Uni Software Defined Infrastructure Notes

Felix Pojtinger

November 15, 2021

Contents

1 Introduction

1.1 Contributing

These study materials are heavily based on professor Goik's "Software Defined Infrastructure" lecture at HdM Stuttgart.

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

Figure 1: QR code to source repository

If you like the study materials, a GitHub star is always appreciated :)

1.2 License

Figure 2: AGPL-3.0 license badge

2 Hosts

Add the following A and AAAA records to a public DNS server (with root domain `alphahorizon.io`):

```
felixs-sdi1     10800   IN      A       138.68.70.72
felixs-sdi1     10800   IN      AAAA    2a03:b0c0:3:d0::e34:5001
*.felixs-sdi1   10800   IN      A       138.68.70.72
*.felixs-sdi1   10800   IN      AAAA    2a03:b0c0:3:d0::e34:5001
felixs-sdi2     10800   IN      A       159.223.25.154
felixs-sdi2     10800   IN      AAAA    2a03:b0c0:3:d0::1092:b001
*.felixs-sdi2   10800   IN      A       159.223.25.154
*.felixs-sdi2   10800   IN      AAAA    2a03:b0c0:3:d0::1092:b001
```

3 User

```
ssh root@felixs-sdi1.alphahorizon.io
adduser pojntfx
usermod -aG sudo pojntfx
su pojntfx
```

4 SSH

```
sudo apt update
sudo apt install -y openssh-server
sudo systemctl enable --now ssh
mkdir -p ~/.ssh
chmod 700 ~/.ssh
curl 'https://github.com/pojntfx.keys' | tee -a ~/.ssh/authorized_keys
chmod 600 ~/.ssh/authorized_keys
exit
```

5 UFW

```
ssh pojntfx@felixs-sdi1.alphahorizon.io
sudo apt update
sudo apt install -y ufw
sudo systemctl enable --now ufw
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow OpenSSH
sudo ufw enable
```

6 APT

```
sudo apt update
sudo apt install -y unattended-upgrades

sudo vi /etc/apt/apt.conf.d/50unattended-upgrades # Now replace/add the following:
Unattended-Upgrade::Origins-Pattern {
  "origin=*";
}
Unattended-Upgrade::Automatic-Reboot "true";
Unattended-Upgrade::Automatic-Reboot-Time "02:00";

sudo dpkg-reconfigure unattended-upgrades # Answer with yes
sudo systemctl enable --now unattended-upgrades
sudo unattended-upgrades --debug # Test the configuration; this will install the available u
sudo reboot # If required
```

7 Traefik

```
$ sudo apt update
$ sudo apt install -y docker.io
$ sudo systemctl enable --now docker
$ sudo mkdir -p /etc/traefik
$ sudo tee /etc/traefik/traefik.yaml<<'EOT'
entryPoints:
  dnsTcp:
    address: ":53"

  dnsUdp:
    address: ":53/udp"

  web:
    address: ":80"

  websecure:
    address: ":443"

  websecurealt:
    address: ":8443"

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

```
api:
  dashboard: true

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

log:
  level: INFO
EOT
$ sudo tee /etc/traefik/services.yaml<<'EOT'
udp:
  routers:
    dns:
      entryPoints:
        - dnsUdp
      service: dns
  services:
    dns:
      loadBalancer:
        servers:
          - address: localhost:54

tcp:
  routers:
    dns:
      entryPoints:
        - dnsTcp
      rule: HostSNI(`*`)
      service: dns
    ssh:
      entryPoints:
        - websecurealt
      rule: HostSNI(`*`)
      service: ssh
    sshOverTLS:
      entryPoints:
        - websecure
      rule: HostSNI(`ssh.felixs-sdi1.alphahorizon.io`)
      service: ssh
      tls:
        certResolver: letsencrypt
        domains:
          - main: ssh.felixs-sdi1.alphahorizon.io
    ldap:
      entryPoints:
        - websecure
      rule: HostSNI(`ldap.felixs-sdi1.alphahorizon.io`)
      service: ldap
      tls:
        certResolver: letsencrypt
        domains:
          - main: ldap.felixs-sdi1.alphahorizon.io
  services:
    dns:
      loadBalancer:
        servers:
          - address: localhost:54
    ssh:
      loadBalancer:
        servers:
          - address: localhost:22
    ldap:
      loadBalancer:
        servers:
          - address: localhost:389

  http:
    routers:
      cockpit:
        rule: Host(`cockpit.felixs-sdi1.alphahorizon.io`)
        tls:
          certResolver: letsencrypt
          domains:
            - main: cockpit.felixs-sdi1.alphahorizon.io
        service: cockpit
        entryPoints:
          - websecure
    dashboard:
      rule: Host(`traefik.felixs-sdi1.alphahorizon.io`)
      tls:
        certResolver: letsencrypt
        domains:
          - main: traefik.felixs-sdi1.alphahorizon.io
      service: api@internal
      entryPoints:
        - websecure
      middlewares:
        - dashboard

  middlewares:
    dashboard:
      basicauth:
        users:
          - 'admin:$apr1$wBh8VM6G$bhZ82XpyH3mX4ha9XBbcL1' # htpasswd -nb admin asdf

  services:
    cockpit:
      loadBalancer:
        serversTransport: cockpit
        servers:
          - url: https://localhost:9090

  serversTransports:
    cockpit:
      insecureSkipVerify: true
EOT
$ sudo docker run -d --net=host -v /var/lib/traefik/:/var/lib/traefik -v /etc/traefik/:/etc/
$ sudo ufw allow 'DNS'
$ sudo ufw allow 'WWW'
$ sudo ufw allow 'WWW Secure' # Now visit https://cockpit.felixs-sdi1.alphahorizon.io/
$ sudo ufw allow '8443/tcp'
$ ssh pojntfx@felixs-sdi1.alphahorizon.io # Connect using SSH without Traefik
$ ssh -p 8443 pojntfx@felixs-sdi1.alphahorizon.io # Connect using SSH over Traefik without T
$ ssh -o ProxyCommand="openssl s_client -connect ssh.felixs-sdi1.alphahorizon.io:443 -quiet'
```

8 Cockpit

```
echo 'deb http://deb.debian.org/debian bullseye-backports main' | sudo tee /etc/apt/sources.
sudo apt update
sudo apt install -t bullseye-backports -y cockpit
```

9 DNS

9.1 Manager

```
sudo apt update
sudo apt install -y bind9 bind9utils
sudo systemctl enable --now named

sudo vi /etc/bind/named.conf.options # Now add the following at the end of the options block
listen-on port 54 { 127.0.0.1; };
listen-on-v6 port 54 { ::1; };
```

```
version "not currently available";
recursion yes;
querylog yes;
allow-transfer { none; };
allow-query { any; };

sudo tee -a /etc/bind/named.conf.local <<EOT
zone "example.pojtinger" {
        type master;
        file "/etc/bind/db.example.pojtinger";
        allow-query { any; };
        allow-transfer { 159.223.25.154; 2a03:b0c0:3:d0::1092:b001; };
};

zone "70.68.138.in-addr.arpa" {
        type master;
        file "/etc/bind/db.70.68.138";
        allow-query { any; };
        allow-transfer { 159.223.25.154; 2a03:b0c0:3:d0::1092:b001; };
};

zone "1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2.ip6.arpa" {
        type master;
        file "/etc/bind/db.1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2";
        allow-query { any; };
        allow-transfer { 159.223.25.154; 2a03:b0c0:3:d0::1092:b001; };
};
EOT

# Increase `1634570712` by one and reload after each change to propagate changes to the work
sudo tee /etc/bind/db.example.pojtinger <<EOT
\$ORIGIN example.pojtinger.
\$TTL 3600

example.pojtinger.      IN      SOA     ns1.example.pojtinger. hostmaster.example.pojtinger.

example.pojtinger.      IN      NS      ns1.example.pojtinger.
example.pojtinger.      IN      NS      ns2.example.pojtinger.

example.pojtinger.      IN      A       138.68.70.72
example.pojtinger.      IN      AAAA    2a03:b0c0:3:d0::e34:5001

ns1.example.pojtinger.  IN      A       138.68.70.72
ns1.example.pojtinger.  IN      AAAA    2a03:b0c0:3:d0::e34:5001
```

```
ns2.example.pojtinger.  IN      A       159.223.25.154
ns2.example.pojtinger.  IN      AAAA    2a03:b0c0:3:d0::1092:b001

example.pojtinger.      IN      MX      1       fb.mail.gandi.net.
www.example.pojtinger.  IN      CNAME   example.pojtinger.
EOT

# Increase `1634570724` by one and reload after each change to propagate changes to the work
sudo tee /etc/bind/db.70.68.138 <<EOT
\$ORIGIN 70.68.138.in-addr.arpa.
\$TTL 3600

@       IN      SOA     ns1.example.pojtinger. hostmaster.example.pojtinger.    ( 1634570724

@       IN      NS      ns1.example.pojtinger.
@       IN      NS      ns2.example.pojtinger.

72      IN      PTR     example.pojtinger.
EOT

# Increase `1634570724` by one and reload after each change to propagate changes to the work
sudo tee /etc/bind/db.1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2 <<EOT
\$ORIGIN 1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2.ip6.arpa.
\$TTL 3600

@       IN      SOA     ns1.example.pojtinger. hostmaster.example.pojtinger.    ( 1634570724

@       IN      NS      ns1.example.pojtinger.
@       IN      NS      ns2.example.pojtinger.

1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2.ip6.arpa.     IN      PTR
EOT

sudo named-checkconf

sudo named-checkzone example.pojtinger /etc/bind/db.example.pojtinger
sudo named-checkzone 70.68.138.in-addr.arpa. /etc/bind/db.70.68.138
sudo named-checkzone 1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2.ip6.arpa

sudo systemctl reload named
```

9.2 Worker

```
sudo apt update
sudo apt install -y bind9 bind9utils
sudo systemctl enable --now named
sudo vi /etc/bind/named.conf.options # Now add the following at the end of the options block
listen-on port 54 { 127.0.0.1; };
listen-on-v6 port 54 { ::1; };

version "not currently available";
recursion yes;
querylog yes;
allow-transfer { none; };
allow-query { any; };

sudo tee -a /etc/bind/named.conf.local <<EOT
zone "example.pojtinger" {
        type slave;
        file "db.example.pojtinger";
        allow-query { any; };
        masters { 138.68.70.72; 2a03:b0c0:3:d0::e34:5001; };
};

zone "70.68.138.in-addr.arpa" {
        type slave;
        file "db.70.68.138";
        allow-query { any; };
        masters { 138.68.70.72; 2a03:b0c0:3:d0::e34:5001; };
};

zone "1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2.ip6.arpa" {
        type slave;
        file "db.1.0.0.5.4.3.e.0.0.0.0.0.0.0.0.0.0.d.0.0.3.0.0.0.0.c.0.b.3.0.a.2";
        allow-query { any; };
        masters { 138.68.70.72; 2a03:b0c0:3:d0::e34:5001; };
};
EOT

sudo named-checkconf
sudo systemctl reload named
```

9.3 Exercises

Use the dig command to query A/CNAME/MX/NS records from various machines/domains of your choice. Then execute reverse lookups as well.

```
# Get A/AAA records from manager server
$ dig +noall +answer @138.68.70.72 example.pojtinger A
example.pojtinger.      3600    IN      A       138.68.70.72
```

```
$ dig +noall +answer @138.68.70.72 example.pojtinger AAAA
example.pojtinger.      3600    IN      AAAA    2a03:b0c0:3:d0::e34:5001

# Get A/AAAA records from worker server
$ dig +noall +answer @159.223.25.154 example.pojtinger A
example.pojtinger.      3600    IN      A       138.68.70.72
$ dig +noall +answer @159.223.25.154 example.pojtinger AAAA
example.pojtinger.      3600    IN      AAAA    2a03:b0c0:3:d0::e34:5001

# Get NS record
$ dig +noall +answer @159.223.25.154 example.pojtinger NS
example.pojtinger.      3600    IN      NS      ns1.example.pojtinger.
example.pojtinger.      3600    IN      NS      ns2.example.pojtinger.

# Get CNAME record
$ dig +noall +answer @159.223.25.154 www.example.pojtinger CNAME
www.example.pojtinger.  3600    IN      CNAME   example.pojtinger.

# Do IPv4 reverse lookup
$ dig +short @159.223.25.154 -x 138.68.70.72
example.pojtinger.

# Do IPv6 reverse lookup
$ dig +short @159.223.25.154 -x '2a03:b0c0:3:d0::e34:5001'
example.pojtinger.
```

Enable recursive queries to parent nameservers enabling your nameserver to resolve external machines like www.w3.org by delegation.

```
# Get AAAA record for felix.pojtinger.com using parent nameservers
$ dig +noall +answer @159.223.25.154 felix.pojtinger.com AAAA
felix.pojtinger.com.    123     IN      CNAME   cname.vercel-dns.com.
```

Provide a mail exchange record pointing to mx1.hdm-stuttgart.de. Test this configuration using dig accordingly.

```
# Get MX record
$ dig +noall +answer @159.223.25.154 example.pojtinger MX
example.pojtinger.      3600    IN      MX      1 fb.mail.gandi.net.
```

10 LDAP

```
sudo apt update
sudo apt install -y slapd ldap-utils certbot

sudo dpkg-reconfigure slapd # ldap.felixs-sdi1.alphahorizon.io, felixs-sdi1
sudo ufw allow 'LDAPS' # TODO: Setup certbot

export CERT_LOCATION="$(sudo bash -c 'find /var/lib/caddy/.local/share/caddy/certificates/*/
export KEY_LOCATION="$(sudo bash -c 'find /var/lib/caddy/.local/share/caddy/certificates/*/l

sudo mkdir -p /etc/openldap
sudo tee -a /etc/openldap/slapd.conf <<EOT
# TODO: Chown/ln and use CERT_LOCATION and KEY_LOCATION for `slapd`'s TLS config according t
EOT
```