How to Install Visual Studio Code - Server IDE on Ubuntu 18.04 LTS

Code-server is a Visual Studio (VS) Code that can be run remotely on the server and which is accessible through a web browser. It allows you to create and have a consistent development environment that can be accessed anytime and everywhere.

In this tutorial, we will show you how to install the Code-server with Nginx as a reverse proxy and SSL Letsencrypt on the latest Ubuntu 18.04 Server.

Prerequisites

For this guide, we will install the Code-server on the Ubuntu 18.04 server with 3GB of RAM, 25GB free disk space, and 2CPUs.

What we will do:

- Add User and Download Code-server Binary
- Setup Code-server as a Systemd Service
- Generate SSL Letsencrypt
- Setup Nginx as a Reverse Proxy for Code-server
- Testing

Step 1 - Add User and Download Code-Server Binary

First, we will add a new user and download the code-server binary file from GitHub.

Add a new user 'code' using the command below.

```
useradd -m -s /bin/bash code
passwd code
```

Now log in as 'code' user and download the code-server binary file.

```
su - code
wget https://github.com/cdr/code-server/releases/download/2.1692-vsc1.39.2/code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
```

Extract the code-server and rename the directory as 'bin'.

```
tar -xf code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
mv code-server2.1692-vsc1.39.2-linux-x86_64/ bin/
```

Now make the code-server as an executable binary file.

```
chmod +x ~/bin/code-server
```

And create a new data directory for storing the user data.

```
mkdir -p ~/data
```

Now you've created a new user 'code' and downloaded the code-server binary to the home directory of user 'code'.

```
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ ls
code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ tar -xf code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
code@ubuntu-vscode:~$ mv code-server2.1692-vsc1.39.2-linux-x86_64/ bin/
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ ls
bin  code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ chmod +x ~/bin/code-server
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ mkdir -p ~/data
code@ubuntu-vscode:~$
code@ubuntu-vscode:~$ tree
.
├── bin
│   ├── code-server
│   ├── LICENSE.txt
│   ├── README.md
│   └── ThirdPartyNotices.txt
├── code-server2.1692-vsc1.39.2-linux-x86_64.tar.gz
└── data

2 directories, 5 files
code@ubuntu-vscode:~$
```

Step 2 - Setup Code-Server as a Systemd Service

In this step, we will set up the code-server to run as a system service.

The code-server service will run under the user 'code', with default port '8080' and enabled password authentication.

Go to the '/etc/systemd/system' directory and create a new service file 'code-server.service' using vim editor.

```
cd /etc/systemd/system/
vim code-server.service
```

Change the 'Environment' option for a password with your own and paste the configuration into it.

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

[Service]
User=code
WorkingDirectory=/home/code
Environment=PASSWORD=hakasevscodeserv
ExecStart=/home/code/bin/code-server --host 127.0.0.1 --user-data-dir /home/code/data --auth password
Restart=always

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

Save and close.

Now reload the system manager.

```
systemctl daemon-reload
```

After that, start the code-server service and add it to the system boot using the following command.

```
systemctl start code-server
systemctl enable code-server
```

```
root@ubuntu-vscode:~#
root@ubuntu-vscode:~# cd /etc/systemd/system/
root@ubuntu-vscode:/etc/systemd/system# vim code-server.service
root@ubuntu-vscode:/etc/systemd/system#
root@ubuntu-vscode:/etc/systemd/system# systemctl daemon-reload
root@ubuntu-vscode:/etc/systemd/system#
root@ubuntu-vscode:/etc/systemd/system# systemctl start code-server
root@ubuntu-vscode:/etc/systemd/system# systemctl enable code-server
Created symlink /etc/systemd/system/multi-user.target.wants/code-server.service → /etc/systemd/system/code-server.service.
root@ubuntu-vscode:/etc/systemd/system#
```

Now Check the code-server service.

```
netstat -plntu
systemctl status code-server
```

And the code-server service is up and running on default port '8080'.

```
root@ubuntu-vscode:/etc/systemd/system#
root@ubuntu-vscode:/etc/systemd/system# netstat -plntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN      2965/code-server
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      733/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1420/sshd
tcp6       0      0 :::22                   :::*                    LISTEN      1420/sshd
udp        0      0 127.0.0.53:53           0.0.0.0:*                           733/systemd-resolve
udp        0      0 10.0.2.15:68            0.0.0.0:*                           2030/systemd-networ
root@ubuntu-vscode:/etc/systemd/system#
root@ubuntu-vscode:/etc/systemd/system# systemctl status code-server
● code-server.service - code-server
   Loaded: loaded (/etc/systemd/system/code-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-12-24 09:12:05 UTC; 1min 5s ago
 Main PID: 2942 (code-server)
    Tasks: 23 (limit: 2317)
   CGroup: /system.slice/code-server.service
           ├─2942 /home/code/bin/code-server --host 127.0.0.1 --user-data-dir /home/code/data --auth password
           └─2965 /home/code/bin/code-server --host 127.0.0.1 --user-data-dir /home/code/data --auth password
```

As a result, you've set up the code-server to run as a system service.

Step 3 - Generate SSL Letsencrypt

In this step, we will generate the SSL letsencrypt using the certbot tool for securing the code-server.

Install the certbot tool using the apt command below.

```
sudo apt install certbot -y
```

Once the installation is complete, generate the SSL letsencrypt using the certbot command below.

```
certbot certonly --standalone --agree-tos -m myemail@gmail.com -d vscode.hakase-labs.io
```

Once it's complete, your certificates will be located at the '/etc/letsencrypt/live/vscode.hakase-labs.io/' directory.

```
ls -lah /etc/letsencrypt/live/vscode.hakase-labs.io/
```

Now you've generated the SSL Letsencrypt for securing the code-server installation using the certbot tool.

Step 4 - Setup Nginx as a Reverse Proxy

In this step, we will install the Nginx web server and set up it as a reverse proxy for the code-server with SSL enabled on top of it.

Install Nginx package using the apt command below.

```
sudo apt install nginx -y
```

Once the installation is complete, go to the '/etc/nginx/sites-available' directory and create a new virtual host configuration 'code-server'.

```
cd /etc/nginx/sites-available/
vim code-server
```

Now change the domain name and path of SSL with your own and paste the configuration into it.

```
server {
 listen 80;
 server_name vscode.hakase-labs.io;
 # enforce https
 return 301 https://$server_name:443$request_uri;
}

server {
 listen 443 ssl http2;
 server_name vscode.hakase-labs.io;

 ssl_certificate /etc/letsencrypt/live/vscode.hakase-labs.io/fullchain.pem;
 ssl_certificate_key /etc/letsencrypt/live/vscode.hakase-labs.io/privkey.pem;

 location / {
 proxy_pass http://127.0.0.1:8080/;
 proxy_set_header Host $host;
 proxy_set_header Upgrade $http_upgrade;
 proxy_set_header Connection upgrade;
 proxy_set_header Accept-Encoding gzip;
 }
}
```

Save and close.

Now activate the 'code-server' virtual host, test the nginx configuration and make sure there is no error.

```
ln -s /etc/nginx/sites-available/code-server /etc/nginx/sites-enabled/
nginx -t
```

```
root@ubuntu-vscode:~#
root@ubuntu-vscode:~#
root@ubuntu-vscode:~# cd /etc/nginx/sites-available/
root@ubuntu-vscode:/etc/nginx/sites-available# vim code-server
root@ubuntu-vscode:/etc/nginx/sites-available#
root@ubuntu-vscode:/etc/nginx/sites-available# ln -s /etc/nginx/sites-available/code-server /etc/nginx/sites-enabled/
root@ubuntu-vscode:/etc/nginx/sites-available# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@ubuntu-vscode:/etc/nginx/sites-available# _
```

After that, restart the nginx service and add it to the system boot.

```
systemctl restart nginx
systemctl enable nginx
```

Now the Nginx service is up and running as a reverse proxy for the code-server. Check it using the command below.

```
netstat -plntu
systemctl status nginx
```

And you will get the result as below.

```
root@ubuntu-vscode:/etc/nginx/sites-available#
root@ubuntu-vscode:/etc/nginx/sites-available# systemctl restart nginx
root@ubuntu-vscode:/etc/nginx/sites-available# systemctl enable nginx
Synchronizing state of nginx.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nginx
root@ubuntu-vscode:/etc/nginx/sites-available#
root@ubuntu-vscode:/etc/nginx/sites-available# netstat -plntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      11561/nginx: master
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      11561/nginx: master
tcp        0      0 127.0.0.1:8080          0.0.0.0:*               LISTEN      11376/code-server
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      23160/systemd-resol
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1420/sshd
tcp6       0      0 :::80                   :::*                    LISTEN      11561/nginx: master
tcp6       0      0 :::22                   :::*                    LISTEN      1420/sshd
udp        0      0 127.0.0.53:53           0.0.0.0:*                           23160/systemd-resol
udp        0      0 10.0.2.15:68            0.0.0.0:*                           23135/systemd-netwo
root@ubuntu-vscode:/etc/nginx/sites-available#
root@ubuntu-vscode:/etc/nginx/sites-available# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2019-12-24 10:13:03 UTC; 1min 1s ago
     Docs: man:nginx(8)
 Main PID: 11561 (nginx)
    Tasks: 3 (limit: 2317)
   CGroup: /system.slice/nginx.service
           ├─11561 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
           ├─11563 nginx: worker process
           └─11564 nginx: worker process
```

The Nginx service is up and running on the Ubuntu 18.04 server with the HTTP and HTTPS ports enabled on top of it.

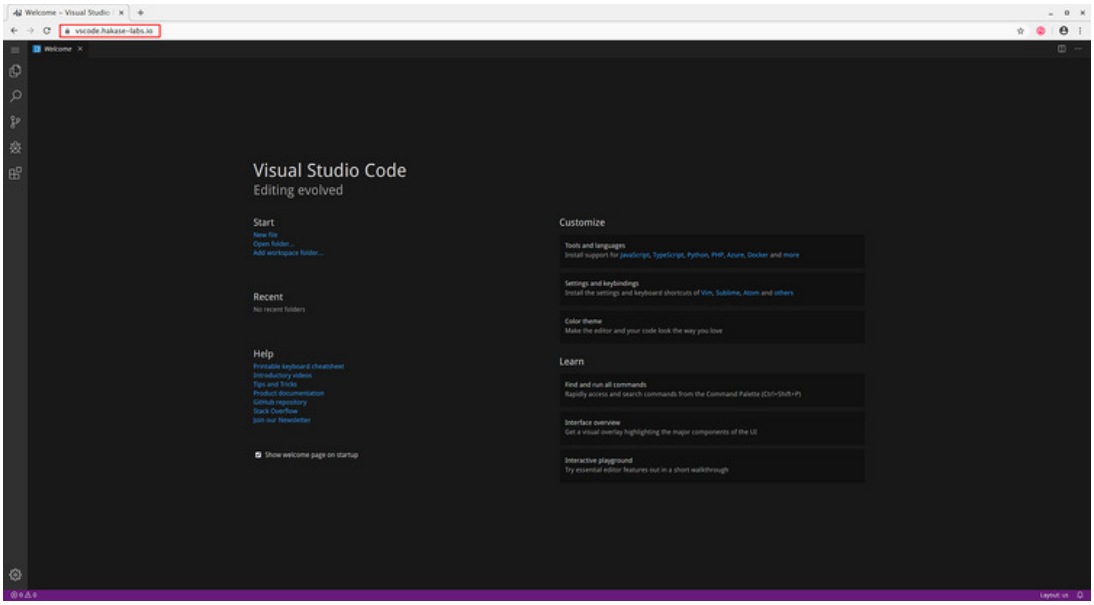
Step 5 - Testing

Open your web browser and type the URL of your code-server installation.

https://vscode.hakase-labs.io/

Log in with your password that you've configured at the code-server service file.

Once the password is correct, you will get the VS Code editor on your web browser as below.



As a result, you've installed the code-server on Ubuntu 18.04 server with Nginx as a reverse proxy and securing the code-server installation with SSL Letsencrypt.

Reference

- https://github.com/cdr/code-server

×

This feature is only available to subscribers. Get your subscription here.