Setting up the Open Source GIS Stack

# Preparing the server:

## Security tools:

#### Unattended upgrades

This will automatically install only security fixes on a continual basis on your server.

sudo apt install unattended upgrades

#### ssh

Disable password authentication for SSH

sudo vim /etc/ssh/sshd\_config

Set this:

PasswordAuthentication no

Then do

sudo systemctl restart sshd.service

#### Crowdsec

<https://crowdsec.net/>

wget -qO - https://s3-eu-west-1.amazonaws.com/crowdsec.debian.pragmatic/crowdsec.asc |sudo apt-key add - && echo "deb https://s3-eu-west-1.amazonaws.com/crowdsec.debian.pragmatic/$(lsb\_release -cs) $(lsb\_release -cs) main" | sudo tee /etc/apt/sources.list.d/crowdsec.list > /dev/null;

sudo apt-get update

sudo apt-get install crowdsec

#### Fail2ban

sudo apt install fail2ban

<https://www.fail2ban.org/wiki/index.php/Main_Page>

#### Firewall

sudo ufw allow ssh

sudo ufw enable

sudo ufw status

Should show something like this:

Status: active

To Action From

-- ------ ----

22/tcp ALLOW Anywhere

22/tcp (v6) ALLOW Anywhere (v6)

We will open more ports as they are needed.

## Status monitoring

**bpytop** is a great console based dashboard for monitoring your server.

sudo snap install bpytop

## Docker

sudo apt install [docker.io](http://docker.io) docker-compose

### Git, rpl and Make

Needed for checking out our docker project and running the various make commands we provide.

sudo apt install git make rpl

# Deploying the server

Note for the unprivileged user throughout here, we use the user name ‘timlinux’ - you should substitute this with your own user.

## User Group

Add yourself to the user group of docker so you don't need to sudo docker commands.

sudo usermod -a -G docker timlinux

Then log out and in again to assume the upgraded permissions.

## Project Checkout

cd /home

sudo mkdir web

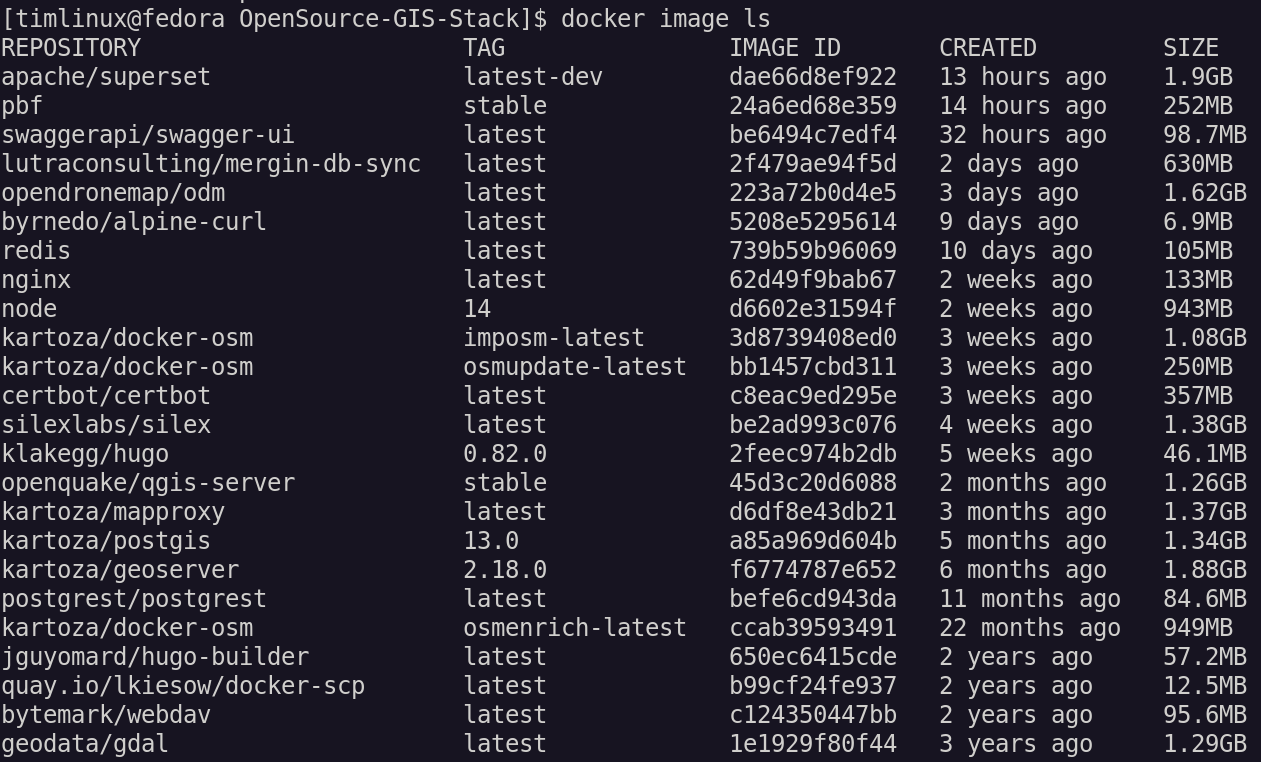
sudo chown [timlinux.timlinux](http://timlinux.timlinux) web

cd web

git clone https://github.com/kartoza/OpenSource-GIS-Stack

cd OpenSource-GIS-Stack

### Fetching Docker Images



### Configuration

Copy the .env boilerplate file and then adjust any settings in it as needed.

cp .env.example .env

Replace terms that should be unique for your installation:

rpl smallholding gis .env

rpl castelo.kartoza.com geoservices.govt.lc .env

rpl castelo.kartoza.com geoservices.govt.lc nginx\_conf/nginx.conf

### SCP File Drop

This is a container intended for users to upload files for publication on the server. It runs on port 2222 so we need to expose that through the firewall:

ufw allow 2222

sudo ufw allow 2222

You can add your public keys from the host e.g.

cat ~/.ssh/authorized\_keys > scp\_conf/gis\_projects

Or copy them in by other means. Each file you create in scp\_conf will be a user name. Each file should contain a list of public keys.

Starting the container:

docker-compose up -d scp

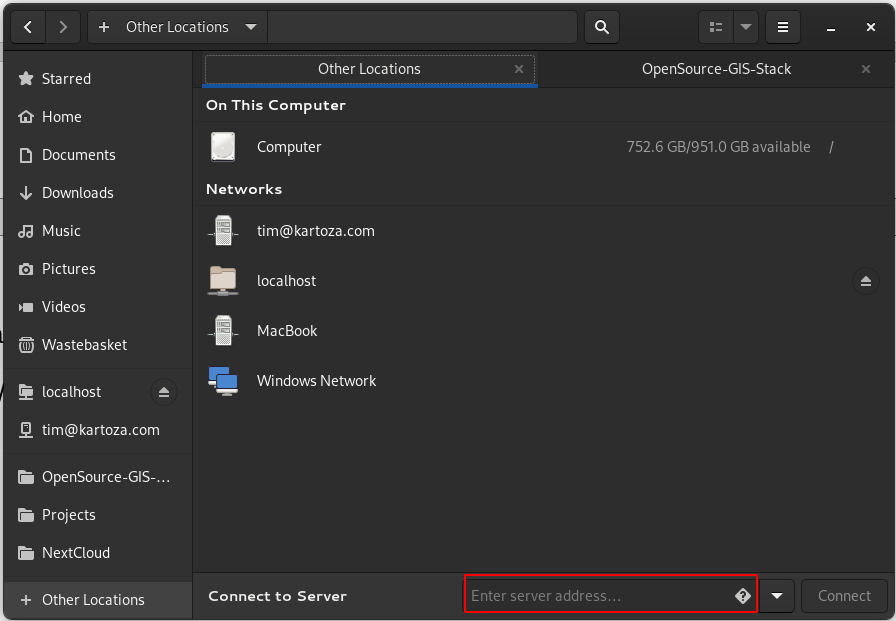
Example copying of data into the container:

scp -P 2222 sample-document.txt localhost:/data//gis\_projects/gis\_projects/gis\_projects

In Nautilus (file manager in Linux Gnome Desktop) you can test by connecting

sftp://localhost:2222/data/gis\_projects

into the red highlighted box below:



After that open a second window and you can drag and drop files too and from the folder.

The scp container has these pre-configured accounts and purposes:

* **geoserver\_data** - data to be made available from a file store in GeoServer
* **qgis\_data** - project folders for publishing QGIS projects via QGIS Server
* **hugo\_data** - markdown files for publishing to hugo static site generator