Get a VPS, domain name and trusted certs

To be able to self-host a matrix server, you need a Linux machine (preferrably Ubuntu 22.04). You can get a VPS server from any provider you like. Optionally, you can also get yourself a domain name to use with your chat app.
If you did get a domain name, make sure to generate and install trusted ssl certs with Let's Encrypt.

To generate the cert, first install certbot:

```
sudo apt update
sudo apt install certbot python3-certbot-nginx
```

Next, generate the cert for your domain

```
sudo certbot certonly --nginx -d yourdomain.com
```

This will generate your certificate to ***/etc/letsencrypt/live/yourdomain.com/fullchain.pem*** and the private key to ***/etc/letsencrypt/live/yourdomain.com/privkey.pem***

Install Synapse and Element as Docker containers

We will run synapse and element as individual docker containers for better security. Make sure you have docker installed and use this docker-compose.yaml file that defines the services to run (synapse and element), configures a private docker network, and assigns an IP in the private network to each service.

```
services:
 synapse:
   image: matrixdotorg/synapse:latest
   restart: unless-stopped
   environment:
     - SYNAPSE_CONFIG_PATH=/data/homeserver.yaml
   volumes:
     - ./synapse:/data
   networks:
     custom_network:
       ipv4_address: 10.10.10.2
 element:
   image: vectorim/element-web:latest
   restart: unless-stopped
   volumes:
     - ./element-config.json:/app/config.json
   networks:
     custom_network:
```

```
      ipv4_address: 10.10.10.3
networks:
 custom_network:
   ipam:
     driver: default
     config:
       - subnet: "10.10.10.0/24"
```

First, generate your homeserver.yaml file with this command

```
docker compose run --rm -e SYNAPSE_SERVER_NAME=yourdomain.com
-e SYNAPSE_REPORT_STATS=yes synapse generate
```

Make sure to replace example.com to your domain name. If you don't have a domain name, you can use your IP address here.

This will generate a new folder called **synapse** and inside it, you will find your **homeserver.yaml** file.

Open it, and add the following lines at the end:

```
## Repactha and registration config
recaptcha_public_key: "####################"
recaptcha_private_key: "####################"


## Enable registration
enable_registration: true
enable_registration_captcha: true



## Disable federation if you want
# federation_domain_whitelist: []
```

Replace the **recaptcha_public_key** with your Google ReCaptcha v2 site key and **recaptcha_private_key** to your Google ReCaptcha v2 secret key. You can create your keys from here: https://www.google.com/recaptcha/admin/create
More details on how to do it is documented here:
https://github.com/matrix-org/synapse/blob/develop/docs/CAPTCHA_SETUP.md

Now, you can start the services in your docker-compose.yaml file:

```
docker compose up
```

The synapse server and element client are now running inside containers and they are part of a private docker network. In order to make them accessible from the Internet, you need to configure an nginx reverse proxy.

Create a proxy configuration file at /etc/nginx/conf.d/proxy.conf

```
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    listen 8448 ssl http2 default_server;
    listen [::]:8448 ssl http2 default_server;

    # SSL cert paths
    ssl_certificate
/etc/letsencrypt/live/yourdomain.com/fullchain.pem;  # Full
chain - CHANGE THIS TO YOUR APPROPRIATE CERT PATH
    ssl_certificate_key
/etc/letsencrypt/live/yourdomain.com/privkey.pem; # Private
key - CHANGE THIS TO YOUR APPROPRIATE KEY PATH

    server_name yourdomain.com; # CHANGE THIS TO YOUR DOMAIN

    location ~ ^(/_matrix|/_synapse/client) {
        proxy_pass http://10.10.10.2:8008; # Container IP for
Synapse
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $host;

        client_max_body_size 50M;

        proxy_http_version 1.1;
    }

    location / {
        proxy_pass http://10.10.10.3:80; # Container IP for
Element Chat
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $host;

        client_max_body_size 50M;
```

```
        proxy_http_version 1.1;
    }
}

server {
    listen 80;  # Listen on port 80 (HTTP)
    server_name yourdomain.com www.yourdomain.com; # CHANGE
THIS TO YOUR DOMAIN NAME

    # Redirect all HTTP requests to HTTPS
    return 301 https://$host$request_uri;
}
```

—

Make sure to replace **ssl_certificate** and **ssl_certificate_key** to the paths of your certficiate and key generated by certbot earlier.

Also, change **server_name** to your domain name.

You can then go ahead and reload nginx for this proxy to turn active.

```
sudo systemctl reload nginx
```

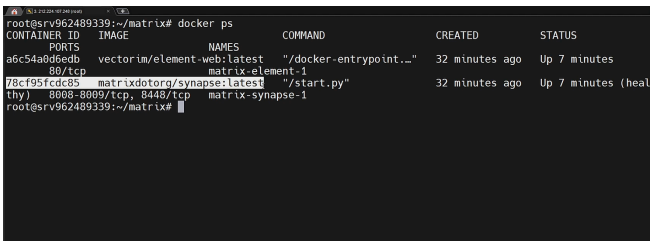
And now, you can go to **yourdomain.com** and you will be able to access the element web client which is communicating with the synapse server in the backend. You can then register yourself and start using your private chat app!

Administration

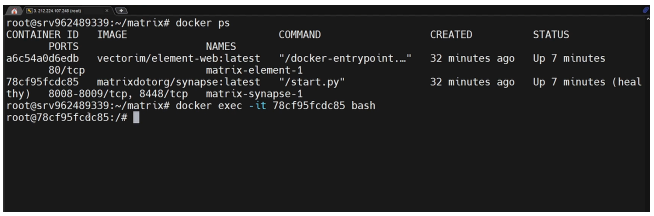
Since you self-hosted your Matrix homeserver, you should also be able to do administration using the Admin API.
You can use this synapse admin web console to do the administration stuff without having to manually interact with the admin api.

But before that, you need to have administrator account on your homeserver. To do this, you need to exec into your synapse docker container.

First, run docker ps to see the running containers and note down the container ID of the synapse server.



Next, exec into it and run bash.
`docker exec -it <container-id> bash`



Once you are inside the container, run the following command:

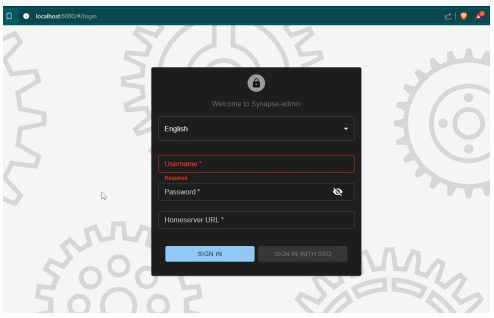
`register_new_matrix_user -c /data/homeserver.yaml http://localhost:8008`

This will prompt you for the username, password and whether you want to make this use an admin - select "yes" and it will create an admin account for you.

Now that the admin account is created, you can go ahead and run the synapse admin web console.

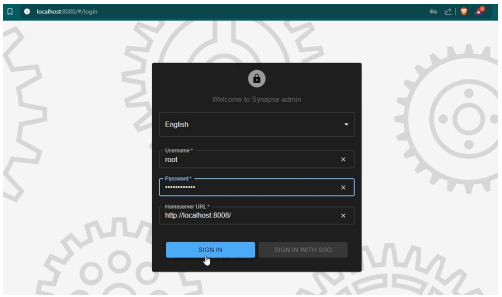
You can run it with docker with this command:
`docker run -p 8080:80 awesometechnologies/synapse-admin`

You don't have to run this on your VPS, instead, you can do it on your local machine as well. This will make the admin console available to use at http://localhost:8080



You need to set your homeserver URL here so that the console will be able to communicate with the admin API. But since we are not exposing our Synapse admin API to the Internet with nginx reverse proxy, you need to create an SSH tunnel from your machine to the private docker network where the Synapse server is actually running. You can do that very easily with this simple command:
`ssh -N -L 127.0.0.1:8008:10.10.10.2:8008 root@<YOUR-VPS-IP>`
This will create a tunnel and redirect all the traffic from your 127.0.0.1 port 8008 to 10.10.10.2 port 8008 in your VPS's docker network where the Synapse server is running.

So, now you can set your homeserver URL to http://127.0.0.1:8008 and then input your admin account credentials to login to the admin console.



You can now manage your homeserver - like manage, add, delete users, create rooms, manage federation, etc.