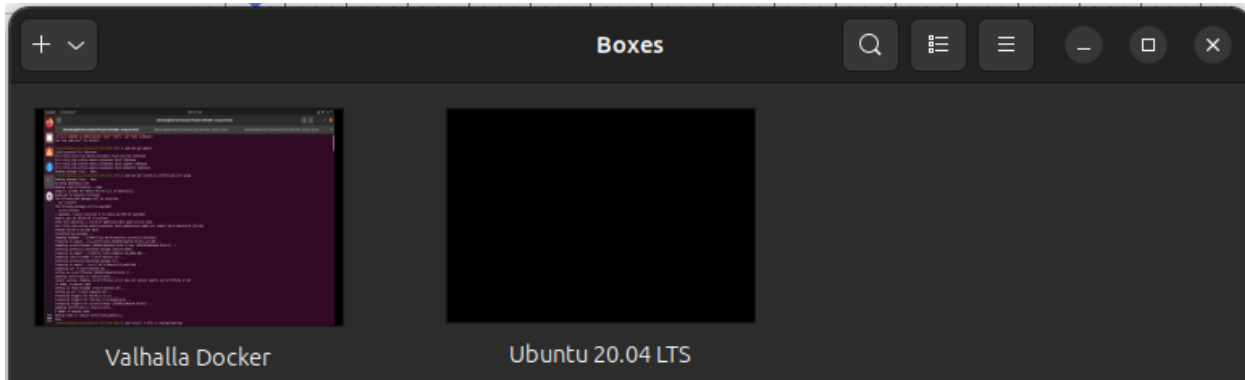


# Running Valhalla using Docker

I have the latest Ubuntu 23.04 installed on my laptop. The tutorials I found were for Ubuntu 20.04 and Valhalla 3.0.9. Therefore, I again created a virtual machine with the recommended Ubuntu 20.04 in Gnome Boxes.

```
vaishnavnegi@vaishnavnegi:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 23.10
Release:       23.10
Codename:      mantic
```

## Gnome Boxes



## Installing `docker` and `docker-compose`:

Since installing Docker is not the focus of our task here, I'll only list the steps I followed to successfully install Docker on my Virtual Machine. These can be found on the [official Docker documentation](https://docs.docker.com/engine/install/ubuntu/).

1. 

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo apt-get update
[sudo] password for lukevaine:
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:2 http://de.archive.ubuntu.com/ubuntu focal InRelease
Hit:3 http://de.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://de.archive.ubuntu.com/ubuntu focal-backports InRelease
```
2. 

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo apt-get install ca-certificates curl gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.2).
gnupg set to manually installed.
The following NEW packages will be installed:
  curl libcurl4
```
3. 

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo install -m 0755 -d /etc/apt/keyrings
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ curl -fsSL https://download.docker.com/linux/ubuntu
/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo chmod a+r /etc/apt/keyrings/docker.gpg
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ echo \
> "deb [arch=$(dpkg --print-architecture)] signed-by=/etc/apt/keyrings/docker.gpg https://downlo
ad.docker.com/linux/ubuntu \
> "$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \
> sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```
4. We need to install a specific version of Docker Engine for Ubuntu 20.04, so we start by listing the available versions in the repository:

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:24.0.7-1~ubuntu.20.04~focal
  Version table:
   5:24.0.7-1~ubuntu.20.04~focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:24.0.6-1~ubuntu.20.04~focal 500
      500 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages
   5:24.0.5-1~ubuntu.20.04~focal 500
```

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ VERSION_STRING=5:24.0.7-1~ubuntu.20.04~focal
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo apt-get install docker-ce=$VERSION_STRING
docker-ce-cli=$VERSION_STRING containerd.io docker-buildx-plugin docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras git git-man liberror-perl pigz slirp4netns
Suggested packages:
```

5.

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-11-14 16:31:29 CET; 53s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 11531 (dockerd)
      Tasks: 18
     Memory: 26.2M
    CGroup: /system.slice/docker.service
            └─11531 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

6.

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:88ec0acaa3ec199d3b7eaf73588f4518c25f9d34f58ce9a0df68429c5af48e8d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
```

7.

Next, we install `docker-compose`. The steps for this can be found on this [link](#).

1. The following command will download and save the executable file at `/usr/local/bin/docker-compose`, which will make this software globally accessible as `docker-compose`.

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo curl -L "https://github.com/docker/compose/releases
/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 12.1M 100 12.1M    0     0 4979k    0  0:00:02  0:00:02 --:--:-- 5548k
```

2. Now we give executable permissions so that the `docker-compose` command is executable:

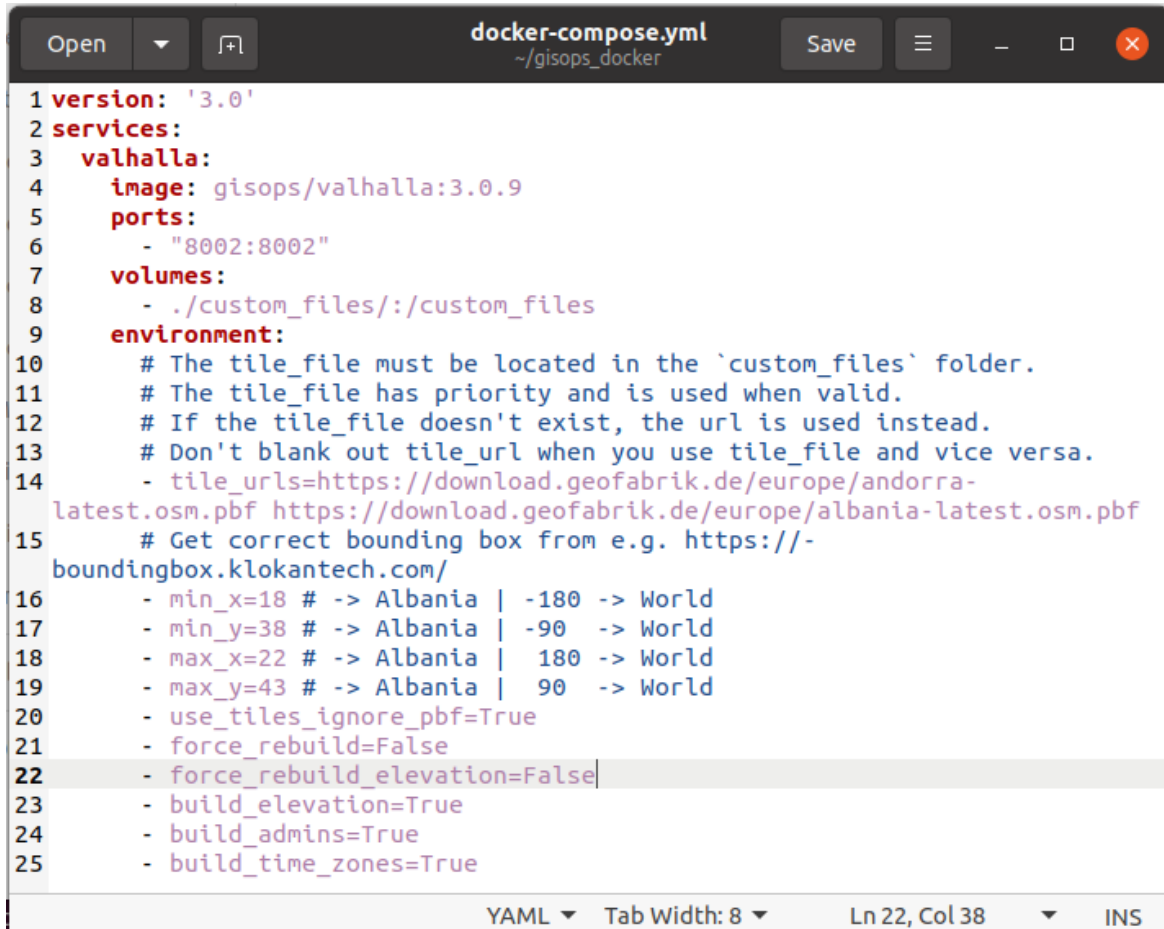
```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ sudo chmod +x /usr/local/bin/docker-compose
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

# Setup Valhalla on Ubuntu 20.04 with Docker

- We create a folder in our home directory to clone the needed docker files and create `docker-compose.yml` file required to run Valhalla:

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~$ mkdir ~/gisops_docker && cd $_
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~/gisops_docker$ nano docker-compose.yml
```

- The content of the `docker-compose.yml` are as follows:



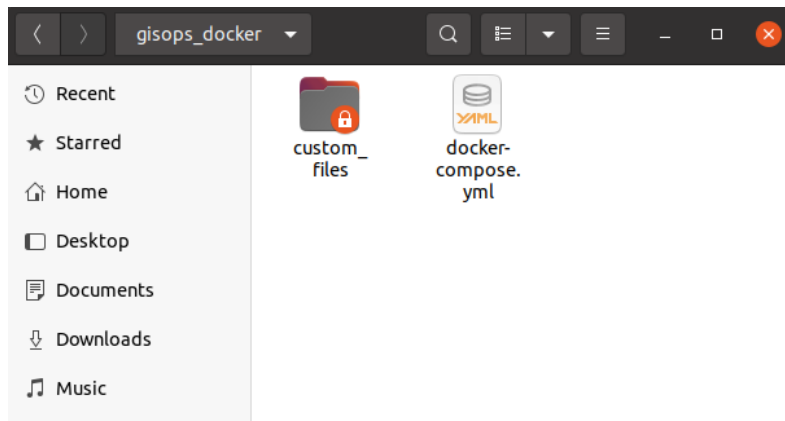
```
1 version: '3.0'
2 services:
3   valhalla:
4     image: gisops/valhalla:3.0.9
5     ports:
6       - "8002:8002"
7     volumes:
8       - ./custom_files:/custom_files
9     environment:
10      # The tile_file must be located in the `custom_files` folder.
11      # The tile_file has priority and is used when valid.
12      # If the tile_file doesn't exist, the url is used instead.
13      # Don't blank out tile_url when you use tile_file and vice versa.
14      - tile_urls=https://download.geofabrik.de/europe/andorra-
15        latest.osm.pbf https://download.geofabrik.de/europe/albania-latest.osm.pbf
16      # Get correct bounding box from e.g. https://-
17        boundingbox.klokantech.com/
18      - min_x=18 # -> Albania | -180 -> World
19      - min_y=38 # -> Albania | -90 -> World
20      - max_x=22 # -> Albania | 180 -> World
21      - max_y=43 # -> Albania | 90 -> World
22      - use_tiles_ignore_pbf=True
23      - force_rebuild=False
24      - force_rebuild_elevation=False
25      - build_elevation=True
26      - build_admins=True
27      - build_time_zones=True
```

- **volumes** is mounting a local folder called **custom\_files** into the docker container. This is the place where the Valhalla container shares its data and also the place where changes to the running instance can be made.
- **tile\_urls** will be downloaded by Valhalla and build if their file hashes don't match with the existing build tiles. It's possible to add as many urls as desired with a space delimiter.
- **min\_\*/max\_\*** defines the bounding box so Valhalla can download the correct elevation tiles. This is only required if **build\_elevation** is set to True.
- **use\_tiles\_ignore\_pbf** will let Valhalla know that it should prioritize tiles that were already built and not overwrite it.
- **force\_rebuild** will force Valhalla to rebuild the tiles in all cases except for **force\_rebuild\_elevation**.
- **force\_rebuild\_elevation** will always skip rebuilding the elevation data if False. That way downloading and processing huge amounts of elevation data can be omitted.
- **build\_\*** will let Valhalla know what to build in detail. All of them are optional for a functioning Valhalla instance.

Now docker-compose will build Valhalla as follows:

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~/gisops_docker$ sudo docker-compose up --build
Pulling valhalla (gisops/valhalla:3.0.9)...
3.0.9: Pulling from gisops/valhalla
5bed26d33875: Pull complete
f11b29a9c730: Pull complete
930bda195c84: Pull complete
78bf9a5ad49e: Pull complete
6237fff93ffb: Pull complete
0f1b9025811c: Pull complete
49bdf9fd54e: Pull complete
```

- Docker will pull the Valhalla image 3.0.9 from [GIS • OPS Dockerhub](https://github.com/OSM-Valhalla/valhalla) as specified in the .yaml file and build it with the OSM files of Albania and Andorra.
- A new folder named custom\_files is created. It has all the files Valhalla required to properly run or rebuild any of its data. This folder can be copied, compressed, or moved to a different system and as long as the same docker-compose.yml is used, Valhalla will be able to immediately start with our data.



- The Valhalla service will start after the build is complete.

```
valhalla_1 | Found config file. Starting valhalla service!
valhalla_1 | 2023/11/14 16:15:30.722509 [INFO] Tile extract successfully loaded with tile count: 242
valhalla_1 | 2023/11/14 16:15:30.755262 [WARN] Corrupt elevation data: /custom_files/elevation_data/N40
valhalla_1 | 2023/11/14 16:15:30.755612 [WARN] Corrupt elevation data: /custom_files/elevation_data/N40
valhalla_1 | 2023/11/14 16:15:30.755962 [WARN] Corrupt elevation data: /custom_files/elevation_data/N40
```

- While the service is running, we will open another terminal tab and write in the following commands to test the service:

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~/gisops_docker$ sudo docker ps -a
[sudo] password for lukevaine:
CONTAINER ID   IMAGE                                COMMAND                  NAMES
PORTS
0e8ff30add7f   gisops/valhalla:3.0.9               "/bin/sh -c ${SCRIPT... 21 minutes ago   Up 21 minutes
0.0.0.0:8002->8002/tcp, :::8002->8002/tcp   gisops_docker_valhalla_1
8b1f384acbb5   nginx:alpine                        "/docker-entrypoint.... 44 minutes ago   Up 44 minutes
0.0.0.0:8000->80/tcp, :::8000->80/tcp       compose-demo_web_1
779f73d1c5a5   hello-world                          "/hello"                 57 minutes ago   Exited (0) 57 minutes a
go
keene_mcnulty
```

```
lukevaine@lukevaine-Standard-PC-Q35-ICH9-2009:~/gisops_docker$ curl http://localhost:8002/route --data
'{"locations":[{"lat":41.318818,"lon":19.461336}, {"lat":41.321001,"lon":19.459598}], "costing": "auto", "d
irections_options":{"units":"miles"}}' | jq '.'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
Dload  Upload   Total     Spent    Left     Speed
100 1743 100 1604 100 139 195k 17375 --:--:-- --:--:-- --:--:-- 212k
{
  "trip": {
    "language": "en-US",
    "status": 0,
    "units": "miles",
    "status_message": "Found route between points",
    "legs": [
      {
        "shape": "yx{xA_lybd@eEzCsVvQwWfSaSdPoShOsVzPiRjN",
        "summary": "r
```