Box

Setup for my personal (pet) server.

Felix Pojtinger

2021-12-11

Contents

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

2 Debian

```
sudo umount /dev/mmcblk0{,p1,p0}
curl -L 'https://raspi.debian.net/tested/20210823_raspi_3_bullseye.img.xz' | xzcat >/tmp/debian.img
sudo dd if=/tmp/debian.img of=/dev/mmcblk0 bs=4M status=progress
sync

sudo mkdir -p /mnt/raspi-boot
sudo mount /dev/mmcblk0p1 /mnt/raspi-boot
{
    echo "root_pw=$(openssl rand -base64 12)"
    echo "root_authorized_key=$(cat ~/.ssh/id_rsa.pub)"
    echo "hostname=jeans-box"
} >>/mnt/raspi-boot/sysconf.txt
sudo umount /dev/mmcblk0{,p1,p0}
```

3 IPv6

```
ssh root@jeans-box
tee /etc/sysctl.d/privacy.conf <<'EOT'
net.ipv6.conf.all.use_tempaddr=2
EOT
sysctl -p

tee /etc/network/interfaces.d/eth0 <<'EOT'
auto eth0
iface eth0 inet dhcp

iface eth0 inet6 static
    address 2001:7c7:2121:8d00::3
    autoconf 1
    accept_ra 2
EOT
systemctl restart networking

tee /etc/resolv.conf <<'EOT'
nameserver 2606:4700:4700::1111
nameserver 2606:4700:4700::1001
EOT
chattr +i /etc/resolv.conf
sed -i /etc/hosts -e 's/\tlocalhost/\tlocalhost jeans-box/g'
```

4 DNS

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

5 SSH

```
ssh root@jeans-box.example.com
apt update
apt install -y sudo curl openssh-server locales
systemctl enable --now ssh

echo "LC_ALL=en_US.UTF-8" | tee -a /etc/environment
echo "en_US.UTF-8 UTF-8" | tee /etc/locale.gen
echo "LANG=en_US.UTF-8" | tee /etc/locale.conf
locale-gen en_US.UTF-8

adduser jean
su jean -c "mkdir -m 700 -p ~/.ssh && curl 'https://github.com/jean.keys' | tee -a ~/.ssh/authorized_keys && chmod 600 ~/.ssh/authorized_keys"
usermod -aG sudo jean

echo 'PermitRootLogin no' | tee /etc/ssh/ssh_config.d/no-root.conf

passwd -d root
passwd -l root
chsh -s /sbin/nologin
rm ~/.ssh/authorized_keys

systemctl restart ssh
```

6 firewalld

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

7 APT

```
ssh jean@jeans-box.example.com
sudo apt update
sudo apt install -y unattended-upgrades

sudo tee /etc/apt/apt.conf.d/50unattended-upgrades <<'EOT'
Unattended-Upgrade::Origins-Pattern {
  "origin=*";
}
Unattended-Upgrade::Automatic-Reboot "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";
EOT
sudo systemctl enable --now unattended-upgrades
sudo unattended-upgrades --debug
```

8 Podman

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

9 Traefik

```
sudo mkdir -p /etc/traefik
sudo tee /etc/traefik/traefik.yaml<<'EOT'
entryPoints:
  web:
    address: ":80"

  websecure:
```

```
    address: ":443"

  sshalt:
    address: ":2222"

  websecurealt:
    address: ":8443"

providers:
  file:
    filename: /etc/traefik/services.yaml
    watch: true

api:
  dashboard: true

certificatesResolvers:
  letsencrypt:
    acme:
      email: jean@example.com
      storage: /var/lib/traefik/acme.json
      httpChallenge:
        entryPoint: web

log:
  level: INFO
EOT

sudo tee /etc/traefik/services.yaml<<'EOT'
tcp:
  routers:
    ssh:
      entryPoints:
        - websecurealt
      rule: HostSNI(`*`)
      service: ssh
    giteaSSH:
      entryPoints:
        - sshalt
      rule: HostSNI(`*`)
      service: giteaSSH
    sshOverTLS:
      entryPoints:
        - websecure
      rule: HostSNI(`ssh.jeans-box.example.com`)
      service: ssh
      tls:
        certResolver: letsencrypt
        domains:
          - main: ssh.jeans-box.example.com
  services:
```

```
    ssh:
      loadBalancer:
        servers:
          - address: localhost:22
    giteaSSH:
      loadBalancer:
        servers:
          - address: localhost:3022

http:
  routers:
    dashboard:
      rule: Host(`traefik.jeans-box.example.com`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: traefik.jeans-box.example.com
      service: api@internal
      entryPoints:
        - websecure
      middlewares:
        - dashboard
    cockpit:
      rule: Host(`cockpit.jeans-box.example.com`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: cockpit.jeans-box.example.com
      service: cockpit
      entryPoints:
        - websecure
    gitea:
      rule: Host(`gitea.jeans-box.example.com`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: gitea.jeans-box.example.com
      service: gitea
      entryPoints:
        - websecure
    dex:
      rule: Host(`dex.jeans-box.example.com`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: dex.jeans-box.example.com
      service: dex
      entryPoints:
        - websecure
    liwasc:
      rule: Host(`liwasc.jeans-box.example.com`)
```

```
      tls:
        certResolver: letsencrypt
        domains:
          - main: liwasc.jeans-box.example.com
      service: liwasc
      entryPoints:
        - websecure
    bofied:
      rule: Host(`bofied.jeans-box.example.com`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: bofied.jeans-box.example.com
      service: bofied
      entryPoints:
        - websecure

  middlewares:
    dashboard:
      basicauth:
        users:
          - "jean:$apr1$dYdt8Zrl$TsEfzaedPGyjdrDk8EfRN." # htpasswd -nb
              htpasswd -nb jean asdf

  services:
    cockpit:
      loadBalancer:
        serversTransport: cockpit
        servers:
          - url: https://localhost:9090
    gitea:
      loadBalancer:
        servers:
          - url: http://localhost:3000
    dex:
      loadBalancer:
        servers:
          - url: http://localhost:5556
    liwasc:
      loadBalancer:
        servers:
          - url: http://localhost:15124
    bofied:
      loadBalancer:
        servers:
          - url: http://localhost:15256

  serversTransports:
    cockpit:
      insecureSkipVerify: true
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
```

```
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.

```
sudo mkdir -p /etc/dex
sudo mkdir -p /var/lib/dex
sudo touch /var/lib/dex/dex.db
sudo chown -R 1001:1001 /var/lib/dex/
sudo tee /etc/dex/config.yaml<<'EOT'
issuer: https://dex.jeans-box.example.com

storage:
    type: sqlite3
    config:
        file: /var/dex/dex.db

web:
    http: 0.0.0.0:5556
    allowedOrigins: ['*']

staticClients:
    - id: liwasc
      redirectURIs:
          - https://pojntfx.github.io/liwasc/
      name: "liwasc"
      public: true
    - id: bofied
```

```
      redirectURIs:
          - https://pojntfx.github.io/bofied/
      name: "bofied"
      public: true

connectors:
    - type: gitea
      id: gitea
      name: Gitea
      config:
          clientID: yourclientidfromgiteahere
          clientSecret: yourclientsecretfromgiteahere
          redirectURI: https://dex.jeans-box.example.com/callback
          baseURL: https://gitea.jeans-box.example.com
EOT
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
sudo podman generate systemd --new dex | sudo tee /lib/systemd/system/dex.service
sudo systemctl daemon-reload
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

```
sudo mkdir -p /var/lib/liwasc
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_OIDCCLIENTID=liwasc -e LIWASC_BACKEND_DEVICENAME=eth0 -e LIWASC_BACKEND_PERIODICSCANCRONEXPRESSION='0 0 * * *' --name liwasc pojntfx/liwasc-backend
```

```
sudo podman generate systemd --new liwasc | sudo tee /lib/systemd/
    system/liwasc.service
sudo systemctl daemon-reload
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

```
sudo mkdir -p /var/lib/bofied
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_OIDCCLIENTID=bofied -e
    BOFIED_BACKEND_ADVERTISEDIP=100.64.154.249 --name bofied pojntfx/
    bofied-backend
sudo podman generate systemd --new bofied | sudo tee /lib/systemd/
    system/bofied.service
sudo systemctl daemon-reload
sudo systemctl enable --now bofied
sudo firewall-cmd --permanent --add-port=67/udp
sudo firewall-cmd --permanent --add-port=69/udp
sudo firewall-cmd --permanent --add-port=4011/udp
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/`