

Web server

- Web server là gì?

Web server là máy chủ cài đặt các chương trình phục vụ các ứng dụng web. Nó có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao thức HTTP hoặc các giao thức khác. Có nhiều web server khác nhau như: Apache, Nginx, IIS, ...

- Cách thức hoạt động:

1. Trình duyệt phân giải tên miền thành địa chỉ IP

Trình duyệt web của bạn trước tiên cần phải xác định địa chỉ IP nào mà tên miền trở về. Trình duyệt sẽ yêu cầu thông tin từ một hoặc nhiều máy chủ DNS (thông qua internet). Máy chủ DNS sẽ cho trình duyệt biết địa chỉ IP nào tên miền sẽ trở đến cũng là nơi đặt trang web.

Lúc này trình duyệt web đã biết địa chỉ IP của trang web, nó có thể yêu cầu URL đầy đủ từ webserver.

2. Webserver gửi lại client trang được yêu cầu

Web server phản hồi bằng cách gửi lại những thông tin client yêu cầu... Nếu trang không tồn tại hoặc có lỗi khác xảy ra, nó sẽ gửi lại thông báo lỗi thích hợp.

3. Trình duyệt hiển thị trang web

Trình duyệt web của bạn nhận lại được các tập tin html css (nhiều file khác)... và render hiển thị trang theo yêu cầu.

Apache Server

- Apache là gì?

Apache là phần mềm web server miễn phí mã nguồn mở, nó cho phép bạn thiết lập một website an toàn mà không tốn nhiều công sức. Và nó thường được chọn

bởi những người kinh doanh tự thân và một doanh nghiệp nhỏ, để tạo thương hiệu trên mạng.

- **Apache Web Server hoạt động như thế nào?**

Apache không phải là server vật lý, nó là một phần mềm chạy trên server đó. Công việc của nó là thiết lập kết nối giữa server và trình duyệt người dùng rồi chuyển file tới và lui giữa chúng. Apache là một phần mềm đa nền tảng, nó hoạt động tốt với cả server Unix và Windows.

ví dụ: trang chủ “Wisky.vn”, trình duyệt người dùng sẽ gửi yêu cầu tải trang web đó lên server và Apache sẽ trả kết quả với tất cả đầy đủ các file cấu thành nên trang Wisky (hình ảnh, chữ, css, html, ...). Server và client giao tiếp với nhau qua giao thức HTTP và Apache chịu trách nhiệm cho việc đảm bảo tiến trình này diễn ra mượt mà và bảo mật giữa 2 máy.

- **Ưu điểm và khuyết điểm của Apache**

Ưu điểm:

1. Phần mềm mã nguồn mở và miễn phí, kể cả cho mục đích thương mại.
2. Phần mềm đáng tin cậy, ổn định.
3. Được cập nhật thường xuyên, nhiều bản vá lỗi bảo mật liên tục.
4. Linh hoạt vì có cấu trúc module.
5. Dễ cấu hình, thân thiện với người mới bắt đầu
6. Đa nền tảng (hoạt động được cả với server Unix và Windows).
7. Hoạt động cực kỳ hiệu quả với WordPress sites.
8. Có cộng đồng lớn và sẵn sàng hỗ trợ với bất kỳ vấn đề nào.

Khuyết điểm:

1. Gặp vấn đề hiệu năng nếu website có lượng truy cập cực lớn.
2. Quá nhiều lựa chọn thiết lập có thể gây ra các điểm yếu bảo mật.

- **Install and Configure Apache Server in CentOS**

- **Install:**

➤ Ta dùng lệnh

```
sudo dnf install httpd
```

```
[root@localhost hoangvo]# sudo dnf install httpd
CentOS Linux 8 - AppStream          436 kB/s | 8.8 MB    00:20
CentOS Linux 8 - BaseOS             366 kB/s | 5.6 MB    00:15
CentOS Linux 8 - Extras             15 kB/s | 10 kB     00:00
Dependencies resolved.
=====
Package                Arch    Version                                Repo                Size
=====
Installing:
httpd                  x86_64  2.4.37-39.module_el8.4.0+778+c970deab appstream            1.4 M
Installing dependencies:
apr                   x86_64  1.6.3-11.el8                        appstream            125 k
apr-util              x86_64  1.6.1-6.el8                         appstream            105 k
centos-logos-httpd    noarch  85.8-1.el8                          baseos                75 k
httpd-filesystem      noarch  2.4.37-39.module_el8.4.0+778+c970deab appstream             38 k
httpd-tools           x86_64  2.4.37-39.module_el8.4.0+778+c970deab appstream            106 k
mod_http2             x86_64  1.15.7-3.module_el8.4.0+778+c970deab appstream            154 k
Installing weak dependencies:
apr-util-bdb          x86_64  1.6.1-6.el8                        appstream             25 k
apr-util-openssl      x86_64  1.6.1-6.el8                        appstream             27 k
Enabling module streams:
httpd                  2.4
```

○ Configure:

➤ Kiểm tra apache đã mở hay chưa:

```
sudo systemctl status httpd
```

```
[root@localhost hoangvo]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
lines 1-4/4 (END)
```

(ở đây của mình đang bị tắt)

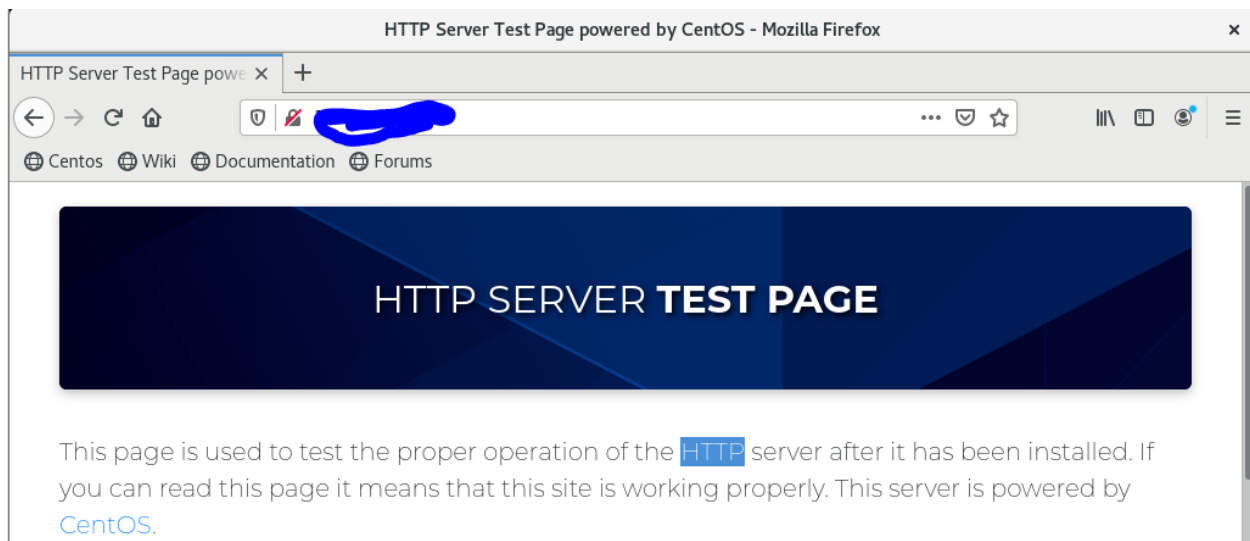
➤ Mở apache ta dùng lệnh:

```
sudo systemctl start httpd
```

```
[root@localhost hoangvo]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
  Docs: man:httpd.service(8)

[root@localhost hoangvo]# sudo systemctl start httpd
[root@localhost hoangvo]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: enabled)
  Active: active (running) since Fri 2021-08-27 04:59:57 EDT; 3s ago
  Docs: man:httpd.service(8)
  Main PID: 36965 (httpd)
  Status: "Started, listening on: port 80"
  Tasks: 213 (limit: 4621)
  Memory: 22.2M
  CGroup: /system.slice/httpd.service
          └─36965 /usr/sbin/httpd -DFOREGROUND
             └─36966 /usr/sbin/httpd -DFOREGROUND
                └─36967 /usr/sbin/httpd -DFOREGROUND
                   └─36968 /usr/sbin/httpd -DFOREGROUND
                      └─36969 /usr/sbin/httpd -DFOREGROUND
```

➤ Để check cài thành công ta vào web browser nhập ip máy



Nếu hiện ra như thế này là đã cài thành công

➤ Để dừng apache ta dùng lệnh stop:

sudo systemctl stop httpd

- Deploy REACT JS In APACHE

- Tạo một file config cho ứng dụng của bạn Sau đó thêm đoạn code thiết lập dưới đây vào file vừa được tạo và lưu lại Sau đó thêm đoạn code thiết lập dưới đây vào file vừa được tạo và lưu lại

nano /etc/httpd/conf.d/demo.conf

```
<VirtualHost *:8080>
  DocumentRoot /var/www/html/build
  <Directory "/var/www/html/build">
    Options FollowSymLinks
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

Vì đây mình dùng port 8080 nên sửa port 8080 bên http

nano /etc/httpd/conf/httpd.conf

sửa chỗ listen thành port mình đã config

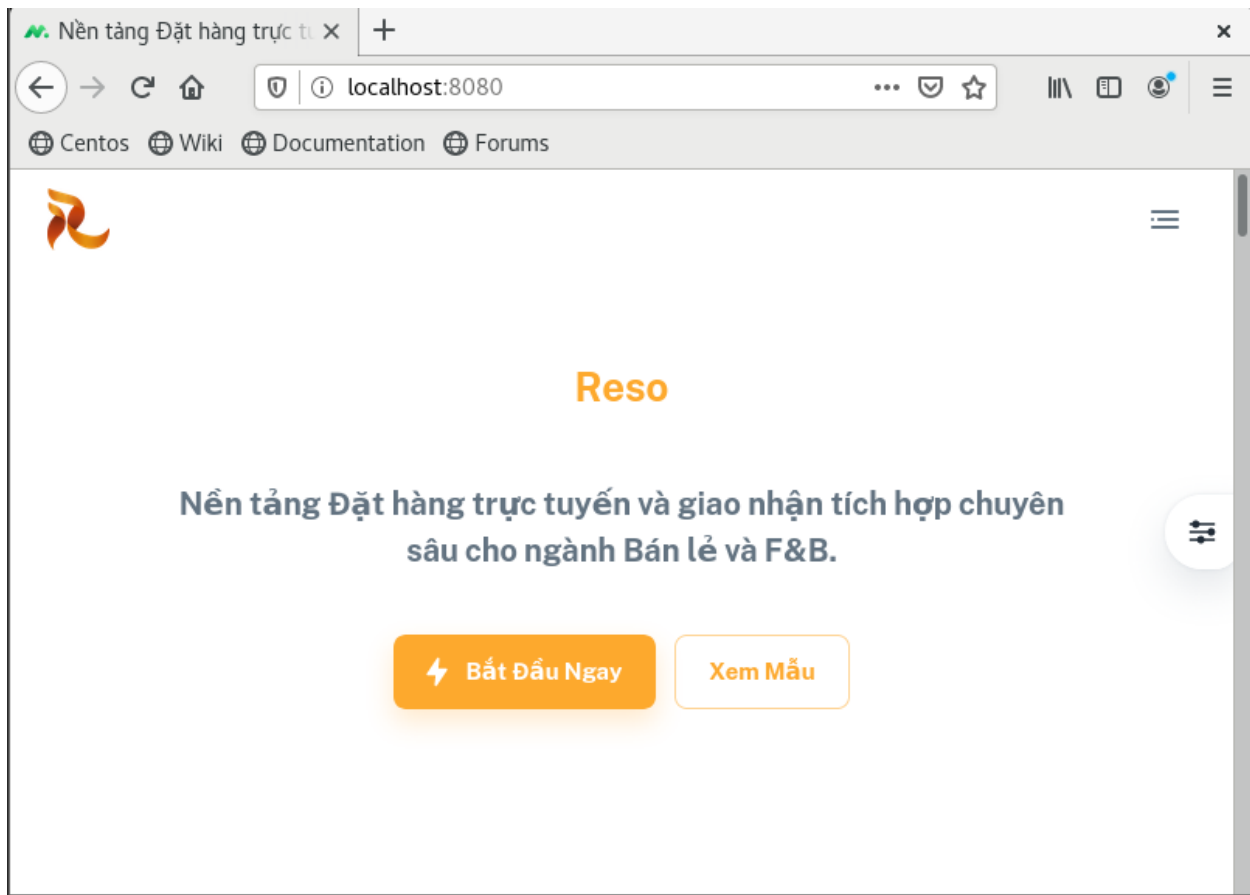
```
#Listen 12.34.56.78:80
Listen 8080
```

- Restart lại http:

Ta dùng lệnh: **systemctl restart httpd**

- Kiểm tra đã thành công

Ta vào firefox và gõ: **localhost:port (port lúc config)**



- Run ngrok
 - Ta dùng lệnh
Ngrok http 8080
Nó sẽ cho ra 2 đường dẫn http và https cho bạn

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Session Expires    1 hour, 58 minutes
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://ccbf-113-172-177-31.ngrok.io -> http://localhost:8
Forwarding           https://ccbf-113-172-177-31.ngrok.io -> http://localhost:

Connections         ttl    opn    rt1    rt5    p50    p90
                   24     0      0.09   0.06   5.01   10.13

HTTP Requests
```

NGINX Server

- NGINX là gì?

NGINX, đọc là “engine-ex” là một phần mềm web server mã nguồn mở có thể hoạt động như là email proxy, reverse proxy và load balancer. Cấu trúc của phần mềm này là bất đồng bộ và hướng sự kiện; vì vậy cho phép phần mềm xử lý nhiều truy vấn cùng lúc. NGINX dễ dàng để mở rộng cho website hơn, đồng nghĩa với việc dịch vụ này có thể đi theo suốt qua trình phát triển của website, cũng như traffic web.

- NGINX server hoạt động như thế nào?

NGINX hoạt động theo kiến trúc bất đồng bộ (asynchronous), hướng sự kiện (event driven). Kiến trúc này có thể hiểu là những threads tương đồng nhau sẽ được quản lý trong một tiến trình (process), và mỗi tiến trình hoạt động chứa các thực thể nhỏ hơn gọi là worker connections. Cả bộ đơn vị này chịu trách nhiệm xử lý các threads.

Worker connections sẽ gửi các truy vấn cho một worker process, worker process sẽ gửi nó tới process cha (master process). Cuối cùng, master process sẽ trả kết quả cho những yêu cầu đó.

- So sánh NGINX Server vs Apache Server

1. Hệ điều hành hỗ trợ

Cả NGINX và Apache đều có thể chạy trên nhiều hệ điều hành khác nhau của hệ thống Unix. Nhưng hiệu năng của NGINX trên Windows lại tỏ ra kém hiệu quả hơn khi hoạt động trên các platform khác.

2. Hỗ trợ người dùng

Mặc dù cả NGINX và Apache đều có hệ thống mailing hỗ trợ và diễn đàn Stack Overflow, nhưng Apache lại thiếu hỗ trợ từ chính công ty của nó.

3. Hiệu năng

NGINX xử lý cùng lúc 1000 kết nối tới nội dung tĩnh nhanh hơn 2 lần so với Apache và dùng ít bộ nhớ hơn. Khi so về hiệu năng trên nội

dung động, cả 2 nền tảng cho tốc độ giống nhau. NGINX là lựa chọn tốt hơn cho những ai có website tĩnh nhiều hơn

- Install and Configure Nginx Server in CentOS

- Install:

- Để cài đặt Nginx ta dùng lệnh

- `sudo dnf install nginx`

```
[root@localhost hoangvo]# sudo dnf install nginx
Last metadata expiration check: 0:31:03 ago on Fri 27 Aug 2021 04:55:05 AM EDT.
Dependencies resolved.
=====
Package                Arch    Version                                Repo                Size
=====
Installing:
  nginx                 x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream            570 k
Installing dependencies:
  nginx-all-modules    noarch  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             23 k
  nginx-filesystem     noarch  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             24 k
  nginx-mod-http-image-filter x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             35 k
  nginx-mod-http-perl   x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             45 k
  nginx-mod-http-xslt-filter x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             33 k
  nginx-mod-mail        x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             64 k
  nginx-mod-stream      x86_64  1:1.14.1-9.module_el8.0.0+184+e34fea82 appstream             85 k
Enabling module streams:
  nginx                 1.14
```

- Configure:

- Mở nginx ta dùng lệnh

- `sudo systemctl enable nginx`

- `sudo systemctl start nginx`


```
[root@localhost hoangvo]# sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[root@localhost hoangvo]# sudo systemctl start nginx
Job for nginx.service failed because the control process exited with error code.
See "systemctl status nginx.service" and "journalctl -xe" for details.
[root@localhost hoangvo]# sudo systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor prese>
   Active: failed (Result: exit-code) since Fri 2021-08-27 05:31:50 EDT; 43s ago
   Process: 39006 ExecStart=/usr/sbin/nginx (code=exited, status=1/FAILURE)
   Process: 39004 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
   Process: 39003 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, statu>

Aug 27 05:31:48 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to 0.>
Aug 27 05:31:48 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to [:>
Aug 27 05:31:49 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to 0.>
Aug 27 05:31:49 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to [:>
Aug 27 05:31:49 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to 0.>
Aug 27 05:31:49 localhost.localdomain nginx[39006]: nginx: [emerg] bind() to [:>
Aug 27 05:31:50 localhost.localdomain nginx[39006]: nginx: [emerg] still could >
Aug 27 05:31:50 localhost.localdomain systemd[1]: nginx.service: Control proces>
Aug 27 05:31:50 localhost.localdomain systemd[1]: nginx.service: Failed with re>
Aug 27 05:31:50 localhost.localdomain systemd[1]: Failed to start The nginx HTT>
```

(nếu bị lỗi này đây là cách fix)

sudo fuser -k 80/tcp

sudo fuser -k 443/tcp

sudo service nginx restart

Và chạy lại: **sudo systemctl start nginx**

```
[root@localhost hoangvo]# sudo fuser -k 80/tcp
80/tcp:          36965 36967 36968 36969
[root@localhost hoangvo]# sudo fuser -k 443/tcp
[root@localhost hoangvo]# sudo service nginx restart
Redirecting to /bin/systemctl restart nginx.service
[root@localhost hoangvo]# sudo systemctl start nginx
[root@localhost hoangvo]# sudo systemctl status nginx.service
● nginx.service - The nginx HTTP and reverse proxy ser>
   Loaded: loaded (/usr/lib/systemd/system/nginx.servi>
   Active: active (running) since Fri 2021-08-27 05:44>
   Process: 39686 ExecStart=/usr/sbin/nginx (code=exite>
   Process: 39683 ExecStartPre=/usr/sbin/nginx -t (code>
   Process: 39682 ExecStartPre=/usr/bin/rm -f /run/ngin>
   Main PID: 39689 (nginx)
     Tasks: 2 (limit: 4621)
    Memory: 3.5M
    CGroup: /system.slice/nginx.service
            └─39689 nginx: master process /usr/sbin/ngi>
               └─39690 nginx: worker process
```

- Điều chỉnh Firewall rules
 - `sudo firewall-cmd --permanent --add-service=http`
- Để xác minh rằng dịch vụ tường lửa http đã được thêm đúng cách
 - `sudo firewall-cmd --permanent --list-all`

```
[root@localhost hoangvo]# sudo firewall-cmd --permanent --add-service=http
success
[root@localhost hoangvo]# sudo firewall-cmd --permanent --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost hoangvo]#
```

- Kiểm tra cài đặt thành công ta lên web browser nhập địa chỉ ip máy



- Deploy REACT JS In NGINX

Tạo một file config cho ứng dụng của bạn

\$ **sudo nano /etc/nginx/sites-available/demo.conf**

```
[hoangvo@localhost ~]$ sudo su
[sudo] password for hoangvo:
[root@localhost hoangvo]# nano /etc/nginx/sites-available/demo.conf
```

Sau đó thêm đoạn code thiết lập dưới đây vào file vừa được tạo và lưu lại

```
server {
    listen 3000;
    server_name react;
    root /usr/share/nginx/build/;
    index index.html index.htm;

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

ở bước này giả định bạn đã chạy lệnh build ứng dụng tại thư mục build trong thư mục dự án

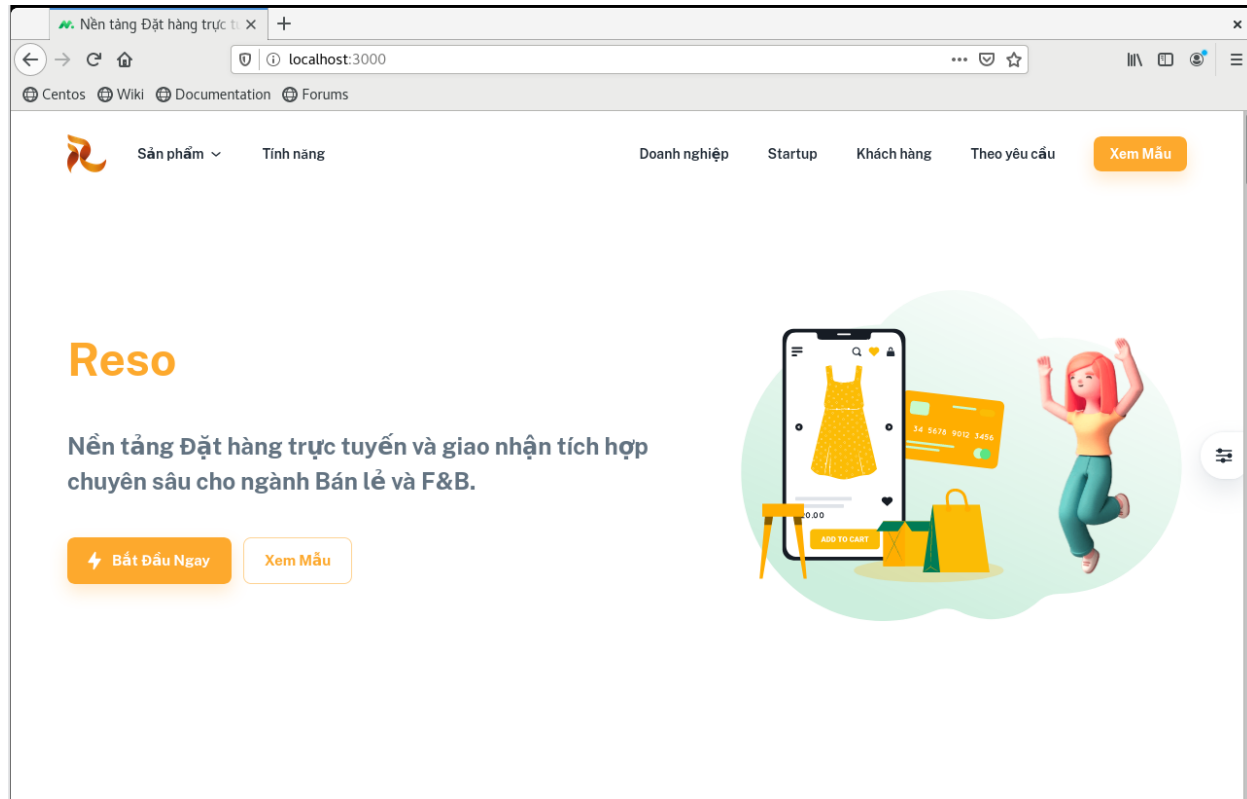
Sau đó dùng lệnh

Nginx -t (để test nginx)

```
[root@localhost sites-available]# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Và chạy lại nginx: **nginx -s reload**

nếu mọi thứ đều trơn tru thì ứng dụng của bạn đã sẵn sàng rồi nhé,
kiểm tra trên đường dẫn: <http://localhost:3000>



- Dùng ngrok để public trên internet

- Cài ngrok

- Trước tiên ta phải cài snap

- <https://snapcraft.io/install/ngrok/centos>

- Cài ngrok:

- `sudo snap install ngrok`

- ```
[root@localhost sites-available]# snap install ngrok
Download snap "snapd" (12883) from channel "stable" 72% 499kB/s 19.0s
```

- Run ngrok

- Ta dùng lệnh

- Ngrok http 3000

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status online
Session Expires 1 hour, 59 minutes
Version 2.3.40
Region United States (us)
Web Interface http://127.0.0.1:4040
Forwarding http://a4ac-113-173-102-97.ngrok.io -> htt
Forwarding https://a4ac-113-173-102-97.ngrok.io -> ht
Connections ttl opn rt1 rt5 p50 p9
 0 0 0.00 0.00 0.00 0.
```

Đây là 2 link mà bạn có thể copy và chạy trên browser và thấy được app của bạn

## IIS Web Server

- IIS là gì?

IIS là viết tắt của từ (Internet Information Services ), là các dịch vụ dành cho máy chủ chạy trên nền hệ điều hành Window nhằm cung cấp và phân tán các thông tin lên mạng, nó bao gồm nhiều dịch vụ khác nhau như Web Server, FTP Server...

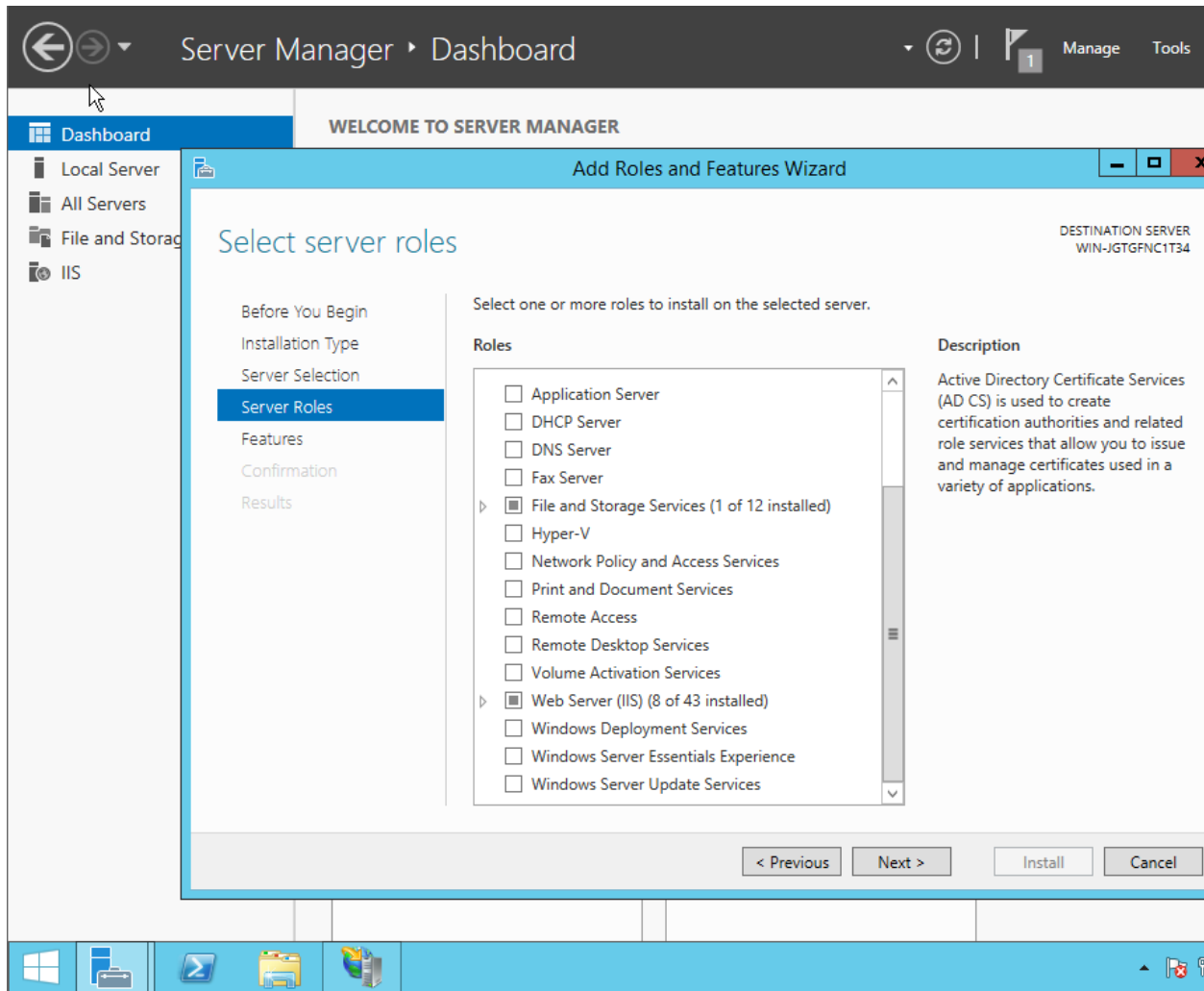
- Các tính năng của IIS

- Một trong những tính năng được sử dụng nhiều nhất của IIS là tạo một ứng dụng web bằng ASP.NET. Bên cạnh đó, IIS hoàn toàn có thể chạy được với các trang web viết bằng các ngôn ngữ khác như PHP, Perl,...
- IIS hỗ trợ một số loại xác thực như Basic access authentication, Digest access authentication, Windows Authentication, Certificate authentication,...; các tính năng bảo mật khác bao gồm hỗ trợ SSL/TLS, Server Name Indication (SNI), thiết lập bảo mật cho máy chủ FTP,...

- Install IIS and Deploy a Web App in IIS

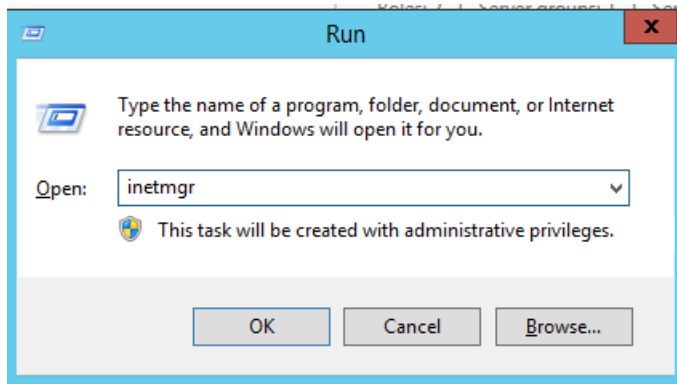
- Install:

- Ở Win server vào server manager chọn add roles and features -> chọn server roles -> chọn web server -> install

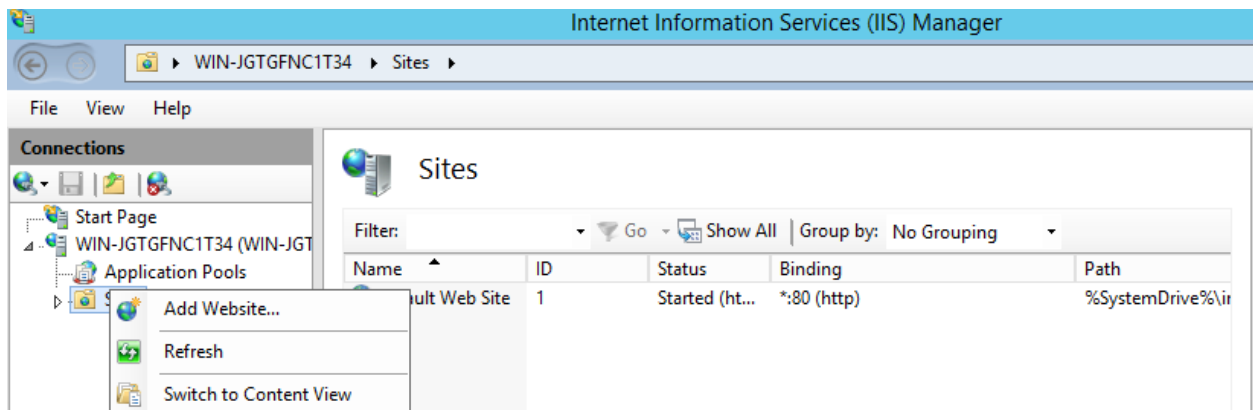


- Deploy web app

- Đầu tiên ta vào Start chọn Run và tìm kiếm inetmgr



- Mở rộng localComputer. Nhấp chuột phải vào Sites. Nhấp vào Add Website như hình dưới đây



- Nhập tên của trang web. Trong trường hợp của mình, đó là "MyWeb". Nhập vị trí của ứng dụng web hoặc trang web. Cung cấp port. Và sau đó nhấn OK (nếu port đã có hãy đổi port khác)

**Add Website**

Site name:  Application pool:

Content Directory

Physical path:

Pass-through authentication

Binding

Type:  IP address:  Port:

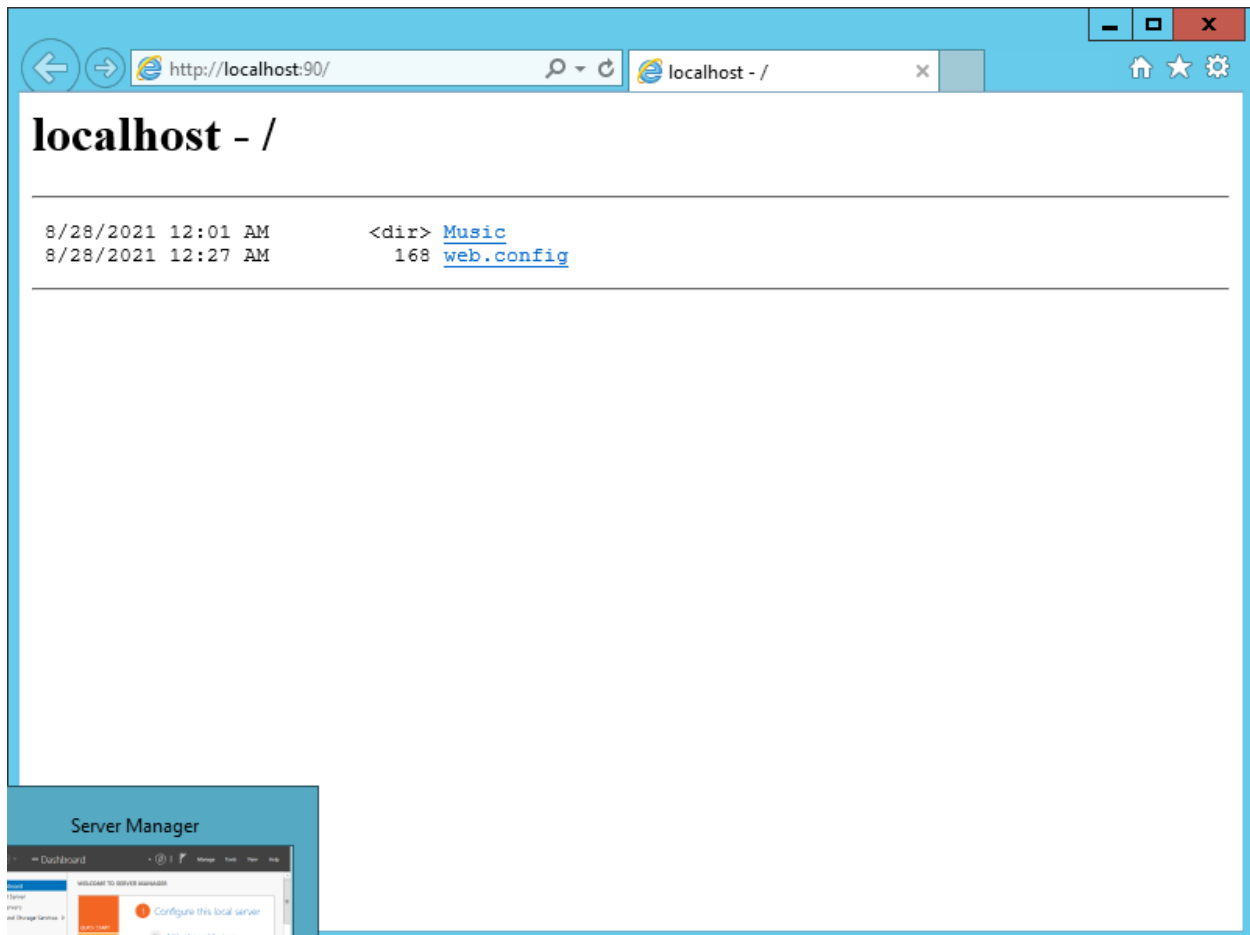
Host name:

Example: www.contoso.com or marketing.contoso.com

☒ Start Website immediately

- Nhấp chuột phải vào tên trang web (trong trường hợp của tôi là "MyWeb"), nhấp vào "Manage website" và "Browse"





## Web Application Firewall

- WAF là gì?

WAF (Web Application Firewall) còn gọi là tường lửa ứng dụng web. WAF là một thiết bị proxy có thể xử lý giao thức HTTP nhằm bảo vệ ứng dụng web. WAF kiểm tra lượng truy cập và sẽ lọc ra các yêu cầu có mối đe dọa xâm hại đến website trước khi đến ứng dụng web.

- Tại sao cần tường lửa ứng dụng web?

WAF bảo vệ khỏi các cuộc tấn công độc hại, chẳng hạn như:

1. SQL Injection : một công nghệ hack được sử dụng để trích xuất thông tin nhạy cảm từ database.

2. Remote Code Execution: một kỹ thuật tấn công cho phép một người thực thi code từ xa sau khi user chấp nhận file độc hại.
3. Cross-site scripting: khi script độc hại được đưa vào code của một trang web đáng tin cậy khác, sẽ cho phép dữ liệu nhạy cảm của user như cookie bị truy cập.

Những mối đe dọa này có thể xâm nhập và làm tê liệt trang web, làm giảm hiệu suất và khiến doanh nghiệp của bạn bị mất dữ liệu.

- **Hoạt động của tường lửa ứng dụng web**

- WAF được triển khai trước các ứng dụng web và phân tích lưu lượng HTTP, kiểm tra cả request GET và POST nhằm phát hiện và chặn bất kỳ thứ gì độc hại.
- Không giống như tường lửa (Firewall) thông thường chỉ đóng vai trò như một cổng an toàn giữa các server, WAF là một biện pháp bảo mật ứng dụng được đặt giữa Web Client và Web Server.
- WAF thực hiện kiểm tra chi tiết mọi request và response đối với tất cả các dạng lưu lượng truy cập web phổ biến. Việc kiểm tra này giúp WAF xác định và chặn các mối đe dọa, ngăn chúng xâm nhập vào server.

- **Lợi ích của Web App Firewall**

1. Tổng chi phí sở hữu (TCO) thấp.
2. Bảo mật toàn diện cho trang web
3. Chi phí băng thông (bandwidth) hiệu quả

- **Các loại tường lửa sử dụng trong ứng dụng web**

1. Network-based WAFs (WAF dựa trên mạng)
2. Host-based WAFs (WAF dựa trên máy chủ)
3. Cloud – hosted WAFs (WAF được lưu trữ trên đám mây)

## **Firewall**

- **Tường lửa Firewall là gì?**

Tường lửa (Firewall) là một hệ thống an ninh mạng, có thể dựa trên phần cứng hoặc phần mềm, sử dụng các quy tắc để kiểm soát traffic vào, ra khỏi hệ thống. Tường lửa hoạt động như một rào chắn giữa mạng an toàn và mạng không an toàn. Nó kiểm soát các truy cập đến nguồn lực của mạng thông qua một mô hình kiểm soát chủ động.

- **Firewall hoạt động như thế nào?**

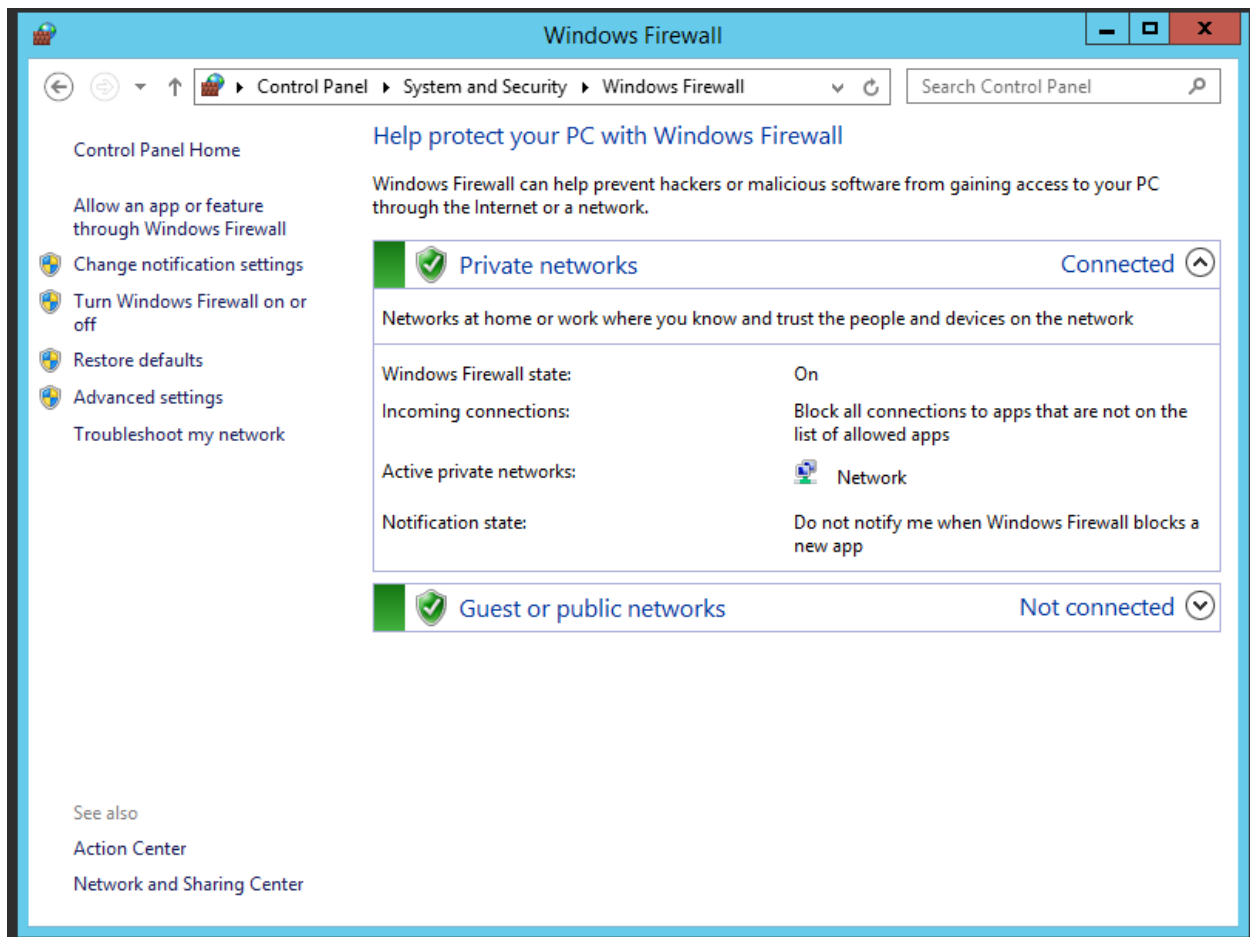
Firewall giúp kiểm soát luồng thông tin giữa Intranet và Internet, chúng phát hiện và phân xét những hành vi được truy cập và không được truy cập vào bên trong hệ thống, đảm bảo tối đa sự an toàn thông tin.

Một firewall cần biết được sự khác biệt giữa lưu lượng hợp pháp như trên với những loại dữ liệu gây hại khác.

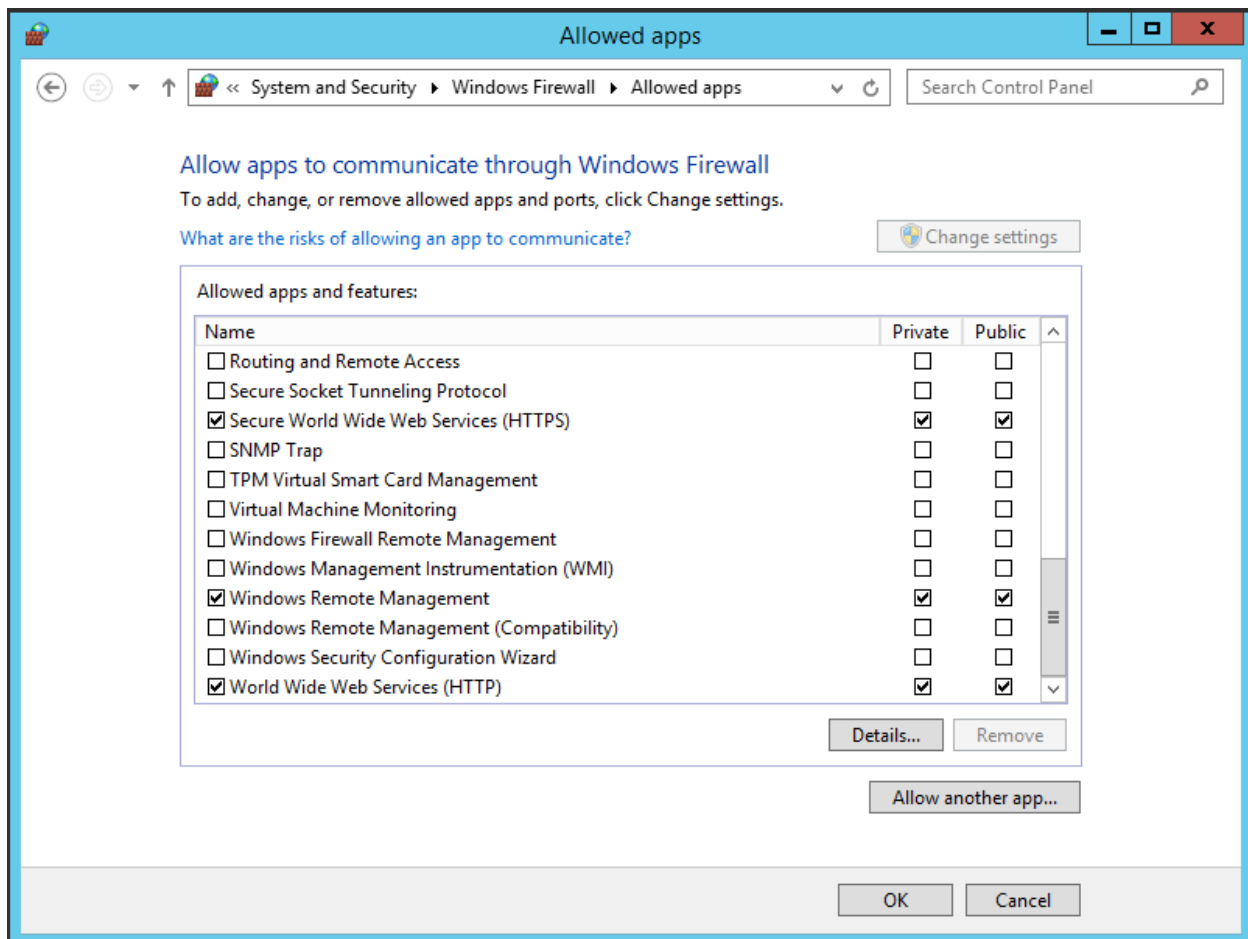
Firewall sử dụng rule hoặc ngoại lệ để làm việc với những kết nối tốt và loại bỏ những kết nối xấu. Nhìn chung, quá trình này được thực hiện ẩn, người dùng không thấy được hoặc không cần tương tác gì cả.

- **Enable Firewall cho web server**

- vào Start → Control Panel → System and Security → Windows Firewall. Khi có cửa sổ xuất hiện, kích vào đường link Allow a program or feature through Windows Firewall trong danh sách bên trái để xem những phần mềm được phép giao tiếp qua firewall.



- Nếu World Wide Web Services (HTTP) không được chọn, vui lòng Kiểm tra World Wide Web Services (HTTP) và nhấp vào OK.



## SSL

- SSL là gì?

SSL là viết tắt của từ Secure Sockets Layer. SSL là tiêu chuẩn của công nghệ bảo mật, truyền thông mã hoá giữa máy chủ Web server và trình duyệt. Tiêu chuẩn này hoạt động và đảm bảo rằng các dữ liệu truyền tải giữa máy chủ và trình duyệt của người dùng đều riêng tư và toàn vẹn. SSL hiện tại cũng là tiêu chuẩn bảo mật cho hàng triệu website trên toàn thế giới, nó bảo vệ dữ liệu truyền đi trên môi trường internet được an toàn.

SSL đảm bảo rằng tất cả các dữ liệu được truyền giữa các máy chủ web và các trình duyệt được mang tính riêng tư, tách rời.

- **Tại sao nên sử dụng SSL?**

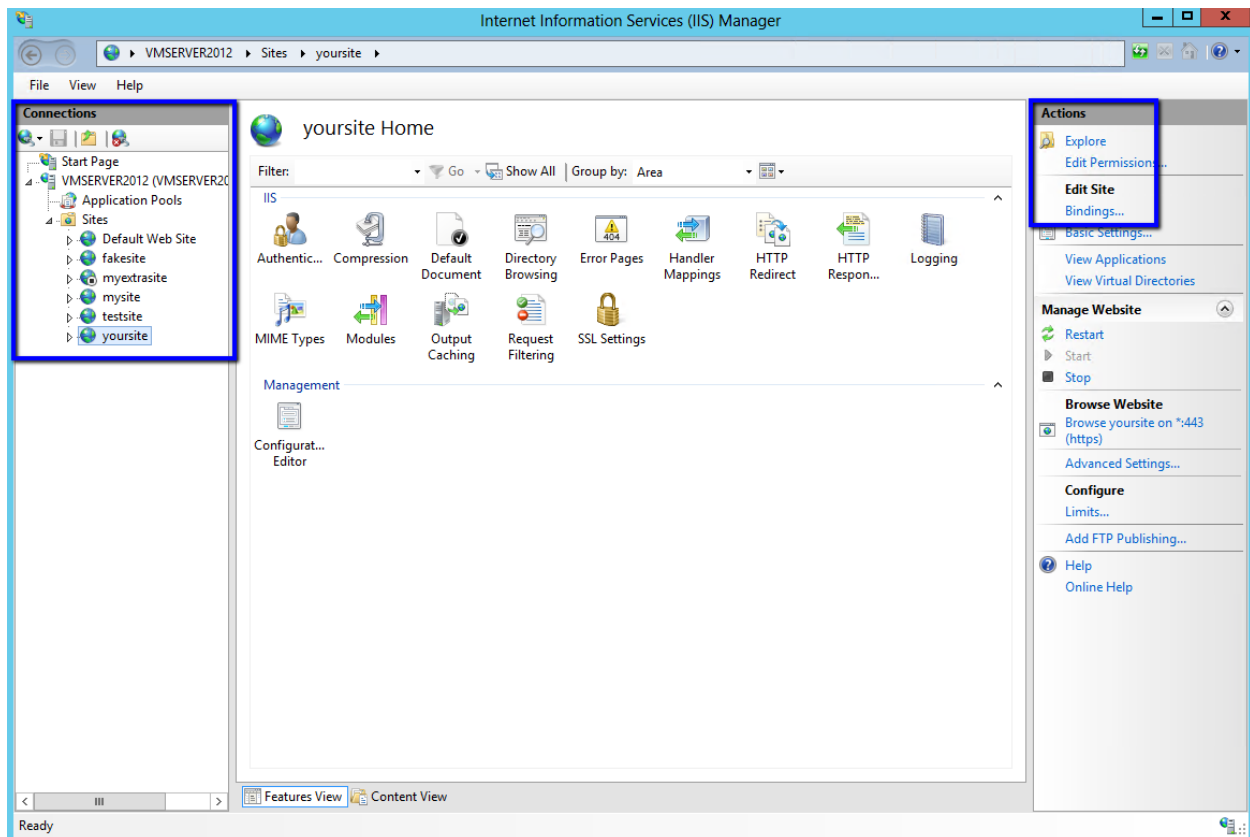
Khi bạn đăng ký tên miền để sử dụng các dịch vụ website, email v.v... luôn có những lỗ hổng bảo mật cho hacker tấn công, SSL bảo vệ website và khách hàng của bạn.

1. An toàn dữ liệu: dữ liệu không bị thay đổi bởi hacker.
2. Bảo mật dữ liệu: dữ liệu được mã hóa và chỉ người nhận đích thực mới có thể giải mã.
3. Chống chối bỏ: đối tượng thực hiện gửi dữ liệu không thể phủ nhận dữ liệu của mình.

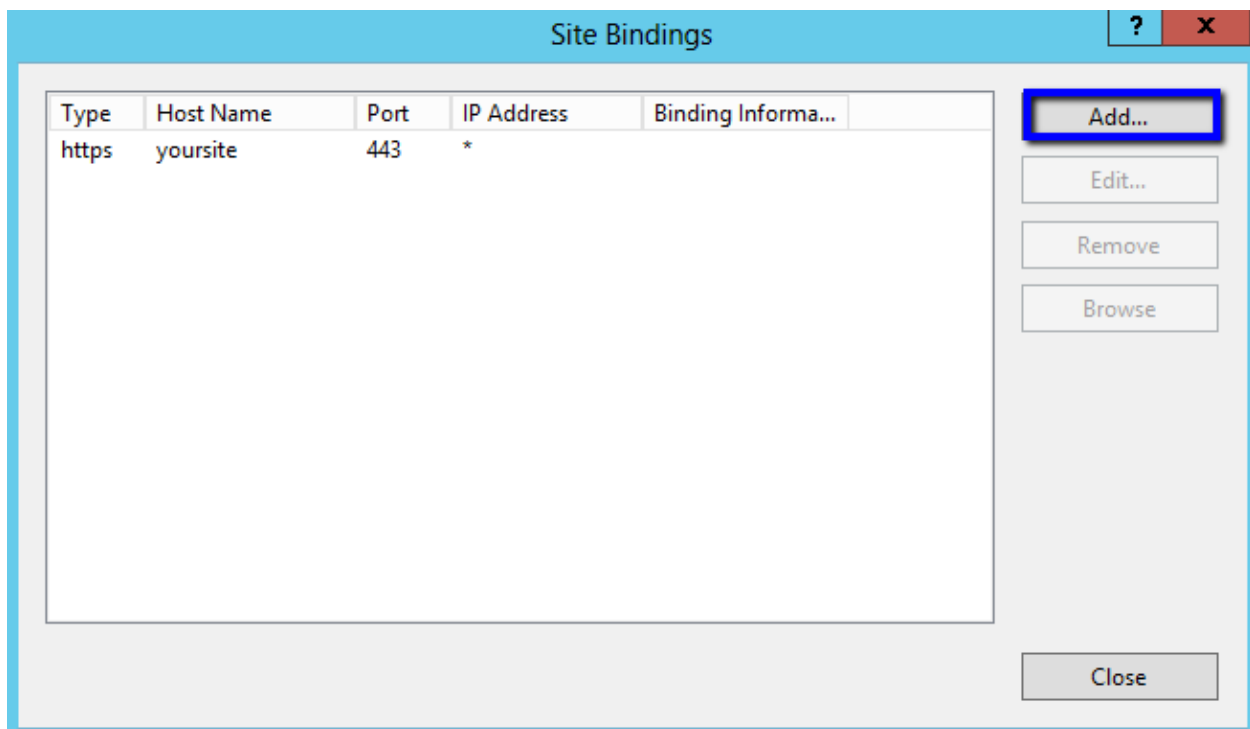
Tiêu chuẩn xác thực – SSL chỉ được cung cấp bởi các đơn vị cấp phát chứng thư (CA) có uy tín trên toàn thế giới sau khi đã thực hiện xác minh thông tin về chủ thể đăng ký rất kỹ càng mang lại mức độ tin cậy cao cho người dùng Internet và tạo nên giá trị cho các website, doanh nghiệp cung cấp dịch vụ

- **Enabling SSL in IIS**

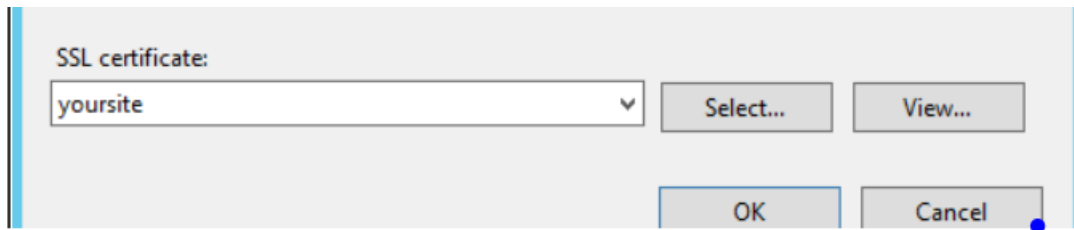
- Từ màn hình Start, nhấp và nhấp vào Internet Information Services (IIS).Hoặc vào Start chọn Run và tìm kiếm inetmgr
- Ở Internet Information Services (IIS) Manager, dưới Connections, expand Server, expand Sites, và chọn website mà bạn muốn cài SSL Certificate.
- Ở Actions menu phía bên phải, dưới Edit Site, chọn Bindings



- Ở cửa sổ Site Bindings, chọn Add



- Ở SSL Certificate chọn certificate mà bạn muốn imported SSL Certificate



- Chọn Ok

## Monitor web server

- Server Monitoring là gì?

Server Monitoring là Giám sát máy chủ. Là quá trình xem xét và phân tích một máy chủ cho tính sẵn sàng, hoạt động, hiệu suất, an ninh và các quá trình hoạt động khác .

- Một số Open Source Monitoring Server

<https://geekflare.com/best-open-source-monitoring-software/>

<https://devopscube.com/best-opensource-monitoring-tools/>

- Tại sao cần Server Monitoring?

- Phân tích trong thời gian thực

Một hệ thống tốt sẽ cung cấp việc giám sát liên tục, không có độ trễ, hoặc độ trễ tối thiểu.

- Hệ thống cảnh báo

Khi một sự cố cụ thể xảy ra, một thông báo được tạo và được gửi đến đúng người. Thông báo có thể được cấu hình các điều kiện kích hoạt (ví dụ: đĩa cứng đạt dung lượng tối đa).

- Thông báo bằng nhiều cách khác nhau (email, SMS,...) đến đúng người.

- Trực quan hóa đồ họa



Để thuận tiện cho việc phân tích dữ liệu, công cụ giám sát sẽ tạo ra các biểu đồ về dữ liệu khá thân thiện và dễ hiểu.

- Sản xuất báo cáo

Hệ thống giám sát cho phép bạn tạo báo cáo, các dữ liệu phục vụ cho hoạt động của doanh nghiệp.

- Có sẵn bản ghi

Tạo một bản ghi của các giám sát trước sẽ giúp bạn so sánh hiệu quả hoạt động của hệ thống. Giúp doanh nghiệp dễ dàng đưa ra các biện pháp trong tương lai, chẳng hạn như mua phần cứng mới.

- Khả năng cài đặt các plug-in

Cài đặt thêm các plug-in, ad hoc nhằm đáp ứng tối đa các nhu cầu của doanh nghiệp.

- Phân biệt theo loại người dùng

Dữ liệu được truy cập bởi mỗi người dùng sẽ khác nhau tùy thuộc vào các quyền có sẵn.

- Deploy monitor web server Zabbix

- Set SELinux to permissive mode

- Ta dùng lệnh:

- ```
setenforce 0 && sed -i 's/^SELINUX=.*SELINUX=permissive/g' /etc/selinux/config
```

- Install Zabbix server, frontend, and agent

- rpm -Uvh

- https://repo.zabbix.com/zabbix/5.4/rhel/8/x86_64/zabbix-release-5.4-1.el8.noarch.rpm

```
[root@localhost hoangvo]# rpm -Uvh https://repo.zabbix.com/zabbix/5.4/rhel/8/x86_64/zabbix-release-5.4-1.el8.noarch.rpm
Retrieving https://repo.zabbix.com/zabbix/5.4/rhel/8/x86_64/zabbix-release-5.4-1.el8.noarch.rpm
warning: /var/tmp/rpm-tmp.kb81KA: Header V4 RSA/SHA512 Signature, key ID a14fe591: NOKEY
Verifying... ##### [100%]
Preparing... ##### [100%]
Updating / installing...
 1:zabbix-release-5.4-1.el8 ##### [100%]
```

- `dnf -y install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-sql-scripts zabbix-agent`

```
[root@localhost hoangvo]# dnf -y install zabbix-server-mysql zabbix-web-mysql zabbix-apache-conf zabbix-sql-scripts zabbix-agent
CentOS Linux 8 - AppStream 3.3 kB/s | 4.3 kB 00:01
CentOS Linux 8 - AppStream 4.0 MB/s | 8.8 MB 00:02
CentOS Linux 8 - BaseOS 4.2 kB/s | 3.9 kB 00:00
CentOS Linux 8 - BaseOS 3.1 MB/s | 6.5 MB 00:02
CentOS Linux 8 - Extras 2.4 kB/s | 1.5 kB 00:00
Extra Packages for Enterprise Linux Modular 8 - 10 kB/s | 9.4 kB 00:00
Extra Packages for Enterprise Linux 8 - x86_64 8.8 kB/s | 9.4 kB 00:01
Extra Packages for Enterprise Linux 8 - x86_64 196 kB/s | 10 MB 00:53
Node.js Packages for Enterprise Linux 8 - x86_64 42 B/s | 2.5 kB 01:00
Zabbix Official Repository - x86_64 28 kB/s | 65 kB 00:02
Zabbix Official Repository non-supported - x86_64 1.2 kB 00:01
Dependencies resolved.

=====
Package Arch Version Repo Size
=====
Installing:
zabbix-agent x86_64 5.4.4-1.el8 zabbix 497 k
zabbix-apache-conf noarch 5.4.4-1.el8 zabbix 20 k
zabbix-server-mysql
```

○ Install and configure database

- a. Install MariaDB

`dnf -y install mariadb-server && systemctl start mariadb && systemctl enable mariadb`

```
[root@localhost hoangvo]# dnf -y install mariadb-server && systemctl start maria
db && systemctl enable mariadb
Last metadata expiration check: 0:02:39 ago on Tue 14 Sep 2021 02:06:50 AM EDT.
Dependencies resolved.
=====
Package                Arch    Version                                Repo                Size
=====
Installing:
mariadb-server          x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream            16 M
Installing dependencies:
mariadb                 x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream             6.0 M
mariadb-common           x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream             64 k
mariadb-errmsg            x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream            234 k
perl-DBD-MySQL           x86_64  4.046-3.module_el8.1.0+203+e45423dc    appstream            156 k
Installing weak dependencies:
mariadb-backup           x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream             6.1 M
mariadb-gssapi-server    x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream             51 k
mariadb-server-utils     x86_64  3:10.3.28-1.module_el8.3.0+757+d382997d appstream             1.1 M
Enabling module streams:
```

- b. Reset root password for database

[mysql_secure_installation](#)

nó hiện ra và setup:

Enter current password for root (enter for none): Press the Enter

Set root password? [Y/n]: Y

New password: <Enter root DB password>

Re-enter new password: <Repeat root DB password>

Remove anonymous users? [Y/n]: Y

Disallow root login remotely? [Y/n]: Y

Remove test database and access to it? [Y/n]: Y

Reload privilege tables now? [Y/n]: Y

- c. Create database

[mysql -uroot -p'your root DB password' -e "create database zabbix character set utf8 collate utf8_bin;"](#)

[mysql -uroot -p' your root DB password ' -e "grant all privileges on zabbix.* to zabbix@localhost identified by 'zabbixDBpass';"](#)

- d. Import initial schema and data

[mysql -uroot -p'rootDBpass' zabbix -e "set global innodb_strict_mode='OFF';"](#)

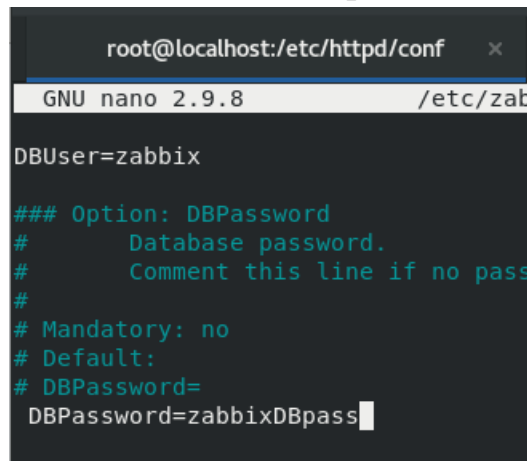
Import database shema for Zabbix server

```
zcat /usr/share/doc/zabbix-sql-scripts/mysql/create.sql.gz |  
mysql -uzabbix -p'zabbixDBpass' zabbix
```

Enable strict mode

```
mysql -uroot -p'rootDBpass' zabbix -e "set global  
innodb_strict_mode='ON';"
```

- e. Enter database password in Zabbix configuration file
Mở: `nano /etc/zabbix/zabbix_server.conf`
Và thêm vào database pass: `DBPassword=zabbixDBpass`

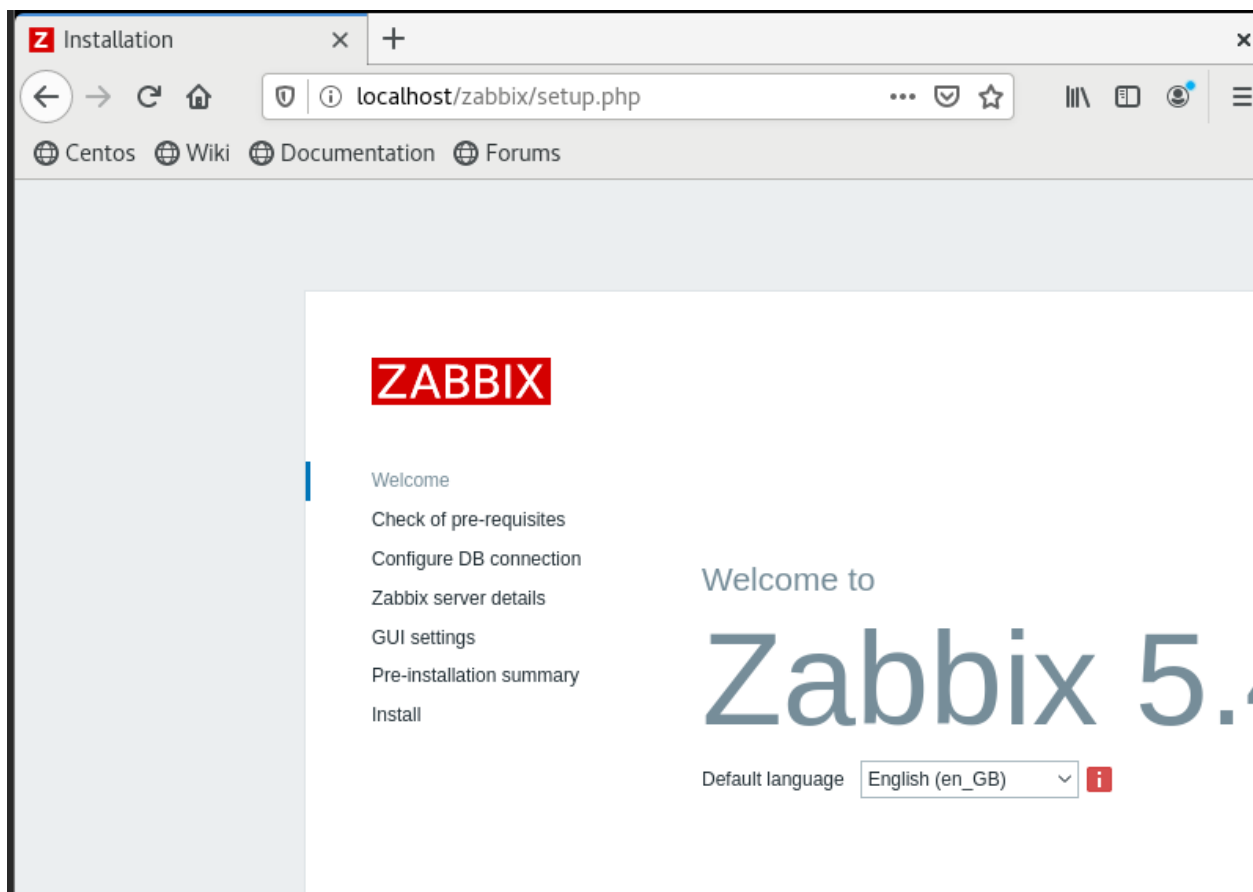


```
root@localhost:/etc/httpd/conf x  
GNU nano 2.9.8 /etc/zab  
DBUser=zabbix  
### Option: DBPassword  
# Database password.  
# Comment this line if no pass  
#  
# Mandatory: no  
# Default:  
# DBPassword=  
DBPassword=zabbixDBpass
```

- Start Zabbix server and agent processes
 - Ta dùng lệnh:
`systemctl restart zabbix-server zabbix-agent`
`systemctl enable zabbix-server zabbix-agent`
- Configure Zabbix frontend
 - a. Configure PHP for Zabbix frontend
`nano /etc/php-fpm.d/zabbix.conf`
xóa dấu “;” ở “; php_value date.timezone Europe/Riga”
và sửa đúng vị trí của bạn ví dụ:
`php_value[date.timezone] = Asia/Ho_Chi_Minh`
 - b. Restart Apache web server and make it start at system boot
`systemctl restart httpd php-fpm`
`systemctl enable httpd php-fpm`

➤ c. Configure web frontend

Vào browser: <http://localhost/zabbix>



Next step tới Configure DB connection



- Welcome
- Check of pre-requisites
- Configure DB connection
- Zabbix server details
- GUI settings
- Pre-installation summary
- Install

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type	<input type="text" value="MySQL"/>
Database host	<input type="text" value="localhost"/>
Database port	<input type="text" value="0"/> 0 - use default port
Database name	<input type="text" value="zabbix"/>
Store credentials in	<input checked="" type="radio"/> Plain text <input type="radio"/> HashiCorp Vault
User	<input type="text" value="zabbix"/>
Password	<input type="password" value="....."/>

Database TLS encryption *Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).*

[Back](#)[Next step](#)

Type your db password and next step



- Welcome
- Check of pre-requisites
- Configure DB connection
- Zabbix server details
- GUI settings
- Pre-installation summary
- Install

Zabbix server details

Please enter the host name or host IP address and port number of the Zabbix server, as well as the name of the installation (optional).

Host	<input type="text" value="localhost"/>
Port	<input type="text" value="10051"/>
Name	<input type="text" value="192.168.10.138"/>

[Back](#)[Next step](#)

Next step:

Võ Minh Hoàng

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

GUI settings

Pre-installation summary

Install

GUI settings

Default time zone

System: (UTC+07:00) Asia/Ho_Chi_Minh

▼

Default theme

Blue

▼

Back

Next step

Next step:

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Zabbix server details

GUI settings

Pre-installation summary

Install

Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.

Database type

MySQL

Database server

localhost

Database port

default

Database name

zabbix

Database user

zabbix

Database password

Database TLS encryption

false

Zabbix server

localhost

Zabbix server port

10051

Zabbix server name

192.168.10.138

Back

Next step

And install:



Install

Welcome
Check of pre-requisites
Configure DB connection
Zabbix server details
GUI settings
Pre-installation summary
Install

Congratulations! You have successfully installed Zabbix frontend.

Configuration file "/etc/zabbix/web/zabbix.conf.php" created.

Back

Finish



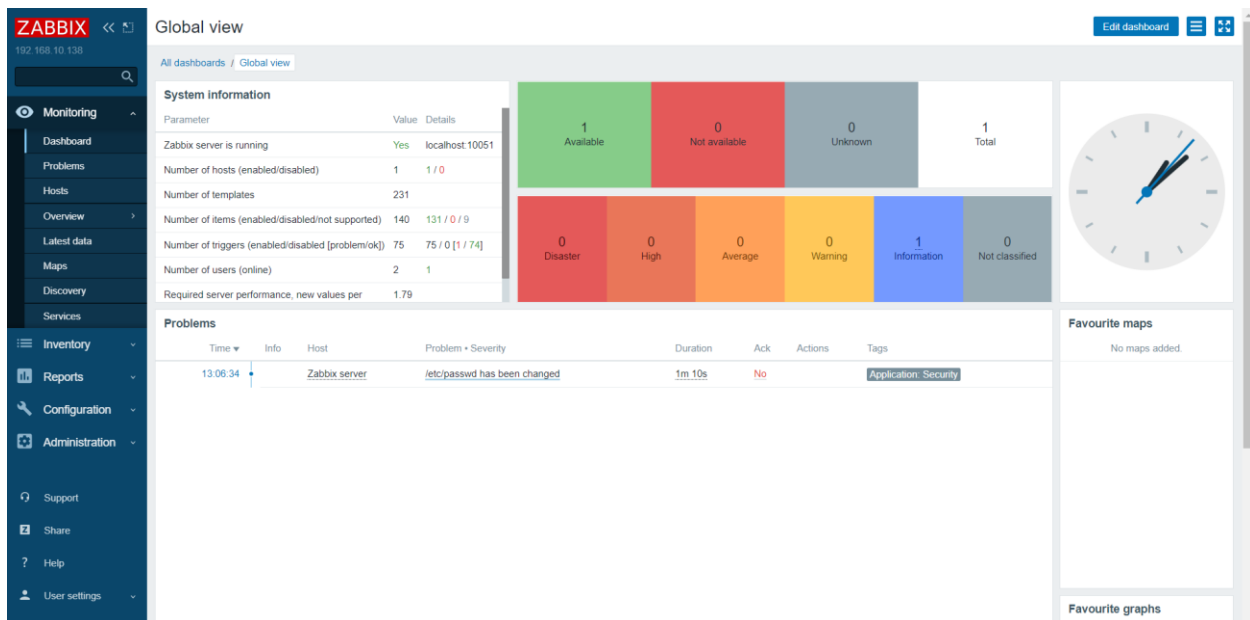
Username

Password

☒ Remember me for 30 days

Sign in

Use Zabbix default admin username “Admin” and password “zabbix”



- Deploy monitor web server Prometheus
 - Install Prometheus
 - Add system user and group for Prometheus

```
sudo groupadd --system prometheus
sudo useradd -s /sbin/nologin --system -g prometheus prometheus
```

- Set NTP Server

- Set timezone

```
sudo timedatectl set-timezone Asia/Ho_Chi_Minh
```

- Install chrony and configure it as NTP client.

```
sudo yum -y install chrony
```

- Set NTP synchronization.

```
sudo timedatectl set-ntp true
```

- Start and enable the service.

```
sudo systemctl enable --now chronyd
```

- Create data directory for Prometheus

```
sudo mkdir /var/lib/prometheus
```

- Create configuration directories for Prometheus

```
for i in rules rules.d files_sd; do
    sudo mkdir -p /etc/prometheus/${i};
done
```

- Download Prometheus

```
[hoangvo@localhost ~]$ curl -s https://api.github.com/repos/prometheus/prometheus/releases/latest \
> | grep browser_download_url \
> | grep linux-amd64 \
> | cut -d '"' -f 4 \
> | wget -qi -
```

Extract the file and move it to directory in your \$PATH

```
tar xvf prometheus-*.tar.gz
cd prometheus-*/
sudo cp prometheus promtool /usr/local/bin/
```

Also copy consoles and console_libraries to /etc/prometheus directory

```
sudo cp -r consoles/ console_libraries/ /etc/prometheus/
```

- Create a Prometheus configuration file

➤ Use: nano /etc/prometheus/prometheus.yml

```
# Global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 10s.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 10s.
  scrape_timeout: 15s # scrape_timeout is set to the global default (10s).

# A scrape configuration containing exactly one endpoint to scrape: # Here it's localhost:9090
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ['localhost:9090']
```

- Create systemd Service unit

➤ Create a file: nano /etc/systemd/system/prometheus.service

➤ Add the following contents to it.

```

[Unit]
Description=Prometheus
Documentation=https://prometheus.io/docs/introduction/overview/
Wants=network-online.target
After=network-online.target

[Service]
Type=simple
Environment="GOMAXPROCS=2"
User=prometheus
Group=prometheus
ExecReload=/bin/kill -HUP $MAINPID
ExecStart=/usr/local/bin/prometheus \
    --config.file=/etc/prometheus/prometheus.yml \
    --storage.tsdb.path=/var/lib/prometheus \
    --web.console.templates=/etc/prometheus/consoles \
    --web.console.libraries=/etc/prometheus/console_libraries \
    --web.listen-address=0.0.0.0:9090 \
    --web.external-url=

SyslogIdentifier=prometheus
Restart=always

[Install]
WantedBy=multi-user.target

```

Set correct directory permissions.

```

for i in rules rules.d files_sd; do
    sudo chown -R prometheus:prometheus /etc/prometheus/${i};
done

```

```
for i in rules rules.d files_sd; do
    sudo chmod -R 775 /etc/prometheus/${i};
done
```

```
sudo chown -R prometheus:prometheus /var/lib/prometheus/
```

Start Prometheus service.

```
sudo systemctl daemon-reload
sudo systemctl start prometheus
```

- Configure firewall

```
sudo firewall-cmd --add-port=9090/tcp --permanent
sudo firewall-cmd --reload
```

- Open Prometheus Server IP/Hostname and port 9090

prometheus (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	4.123s ago	11.561ms	

- Install Node_exporter

- Install

➤ Ta dùng lệnh:

wget

https://github.com/prometheus/node_exporter/releases/download/v1.2.2/node_exporter-1.2.2.linux-amd64.tar.gz

➤ Giải nén file

tar -xzf node_exporter-1.2.2.linux-amd64

➤ Config lại Prometheus

Ta vào: nano /etc/prometheus/prometheus.yml

Và thêm vào:

```
- job_name: 'node_exporter'
  # metrics_path defaults to '/metrics'
  # scheme defaults to 'http'.

  static_configs:
    - targets: ['localhost:9100']
```

Sau đó ta vào: `cd node_exporter-1.2.2.linux-amd64/`
 Nhập: `./node_exporter`

```
t
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=tap
stats
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=tex
tfile
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=the
rmal_zone
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=tim
e
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=tim
ex
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=udp
_queues
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=una
me
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=vms
tat
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=xf
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:115 collector=zfs
level=info ts=2021-09-20T10:09:13.519Z caller=node_exporter.go:199 msg="Listenin
g on" address=:9100
level=info ts=2021-09-20T10:09:13.520Z caller=tls_config.go:191 msg="TLS is disa
bled." http2=false
```

○ Restart Prometheus

Targets

All Unhealthy Collapse All

node_exporter (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter"	13.714s ago	27.764ms	

prometheus (1/1 up) [show less](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	4.123s ago	11.561ms	

○ Install grafana

○ Add Grafana

```
cat <<EOF | sudo tee /etc/yum.repos.d/grafana.repo
[grafana]
name=grafana
baseurl=https://packages.grafana.com/oss/rpm
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packages.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
EOF
```

- Install Grafana

```
sudo dnf -y install grafana
```

- Start Grafana Service

```
sudo systemctl enable --now grafana-server.service
```

- Open firewall port for Grafana

```
sudo firewall-cmd --add-port=3000/tcp --permanent
sudo firewall-cmd --reload
```

- Open Grafana IP/Hostname and port 3000

- The default logins are:

username: admin

Password: admin

- Import Prometheus in Grafana

- Vào import nhập 1860 và chọn load

- Ngay chỗ Prometheus chọn Prometheus.

- Tương tự với 405



Import

Import dashboard from file or Grafana.com

Upload JSON file

Import via grafana.com

Load

Import via panel json

Load

Prometheus

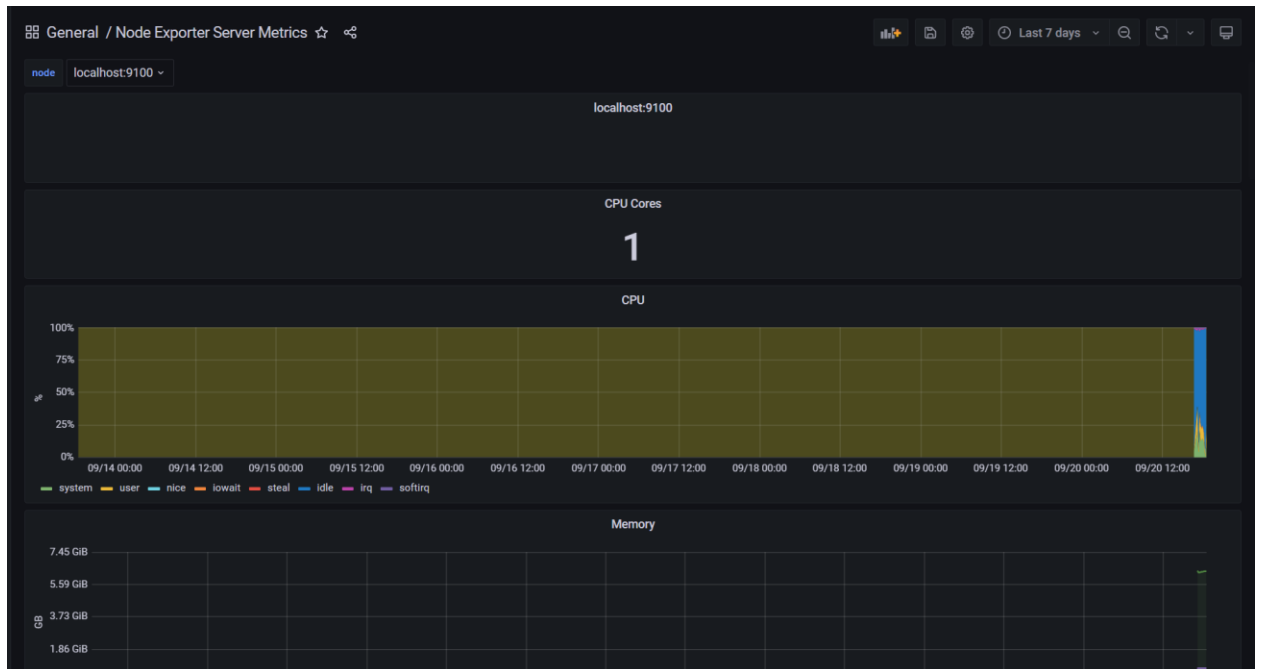
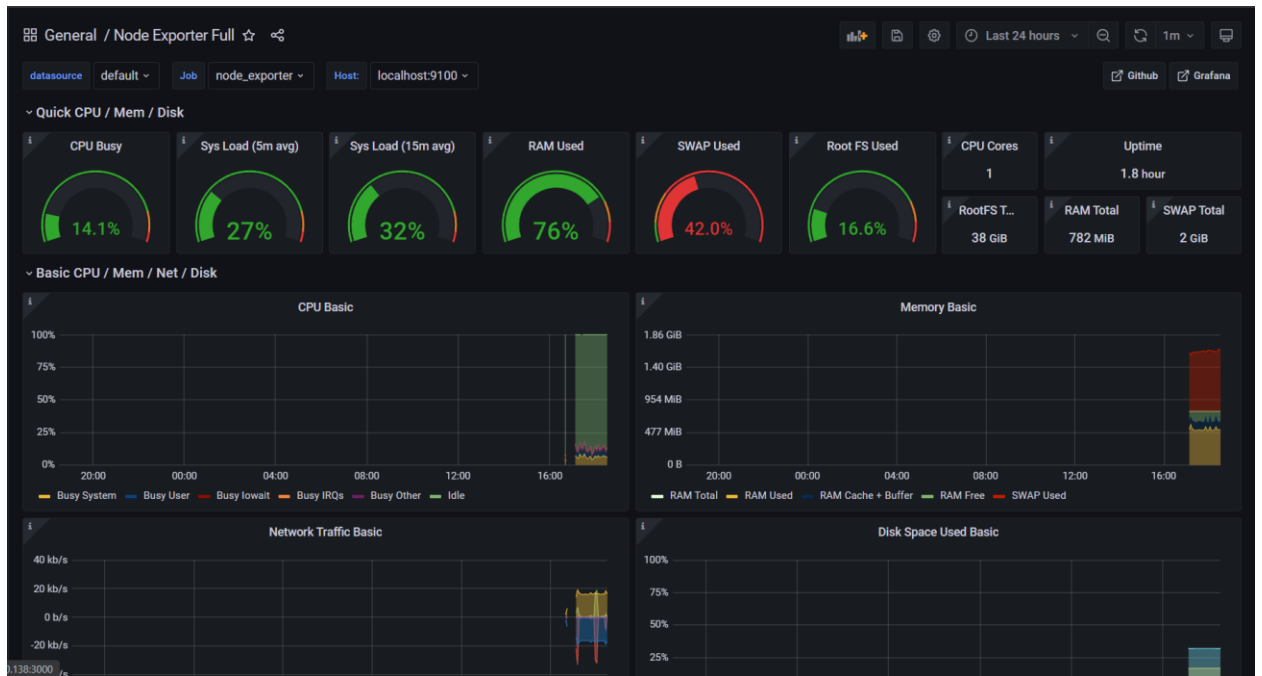


Import via grafana.com

Load

○ Result

Võ Minh Hoàng



Võ Minh Hoàng