Community

TUTORIAL

How To Set Up the code-server Cloud IDE Platform on Ubuntu 20.04

Development Ubuntu 20.04

By Savic

Published on May 20, 2020 16.4k

English

Not using Ubuntu 20.04?
Choose a different version or distribution.

Ubuntu 20.04

The author selected the Free and Open Source Fund to receive a donation as part of the Write for DOnations program.

Introduction

With developer tools moving to the cloud, creation and adoption of cloud IDE (Integrated Development Environment) platforms is growing. Cloud IDEs allow for real-time collaboration between developer teams to work in a unified development environment that minimizes incompatibilities and enhances productivity. Accessible through web browsers, cloud IDEs are available from every type of modern device.

code-server is Microsoft Visual Studio Code running on a remote server and accessible directly from your browser. Visual Studio Code is a modern code editor with integrated Git support, a code debugger, smart autocompletion, and customizable and extensible features. This means that you can use various devices running different operating systems, and always have a consistent development environment on hand.

SCROLL TO TOP

In this tutorial, you will set up the code-server cloud IDE platform on your Ubuntu 20.04 machine and expose it at your domain, secured with free Let's Encrypt TLS certificates. In the end, you'll have Microsoft Visual Studio Code running on your Ubuntu 20.04 server, available at your domain and protected with a password.

Prerequisites

- A server running Ubuntu 20.04 with at least 2GB RAM, root access, and a sudo, non-root account. You can set this up by following this initial server setup guide.
- Nginx installed on your server. For a guide on how to do this, complete Steps 1 to 4 of How To Install Nginx on Ubuntu 20.04.
- A fully registered domain name to host code-server, pointed to your server. This tutorial will use `code-server.your-domain` throughout. You can purchase a domain name on Namecheap, get one for free on Freenom, or use the domain registrar of your choice. For DigitalOcean, you can follow this introduction to DigitalOcean DNS for details on how to add them.

Step 1 — Installing code-server

In this section, you will set up code-server on your server. This entails downloading the latest version and creating a `systemd` service that will keep code-server always running in the background. You'll also specify a restart policy for the service, so that code-server stays available after possible crashes or reboots.

You'll store all data pertaining to code-server in a folder named `~/code-server`. Create it by running the following command:

```
$ mkdir ~/code-server
```

Navigate to it:

```
$ cd ~/code-server
```

You'll need to head over to the Github releases page of code-server and pick the latest Linux build (the file will contain 'linux' in its name). At the time of writing, th SCROLL TO TOP was 3.3.1. Download it using `wget` by running the following command:

```
$ wget https://github.com/cdr/code-server/releases/download/v3.3.1/code-server-3.3.1-linu
```

Then, unpack the archive by running:

```
$ tar -xzvf code-server-3.3.1-linux-amd64.tar.gz
```

You'll get a folder named exactly as the original file you downloaded, which contains the code-server source code. Copy it to `/usr/lib/code-server` so you'll be able to access it system wide by running the following command:

```
$ sudo cp -r code-server-3.3.1-linux-amd64 /usr/lib/code-server
```

Then, create a symbolic link at `/usr/bin/code-server`, pointing to the code-server executable:

```
$ sudo ln -s /usr/lib/code-server/bin/code-server /usr/bin/code-server
```

Next, create a folder for code-server, where it will store user data:

```
$ sudo mkdir /var/lib/code-server
```

Now that you've downloaded code-server and made it available system-wide, you will create a systemd service to keep code-server running in the background at all times.

You'll store the service configuration in a file named `code-server.service`, in the `/lib/systemd/system` directory, where systemd stores its services. Create it using your text editor:

```
$ sudo nano /lib/systemd/system/code-server.service
```

Add the following lines:

SCROLL TO TOP

/lib/systemd/system/code-server.service

```
[Unit]
Description=code-server
After=nginx.service

[Service]
Type=simple
Environment=PASSWORD=your_password
ExecStart=/usr/bin/code-server --bind-addr 127.0.0.1:8080 --user-data-dir /var/lib/code-serv
Restart=always

[Install]
WantedBy=multi-user.target
```

Here you first specify the description of the service. Then, you state that the `nginx` service must be started before this one. After the `[Unit]` section, you define the type of the service (`simple` means that the process should be simply run) and provide the command that will be executed.

You also specify that the global code-server executable should be started with a few arguments specific to code-server. `--bind-addr 127.0.0.1:8080` binds it to `localhost` at port `8080`, so it's only directly accessible from inside of your server. `--user-data-dir /var/lib/code-server` sets its user data directory, and `--auth password` specifies that it should authenticate visitors with a password, specified in the `PASSWORD` environment variable declared on the line above it.

Remember to replace `your_password` with your desired password, then save and close the file.

The next line tells systemd to restart code-server in all malfunction events (for example, when it crashes or the process is killed). The `[Install]` section orders systemd to start this service when it becomes possible to log in to your server.

Start the code-server service by running the following command:

```
$ sudo systemctl start code-server
```

Check that it's started correctly by observing its status:

SCROLL TO TOP

```
$ sudo systemctl status code-server
```

You'll see output similar to:

```
Output
● code-server.service - code-server
     Loaded: loaded (/lib/systemd/system/code-server.service; disabled; vendor preset: enablec
     Active: active (running) since Wed 2020-05-20 13:03:40 UTC; 12s ago
   Main PID: 14985 (node)
      Tasks: 18 (limit: 2345)
     Memory: 26.1M
     CGroup: /system.slice/code-server.service
             ├─14985 /usr/lib/code-server/bin/../lib/node /usr/lib/code-server/bin/.. --bind-a
             └─15010 /usr/lib/code-server/lib/node /usr/lib/code-server --bind-addr 127.0.0.1:

May 20 13:03:40 code-server-update-2004 systemd[1]: Started code-server.

May 20 13:03:40 code-server-update-2004 code-server[15010]: info  Wrote default config file tc
May 20 13:03:40 code-server-update-2004 code-server[15010]: info  Using config file ~/.config/
May 20 13:03:40 code-server-update-2004 code-server[15010]: info  Using user-data-dir /var/lit
May 20 13:03:40 code-server-update-2004 code-server[15010]: info  code-server 3.3.1 6f1309795e
May 20 13:03:40 code-server-update-2004 code-server[15010]: info  HTTP server listening on htt
May 20 13:03:40 code-server-update-2004 code-server[15010]: info      - Using password from $F
May 20 13:03:40 code-server-update-2004 code-server[15010]: info      - To disable use `--auth
May 20 13:03:40 code-server-update-2004 code-server[15010]: info    - Not serving HTTPS
```

To make code-server start automatically after a server reboot, enable its service by running the following command:

```
$ sudo systemctl enable code-server
```

In this step, you've downloaded code-server and made it available globally. Then, you've created a systemd service for it and enabled it, so code-server will start at every server boot. Next, you'll expose it at your domain by configuring Nginx to serve as a reverse proxy between the visitor and code-server.

Step 2 — Exposing code-server at Your Domain

SCROLL TO TOP

In this section, you will configure Nginx as a reverse proxy for code-server.

As you have learned in the Nginx prerequisite step, its site configuration files are stored under `/etc/nginx/sites-available` and must later be symlinked to `/etc/nginx/sites-enabled` to become active.

You'll store the configuration for exposing code-server at your domain in a file named `code-server.conf`, under `/etc/nginx/sites-available`. Start off by creating it using your editor:

```
$ sudo nano /etc/nginx/sites-available/code-server.conf
```

Add the following lines:

/etc/nginx/sites-available/code-server.conf

```
server {
    listen 80;
    listen [::]:80;

    server_name code-server.your-domain;

    location / {
      proxy_pass http://localhost:8080/;
      proxy_set_header Upgrade $http_upgrade;
      proxy_set_header Connection upgrade;
      proxy_set_header Accept-Encoding gzip;
    }
}
```

Replace `code-server.your-domain` with your desired domain, then save and close the file.

In this file, you define that Nginx should listen to HTTP port `80`. Then, you specify a `server_name` that tells Nginx for which domain to accept requests and apply this particular configuration. In the next block, for the root location (`/`), you specify that requests should be passed back and forth to code-server running at `localhost:8080`. The next three lines (starting with `proxy_set_header`) order Nginx to carry over some HTTP request headers that are needed for correct functioning of WebSockets, which code-server extensively uses.

To make this site configuration active, you will need to create a symlink of it in the `/etc/nginx/sites-enabled` folder by running:

SCROLL TO TOP

```
$ sudo ln -s /etc/nginx/sites-available/code-server.conf /etc/nginx/sites-enabled/code-se
```

To test the validity of the configuration, run the following command:

```
$ sudo nginx -t
```

You'll see the following output:

```
Output
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

For the configuration to take effect, you'll need to restart Nginx:

```
$ sudo systemctl restart nginx
```

You now have your code-server installation accessible at your domain. In the next step, you'll secure it by applying a free Let's Encrypt TLS certificate.

Step 3 — Securing Your Domain

In this section, you will secure your domain using a Let's Encrypt TLS certificate, which you'll provision using Certbot.

To install the latest version of Certbot and its Nginx plugin, run the following command:

```
$ sudo apt install certbot python3-certbot-nginx
```

As part of the prerequisites, you have enabled `ufw` (Uncomplicated Firewall) and configured it to allow unencrypted HTTP traffic. To be able to access the secured site, you'll need to configure it to accept encrypted traffic by running the following command:

```
$ sudo ufw allow https
```

SCROLL TO TOP

The output will be:

```
Output
Rule added
Rule added (v6)
```

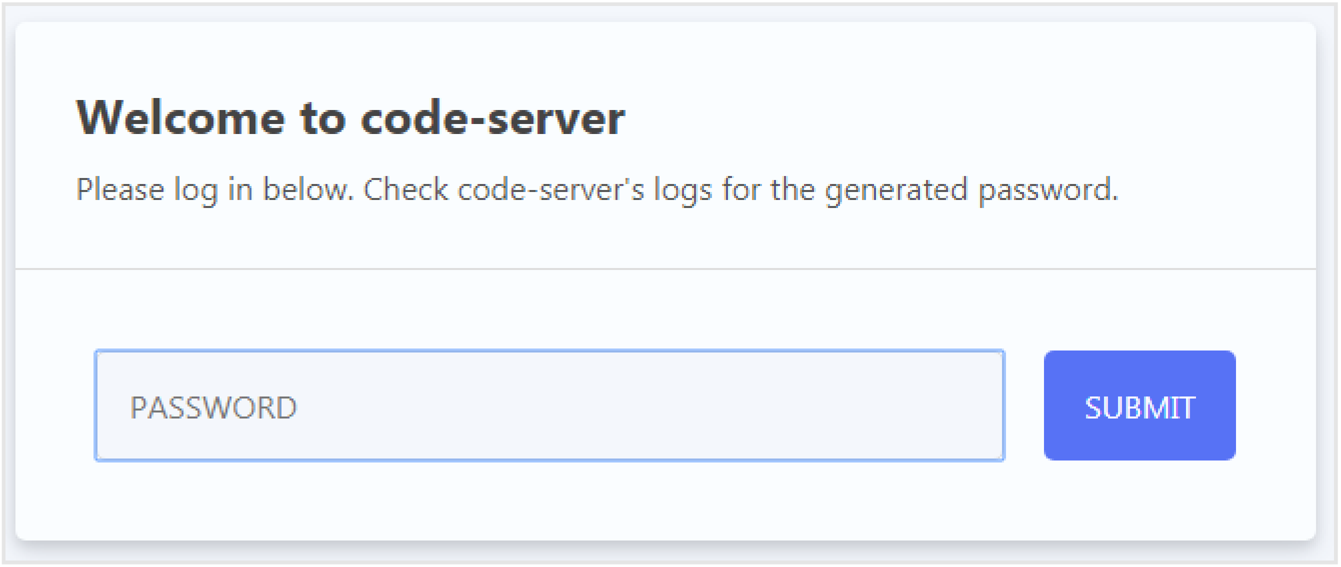
Similarly to Nginx, you'll need to reload it for the configuration to take effect:

```
$ sudo ufw reload
```

The output will show:

```
Output
Firewall reloaded
```

Then, in your browser, navigate to the domain you used for code-server. You will see the code-server login prompt.

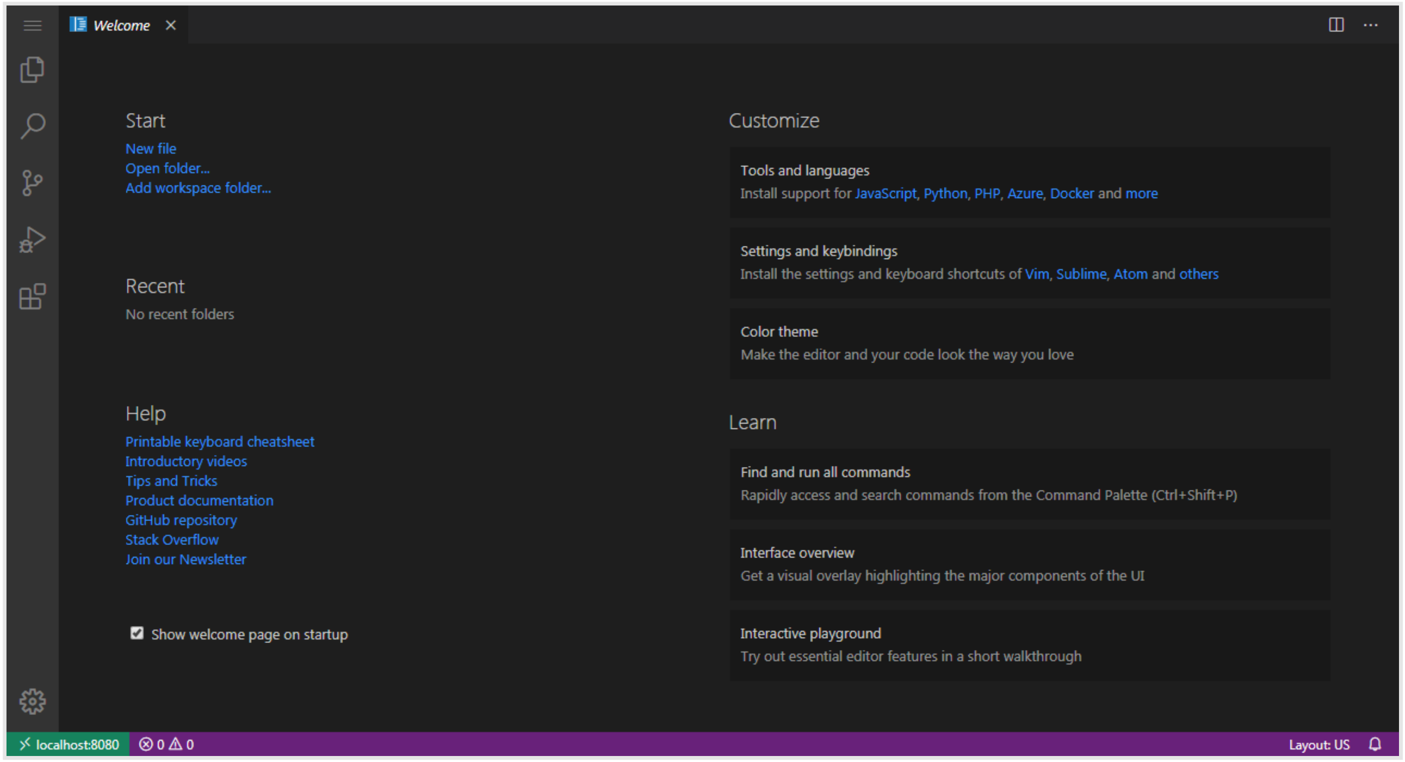


Welcome to code-server

Please log in below. Check code-server's logs for the generated password.

code-server is asking you for your password. Enter the one you set in the previous step and press **Enter IDE**. You'll now enter code-server and immediately see its editor GUI.

SCROLL TO TOP


Welcome
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Join our Newsletter
Show welcome page on startup
Customize
Tools and languages
Install support for JavaScript, Python, PHP, Azure, Docker and more
Settings and keybindings
Install the settings and keyboard shortcuts of Vim, Sublime, Atom and others
Color theme
Make the editor and your code look the way you love
Learn
Find and run all commands
Rapidly access and search commands from the Command Palette (Ctrl+Shift+P)
Interface overview
Get a visual overlay highlighting the major components of the UI
Interactive playground
Try out essential editor features in a short walkthrough
localhost:8080
0 0
Layout: US

Now that you've checked that code-server is correctly exposed at your domain, you'll install Let's Encrypt TLS certificates to secure it, using Certbot.

To request certificates for your domain, run the following command:

```
$ sudo certbot --nginx -d code-server.your-domain
```

In this command, you run `certbot` to request certificates for your domain—you pass the domain name with the `-d` parameter. The `--nginx` flag tells it to automatically change Nginx site configuration to support HTTPS. Remember to replace `code-server.your-domain` with your domain name.

If this is your first time running Certbot, you'll be asked to provide an email address for urgent notices and to accept the EFF's Terms of Service. Certbot will then request certificates for your domain from Let's Encrypt. It will then ask you if you'd like to redirect all HTTP traffic to HTTPS:

```
Output
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
```

SCROLL TO TOP

```
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel):
```

It is recommended to select the second option in order to maximize security. After you input your selection, press `ENTER`.

The output will be similar to this:

```
Output
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
 /etc/letsencrypt/live/code-server.your-domain/fullchain.pem
 Your key file has been saved at:
 /etc/letsencrypt/live/code-server.your-domain/privkey.pem
 Your cert will expire on ... To obtain a new or tweaked
 version of this certificate in the future, simply run certbot again
 with the "certonly" option. To non-interactively renew *all* of
 your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
 configuration directory at /etc/letsencrypt. You should make a
 secure backup of this folder now. This configuration directory will
 also contain certificates and private keys obtained by Certbot so
 making regular backups of this folder is ideal.
- If you like Certbot, please consider supporting our work by:

 Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
 Donating to EFF:                    https://eff.org/donate-le
```

This means that Certbot has successfully generated TLS certificates and applied them to the Nginx configuration for your domain. You can now reload your code-server domain in your browser and observe a padlock to the left of the site address, which means that your connection is properly secured.

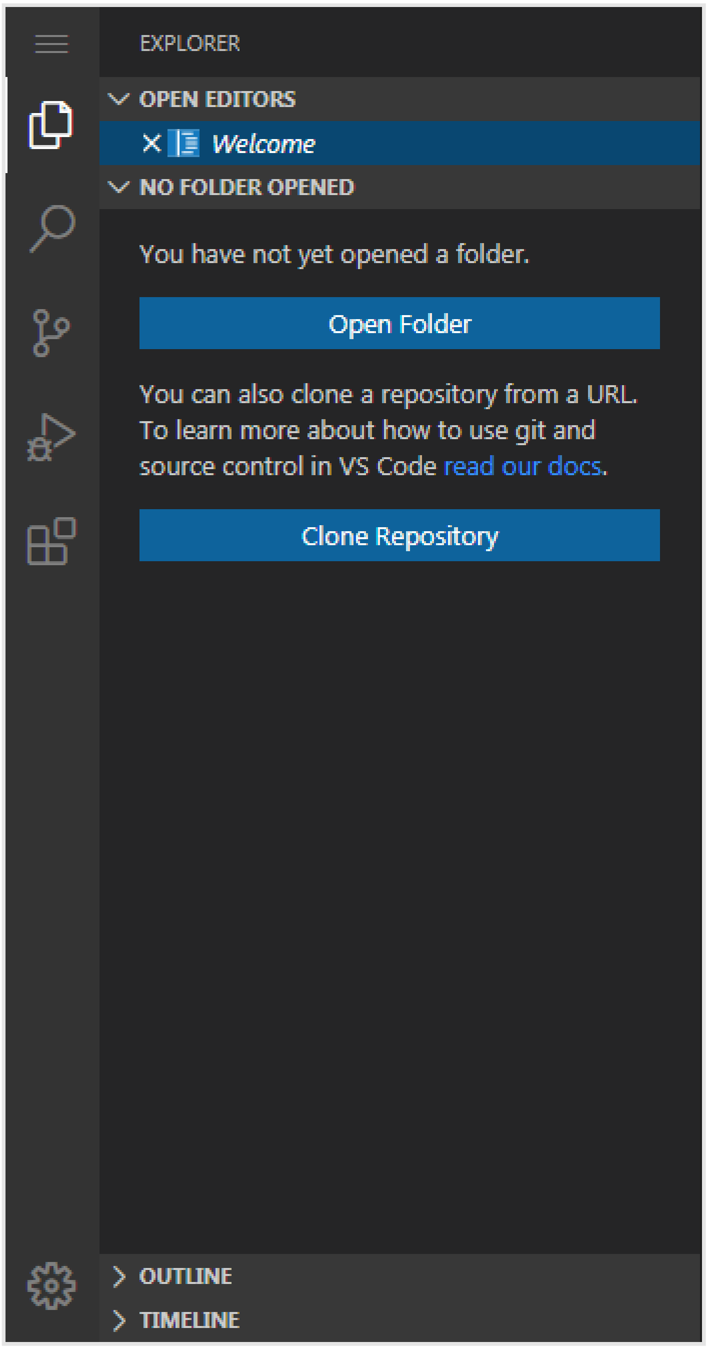
Now that you have code-server accessible at your domain through a secured Nginx reverse proxy, you're ready to review the user interface of code-server.

SCROLL TO TOP

Step 4 — Using the code-server Interface

In this section, you'll use some of the features of the code-server interface. Since code-server is Visual Studio Code running in the cloud, it has the same interface as the standalone desktop edition.

On the left-hand side of the IDE, there is a vertical row of six buttons opening the most commonly used features in a side panel known as the Activity Bar.

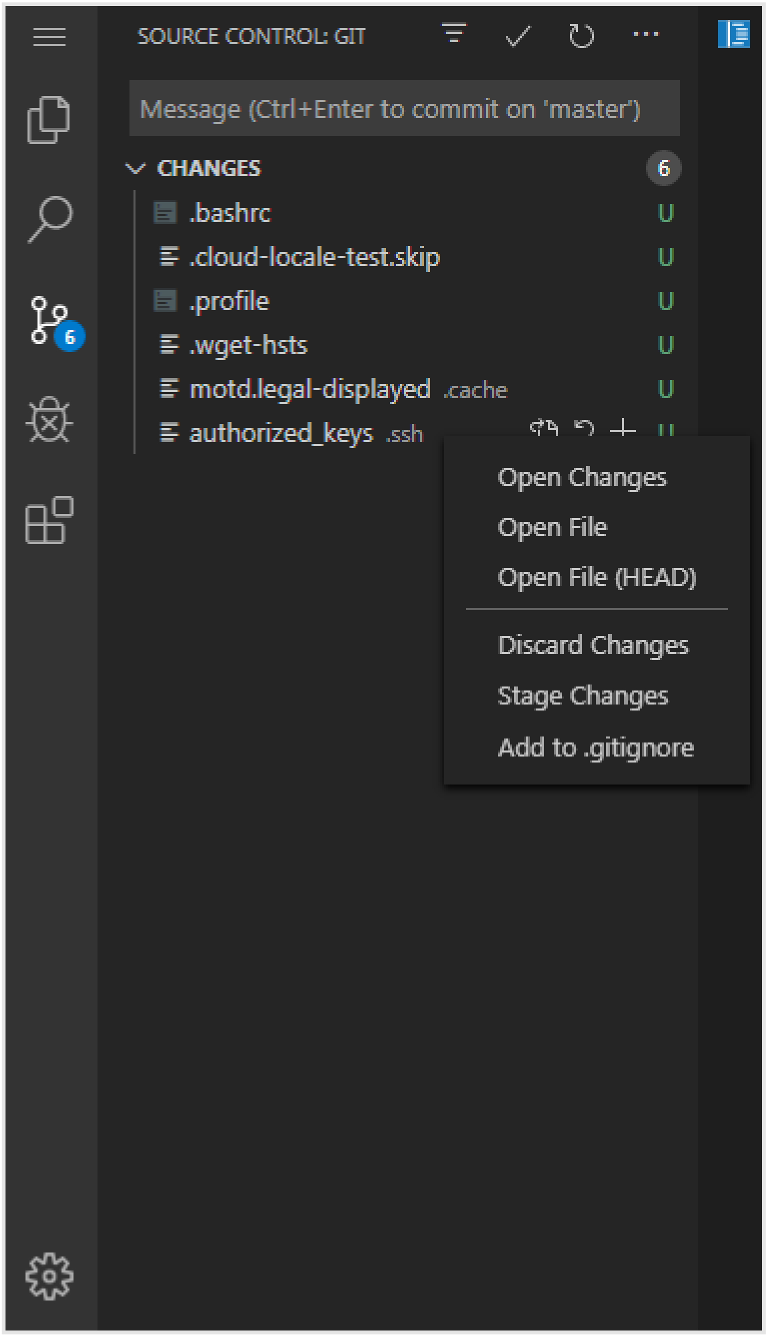


You have not yet opened a folder.

You can also clone a repository from a URL. To learn more about how to use git and source control in VS Code read our docs.

This bar is customizable so you can move these views to a different order or remove them from the bar. By default, the first button opens the general menu in a dropdown, while the second view opens the Explorer panel that provides tree-like navigation of SCROLL TO TOP structure. You can manage your folders and files here—creating, deleting, moving, and renaming them as necessary. The next view provides access to a search and replace functionality.

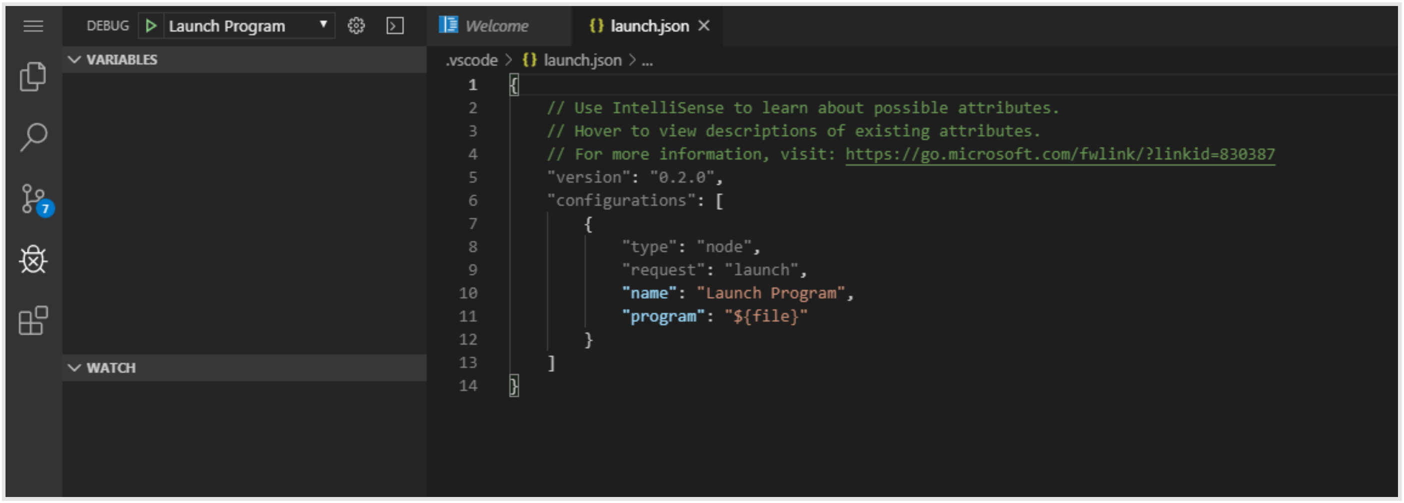
Following this, in the default order, is your view of the source control systems, like Git. Visual Studio code also supports other source control providers and you can find further instructions for source control work flows with the editor in this documentation.



The debugger option on the Activity Bar provides all the common actions for debugging in the panel. Visual Studio Code comes with built-in support for the Node.js runtime debugger and any language that transpiles to Javascript. For other languages you ca

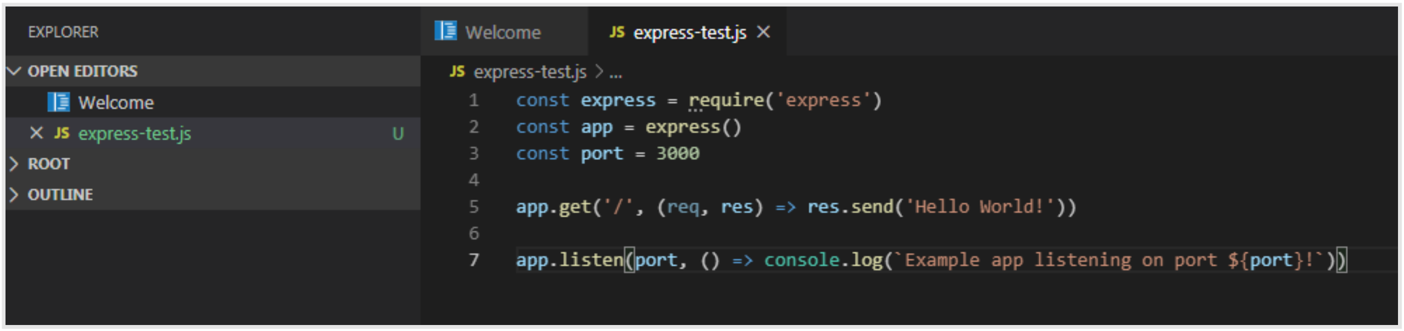
SCROLL TO TOP

extensions for the required debugger. You can save debugging configurations in the `launch.json` file.



```
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "type": "node",
            "request": "launch",
            "name": "Launch Program",
            "program": "${file}"
        }
    ]
}
```

The final view in the Activity Bar provides a menu to access available extensions on the Marketplace.



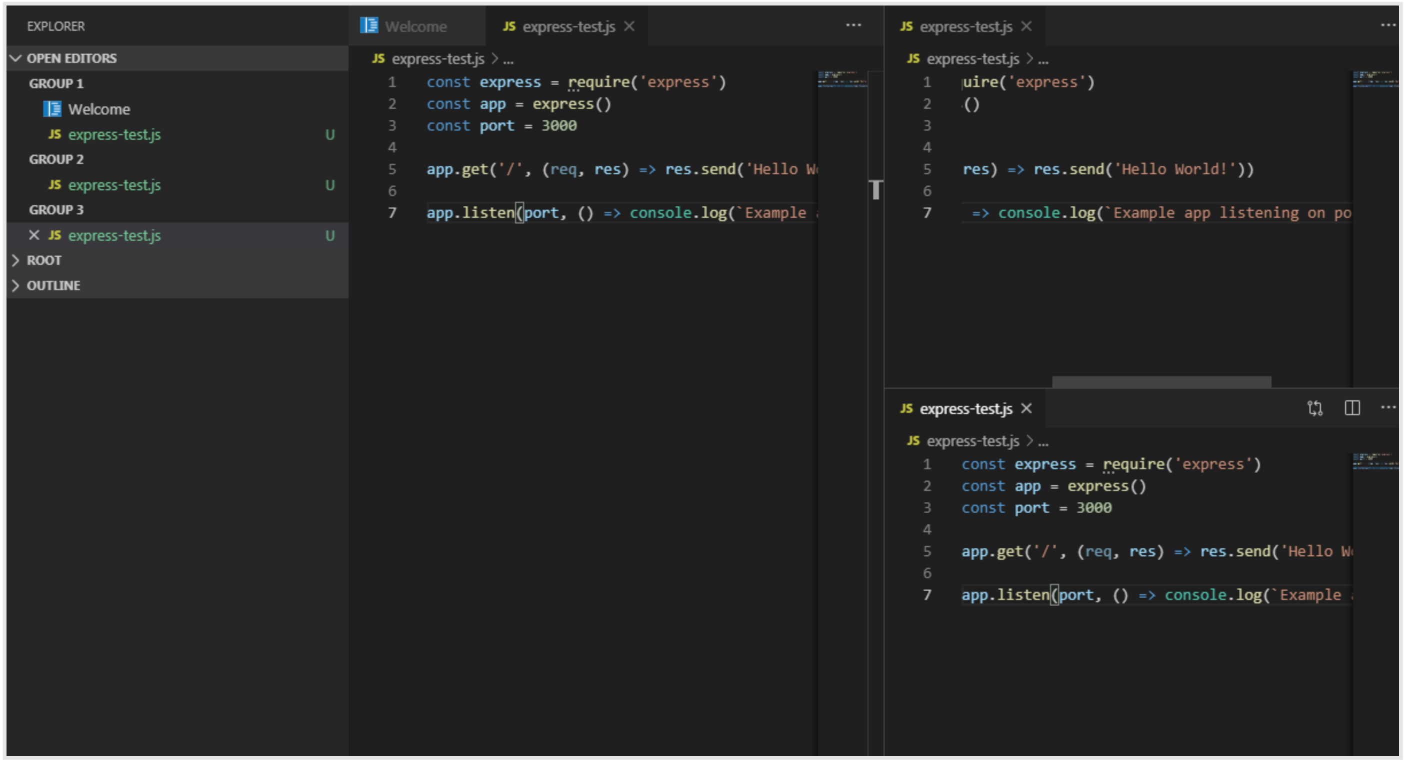
```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(port, () => console.log(`Example app listening on port ${port}!`))
```

The central part of the GUI is your editor, which you can separate by tabs for your code editing. You can change your editing view to a grid system or to side-by-side files.

SCROLL TO TOP



```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello W

app.listen(port, () => console.log(`Example
```

```
uire('express')
()


res) => res.send('Hello World!'))

 => console.log(`Example app listening on po
```

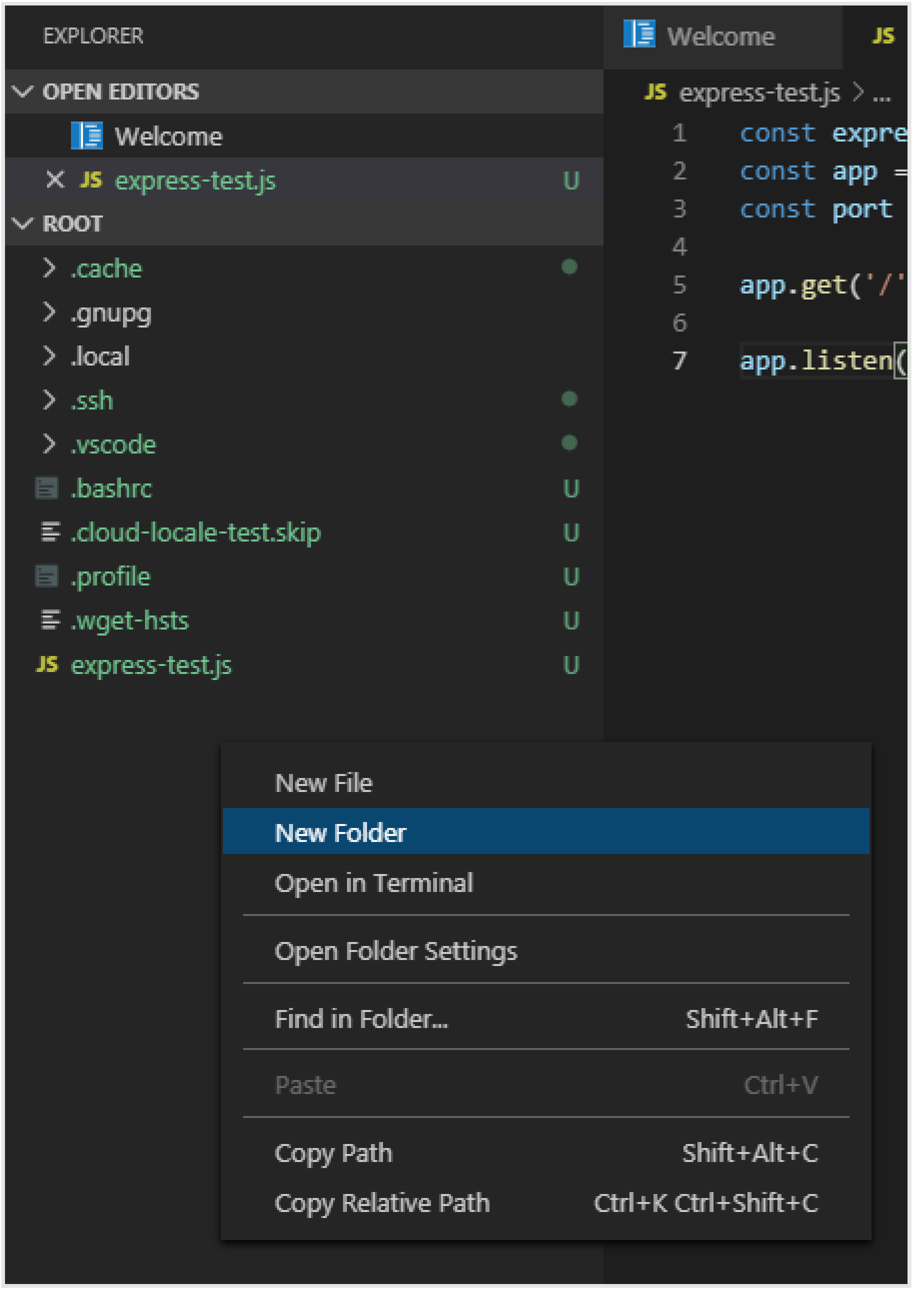
```
const express = require('express')
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello W

app.listen(port, () => console.log(`Example
```

After creating a new file through the **File** menu, an empty file will open in a new tab, and once saved, the file's name will be viewable in the Explorer side panel. Creating folders can be done by right clicking on the Explorer sidebar and clicking on **New Folder**. You can expand a folder by clicking on its name as well as dragging and dropping files and folders to upper parts of the hierarchy to move them to a new location.

SCROLL TO TOP



```
const expre
const app =
const port

app.get('/'

app.listen(
```

You can gain access to a terminal by entering `CTRL+SHIFT+``, or by clicking on **Terminal** in the upper menu dropdown, and selecting **New Terminal**. The terminal will open in a lower panel and its working directory will be set to the project's workspace, which contains the files and folders shown in the Explorer side panel.

You've explored a high-level overview of the code-server interface and reviewed some of the most commonly used features.

Conclusion

You now have code-server, a versatile cloud IDE, installed on your Ubuntu 20.04 server exposed at your domain and secured using Let's Encrypt certificates. You c

SCROLL TO TOP

projects individually, as well as in a team-collaboration setting. Running a cloud IDE frees resources on your local machine and allows you to scale the resources when needed. For further information, see the Visual Studio Code documentation for additional features and detailed instructions on other components of code-server.

If you would like to run code-server on your DigitalOcean Kubernetes cluster check out our tutorial on How To Set Up the code-server Cloud IDE Platform on DigitalOcean Kubernetes.

About the authors

Savic

has authored 43 tutorials.

Kathryn Hancox

Editor

Still looking for an answer?

How To Build a Responsive About Me Page with Laravel, Sail, and Tailwind CSS

Tutorial

How To Use Enums in TypeScript

Tutorial

Comments

8 Comments

Leave a comment...

Sign In to Comment

0 palazuelosgg May 31, 2020

Hello. All was good until

Step 2 — Exposing code-server at Your Domain

It just does not work on my side.

SCROLL TO TOP

What did I setup on Namecheap.com (my registrar)?
I created one “A Record”, for host @ pointing to my DO VPS IP address. So far my “index.html” is displayed as expected.

Then I did the setup suggested in Step #2, for nginx and the code-server.mydomain.com.

- I created new nginx code-server.conf file, while indicating **code-server.mydomain.com;**
- I tested and restarted nginx
- I tried http://code-server.mydomain.com ← in Google Chrome and Firefox errors.

Chrome:
This site can’t be reached
code-server.mydomain.com’s server IP address could not be found.
Try running Windows Network Diagnostics.
DNSPROBEFINISHED_NXDOMAIN

Firefox:
*Hmm. We’re having trouble finding that site.

We can’t connect to the server at code-server.palazuelos.xyz.

If that address is correct, here are three other things you can try:

```
Try again later.
Check your network connection.
If you are connected but behind a firewall, check that Firefox has permission to acces
```

Any idea or suggestion?
thanks,

Reply Report

0 jihad April 16, 2021
I wasted 3 hours trying fixing it — when i decided to do the last step anyways (Securing your domain), it worked :) not sure if it was that or due to DNS taking couple of hours to refresh.

Reply Report

SCROLL TO TOP

0 sdg1972 June 21, 2020

Fantastic documentation! Every step went exactly as you said it would and everything worked flawlessly. Thanks for the terrific work!

Reply Report

0 myerspa August 24, 2020

I got this all up and running, thanks for the instructions! However, whenever I run the terminal within code-server, I get 'Command not found' errors. It doesn't seem to be able to find anything like nvm, npm, or yarn that all work when using the terminal directly on the box.

Any idea where I went wrong there or how to fix?

Reply Report

0 aykutapps November 14, 2020

Thank you for the documentation. I have followed the steps and getting the error "**WebSocket close with status code 1006**" when I open the code server in the browser. Do you know what can be the issue?

Kind regards,
Aykut

Reply Report

0 yiannismarios75 February 3, 2021

Hi,

What you are doing:

sudo systemctl enable code-server

is wrong because it will start code-server as root which is dangerous! You can see this if you open the integrated terminal where it says "root@".

Instead you should enable the service on start-up like this:

sudo systemctl enable –now code-server@$USER

then:

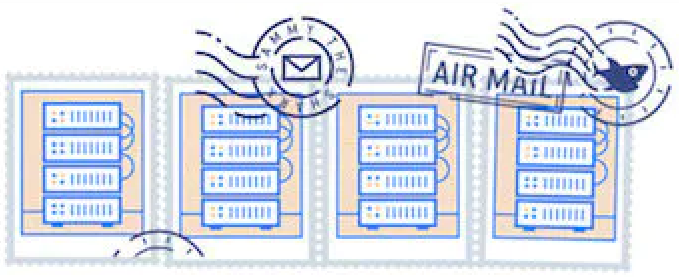
sudo service code-server start

Reply Report

gouravkr May 15, 2021

SCROLL TO TOP

0 This comment has been marked as resolved by **gouravkr**. SHOW COMMENT



GET OUR BIWEEKLY NEWSLETTER

Sign up for Infrastructure as a Newsletter.

HOLLIE'S HUB FOR GOOD

Working on improving health and education, reducing inequality, and spurring economic growth?

SCROLL TO TOP

We'd like to help.

BECOME A CONTRIBUTOR

You get paid; we donate to tech nonprofits.

Featured on Community Kubernetes Course Learn Python 3 Machine Learning in Python Getting started with Go Intro to Kubernetes

DigitalOcean Products Virtual Machines Managed Databases Managed Kubernetes Block Storage Object Storage Marketplace VPC Load Balancers

Welcome to the developer cloud

DigitalOcean makes it simple to launch in the cloud and scale up as you grow – whether you're running one virtual machine or ten thousand.

Learn More

SCROLL TO TOP

Company

About
Leadership
Blog
Careers
Partners
Referral Program
Press
Legal
Security & Trust Center

Products

Pricing
Products Overview
Droplets
Kubernetes
Managed Databases
Spaces
Marketplace
Load Balancers
Block Storage
API Documentation
Documentation
Release Notes

Community

Tutorials
Q&A
Tools and Integrations
Tags
Product Ideas
Write for DigitalOcean
Presentation Grants
Hatch Startup Program
Shop Swag
Research Program
Open Source
Code of Conduct

Contact

Get Support
Trouble Signing In?
Sales
Report Abuse
System Status

SCROLL TO TOP