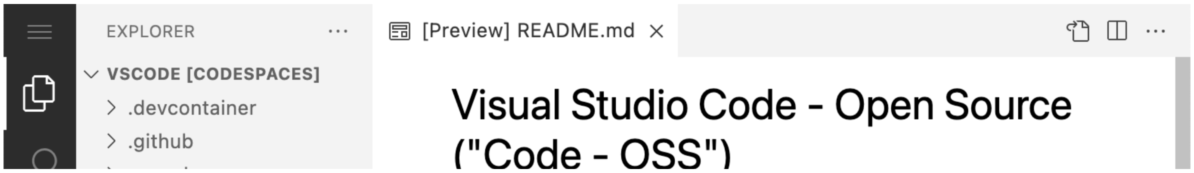
The Repository

This repository ("Code - OSS") is where we (Microsoft) develop the Visual Studio Code product together with the community. Not only do we work on code and issues here, we also publish our roadmap, monthly iteration plans, and our endgame plans. This source code is available to everyone under the standard MIT license.

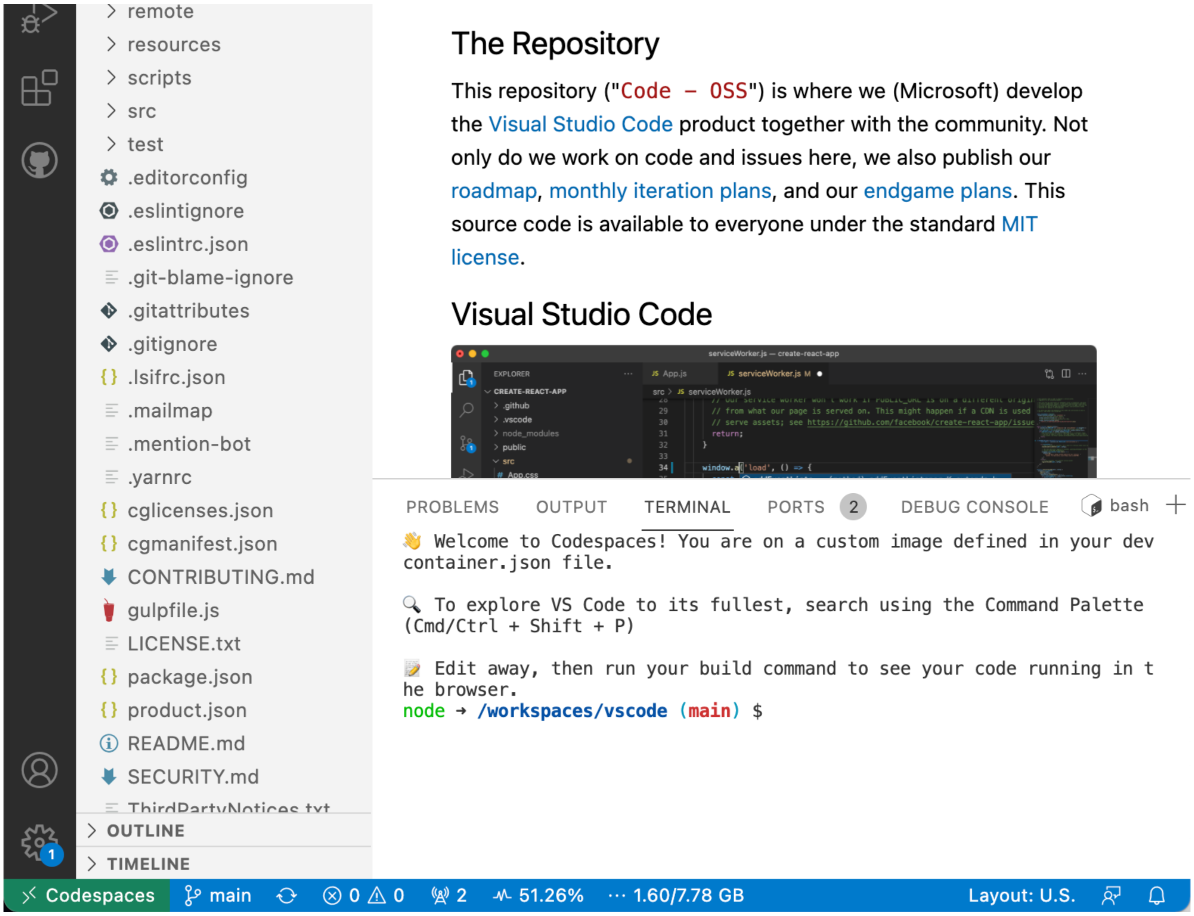
Visual Studio Code

```
=/workspace/
.gitpod/cmd-
2 history -r
; {
> export NOD
E_ENV=develo
pment
> export VSC
ODE_DEV=1
> gp sync-aw
ait init
> node out/g
itpod.js
>
> }
```

```
=/workspace
.gitpod/cm
1 history 
; {
> export N
E_ENV=deve
pment
> export V
ODE_DEV=1
> gp sync-
ait init
> yarn git
d:watch
>
> }
```



Visual Studio Code - Open Source ("Code - OSS")



The Repository

This repository ("Code - OSS") is where we (Microsoft) develop the Visual Studio Code product together with the community. Not only do we work on code and issues here, we also publish our roadmap, monthly iteration plans, and our endgame plans. This source code is available to everyone under the standard MIT license.

Visual Studio Code

```
👋 Welcome to Codespaces! You are on a custom image defined in your dev
container.json file.

🔍 To explore VS Code to its fullest, search using the Command Palette
(Cmd/Ctrl + Shift + P)

📝 Edit away, then run your build command to see your code running in t
he browser.
node ➜ /workspaces/vscode (main) $
```

VS Code Extension Marketplace for Codespaces and Gitpod

Having the VS Code team behind the product and being able to access all proprietary extensions (like Liveshare, for example) through the Microsoft-controlled Visual Studio Marketplace is a plus for Codespaces.

As a response, Gitpod created https://open-vsx.org/ (now hosted under the Eclipse Foundation), which is a vendor-neutral marketplace run by the Eclipse Foundation for VS Code extensions that is accessible through Gitpod.

Though there is almost extension parity for the most popular VS Code extensions, Microsoft's (excellent) proprietary extensions have not found a way to OpenVSX.

In case you don't find an open-source extension on OpenVSX, you can trigger the publish

Remote Development from Desktop VS Code

I am a big fan of being able to connect from my local Desktop VS Code into a dev environment running on somebody else's computer.

Both products offer that feature, but the flow that Codespaces ships out of the box is superior compared to the approach from Gitpod.

While you can achieve the same with Gitpod's local app, the setup involves more work and friction from a user's perspective. Asking the Gitpod team about this, I received feedback that they will release a similar one-click experience at the end of August.

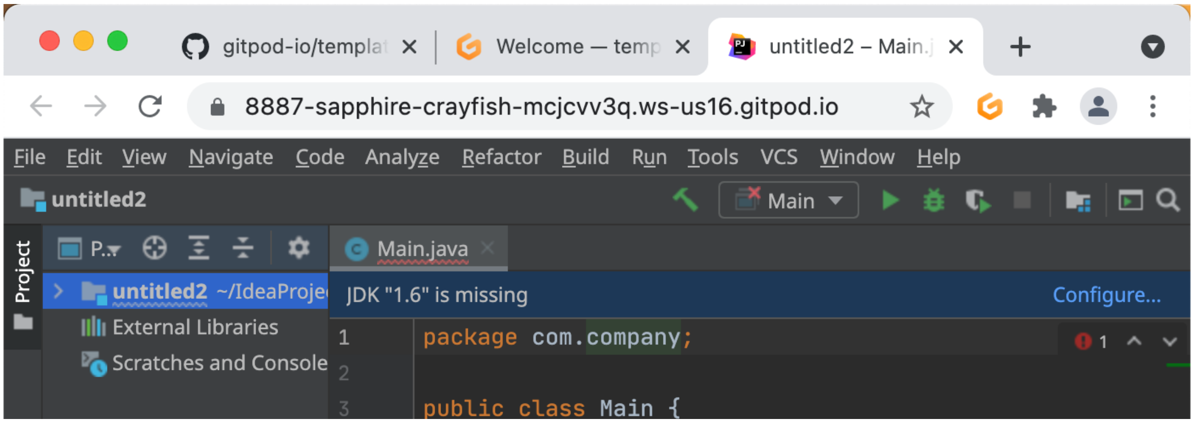
What about other IDEs than VS Code?

For GitHub Codespaces I could not find any information for other IDEs than VS Code. And given that Microsoft is behind both projects, I'd expect that they'll probably focus on VS Code.

In contrast, Gitpod tries to stress that what they have built is, on an architectural level, IDE independent and allows you to run any IDE image you can run remotely in their container.

I found some templates in their GitHub Repo that allow you to run the Jetbrains product fleet (based on https://github.com/JetBrains/projector-installer).

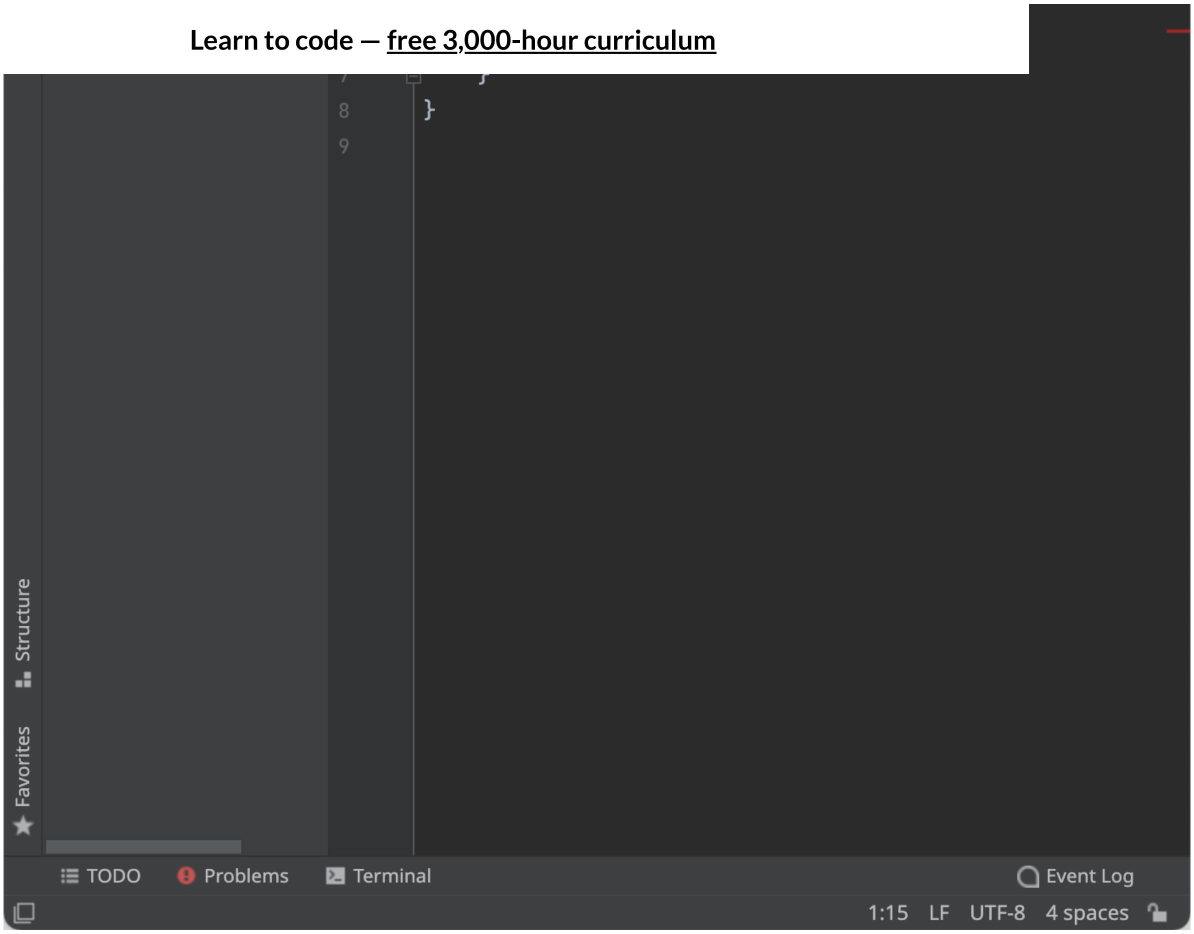
The developer experience there still feels a bit clunky. Digging a bit deeper I found this interesting discussion around one of the most requested features from the Jetbrains community that would enable remote support via SSH out of the box. I would love to see that supported by Gitpod and Codespaces.



JDK "1.6" is missing Configure...

```
package com.company;

public class Main {
```



```
}
```

Here are some resources for further reading:

- https://www.theregister.com/2021/04/08/gitpod_talks_up_importance_of/
- https://www.gitpod.io/blog/local-app
- https://www.gitpod.io/docs/integrations/jetbrains

Codespaces vs Gitpod Runtime

GitHub Codespaces runs on virtual machines which come with great isolation out of the box and are easier to provision and manage. However, they also carry the overhead of the full operating system, making them larger and slower to start as well as more costly.

Gitpod runs on lightweight containers that spin up quickly and enjoy a far higher cloud density (as more processes can run in parallel on the same underlying hardware, there is less idle compute). This allows Gitpod workspaces to spin-up faster and be more resource – that is cost and energy – efficient.

Before last year, sudo-rights (and with that docker-in-docker) were not possible in Gitpod workspaces. Earlier this year they implemented namespace isolation features that made both of those things possible (see their head of engineering explaining how they achieved that).

Both Codespaces and Gitpod support docker-in-docker for Docker Compose scenarios and nested virtualization which enables running operating systems or appliances in your browser.

Here are some more resources for you on this:

- https://www.gitpod.io/blog/root-docker-and-vscode
- https://github.com/gitpod-io/template-nixos

Automation with Codespaces and Gitpod

Both heavily embrace the notion of dev environment as code that Vagrant first coined and Gitpod then further built out.

The basic idea behind that is applying ideas from Infrastructure as Code to dev environments. Codespaces uses a devcontainer.json format as a configuration file while Gitpod uses a .gitpod.yml.

Given the reach and distribution power of Codespaces and VS Code, my prediction is that the devcontainer.json format will win long-term as the industry standard for configuring developer environments. As listed on their roadmap, Gitpod plans to support that as well. Right now they don't and run on a .gitpod.yml.

Creating and setting up your dev-env-as-code is only the first step towards fully automated per-task developer environments. As you do not want to wait for dependencies to download and code to build every time you start a developer environment, the workspaces/codespaces need to be prebuilt before you even start.

Gitpod supports prebuilds for that (think about them as a CI/CD server where Gitpod prebuilds the full workspace / runs the automation on every commit to Git). Automation y they already use them. Therefore it should be a matter of months until they also publicly release that feature.

You can read more here:

- https://www.gitpod.io/screencasts/continuously-prebuild-your-project
- https://github.blog/2021-08-11-githubs-engineering-team-moved-codespaces/

Codespaces and Gitpod Open Source and Ecosystem

The biggest difference between the two is that Gitpod is open source and everyone can contribute to the project. You can self-host it on GKE, EKS, and vanilla Kubernetes.

Their roadmap and development work is public and everyone can contribute to the project. In addition to this, maintainers and core contributors to open-source projects get Gitpod for free.

With GitHub, Microsoft owns the market leading social platform for developers and they integrated GitHub Codespaces as a first-class citizen into the developer experience on GitHub. This means that by default the UI in GitHub shows you an Open in Codespaces button, which nicely embeds into your development workflow.

For Gitpod to achieve the same level of integration you need to download either the browser extension or a bookmarklet. GitLab has a strategic partnership with Gitpod and has on every repository and merge request an “Open in Gitpod” button built into the GitLab UI.

Contrary to Microsoft, Gitpod follows a strategy where it is neutral by default, not owned by big tech and designed in a way to integrate with whatever tooling developers want to use.

This means it not only works with GitHub but also other Git providers such as GitLab and Bitbucket and you can deploy it on your own infrastructure.

Availability, Pricing, and Specs of

Gitpod has been available for more than 2.5 years, while GitHub Codespaces came out of beta on the 11th of August 2021 for customers with a GitHub Team or GitHub Enterprise subscription.

Based on a tweet from their CEO, Nat Friedman, we can expect that individual developers will have access to GitHub Codespaces at the end of the year.

GitHub Codespaces is currently free for all organizations on a GitHub Team or GitHub Enterprise subscription until September 10th. The billing increment is for the amount of minutes a workspace is active, and for the amount of storage used on disk for each workspace until a user deletes the workspace.

You can read more about that here.

Gitpod is free for public and private repositories for 50 hours per month. Maintainers and core contributors of well-established open-source projects can apply for a voucher that upgrades their account to unlimited hours per month.

Read more about Gitpod pricing in these resources:

- https://www.gitpod.io/blog/cloud-based-development-for-everyone
- https://www.gitpod.io/pricing
- https://www.gitpod.io/docs/professional-open-source

To simplify comparing the offerings from GitHub Codespaces and Gitpod, here are three different scenarios using the same amounts of CPU, memory, and storage for each use case.

	Product Manager	Developer	Power Developer
Working hours	2h per day, 21 working days per month	5h per day, 21 working days per month	10h per day, 21 working days per month.
Total Hours per month	42 hours	105 hours	210 hours
Amount of CPUs	8 cores	8 cores	8 cores
Amount of Memory	12GB	12GB	12GB

Total Cost on GitHub Codepsaces	$30.24 USD/month compute and $2.24 USD/month in additional storage fees (first 32GB per workspace is free)	$75.6 USD/month compute and $2.24 USD/month in additional storage fees (first 32GB per workspace is free)	$151.2 USD/month compute and $2.24 USD/month in additional storage fees (first 32GB per workspace is free)
Total Cost on Gitpod	$9 USD/month	$25 USD/month	$39 USD/month

The amount of CPU and memory for each workspace launched is configurable in GitHub Codespaces and they offer SKUs of up to 32 CPU cores and 64GB of RAM. Gitpod at this time does not yet offer a way to have more or less compute resources.

See it in action

I hope you're as excited as I am after first hearing about the concept of dev environments as code in the cloud! Give both a try and see for yourself how you like them.

Right now I prefer Gitpod as it's free for my use cases, has pre-builds, is accessible to anyone who wants to use it, is open source, and can be used with any Git provider including GitHub.

How to Get Started with Gitpod:

- Prefix any GitLab, GitHub or Bitbucket URL with gitpod.io/# to dive right in
- Or use their browser extension or browser bookmarklet for starting workspaces from any git context
- Or try their quickstart templates
- Check out my guides here and here

How to Get Started with Codespaces:

- If you're a GitHub Teams or Enterprise customer, look for an "Open in GitHub" button next to a repo
- Individual developers need to wait until it launches more widely
- Check out this video for an overview.

Nader Dabit

Software developer, author, Developer Relations Engineer at Edge & Node

If you read this far, tweet to the author to show them you care. Tweet a thanks

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers. Get started

Trending Guides

R Value in Statistics

What is Economics?

Module Exports

Python VS JavaScript

Model View Controller

React Testing Library

ASCII Table Chart

ADB

MBR VS GPT

Debounce

Helm Chart

80-20 Rule

OSI Model

HTML Link Code