
Box

Setup for my personal (pet) server.

Felix Pojtinger

2021-12-11

Contents

1	Introduction	3
1.1	Contributing	3
1.2	License	3
2	Debian	4
3	IPv6	4
4	DNS	4
5	SSH	5
6	firewalld	5
7	APT	6
8	Podman	6
9	Traefik	6
10	Cockpit	10
11	Gitea	10
12	Dex	11
13	liwasc	12
14	bofied	13

1 Introduction

1.1 Contributing

Found an error or have a suggestion? Please open an issue on GitHub (github.com/pojntfx/box):



Figure 1: QR-Code to the source code on GitHub

1.2 License

This document and included source code is Free Culture/Free Software.



Figure 2: Badge of the AGPL-3.0 license

Box (c) 2021 Felix Pojtinger

SPDX-License-Identifier: AGPL-3.0

2 Debian

```
1 sudo umount /dev/mmcblk0{,p1,p0}
2 curl -L 'https://raspi.debian.net/tested/20210823_raspi_3_bullseye.img.xz' | xzcat >/tmp/debian.img
3 sudo dd if=/tmp/debian.img of=/dev/mmcblk0 bs=4M status=progress
4 sync
5
6 sudo mkdir -p /mnt/raspi-boot
7 sudo mount /dev/mmcblk0p1 /mnt/raspi-boot
8 {
9     echo "root_pw=$(openssl rand -base64 12)"
10    echo "root_authorized_key=$(cat ~/.ssh/id_rsa.pub)"
11    echo "hostname=jeans-box"
12 } >>/mnt/raspi-boot/sysconf.txt
13 sudo umount /dev/mmcblk0{,p1,p0}
```

3 IPv6

```
1 ssh root@jeans-box
2 tee /etc/sysctl.d/privacy.conf <<'EOT'
3 net.ipv6.conf.all.use_tempaddr=2
4 EOT
5 sysctl -p
6
7 tee /etc/network/interfaces.d/eth0 <<'EOT'
8 auto eth0
9 iface eth0 inet dhcp
10
11 iface eth0 inet6 static
12     address 2001:7c7:2121:8d00::3
13     autoconf 1
14     accept_ra 2
15 EOT
16 systemctl restart networking
17
18 tee /etc/resolv.conf <<'EOT'
19 nameserver 2606:4700:4700::1111
20 nameserver 2606:4700:4700::1001
21 EOT
22 chatr +i /etc/resolv.conf
23 sed -i /etc/hosts -e 's/\tlocalhost/\tlocalhost jeans-box/g'
```

4 DNS

1	jeans-box	10800	IN	AAAA	2001:7c7:2121:8d00::3
2	*.jeans-box	10800	IN	AAAA	2001:7c7:2121:8d00::3

5 SSH

```
1 ssh root@jeans-box.example.com
2 apt update
3 apt install -y sudo curl openssh-server locales
4 systemctl enable --now ssh
5
6 echo "LC_ALL=en_US.UTF-8" | tee -a /etc/environment
7 echo "en_US.UTF-8 UTF-8" | tee /etc/locale.gen
8 echo "LANG=en_US.UTF-8" | tee /etc/locale.conf
9 locale-gen en_US.UTF-8
10
11 adduser jean
12 su jean -c "mkdir -m 700 -p ~/.ssh && curl 'https://github.com/jean.
    keys' | tee -a ~/.ssh/authorized_keys && chmod 600 ~/.ssh/
    authorized_keys"
13 usermod -aG sudo jean
14
15 echo 'PermitRootLogin no' | tee /etc/ssh/ssh_config.d/no-root.conf
16
17 passwd -d root
18 passwd -l root
19 chsh -s /sbin/nologin
20 rm ~/.ssh/authorized_keys
21
22 systemctl restart ssh
```

6 firewalld

```
1 ssh jean@jeans-box.example.com
2 sudo apt update
3 sudo apt install -y firewalld
4 sudo systemctl enable --now firewalld
5 sudo firewall-cmd --zone=public --add-interface=eth0 --permanent
6 sudo firewall-cmd --permanent --add-service=mdns
7 sudo firewall-cmd --permanent --add-service=llmnr
8 sudo firewall-cmd --reload
```

7 APT

```
1 ssh jean@jeans-box.example.com
2 sudo apt update
3 sudo apt install -y unattended-upgrades
4
5 sudo tee /etc/apt/apt.conf.d/50unattended-upgrades <<'EOT'
6 Unattended-Upgrade::Origins-Pattern {
7     "origin=*";
8 }
9 Unattended-Upgrade::Automatic-Reboot "true";
10 Unattended-Upgrade::Automatic-Reboot-Time "02:00";
11 EOT
12 sudo systemctl enable --now unattended-upgrades
13 sudo unattended-upgrades --debug
```

8 Podman

```
1 ssh jean@jeans-box.example.com
2 echo 'deb https://download.opensuse.org/repositories/devel:/kubic:/
   libcontainers:/stable/Debian_11/ /' | sudo tee /etc/apt/sources.list
   .d/libcontainers.list
3 curl -L "https://download.opensuse.org/repositories/devel:/kubic:/
   libcontainers:/stable/Debian_11/Release.key" | sudo apt-key add -
4 sudo apt update
5 sudo apt upgrade -y # Prevent conflicts with eventual prior Podman
   install from Debian repos
6 sudo apt install -t Debian_11 -y podman
7 echo 'unqualified-search-registries=["docker.io"]' | sudo tee /etc/
   containers/registries.conf.d/docker.conf
8 sudo systemctl unmask podman-auto-update.service
9 sudo systemctl unmask podman-auto-update.timer
10 sudo systemctl enable --now podman-auto-update.timer
11 sudo systemctl enable --now podman-restart
```

9 Traefik

```
1 sudo mkdir -p /etc/traefik
2 sudo tee /etc/traefik/traefik.yaml<<'EOT'
3 entryPoints:
4   web:
5     address: ":80"
6
7   websecure:
```

```
8     address: ":443"
9
10    sshalt:
11      address: ":2222"
12
13    websecurealt:
14      address: ":8443"
15
16    providers:
17      file:
18        filename: /etc/traefik/services.yaml
19        watch: true
20
21    api:
22      dashboard: true
23
24    certificatesResolvers:
25      letsencrypt:
26        acme:
27          email: jean@example.com
28          storage: /var/lib/traefik/acme.json
29          httpChallenge:
30            entryPoint: web
31
32    log:
33      level: INFO
34    EOT
35
36    sudo tee /etc/traefik/services.yaml<<'EOT'
37    tcp:
38      routers:
39        ssh:
40          entryPoints:
41            - websecurealt
42          rule: HostSNI(`*`)
43          service: ssh
44        giteaSSH:
45          entryPoints:
46            - sshalt
47          rule: HostSNI(`*`)
48          service: giteaSSH
49        sshOverTLS:
50          entryPoints:
51            - websecure
52          rule: HostSNI(`ssh.jeans-box.example.com`)
53          service: ssh
54        tls:
55          certResolver: letsencrypt
56          domains:
57            - main: ssh.jeans-box.example.com
58    services:
```

```
59     ssh:
60         loadBalancer:
61             servers:
62                 - address: localhost:22
63     giteaSSH:
64         loadBalancer:
65             servers:
66                 - address: localhost:3022
67
68 http:
69     routers:
70         dashboard:
71             rule: Host(`traefik.jeans-box.example.com`)
72             tls:
73                 certResolver: letsencrypt
74                 domains:
75                     - main: traefik.jeans-box.example.com
76             service: api@internal
77             entryPoints:
78                 - websecure
79             middlewares:
80                 - dashboard
81     cockpit:
82         rule: Host(`cockpit.jeans-box.example.com`)
83         tls:
84             certResolver: letsencrypt
85             domains:
86                 - main: cockpit.jeans-box.example.com
87         service: cockpit
88         entryPoints:
89             - websecure
90     gitea:
91         rule: Host(`gitea.jeans-box.example.com`)
92         tls:
93             certResolver: letsencrypt
94             domains:
95                 - main: gitea.jeans-box.example.com
96         service: gitea
97         entryPoints:
98             - websecure
99     dex:
100         rule: Host(`dex.jeans-box.example.com`)
101         tls:
102             certResolver: letsencrypt
103             domains:
104                 - main: dex.jeans-box.example.com
105         service: dex
106         entryPoints:
107             - websecure
108     liwasc:
109         rule: Host(`liwasc.jeans-box.example.com`)
```



```
110     tls:
111         certResolver: letsencrypt
112         domains:
113             - main: liwasc.jeans-box.example.com
114     service: liwasc
115     entryPoints:
116         - websecure
117     bofied:
118         rule: Host(`bofied.jeans-box.example.com`)
119         tls:
120             certResolver: letsencrypt
121             domains:
122                 - main: bofied.jeans-box.example.com
123     service: bofied
124     entryPoints:
125         - websecure
126
127     middlewares:
128         dashboard:
129             basicauth:
130                 users:
131                     - "jean:$apr1$dYdt8Zrl$TsEfzaedPGyjdrDk8EfRN." # htpasswd -nb
132                               htpasswd -nb jean asdf
133
134     services:
135         cockpit:
136             loadBalancer:
137                 serversTransport: cockpit
138                 servers:
139                     - url: https://localhost:9090
140         gitea:
141             loadBalancer:
142                 servers:
143                     - url: http://localhost:3000
144         dex:
145             loadBalancer:
146                 servers:
147                     - url: http://localhost:5556
148         liwasc:
149             loadBalancer:
150                 servers:
151                     - url: http://localhost:15124
152         bofied:
153             loadBalancer:
154                 servers:
155                     - url: http://localhost:15256
156
157     serversTransports:
158         cockpit:
159             insecureSkipVerify: true
160 EOT
```

```
160
161 sudo podman run -d --restart=always --label "io.containers.autoupdate=
    image" --net=host -v /var/lib/traefik:/var/lib/traefik -v /etc/
    traefik:/etc/traefik --name traefik traefik
162 sudo podman generate systemd --new traefik | sudo tee /lib/systemd/
    system/traefik.service
163 sudo systemctl daemon-reload
164 sudo systemctl enable --now traefik
165
166 sudo firewall-cmd --permanent --add-service=http
167 sudo firewall-cmd --permanent --add-service=https
168 sudo firewall-cmd --permanent --add-port=8443/tcp
169 sudo firewall-cmd --reload
170
171 curl -Lu jean:asdf https://traefik.jeans-box.example.com/ # Test the
    Traefik dashboard
172 ssh -p 8443 jean@jeans-box.example.com # Test SSH over TCP
173 ssh -o ProxyCommand="openssl s_client -connect ssh.jeans-box.example.
    com:443 -quiet" jean # Test SSH over TLS
```

10 Cockpit

```
1 echo 'deb http://deb.debian.org/debian bullseye-backports main' | sudo
    tee /etc/apt/sources.list.d/backports.list
2 sudo apt update
3 sudo apt install -t bullseye-backports -y cockpit cockpit-podman
    cockpit-pcp
4
5 curl https://cockpit.jeans-box.example.com/ # Test Cockpit
```

11 Gitea

```
1 sudo mkdir -p /var/lib/gitea
2 sudo podman run -d --restart=always --label "io.containers.autoupdate=
    image" --net slirp4netns:allow_host_loopback=true,enable_ipv6=true -
    p 3000:3000 -p 3022:22 -v /var/lib/gitea:/data -v /etc/timezone:/
    etc/timezone:ro -v /etc/localtime:/etc/localtime:ro -e 'USER_UID
    =1000' -e 'USER_GID=1000' --name gitea gitea/gitea
3 sudo podman generate systemd --new gitea | sudo tee /lib/systemd/system
    /gitea.service
4 sudo systemctl daemon-reload
5 sudo systemctl enable --now gitea
6 sudo firewall-cmd --permanent --add-port=2222/tcp
7 sudo firewall-cmd --reload
8
```

```
9 curl https://gitea.jeans-box.example.com/ # Test Cockpit
```

Now visit <https://gitea.jeans-box.example.com/> and run the Wizard; use the following values:

- SSH Server Domain: `gitea.jeans-box.example.com`
- SSH Server Port: `2222`
- Gitea Base URL: `https://gitea.jeans-box.example.com/`
- Use your email SMTP server in [Email Settings](#), enable [Email Notifications](#) and [Require Email Confirmation to Register](#)
- Under [Server and Third-Party Service Settings](#), enable [Disable Self-Registration](#) (if you want to prevent others from using Gitea)
- Under [Administrator Account Settings](#), create your admin account

Note that the installation might take a while (about 1 minute)

12 Dex

First, setup Gitea by visiting <https://gitea.jeans-box.example.com/user/settings/applications> and adding a new OAuth2 application with Application Name `Dex` and Redirect URI `https://dex.jeans-box.example.com/callback`. Note the client ID and client secret; we'll need them in the following.

```
1 sudo mkdir -p /etc/dex
2 sudo mkdir -p /var/lib/dex
3 sudo touch /var/lib/dex/dex.db
4 sudo chown -R 1001:1001 /var/lib/dex/
5 sudo tee /etc/dex/config.yaml<<'EOT'
6 issuer: https://dex.jeans-box.example.com
7
8 storage:
9   type: sqlite3
10  config:
11    file: /var/dex/dex.db
12
13 web:
14   http: 0.0.0.0:5556
15   allowedOrigins: ['*']
16
17 staticClients:
18   - id: liwasc
19     redirectURIs:
20       - https://pojntfx.github.io/liwasc/
21     name: "liwasc"
22     public: true
23   - id: bofied
```

```

24     redirectURIs:
25         - https://pojntfx.github.io/bofied/
26     name: "bofied"
27     public: true
28
29     connectors:
30         - type: gitea
31           id: gitea
32           name: Gitea
33           config:
34             clientID: yourclientidfromgiteahere
35             clientSecret: yourclientsecretfromgiteahere
36             redirectURI: https://dex.jeans-box.example.com/callback
37             baseURL: https://gitea.jeans-box.example.com
38 EOT
39 sudo podman run -d --restart=always --label "io.containers.autoupdate=
    image" --net slirp4netns:allow_host_loopback=true,enable_ipv6=true -
    p 5556:5556 -v /var/lib/dex:/var/dex -v /etc/dex:/etc/dex --name dex
    ghcr.io/dexidp/dex dex serve /etc/dex/config.yaml
40 sudo podman generate systemd --new dex | sudo tee /lib/systemd/system/
    dex.service
41 sudo systemctl daemon-reload
42 sudo systemctl enable --now dex

```

You can test it out by visiting <https://pojntfx.github.io/liwasc/> and trying to log in using the following credentials:

- Backend URL: `ws://example.com/` (we'll set this later; this is just to try out the login)
- OIDC Issuer: `https://dex.jeans-box.example.com`
- OIDC Client ID: `liwasc`
- OIDC Redirect URL: `https://pojntfx.github.io/liwasc/`

And authorization prompt from Gitea and Dex should show up, after which liwasc's home page should load (showing an error like `Failed to construct 'WebSocket': An insecure WebSocket connection may not be initiated from a page loaded over HTTPS.`).

13 liwasc

```

1 sudo mkdir -p /var/lib/liwasc
2 sudo podman run -d --restart=always --label "io.containers.autoupdate=
    image" --net host --cap-add NET_RAW --ulimit nofile=16384:16384 -v /
    var/lib/liwasc:/root/.local/share/liwasc -e
    LIWASC_BACKEND_OIDCISSUER=https://dex.jeans-box.example.com -e
    LIWASC_BACKEND_OIDCLIENTID=liwasc -e LIWASC_BACKEND_DEVICENAME=eth0
    -e LIWASC_BACKEND_PERIODICSCANCRONEXPRESSION='0 0 * * *' --name
    liwasc pojntfx/liwasc-backend

```

```
3 sudo podman generate systemd --new liwasc | sudo tee /lib/systemd/
  system/liwasc.service
4 sudo systemctl daemon-reload
5 sudo systemctl enable --now liwasc
```

Now visit <https://pojntfx.github.io/liwasc/> as we did before and use `wss://liwasc.jeans-box.example.com/` as the backend URL (note the trailing slash!).

14 bofied

```
1 sudo mkdir -p /var/lib/bofied
2 sudo podman run -d --restart=always --label "io.containers.autoupdate=
  image" --net host --cap-add NET_BIND_SERVICE -v /var/lib/bofied:/
  root/.local/share/bofied -e BOFIED_BACKEND_OIDCISSUER=https://dex.
  jeans-box.example.com -e BOFIED_BACKEND_OIDCLIENTID=bofied -e
  BOFIED_BACKEND_ADVERTISEDIP=100.64.154.249 --name bofied pojntfx/
  bofied-backend
3 sudo podman generate systemd --new bofied | sudo tee /lib/systemd/
  system/bofied.service
4 sudo systemctl daemon-reload
5 sudo systemctl enable --now bofied
6 sudo firewall-cmd --permanent --add-port=67/udp
7 sudo firewall-cmd --permanent --add-port=69/udp
8 sudo firewall-cmd --permanent --add-port=4011/udp
9 sudo firewall-cmd --reload
```

Now visit <https://pojntfx.github.io/bofied/> and login using the following credentials:

- Backend URL: `https://bofied.jeans-box.example.com/`
- OIDC Issuer: `https://dex.jeans-box.example.com`
- OIDC Client ID: `bofied`
- OIDC Redirect URL: `https://pojntfx.github.io/bofied/`