# Install using Docker

Docker is the easiest way to start a OpenHIM instance. Follow these instructions to get up and running using Docker:

<https://openhim.readthedocs.io/en/latest/getting-started.html#installation-using-docker>

Before running “docker-compose up -d”, ensure that the file ‘default.json’ has host set to the public IP address of the server:

{

"version": "1.10.0",

"minimumCoreVersion": "3.4.0",

"protocol": "https",

"host": "<server\_public\_ip>",

"port": 8080,

"title": "Admin Console",

"footerTitle": "OpenHIM Administration Console",

"footerPoweredBy": "<a href='http://openhim.org/' target='\_blank'>Powered by OpenHIM</a>",

"loginBanner": "",

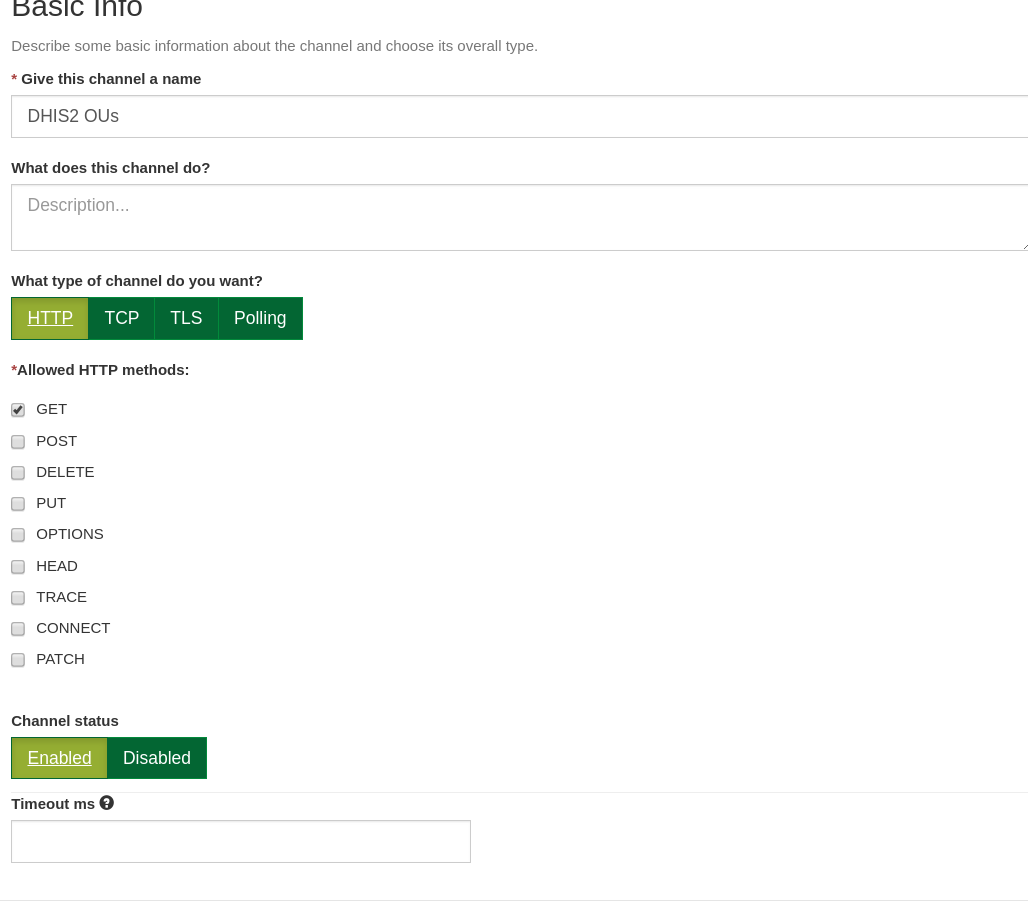
"mediatorLastHeartbeatWarningSeconds": 60,

"mediatorLastHeartbeatDangerSeconds": 120

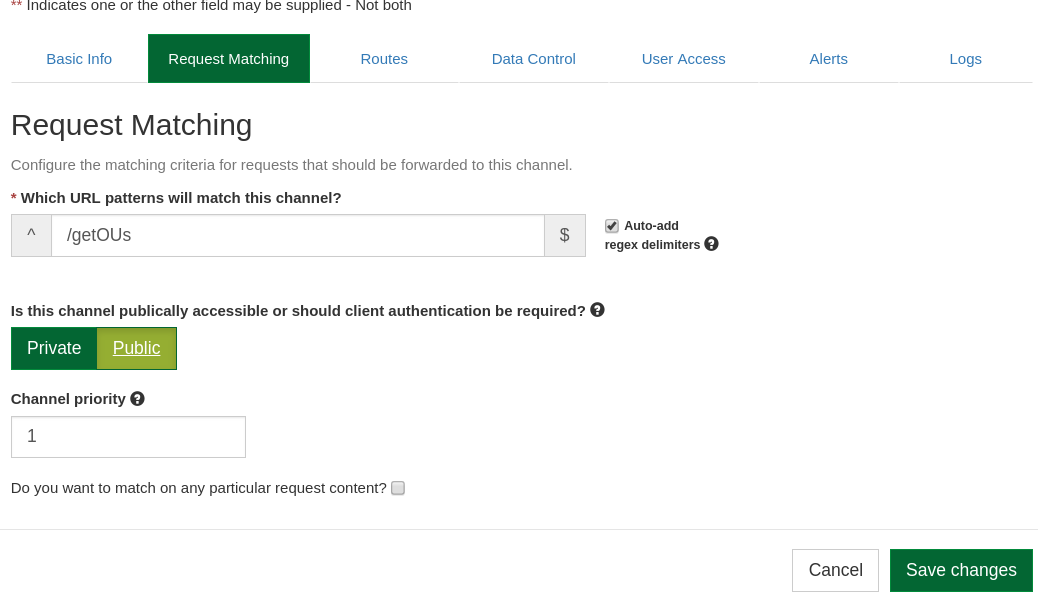
}

Open ports 5000 (for channels), 8080 (OpenHIM heartbeat and API) and 9000 (OpenHIM Web Interface).

Let’s replicate the same OU service that we previously implemented in Zato using OpenHIM. Start by creating a channel and only allow GET requests:

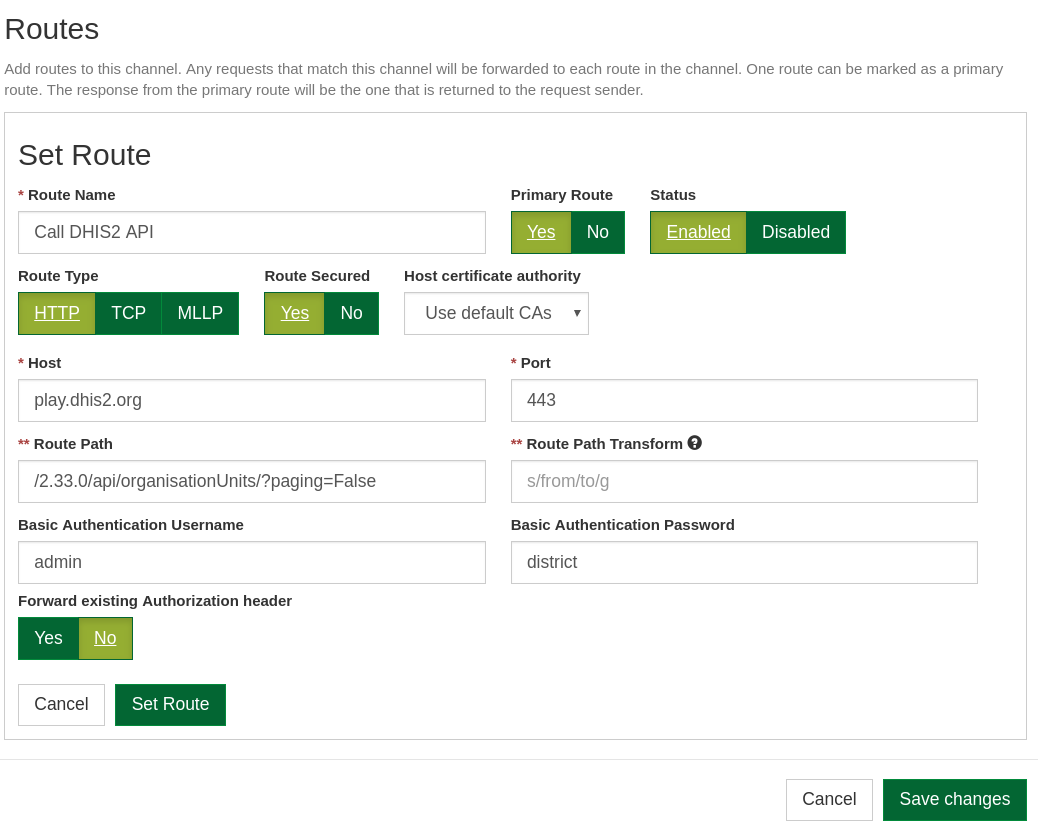


On “Request Matching” tab, set the following:



This is the URL clients will use to get OUs.

In the Routes tab, set the following details:



We have two options to match a URL to: either to a mediator service or another route like the one as shown above. The good thing about routes like this is we can write additional logic in any language or framework and have OpenHIM forward the request to it.

Save the channel and try running the following to test whether it’s working or not:

$ curl localhost:5000/getOUs

You should get a response like this:



The channel is working now.

## Note: Body Retention

OpenHIM, by default, stores requests and responses. However responses like OU lists can be quite large. We can configure a particular OpenHIM channel to not store responses by configuring the following settings:

