How to Host a Joplin Server with Docker on Ubuntu

**Introduction**

Joplin is a robust note-taking app that is open-source and allows for synchronization on various devices. Running your own Joplin server gives you complete authority over your data and guarantees privacy. This manual will guide you on how to establish and manage a Joplin server on Ubuntu by utilizing Docker and Docker Compose.

Before proceeding, certain requirements must be met.

Prior to starting, below are the subsequent measures to take  
At Vultr, install a fresh Ubuntu 22.04 server.

In your DNS, construct a fully qualified domain name that points to the server's IP address.

Throughout the manual, you will find the URL joplin1.example1.com mentioned repeatedly.

To gain sudo access to the server as a non-root user, use SSH.

Set up Docker Compose in conjunction with Nginx.

Fundamental knowledge of Docker and managing systems.

1. Joplin Server Installation

1) Initiate a fresh folder in the /opt directory to house the Joplin Docker files.

*$ sudo mkdir /opt/joplin*

2) Go to the directory.

*$ cd /opt/joplin/*  
3) Employ a text editor to generate a fresh Joplin Docker Compose document.

*$ sudo nano joplin-docker-compose.yml*

4) Include the following information within the document

-> Feel free to modify the numbers 3454 and 8081 as needed; these are merely illustrations  
 -> Substitute https://joplin1.example1.com with your complete domain name.  
-> Adjust the values of POSTGRES\_PASSWORD, POSTGRES\_DATABASE, and POSTGRES\_USER to your preferred configuration.

version: '3'

services:

db:

image: postgres:13

volumes:

- ./data/postgres:/var/lib/postgresql/data

ports:

- "5432:5432"

restart: always

environment:

- POSTGRES\_PASSWORD=Your-Password-here

- POSTGRES\_USER=joplin-user

- POSTGRES\_DB=joplindb

app:

image: joplin/server:latest

container\_name: joplin-server

depends\_on:

- db

ports:

- "8081: 8081"

restart: always

environment:

- APP\_PORT=8081

- APP\_BASE\_URL=https://joplin1.example1.com

- DB\_CLIENT=pg

- POSTGRES\_PASSWORD=Your-Password-here

- POSTGRES\_DATABASE=joplindb

- POSTGRES\_USER=joplin-user

- POSTGRES\_PORT=5432

- POSTGRES\_HOST=db

This Docker Compose setup establishes a fresh Postgres database container alongside a Joplin Server that's active on port 8081.

5.Save the document, then close it.

6.Get the Joplin Server initiated.

$ sudo docker-compose -f joplin-docker-compose.yml up -d

7.Validate whether the Joplin Server is operational.

$ sudo docker ps

The expected outcome should align with the illustration presented below.

1. **Configure a Reverse Proxy with Nginx**

Establish a fresh configuration file for Nginx.

*$ sudo nano /etc/nginx/conf.d/joplin-server.conf*

It is important to incorporate the required information into this file.

The document must include the below mentioned information.

server {

listen 80;

listen [::]:80;

server\_name joplin1.example1.com;

error\_log /var/log/nginx/joplin-server.error;

location / {

proxy\_pass http://127.0.0.1: 8081;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

}

}

Save the document, and close it.

Examine the document's syntax and verify if there are any configuration problems with Nginx.

$ sudo nginx -t

To implement the adjustments, restart Nginx.

$ sudo service nginx restart

3. Security  
UFW comes pre-activated on Ubuntu installations. Following are the procedures to adjust the firewall settings to permit HTTP and HTTPS connections on the server.  
Enable and permit access for both HTTP and HTTPS on the designated port let say 80 and 443 respectively. Followed by the refreshing the firewall rules to implement the modifications,   
which can be achieved by executing the below command

$ sudo ufw allow 80/tcp

$ sudo ufw allow 443/tcp

$ sudo ufw reload

4. Set up access via HTTPS

Set up the Let's Encrypt Certbot client.

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

Get an SSL certificate for free. Change admin1@example1.com to your email address and joplin1.example1.com to your fully-qualified domain name.

$ sudo certbot -d joplin1.example1.com -m admin1@example1.com

Certbot will establish  a new certificate for your Nginx server, initiate a request for a new certificate, and arrange a renewal event. For additional details, please refer to our Certbot guide.

To make the modifications take effect, restart Nginx.

$ sudo service nginx restart

5. Access the Joplin Server

To assess the Joplin server, please open a web browser and go to your subdomain at: https://joplin1.example1.com. Log in using the default credentials listed below: Email: admin@localhost Password: admin. Once you are logged in, click on your username, Admin, located in the upper right corner. You can then update your Administrator profile by filling in the Full name, Email, and Password fields. It is recommended to select a strong and secure password to enhance the Joplin server's security, followed by clicking Update Profile. Next, click on Admin in the top navigation menu. Proceed to the Users section and select

Incorporate the user to facilitate the creation of a new non-administrative account intended for syncing with your Joplin Client.

6. Link the Joplin Server and Clients  
  
Go to the official website and install the Joplin client application.  
  
Start the Joplin client.   
  
The Preferences menu is chosen by Mac users.   
The Settings menu is chosen by Windows users.   
The Configuration menu is chosen by mobile users.  
  
Go to  synchronisation. then Click synchronisation target drop-down.  
  
Select the Joplin Server from the available options.  
  
In the Joplin Server URL section, type your fully qualified domain name.  
  
Enter the email address associated with your account in the Joplin Server email field.  
  
In the Joplin Server password section, type your password.  
  
Kindly enter your preferred timing in the Synchronization interval box.

Save the modifications made  
  
Apply is clicked by desktop users.

Check Synchronisation Configuration is tapped by mobile users.

7. Troubleshooting

Using curl, you may check the host port of Joplin Server from the command line.

$127.0.0.1:8080 $ curl

Below is a list of frequent problems you might encounter, accompanied by solutions. One such issue you may encounter is a Poor Gateway error.

502 Bad Gateway

Verify that the Joplin Server container is operational and actively listening on the designated port specified in your Docker Compose file. For illustration, this article references port 8081. Ensure that the Joplin Server is functioning within Docker and is responsive on the host port.

**Invalid origin on 127.0.0.1**

You might encounter an error message that states:

Invalid origin: <http://127.0.0.1:8080>

If this situation occurs, please investigate the Docker logs for any hidden issues that may be present.

$ sudo docker logs joplin-server

**Not a valid URL**

Furthermore, the Joplin client may display another error message, which is:

Error. Please check that URL, username, password, etc. are correct and that the sync target is accessible. The reported error was:

Not a valid URL: joplin1.example1.com/api/sessions

To address the above error, prepend https:// to the URLand double-check that your username and password are accurate.

**Invalid origin on fully-qualified domain name**

Another potential error you could face is:

Invalid origin: https://joplin1.example1.com

Examine the - APP\_BASE\_URL= environment variable within the Joplin Docker Compose file to ensure that the appropriate domain name is specified for accessing the application. The variable should appear as illustrated below. Ensure you replace joplin1.example1.com with your fully-qualified domain name.

- APP\_BASE\_URL=https://joplin1.example1.com

In summarize, you have effectively deployed Joplin Server using Docker on an Ubuntu system and established synchronization with the Joplin Client application Furthermore, you have the ability to establish numerous user accounts and keep them in sync with the server.

https://1text.com/plagiarismchecker/673f6605eb7eb