

---

## Checklist

### Review

- ☐ Share with Ben, Geoff,
  - ☒ Ben review
  - ☐ Ben second review
  - ☒ Geoff review
- ☐ Publish on Coder Blog
- ☐ Share on social media

### Readability

- ☐ Passes [a Grammarly check](#)
- ☒ [Flesch Kincaid Reading Ease](#)  $\rightarrow \geq 60$

---

## Research

### List of discussions

- [Can I share a code-server instance for multiple, isolated users?](#)
- [Can you have multiple users](#)
- [Many solutions](#)
- [3 users share a machine](#)

### Reasons

- *would really be a hassle to have to have a different VM for each code-server instance*
- *Multiple users*
- *how it's possible to maintain multiple concurrent connexions to multiple code environments*
- *To save on costs and provisioning time our naive solution would involve a set of dockerized code-server containers behind a load balancer on kubernetes that all users access*
- *Let multiple different users work on my code-server, without having conflicting files*

- *want to have my own editor instance in my browser and my friend to have his own editor instance in his browser*
  - *I want to have a single access domain, for example code.domain.com*
- 

[add cover image here]

# Using code-server with Multiple Users

**Slug:** code-server-multiple-users

## Summary

In this article, we'll explore various solutions to using code-server with multiple users. Think of this as a high-level overview of real-world approaches to solving this problem. All solutions are based on real things we've seen from the community.

## From Single Player to Multiplayer

When [code-server](#) was launched back in 2019, we had one goal in mind: bring VS Code to the web. Since then the project has grown beyond what we could have imagined. One of those ways is using code-server with multiple users.

While we've developed our [own solution](#) to this problem, I want to explore all the solutions we've seen implemented by the community. The goal is to provide a holistic overview of each so you can decide which suits the needs of you and your team.

## Show me the code

If you want to jump straight into code examples, take a look at these code-server setups for teams:

- [Global search via Sourcegraph for code-server](#)
- [code-server instance manager](#) and [router](#)

## Solutions for Multiplayer Mode

As a reminder, code-server itself is designed for single-player mode. You get a machine, you install code-server via a package manager or the release binaries. You start it up and boom, you have your IDE in a browser. This drastically reduces the number of tools needed for your dev workflow to only two:

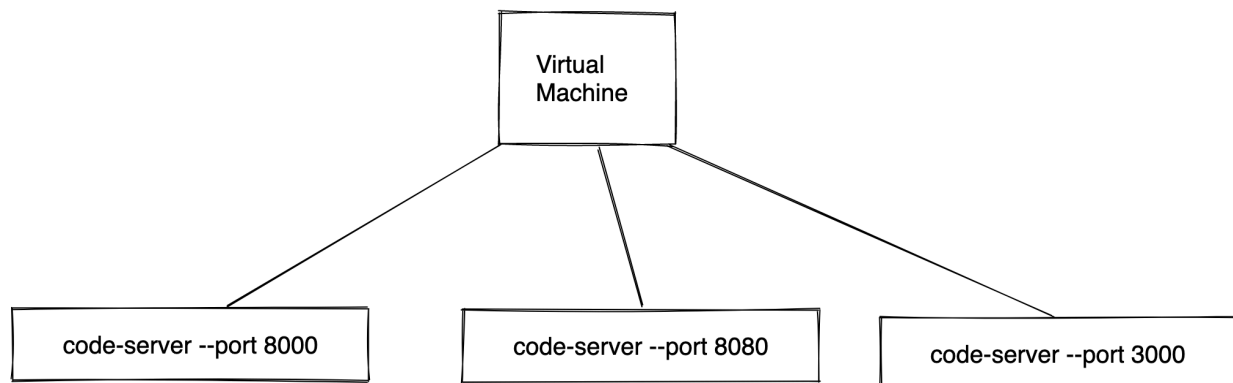
1. an internet connection
2. a web browser

To add more users, we now need a solution. Over the years, we've seen many discussions come up around this problem. Let's explore the ones that folks have shared with us and the pros and cons of each.

### Running Multiple code-server Instances on Different Ports

This is by far the simplest approach we've seen used. Your team has one machine with code-server installed. For each user, you run their own instance of code-server on different ports. Here's a diagram with more details:

### One Machine, Different Ports



#### Pros

- Only requires one machine
- Configuration is simple

#### Cons

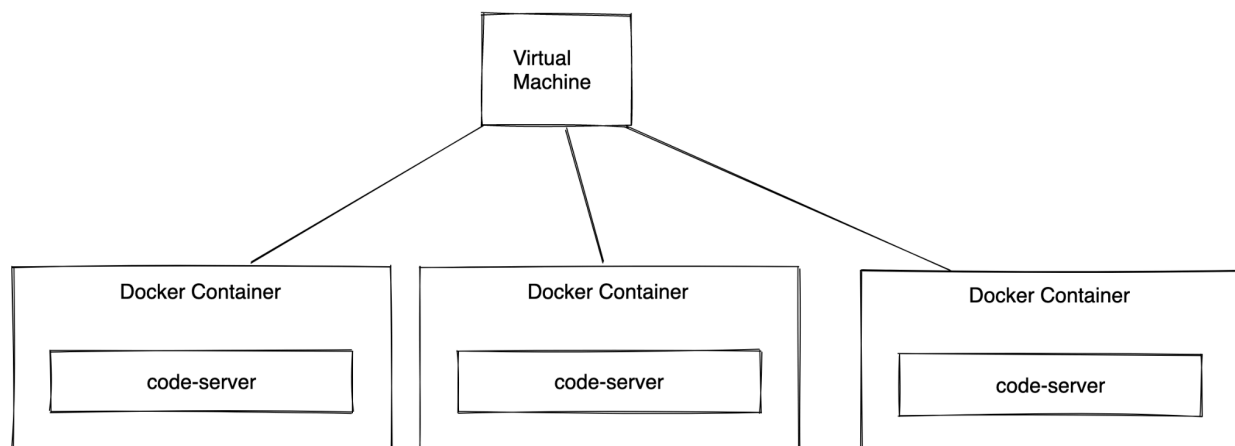
- May use lots of RAM
- Possible file conflicts if admin isn't super careful (i.e: admin should [restrict user to home directory](#))
- Possible limit to number of users

If you have less than five users, this approach may work for your team. More than that, you're likely to run into problems.

## Running Multiple Docker Containers with code-server

A common approach we've seen is to have one machine run multiple Docker containers each with their own code-server instance. We use this approach for [dev.coder.com](https://dev.coder.com) but use coder/coder to orchestrate it. This is similar to the previous approach except the Docker containers run in isolation. That means no file conflicts. Here's a diagram with more details:

### One Machine, Different Containers



#### Pros

- Only requires one machine
- Configuration is simple

#### Cons

- May use lots of RAM
- Possible limit to number of users

Similar to the previous approach, it can work for small teams but you may run into limitations and need more machines.

## Running [minikube](https://minikube.sigs.k8s.io) with a code-server Instance

One of the less popular approaches but still worth knowing is using minikube to set up a local Kubernetes cluster on your machine. Each container then has its own code-server instance.

This is similar to the previous approach with the exception being that it would be easier to migrate to a production Kubernetes environment (i.e. Google Kubernetes Engine) if needed. Here's a diagram with more details:

[insert diagram]

#### **Pros**

- Only requires one machine
- Configuration is simple

#### **Cons**

- May use lots of RAM
- Possible limit to number of users

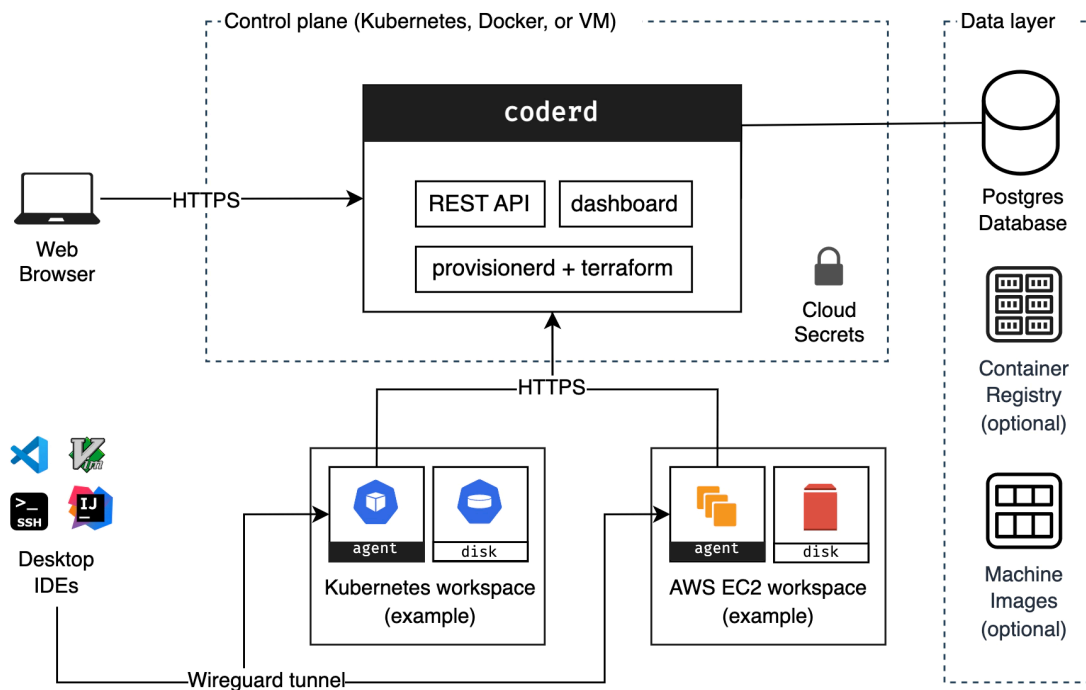
Similar to the previous approach, it can work for small teams but you may run into limitations and need more machines.

#### Running [coder/coder](#)

This is our solution to this problem: a tool that provisions remote development environments via Terraform. With this solution, you can also support multiple IDEs beyond code-server including:

- JetBrains
- VS Code Remote SSH
- JupyterLab
- Emacs

Coder is a flexible solution and could be used to do all the above solutions so it's not exactly an alternative. But it does have more features out-of-the-box and is much easier to set up. Taken from our [docs](#), here's a diagram with more details:



## Pros

- Only requires one machine
- Install a single binary
- Extensible via Terraform
- Includes authentication, user management and audit logging
- Supports other IDEs beyond code-server

## Cons

- Requires knowledge of Terraform

There's definitely some bias here but this is the most proven solution for providing code-server to large teams in a way that scales. It's dogfooded, battle-tested and used by large enterprise companies whose names you probably know. Even if you're not sure, it's worth exploring this route.

## Resources

For more in-depth info on these approaches, take a look at the following resources:

- code-server Discussions
  - [Can I share a code-server instance for multiple, isolated users?](#)
  - [Can you have multiple users](#)

- [Multiple instances on same server?](#)
  - [Can I share a code-server instance for multiple, isolated users?](#)
- [coder/coder docs](#)

## Summary

As you can see, there are various options for running code-server with multiple users. We covered the pros and cons of each solution. It's up to you and your team to decide which will work best for your needs.

If you're interested in setting up [coder/coder](#) for your team and need help, feel free to email me at [joe@coder.com](mailto:joe@coder.com) or join our [Discord](#).

Good luck to you and your team! Happy coding!