

Please visit website: <http://cxyroad.com>

局域网聊天软件 matrix

=====

窝有 3 只 Android 手机 (3 号手机, 6 号手机, 9 号手机),
2 台 ArchLinux PC (4 号 PC, 6 号 PC), 1 台 Fedora CoreOS 服务器 (5 号).
(作为穷人, 窝使用的基本上是老旧的二手设备, 比如 5 年前的手机,
9 年前的笔记本, 10 年前的古老 e5v3 主机, 都比较便宜.)

窝经常需要 (想) 从一台设备发消息/文件等到另一台设备.
这个功能虽然使用 QQ / 等也能实现, 但是有很多问题:

* (1) **需要注册很多账号**.

比如一个 QQ 号只能同时在一只手机上登录, 有多只手机互相发消息, 就需要多个账号.

这很可能意味着需要有多多个手机号, 很是麻烦, 花钱也多.

* (2) **通信效率不高**.

这些设备都在同一个局域网, 连接在同一个 wifi 路由器之下.

但是, 发消息却需要通过上述聊天软件在公网的服务器, 绕一圈回来.
直接在局域网内部, 走近路它不香嘛?

* (3) **无法在断网时使用**.

上述软件由于依赖公网的服务器, 如果宽带接入中断, 就无法使用了.

综上, 需要搭建一套在本地局域网使用的聊天软件.

相关文章: 《使用多用户增强服务器的安全性》 juejin.cn/post/738354...

目录

--

* 1 服务端的安装 (synapse)

* 2 客户端软件

+ 2.1 手机客户端 (fluffychat)

+ 2.2 PC 客户端 (fractal)

+ 2.3 更多可选的客户端 (cinny, neochat, nheko, moment, element)

- * 3 聊天测试
- + 3.1 客户端登录
- + 3.2 创建群组 (聊天室)
- + 3.3 发送消息
- * 4 总结与展望

1 服务端的安装 (synapse)

`matrix` 是一个开源的通信协议,也就是一种标准规范.
matrix 使用 C/S (客户端/服务器) 架构,也就是分为中心服务器和客户端.
其中服务器 (homeserver) 集中存储账号聊天数据,
客户端软件运行在 PC / 手机上,用户使用客户端收发消息.

有多种具体的服务端软件可以选择,此处采用的是比较成熟的一个,功能比较完善.

参考资料: [element-hq.github.io/synapse/latest/](http://cxyroad.com/"https://element-hq.github.io/synapse/latest/")

* (1) 制作容器镜像.

相关文章: 《构建 deno/fresh 的 docker 镜像》 juejin.cn/post/737951...

使用类似的方法来制作镜像,如下:

...

```
> cat Dockerfile
FROM quay.io/jitesoft/alpine:latest

RUN sed -i 's/ftp.halifax.rwth-aachen.de/mirrors.sjtug.sjtu.edu.cn/g'
/etc/apk/repositories
RUN apk update && apk upgrade && apk add synapse curl icu-data-full
&& apk cache clean
```

```
EXPOSE 8008/tcp 8009/tcp 8448/tcp
```

```
ENTRYPOINT ["/usr/bin/synapse_homeserver"]
```

```
HEALTHCHECK --start-period=5s --interval=15s --timeout=5s \
  CMD curl -fSs http://localhost:8008/health || exit 1
```

```
#CMD ["/bin/ash"]
```

```
...
```

构建命令:

```
...
```

```
docker build -t synapse .
```

```
...
```

制作的镜像:

```
...
```

```
> docker images
```

REPOSITORY	SIZE	TAG	IMAGE ID	
synapse		latest	e01a809a9821	About a
minute ago	256MB			
quay.io/jitesoft/alpine		latest	1bd690c0f25c	6 days
ago	7.82MB			

```
...
```

导出镜像:

```
...
```

```
> docker save synapse | zstd > synapse.tar.zst
```

```
> ls -lh synapse.tar.zst
```

```
-rw-r--r-- 1 s2 s2 78M 6月26日 16:34 synapse.tar.zst
```

```
...
```

* (2) 服务端配置文件.

(Fedora CoreOS 服务器详见文章 《使用多用户增强服务器的安全性》)

相关配置文件如下 (仅供参考):

```
...  
#fc-test@MiWiFi-RA74-srv:~$ cat /mnt/data/d1/fc-test/srv/synapse/homeserver.yaml  
# Configuration file for Synapse.  
server_name: "synapse.fc-server.test"  
pid_file: /homeserver.pid  
listeners:  
  - port: 8008  
    tls: false  
    type: http  
    x_forwarded: true  
    bind_addresses: ['::', '0.0.0.0']  
    resources:  
      - names: [client, federation]  
        compress: false  
database:  
  name: sqlite3  
  args:  
    database: /var/homeserver.db  
media_store_path: /var/media_store  
log_config: "/var/log_config.yaml"  
  
# 随机字符串  
registration_shared_secret: "随机秘密, 请修改此处"  
macaroon_secret_key: "随机秘密, 请修改此处"  
form_secret: "随机秘密, 请修改此处"  
  
report_stats: true  
signing_key_path: "/etc/synapse/my.domain.name.signing.key"  
trusted_key_servers:  
  - server_name: "localhost"  
  
#fc-test@MiWiFi-RA74-srv:~$ cat /mnt/data/d1/fc-test/srv/synapse/log_config.yaml  
# Log configuration for Synapse.  
version: 1  
  
formatters:  
  precise:  
    format: '%(asctime)s - %(name)s - %(lineno)d - %(levelname)s -  
%(request)s - %(message)s'
```

```
handlers:
  file:
    class: logging.handlers.TimedRotatingFileHandler
    formatter: precise
    filename: /var/homeserver.log
    when: midnight
    backupCount: 3
    encoding: utf8

  buffer:
    class: synapse.logging.handlers.PeriodicallyFlushingMemoryHandler
    target: file

    capacity: 10
    flushLevel: 30
    period: 5

  console:
    class: logging.StreamHandler
    formatter: precise

loggers:
  synapse.storage.SQL:
    level: INFO

root:
  level: INFO
  #level: DEBUG

  handlers: [buffer]

disable_existing_loggers: false
```

...

****警告: 此处的配置并没有充分考虑服务器的安全性, 仅适用于在局域网内部使用 !!****
****免责声明: 请勿将服务器暴露在公网, 可能出现严重的安全问题, 后果自负 !!****
* (3) 运行服务端容器.

(3.1) 加载容器镜像:

...

```
fc-test@MiWiFi-RA74-srv:~$ ls -lh
total 155M
```

```

-rw-r--r--. 1 fc-test fc-test 77M Jun 23 08:16 my-app.tar.zst
-rw-r--r--. 1 fc-test fc-test 78M Jun 26 08:44 synapse.tar.zst
fc-test@MiWiFi-RA74-srv:~$ podman load < synapse.tar.zst
Getting image source signatures
Copying blob 9aa2e4323f1d done |
Copying blob feed612d9b64 done |
Copying blob 299020072df1 done |
Copying blob fc99c43dcd02 done |
Copying config e01a809a98 done |
Writing manifest to image destination
Loaded image: docker.io/library/synapse:latest
fc-test@MiWiFi-RA74-srv:~$ podman images
REPOSITORY                                TAG          IMAGE ID      CREATED
SIZE
docker.io/library/synapse                 latest      e01a809a9821  13
minutes ago 266 MB
registry.fedoraproject.org/fedora-toolbox 40          fe913ee7ac45  27
hours ago 2.19 GB
docker.io/library/my-app                   latest      83173f90cca5  13 days
ago 238 MB
fc-test@MiWiFi-RA74-srv:~$

```

...

(3.2) 配置文件:

...

```

fc-test@MiWiFi-RA74-srv:~$ cat
~/.config/containers/systemd/synapse.container
[Unit]
Description=synapse server
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=5s
StartLimitBurst=1

[Container]
Image=synapse
PublishPort=8008:8008
Pull=never

Exec= -c /var/homeserver.yaml
Volume=/mnt/data/d1/fc-test/srv/synapse/:/var:z

[Service]

```

Restart=always

[Install]

WantedBy=default.target

...

(3.3) 重新载入配置文件:

...

```
fc-test@MiWiFi-RA74-srv:~$ systemctl --user daemon-reload
fc-test@MiWiFi-RA74-srv:~$ systemctl --user status synapse
○ synapse.service – synapse server
  Loaded: loaded (/var/home/fc-
test/.config/containers/systemd/synapse.container; generated)
  Drop-In: /usr/lib/systemd/user/service.d
           └─10-timeout-abort.conf
  Active: inactive (dead)
fc-test@MiWiFi-RA74-srv:~$
```

...

(3.4) 运行容器:

...

```
fc-test@MiWiFi-RA74-srv:~$ systemctl --user start synapse
```

...

* (4) 创建用户.

(4.1) 查看运行的容器:

...

```
fc-test@MiWiFi-RA74-srv:~$ podman ps
CONTAINER ID  IMAGE                                COMMAND
CREATED      STATUS    PORTS                NAMES
bf7e6be170fa  docker.io/library/my-app:latest     /usr/bin/deno run... 7
hours ago   Up 7 hours  0.0.0.0:8000->8000/tcp  systemd-my-app
17e3fb784136  docker.io/library/synapse:latest    -c /var/homeserve... 2
minutes ago Up 2 minutes 0.0.0.0:8008->8008/tcp  systemd-synapse
```

...

(4.2) 在容器中运行命令:

...

```
fc-test@MiWiFi-RA74-srv:~$ podman exec -it systemd-synapse
/bin/ash
/ # register_new_matrix_user -c /var/homeserver.yaml
New user localpart [root]: p3
Password:
Confirm password:
Make admin [no]: yes
Sending registration request...
Success!
/ #
```

...

使用 `register_new_matrix_user` 命令创建新用户, 输入用户名和密码.

2 客户端软件

支持 matrix 协议的客户端软件也有很多可供选择.

2.1 手机客户端 (fluffychat)

这个可能是最可爱的客户端软件, 支持 Android 手机, 可爱无敌 !!

网址: [fluffychat.im/](http://cxyroad.com/ "https://fluffychat.im/")

2.2 PC 客户端 (fractal)

* (1) fractal 是一个具有 GNOME 界面风格的 PC 客户端.
可以从 flathub 安装:

...

flatpak install flathub org.gnome.Fractal

...

[flathub.org/apps/org.gn...](http://cxyroad.com/
"https://flathub.org/apps/org.gnome.Fractal")

[gitlab.gnome.org/World/fract...](http://cxyroad.com/
"https://gitlab.gnome.org/World/fractal")

* (2) fluffychat 也有 PC 版, 但是目前体验稍微差一些.

[flathub.org/apps/im.flu...](http://cxyroad.com/
"https://flathub.org/apps/im.fluffychat.Fluffychat")

[fluffychat.im/](http://cxyroad.com/ "https://fluffychat.im/")

2.3 更多可选的客户端 (cinny, neochat, nheko, moment, element)

* (1) `cinny`

[flathub.org/apps/in.cin...](http://cxyroad.com/
"https://flathub.org/apps/in.cinny.Cinny")

[cinny.in/](http://cxyroad.com/ "https://cinny.in/")

* (2) `neochat`

[flathub.org/apps/org.kd...](http://cxyroad.com/
"https://flathub.org/apps/org.kde.neochat")

[apps.kde.org/de/neochat/](http://cxyroad.com/
"https://apps.kde.org/de/neochat/")

* (3) `nheko`

[flathub.org/apps/im.nhe...](http://cxyroad.com/
"https://flathub.org/apps/im.nheko.Nheko")

* (4) `moment`

[flathub.org/apps/xyz.mx...](http://cxyroad.com/
"https://flathub.org/apps/xyz.mx_moment.moment")

* (5) `element`

[element.io/download](http://cxyroad.com/
"https://element.io/download")

* (6) `matrix-commander`

这个是命令行界面的, 酷!

[github.com/8go/matrix-...](http://cxyroad.com/
"https://github.com/8go/matrix-commander")

* 更多推荐列表: [itsfoss.com/best-matrix...](http://cxyroad.com/
"https://itsfoss.com/best-matrix-clients/")

以上只是列举了一部分, 真的有很多的兼容软件可以选择!

3 聊天测试

下面简单演示一下客户端软件的使用, 以 fluffychat 举例.

3.1 客户端登录

首先需要登录账号. 打开 fluffychat 手机应用, 如图:

![31-login-1.jpg](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/b7d0dfdfb9f941859c3c155a8219ded0~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=190801&e=jpg&b=fbf6fd)

服务器地址输入 `http://synapse.fc-server.test:8008` 并确认.
其中 `synapse.fc-server.test` 为本地局域网服务器的 IP 地址.
如果地址输入错误, 就连不上服务器, 请仔细检查是否有微小的错误.

![31-login-2.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1add84f7cf47453fafb1e828e4341016~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=318977&e=jpg&b=faf4fc)

连接服务器成功, 点击 `使用密码登录` 按钮.

![31-login-3.jpg](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/0108a67b1d884b338e2ac82c879fe89c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=406323&e=jpg&b=f6f0f9)

![31-login-4.jpg](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/6adf7c2ce3eb4a34961f6a0312e42c61~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=382861&e=jpg&b=f6f0f9)

输入用户名/密码, 进行登录.

![31-login-5.jpg](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/88bb478ac86942088d7ef10abbb59891~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=87730&e=jpg&b=fbf6fe)

登录成功界面.

![31-n-1.jpg](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/432ce35eac7e404a90ea71b16f481984~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=154774&e=jpg&b=f9f4fc)

点击右上角头像, 然后点击 `设置` 菜单项, 可以进行简单的设置.

![31-n-2.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/b0235296c9ee477faac56e7261903aa3~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=188949&e=jpg&b=fdf8ff)

比如可以修改自己的名字.

![31-n-3.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/98fbfcfc264a4338969a35497ae2a329~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=186621&e=jpg&b=fdf8ff)

3.2 创建群组 (聊天室)

![32-r-1.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4f64f7daed1a401ca590df0e8566cb84~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=145050&e=jpg&b=f8f3fa)

进入创建群组界面.

![32-r-2.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/46ce763581aa4543989733149af2bc73~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=224550&e=jpg&b=f5e)

ff7)

输入名称和描述, 进行创建.

![32-r-3.jpg](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/2168bdd746f2479480b51f345c2736aa~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=258562&e=jpg&b=fbf6fd)

创建成功.

![32-r-4.jpg](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/31281a24d32d4f3cb1a46f048ef7db87~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=270690&e=jpg&b=fbf6fd)

3.3 发送消息

重复上述步骤: 首先在服务器上, 为每个设备创建账号.
然后在每个设备上登录对应账号, 并加入群组.

然后画风就变成这样了:

![33-t-1.jpg](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/c18f05cfcf54444a9f3548c79149c03f~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1080&h=2340&s=315408&e=jpg&b=f8f3fb)

这是在手机上使用 fluffychat 客户端.

![33-t-2.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1a57a7b767944be6bf03781d6ec5d6a6~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=1092&h=1158&s=104776&e=png&a=1&b=232323)

这是在 PC 上使用 fractal 客户端.

众喵集齐, 喵呜 ~~

4 总结与展望

matrix 协议为 C/S 架构, 分为中心服务器和客户端.
服务端软件和客户端软件都有很多种可供选择, 本文使用其中 2 种进行举例.

服务端软件以容器的方式运行, 需要自己制作容器镜像, 部署镜像并编写所需配置文件.
服务端启动之后, 可以创建新的账号.

客户端软件的安装就很简单了, 输入服务器地址, 以及用户名/密码进行登录.
登录之后可以创建/加入群组 (聊天室), 然后就可以愉快的发送消息啦 ~

通过局域网聊天软件, 几只手机和 PC 终于可以齐聚一堂, 共同玩耍了 !

synapse 服务端目前使用 sqlite 数据库, 但是这个数据库仅供测试, 读写性能很差.
后续应该升级成使用 postgresql 数据库.

本文使用 CC-BY-SA 4.0 许可发布.

原文链接: <https://juejin.cn/post/7384652811554619443>