

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 with a different VM OS. Use the codepad links e.g. [try it on codepad!](#) to copy and paste configurations.

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 certificates 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 (or signed) 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 Letsencrypt.



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
- nginx & unzip installed on the VM

Install nginx and unzip

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:

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

on Amazon EC2 VM (University VM):

```
1 sudo amazon-linux-extras install -y nginx1
2 sudo systemctl enable nginx
3 sudo systemctl restart nginx
```

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



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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

Install SSL certificates onto the VM

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

```
scp -i ~/TeamProject2022-0.pem mc.bham.team.zip root@mc.bham.team:~/mc.zip
```

unzip the certificates:

```
ssh -i ~/TeamProject2022-0.pem 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.

Configure nginx

Now we will configure nginx to:

- Use SSL 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
```

```
1 server {
2     listen 80 default_server;
3     listen [::]:80 default_server;
4
5     root /var/www/html;
6
7     index index.html index.htm index.nginx-debian.html;
8
9     server_name _;
10
11     location / {
12         try_files $uri $uri/ =404;
13     }
14 }
```

try it on codepad!

Use SSL certificates

on Ubuntu: nano /etc/nginx/sites-available/default

on EC2 uni VM: sudo nano /etc/nginx/conf.d/default.conf

```
1 server {
2     listen 443 ssl default_server;
3     listen [::]:443 ssl default_server;
4
5     ssl_certificate /root/mc.bham.team/fullchain1.pem;
6     ssl_certificate_key /root/mc.bham.team/privkey1.pem;
7     ssl_protocols TLSv1.2 TLSv1.3;
8
9     root /var/www/html;
10
11     index index.html index.htm index.nginx-debian.html;
12
13     server_name mc.bham.team;
14
15     location / {
16         try_files $uri $uri/ =404;
17     }
18 }
```

try it on codepad!

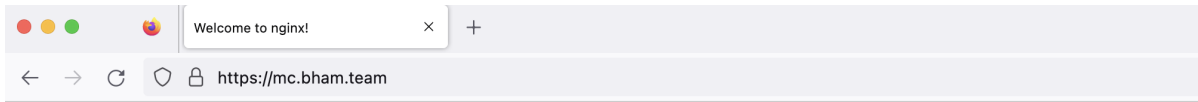
The home directory is /home/ec2-user/ not /root/ on Amazon VMs

Test the config and restart nginx:

```
sudo nginx -t
```

```
sudo nginx -s reload
```

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



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

Redirect plain http to https

Now `http://mc.bham.team:80` does not work.

On an EC2 uni VM edit the server section in `sudo nano /etc/nginx/nginx.conf` so it is as below:

on Ubuntu, add the following server section, after the last `}` in `/etc/nginx/sites-available/default`:

```
1 server {
2     listen 80 default_server;
3     listen [::]:80 default_server;
4     return 301 https://$host$request_uri;
5 }
```

try it on codepad!

which redirects any `http://` requests to `https://`

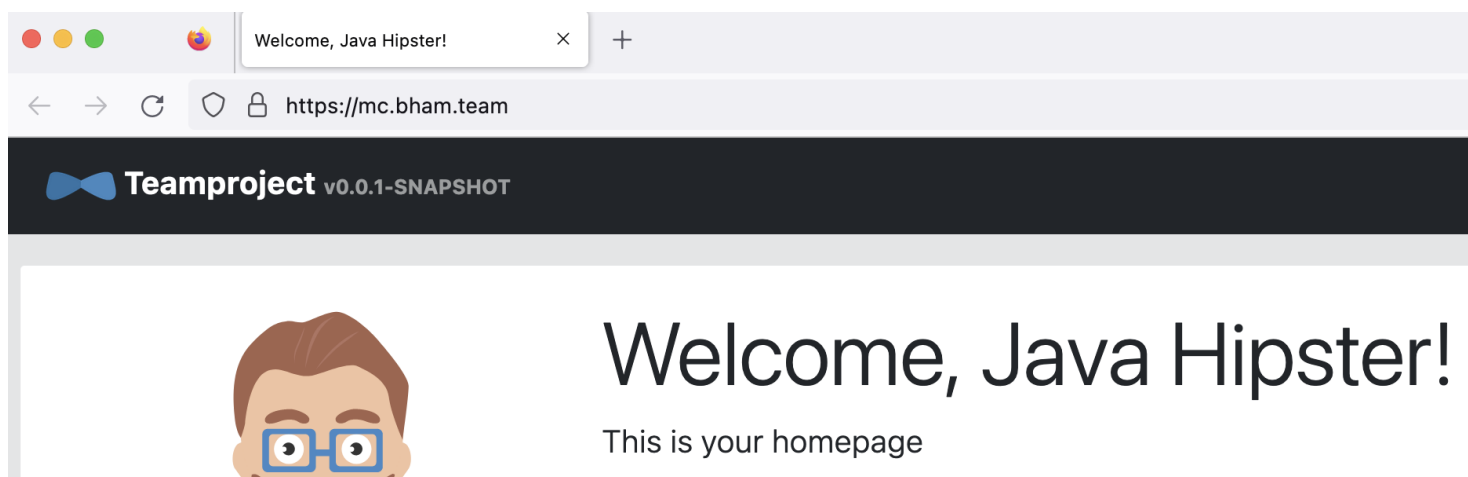
Redirect to JHipster

Change the location / section:

```
1 location / {
2     proxy_pass http://127.0.0.1:8080/;
3 }
```

try it on codepad!

which redirects `https://` requests to JHipster running on `http://127.0.0.1:8080/`.



We should make sure `src/main/docker/app.yml` line 14 is `127.0.0.1:8080:8080` to stop public connections on port 8080. The URL `http://mc.bham.team:8080` should no longer work. The app will need to be restarted using `docker compose` or gitlab ci redeploy for the change to take effect.

Transparent redirects to JHipster

To solve some errors with cookies and CORS for JHipster:

```
1 location / {  
2     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
3     proxy_set_header X-Forwarded-Proto $scheme;  
4     proxy_set_header X-Real-IP $remote_addr;  
5     proxy_set_header Host $http_host;  
6     proxy_pass http://127.0.0.1:8080/;  
7 }
```

try it on codepad!

Optional- Other HTTPS Setup Mechanisms

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 letsencrypt 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 University VM in milestone 2+.

N.b. there are [rate limits](#) for LetsEncrypt issuing new certificates, which we may hit on this module for the sub domains `*.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>