Network Administration/System Administration (NTU CSIE, Spring 2024) Homework #11 - Nginx

B12902110 呂承諺

May 12, 2024

Web Terminology

1.
2.
3.
4.
5.

Web Server Configurations

6. **Steps**

(1) Run the following commands.

```
$ cp -r /tmp2/nasa-hw11 /tmp2/b1290110
$ cd /tmp2/b12902110/nasa-hw11
$ qemu-img create -f qcow2 disk0.qcow2 20G
```

(2) Create `/tmp2/b12902110/nasa-hw11/run_vm.sh` as the following.

```
#!/bin/bash

readonly MACHINE_IP="$(ip -4 a s net0.30 | grep -oP '(?<=inet\s)\d+(\.\d+){3}')"
qemu-system-x86_64 \
  -enable-kvm \
  -cpu host \
  -smp 8,sockets=1,cores=4,threads=2 \
  -m 8G \
  -nic user,hostfwd=tcp::11022-:22,hostfwd=tcp::11080-:80,hostfwd=tcp::11043-:443 \
  -monitor stdio \
  -vga virtio \
  -vnc ${MACHINE_IP}:0,to=10000,password=on \
  -drive file=disk0.qcow2 \
  -drive file=debian.iso,media=cdrom
```

(3) Boot up the VM, connect to it via QEMU's VNC, and follow the Debian installation guide. After the installation finished, reboot into the VM.

(4) Configure sudo as the root user.

```
$ su
# apt install -y sudo
# usermod -aG sudo b12902110
```

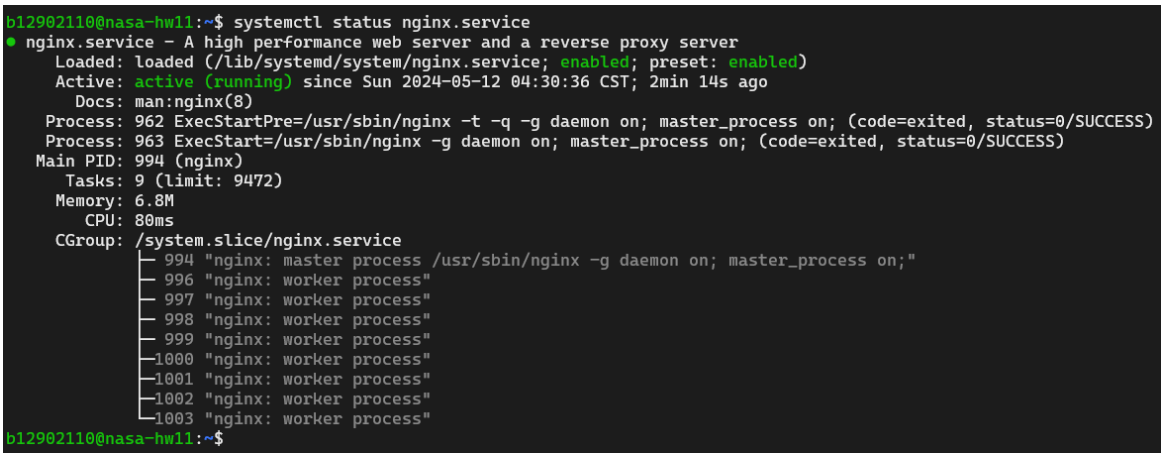
(5) Re-login as user b12902110. Install the necessary package for our web server.

```
$ sudo apt install -y nginx
```

(6) Start the nginx service.

```
$ sudo systemctl start nginx.service
```

Result



References

- command usermod not found

7. Steps

(1) Install `ufw`.

```
$ sudo apt install -y ufw
```

(2) Configure firewall rules with the following commands.

```
$ sudo ufw default deny
$ sudo ufw allow 22
$ sudo ufw allow 80
$ sudo ufw allow 443
$ sudo ufw enable
```

Result

This part is done after the last problem, so we have more services than an HTTP and an SSH service.

In the QEMU console, add another port forwarding rule: 11088 on the host to 8888 on the VM.

```
(qemu) hostfwd_add tcp::11088-:8888
```

Therefore, we have 4 port forwarding rules.

Source	Destination
ws2.csie.ntu.edu.tw:11022	nasa-hw11:22
ws2.csie.ntu.edu.tw:11080	nasa-hw11:80
ws2.csie.ntu.edu.tw:11043	nasa-hw11:443
ws2.csie.ntu.edu.tw:11088	nasa-hw11:8888

All of the 4 ports has a service running on it.

```
b12902110@nasa-hw11:/var/www/hostB$ netstat -tl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:9999            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:http            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:https           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8888            0.0.0.0:*               LISTEN
tcp6       0      0 [::]:9999               [::]:*                  LISTEN
tcp6       0      0 [::]:http               [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 [::]:https              [::]:*                  LISTEN
tcp6       0      0 [::]:8888               [::]:*                  LISTEN
b12902110@nasa-hw11:/var/www/hostB$
```

However, only ports 22, 80, and 443 are accessible from outside the VM.

```
b12902110@ws2: /tmp2/b12902110/nasa-hw11
$ ssh -p 11022 b12902110@127.0.0.1
The authenticity of host '[127.0.0.1]:11022 ([127.0.0.1]:11022)' can't be established.
ED25519 key fingerprint is SHA256:EJBy9bOgAc78EF9Fs+NGOvtWEneStJlGuYyFv6KuNog.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? ^C
[130]

b12902110@ws2: /tmp2/b12902110/nasa-hw11
$ curl http://localhost:11080
<!DOCTYPE html>
<html>
  <head>
    <title>Hello! My name is b12902110!</title>
  </head>
  <body>
    <h1>Hello! My name is b12902110!</h1>
  </body>
</html>

b12902110@ws2: /tmp2/b12902110/nasa-hw11
$ curl --insecure https://localhost:11043
<!DOCTYPE html>
<html>
  <head>
    <title>Hello! My name is b12902110!</title>
  </head>
  <body>
    <h1>Hello! My name is b12902110!</h1>
  </body>
</html>

b12902110@ws2: /tmp2/b12902110/nasa-hw11
$ curl http://localhost:11088
curl: (56) Recv failure: Connection reset by peer
[56]
```

References

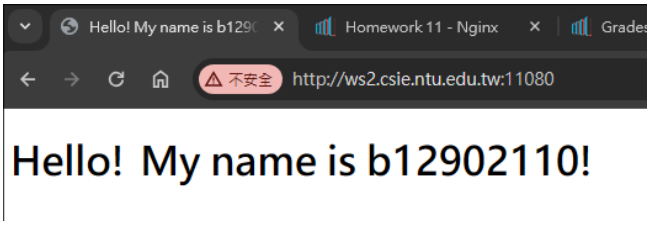
- How To Open a Port on Linux | DigitalOcean
- How to Set Up a Firewall with UFW on Ubuntu | DigitalOcean

8. **Steps**

Create `/var/www/html/index.html` as the following.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello! My name is b12902110!</title>
  </head>
  <body>
    <h1>Hello! My name is b12902110!</h1>
  </body>
</html>
```

Result



9. **Steps**

(1) Add the following `location` block into `/etc/nginx/sites-available/default`.

```
server {
  ...

  location ~ ^/~(.*?)/(.*) {
    alias /home/$1/htdocs/$2;
  }

  ...
}
```

(2) Run the following commands.

```
$ chmod 755 /home/b12902110
$ mkdir /home/b12902110/htdocs
```

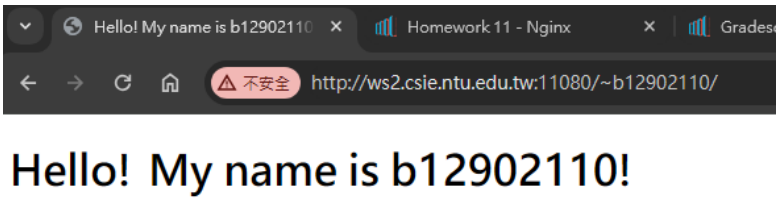
(3) Create `/home/b12902110/htdocs/index.html` as the following.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello! My name is b12902110!</title>
  </head>
  <body>
    <h1>Hello! My name is b12902110!</h1>
  </body>
</html>
```

(4) Reload the nginx service.

```
$ sudo systemctl reload nginx.service
```

Result



References

- nginx user public home without - Stack Overflow
- Beginner's Guide
- Module ngx_http_core_module

10. **Steps**

(1) Add the following `location` block into `/etc/nginx/sites-available/default`.

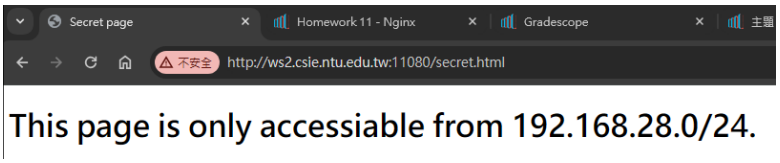
```
server {
  ...

  location = /secret.html {
    allow 192.168.28.0/24;
    deny all;
  }

  ...
}
```

(2) Reload the nginx service.

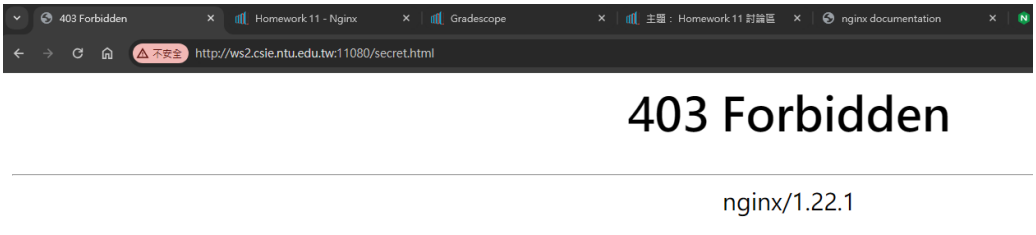
```
$ sudo systemctl reload nginx.service
```

Result

Before restriction:



After restriction:



References

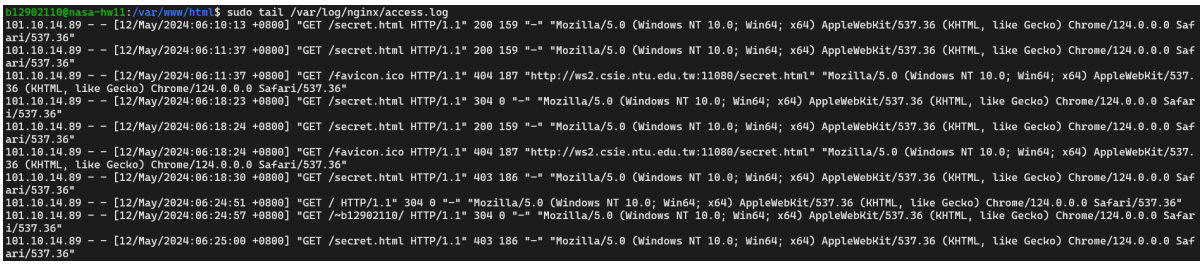
- Module ngx_http_access_module

11. **Steps**

View the last few lines of `/var/log/nginx/access.log/`

```
$ sudo tail /var/log/nginx/access.log
```

Result



References

- Configuring Logging | NGINX Documentation

12. (a)

    (b)

    (c) **Steps (Server)**

    (1) Create CA key and certificate.

```
$ openssl req -new -x509 -noenc -days 365000 -keyout ca-key.pem \
    -out ca-cert.pem

...

-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taiwan
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NTU CSIE
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:
```

    (2) Create server key and certificate request.

```
$ openssl req -new -noenc -keyout server-key.pem -out server-req.pem

...

-----
Country Name (2 letter code) [AU]:TW
State or Province Name (full name) [Some-State]:Taiwan
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NTU CSIE
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:nasa-hw11
Email Address []:
```

    (3) Sign the server certificate.

```
$ openssl x509 -req -days 365000 -in server-req.pem \
    -out server-cert.pem -CA ca-cert.pem -CAkey ca-key.pem
Certificate request self-signature ok
subject=C = TW, ST = Taiwan, O = NTU CSIE, CN = nasa-hw11
```

    (4) Install the certificates to `/etc/nginx`.

```
$ sudo cp server-key.pem server-cert.pem /etc/nginx
$ sudo chown www-data:www-data /etc/nginx/server-key.pem
```

(5) Configure the following `server` settings in `/etc/nginx/sites-available/default`.

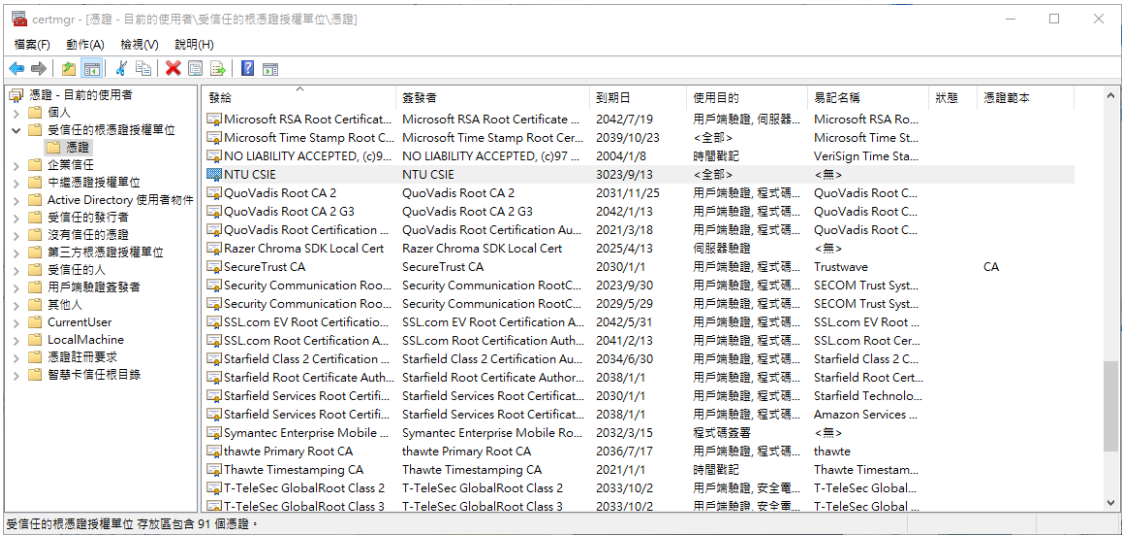
```
server {
  ...

  listen 443 ssl default_server;
  listen [::]:443 ssl default_server;
  ssl_certificate server-cert.pem;
  ssl_certificate_key server-key.pem;

  ...
}
```

(6) Reload the nginx service.

```
$ sudo systemctl reload nginx.service
```

Steps (Windows Client)

Run `certmgr.msc`, and install `ca-cert.pem` to "Trusted Root Certification Authorities".



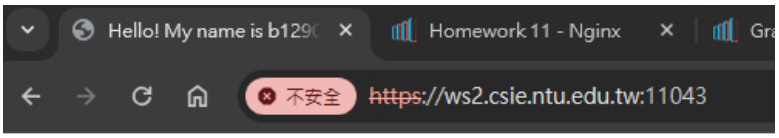
Result

Certificates:

```
b12902110@nasa-hw11:~$ openssl x509 -in ca-cert.pem -text -nooout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            05:21:a8:a9:65:79:b1:fe:79:ec:ac:1b:79:e2:66:ea:40:28:9b:37
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = TW, ST = Taiwan, O = NTU CSIE
        Validity
            Not Before: May 11 22:56:35 2024 GMT
            Not After : Sep 12 22:56:35 3023 GMT
        Subject: C = TW, ST = Taiwan, O = NTU CSIE
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:da:a6:ab:6d:fa:7b:90:43:79:6d:7e:4b:1d:e2:
                    67:35:c8:d4:ef:f5:c8:39:41:5a:15:67:a7:7f:a6:
                    e2:25:e5:8f:3e:77:32:e0:94:b4:eb:44:f7:35:05:
                    fa:19:a7:c6:82:c5:e7:32:6e:56:a3:8a:bb:42:f0:
                    ff:f8:c0:2c:42:eb:c9:85:94:93:f0:d5:6c:5c:a9:
                    d1:58:e1:77:e3:93:fc:12:b9:0a:86:04:3f:6e:86:
                    ea:fa:f5:30:1a:b7:49:be:62:18:09:1f:83:c2:18:
                    c4:5f:b9:8b:fc:84:d8:68:af:a3:3f:23:be:4e:d2:
                    a3:ca:ea:04:7d:a7:c4:c1:f8:c4:22:21:69:59:7b:
                    3a:6a:f5:93:6c:17:6b:b3:7a:3f:31:a2:8b:47:c3:
                    30:43:e9:08:14:88:62:a5:d7:fa:8a:d3:e3:45:29:
                    7c:57:c1:a4:a1:c4:27:aa:f5:ac:4e:be:cd:82:d9:
                    1b:8a:83:5b:06:45:5e:52:77:71:0b:db:ff:b8:27:
                    bc:c0:6f:8e:d5:2d:98:b8:c0:cf:c7:76:65:60:84:
                    2f:a0:f3:9f:dc:9a:08:69:22:77:46:b9:03:3e:fa:
                    71:d0:3f:9e:c7:ea:79:11:00:39:bc:61:1d:79:c8:
                    44:56:9b:8f:ba:70:a2:f7:75:c1:43:f9:4d:cc:e6:
                    e7:eb
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                A7:CF:3A:78:A3:00:B7:29:5D:77:99:91:D2:5F:14:F1:92:2F:BF:AC
            X509v3 Authority Key Identifier:
                A7:CF:3A:78:A3:00:B7:29:5D:77:99:91:D2:5F:14:F1:92:2F:BF:AC
            X509v3 Basic Constraints: critical
                CA:TRUE
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        77:75:22:63:a0:0c:2c:3f:9e:3b:88:8a:73:14:63:e6:28:a3:
        2c:0d:a7:2c:44:c8:e0:5f:84:17:38:74:d8:e2:8d:c3:e6:0e:
        a0:cf:43:45:7f:22:c9:fc:9b:6e:8f:cd:1b:a8:00:ae:3f:9e:
        73:f8:0b:89:d5:64:55:87:a8:da:6f:3c:4e:e5:86:52:df:2c:
        ec:24:cb:1d:7b:97:f6:d6:fd:76:74:6a:65:14:f9:6e:28:44:
        4a:8c:7d:5d:db:f0:34:ba:b8:d8:c0:7e:b1:cc:d0:92:f6:12:
        03:3a:24:b7:f6:1b:68:d9:a6:f1:36:57:d7:a9:e9:59:e5:8b:
        be:f7:5d:59:0f:c2:58:57:2e:6b:5e:05:11:04:98:9c:e9:a9:
        c6:a3:f7:1c:2f:6a:6f:e9:d1:24:c3:e6:b4:24:ad:26:ec:ef:
        9f:01:05:af:f5:eb:08:7c:cc:94:99:03:98:e3:fc:b9:67:34:
        11:f0:34:58:ec:8c:3d:c2:9f:9b:b0:5e:80:25:4d:e9:90:4a:
        24:31:2a:1c:3f:49:74:cc:b1:19:98:27:56:7a:0f:a2:90:cf:
        ee:c2:49:a9:05:46:1d:4b:30:33:db:54:98:40:48:6c:64:73:
        ee:a3:bf:d2:61:61:a6:2b:22:5d:4c:83:7e:d4:8b:f2:41:70:
        c6:a5:15:c3
```

```
b12902110@nasa-hw11:~$ openssl x509 -in server-cert.pem -text -nooout
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number:
            1d:3a:d4:1e:c4:99:94:64:63:43:b5:c5:7f:18:e9:04:05:f2:d4:fc
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: C = TW, ST = Taiwan, O = NTU CSIE
        Validity
            Not Before: May 11 23:15:54 2024 GMT
            Not After : Sep 12 23:15:54 3023 GMT
        Subject: C = TW, ST = Taiwan, O = NTU CSIE, CN = default_server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d2:50:7a:d3:16:72:df:bb:df:13:d4:57:ed:fc:
                    23:a2:e8:99:74:2c:d8:20:f7:55:cd:a2:8d:3b:85:
                    65:f4:cb:bd:36:e4:70:0a:38:22:46:42:7e:59:39:
                    77:5d:ed:4e:f5:f3:04:b5:e2:d8:e6:92:23:ce:8c:
                    6f:bf:55:af:37:b1:eb:a3:3c:46:c3:e7:35:29:ac:
                    d2:58:17:a4:dc:8a:09:9e:a5:0d:55:3c:54:26:35:
                    56:c2:12:a1:79:22:bb:da:5f:72:82:c7:6a:f7:31:
                    49:e5:b9:fc:23:63:24:46:a5:13:95:9f:d4:80:96:
                    77:35:ad:2f:42:c7:eb:1c:64:92:b3:bf:50:4a:1e:
                    c6:d2:3d:0c:62:03:13:ad:68:b2:4d:28:98:df:aa:
                    b1:ed:60:21:0b:e6:41:f8:69:fb:28:2d:34:1f:fa:
                    d3:cc:a1:2e:99:6c:9f:57:ef:59:9b:72:50:f5:87:
                    0f:f7:78:1d:86:0a:46:36:64:a8:13:65:f2:3f:86:
                    0d:7c:e0:98:96:0c:a9:13:56:30:b4:b6:d9:84:2a:
                    ec:02:1e:35:fb:8f:c8:ea:e3:e9:05:fb:81:68:a2:
                    d3:f5:6e:71:b6:8c:9a:a2:47:bf:0d:6b:0c:59:ac:
                    f2:74:c8:68:ec:76:72:75:af:70:8e:3b:25:bd:f9:
                    cf:e7
                Exponent: 65537 (0x10001)
    Signature Algorithm: sha256WithRSAEncryption
    Signature Value:
        6f:19:39:d8:9d:e9:01:7f:94:e2:3d:30:d8:10:ae:3e:0e:7e:
        94:9f:e9:cd:b7:af:d3:ce:6c:ce:20:a1:04:21:f9:a4:65:b8:
        a4:89:a6:a6:6a:8e:ba:ec:14:9d:4e:0c:5f:47:6a:96:e6:e2:
        c1:70:8e:87:eb:0c:66:1e:62:34:e2:72:71:cb:aa:9d:98:91:
        92:f0:74:54:6d:c1:fe:23:b0:9f:40:a5:3b:2a:21:bf:27:19:
        71:f1:a1:c0:d9:e9:87:9a:8b:9e:cb:15:4f:10:62:7f:77:2c:
        92:89:d1:42:c3:91:30:be:8e:8f:3c:a3:9f:e0:c7:65:66:1b:
        a7:6c:4c:94:b4:49:0f:69:ad:ee:37:c3:8e:e2:51:27:e1:0e:
        5f:dc:bf:55:e3:da:e9:95:27:eb:d5:a6:b8:da:8e:35:2a:73:
        38:dd:2c:1a:9b:b3:97:49:40:f5:29:9e:79:7c:45:89:74:41:
        eb:26:fe:72:61:2a:b2:c4:8f:a0:7e:f1:21:c5:92:86:9b:c3:
        2e:db:4e:49:7b:b6:5a:89:6e:e2:08:bf:bd:9b:46:b7:b2:ed:
        50:5d:dc:90:f6:79:08:e1:c5:bf:8b:61:07:00:3f:24:d4:4b:
        07:d6:18:b7:48:da:a8:16:4b:0b:c4:ef:01:a4:34:ad:f7:82:
        5a:4e:be:73
```

Browser:



Hello! My name is b12902110!

Since we're using NAT here, it still shows up as insecure due to `NET::ERR_CERT_COMMON_NAME_INVALID`.

References

- Create Self-Signed Certificates and Keys with OpenSSL —MariaDB Documentation
- /docs/man3.0/man1/openssl-req.html
- /docs/man3.0/man1/openssl-x509.html

Reverse Proxy

13. **Steps**

(1) Create `/etc/nginx/sites-available/hostA` as the following.

```
server {
  listen 8888 default_server;
  listen [::]:8888 default_server;
```

```
  root /var/www/hostA;
  index index.html;

  server_name hostA;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

(2) Create `/etc/nginx/sites-available/hostB` as the following.

```
server {
  listen 9999 default_server;
  listen [::]:9999 default_server;

  root /var/www/hostB;
  index index.html;

  server_name hostB;

  location / {
    try_files $uri $uri/ =404;
  }
}
```

(3) Enable both `hostA` and `hostB`.

```
$ sudo ln -s /etc/nginx/sites-available/hostA \
    /etc/nginx/sites-enabled/hostA
$ sudo ln -s /etc/nginx/sites-available/hostB \
    /etc/nginx/sites-enabled/hostB
```

(4) Create `/var/www/hostA/index.html` as the following.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Host A index.html</title>
  </head>
  <body>
    <h1>Hi! This is host A.</h1>
  </body>
</html>
```

(5) Create `/var/www/hostB/index.html` as the following.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Host B index.html</title>
  </head>
  <body>
```

```
    <h1>Hi! This is host B.</h1>
  </body>
</html>
```

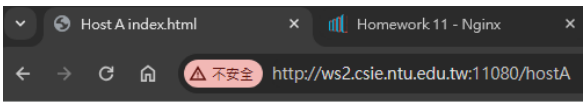
(6) Add `/hostA` and `/hostB` location blocks to `/etc/nginx/sites-available/default`.

```
server {
  ...

  location /hostA {
    proxy_pass http://127.0.0.1:8888/;
  }

  location /hostB {
    proxy_pass http://127.0.0.1:9999/;
  }

  ...
}
```

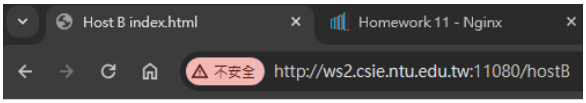
(7) Reload the nginx service.

```
$ sudo systemctl reload nginx.service
```

Result



Hi! This is host A.



Hi! This is host B.

References

- Beginner's Guide
- Module ngx_http_proxy_module