

Other

- [What to Do If You Have Exceeded Your Bandwidth Limit](#)
- [What to Do If Your Website Is Flagged as an Attack Site](#)
- [DotRoll Web Hosting Security Features and Solutions](#)
- [How to Set Up a Cron Job](#)
- [What Is SSL](#)
- [How to Renew Your Hosting Service](#)
- [How to Create and Edit Files via SSH](#)
- [How to Transfer a Service to Another Billing Account](#)
- [Setting Up Supabase Using Docker Desktop](#)
- [Setting Up Supabase via Command Line](#)

What to Do If You Have Exceeded Your Bandwidth Limit

If you have received a notification email for accessing the bandwidth traffic limit, you can opt in to <https://admin.dotroll.com> for additional data traffic after logging in.

Steps for ordering:

1. After logging in, click on **Services / My Services**.
2. In the **Actions** box in the left sidebar, click the **View Available Addons**.
3. On the page that appears, select the data bundle you want to purchase, then click **Order Now**.
4. Then click the **Checkout** button and check the billing information.
5. Select the form of payment and select "I have read and agree to the Terms of Service" and click **Complete Order**.

What to Do If Your Website Is Flagged as an Attack Site

Unfortunately we are unable to help with the reported attack site because we did not put this type of blocking on Google, but we did not. This type of blocking is useful because the visitor's tool does not get infected, but it's pretty uncomfortable.

To resolve this issue, follow these steps:

1. For devices from which you signed in to your site, or to scan for a virus on your web site.
2. Then log on to <https://admin.dotroll.com> and modify your hosting password as well as your FTP users' passwords.
3. If you have received a CXS notification, you can view the files in the notification email if you do not find the code code for the program code in the code then remove them. If you are using a content management system (CMS), you should definitely upgrade both the content manager and the installed add-on modules and templates.

If you have not created the website, contact the developer of the website, as we will not be able to help you with programming issues.

Google will review websites for up to 90 days, if you do not want to wait for it, you can do this by following these steps to speed up blocking.

1. If the site is cleaned from harmful code sequences, please read the [Google Search Console](#)

In addition, the StopBadware site, which is <http://www.stopbadware.org/>

2. Next, you'll need to look at the Google Diagnostic page that you can access (at the end of the URL you want to replace):

http://www.google.com/safebrowsing/diagnostic?site=http://your_domain_name.tld

Note: At the end of the URL, replace the yourn_domain_name.tld with the domain name that is experiencing the problem.

3. Then, sign in to [Google Webmaster Tools](#) where you can apply for a review.
4. Find it on your site with the Badware Website Clearinghouse, then click the links in the Report column to get a review of the pages by clicking the "Click to Request Review" link on the left. <http://www.stopbadware.org/home/reportsearch>

5. If you have been granted phishing as a page, please also review the link below.

http://sb.google.com/safebrowsing/report_error/

Wait for a few days, or maybe seven weeks for the systems to pass through the news of infection, and will also be able to tell you about success in Webmaster Tools.

If you have any further questions, please feel free to contact our Customer Service.

DotRoll Web Hosting

Security Features and Solutions

While developing DotRoll non-free hosting, we paid special attention to security. These security features include:

SYSTEM USERS

Unlike usual hosting solutions, security starts at the system level. Every hosting account is a system user. This allows us to run applications like Perl, Python, Ruby on Rails.

HARDENED KERNEL

Our kernel has been hardened to suit the security needs of our clients. For example, users may only see their own processes.

VIRTUAL MINI ENVIRONMENT

Every user has a very own miniature environment (chroot) in which the applications run and FTP is jailed into. This allows us to provide users with SSH access.

OUTGOING CONNECTIONS

As we reckon the need for using API's, we have developed a solution to monitor abuse and thus allow free access to external sites to our users. PHP, for example has been fitted with all the technology we could pack into it (SOAP, XML-RPC, etc).

E-MAIL SOLUTIONS

All outgoing mail is bound to users transparently. Every user may send only from a domain belonging to that very account. Outgoing mail is digitally signed, which more and more e-mail providers recognize as a sign of trust and verifiability. Blacklists and e-mail traffic are constantly monitored so your legitimate e-mail may pass unhindered and get right where it belongs - in your

user's inboxes.

How to Set Up a Cron Job

There are two options for setting.

1. For the file you want to run on a file in a timely manner, at least the owner has to run the run right (0700). Also, at the beginning of the file (in the first line) you have to specify the appropriate interpreter for the file type.

For PHP file:

```
#!/usr/bin/php
```

For Shell script:

```
#!/bin/bash
```

In this case, the file should be given in the timed scripts menu in the following format:
/home/username/public_html/cron.php

2. Only enter the interpreter in the Timed Scripts menu before the file to run, in the following format:

For PHP file:

```
php /home/username/public_html/cron.php
```

For Shell script:

```
/bin/sh /home/username/public_html/file.sh
```

In the latter case, you do not have to configure the file for the file.

What Is SSL

SSL / TLS is a protocol that provides secure data transfer. If supported by two endpoints of the connection, almost any other protocol can be transmitted via SSL.

An important feature of SSL is its signature. The keys can be signed and authenticated. Software vendors (such as browsers) embed embedded browsers into a list of the signatories they recognize, which are automatically accepted by the software. In other cases, the software will ask you to accept the key.

How to Renew Your Hosting Service

Follow these steps to renew:

1. Log in at <https://admin.dotroll.com/> with the email address you entered, when you registered your account.
2. In the top menu bar, click on **Services / My Services**.
3. Click on the green **Active** icon.
4. After the page loaded, below Hosting Renewals, click on **Renew service for 1 year** button.

Our system then draws up a billing account that allows you to extend the service immediately via credit card or PayPal.

How to Create and Edit Files via SSH

There are several ways to create a file through SSH. In this article, we will present the file creation with two types of word processors.

- **nano** - for beginner users
- **vim** - for advanced users

If you are editing files from a cumbersome command line, we recommend that you create a file via FTP, you can use this description below:

[Create and modify a file via FTP](#)

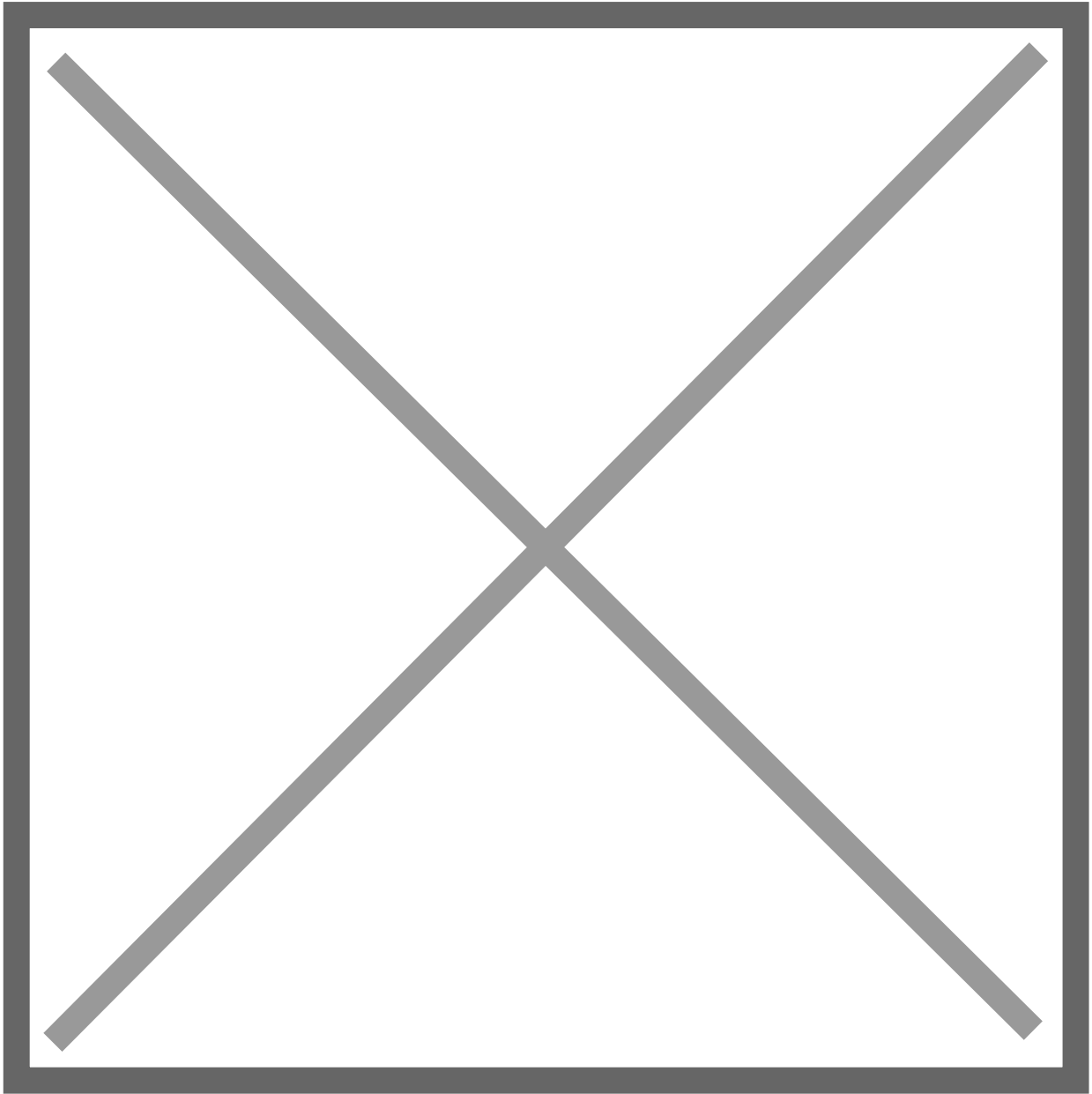
The following description is just an example of how to create or modify a file using SSH. Following the description, make sure you create the file at the right place. If you modify it, edit the file you want.

CREATING A FILE USING NANO

1. Log in to the server on SSH.
2. Navigate to the directory where you want to create or edit the file.
3. Next, type the **nano** command and then divide by space, enter the name of the file you want to create/edit.

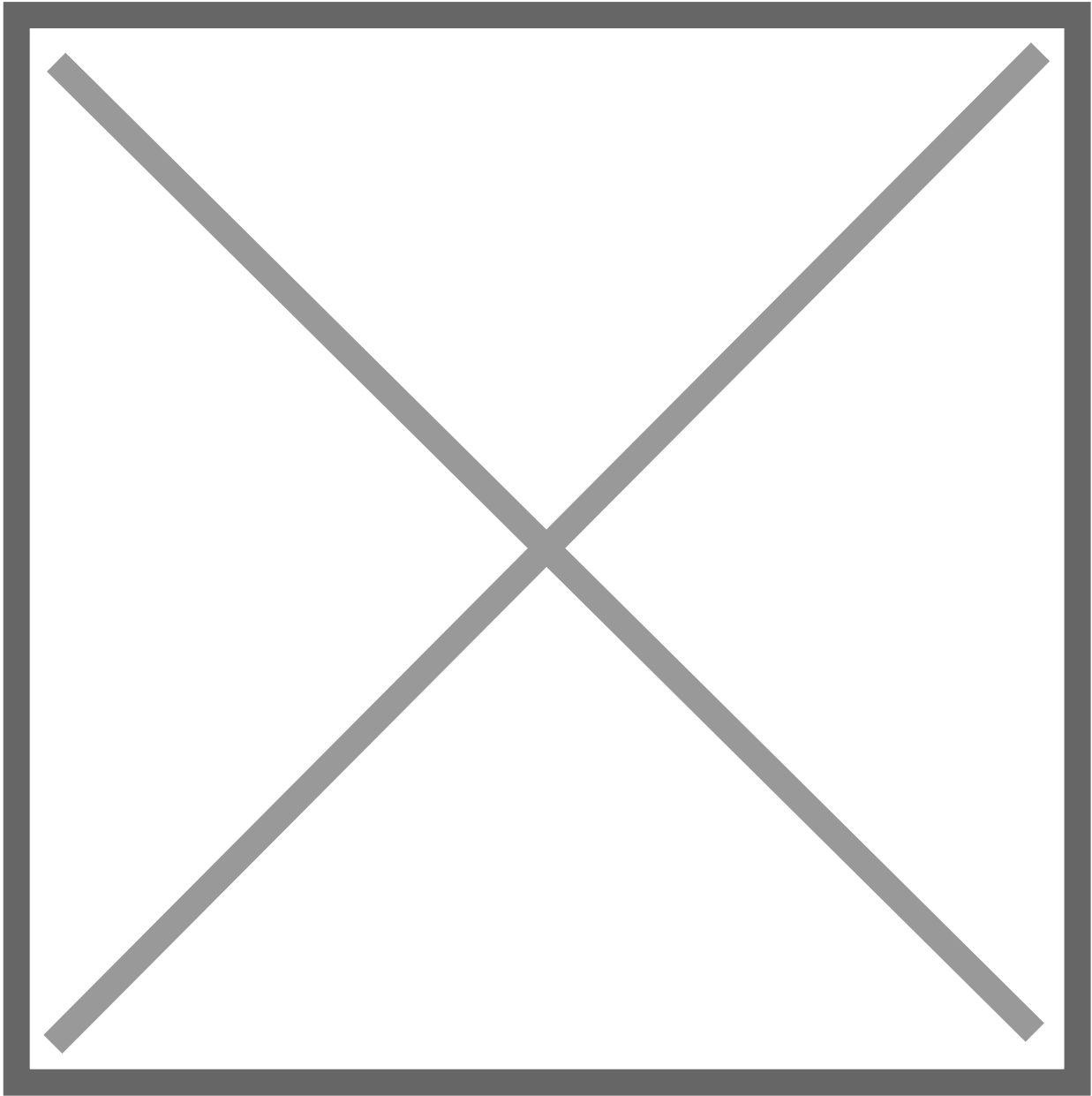
```
[server] $ nano .htaccess
```

4.



Next, start typing the contents of the file.

5. When you have uploaded the file with the required content, hold down the **Ctrl** key and press the **O** key (Ctrl + O).



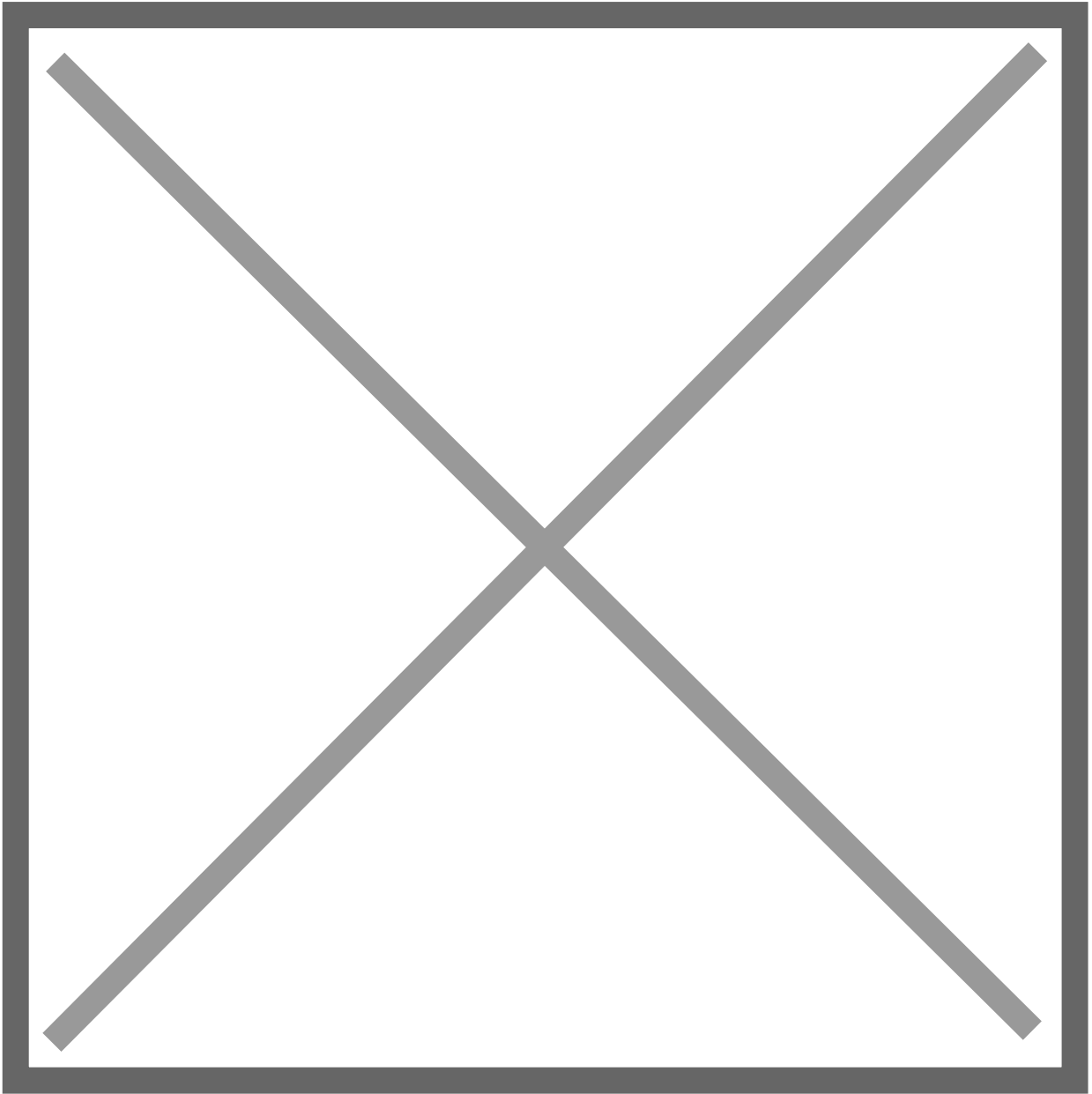
At the bottom of the screen, the editor will ask you again about the name of the file you previously specified.

6. You can confirm the file by pressing Enter.
7. Press **Ctrl+X** to exit the word processor.

CREATING A FILE USING VIM

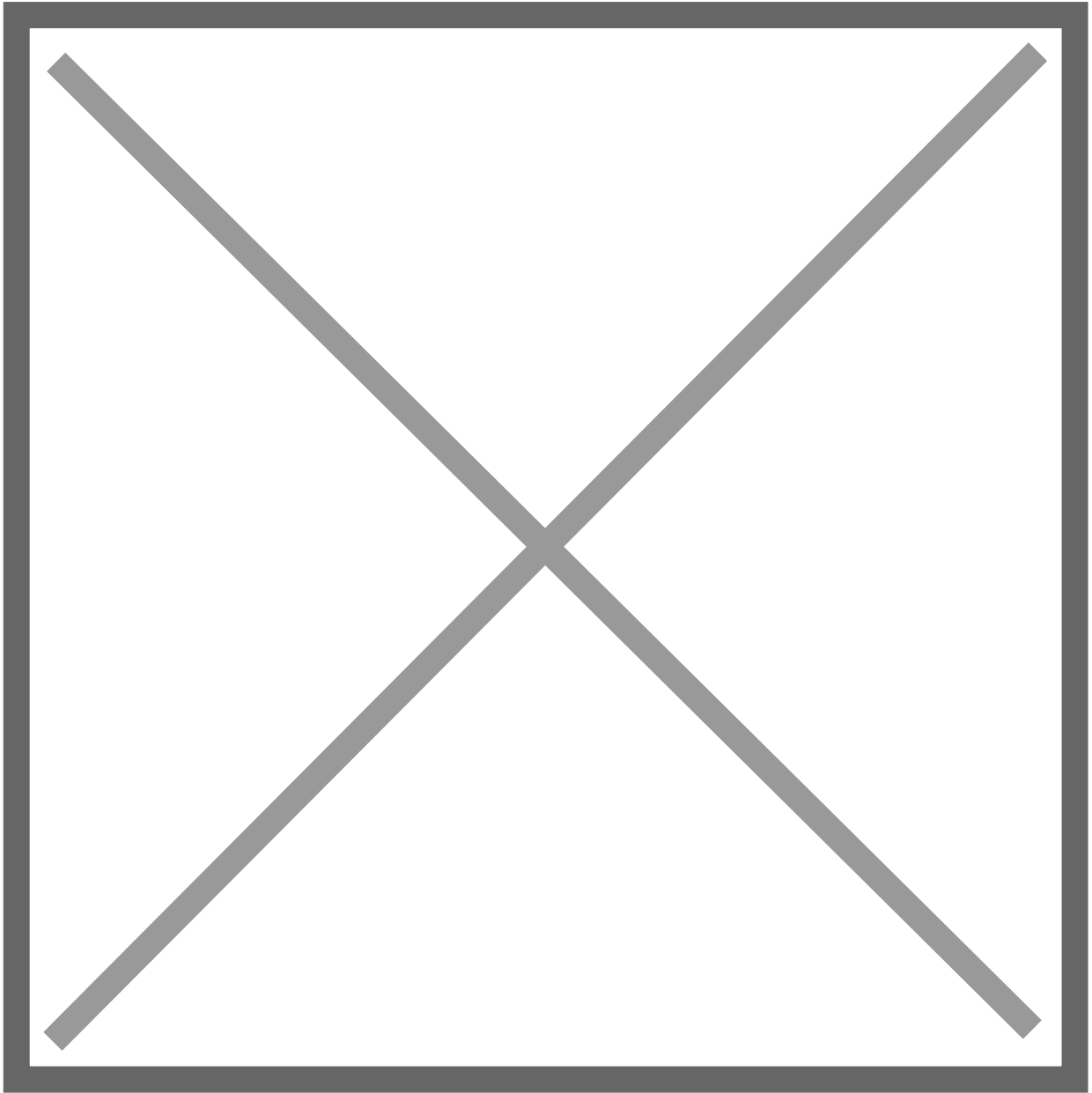
1. Log in to the server on SSH.
2. Navigate to the directory where you want to create or edit the file.
3. Next, type the **vim** command, and then divide by space, enter the name of the file you want to create/edit.

```
[server] $ vim .htaccess
```



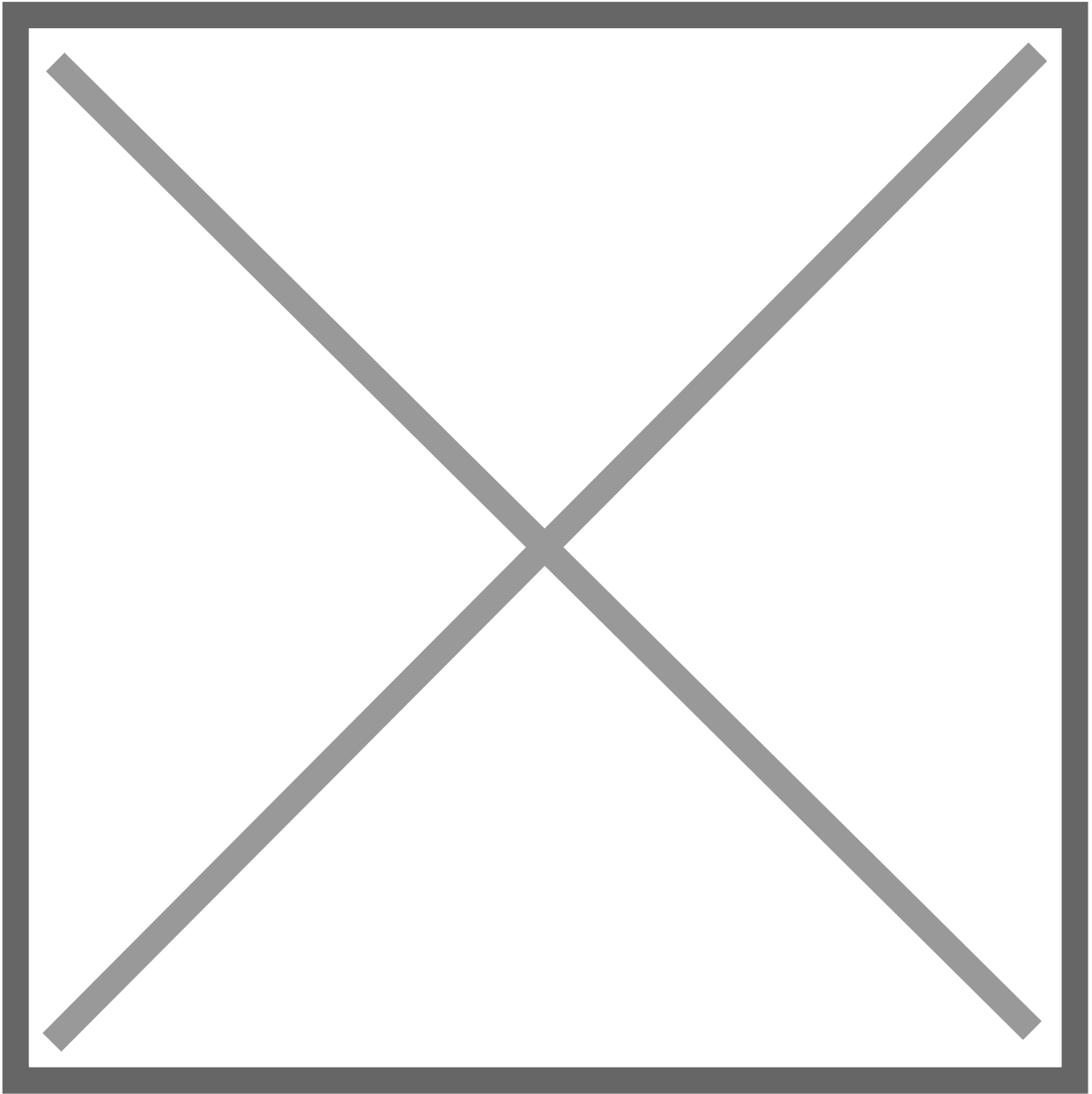
4.

After the word processor starts, you can upload the file by pressing the **⌘** key as it changes to INSERT mode.



5. When you upload the file with the required content, use **Esc** to exit the insertion mode.

6.



To save the file, type **:wq**, and then use the **Enter** key to save the file.

How to Transfer a Service to Another Billing Account

Transfer steps:

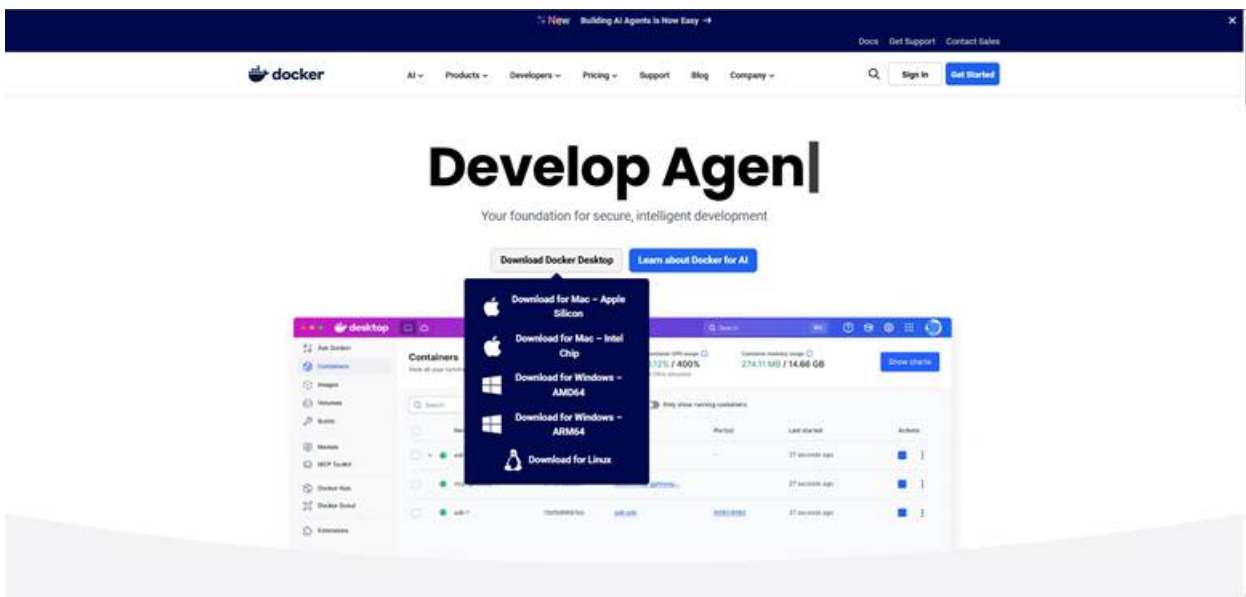
- Log in to our system with your email address and password, then click on the **Services / My Services** menu item in the top menu line.
- Next, click the service you want to transfer, and then click **Transfer service to user** in the **Actions** box on the left.
- On the page that appears, in the **Transfer to user:** field, enter the email address or username of the person or organization to whom you want to transfer the service name, and then click **Check user name** button.
- Our system then checks to see if this is a genuine bill payer based on the information you provided, and then displays the information. If you find the information correct, click the **Confirm** button.
- The interface will then display the services already handed over, here you have the option to cancel the previous handover.
- The system alerts the other user after logging in that services are waiting to be received.

Setting Up Supabase Using Docker Desktop

If you're new to self-hosting or want a simple, visual setup, Docker Desktop offers the easiest way to run Supabase locally. It provides a graphical interface for managing containers, configuring ports, and ensuring your data persists across restarts—no command-line work required.

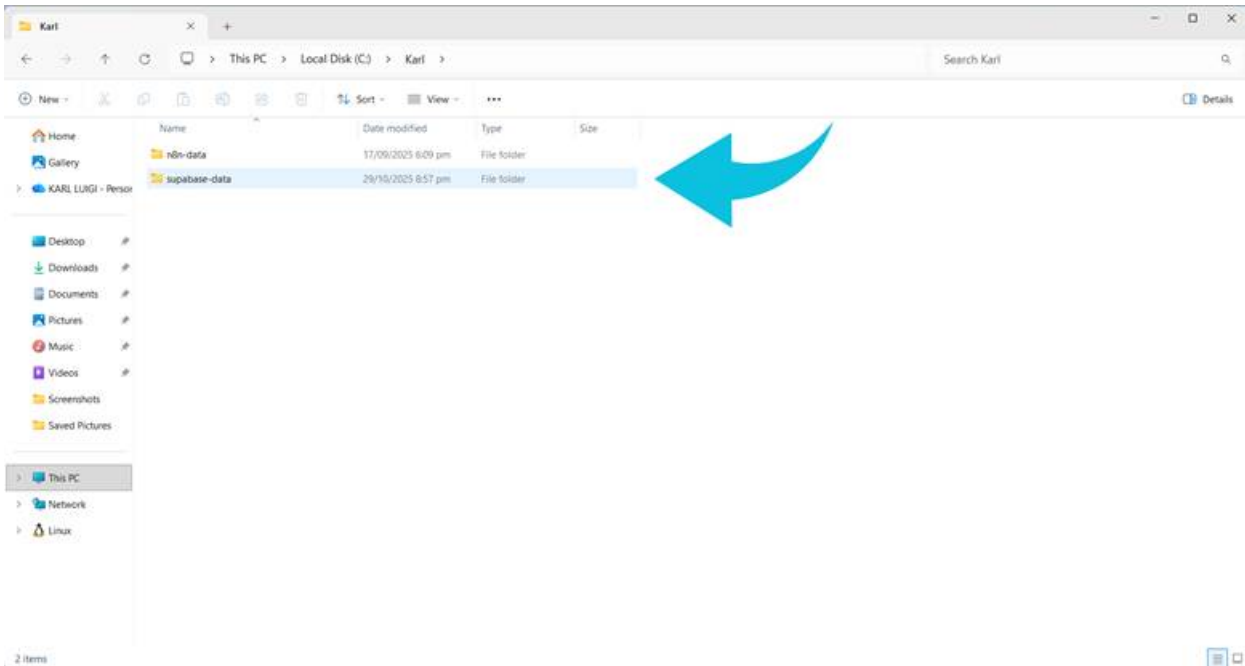
Step 1. Install Docker Desktop

- [Download Docker Desktop](#) for your operating system.
- Install it following the installer instructions.
Windows users: Enable **WSL2** during installation and restart if prompted.
- Open Docker Desktop and make sure it's running. You should see the Docker whale icon in your system tray or menu bar.



Step 2. Prepare a Folder for Data

- Supabase stores data (PostgreSQL database, file uploads) in volumes. To make your data persistent:
 - **macOS / Linux:** `~/supabase-data`
 - **Windows:** `C:\supabase-data`



This folder ensures your data is safe even if you remove or update containers.

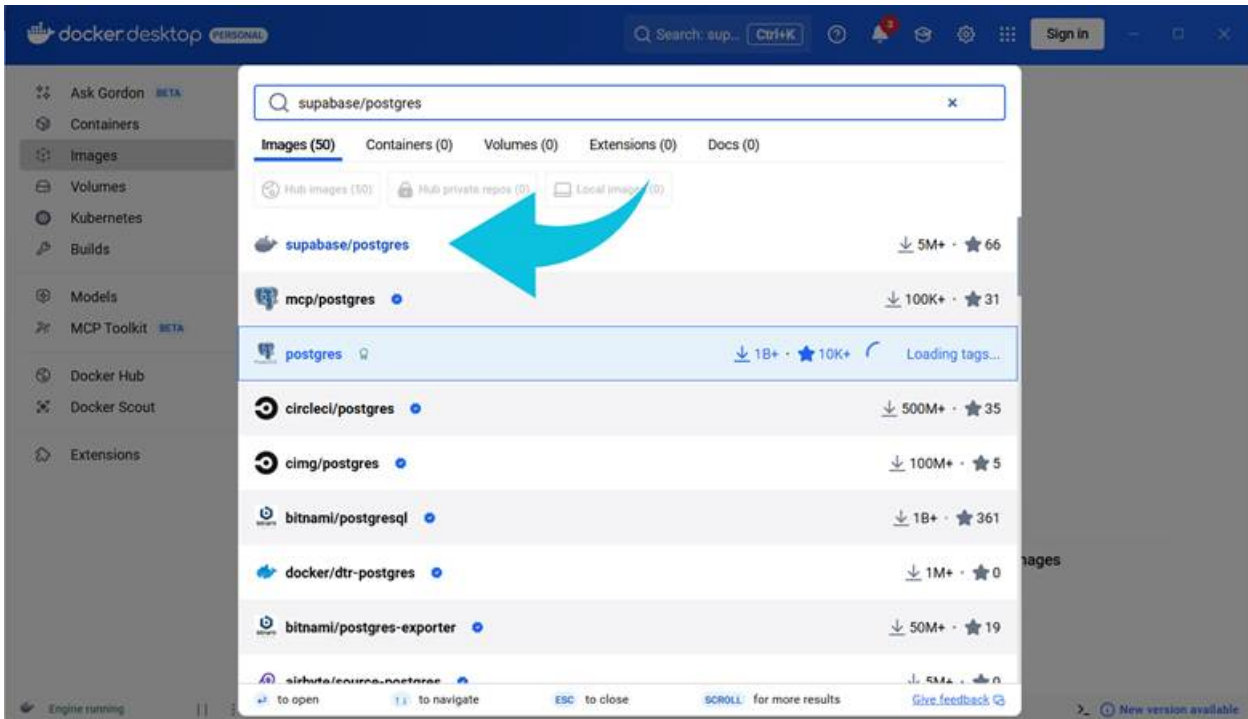
Step 3. Pull the Official Supabase Images

Supabase uses multiple Docker images for its services: PostgreSQL, Auth, Realtime, Storage, API, and Studio. You can pull them directly in Docker Desktop:

1. Open Docker Desktop and go to the **Images** tab.
2. In the search bar, enter the official image names:

- `supabase/postgres`
- `supabase/gotrue`
- `supabase/realtime`
- `supabase/storage-api`
- `supabase/postgrest`
- `supabase/studio`

3. Click **Pull** for each image to download it to your machine. Repeat for all images.



Step 4. Create Containers for Each Service

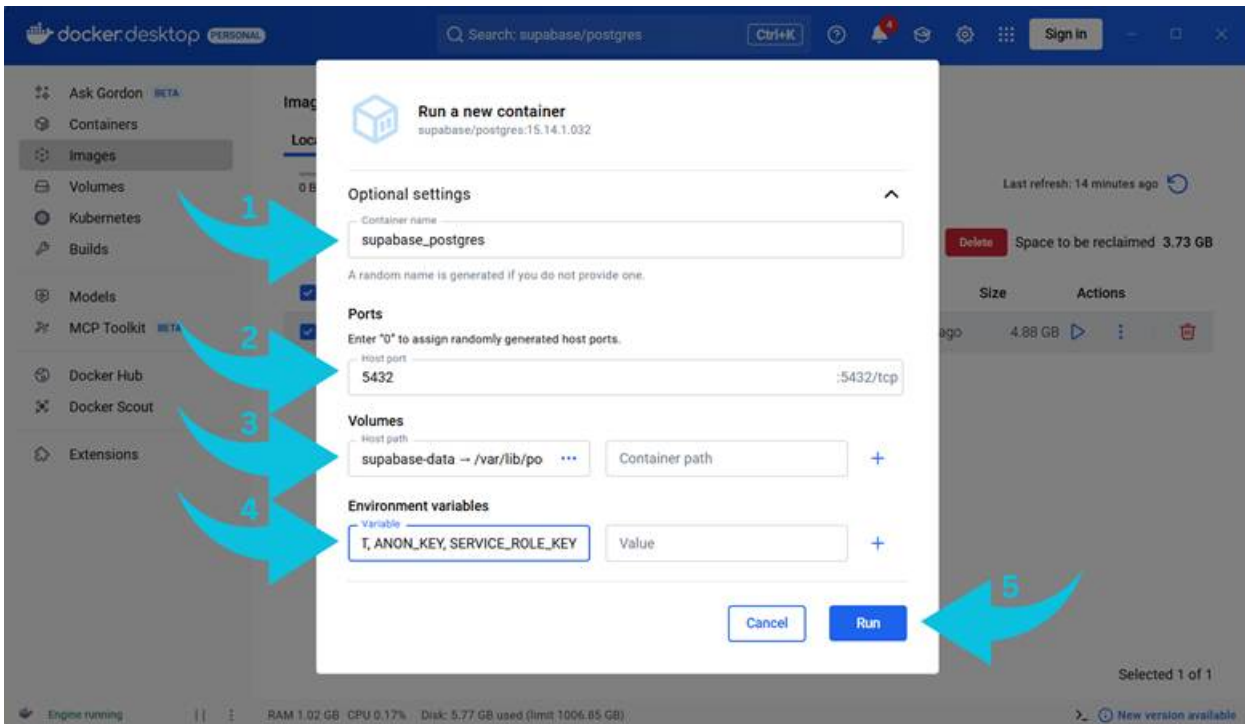
Once the images are downloaded, you need to create a container for each service:

1. Go to the **Containers/Apps** tab in Docker Desktop and click **+ Add Container**.
2. Select the image you just pulled.
3. Configure settings for each container:
 - **Name:** Give it a meaningful name (e.g., `supabase_postgres`, `supabase_auth`).
 - **Ports:** Map the container ports to host ports:
 - PostgreSQL: `5432`
 - Supabase Studio: `3000`
 - Others use default ports (check each service documentation).
 - **Volumes:** Bind your host folder to the container for persistence:
 - Example: `~/supabase-data` → `/var/lib/postgresql/data` for PostgreSQL
 - **Environment Variables:** Set essential keys:

- `POSTGRES_PASSWORD`, `JWT_SECRET`, `ANON_KEY`, `SERVICE_ROLE_KEY`

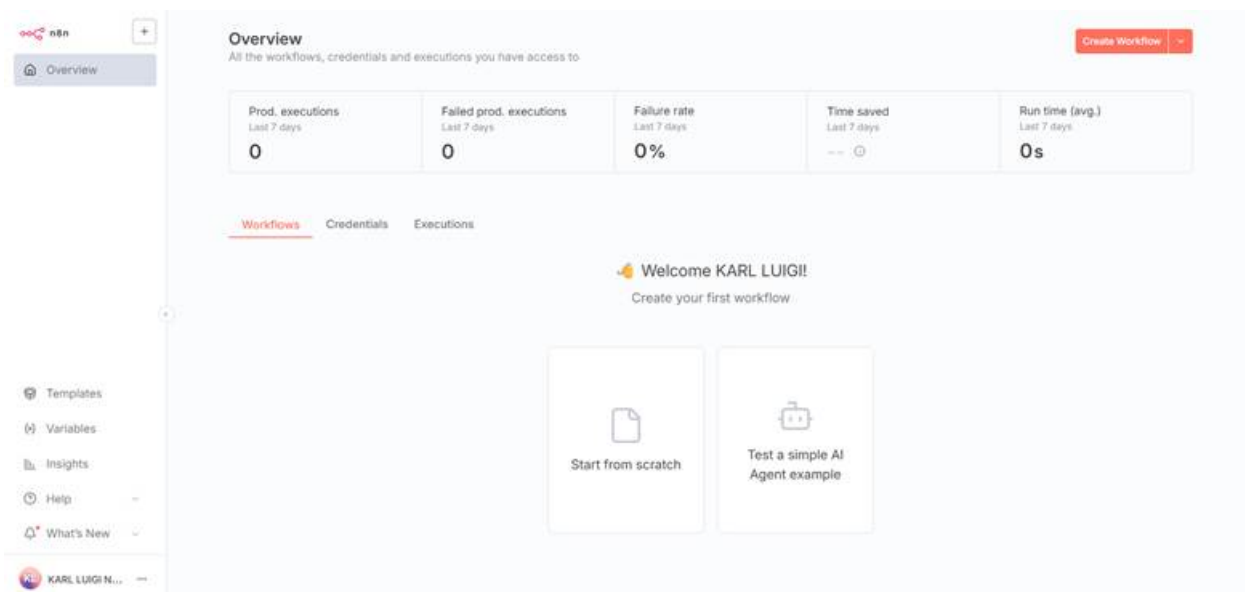
- **Restart Policy:** Always or Unless Stopped

4. Click **Run** to launch the container. Repeat for all Supabase services.



Step 5. Access Supabase Studio

- Once all containers are running, open your browser and go to: <http://localhost:3000>
- You now have full access to Supabase Studio and can manage your database, authentication, and storage visually.



Step 6. Updating Supabase

When new Supabase releases are available:

1. Stop and remove the containers using Docker Desktop.
2. Pull the updated images (Images tab → Pull).
3. Recreate containers using the same settings and volume mappings.

Because your data is stored in persistent volumes, it **remains safe** during updates.

Step 7. Maintenance and Monitoring

- Docker Desktop lets you monitor resource usage, logs, and container health.
- Regularly check logs for errors and ensure your data folder has sufficient disk space.
- For production or long-term local usage, consider automating container updates using **Watchtower**.

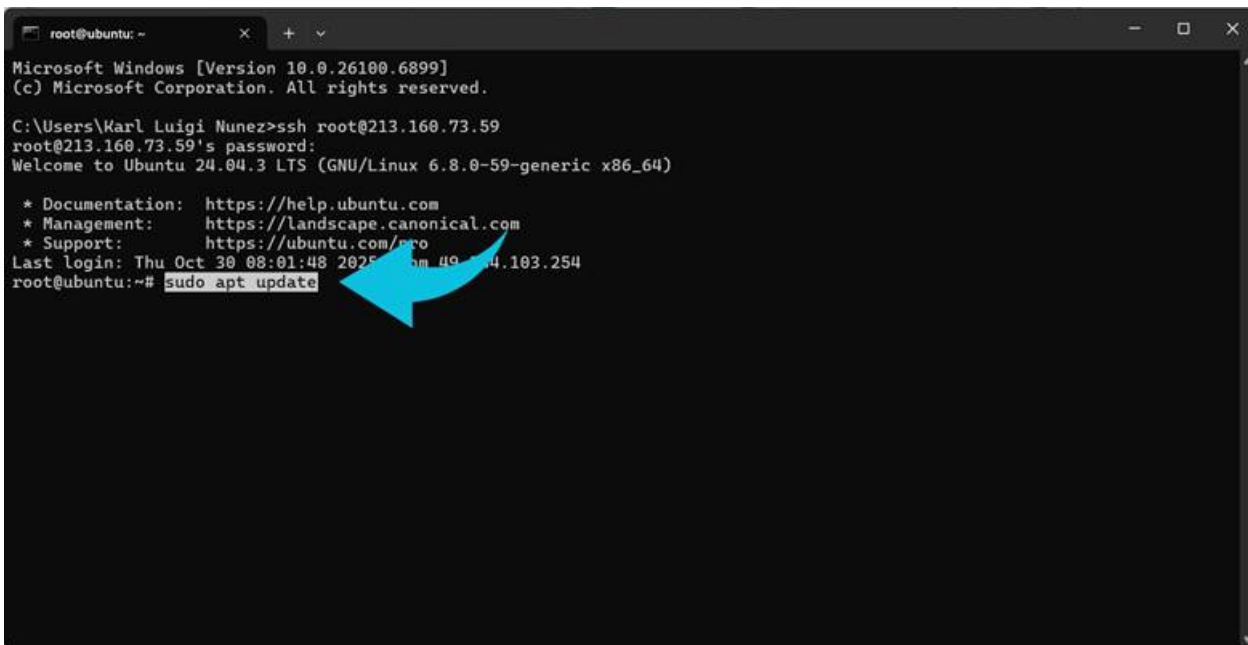
Setting Up Supabase via Command Line

For developers and teams who prioritize automation, version control, or deployment on remote servers, the command-line approach provides flexibility and repeatability. Using Docker and Docker Compose directly allows you to manage Supabase stacks efficiently and integrate the process into scripts or CI/CD pipelines.

Step 1. Install Docker and Docker Compose

- Linux:

`sudo apt update`



```
root@ubuntu: ~  
Microsoft Windows [Version 10.0.26100.6899]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\Karl Luigi Nunez>ssh root@213.160.73.59  
root@213.160.73.59's password:  
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-59-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:       https://ubuntu.com/pro  
Last login: Thu Oct 30 08:01:48 2025 from 49.14.103.254  
root@ubuntu:~# sudo apt update
```

`sudo apt install docker.io docker-compose -y`

```
root@ubuntu: ~
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Karl Luigi Nunez>ssh root@213.160.73.59
root@213.160.73.59's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-59-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
Last login: Thu Oct 30 08:01:48 2025 from 49.144.103.254
root@ubuntu:~# sudo apt update
Hit:1 http://de.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://de.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://de.archive.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease
Hit:6 https://deb.nodesource.com/node_22.x nodistro InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ubuntu:~# sudo apt install docker.io docker-compose -y
```

- **Windows/macOS:** Docker Desktop includes both tools, so no additional installation is needed.

Verify Docker is running before proceeding.

Step 2. Clone the Supabase Repository

`git clone https://github.com/supabase/supabase.git`

`cd supabase/docker`

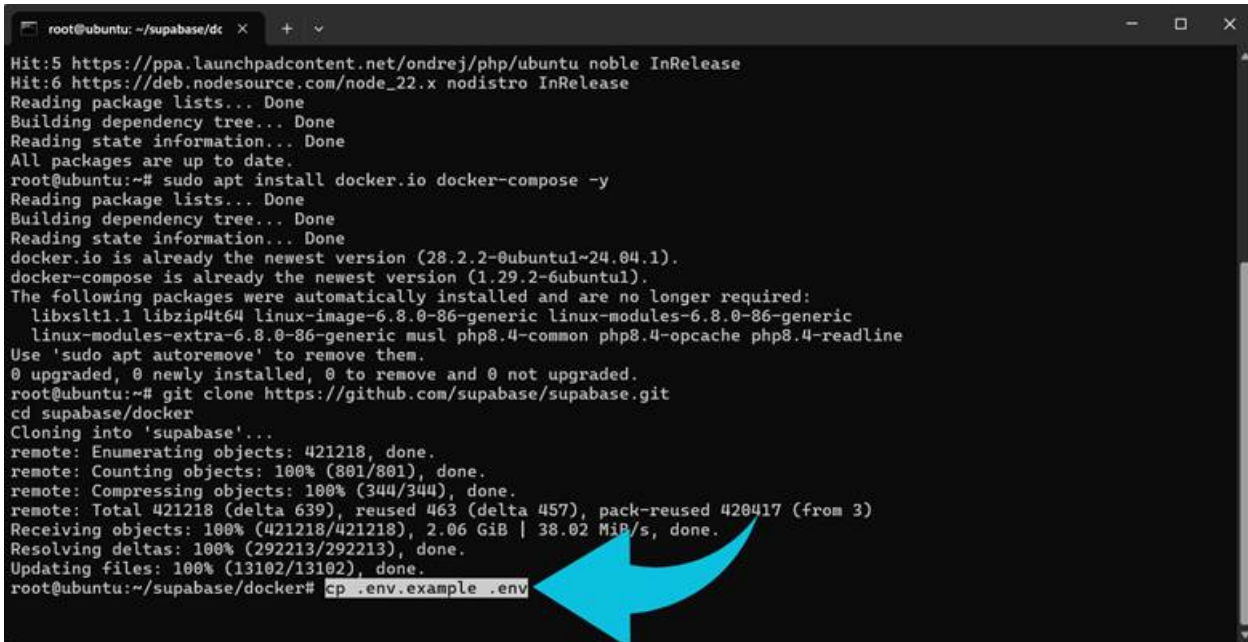
```
root@ubuntu: ~
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro
Last login: Thu Oct 30 08:01:48 2025 from 49.144.103.254
root@ubuntu:~# sudo apt update
Hit:1 http://de.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://de.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://de.archive.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease
Hit:6 https://deb.nodesource.com/node_22.x nodistro InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ubuntu:~# sudo apt install docker.io docker-compose -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (28.2.2-0ubuntu1~24.04.1).
docker-compose is already the newest version (1.29.2-6ubuntu1).
The following packages were automatically installed and are no longer required:
  libxslt1.1 libzip4t64 linux-image-6.8.0-86-generic linux-modules-6.8.0-86-generic
  linux-modules-extra-6.8.0-86-generic musl php8.4-common php8.4-opcache php8.4-readline
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# git clone https://github.com/supabase/supabase.git
cd supabase/docker
```

This repository contains the [docker-compose.yml](#) configuration defining all core services: PostgreSQL, APIs, authentication, Realtime, storage, and Supabase Studio.

Step 3. Configure Environment Variables

Copy the example environment file and customize it:

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



```
root@ubuntu: ~/supabase/dc
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease
Hit:6 https://deb.nodesource.com/node_22.x nodistro InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ubuntu:~# sudo apt install docker.io docker-compose -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (28.2.2-0ubuntu1~24.04.1).
docker-compose is already the newest version (1.29.2-6ubuntu1).
The following packages were automatically installed and are no longer required:
  libxslt1.1 libzip4t64 linux-image-6.8.0-86-generic linux-modules-6.8.0-86-generic
  linux-modules-extra-6.8.0-86-generic musl php8.4-common php8.4-opcache php8.4-readline
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# git clone https://github.com/supabase/supabase.git
cd supabase/docker
Cloning into 'supabase'...
remote: Enumerating objects: 421218, done.
remote: Counting objects: 100% (801/801), done.
remote: Compressing objects: 100% (344/344), done.
remote: Total 421218 (delta 639), reused 463 (delta 457), pack-reused 420417 (from 3)
Receiving objects: 100% (421218/421218), 2.06 GiB | 38.02 MiB/s, done.
Resolving deltas: 100% (292213/292213), done.
Updating files: 100% (13102/13102), done.
root@ubuntu:~/supabase/docker# cp .env.example .env
```

Edit `.env` to set your credentials and keys:

- `POSTGRES_PASSWORD`: PostgreSQL database password
- `JWT_SECRET`: authentication token secret
- `ANON_KEY` and `SERVICE_ROLE_KEY`: API access keys

Properly securing these values is critical, especially in production environments.

Step 4. Launch Supabase

Start the services in detached mode:

```
docker compose up -d
```

```
root@ubuntu: ~/supabase/dc X +
Hit:5 https://ppa.launchpadcontent.net/ondrej/php/ubuntu noble InRelease
Hit:6 https://deb.nodesource.com/node_22.x nodistro InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
root@ubuntu:~# sudo apt install docker.io docker-compose -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
docker.io is already the newest version (28.2.2-0ubuntu1~24.04.1).
docker-compose is already the newest version (1.29.2-6ubuntu1).
The following packages were automatically installed and are no longer required:
  libxslt1.1 libzip4t64 linux-image-6.8.0-86-generic linux-modules-6.8.0-86-generic
  linux-modules-extra-6.8.0-86-generic musl php8.4-common php8.4-opcache php8.4-readline
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ubuntu:~# git clone https://github.com/supabase/supabase.git
cd supabase/docker
Cloning into 'supabase'...
remote: Enumerating objects: 421218, done.
remote: Counting objects: 100% (801/801), done.
remote: Compressing objects: 100% (344/344), done.
remote: Total 421218 (delta 639), reused 463 (delta 457), pack-reused 420417 (from 3)
Receiving objects: 100% (421218/421218), 2.06 GiB | 38.8 MiB/s, done.
Resolving deltas: 100% (292213/292213), done.
Updating files: 100% (13102/13102), done.
root@ubuntu:~/supabase/docker# cp .env.example .env
root@ubuntu:~/supabase/docker# docker compose up -d
```

Docker Compose pulls all necessary images and runs the containers, including database, authentication, Realtime, API, and Studio.

Step 5. Verify and Access the Platform

Check that all containers are running:

```
docker ps
```

```
root@ubuntu: ~/supabase/dc X +
✓kong Pulled 43.4s
✓vector Pulled 45.0s
✓rest Pulled 50.6s
✓db Pulled 101.5s
✓analytics Pulled 138.6s
✓meta Pulled 87.4s
✓imgproxy Pulled 45.9s
✓supavisor Pulled 148.6s
✓realtime Pulled 115.2s
✓auth Pulled 63.1s
✓studio Pulled 136.5s
✓storage Pulled 131.6s
✓functions Pulled 126.1s
[+] Running 15/15
✓Network supabase_default Created 0.8s
✓Volume supabase_db-config Created 0.1s
✓Container supabase-vector Healthy 13.7s
✓Container supabase-imgproxy Started 8.2s
✓Container supabase-db Healthy 61.9s
✓Container supabase-analytics Healthy 58.0s
✓Container supabase-auth Started 59.1s
✓Container supabase-edge-functions Started 59.3s
✓Container supabase-rest Started 59.1s
✓Container realtime-dev.supabase-realtime Started 59.2s
✓Container supabase-meta Started 59.1s
✓Container supabase-kong Started 59.2s
✓Container supabase-pooler Started 59.3s
✓Container supabase-studio Started 59.0s
✓Container supabase-storage Started 62.4s
root@ubuntu:~/supabase/docker# docker ps
```

Then open Supabase Studio in your browser:

<http://localhost:3000>

The screenshot displays the n8n Overview dashboard. At the top left, the n8n logo and a '+' icon are visible. Below them is a navigation menu with 'Overview' selected. On the right, there is a 'Create Workflow' button. The main content area is titled 'Overview' and contains a summary of workflow statistics for the last 7 days:

Prod. executions	Failed prod. executions	Failure rate	Time saved	Run time (avg.)
0	0	0%	--	0s

Below the statistics, there are tabs for 'Workflows', 'Credentials', and 'Executions'. A welcome message reads 'Welcome KARL LUGI!' with the instruction 'Create your first workflow'. Two main action buttons are provided: 'Start from scratch' (with a document icon) and 'Test a simple AI Agent example' (with a robot icon). On the left sidebar, there are links for 'Templates', 'Variables', 'Insights', 'Help', 'What's New', and a user profile for 'KARL LUGI N...'.

You now have full access to your self-hosted backend.