# Starting from Scratch

Notes

* All paths in this section are relative to ~/openrouteservice repo root folder
* Any commands in the CLI that start with docker should be run within the docker folder.

Step-by-step instructions

1. Install and set up Docker (google how to install for your system)—you MUST have Docker running before you can proceed with next steps.
2. Clone openroutesource GitHub repository - <https://github.com/GIScience/openrouteservice.git>
3. Per <https://giscience.github.io/openrouteservice/installation/Running-with-Docker.html>:
   1. Inside of docker folder, run in command line interface (CLI)
      1. mkdir conf elevation\_cache graphs logs/ors logs/tomcat to create necessary folders (use backslashes if in Windows)
      2. docker-compose up -d --build,which will run docker/docker-compose.yml (“the YML”)and build a Docker image for openrouteservice per the parameters in the YML.
4. Confirm everything installed as expected by running:
   1. Run docker container ls, and ensure the ors-app container is running
   2. Run docker images, and ensure the openrouteservice/openrouteservice image appears
5. Test that everything works by entering <http://localhost:8080/ors/v2/isochrones> into a web browser. It should return a JSON string saying "error":{"code":3001 followed by some other stuff. This is good. It means the image was created and the ors-app container is successfully running and there is an API end point that scripts can use.
   1. If it returns a “page not found” or similar message, try running docker container restart

# Configuring to your needs after creating docker image

By default, the ORS backend uses:

* OSM PBF file of Heidelberg Germany
* Assumes any routing or graphing (graphs are representations of the OSM network that are navigable and mode-specific) are for driving in a car.

## Edit travel modes assumed

The default installation assumes driving as the mode of travel. To set this:

1. Open docker/conf/ors-config.json
2. Specify the modes you want routing done for under services.routing.profiles.active
   1. Acceptable modes to include are "car", "hgv", "bike-regular", "bike-mountain", "bike-road", "bike-electric", "walking", "hiking" or "wheelchair"
3. Delete the docker/graphs folder
4. Restart the container by running docker container restart
5. Confirm that the correct mode(s) are shown inside docker/graphs, they should be listed as folder names (e.g. “car”, “bike-regular”, etc.)

More on ors-config.json - <https://giscience.github.io/openrouteservice/installation/Configuration.html>

Troubleshooting:

* If the docker/graphs folder is empty (should have folders with names of modes specified in ors-config.json):
  + WAIT 10-15mins, because sometimes it takes time for graphs to build. But while you’re waiting, confirm:
    - The value you entered “routing”:”profiles”:”active” was spelled correctly and is one of the values listed in the [config documentation](https://giscience.github.io/openrouteservice/installation/Configuration.html#orsservicesroutingprofiles)

## Changing the OSM file used

* Follow [start-from-scratch](#_Starting_from_Scratch) instructions
* CAREFULLY follow [instructions](https://giscience.github.io/openrouteservice/installation/Advanced-Docker-Setup.html#running-an-already-built-container). A Hi-level recap of these steps:
  1. Stop and remove the image you created through the from-scratch directions by running docker-compose down
  2. In the docker-compose YML:
     1. Change build: parameters to specify which config file and which OSM file you wanna use.
  3. Uncomment and set the ./data mount volume mapping relationship to be <path to desired OSM file>:<path that ORS uses
  4. Run docker-compose up.
     1. If you don’t specify -d, it will not run as a background process (cmd window will stay open). In theory you should be able to specify -d, but for some reason the HTTP no-site-found error will come up if you do specify -d.
        1. Possibility is to also add the --build flag?
     2. NOTE – it takes a while for the service to fully start (maybe 10 minutes), so once it’s started think twice before shutting it down.

### Notes

* Running docker-compose down (in theory) clears everything: removes the image and any associated containers—better way than stopping containers and doing docker image rm

# Using the ORS Backend in Scripts

## General Steps

1. Ensure Docker is started and running\*
2. Make sure you have [created a docker image](#_Starting_from_Scratch), indicated your desired travel modes via [ors-config](#_Edit_travel_modes), and indicated the OSM file you want to use via [docker-compose.yml](#_Changing_the_OSM)
3. Open a CMD line and navigate to the ORS Docker folder (containing the docker-compose.yml file)
4. Run docker container start ors-app
   1. NOTE – if you get an Error response from daemon… error, try restarting Docker then running the command again
5. Run any applicable scripts
6. When done, run docker container stop ors-app
   1. Note – if the container is running in the cmd window, you can stop it by opening a new CMD window and running the stop command

\*While you can leave Docker running all the time, it consumes several GB of memory while running.

## Scripts Currently Calling on the Cloud API

Any script currently using the cloud API can easily be switched to use the local API by the following replacement:

Replace:

<https://api.openrouteservice.org/v2/isochrones>

With:

<http://localhost:8080/ors/v2/isochrones>

# More information

[From “Getting startied with ORS via docker”](https://ask.openrouteservice.org/t/getting-started-with-ors-via-docker-and-localhost/1415/11)