Getting started with HTTPS for team project: masterclass

Team Project 2022-23

This session is on setting up secure client communication with our applications. We have assigned each team a Domain Name in the public DNS to the IP address of your Virtual Machine e.g. `mc.bham.team` points to `138.68.150.235`. Your Domain is `team00-22.bham.team` - where *team00-22* is your Canvas team name.

This session will use the terminal and an Ubuntu Virtual Machine. However, after installation of nginx, the principles are the same as if using an different VM OS.

This session is not exhaustive. Similar to git, CI and JHipster many of the commands are rarely used and it is completely fine to consult the documentation for nginx when needed.

Terminology

`nginx` - pronounced 'Engine X' a web server that can be configured to handle secure communication using HTTPS. After handling HTTPS, Nginx can forward requests to another application like JHipster, which makes nginx a reverse proxy. **In this masterclass:** we will configure nginx for secure communication.

DNS - Domain Name System URLs must be converted to IP addresses using a DNS server, before further connection. Every Domain Name in the public DNS is registered to someone. **In this masterclass:** we will use the public domain for our VM and SSL certificates specific to our domain.

HTTPS - is a secure version of the older Hyper Text Transfer Protocol (HTTP) that uses port 80. HTTPS uses port 443 and SSL certificates to establish encrypted communication. **In this masterclass:** a VM will be configured to accept HTTPS connections from e.g. `https://mc.bham.team`

SSL certificates - Secure Sockets Layer certs have a public key, private key and domain. Browsers check a web server is providing a valid and trusted SSL certificate for the domain. The public key in the certificate is used to encrypt communication sent to the server. Only the private key can decrypt SSL client communication. Trusted SSL certificates can only be issued to the Domain Name owner by a Certificate Authority. **In this masterclass:** we will install SSL certificates into nginx.

Letsencrypt - is a Certificate Authority that provides free certificates if you can prove ownership of a domain. **In this masterclass:** we will use SSL certificates from Letsencypt.

`certbot` - certbot is a command program to get free SSL certificates from letsencrypt. If a server is public, ownership of a domain can be verified by certbot and a web server automatically. The domain must point to the VM IP in DNS. **Not** in this masterclass: use certbot to request SSL certificates on a public VM.

Pre-requisites

For this masterclass you will need:

- ssh access to a VM
- ssl certificates - check the files section of your team in Canvas
- `unzip` installed on the VM

Install `nginx`

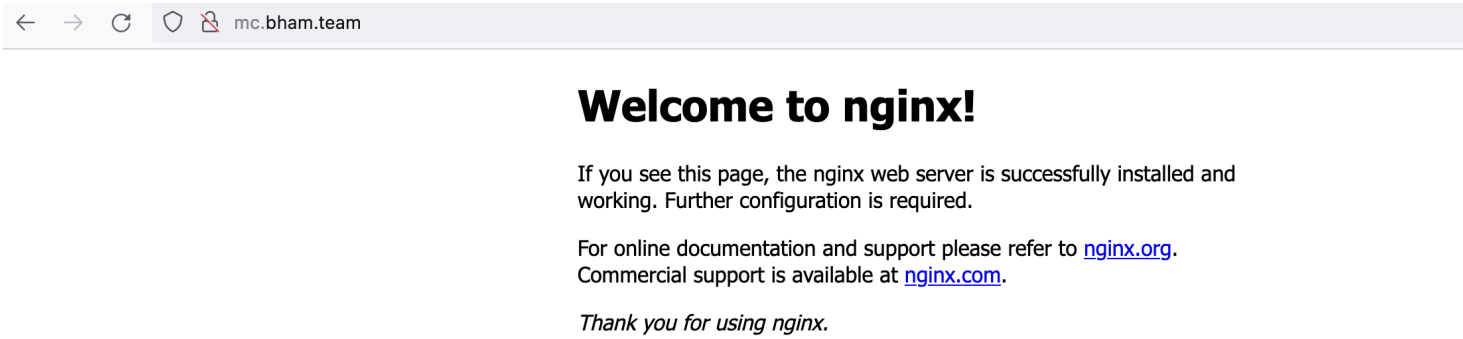
The first step is to install `nginx` on the VM, using SSH. The command used to install depends on the OS of the VM. For Ubuntu:

```
sudo apt update && sudo apt install nginx unzip -y
sudo ufw allow 'Nginx Full'                                    try it on codepad!
```

on Amazon EC2 VM (University VM):

```
sudo yum install nginx unzip
sudo /etc/init.d/nginx start                                   try it on codepad!
```

Now connect to the `nginx` server to test it works: http://mc.bham.team:80



Note the 'insecure' padlock red line- on a mobile device you may not be able to connect at all.

Copy and Install SSL certificates onto VM

Copy the zip file with the certificates to your VM using SCP:

```
scp mc.bham.team.zip root@mc.bham.team:~/mc.zip
```

unzip the certificates:

```
ssh root@mc.bham.team
unzip mc.zip
chmod o-rx -R mc.bham.team
```

n.b. `mc.bham.team` must be the domain or IP of your VM

n.b. the private key `privkey.pem` must be kept private and secure. Do not add this to your git repository.

Now Configure nginx to:

- Use the certificates and accept requests on port 443
- Redirect requests from port 80 to 443
- Forward requests to JHipster- running internally on port 8080.

Default nginx configuration

Without comments, the default nginx config looks like:

cat /etc/nginx/sites-available/default

```
server {
  listen 80 default_server;
  listen [::]:80 default_server;

  root /var/www/html;

  index index.html index.htm index.nginx-debian.html;

  server_name _;

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

try it on codepad!

Use SSL certificates

nano /etc/nginx/sites-available/default

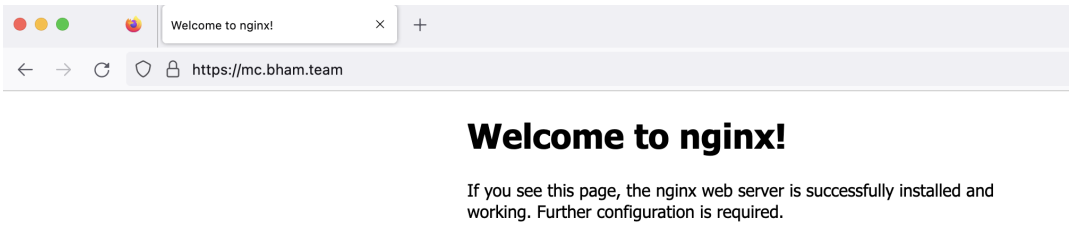
```
server {
  listen 443 ssl default_server;
  listen [::]:443 ssl default_server;

  ssl_certificate /root/mc.bham.team/fullchain1.pem;
  ssl_certificate_key /root/mc.bham.team/privkey1.pem;

  root /var/www/html;

  index index.html index.htm index.nginx-debian.html;

  server_name mc.bham.team;

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

try it on codepad!

Test the config and restart nginx:

```
nginx -t
nginx -s reload
```

Now the following works: https://mc.bham.team



Redirect plain http to https

Now http://mc.bham.team:80 does not work. Add the following server section, after the last }

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;
    return 301 https://$host$request_uri;
}
```
try it on codepad!

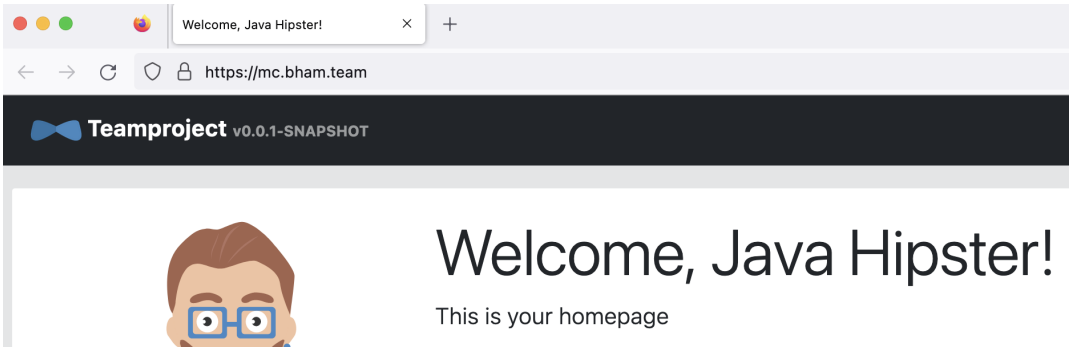
which redirects any requests to `https://`

Redirect to JHipster

Change the `location /` section:

```
location / {
    proxy_pass http://127.0.0.1:8080/;
}
```
try it on codepad!

which redirects any request to JHipster running on `http://127.0.0.1:8080/`



After this step we can edit `src/main/docker/app.yml` line 14 to `127.0.0.1:8080:8080` to stop public connections on port 8080, so that the http://mc.bham.team:8080 no longer works.

Transparent redirects to JHipster

To solve some errors with cookies and CORS for JHipster:

```
location / {
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header Host $http_host;
    proxy_pass http://127.0.0.1:8080/;
}
```
try it on codepad!

Optional

There is more than one way to achieve HTTPS for your apps. Other options for setting up HTTPS include:

- Run certbot* directly on the VM to get the certificates
- Run certbot* and nginx via Docker in the CI pipeline
- Register a domain for your team - free options for domains are available in the GitHub Student Developer Pack, then get a certificate from you DNS provider or letsenrypt via certbot*
- (not recommended) Install SSL certificates using Spring Boot `server.ssl` configuration keys per https://www.jhipster.tech/production/#https-support, via the CI pipeline

N.b. *running `certbot` requires the DNS entry to point to a VM with port 80 open i.e. a non-University VM, or milestone 2+.

N.b. there are rate limits for Letsencypt, which we may hit on this module for `*.bham.team` There is no restriction on renewals.

Deployment

The deployed app from the masterclass can be found here:

https://mc.bham.team

and check the certificate status on SSL labs:

https://www.ssllabs.com/ssltest/analyze.html?d=mc.bham.team