Matrix Synapse Miscellaneous

Different things, sometimes advanced and some things that just did not fit in the regular guide.

Show the public rooms on the server

https://example.com/_matrix/client/r0/publicRooms

Add an SRV Record

To use the domain `matrix.example.com` and still just use `@user:example.com` instead of `@user:matrix.example.com` we need to set up an SRV record. At first an A (possibly AAAA) record is needed for `matrix.example.com`.

After the A record is set up, create a SRV Record that looks like

```
_matrix._tcp.<yourdomain.com> <ttl> IN SRV 10 0 <port> <synapse.server.name>
```

So, for example use

```
_matrix._tcp.example.com. 3600 IN SRV 10 0 8448 synapse.example.com.
```

and when creating the Synapse server use `example.com` for the servername.

Stefan

So, when I'm using the domain matrix.example.com and want to use exmpale.com for the registration and so on, i should use SRV. If i'm using example.com and making the redirect with nginx, there is no need for a SRV record?

kythyria

The SRV record is how other servers find your server (and putting matrix federation behind a reverse proxy is a bit fragile)

Mathijs the relevant url is where the federated servers can connect to you

kythyria

It's so your server_name (what's in MXIDs) doesn't have to be the hostname of the machine synapse is running on.

Example 1

`example.com` points to the physical machine where synapse is installed and Port 8448 is used.

Example 2

`example.com` points to the phsycial machine where synapse is installes and port 8448 can not be used.

SRV needed.

Example 3

Fix Me! `example.com` point to the physical machine with the URL `example1.com`

SRV needed?

.well-known section

Mathijs
it's a little early, but you could also add a section about .well-known which just means you have nginx serve a json file on example.com/.well-known/matrix/client I did it for apache, but it's probably fairly easy for nginx as well

Mathijs
https://matrix.org/docs/spec/client_server/r0.4.0.html#server-discovery if you like reading spec :)

Proper explanation

.well-known will be checked before SRV gets checked.

Coturn

Coturn is a turn server and it is used for 1:1 voip calls through the client (example: riot).

```
apt install coturn
```

Edit Config files

opening ports

Optional Adminshell

If you forgot to write `sudo` and don't want to rewrite the whole command, just type `sudo !!` to execute the last command with sudo rights.

There is another way but you should **not** work like that all the time because its not secure. There is a reason you have to write `sudo` for a lot of commands. For an initial server setup you may have to type a lot of commands, to avoid typing `sudo *` all the time, it is possible to type in `sudo su`. This opens a admin-shell and you should be able to work without typing `sudo` in. Keep in mind, `exit` after step 8 and don't use `sudo su` when you just have to type in some commands.

Synpase maintenance tools

https://github.com/matrix-org/synapse/wiki/Synapse-database-maintenance-tools

Installing Bots

Matrix Synapse currently does not have a concept of bots, a bot is just a normal user. Usally there is a config and a `.py` file, download the files, unpack them and run the `.py` file. Running and downloading things via pip is not recommended.

Install python3

```
sudo apt install python3
```

then install pip

```
sudo apt install python3-pip
```

then matrix-bot-api from pyp

```
pip3 install matrix-bot-api
```

and finally fill out the config and start it

```
python3 pollbot.py
```

How calls work

Info about WebRTC: https://www.pkc.io/blog/untangling-the-webrtc-flow/
more or less like this? https://i.imgur.com/dWAwi7f.png with jitsi.riot.im instead of scalar.vector.im

more or less, but you can also host your own jitsi server if you want, jitsi is still FOSS and self-hostable

also, it's a bit nitpicky, but if you use turn, you may want to add a turn-server on the left side, because it's not synapse that does the webrtc relay

jitsi is its own thing, riot just allows integrating jitsi in riot

note that you can self-host jitsi, but if you want riot to use your selfhosted jitsi by default when opening conference calls you'll also want to host an integrations server (ie dimension)

everybody assumes it works like this, but it doesn't: assuming both HSs have turn, you actually want red lines via homeserver 1, homeserver 2 and then one going through both ie. the media may go through one or both turn servers (or none, as in your green line)

@dave:matrix.org not sure I quite understand the second diagram in the first. media won't always go through both turn servers the key to thinking about turn, I find, is that the TURN server is a thing that essentially pretends to be your client, but somewhere else on the internet and it tunnels the traffic back to you so you have a point of presence with your own internet connection, and then a second point of presence in your turn server once its opened a channel for you and you then present both of those options to ther other side as ways to talk to you equally you can also use your turn server as a route to send packets out to the internet to talk to the other party