

# Quickstart for GitHub Codespaces

## In this article

Introduction

Creating your codespace

Running the application

Edit the application and view changes

Committing and pushing your changes

Personalizing with an extension

Next steps

Further reading

Try out GitHub Codespaces in 5 minutes.

## Introduction

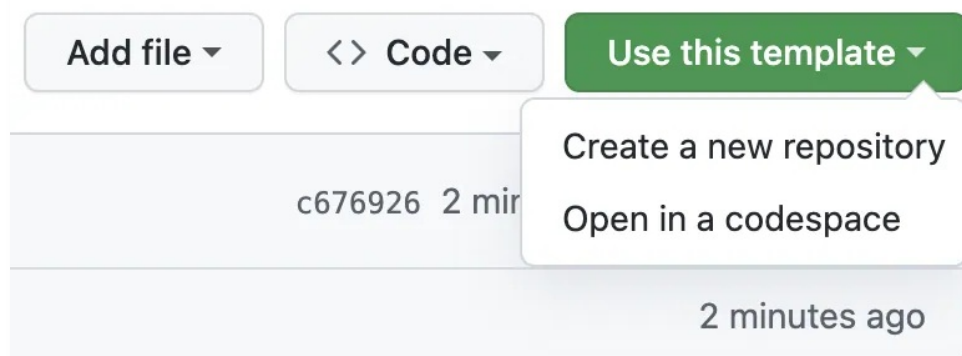
In this guide, you'll create a codespace from a template repository and explore some of the essential features available to you within the codespace. You'll work in the browser version of Visual Studio Code, which is initially the default editor for GitHub Codespaces. After trying out this quickstart you can use Codespaces in other editors, and you can change the default editor. Links are provided at the end of this guide.

From this quickstart, you'll learn how to create a codespace, connect to a forwarded port to view your running application, publish your codespace to a new repository, and personalize your setup with extensions.

For more information on exactly how GitHub Codespaces works, see the companion guide "[Deep dive into GitHub Codespaces](#)."

## Creating your codespace

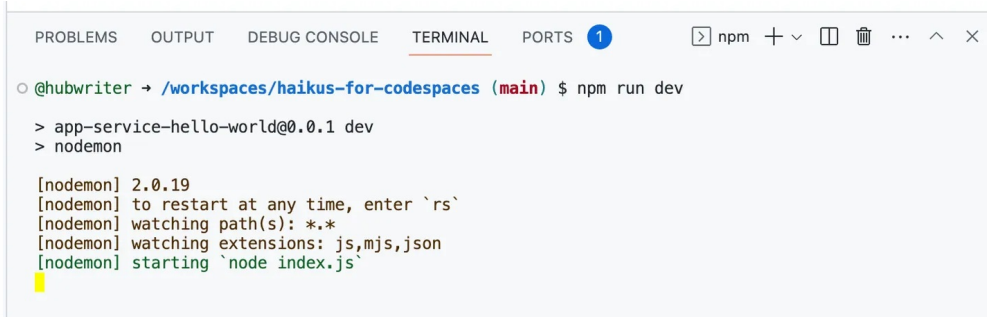
- 1 Navigate to the [github/haikus-for-codespaces](#) template repository.
- 2 Click **Use this template**, then click **Open in a codespace**.



## Running the application

Once your codespace is created, the template repository will be automatically cloned into it. Now you can run the application and launch it in a browser.

- 1 When the terminal becomes available, enter the command `npm run dev`. This example uses a Node.js project, and this command runs the script labeled "dev" in the `package.json` file, which starts up the web application defined in the sample repository.

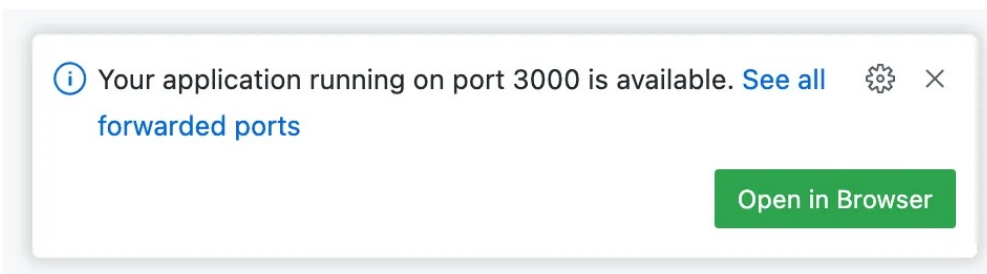


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 npm + - ... ^ x
@hubwriter → /workspaces/haikus-for-codespaces (main) $ npm run dev
> app-service-hello-world@0.0.1 dev
> nodemon

[nodemon] 2.0.19
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
```

If you're following along with a different application type, enter the corresponding start command for that project.


- 2 When your application starts, the codespace recognizes the port the application is running on and displays a pop-up message to let you know that the port has been forwarded.



- 3 Click **Open in Browser** to view your running application in a new tab.

## Edit the application and view changes

- 1 Switch back to your codespace and open the `haikus.json` file by clicking it in the Explorer.
- 2 Edit the `text` field of the first haiku to personalize the application with your own haiku.
- 3 Go back to the running application tab in your browser and refresh to see your changes.

 If you've closed the browser tab, click the Ports tab in VS Code, hover over the **Local Address** value for the running port, and click the **Open in Browser** icon.

PORTS 1			
Port	Local Address	Running Process	Visibility
3000	https://hubwriter-re...	/usr/local/share/nvm/versio...	Private
Add Port			

## Committing and pushing your changes [🔗](#)

Now that you've made a few changes, you can use the integrated terminal or the source view to publish your work to a new repository.

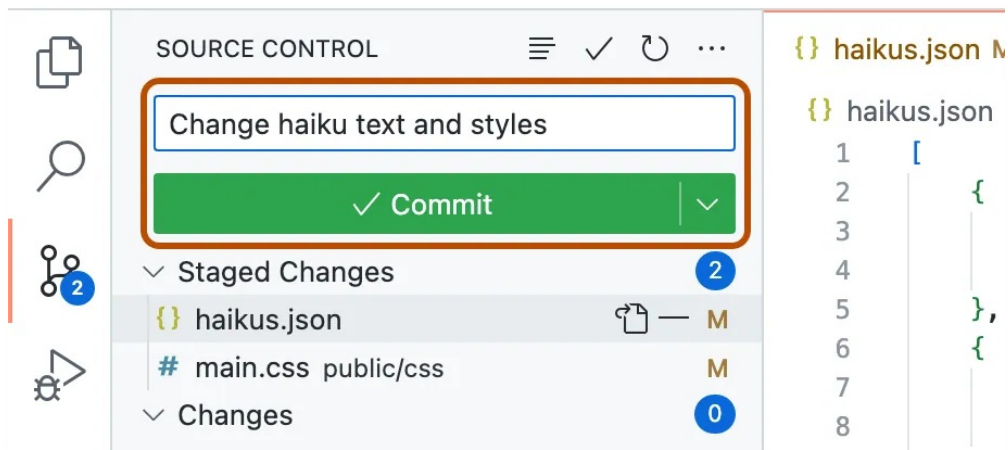
- 1 In the Activity Bar, click the **Source Control** view.

Source Control view showing a commit message input field, a 'Commit' button, and a list of changes. The file `haikus.json` is selected in the 'Changes' list. A blue circle with the number 1 is next to the file name.

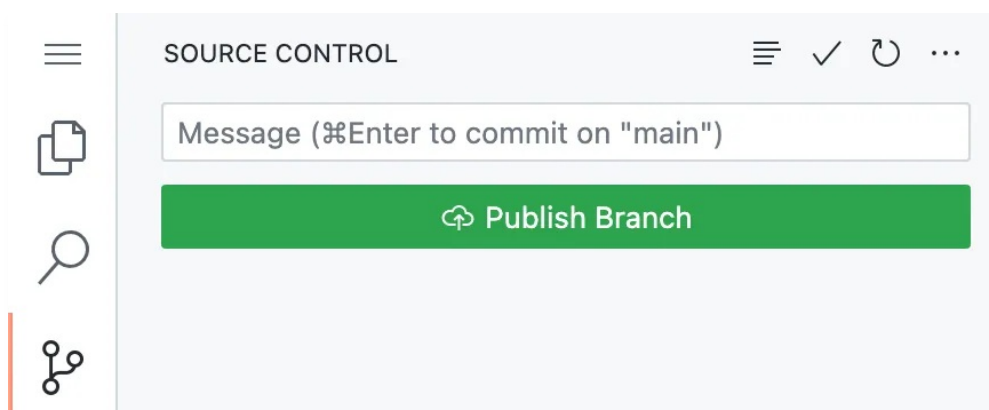
- 2 To stage your changes, click + next to the `haikus.json` file, or next to **Changes** if you've changed multiple files and you want to stage them all.

Source Control view showing a commit message input field, a 'Commit' button, and a list of changes. The file `haikus.json` is selected in the 'Changes' list. A blue circle with the number 2 is next to the '+' icon in the 'Changes' list.

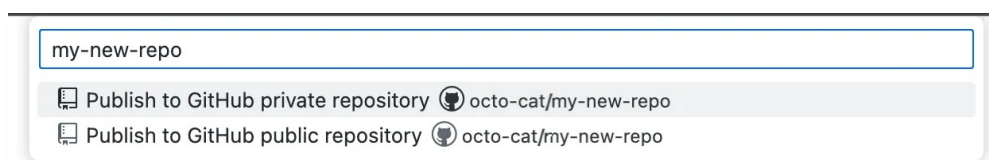
- 3 To commit your staged changes, type a commit message describing the change you've made, then click **Commit**.



- 4 Click **Publish Branch**.

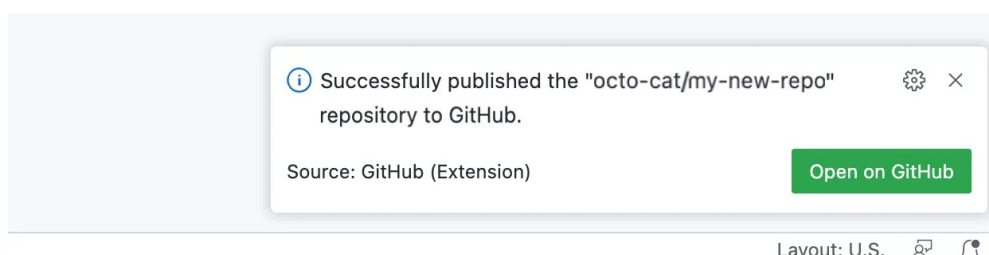


- 5 In the "Repository Name" dropdown, type a name for your new repository, then select **Publish to GitHub private repository** or **Publish to GitHub public repository**.



The owner of the new repository will be the GitHub account with which you created the codespace.

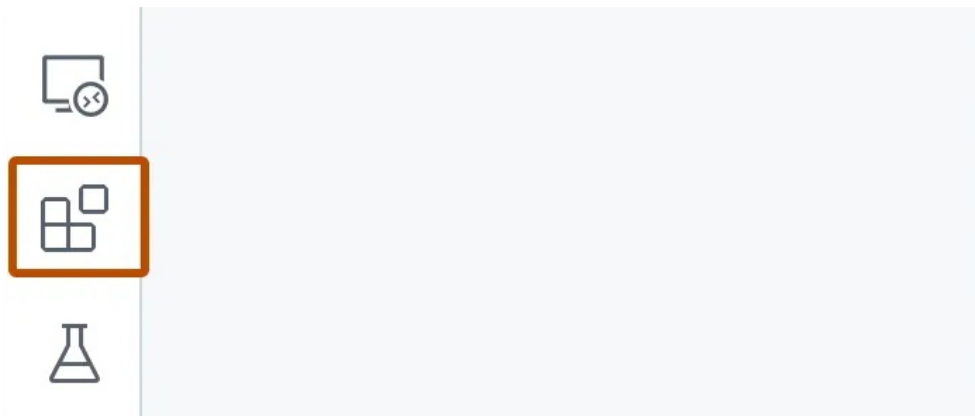
- 6 In the pop-up that appears in the lower right corner of the editor, click **Open on GitHub** to view the new repository on GitHub.com. In the new repository, view the `haikus.json` file and check that the change you made in your codespace has been successfully pushed to the repository.



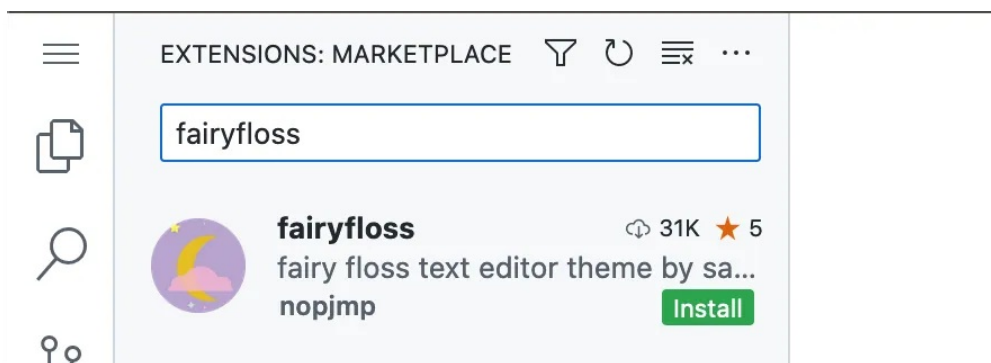
## Personalizing with an extension [↗](#)

When you connect to a codespace using the browser, or the Visual Studio Code desktop application, you can access the Visual Studio Code Marketplace directly from the editor. For this example, you'll install a VS Code extension that alters the theme, but you can install any extension that's useful for your workflow.

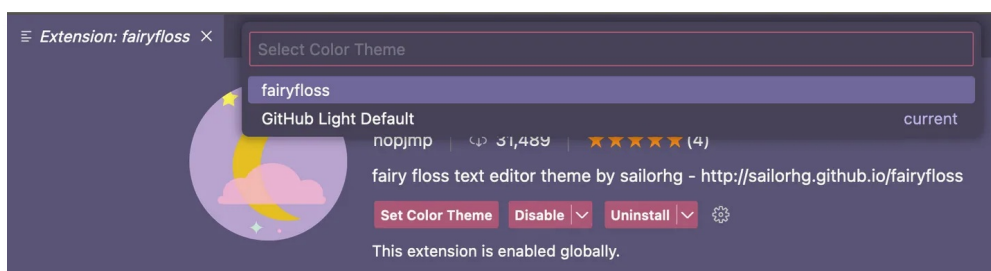
- 1 In the Activity Bar, click the Extensions icon.



- 2 In the search bar, type `fairyfloss` and click **Install**.



- 3 Select the `fairyfloss` theme by selecting it from the list.



## About Settings Sync [↗](#)

You can enable Settings Sync to sync extensions and other settings across devices and instances of VS Code. Your synced settings are cached in the cloud. If Settings Sync is turned on in a codespace, any updates you make to your settings in the codespace are pushed to the cloud, and any updates you push to the cloud from elsewhere are pulled into your codespace. For more information, see "[Personalizing GitHub Codespaces for your account](#)."

## Next steps

---

You've successfully created, personalized, and run your first application within a codespace but there's so much more to explore! Here are some helpful resources for taking your next steps with GitHub Codespaces.

- "[Deep dive into GitHub Codespaces](#)": This quickstart presented some of the features of GitHub Codespaces. The deep dive looks at these areas from a technical standpoint.
- "[Adding a dev container configuration to your repository](#)": These guides provide information on setting up your repository to use GitHub Codespaces with specific languages.
- "[Introduction to dev containers](#)": This guide provides details on creating a custom configuration for Codespaces for your project.

## Further reading

---

- "[Enabling or disabling GitHub Codespaces for your organization](#)"
- "[Using GitHub Codespaces in Visual Studio Code](#)"
- "[Using GitHub Codespaces in your JetBrains IDE](#)"
- "[Using GitHub Codespaces with GitHub CLI](#)"
- "[Setting your default editor for GitHub Codespaces](#)."
- "[Managing the cost of GitHub Codespaces in your organization](#)"

### Legal

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Status](#) [Pricing](#) [Expert services](#) [Blog](#)