

# How to Install Supabase on Ubuntu 20.04

Learn how to install Supabase on Ubuntu 20.04 with our step-by-step guide. Set up this powerful open-source Firebase alternative for your next project quickly and easily.

---

# Contents

01	Introduction	3
02	Prerequisites	3
03	Installation	3
04	Securing Supabase with an Nginx Reverse Proxy	7
05	Finishing Steps	9

# Introduction

---

In this tutorial, you will learn how to install and configure Supabase on Ubuntu 20.04. You will also learn how to secure your Supabase instance with secure secrets and a reverse proxy.

Supabase is a Firebase alternative that offers a PostgreSQL database, user authentication, storage, and a real-time API through a web interface. Supabase also has open-source API libraries that allow for easy interaction with other applications.

## Prerequisites

---

Before you begin, you should:

- [Deploy an Ubuntu 20.04 server](#) with at least 2 GB of RAM.
- Create a non-root user with sudo privileges.
- Log in to your server as a non-root user.
- [Ensure the server is fully updated](#).
- Open port 433 on your [Vultr firewall](#) or your [ufw](#) (if applicable).

You will also need a domain name that points to your server. This is because the SSL certificate generator (Certbot/Let's Encrypt) does not offer SSL certificates for IP addresses. You will need to create a domain name that points to your server. It is also possible to use HTTP instead, but this is not recommended.

## Installation

---

### Ngix, Certbot, and Git

1. Install Nginx and Git.

```
$ sudo apt install nginx git
```

## 2. Uninstall any old versions of Certbot and Docker.

```
$ sudo apt remove certbot docker docker.io containerd runc
```

## 3. Update the snap installer.

```
$ sudo snap install core; sudo snap refresh core
```

## 4. Install Certbot using `snap`.

```
$ sudo snap install --classic certbot
```

## 5. Run Certbot and follow the prompts to enter your domain name and redirect all traffic to HTTPS.

```
$ sudo certbot certonly --standalone
```

## 6. Take note of your certificate and private key paths when provided. It will be different depending on the domain used.

```
Certificate Path: /etc/letsencrypt/live/example.com/fullchain.pem  
Private Key Path: /etc/letsencrypt/live/example.com/privkey.pem
```

If you used a different SSL provider, ensure the certificate and private key files are stored somewhere on your system and that you know the full file path to them.

## Docker

### 1. Install Docker using `snap`.

```
$ sudo snap install docker
```

### 2. Clone the Supabase repository.

```
$ git clone --depth 1 https://github.com/supabase/supabase.git
```

3. Open the `docker` folder.

```
$ cd supabase/docker
```

4. Copy the `.env.example` file to `.env`.

```
$ cp .env.example .env
```

5. Open the `.env` file in your text editor.

```
$ nano .env
```

6. Open a strong password generator like Bitwarden in your browser, and generate a new password. It should contain more than 25 characters.

```
https://bitwarden.com/password-generator/
```

7. Replace the `POSTGRES_PASSWORD` value in the `.env` file with the password you generated.

```
POSTGRES_PASSWORD=<password>
```

8. Generate another password with more than 32 characters and no special characters. Replace the `JWT_SECRET` value in the `.env` file with the newly generated password.

```
JWT_SECRET=<new password>
```

9. Use your `JWT_SECRET` to generate an `ANON_KEY` on the Supabase website. Copy and paste your `JWT_SECRET`, switch the **Preconfigured Payload** type to `ANON_KEY`, press **Generate JWT**, and copy the **Generated Token** result into the `ANON_KEY` value in the `.env` file.

```
https://supabase.com/docs/guides/hosting/overview#api-keys
```

```
ANON_KEY=<generated key>
```

10. Do the same for the `SERVICE_KEY` value while using the same `JWT_SECRET`. Paste it as the `SERVICE_ROLE_KEY` value in the `.env` file.

```
SERVICE_ROLE_KEY=<generated key>
```

11. Close your text editor and save your changes by using `Control + X`, then `Y`, followed by `Enter`.
12. Navigate to the `volumes/api` folder and open `kong.yml` in your text editor.

```
$ cd volumes/api
$ nano kong.yml
```

13. Under `consumers`, replace the `anon` user's `key` with the `ANON_KEY` value from the `.env` file.

```
consumers:
- username: anon
  keyauth_credentials:
  - key: [anon key]
```

14. Replace the `service_role` user's `key` with the `SERVICE_ROLE_KEY` value from the `.env` file.

```
consumers:
- username: anon
  keyauth_credentials:
  - key: [anon key]
- username: service_role
  keyauth_credentials:
  - key: [service_role key]
```

15. Close your text editor again by using `Control + X`, then `Y`, followed by `Enter`.
16. Run Supabase by using `docker-compose` in detached mode. This may take 10-15 minutes.

```
$ sudo docker-compose up -d
```

17. Check that Supabase is running by using `docker`. The status should be `Up`.

```
$ sudo docker ps
```

```
STATUS
```

```
Up x seconds/minutes
```

You have now successfully installed Supabase and have obtained a signed SSL certificate.

## Securing Supabase with an Nginx Reverse Proxy

You can now use your SSL certificate and Nginx to secure your Supabase installation. Make sure to replace `example.com` with your chosen domain name or IP address.

1. Remove the Nginx default configuration file.

```
$ sudo rm /etc/nginx/sites-enabled/default
```

2. Create and open the new configuration file in Nginx's `sites-available` directory in your text editor.

```
$ sudo nano /etc/nginx/sites-available/supabase
```

3. Paste the following into the file and replace `example.com` with your domain name or IP address. Ensure that the `ssl_certificate` and `ssl_certificate_key` lines point to your SSL certificate.

```
upstream supabase {
    server localhost:3000;
}
```

```
server {
    listen 443 ssl http2;
    server_name example.com;

    gzip on;

    ssl_certificate /etc/letsencrypt/live/example.com/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/example.com/privkey.pem;
    ssl_session_cache builtin:1000 shared:SSL:10m;
    ssl_session_cache shared:MySSL:10m;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    # REST API
    location ~ ^/rest/v1/(.*)$ {
        proxy_set_header Host $host;
        proxy_pass http://kong:8000;
        proxy_redirect off;
    }

    # Authentication
    location ~ ^/auth/v1/(.*)$ {
        proxy_set_header Host $host;
        proxy_pass http://kong:8000;
        proxy_redirect off;
    }

    # Realtime
    location ~ ^/realtime/v1/(.*)$ {
        proxy_redirect off;
        proxy_pass http://kong:8000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection $connection_upgrade;
        proxy_set_header Host $host;
    }
}
```

This Nginx configuration will serve Supabase on port 443, and will use the SSL certificate and private key you generated earlier. It will also point the `/rest/v1/`, `/auth/v1/` and `/realtime/v1/` routes to the Kong API server.

4. Exit your text editor and save changes by pressing Control + X, then Y, followed by Enter.
5. Create a link to the new configuration file in Nginx's `sites-enabled` directory.

```
$ sudo ln -s /etc/nginx/sites-available/supabase /etc/nginx/sites-enabled/supabase.conf
```

6. Test the configuration file. If the test is successful, you will see the `syntax is ok`, and the `test is successful` messages.

```
$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

7. Reload Nginx to apply your changes.

```
$ sudo /etc/init.d/nginx reload
```

## Finishing Steps

You should now navigate to your Supabase installation.

```
https://example.com
```

From there you can configure your database, authentication, and file storage.

Congratulations, you have successfully installed Supabase and secured it using an SSL certificate and an Nginx reverse proxy.

### Additional Resources

- [Supabase Documentation](#)
- [Supabase Website](#)
- [Nginx Documentation](#)



**VULTR**

