CIT 215—Web Site Management

Miroslav Krajca

Contents

preface

Many images and pictures were obtained from: https://pixabay.com/
under the free for commercial use, no attribution required.

What is a web server

Web server admin

- **Web Server** —A computer that provides World Wide Web (WWW) services on the Internet. It includes the hardware, operating system, Web server software, and Web site content (Web pages). If the Web server is used internally and not by the public, it may be known as an "intranet server."

- **Web Server Administrator** —The Web server equivalent of a system administrator. Web server administrators are system architects responsible for web servers' overall design, implementation, and maintenance. They may or may not be responsible for Web content, which is traditionally the responsibility of the Webmaster.

- **Webmaster** —A webmaster is a person that is responsible for the implementation of a Web site. Webmasters must be proficient in HTML and one or more scripting and interface languages, such as JavaScript and Perl. They may or may not be responsible for the underlying server, which is traditionally the responsibility of the Web administrator (see above).

https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-44ver2.pdf

The term web server can refer to hardware or software, or both of them working together.

Parts of a web server

- **On the computer hardware side, a web server is a computer that stores web server software and a website's component files (for example, HTML documents, images, CSS stylesheets, and JavaScript files). A web server connects to the Internet or a local network and supports physical data interchange with other devices connected to the web.**

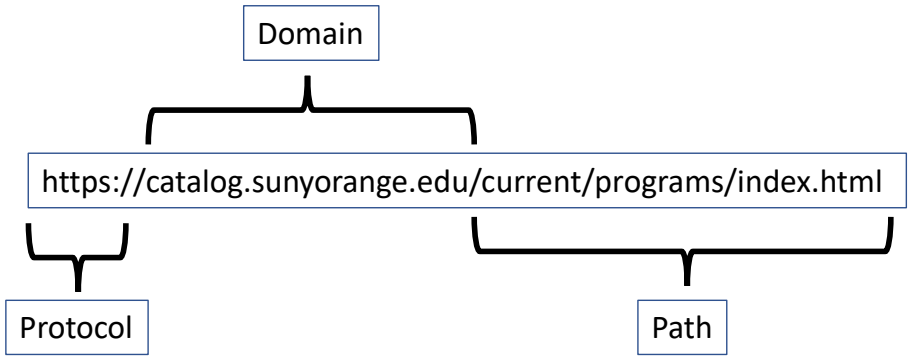
- **On the software side, a web server includes several parts that control how web users access hosted files. At a minimum, this is an HTTP server. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores or just by ip address, and it delivers the content of these hosted websites to the end user's device(web browser).**

When a web browser, like Google Chrome, Firefox, or MS Edge, needs a file hosted on a web server, the browser will request the file by HTTP/S(Hypertext Transfer Protocol/Hypertext Transfer Protocol Secure). When the web server receives the request, the HTTP/S server will accept it, find the content and send it back to the browser through HTTP/S.

More specifically, when a browser requests a page from a web server, the process will follow a series of steps. First, a person will specify a URL(Uniform Resource Locator) in a web browser's address bar. The web browser will then obtain the IP address of the domain name – either translating the URL through DNS (Domain Name System) or by searching in its cache. This will bring the browser to a web server. The browser will then request the specific file from the web server by an HTTP/S request. The web server will respond, sending the browser the requested page again through HTTP/S. If the requested page does not exist or if something goes wrong, the web server will respond with an error message. The browser will then be able to display the webpage.

Multiple domains also can be hosted on one web server.

1.0.1 Parts of a URL



Using the URL https://en.wikipedia.org/wiki/URL#Syntax as an example, components of a URL can include:

- **The protocol or scheme**. It is used to access a resource on the internet. Protocols include HTTP, HTTPS, ftps, mailto, and file. The resource is reached through the domain name system (DNS) name. In this example, the protocol is HTTPS.

- **Host name or domain name**. The unique reference represents a webpage. For this example, en.wikipedia.org.

- **Port name.** Usually not visible in URLs, but necessary. Always following a colon, port 80 is the default port for web servers, but there are other options. For example, :port80.

- **Path.** A path refers to a file or location on the web server. For this example,

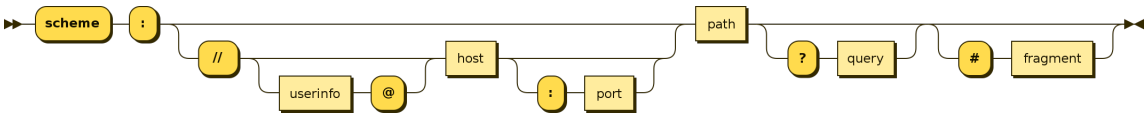
search/query.

- **Query.** Found in the URL of dynamic pages. The query consists of a question mark, followed by parameters. For this example, ?.

- **Parameters.** Pieces of information in a query string of a URL. Multiple parameters can be separated by ampersands (&). For this example, q=URL.

- **Fragment.** This is an internal page reference, which refers to a section within the webpage. It appears at the end of a URL and begins with a hashtag (#). Although not in the example above, an example could be #history in the URL https://en.wikipedia.org/wiki/Internet#History.

Every HTTP/S URL conforms to the syntax of a generic URI(Uniform Resource Identifier). The URI generic syntax consists of components organized hierarchically in order of decreasing significance from left to right.



https://en.wikipedia.org/wiki/URL#Syntax

What does a web server admin do?

Web Administrators manage web environment design, deployment, development, and maintenance activities. In addition, perform testing and quality assurance of web sites and web applications. They also monitor systems for intrusions or denial of service attacks, and report security breaches to appropriate personnel.

Other tasks include:

- Identify or document backup or recovery plans.
- Back up or modify applications and related data to provide for disaster recovery.
- Correct testing-identified problems, or recommend actions for their resolution.
- Identify, standardize, and communicate levels of access and security.
- Determine sources of Web page or server issues, and take action to correct such problems.

- Implement updates, upgrades, and patches in a timely manner to limit the loss of service.

Skills

People in this career often have these skills:

- **Critical Thinking** Thinking about the pros and cons of different ways to solve a problem.
- **Reading Comprehension** Reading work-related information.
- **Complex Problem Solving** Noticing a problem and figuring out the best way to solve it.
- **Active Listening** Listening to others, not interrupting, and asking good questions.
- **Monitoring** Keeping track of how well people and/or groups are doing in order to make improvements.
- **Systems Evaluation** Measuring how well a system is working and how to improve it.
- **Judgment and Decision Making** Thinking about the pros and cons of different options and picking the best one.

There are several ways to install a webserver. The simplest is to use some pre made installation provided by your Operation system. (Linux or, in the case of IIS, Microsoft) The other which will need to be done on some seldom installations is to compile from the source code and do a complete custom install. This method is sometimes necessary on Linux if we need to install proprietary modules or connections to non open source databases. However, this method is not available on Microsoft Windows, only on Linux/Unix Operating systems.

All webservers and for that matter almost all servers will need to have a static ip assigned to them. Make sure that you configured your installation of your OS by having a static IP address or if that is not possible you are running some sort of dynamic DNS service.(Mostly for home not production).

All web servers, and for that matter almost all servers, will need to have a static IP assigned to them. So make sure that you configured your OS installation by having a static IP address, or if that is not possible, you are running some sort of dynamic DNS service. (Mostly for at home, not production). You will also need access to a DNS server where you add your domain name and other ANAMES or CNAMES(conical name, Alias). This will be necessary in order to host multiple sites on your webserver(Virtual servers, NOT VMs, but different domains)

2.0.1 Apache web server

The Apache web server is the most widely-used web server (at this time, although nginx is gaining in popularity) worldwide. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software. Apache runs on Windows but is mainly used on Linux/Unix Operating Systems.

2.0.1.0 Linux Ubuntu

Prerequisites

Before you begin, you will need an Ubuntu 22.04 server set up with a non-root user with **sudo** privileges, and a firewall enabled to block non-essential ports. Non-essential ports must be disabled, or source IP restricted to maintain security. This is because in many cases the web-server will be internet facing and if un-needed ports are available hackers have a greater chance of breaking in.
The first step is to ensure that our distribution is updated and all security patches are applied.

Commands

```
ccc@occc-VirtualBox:~$ sudo apt update
[sudo] password for occc:
occc@occc-VirtualBox:~$ sudo apt upgrade
```

Enter a password when prompted and during the upgrade choose ”Y” to apply upgrades.
Then, install the **apache2** package:

Commands

```
occc@occc-VirtualBox:~$ sudo apt install apache2
```

When done installing make changes to the firewall. There are two ways to do this.
1) use the **ufw** utility.
During installation, Apache registers itself with UFW to provide a few application profiles that can be used to enable or disable access to Apache through the firewall.

Commands

```
occc@occc-VirtualBox:~$ sudo ufw app list
```

Output

```
occc@occc-VirtualBox:~$ sudo ufw app list
Available applications:
Apache
Apache Full
Apache Secure
Bind9
CUPS
OpenSSH
occc@occc-VirtualBox:~$
```

The output may wary slightly depending on other server applications installed. From the above output we see three profiles available for us to choose from.

- **Apache:** This profile opens only port **80** (normal, unencrypted web traffic)
- **Apache Full:** This profile opens both port **80** (normal, unencrypted web traffic) and port **443** (TLS/SSL encrypted traffic)
- **Apache Secure:** This profile opens only port **443** (TLS/SSL encrypted traffic)

You should enable the most restrictive profile that will still allow the traffic you've configured.
Since we have not configured SSL encryption we will have to open the 80 and for later on have the 443 opened as well. We can come back and disable the port 80 HTTP port.

Commands

```
occc@occc-VirtualBox:~$ sudo ufw allow 'Apache Full'
[sudo] password for occc:
Rules updated
Rules updated (v6)
occc@occc-VirtualBox:~$
```

We can verify the change by checking the status:

Commands

```
occc@occc-VirtualBox:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Bind9                      ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
Bind9 (v6)                 ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)

occc@occc-VirtualBox:~$
```

The Apache Full now has allowed status. There is also Apache Full(v6) this is for IPV6 the first one is for IPV4.

Commands

```
occc@occc-VirtualBox:~$ sudo ufw status
Status: inactive
occc@occc-VirtualBox:~$
```

If you got the inactive answer you did not configure a firewall.

Commands

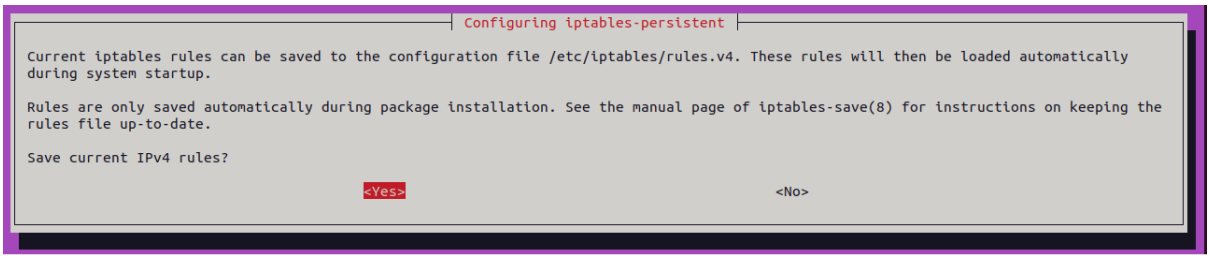
```
occc@occc-VirtualBox:~$ sudo ufw status
Status: inactive
occc@occc-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
occc@occc-VirtualBox:~$
```

Then do the status again.

2) use iptables directly.

Commands

```
root@occc-VirtualBox:~# apt install iptables-persistent
```



choose yes for both ipv4 and ipv6; look at the /etc/iptables/rules.v4

Commands

```
root@occc-VirtualBox:~# more  /etc/iptables/rules.v4
# Generated by iptables-save v1.8.7 on Tue Oct 11 16:19:50 2022
*filter
:INPUT DROP [1163:101992]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [7:2011]
:ufw-after-forward - [0:0]
:ufw-after-input - [0:0]
:ufw-after-logging-forward - [0:0]
:ufw-after-logging-input - [0:0]
:ufw-after-logging-output - [0:0]
:ufw-after-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-before-input - [0:0]
:ufw-before-logging-forward - [0:0]
:ufw-before-logging-input - [0:0]
:ufw-before-logging-output - [0:0]
:ufw-before-output - [0:0]
:ufw-logging-allow - [0:0]
:ufw-logging-deny - [0:0]
:ufw-not-local - [0:0]
:ufw-reject-forward - [0:0]
:ufw-reject-input - [0:0]
:ufw-reject-output - [0:0]
:ufw-skip-to-policy-forward - [0:0]
:ufw-skip-to-policy-input - [0:0]
:ufw-skip-to-policy-output - [0:0]
:ufw-track-forward - [0:0]
:ufw-track-input - [0:0]
:ufw-track-output - [0:0]
:ufw-user-forward - [0:0]
```

```
:ufw-user-input - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
:ufw-user-logging-forward - [0:0]
:ufw-user-logging-input - [0:0]
:ufw-user-logging-output - [0:0]
:ufw-user-output - [0:0]
-A INPUT -j ufw-before-logging-input
-A INPUT -j ufw-before-input
-A INPUT -j ufw-after-input
-A INPUT -j ufw-after-logging-input
-A INPUT -j ufw-reject-input
-A INPUT -j ufw-track-input
-A FORWARD -j ufw-before-logging-forward
-A FORWARD -j ufw-before-forward
-A FORWARD -j ufw-after-forward
-A FORWARD -j ufw-after-logging-forward
-A FORWARD -j ufw-reject-forward
-A FORWARD -j ufw-track-forward
-A OUTPUT -j ufw-before-logging-output
-A OUTPUT -j ufw-before-output
-A OUTPUT -j ufw-after-output
-A OUTPUT -j ufw-after-logging-output
-A OUTPUT -j ufw-reject-output
-A OUTPUT -j ufw-track-output
-A ufw-after-input -p udp -m udp --dport 137 -j ufw-skip-to-
policy-input
-A ufw-after-input -p udp -m udp --dport 138 -j ufw-skip-to-
policy-input
-A ufw-after-input -p tcp -m tcp --dport 139 -j ufw-skip-to-
policy-input
-A ufw-after-input -p tcp -m tcp --dport 445 -j ufw-skip-to-
policy-input
-A ufw-after-input -p udp -m udp --dport 67 -j ufw-skip-to-
policy-input
-A ufw-after-input -p udp -m udp --dport 68 -j ufw-skip-to-
policy-input
-A ufw-after-input -m addrtype --dst-type BROADCAST -j ufw-
skip-to-policy-input
```

```
-A ufw-after-logging-forward -m limit --limit 3/min --limit-
burst 10 -j LOG --log-prefix "[UFW BLOCK] "
-A ufw-after-logging-input -m limit --limit 3/min --limit-
burst 10 -j LOG --log-prefix "[UFW BLOCK] "
-A ufw-before-forward -m conntrack --ctstate RELATED,ESTABLISHED -
j ACCEPT
-A ufw-before-forward -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A ufw-before-forward -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A ufw-before-forward -p icmp -m icmp --icmp-type 12 -j ACCEPT
-A ufw-before-forward -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A ufw-before-forward -j ufw-user-forward
-A ufw-before-input -i lo -j ACCEPT
-A ufw-before-input -m conntrack --ctstate RELATED,ESTABLISHED -
j ACCEPT
-A ufw-before-input -m conntrack --ctstate INVALID -j ufw-
logging-deny
-A ufw-before-input -m conntrack --ctstate INVALID -j DROP
-A ufw-before-input -p icmp -m icmp --icmp-type 3 -j ACCEPT
-A ufw-before-input -p icmp -m icmp --icmp-type 11 -j ACCEPT
-A ufw-before-input -p icmp -m icmp --icmp-type 12 -j ACCEPT
-A ufw-before-input -p icmp -m icmp --icmp-type 8 -j ACCEPT
-A ufw-before-input -p udp -m udp --sport 67 --dport 68 -
j ACCEPT
-A ufw-before-input -j ufw-not-local
-A ufw-before-input -d 224.0.0.251/32 -p udp -m udp --
dport 5353 -j ACCEPT
-A ufw-before-input -d 239.255.255.250/32 -p udp -m udp --
dport 1900 -j ACCEPT
-A ufw-before-input -j ufw-user-input
-A ufw-before-output -o lo -j ACCEPT
-A ufw-before-output -m conntrack --ctstate RELATED,ESTABLISHED -
j ACCEPT
-A ufw-before-output -j ufw-user-output
-A ufw-logging-allow -m limit --limit 3/min --limit-burst 10 -
j LOG --log-prefix "[UFW ALLOW] "
-A ufw-logging-deny -m conntrack --ctstate INVALID -m limit --
limit 3/min --limit-burst 10 -j RETURN
-A ufw-logging-deny -m limit --limit 3/min --limit-burst 10 -
j LOG --log-prefix "[UFW BLOCK] "
```

```
-A ufw-not-local -m addrtype --dst-type LOCAL -j RETURN
-A ufw-not-local -m addrtype --dst-type MULTICAST -j RETURN
-A ufw-not-local -m addrtype --dst-type BROADCAST -j RETURN
-A ufw-not-local -m limit --limit 3/min --limit-burst 10 -
j ufw-logging-deny
-A ufw-not-local -j DROP
-A ufw-skip-to-policy-forward -j DROP
-A ufw-skip-to-policy-input -j DROP
-A ufw-skip-to-policy-output -j ACCEPT
-A ufw-track-output -p tcp -m conntrack --ctstate NEW -
j ACCEPT
-A ufw-track-output -p udp -m conntrack --ctstate NEW -
j ACCEPT
-A ufw-user-input -p tcp -m tcp --dport 53 -m comment --
comment "\'dapp_Bind9\'" -j ACCEPT
-A ufw-user-input -p udp -m udp --dport 53 -m comment --
comment "\'dapp_Bind9\'" -j ACCEPT
-A ufw-user-input -p tcp -m multiport --dports 80,443 -
m comment --comment "\'dapp_Apache%20Full\'" -j ACCEPT
-A ufw-user-limit -m limit --limit 3/min -j LOG --log-
prefix "[UFW LIMIT BLOCK] "
-A ufw-user-limit -j REJECT --reject-with icmp-port-
unreachable
-A ufw-user-limit-accept -j ACCEPT
COMMIT
# Completed on Tue Oct 11 16:19:50 2022
root@occc-VirtualBox:~#
```

we can add: **-A ufw-user-input -p tcp -m multiport –dports 80,443 -m comment –comment ”đapp_Apache%20Full” -j ACCEPT** to the file and make sure that iptables persistant is enabled.

Commands

```
root@occc-VirtualBox:~# sudo systemctl is-enabled netfilter-
persistent.service
enabled
root@occc-VirtualBox:~#
```

if not then enable it:

Commands

```
sudo systemctl enable netfilter-persistent.service
```

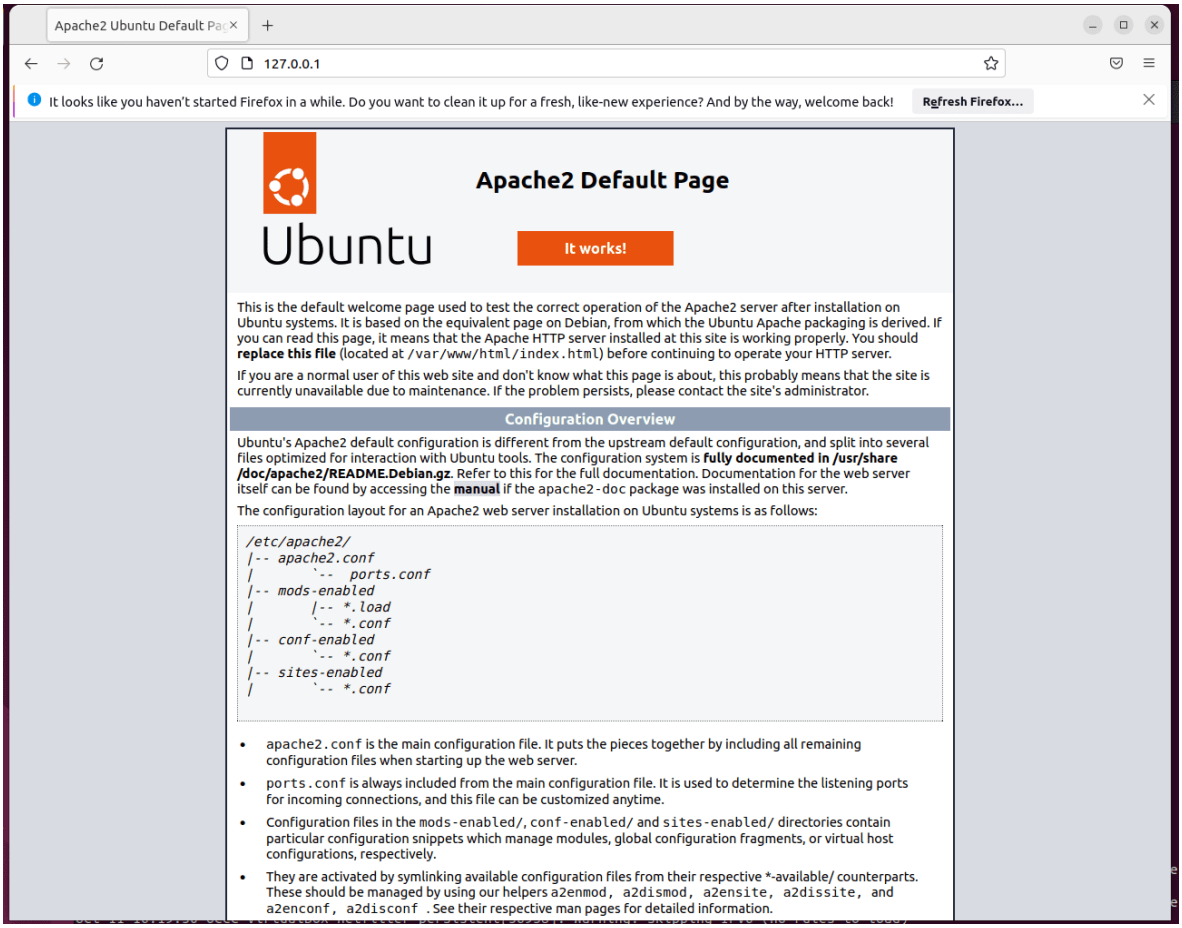
Get status:

Commands

```
root@occc-VirtualBox:~# sudo systemctl is-enabled netfilter-
persistent.service
enabled
root@occc-VirtualBox:~# sudo systemctl status netfilter-
persistent.service
  netfilter-persistent.service - netfilter persistent configuration
     Loaded: loaded (/lib/systemd/system/netfilter-
persistent.service; enabled; vendor preset: enabled)
    Drop-In: /etc/systemd/system/netfilter-
persistent.service.d
             iptables.conf
      Active: active (exited) since Tue 2022-10-
11 16:19:50 EDT; 10min ago
        Docs: man:netfilter-persistent(8)
    Main PID: 56934 (code=exited, status=0/SUCCESS)
         CPU: 2ms

Oct 11 16:19:50 occc-VirtualBox systemd[1]: Starting netfilter persist
Oct 11 16:19:50 occc-VirtualBox netfilter-
persistent[56936]: run-parts: executing /usr/share/netfilter-
persistent/plugins.d/15-ip4tables start
Oct 11 16:19:50 occc-VirtualBox netfilter-
persistent[56937]: Warning: skipping IPv4 (no rules to load)
Oct 11 16:19:50 occc-VirtualBox netfilter-
persistent[56936]: run-parts: executing /usr/share/netfilter-
persistent/plugins.d/25-ip6tables start
Oct 11 16:19:50 occc-VirtualBox netfilter-
persistent[56938]: Warning: skipping IPv6 (no rules to load)
Oct 11 16:19:50 occc-VirtualBox systemd[1]: Finished netfilter persist
root@occc-VirtualBox:~#
```

At this point, check that we can get the web server's default webpage.

go to http:\\127.0.0.1 or http:\\localhost or http:\\<your ip address>



If you don't have the above, you need to check the logs and see if everything is configured correctly.

Managing the Apache Process

Now that your web server is up and running let's examine some basic management commands using **systemctl**.

To stop your web server, run:

Commands

```
sudo systemctl stop apache2
```

To start the web server when it is stopped, run:

Commands

```
sudo systemctl start apache2
```

To stop and then start the service again, run:

Commands

```
sudo systemctl restart apache2
```

If you simply make configuration changes, Apache can reload without dropping connections. To do this, use the following command:

Commands

```
sudo systemctl reload apache2
```

By default, Apache is configured to start automatically when the server boots. If this is not what you want, disable this behavior by running:

Commands

```
sudo systemctl disable apache2
```

To re-enable the service to start up at boot, run:

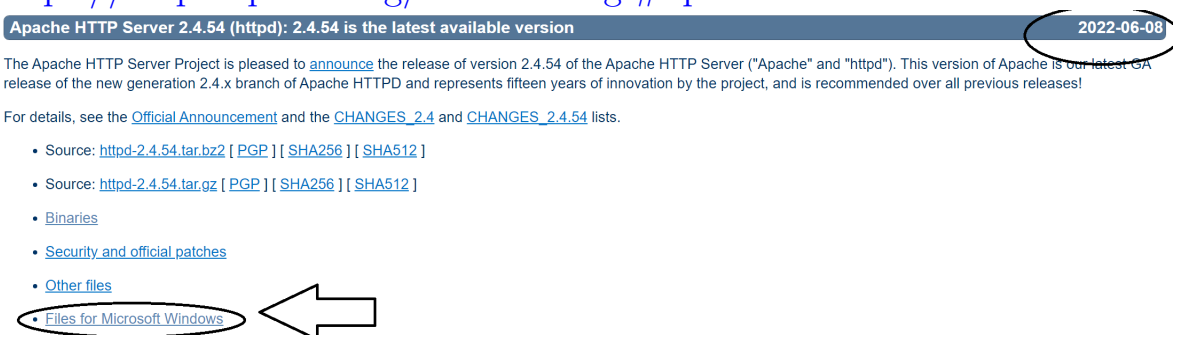
Commands

```
sudo systemctl enable apache2
```

Apache will now start automatically when the server boots again.

2.0.1.0 Windows

Latest as of 10/10/2022
https://httpd.apache.org/download.cgi#apache24



2.0.2 Nginx webserver

NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption
NGINX powers several high-visibility sites, such as Netflix, Hulu, Pinterest, CloudFlare, Airbnb, WordPress.com, GitHub, SoundCloud, Zynga, Eventbrite, Zappos, Media Temple, Heroku, RightScale, Engine Yard, TikToc, StackPath, CDN77 and many others.

2.0.2.0 Linux Ubuntu

Prerequisites

Before you begin, you will need an Ubuntu 22.04 server set up with a non-root user with **sudo** privileges and a firewall enabled to block non-essential ports. Non essential ports have to be disables or source ip restricted to maintain security. In many cases the web-server will be internet facing and if un-needed ports are available hackers have a greater chance of breaking in.
First step is to ensure that our distribution is updated and all security patches applied.

Because Nginx is available in Ubuntu's default repositories, it is possible to install it from these repositories using the **apt** packaging system.

Commands

```
sudo apt update
sudo apt install nginx
```

Press **Y** when prompted to confirm installation. If you are prompted to restart any services, press **ENTER** to accept the defaults and continue. **apt** will install Nginx and any required dependencies to your server.

Adjusting the Firewall

Before testing Nginx, the firewall software needs to be configured to allow access to the service. Nginx registers itself as a service with **ufw** upon installation, making it straightforward to allow Nginx access.

List the application configurations that **ufw** knows how to work with by typing:

Commands

```
sudo ufw app list
```

You should get a listing of the application profiles:

Commands

```
occc@occc-VirtualBox:~$ sudo ufw app list
Available applications:
Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH
occc@occc-VirtualBox:~$
```

List of apps might be different depending on what was installed previously. As demonstrated by the output, there are three profiles available for Nginx:

- **Nginx Full:** This profile opens both port **80** (normal, unencrypted web traffic) and port **443** (TLS/SSL encrypted traffic)
- **Nginx HTTP:** This profile opens only port **80** (normal, unencrypted web traffic)
- **Nginx HTTPS:** This profile opens only port **443** (TLS/SSL encrypted traffic)

It is recommended that you enable the most restrictive profile that will still allow the traffic you've configured.
Since we have not configured SSL encryption we will have to open the 80 and for later on have the 443 opened as well. We can come back and disable the port 80 HTTP port.

Commands

```
occc@occc-VirtualBox:~$ sudo ufw allow 'Nginx Full'
Rules updated
Rules updated (v6)
occc@occc-VirtualBox:~$
```

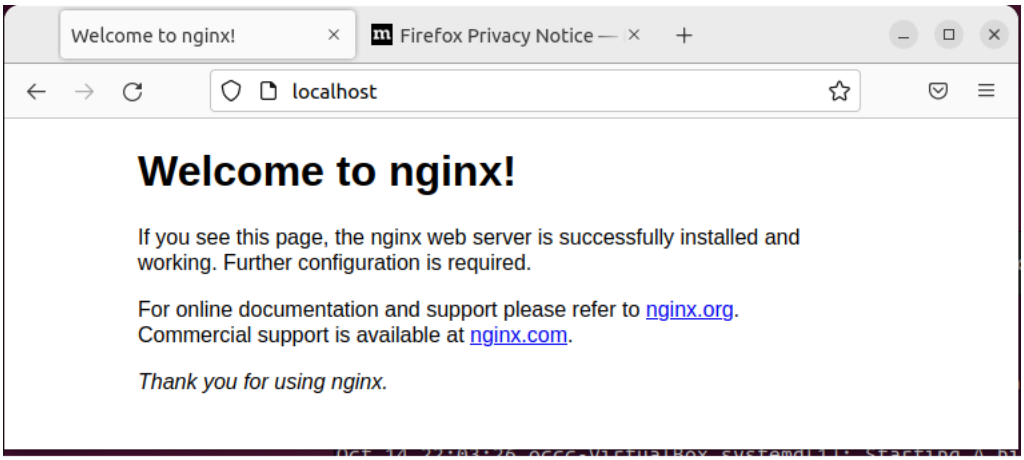
Commands

```
occc@occc-VirtualBox:~$ sudo ufw status
Status: inactive
occc@occc-VirtualBox:~$ sudo ufw enable
Firewall is active and enabled on system startup
occc@occc-VirtualBox:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Nginx Full                 ALLOW       Anywhere
Nginx Full (v6)            ALLOW       Anywhere (v6)

occc@occc-VirtualBox:~$
```

Commands

```
occc@occc-VirtualBox:~$ systemctl status nginx
  nginx.service - A high performance web server and a reverse proxy se
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor pre
Active: active (running) since Fri 2022-10-
14 22:03:26 EDT; 9min ago
Docs: man:nginx(8)
Main PID: 3186 (nginx)
Tasks: 2 (limit: 9457)
Memory: 3.2M
CPU: 11ms
CGroup: /system.slice/nginx.service
          3186 "nginx: master process /usr/sbin/nginx -
g daemon on; master>
          3189 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" ""

Oct 14 22:03:26 occc-VirtualBox systemd[1]: Starting A high performanc
Oct 14 22:03:26 occc-VirtualBox systemd[1]: Started A high performance
occc@occc-VirtualBox:~$
```

Open your web browser to localhost or your server ip address or FQDN.



Managing the Nginx Process

To stop your web server, type:

Commands

```
sudo systemctl stop nginx
```

To start the web server when it is stopped, type:

Commands

```
sudo systemctl start nginx
```

To stop and then start the service again, type:

Commands

```
sudo systemctl restart nginx
```

If you are only making configuration changes, Nginx can often reload without dropping connections. To do this, type:

Commands

```
sudo systemctl reload nginx
```

By default, Nginx is configured to start automatically when the server boots. If this is not what you want, you can disable this behavior by typing:

Commands

```
sudo systemctl disable nginx
```

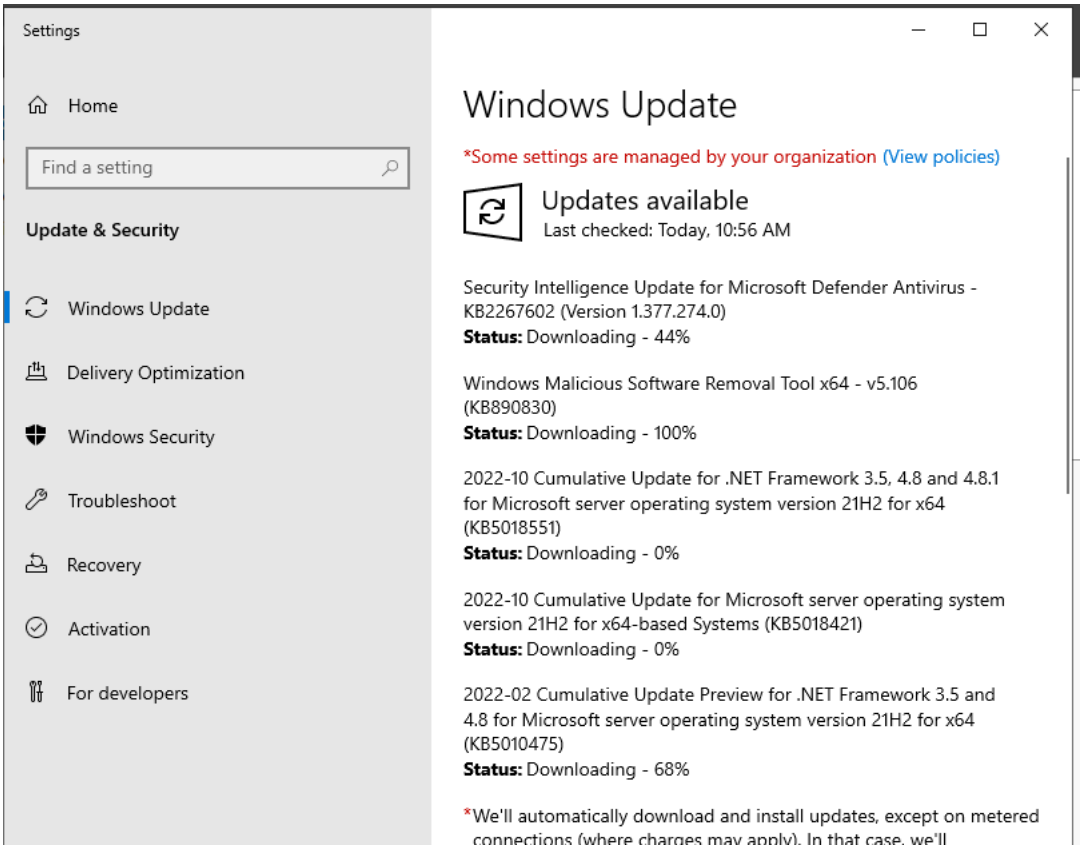
To re-enable the service to start up at boot, you can type:

Commands

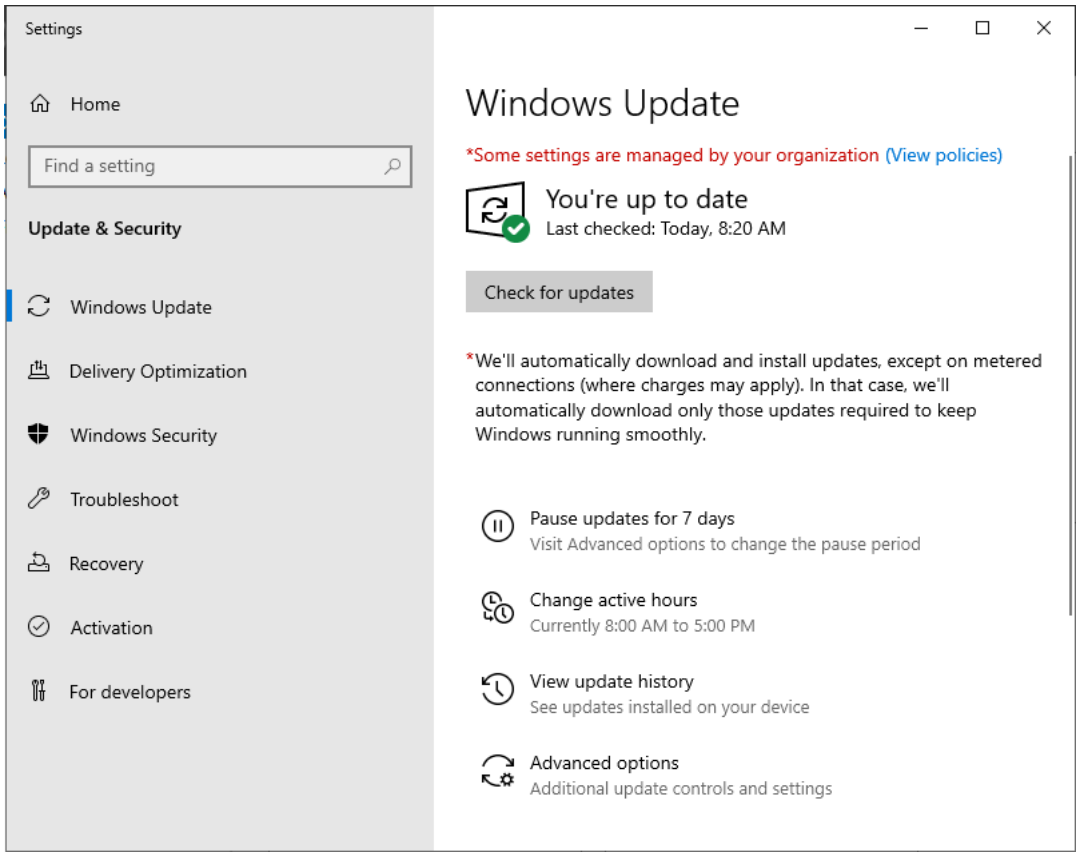
```
sudo systemctl enable nginx
```

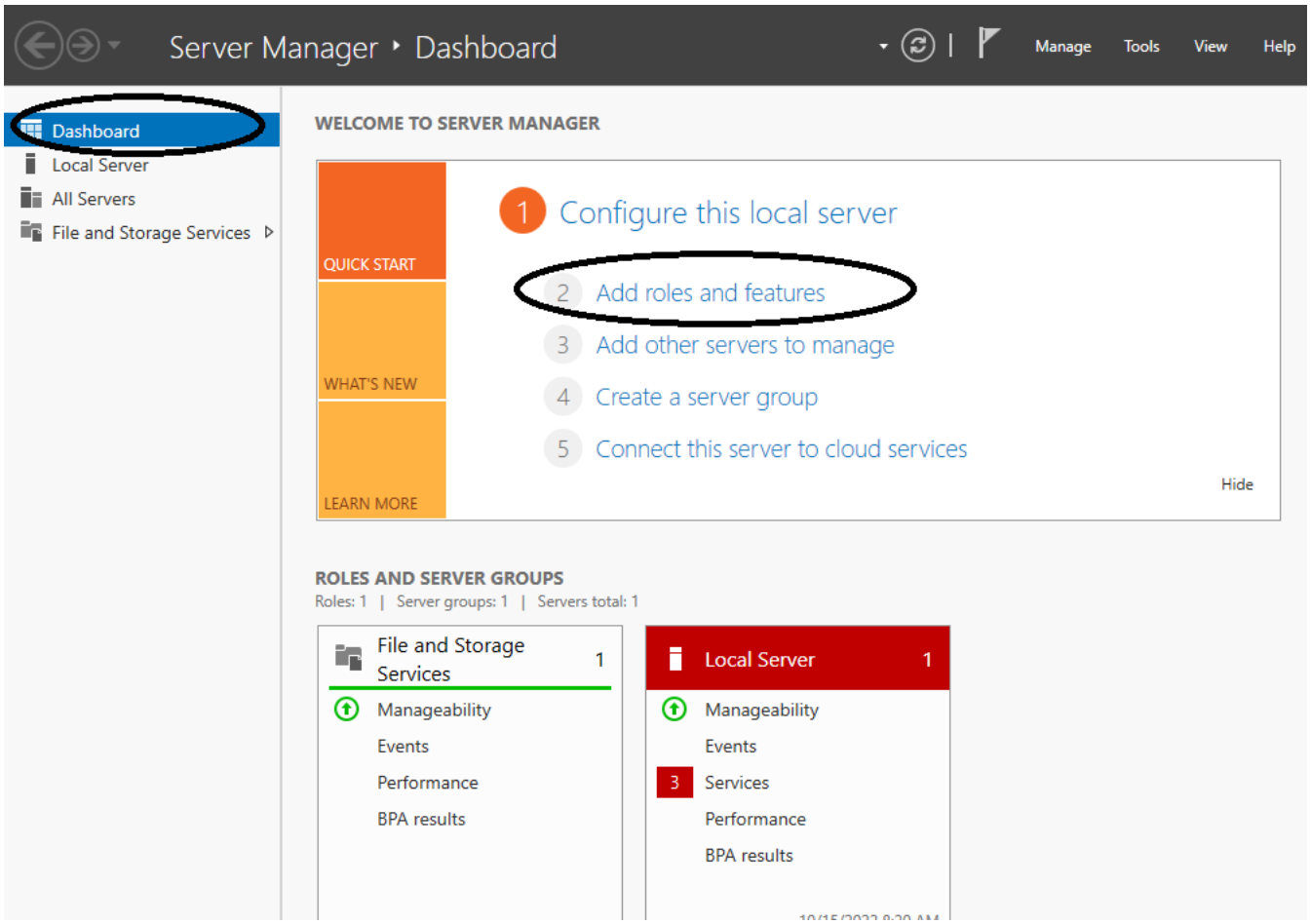
2.0.3 Windows IIS

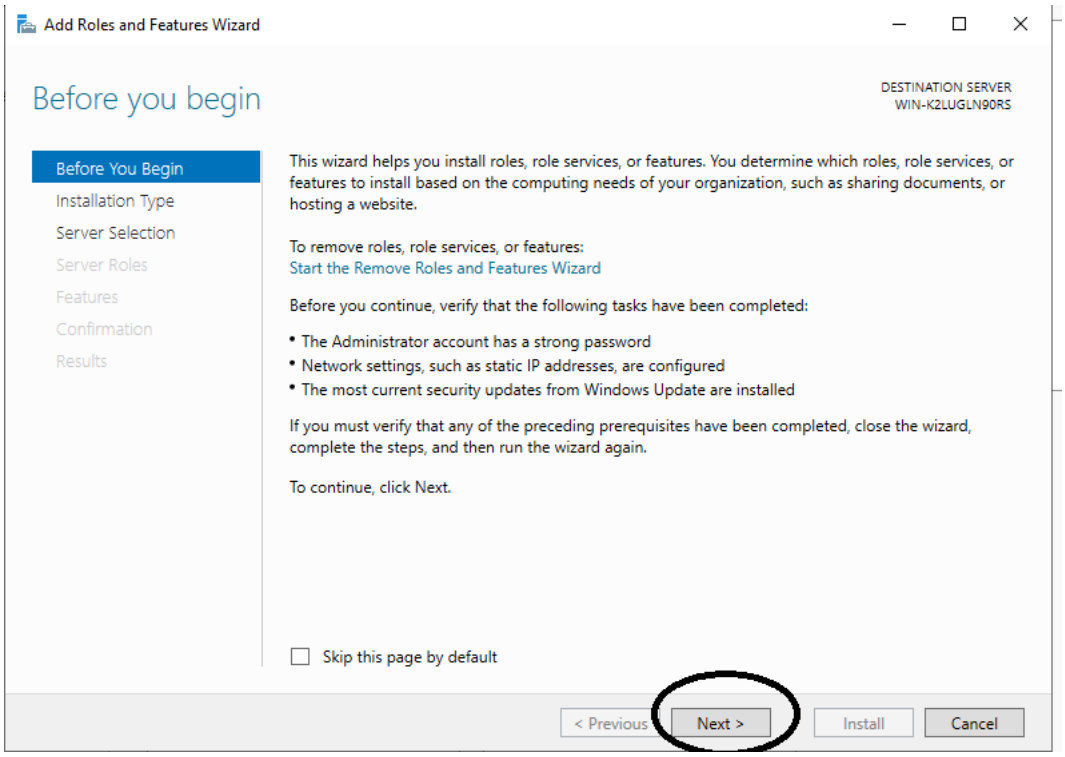
Make sure you have all the latest security updates installed.
This may not be possible on production systems during business hours and would need to be scheduled.

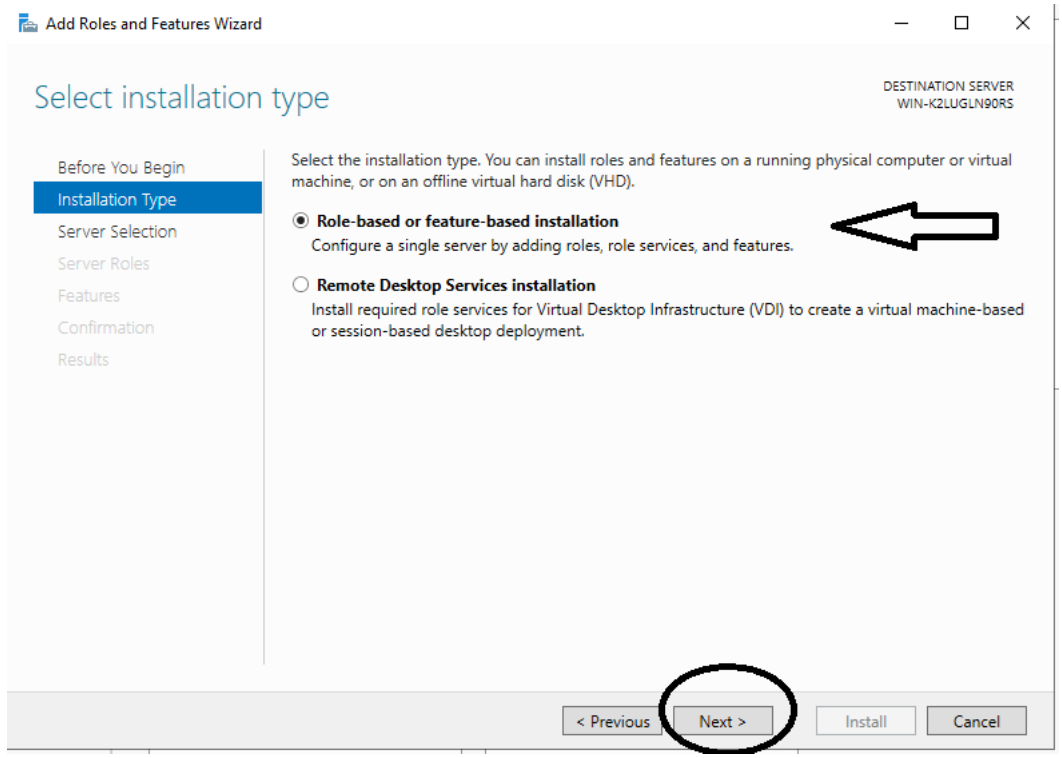


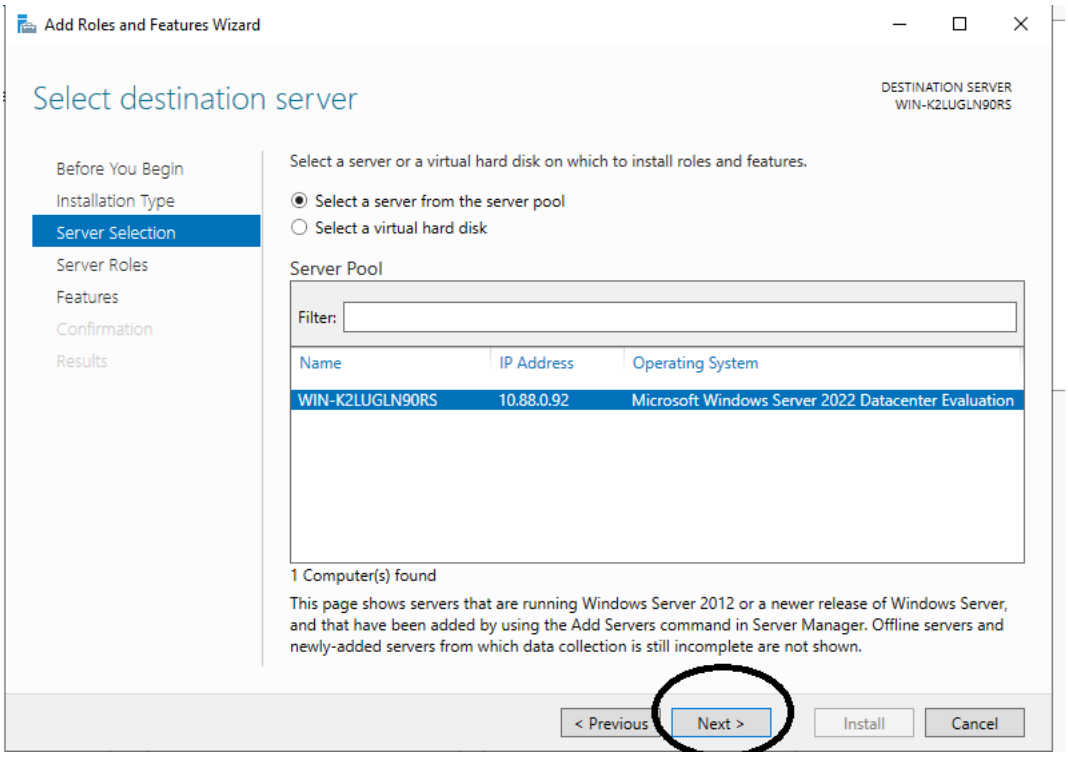
Once you have all the updates installed.

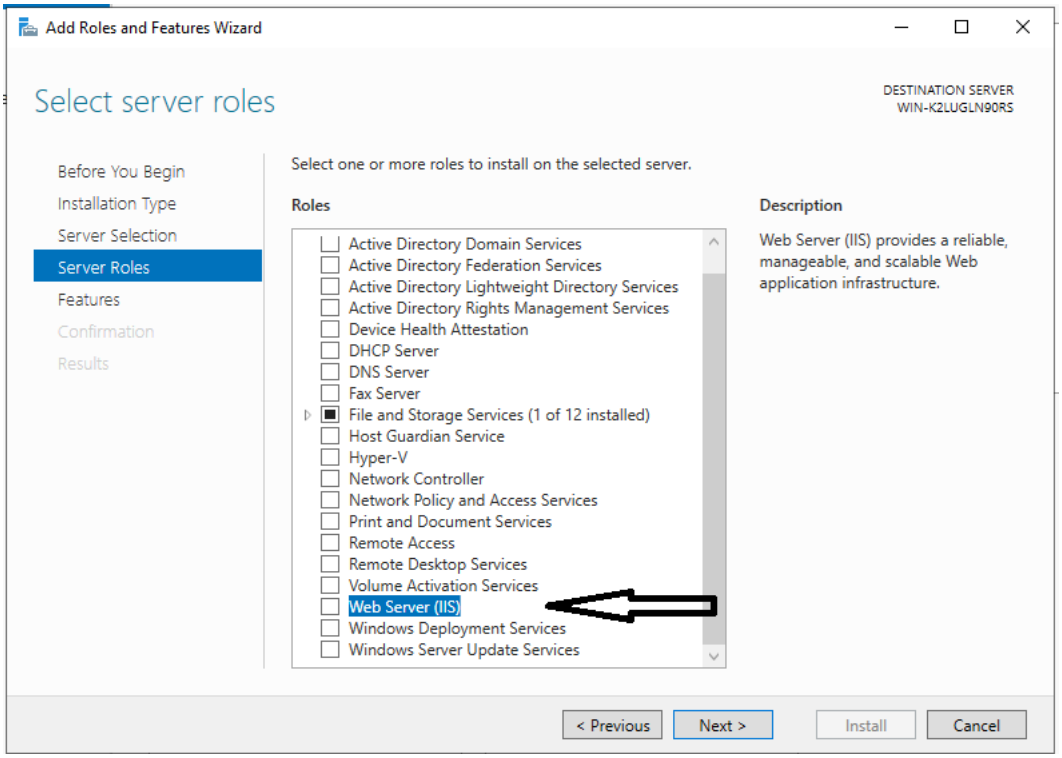

Settings
Home
Find a setting
Update & Security
Windows Update
Delivery Optimization
Windows Security
Troubleshoot
Recovery
Activation
For developers
Windows Update
*Some settings are managed by your organization (View policies)
You're up to date
Last checked: Today, 8:20 AM
Check for updates
*We'll automatically download and install updates, except on metered connections (where charges may apply). In that case, we'll automatically download only those updates required to keep Windows running smoothly.
Pause updates for 7 days
Visit Advanced options to change the pause period
Change active hours
Currently 8:00 AM to 5:00 PM
View update history
See updates installed on your device
Advanced options
Additional update controls and settings


Server Manager ‣ Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
File and Storage Services
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local server
2 Add roles and features
3 Add other servers to manage
4 Create a server group
5 Connect this server to cloud services
Hide
ROLES AND SERVER GROUPS
Roles: 1 | Server groups: 1 | Servers total: 1
File and Storage Services 1
Manageability
Events
Performance
BPA results
Local Server 1
Manageability
Events
3 Services
Performance
BPA results

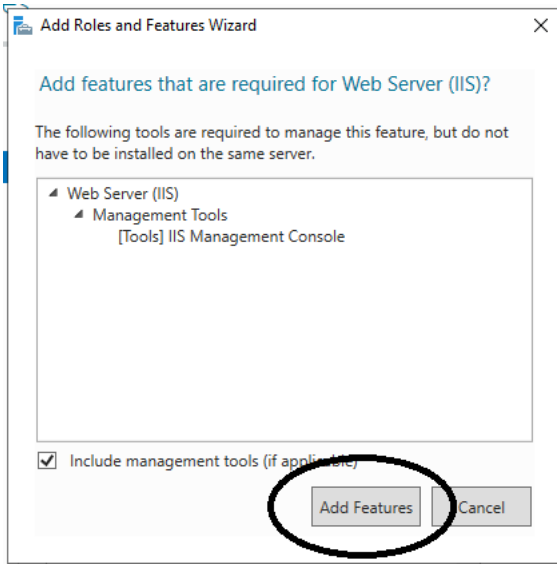

Add Roles and Features Wizard
Before you begin
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.
To remove roles, role services, or features:
Start the Remove Roles and Features Wizard
Before you continue, verify that the following tasks have been completed:
• The Administrator account has a strong password
• Network settings, such as static IP addresses, are configured
• The most current security updates from Windows Update are installed
If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.
To continue, click Next.
Skip this page by default
< Previous
Next >
Install
Cancel

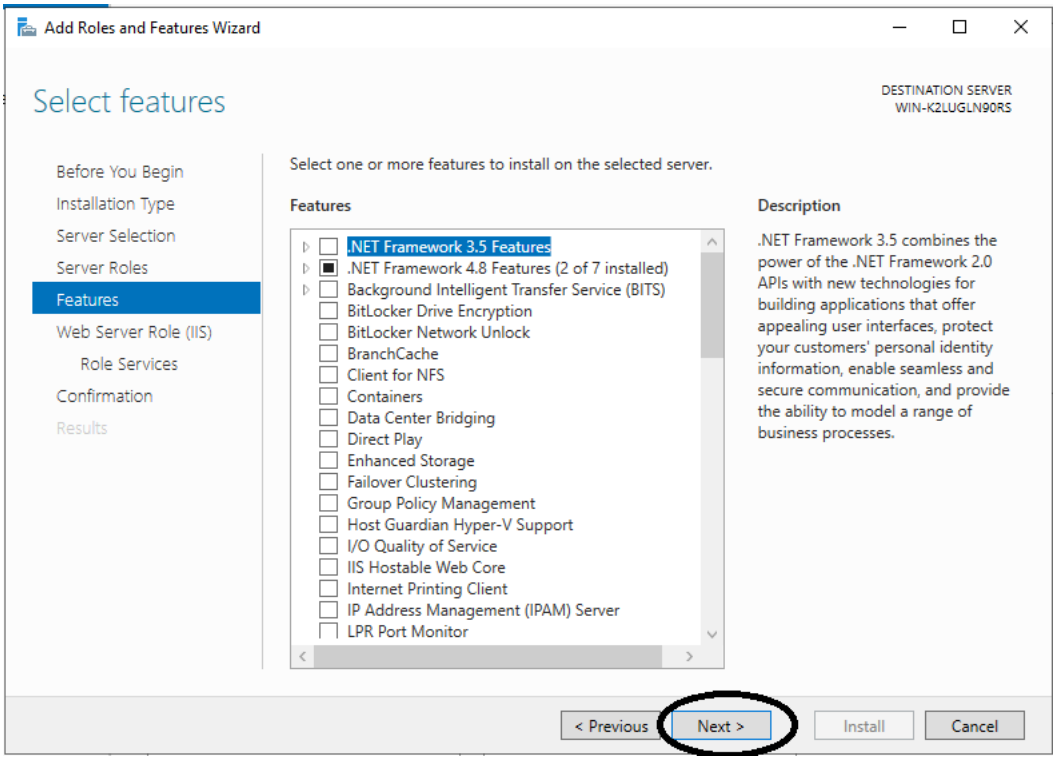

Add Roles and Features Wizard
Select installation type
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).
Role-based or feature-based installation
Configure a single server by adding roles, role services, and features.
Remote Desktop Services installation
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.
< Previous
Next >
Install
Cancel

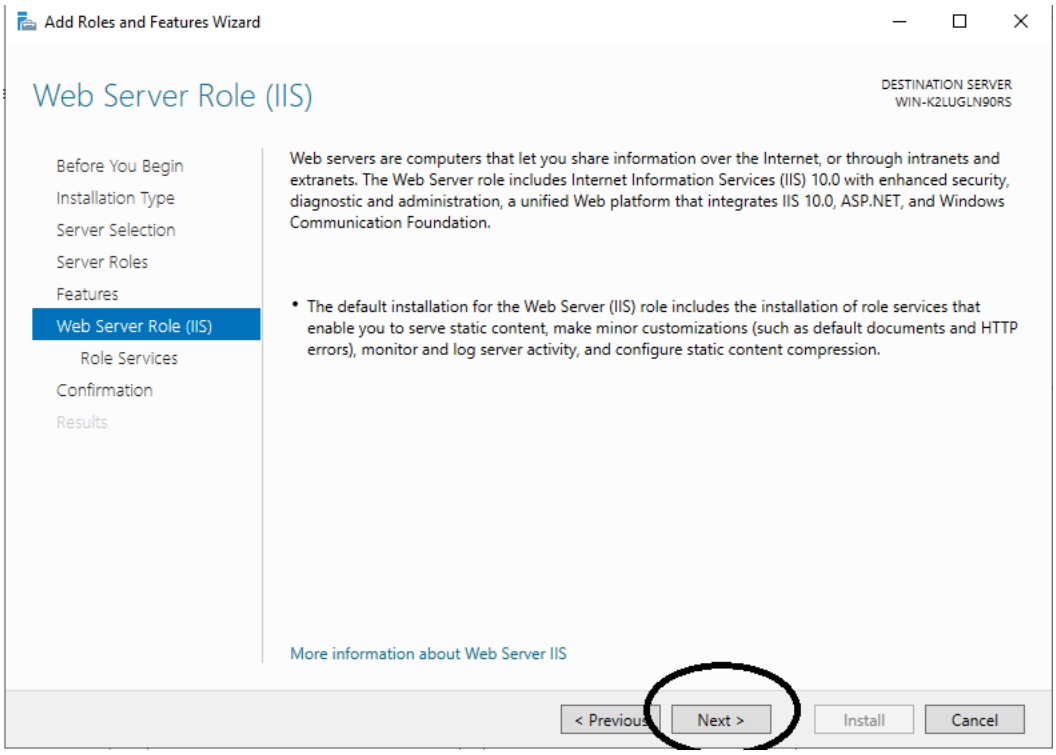


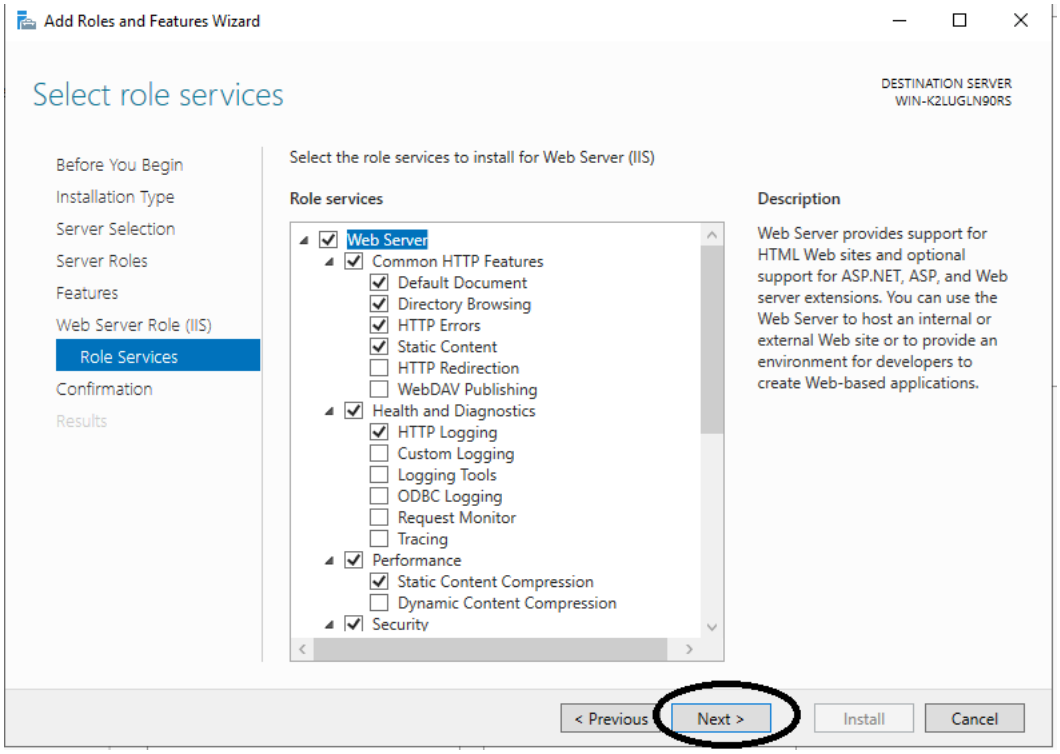


check IIS webserver.

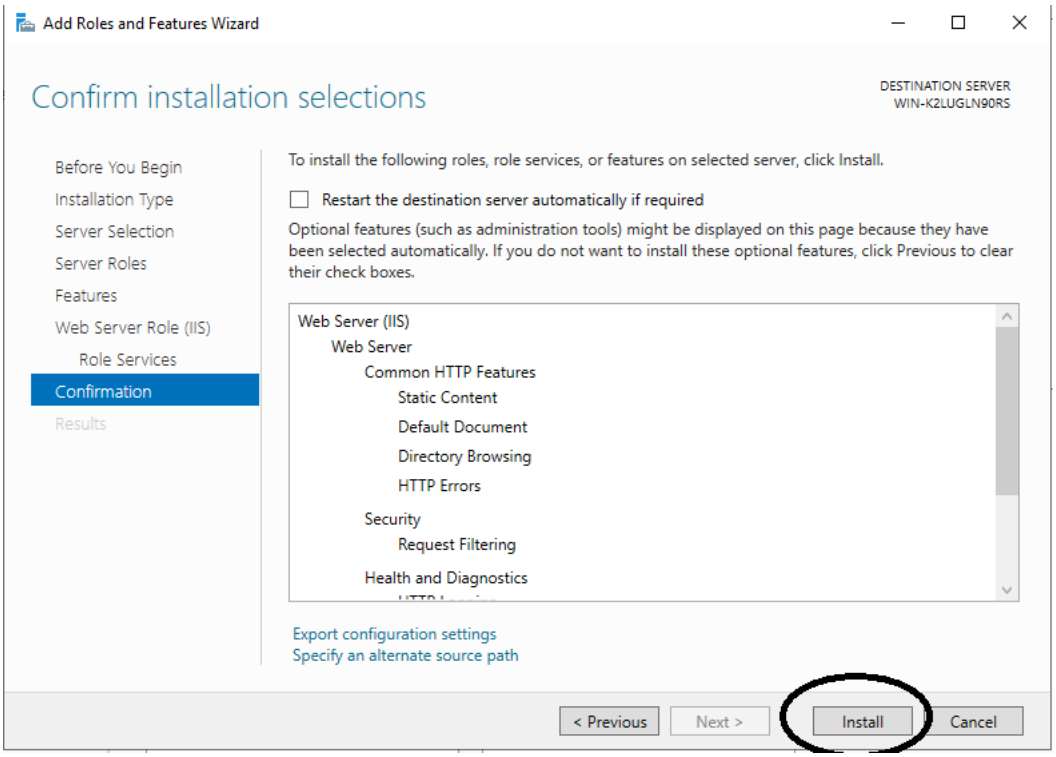

Add Roles and Features Wizard
Add features that are required for Web Server (IIS)?
The following tools are required to manage this feature, but do not have to be installed on the same server.
Web Server (IIS)
Management Tools
[Tools] IIS Management Console
Include management tools (if applicable)
Add Features
Cancel

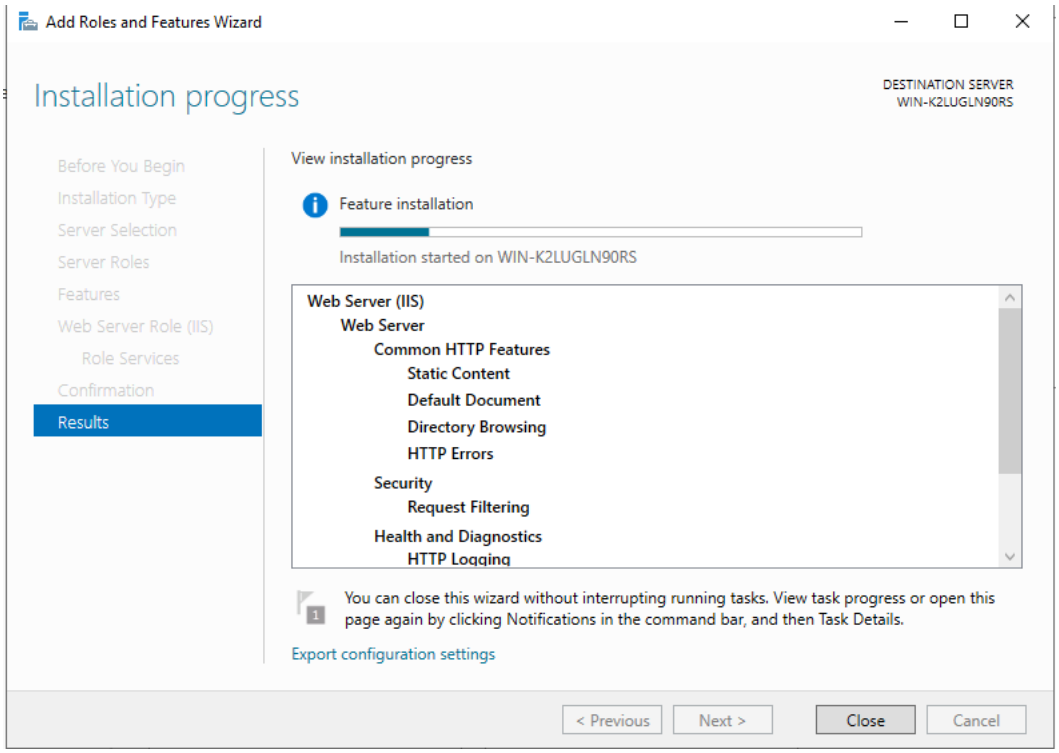

Add Roles and Features Wizard
Select features
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Web Server Role (IIS)
Role Services
Confirmation
Results
Select one or more features to install on the selected server.
Features
.NET Framework 3.5 Features
.NET Framework 4.8 Features (2 of 7 installed)
Background Intelligent Transfer Service (BITS)
BitLocker Drive Encryption
BitLocker Network Unlock
BranchCache
Client for NFS
Containers
Data Center Bridging
Direct Play
Enhanced Storage
Failover Clustering
Group Policy Management
Host Guardian Hyper-V Support
I/O Quality of Service
IIS Hostable Web Core
Internet Printing Client
IP Address Management (IPAM) Server
LPR Port Monitor
Description
.NET Framework 3.5 combines the power of the .NET Framework 2.0 APIs with new technologies for building applications that offer appealing user interfaces, protect your customers' personal identity information, enable seamless and secure communication, and provide the ability to model a range of business processes.
< Previous
Next >
Install
Cancel



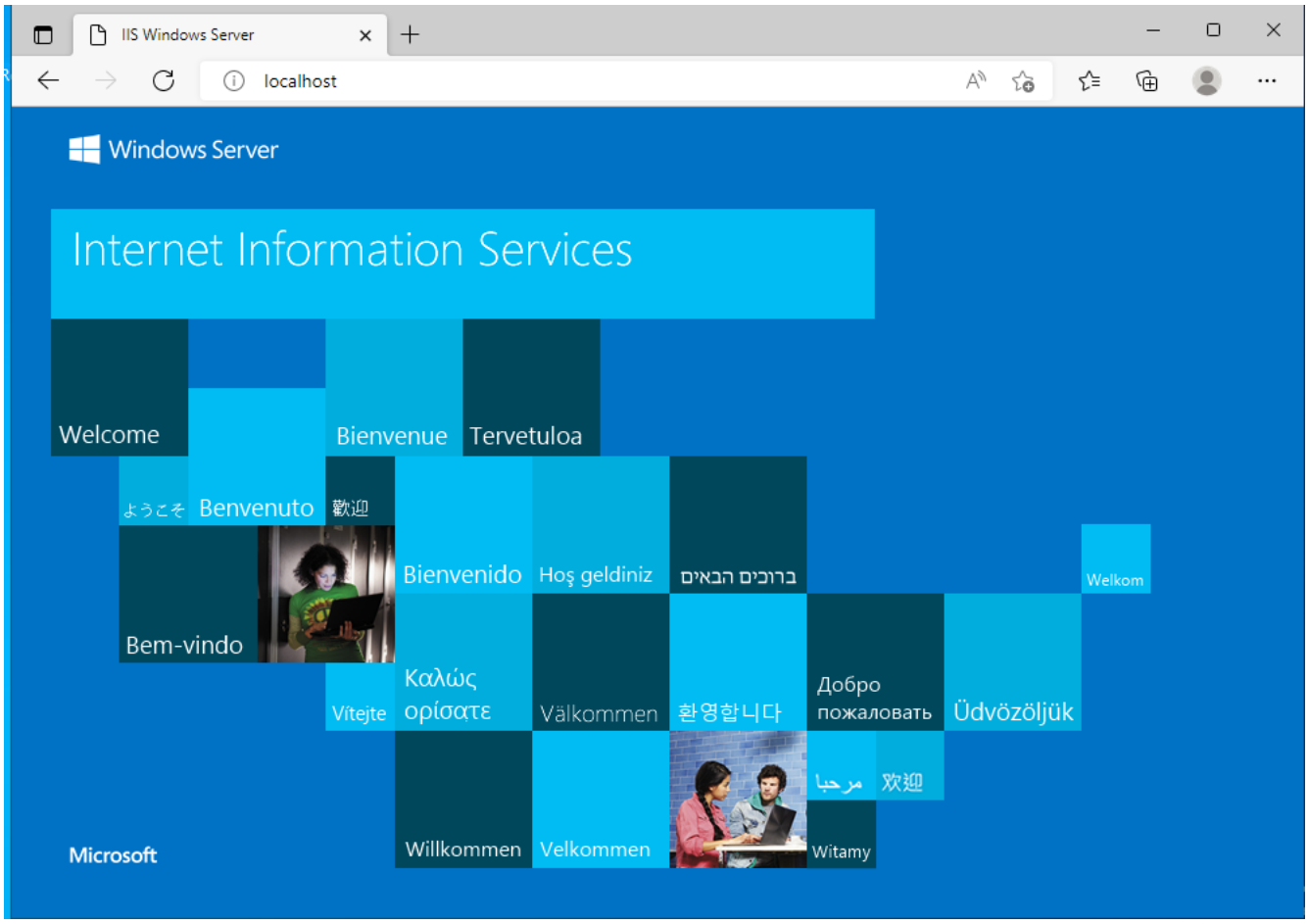


Click next. We can add features later on.



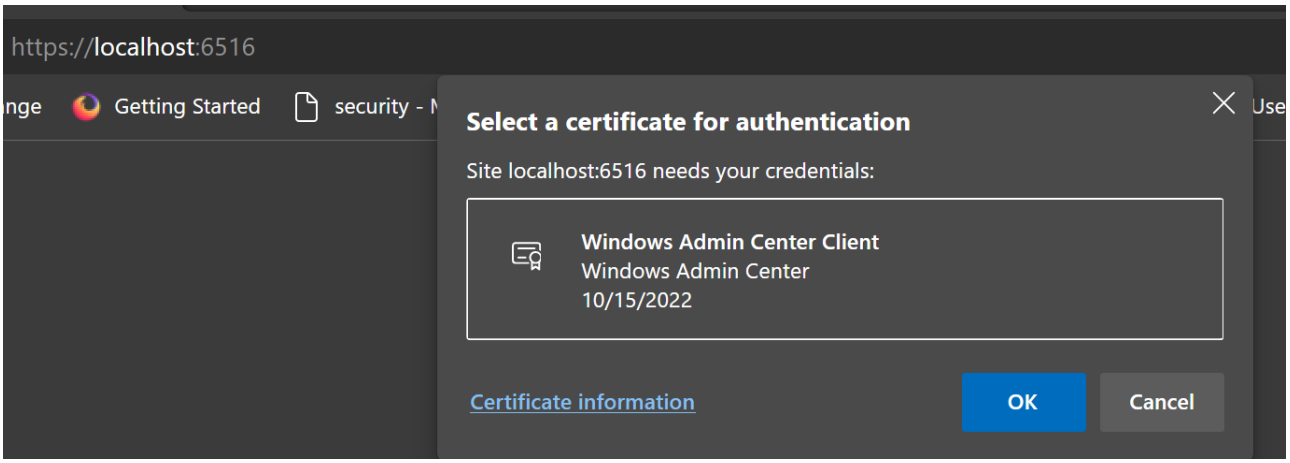


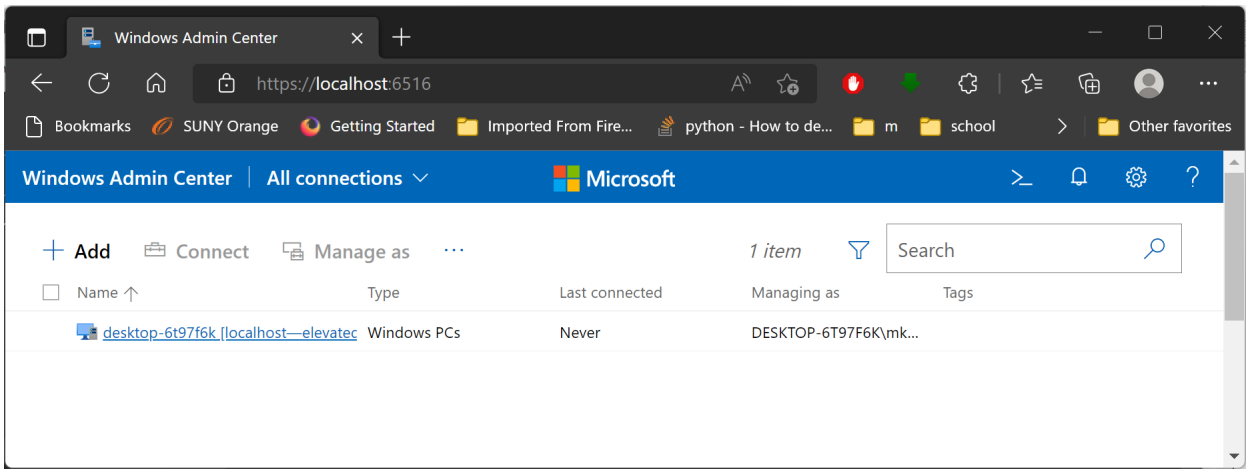
Click close when done.
Test that it is up and running.



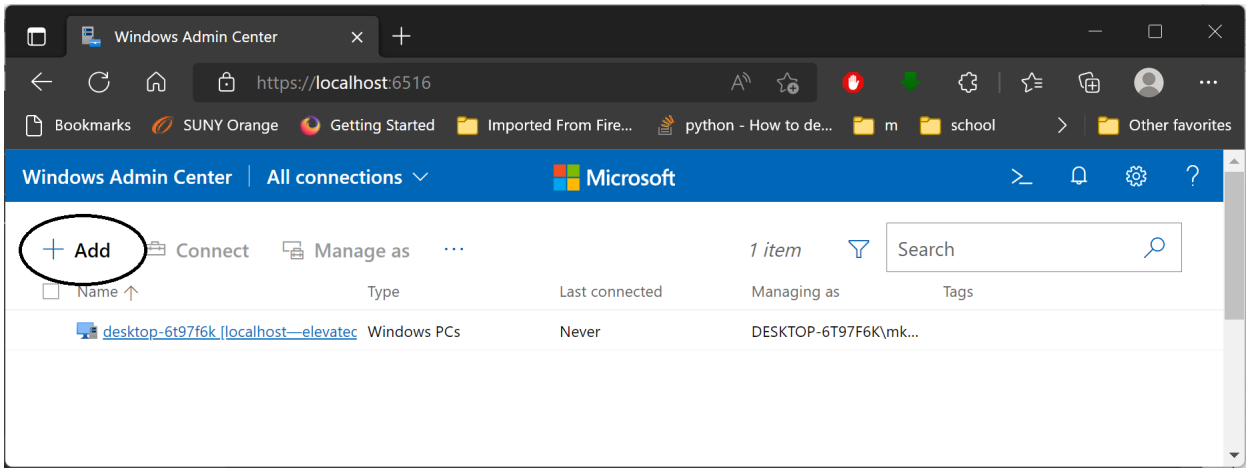
2.0.3.0 Windows IIS admin center install

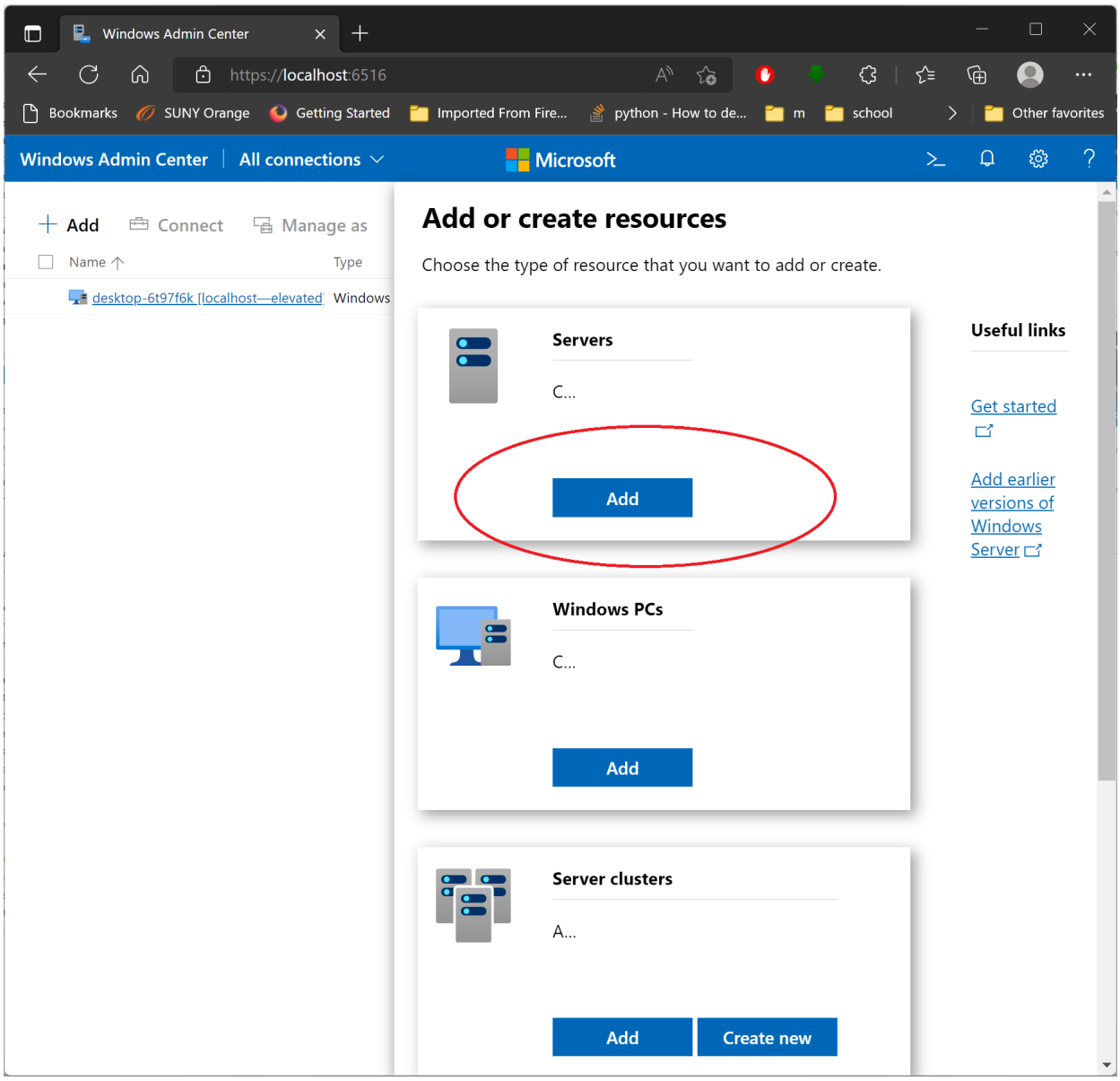
If administering a console only edition of Windows server or remote desktop connection is not possible. We can use Windows remote admin console.
https://www.microsoft.com/en-us/windows-server/windows-admin-center
After installing admin center open Edge browser to https:
localhost:6516
select credentials and click OK.

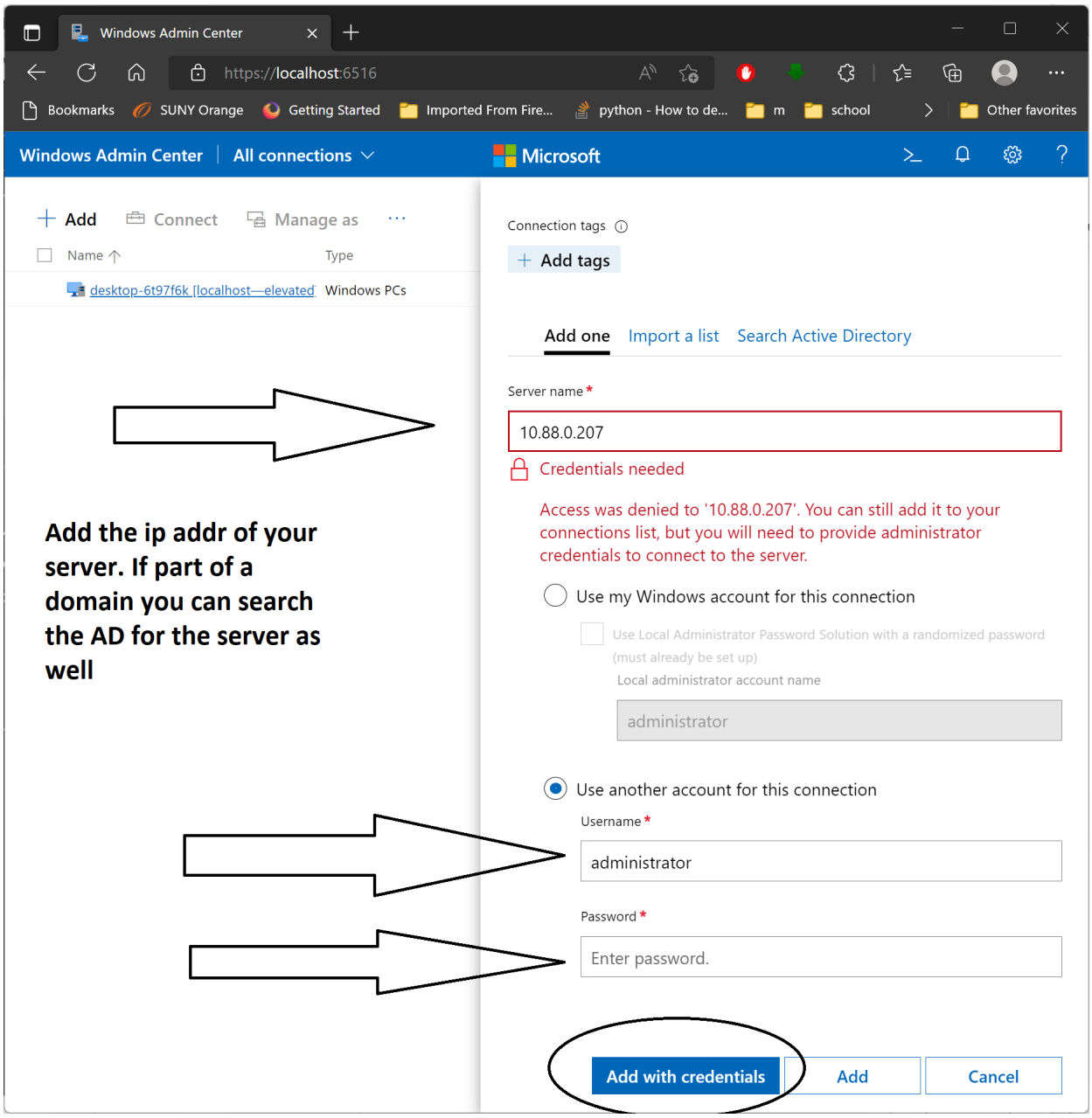

https://localhost:6516
Getting Started
security - N
Use
Select a certificate for authentication
Site localhost:6516 needs your credentials:
Windows Admin Center Client
Windows Admin Center
10/15/2022
Certificate information
OK
Cancel


Windows Admin Center
https://localhost:6516
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
All connections
Microsoft
Add
Connect
Manage as
1 item
Search
Name
Type
Last connected
Managing as
Tags
desktop-6t97f6k [localhost—elevatec
Windows PCs
Never
DESKTOP-6T97F6K\mk...

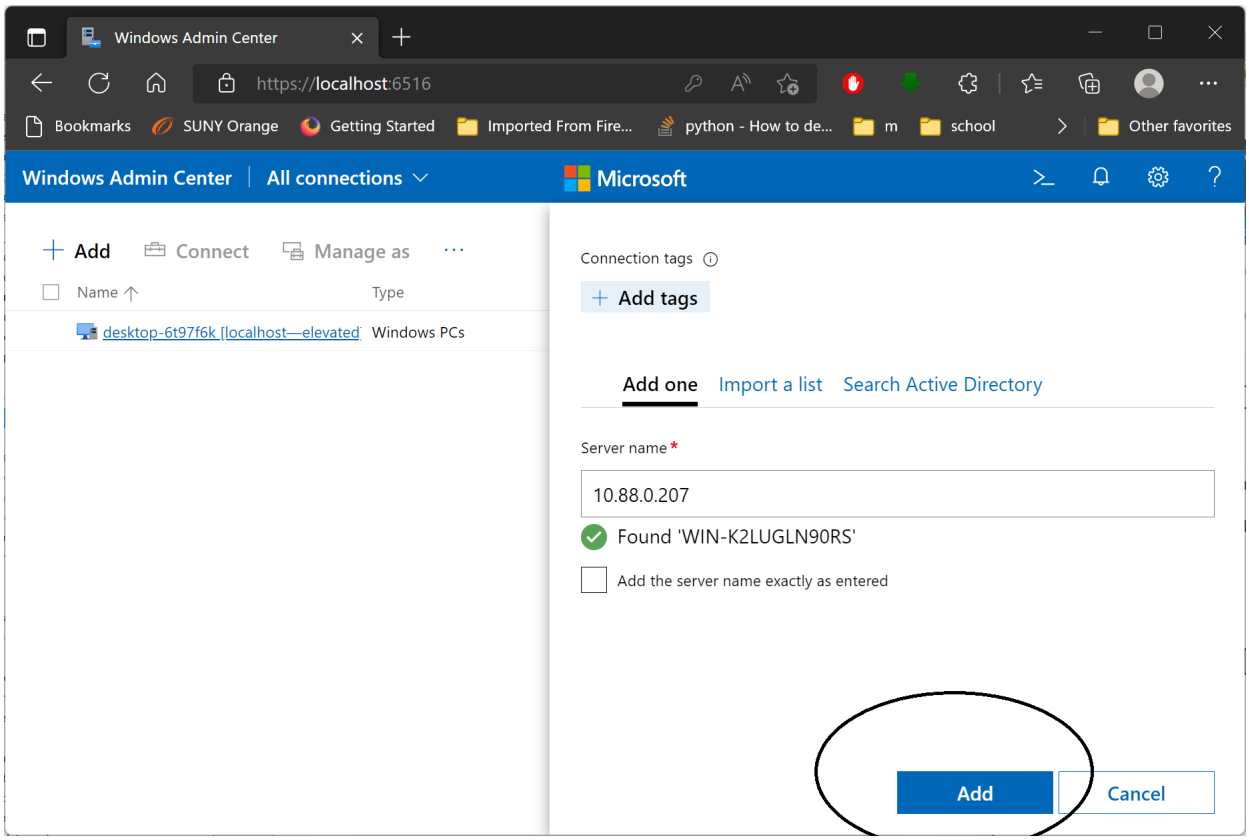
Click ADD.

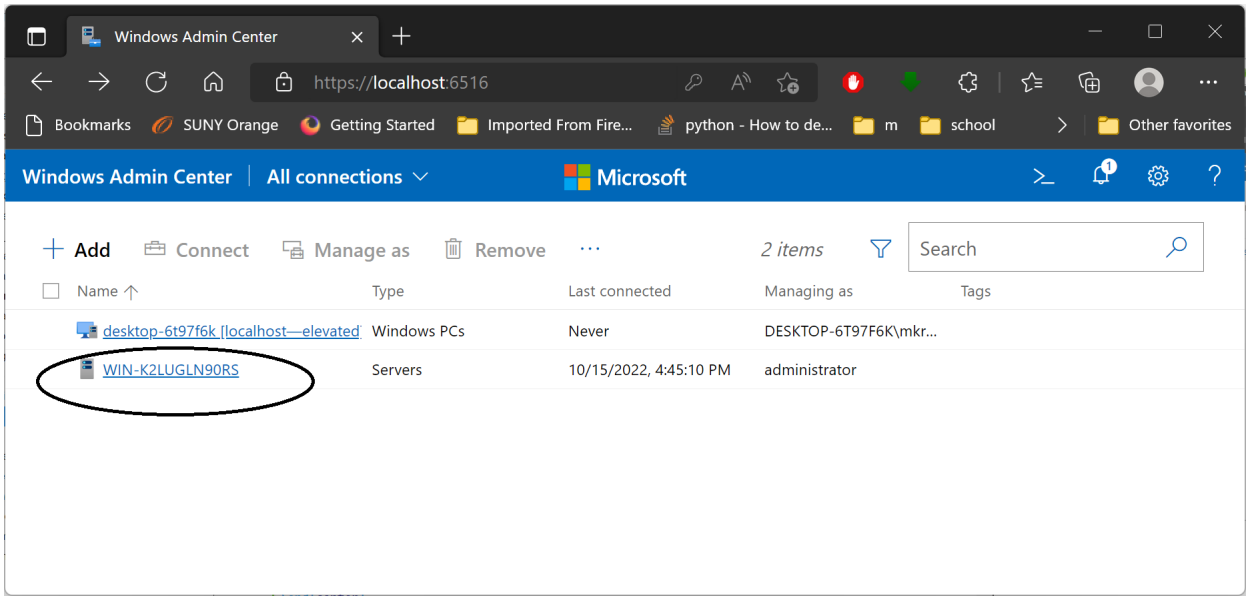

Windows Admin Center
https://localhost:6516
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
All connections
Microsoft
Add
Connect
Manage as
1 item
Search
Name
Type
Last connected
Managing as
Tags
desktop-6t97f6k [localhost—elevatec
Windows PCs
Never
DESKTOP-6T97F6K\mk...

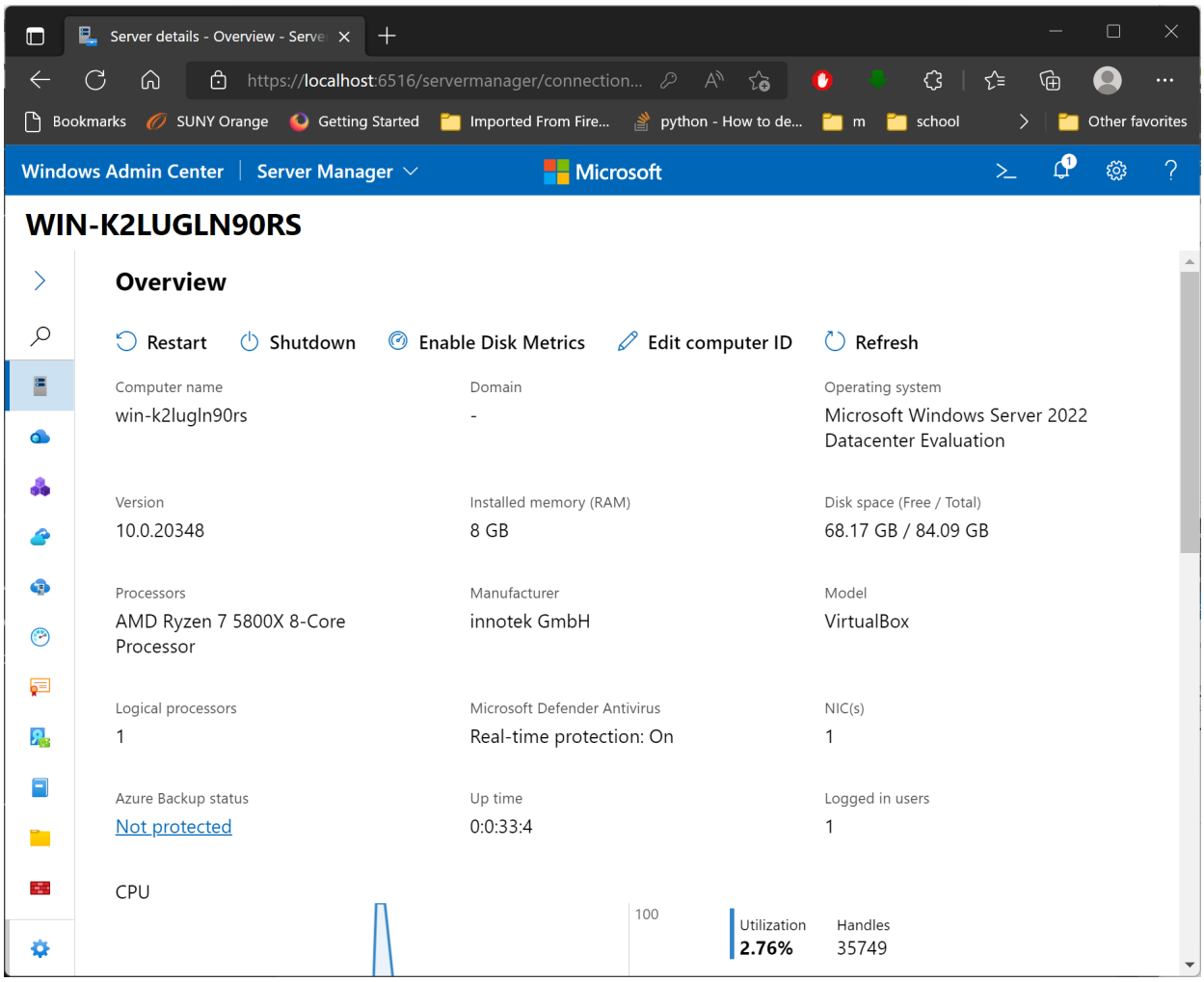

Windows Admin Center
https://localhost:6516
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
All connections
Microsoft
Add
Connect
Manage as
Name
Type
desktop-6t97f6k [localhost—elevated
Windows
Add or create resources
Choose the type of resource that you want to add or create.
Servers
C...
Add
Windows PCs
C...
Add
Server clusters
A...
Add
Create new
Useful links
Get started
Add earlier versions of Windows Server



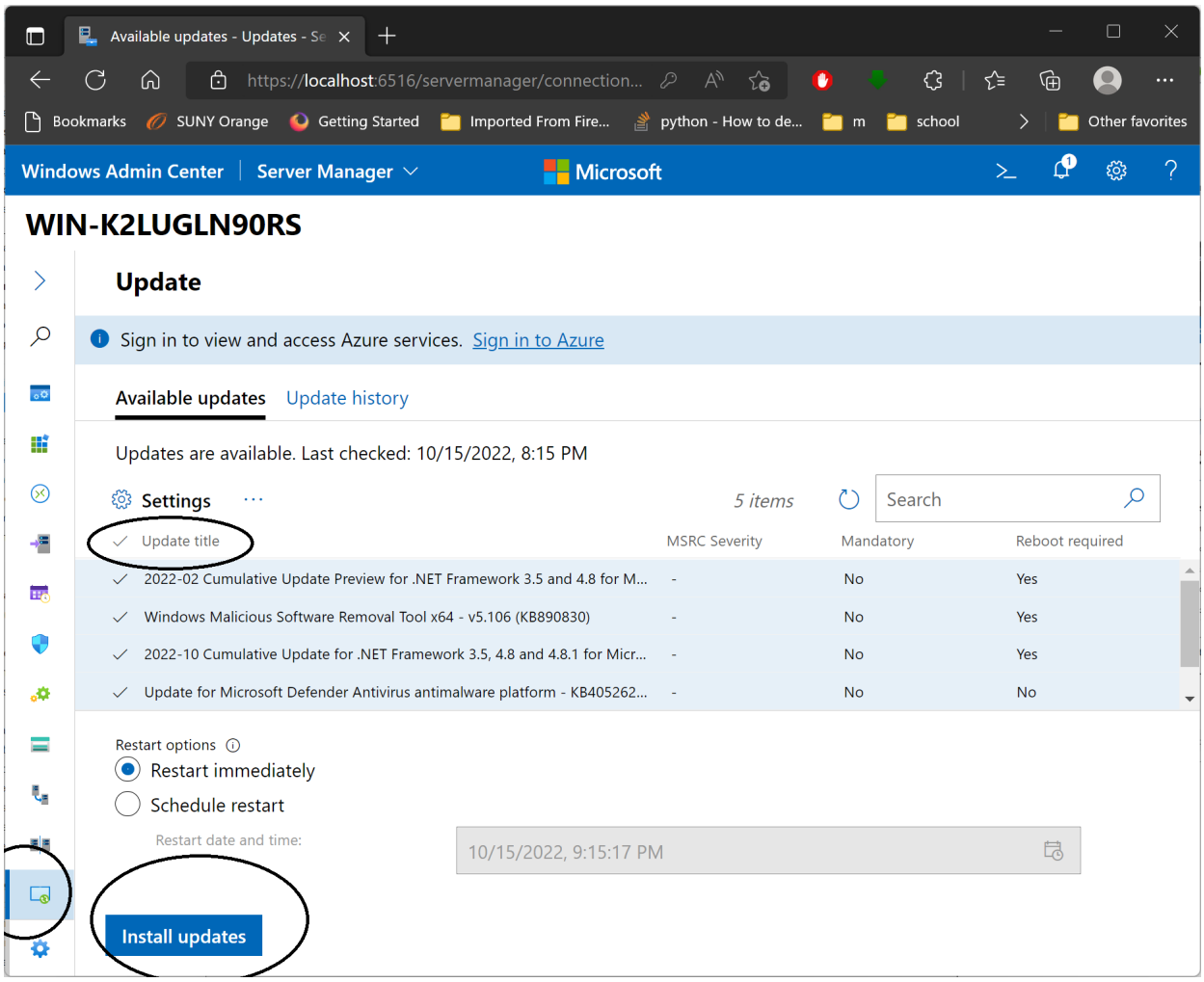
Click add with credentials.

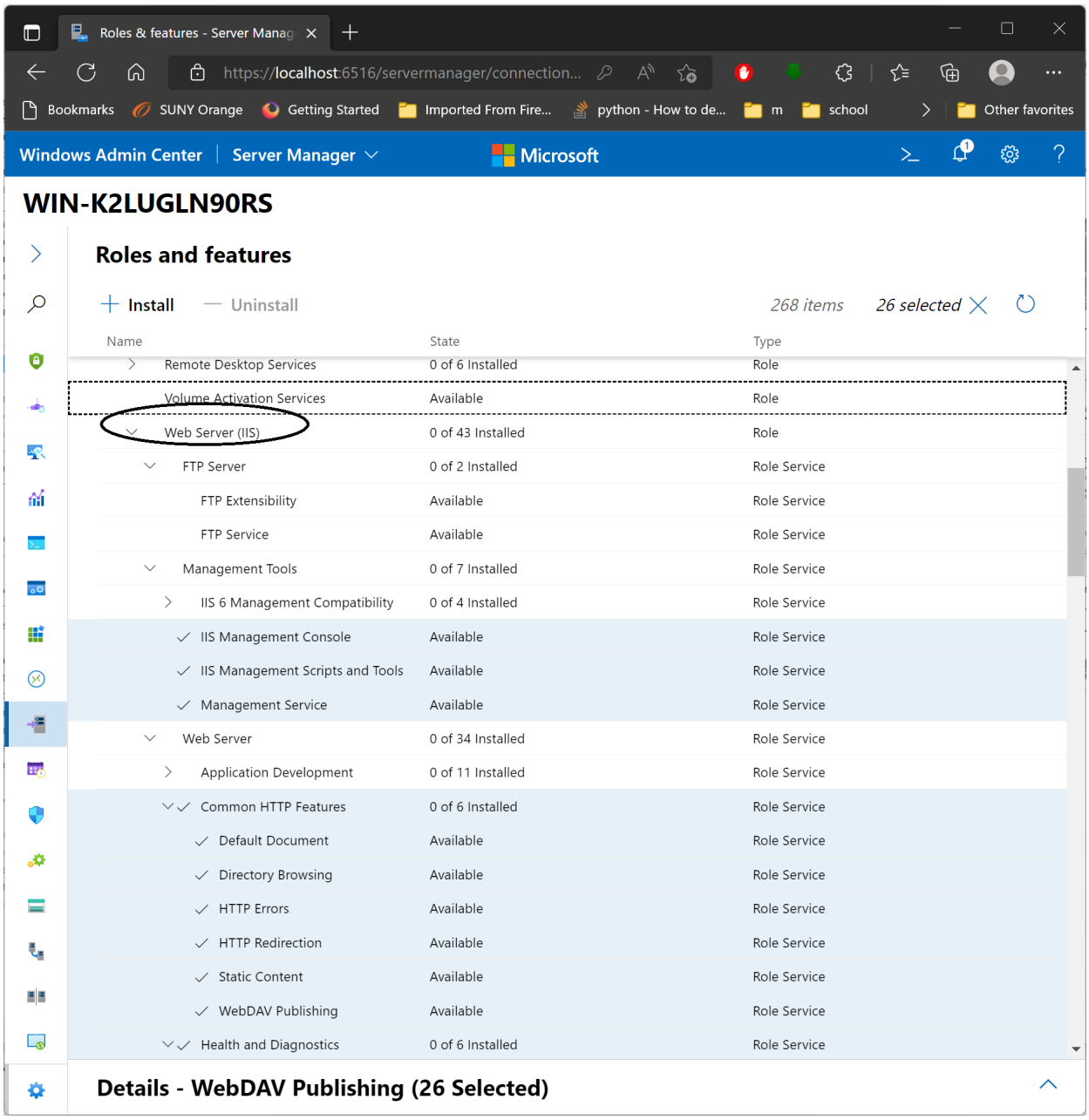

Windows Admin Center
https://localhost:6516
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
All connections
Microsoft
Add
Connect
Manage as
Name
Type
desktop-6t97f6k [localhost—elevated
Windows PCs
Connection tags
Add tags
Add one
Import a list
Search Active Directory
Server name *
10.88.0.207
Found 'WIN-K2LUGLN90RS'
Add the server name exactly as entered
Add
Cancel

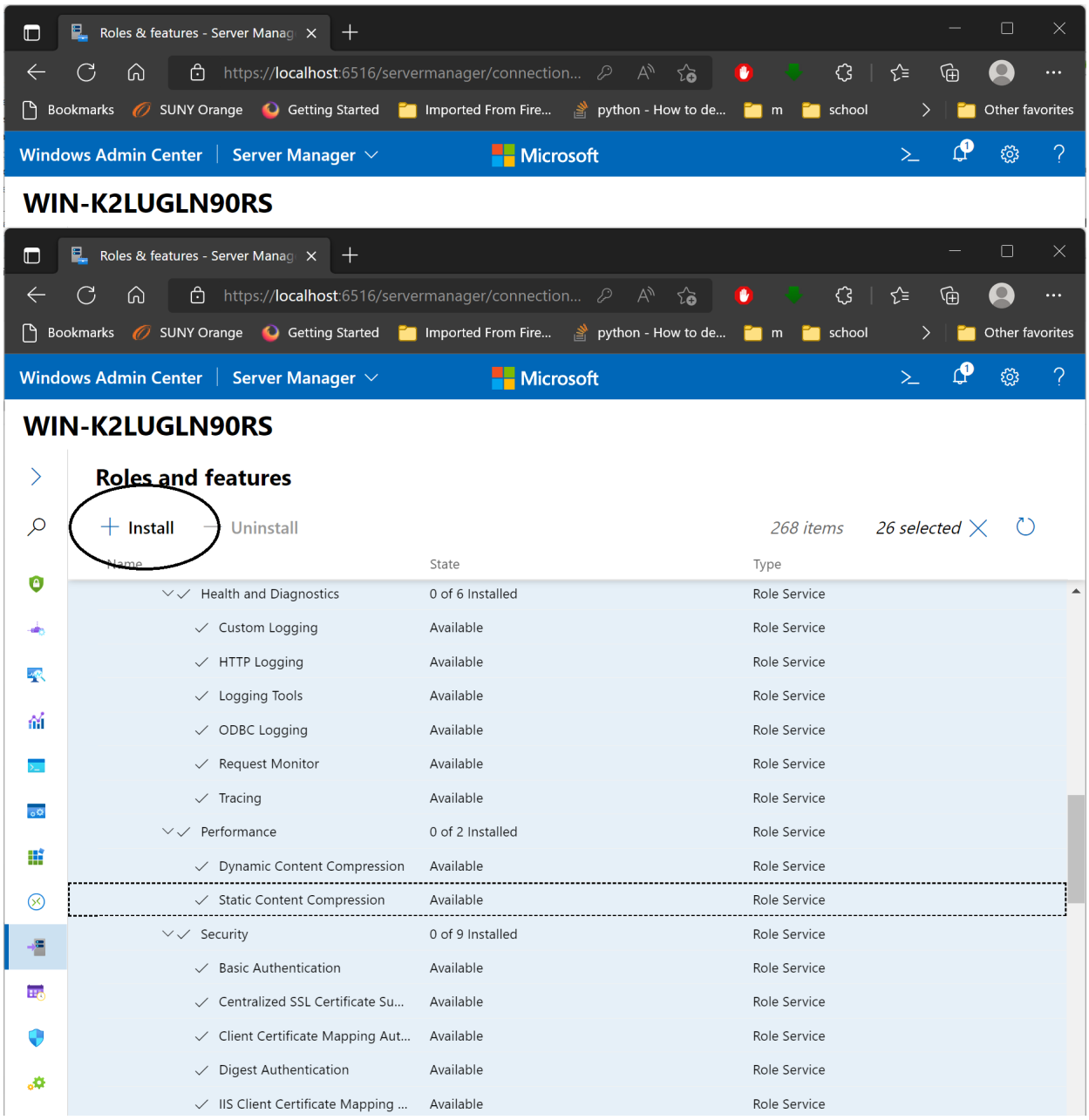

Windows Admin Center
https://localhost:6516
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
All connections
Microsoft
Add
Connect
Manage as
Remove
2 items
Search
Name
Type
Last connected
Managing as
Tags
desktop-6t97f6k [localhost—elevated
Windows PCs
Never
DESKTOP-6T97F6K\mkr...
WIN-K2LUGLN90RS
Servers
10/15/2022, 4:45:10 PM
administrator

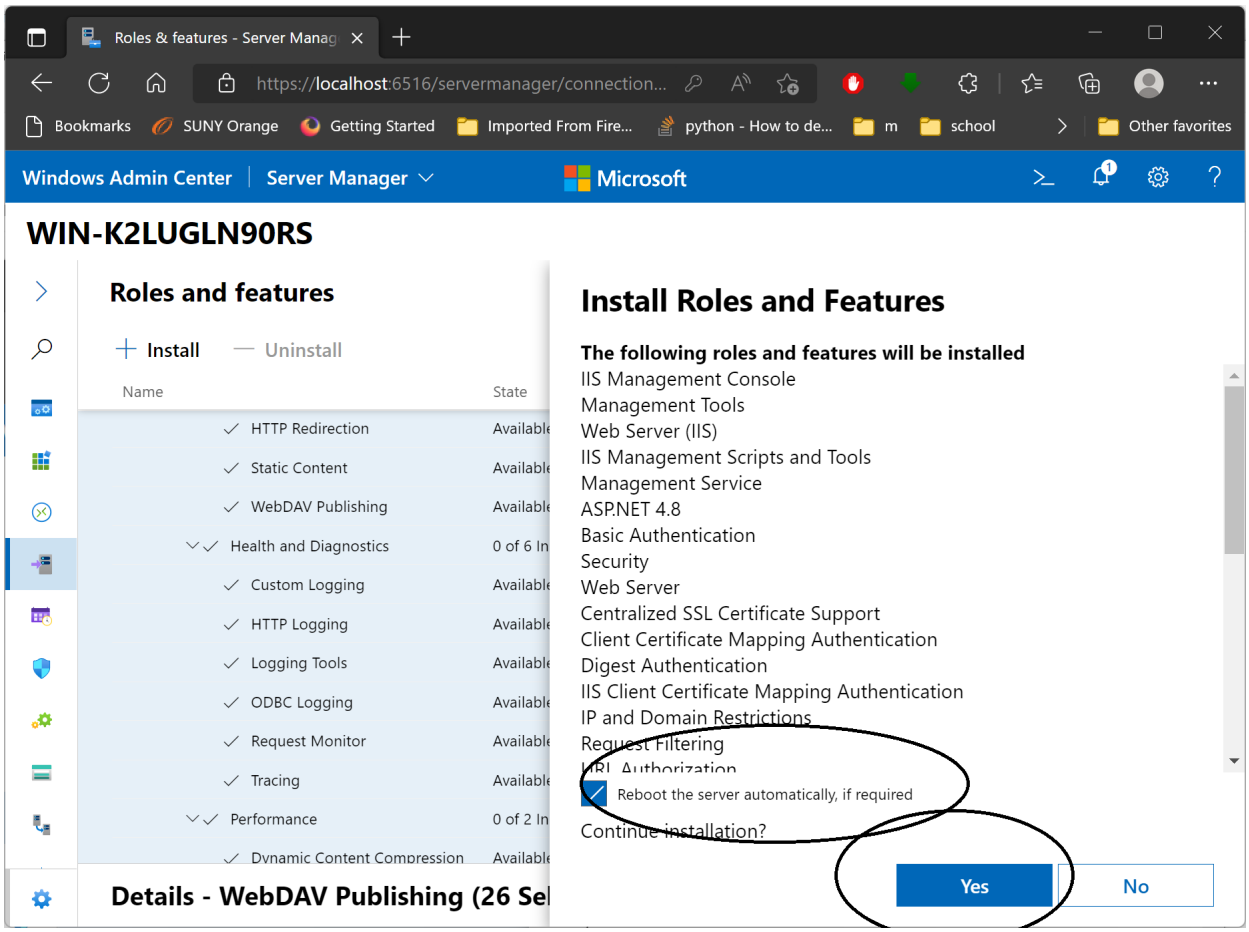


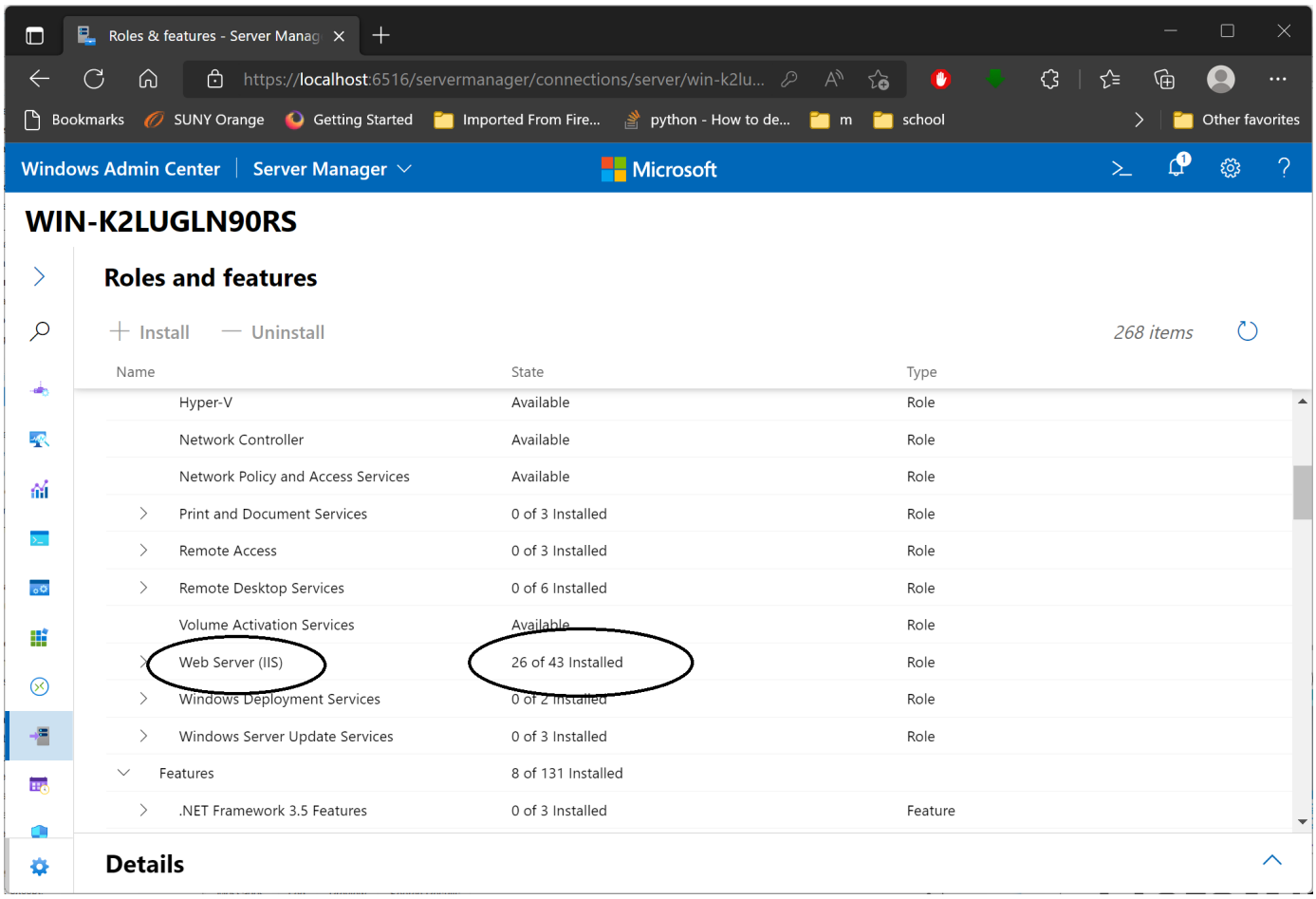
Install Updates.

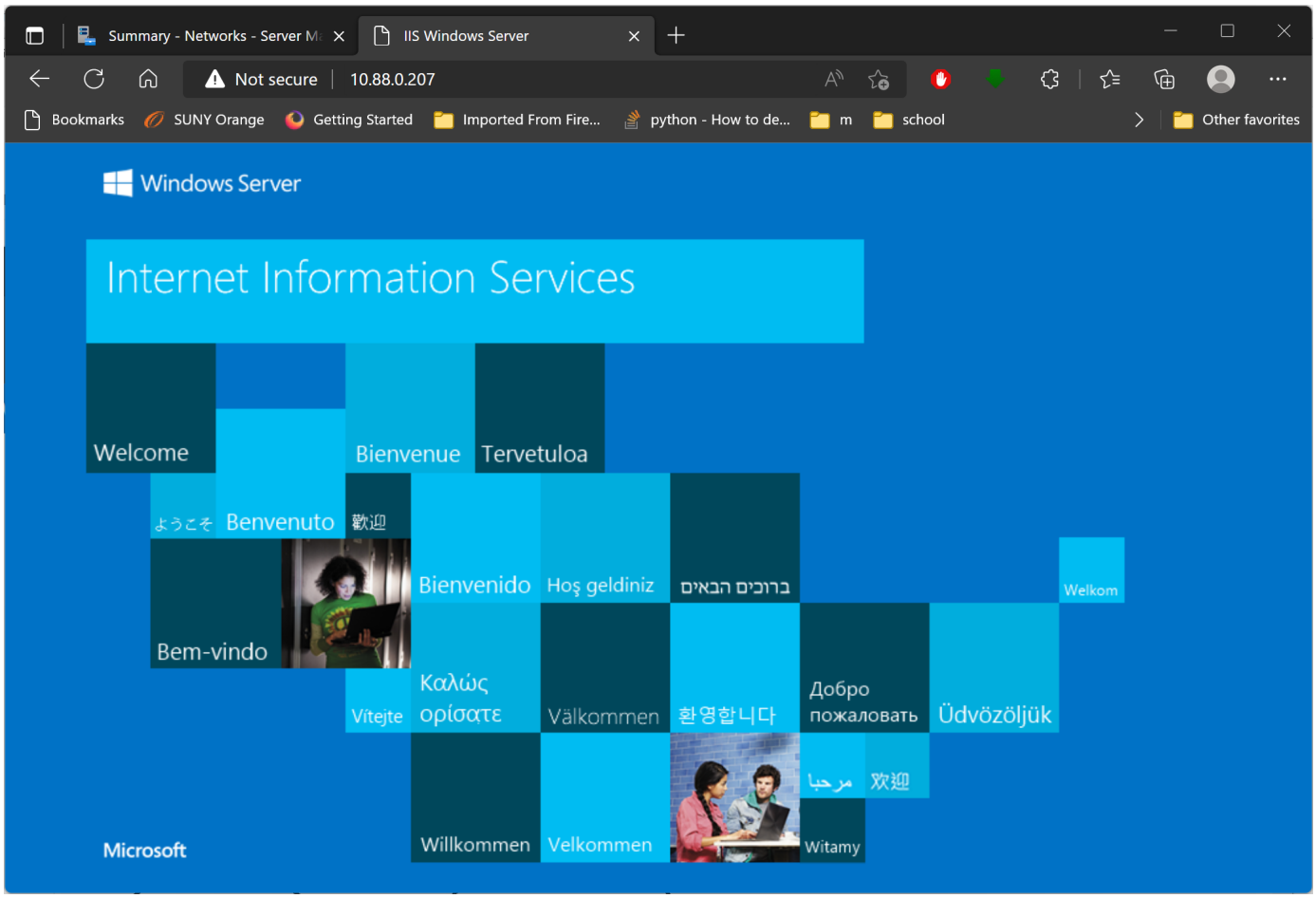

Available updates - Updates - Se
https://localhost:6516/servermanager/connection...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Update
Sign in to view and access Azure services. Sign in to Azure
Available updates
Update history
Updates are available. Last checked: 10/15/2022, 8:15 PM
Settings
5 items
Search
Update title
MSRC Severity
Mandatory
Reboot required
2022-02 Cumulative Update Preview for .NET Framework 3.5 and 4.8 for M... - No Yes
Windows Malicious Software Removal Tool x64 - v5.106 (KB890830) - No Yes
2022-10 Cumulative Update for .NET Framework 3.5, 4.8 and 4.8.1 for Micr... - No Yes
Update for Microsoft Defender Antivirus antimalware platform - KB405262... - No No
Restart options
Restart immediately
Schedule restart
Restart date and time:
10/15/2022, 9:15:17 PM
Install updates


Roles & features - Server Manag
https://localhost:6516/servermanager/connection...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Roles and features
Install
Uninstall
268 items
26 selected
Name
State
Type
Remote Desktop Services
0 of 6 Installed
Role
Volume Activation Services
Available
Role
Web Server (IIS)
0 of 43 Installed
Role
FTP Server
0 of 2 Installed
Role Service
FTP Extensibility
Available
Role Service
FTP Service
Available
Role Service
Management Tools
0 of 7 Installed
Role Service
IIS 6 Management Compatibility
0 of 4 Installed
Role Service
IIS Management Console
Available
Role Service
IIS Management Scripts and Tools
Available
Role Service
Management Service
Available
Role Service
Web Server
0 of 34 Installed
Role Service
Application Development
0 of 11 Installed
Role Service
Common HTTP Features
0 of 6 Installed
Role Service
Default Document
Available
Role Service
Directory Browsing
Available
Role Service
HTTP Errors
Available
Role Service
HTTP Redirection
Available
Role Service
Static Content
Available
Role Service
WebDAV Publishing
Available
Role Service
Health and Diagnostics
0 of 6 Installed
Role Service
Details - WebDAV Publishing (26 Selected)

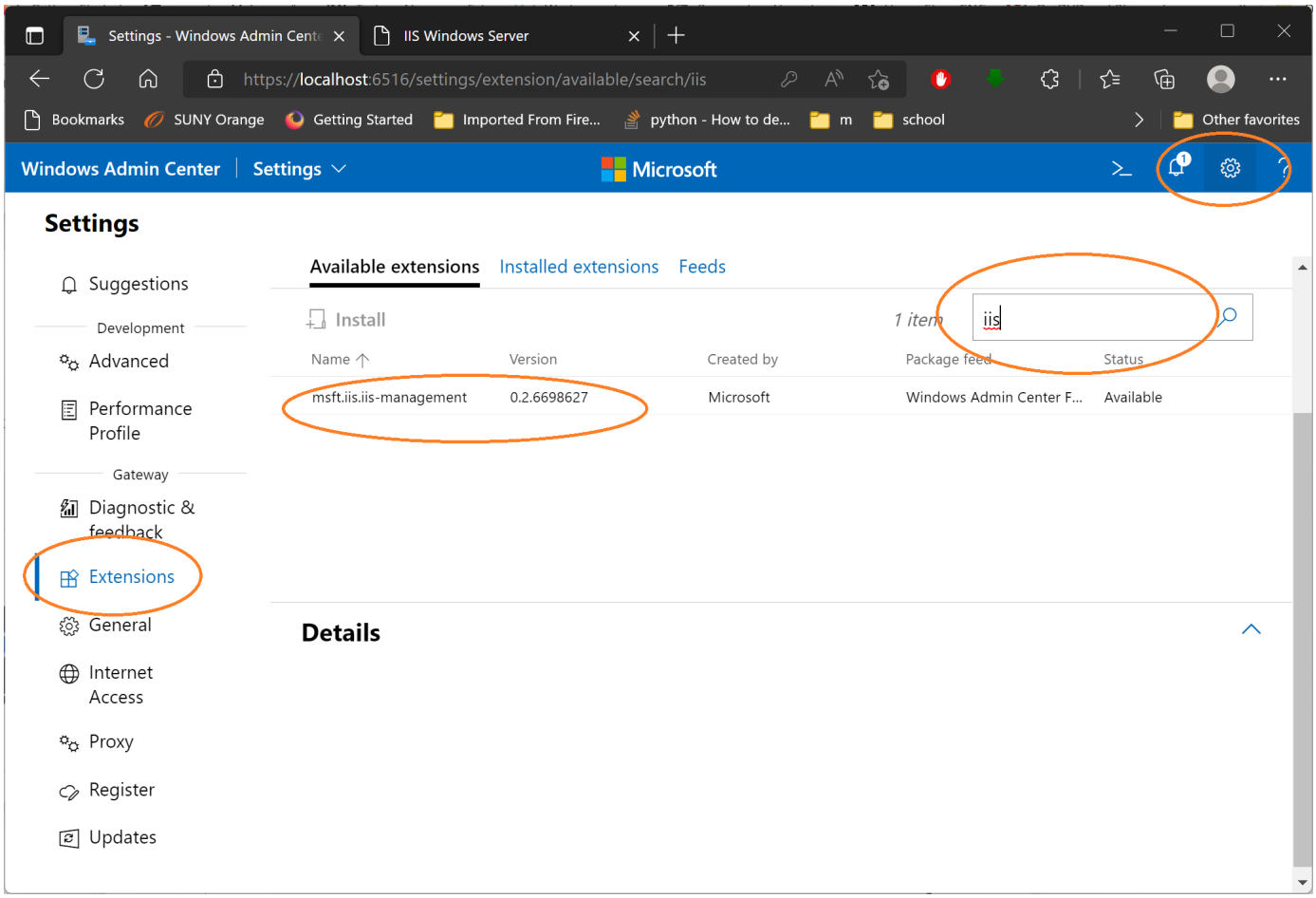

Roles & features - Server Manag
https://localhost:6516/servermanager/connection...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Roles and features
Install
Uninstall
268 items
26 selected
Name
State
Type
Health and Diagnostics
0 of 6 Installed
Role Service
Custom Logging
Available
Role Service
HTTP Logging
Available
Role Service
Logging Tools
Available
Role Service
ODBC Logging
Available
Role Service
Request Monitor
Available
Role Service
Tracing
Available
Role Service
Performance
0 of 2 Installed
Role Service
Dynamic Content Compression
Available
Role Service
Static Content Compression
Available
Role Service
Security
0 of 9 Installed
Role Service
Basic Authentication
Available
Role Service
Centralized SSL Certificate Su...
Available
Role Service
Client Certificate Mapping Aut...
Available
Role Service
Digest Authentication
Available
Role Service
IIS Client Certificate Mapping ...
Available
Role Service

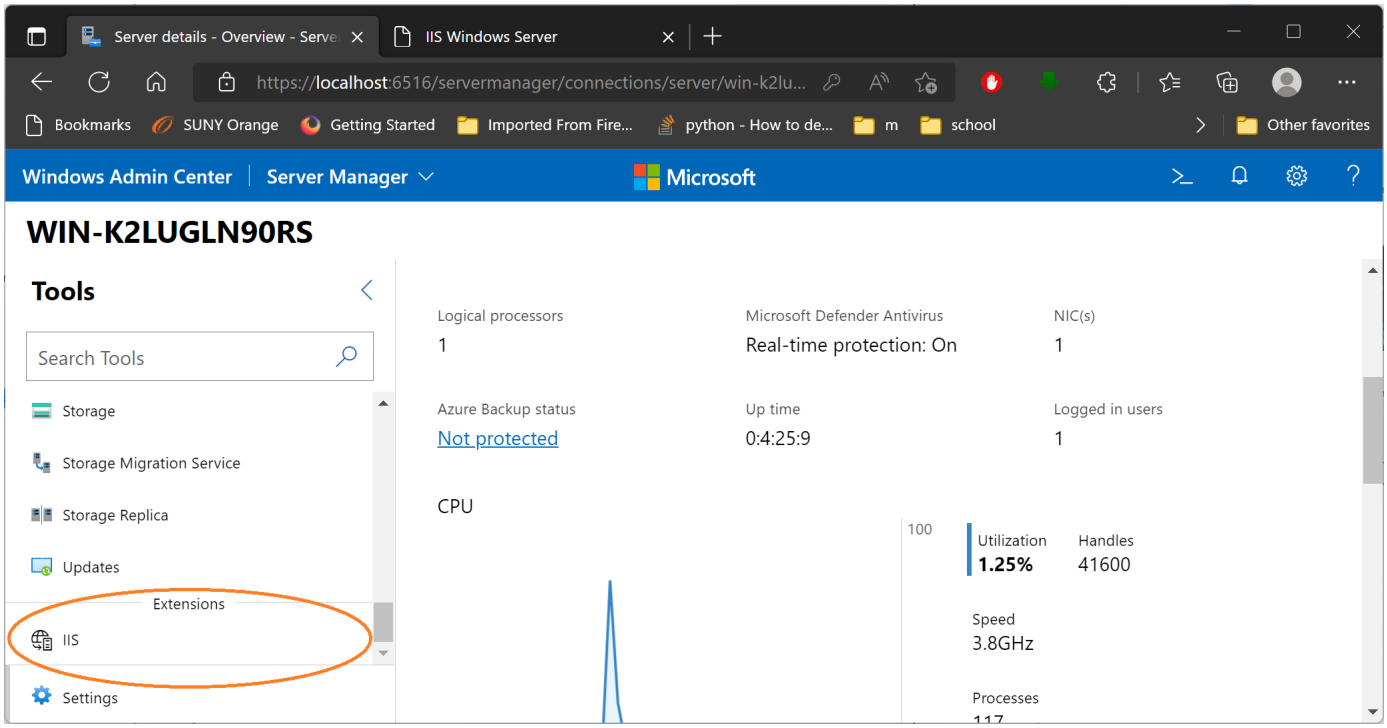

Roles & features - Server Manag
https://localhost:6516/servermanager/connection...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Roles and features
Install
Uninstall
Name
State
HTTP Redirection
Static Content
WebDAV Publishing
Health and Diagnostics
0 of 6 In
Custom Logging
HTTP Logging
Logging Tools
ODBC Logging
Request Monitor
Tracing
Performance
0 of 2 In
Dynamic Content Compression
Details - WebDAV Publishing (26 Sel
Install Roles and Features
The following roles and features will be installed
IIS Management Console
Management Tools
Web Server (IIS)
IIS Management Scripts and Tools
Management Service
ASP.NET 4.8
Basic Authentication
Security
Web Server
Centralized SSL Certificate Support
Client Certificate Mapping Authentication
Digest Authentication
IIS Client Certificate Mapping Authentication
IP and Domain Restrictions
Request Filtering
Reboot the server automatically, if required
Continue installation?
Yes
No

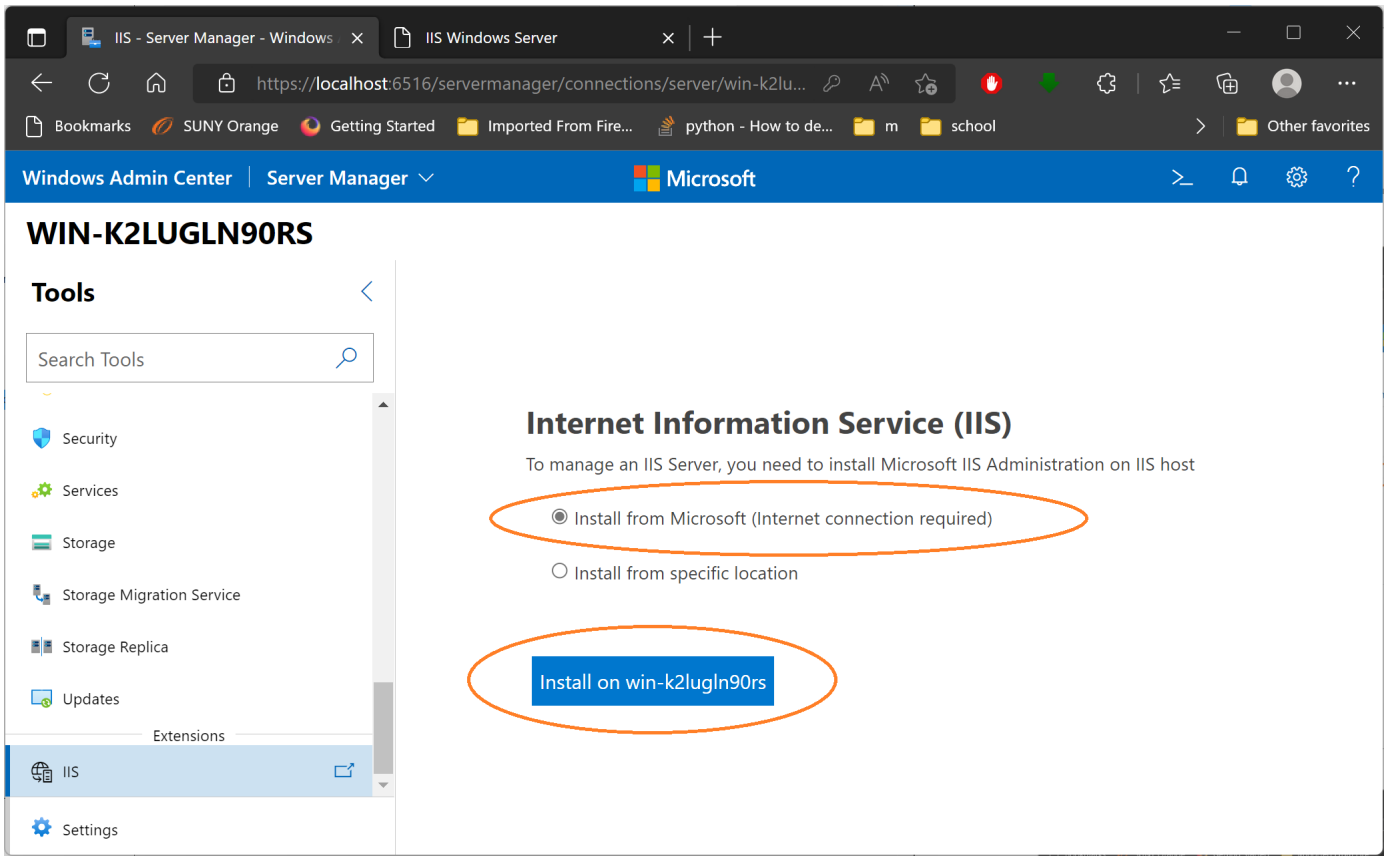

Roles & features - Server Manag
https://localhost:6516/servermanager/connections/server/win-k2lu...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Roles and features
Install
Uninstall
268 items
Name
State
Type
Hyper-V
Available
Role
Network Controller
Available
Role
Network Policy and Access Services
Available
Role
Print and Document Services
0 of 3 Installed
Role
Remote Access
0 of 3 Installed
Role
Remote Desktop Services
0 of 6 Installed
Role
Volume Activation Services
Available
Role
Web Server (IIS)
26 of 43 Installed
Role
Windows Deployment Services
0 of 2 Installed
Role
Windows Server Update Services
0 of 3 Installed
Role
Features
8 of 131 Installed
.NET Framework 3.5 Features
0 of 3 Installed
Feature
Details

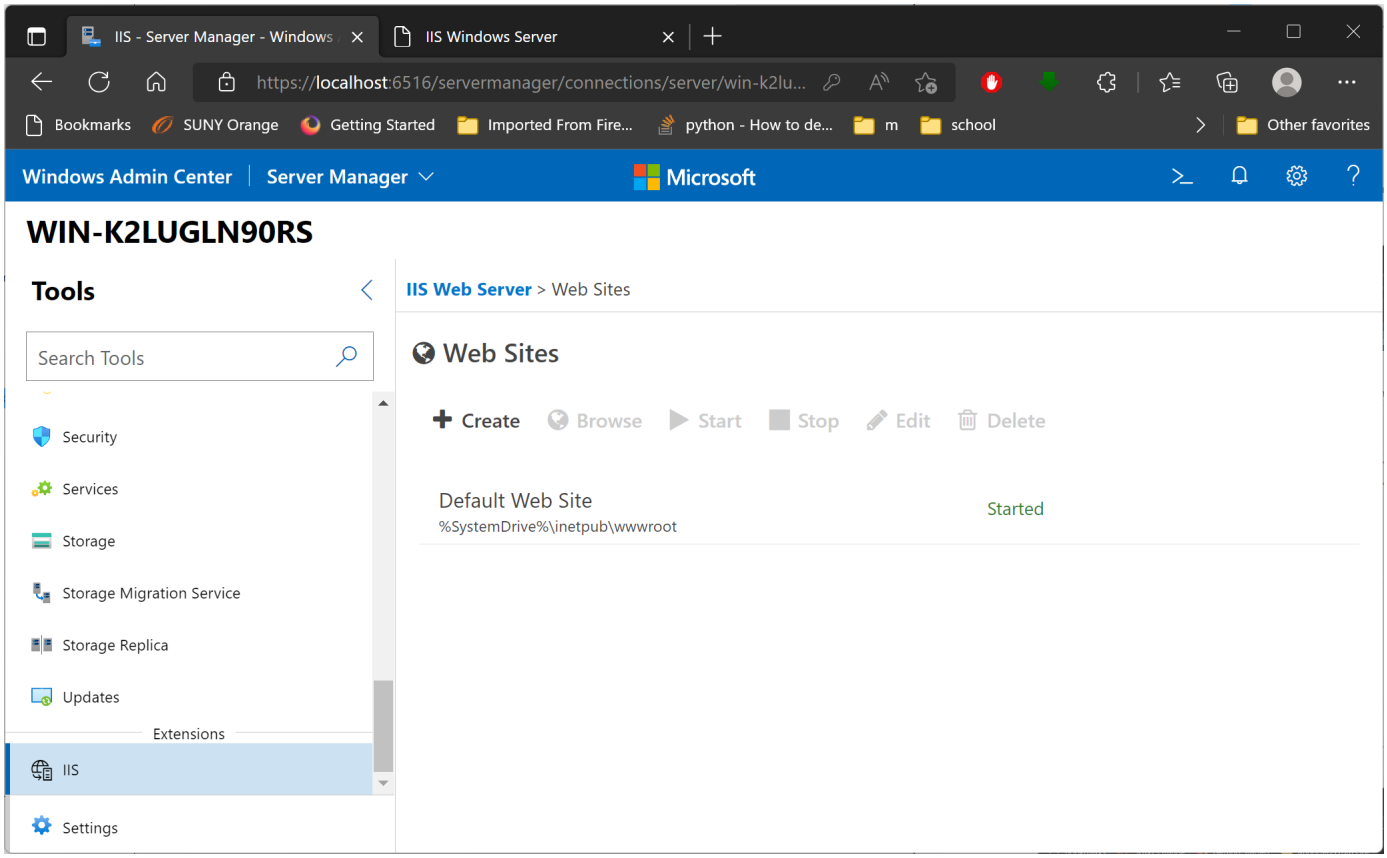


install admin center iis extensions.

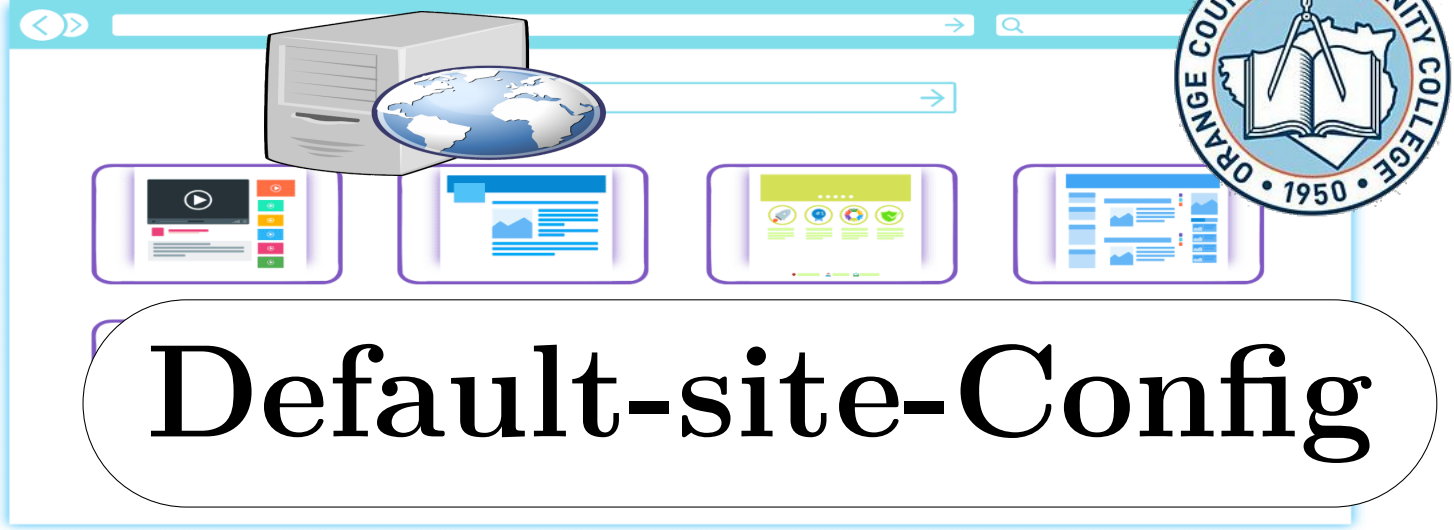

Settings - Windows Admin Cente
IIS Windows Server
https://localhost:6516/settings/extension/available/search/iis
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Settings
Microsoft
Settings
Suggestions
Development
Advanced
Performance Profile
Gateway
Diagnostic & feedback
Extensions
General
Internet Access
Proxy
Register
Updates
Available extensions
Installed extensions
Feeds
Install
1 item
iis
Name
Version
Created by
Package feed
Status
msft.iis.iis-management
0.2.6698627
Microsoft
Windows Admin Center F...
Available
Details


Server details - Overview - Serve
IIS Windows Server
https://localhost:6516/servermanager/connections/server/win-k2lu...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Tools
Search Tools
Storage
Storage Migration Service
Storage Replica
Updates
Extensions
IIS
Settings
Logical processors
1
Microsoft Defender Antivirus
Real-time protection: On
NIC(s)
1
Azure Backup status
Not protected
Up time
0:4:25:9
Logged in users
1
CPU
100
Utilization
1.25%
Handles
41600
Speed
3.8GHz
Processes


IIS - Server Manager - Windows
IIS Windows Server
https://localhost:6516/servermanager/connections/server/win-k2lu...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Tools
Search Tools
Security
Services
Storage
Storage Migration Service
Storage Replica
Updates
Extensions
IIS
Settings
Internet Information Service (IIS)
To manage an IIS Server, you need to install Microsoft IIS Administration on IIS host
Install from Microsoft (Internet connection required)
Install from specific location
Install on win-k2lugln90rs


IIS - Server Manager - Windows
IIS Windows Server
https://localhost:6516/servermanager/connections/server/win-k2lu...
Bookmarks
SUNY Orange
Getting Started
Imported From Fire...
python - How to de...
m
school
Other favorites
Windows Admin Center
Server Manager
Microsoft
WIN-K2LUGLN90RS
Tools
Search Tools
Security
Services
Storage
Storage Migration Service
Storage Replica
Updates
Extensions
IIS
Settings
IIS Web Server > Web Sites
Web Sites
Create
Browse
Start
Stop
Edit
Delete
Default Web Site
%SystemDrive%\inetpub\wwwroot
Started

CIT 215



Default-site-Config

The Default Website is what its name says, a convenience website that is created when the webserver is installed. It can be deleted and replaced by new websites, but its advantage is that it will work "out-of-the-box".
Also the default website is the one that will be accessed by the server ip address. Other domains or virtual websites have to be accessed by the the domain name even if that domain is sharing the same ip's.

3.0.1 Apache web server

By default Apache on Ubuntu has its default webroot in:
/var/www/html

Commands

```
occc@occc-VirtualBox:/var/www/html$ pwd
/var/www/html
occc@occc-VirtualBox:/var/www/html$ ls
index.html
occc@occc-VirtualBox:/var/www/html$
```

We might want to change that to someplace else for security and perhaps to locate it to larger hard drive(mount point).
All the config files for apache2 on most linux OS's is located in the **/etc/apache2** directory.

Commands

```
occc@occc-VirtualBox:/var/www/html$ cd /etc/apache2/
occc@occc-VirtualBox:/etc/apache2$ ls -l
total 80
-rw-r--r-- 1 root root  7224 Mar 24  2022 apache2.conf
drwxr-xr-x 2 root root  4096 Jul 15 02:34 conf-available
drwxr-xr-x 2 root root  4096 Apr  3  2022 conf-enabled
-rw-r--r-- 1 root root  1782 Mar 22  2022 envvars
-rw-r--r-- 1 root root 31063 Mar 22  2022 magic
drwxr-xr-x 2 root root 12288 Jul 15 02:34 mods-available
drwxr-xr-x 2 root root  4096 Apr  3  2022 mods-enabled
-rw-r--r-- 1 root root   320 Mar 22  2022 ports.conf
drwxr-xr-x 2 root root  4096 Jul 15 02:34 sites-available
drwxr-xr-x 2 root root  4096 Apr  3  2022 sites-enabled
occc@occc-VirtualBox:/etc/apache2$
```

For example we might want the default website or any other websites after be located in /webroot directory.
Create the directory by issuing the following commands:

Commands

```
occc@occc-VirtualBox:/etc/apache2$ sudo mkdir /webroot
[sudo] password for occc:
occc@occc-VirtualBox:/etc/apache2$ sudo chown root:root /webroot
occc@occc-VirtualBox:/etc/apache2$ sudo chmod 755 /webroot
occc@occc-VirtualBox:/etc/apache2$ ls -l / | grep webroot
drwxr-xr-x   2 root root       4096 Oct 16 19:07 webroot
occc@occc-VirtualBox:/etc/apache2$
```

Change to the /etc/apache2/sites-enabled directory. Then edit the 000-default.conf file.

Commands

```
occc@occc-VirtualBox:/etc/apache2/sites-enabled$ pwd
/etc/apache2/sites-enabled
occc@occc-VirtualBox:/etc/apache2/sites-enabled$ ls -l
total 0
lrwxrwxrwx 1 root root 35 Apr  3  2022 000-
default.conf -> ../sites-available/000-default.conf
```

```
occc@occc-VirtualBox:/etc/apache2/sites-enabled$
```

For now just change the DocumentRoot to **webroot** /.
original: **DocumentRoot /var/www/html**
change to: **DocumentRoot /webroot**
Also add.

add

```
<Directory /webroot/>
   Options Indexes FollowSymLinks
   AllowOverride None
   Require all granted
</Directory>\\
```

Edit

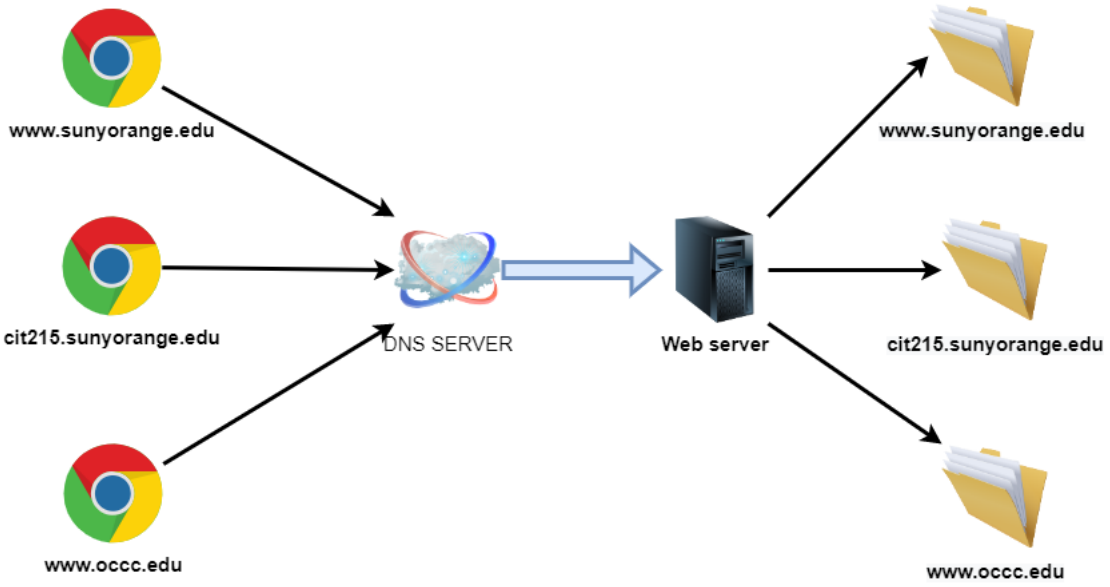
```
sudo vim 000-default.conf
```

basic apache conf file

```
1  <VirtualHost *:80>
2  # The ServerName directive sets the request scheme,
   ↪   hostname and port that
3  # the server uses to identify itself. This is used
   ↪  when creating
4  # redirection URLs. In the context of virtual hosts
   ↪  , the ServerName
5  # specifies what hostname must appear in the
   ↪  request's Host: header to
6  # match this virtual host. For the default virtual
   ↪  host (this file) this
7  # value is not decisive as it is used as a last
   ↪  resort host regardless.
8  # However, you must set it for any further virtual
   ↪  host explicitly.
9  #ServerName www.example.com
10
11 ServerAdmin webmaster@localhost
12 DocumentRoot /webroot
13 <Directory /webroot/>
14    Options Indexes FollowSymLinks
```

```
    AllowOverride None
    Require all granted
</Directory>


# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

VirtualHost allows us to have a separate configuration per virtual web-server running on this physical machine.
https://httpd.apache.org/docs/2.4/vhosts/



ServerAdmin The ServerAdmin sets the contact address that the server includes in any error messages it returns to the client.
DocumentRoot This directive sets the directory from which httpd will serve files. Unless matched by a directive like Alias, the server appends the path from the requested URL to the document root to make the path to the document. Example:

DocumentRoot "/webroot" then access to http://my.example.com/index.html refers to /webroot/index.html. If the directory path is not absolute, it is assumed to be relative to the ServerRoot.

The DocumentRoot should be specified without a trailing slash.
Directory https://httpd.apache.org/docs/2.4/mod/core.htm#directory
Options for directory https://httpd.apache.org/docs/2.4/mod/core.html#options

Save the file. Add some index.html code into the /webroot directory.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CIT 215</title>
</head>
<body>

<h1>CIT 215</h1>
<p>SUNY Orange</p>

</body>
</html>
```

Commands

```
occc@occc-VirtualBox:/etc/apache2/sites-enabled$ sudo su -
root@occc-VirtualBox:~# cd /webroot
root@occc-VirtualBox:/webroot# vi index.html
root@occc-VirtualBox:/webroot# ls -l
total 4
-rw-r--r-- 1 root root 240 Oct 16 19:18 index.html
root@occc-VirtualBox:/webroot#
```

Lets test our configuration to make sure nothing is wrong with the conf file.

Commands

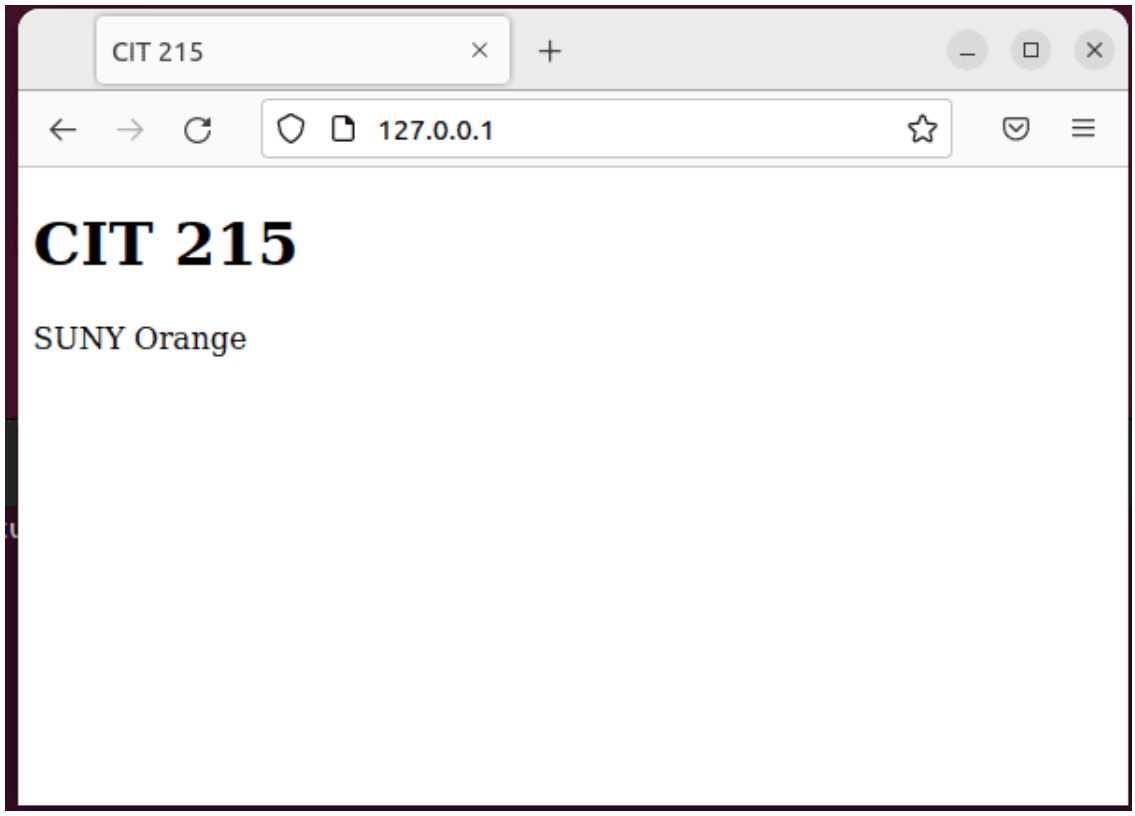
```
occc@occc-VirtualBox:/etc/apache2/sites-
enabled$ apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qual
Syntax OK
occc@occc-VirtualBox:/etc/apache2/sites-enabled$
```

For now ignore the FQDN warning.
The config file is correctly formatted if we get Syntax OK. Reload the config and test the status of the web server.
reload the config and tests the status of the webserver.

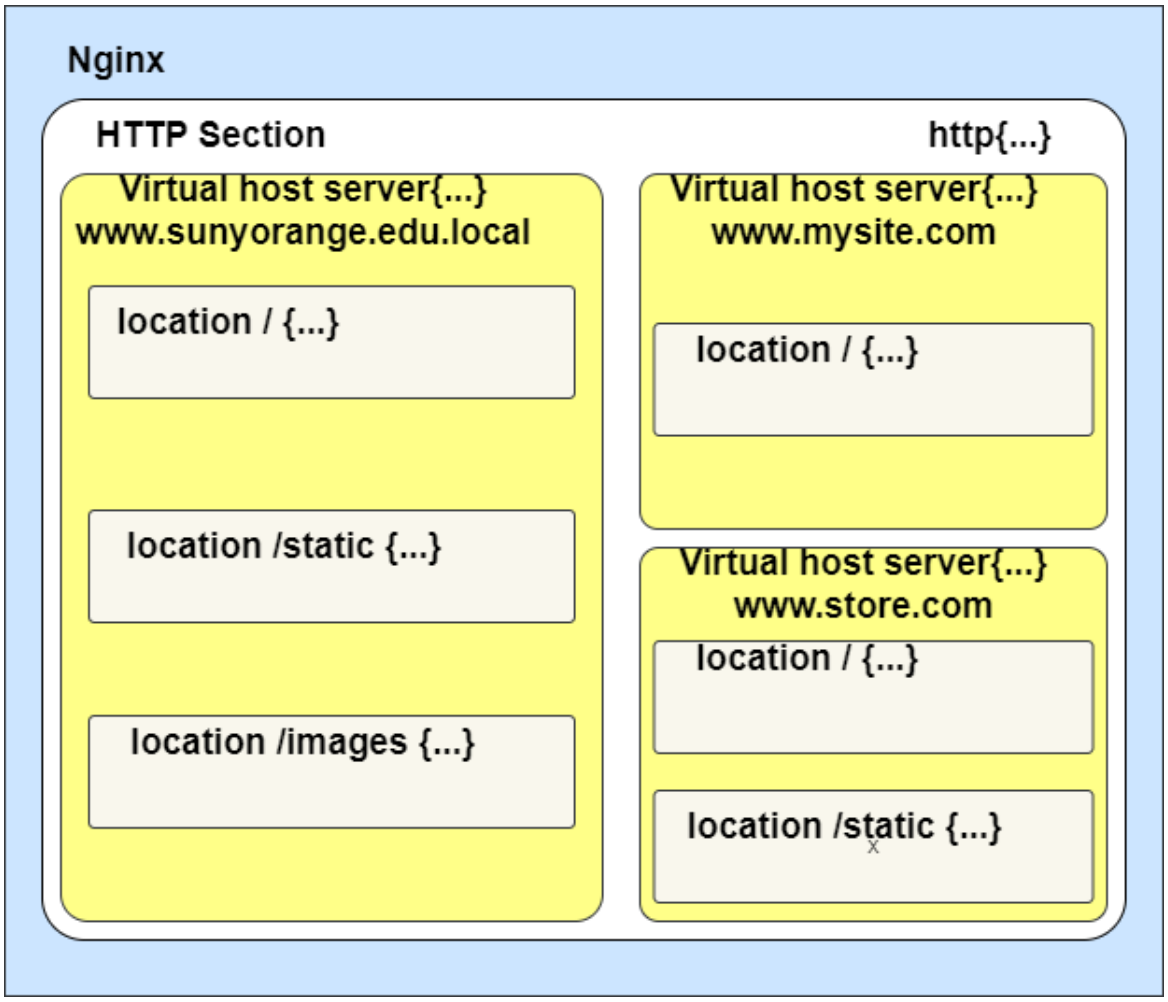
Commands

```
occc@occc-VirtualBox:/etc/apache2/sites-
enabled$ sudo systemctl reload apache2
occc@occc-VirtualBox:/etc/apache2/sites-
enabled$ sudo systemctl status apache2
  apache2.service - The Apache HTTP Server
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor p
Active: active (running) since Sun 2022-10-
16 18:43:38 EDT; 42min ago
Docs: https://httpd.apache.org/docs/2.4/
Process: 818 ExecStart=/usr/sbin/apachectl start (code=exited, status=
Process: 4260 ExecReload=/usr/sbin/apachectl graceful (code=exited, st
Main PID: 832 (apache2)
Tasks: 55 (limit: 9457)
Memory: 7.7M
CPU: 129ms
CGroup: /system.slice/apache2.service
  832 /usr/sbin/apache2 -k start
 4264 /usr/sbin/apache2 -k start
 4265 /usr/sbin/apache2 -k start

Oct 16 18:43:38 occc-VirtualBox systemd[1]: Starting The Apache HTTP S
Oct 16 18:43:38 occc-VirtualBox apachectl[829]: AH00558: apache2: Coul
Oct 16 18:43:38 occc-VirtualBox systemd[1]: Started The Apache HTTP Se
Oct 16 19:26:14 occc-VirtualBox systemd[1]: Reloading The Apache HTTP
Oct 16 19:26:14 occc-VirtualBox apachectl[4263]: AH00558: apache2: Cou
Oct 16 19:26:14 occc-VirtualBox systemd[1]: Reloaded The Apache HTTP S
occc@occc-VirtualBox:/etc/apache2/sites-enabled$
```

Verify that the new webroot and index.html is working.



3.0.2 NGINX server



The HTTP section, defined by the **http{...}** block, encompasses the entire web-related configuration. It may contain one or more **server{...}** blocks, defining the domains and subdomains you are hosting. For each of these websites, you can specify location blocks and other parameters that let you apply additional settings to a particular request URI, or request URIs matching a pattern.

Keep in mind the principle of setting configuration inheritance applies here. If you define a setting or parameter at the **server{...}** block level (for example, gzip on to enable gzip compression), the setting will preserve its value in the potentially incorporated server and location blocks.

Commands

```
root@occc-VirtualBox:/etc/nginx/sites-enabled# cd
root@occc-VirtualBox:~# cd /etc/nginx/sites-enabled/
root@occc-VirtualBox:/etc/nginx/sites-enabled# ls
default
```

```
root@occc-VirtualBox:/etc/nginx/sites-enabled# vi default
```

Change **root /var/www/html;**
to **root /webroot;**

basic nginx conf file

```
##
# You should look at the following URL's in order to grasp a solid understanding
# of Nginx configuration files in order to fully unleash the power of Nginx.
# https://www.nginx.com/resources/wiki/start/
# https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitfalls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-enabled/ and
# leave it as reference inside of sites-available where it will continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by other
# applications, such as Drupal or Wordpress. These applications will be made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed examples.
##

# Default server configuration
#
server {
    listen 80 default_server;
    listen [::]:80 default_server;
```

```
24
25      # SSL configuration
26      #
27      # listen 443 ssl default_server;
28      # listen [::]:443 ssl default_server;
29      #
30      # Note: You should disable gzip for SSL traffic↙
          ↳  .
31      # See: https://bugs.debian.org/773332
32      #
33      # Read up on ssl_ciphers to ensure a secure ↙
          ↳  configuration.
34      # See: https://bugs.debian.org/765782
35      #
36      # Self signed certs generated by the ssl-cert ↙
          ↳  package
37      # Don't use them in a production server!
38      #
39      # include snippets/snakeoil.conf;
40
41      #root /var/www/html;
42      root /webroot;
43      # Add index.php to the list if you are using ↙
          ↳  PHP
44      index index.html index.htm index.nginx-debian.↙
          ↳  html;
45
46      server_name _;
47
48      location / {
49          # First attempt to serve request as file, ↙
              ↳  then
50          # as directory, then fall back to ↙
              ↳  displaying a 404.
51          try_files $uri $uri/ =404;
52      }
53
54      # pass PHP scripts to FastCGI server
```

```
55      #
56      #location ~ \.php$ {
57          #   include snippets/fastcgi-php.conf;
58          #
59          #   # With php-fpm (or other unix sockets):
60          #   fastcgi_pass unix:/run/php/php7.4-fpm.sock;
61          #   # With php-cgi (or other tcp sockets):
62          #   fastcgi_pass 127.0.0.1:9000;
63          #}
64
65      # deny access to .htaccess files, if Apache's document root
66      # concurs with nginx's one
67      #
68      #location ~ /\.ht {
69          #   deny all;
70          #}
71  }
72
73
74  # Virtual Host configuration for example.com
75  #
76  # You can move that to a different file under sites-available/ and symlink that
77  # to sites-enabled/ to enable it.
78  #
79  #server {
80      #   listen 80;
81      #   listen [::]:80;
82      #
83      #   server_name example.com;
84      #
85      #   root /var/www/example.com;
86      #   index index.html;
87      #
88      #   location / {
89          #       try_files $uri $uri/ =404;
```

```
90              #     }
91          #}
```

Server block
https://www.nginx.com/resources/wiki/start/topics/examples/server_blocks/

listen 80 default_server;
listen [::]:80 default_server;
listen tells the server which port to listen in on. The default_server tells the nginx that if it cannot match any known server names to use this configuration for all unmatched requests. [::] 80 tells the server to listen in on IPv6 address.
root , this is the directory location where the server will find all of the data to serve back to the client(web browser)
index if no specific file is requested, then this will be the name of the html file that will be sent back to the client by default.
server_name _; defines an invalid server name that never intersects with any real name. It is just a non-match. So in the event of no matches, nginx will select the first server block and use that.
To conclude, you can use **server_name _;** for catch-all server block
location how will we handle the URL path? Different directives can serve different paths.
ex: www.sunyorange.edu will be served by the / directive.
www.sunyorange.edu/compsci can be served by an additional directive such as:

```
location /compsci/ {
     [ configuration for  compsci ]
}
```

http://nginx.org/en/docs/http/ngx_http_core_module.html#location

The **$uri** variable is set to the URI that nginx is **currently processing** , but it is also subject to normalization, including:

- Removal of the ? and query string
- Consecutive / characters are replaced by a single /

- URL-encoded characters are decoded

The value of **$request_uri** is always the original URI and is not subject to any of the above normalization's.

Most of the time, you would use $uri, because it is normalized. Using $request_uri in the wrong place can cause URL encoded characters to become doubly encoded.

Add the index.html to /webroot

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CIT 215</title>
</head>
<body>

<h1>CIT 215</h1>
<p>SUNY Orange</p>

</body>
</html>
```

Check that the config file is appropriately formatted..

Commands

```
root@occc-VirtualBox:/etc/nginx/sites-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@occc-VirtualBox:/etc/nginx/sites-enabled#
```

reload the config file.

To reload your configuration, you can stop or restart NGINX, or send signals to the primary process. A signal can be sent by running the **nginx** command

(invoking the NGINX executable) with the **-s** argument.

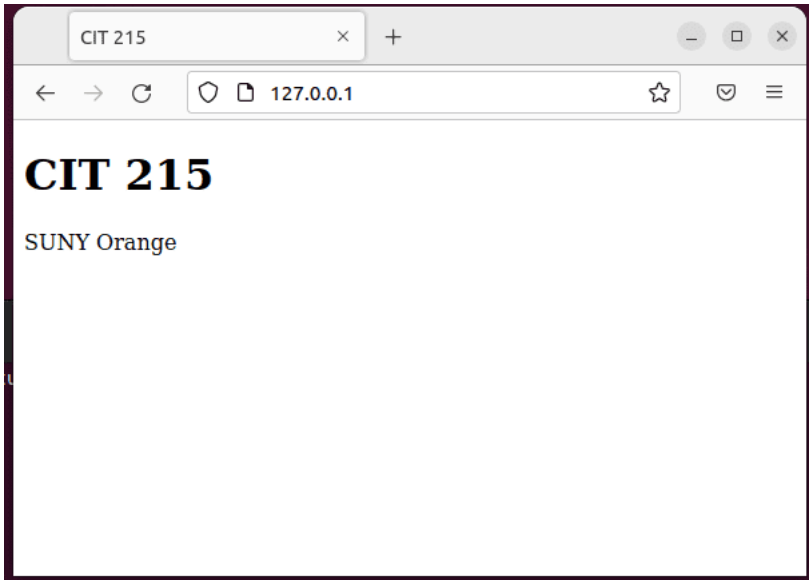
```
nginx -s <SIGNAL>
```

where **<SIGNAL>** can be one of the following:

- **quit** – Shut down gracefully (the **SIGQUIT** signal)
- **reload** – Reload the configuration file (the **SIGHUP** signal)
- **reopen** – Reopen log files (the **SIGUSR1** signal)
- **stop** – Shut down immediately (or fast shutdown, the **SIGTERM** signal)

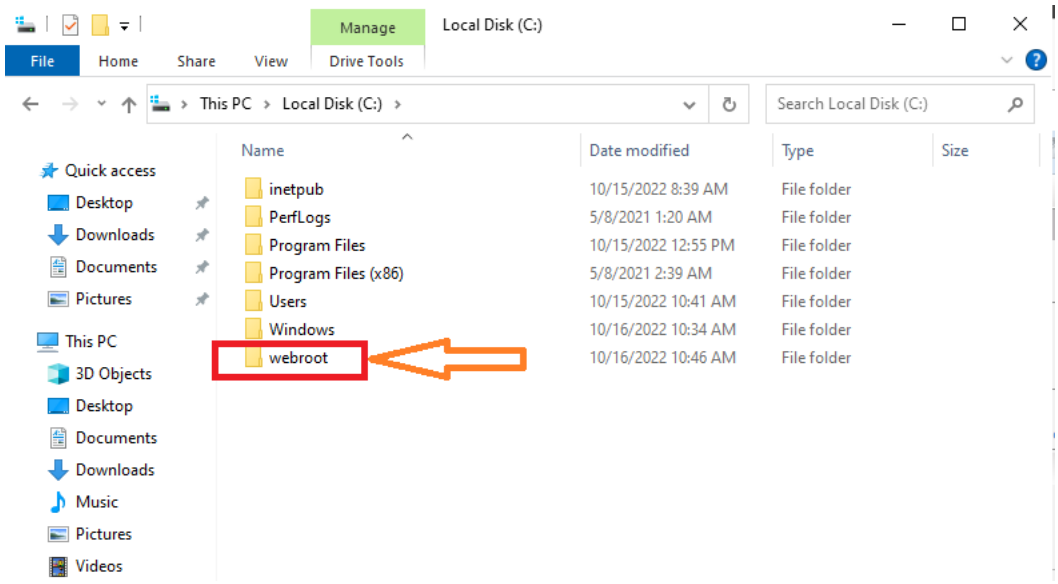
Commands

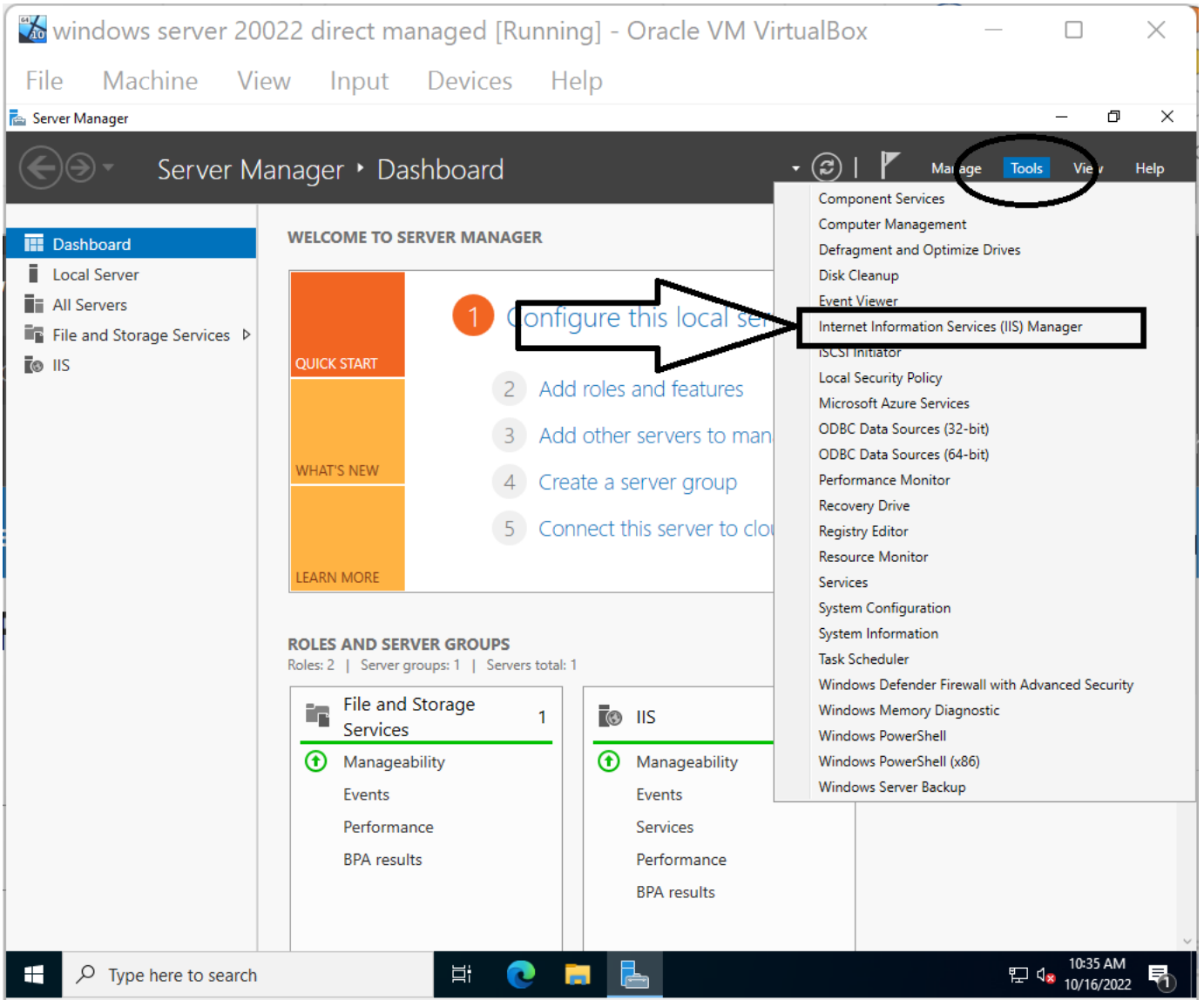
```
root@occc-VirtualBox:/etc/nginx/sites-enabled# nginx -s reload
root@occc-VirtualBox:/etc/nginx/sites-enabled#
```

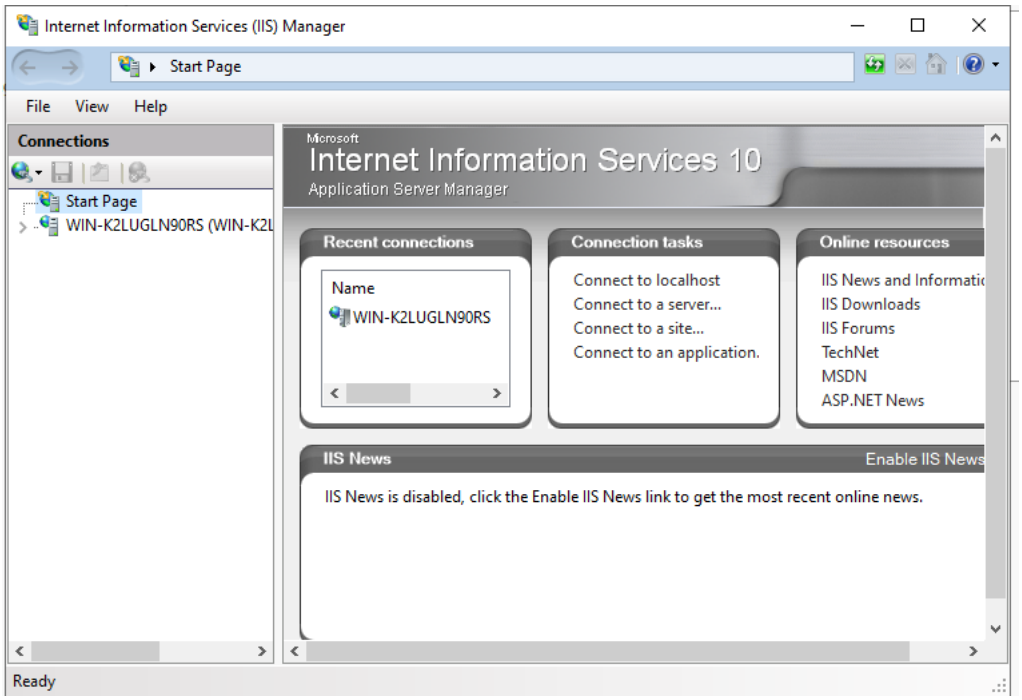


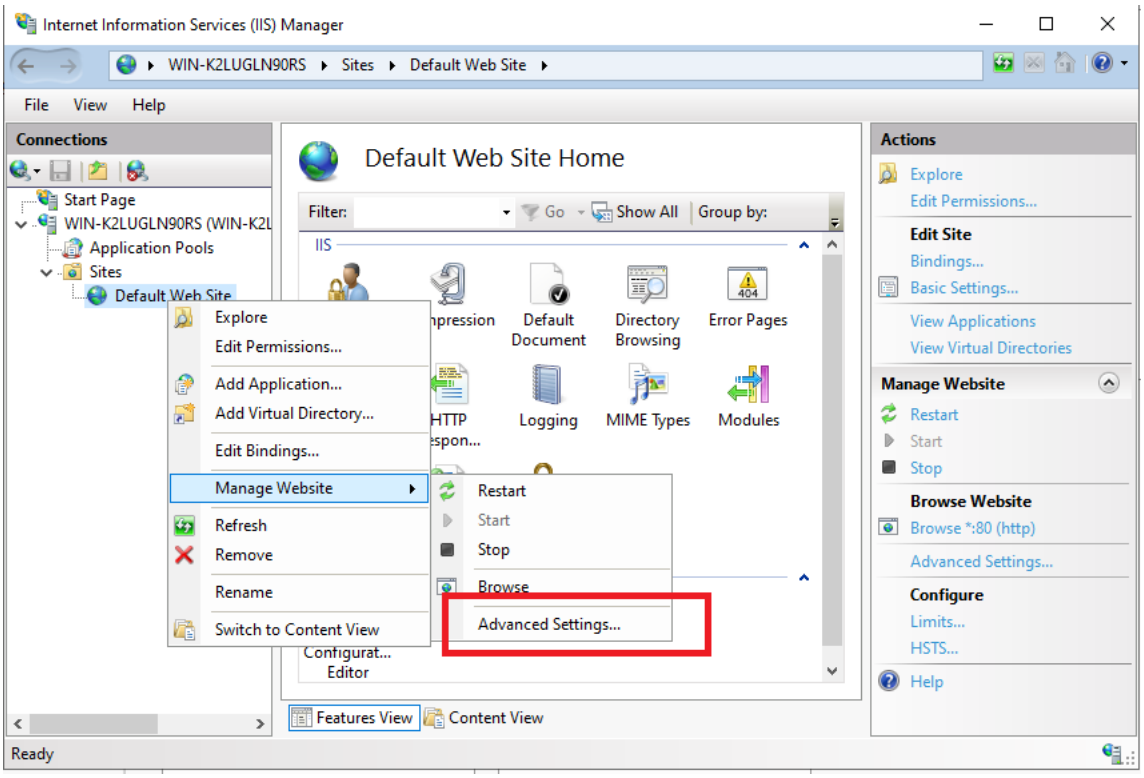
3.0.3 Windows IIS

Create a directory on your system where the main web page will reside.

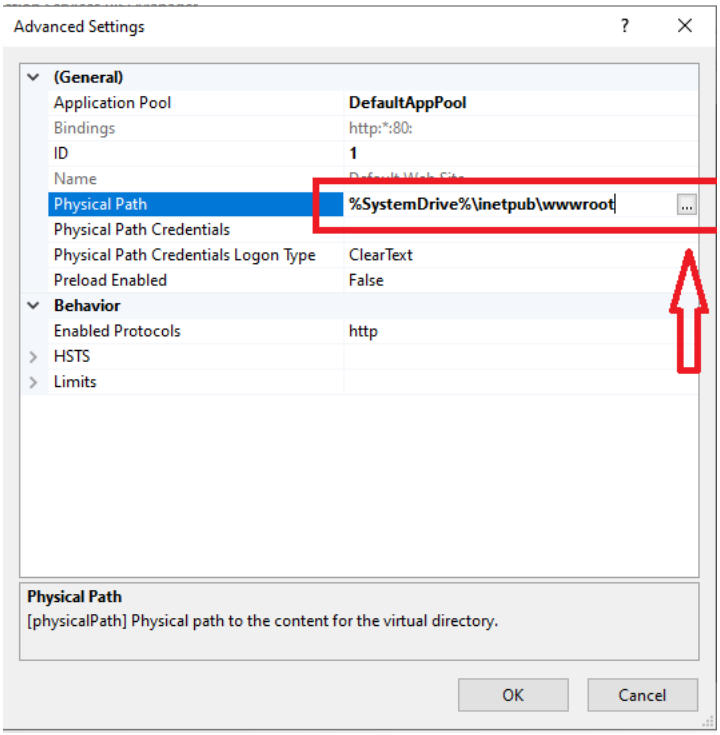

Manage
Local Disk (C:)
File
Home
Share
View
Drive Tools
This PC › Local Disk (C:) ›
Search Local Disk (C:)
Quick access
Desktop
Downloads
Documents
Pictures
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Name
Date modified
Type
Size
inetpub 10/15/2022 8:39 AM File folder
PerfLogs 5/8/2021 1:20 AM File folder
Program Files 10/15/2022 12:55 PM File folder
Program Files (x86) 5/8/2021 2:39 AM File folder
Users 10/15/2022 10:41 AM File folder
Windows 10/16/2022 10:34 AM File folder
webroot 10/16/2022 10:46 AM File folder


windows server 20022 direct managed [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Server Manager
Server Manager • Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
File and Storage Services
IIS
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local se
2 Add roles and features
3 Add other servers to man
4 Create a server group
5 Connect this server to clo
ROLES AND SERVER GROUPS
Roles: 2 | Server groups: 1 | Servers total: 1
File and Storage Services 1
Manageability
Events
Performance
BPA results
IIS
Manageability
Events
Services
Performance
BPA results
Component Services
Computer Management
Defragment and Optimize Drives
Disk Cleanup
Event Viewer
Internet Information Services (IIS) Manager
iSCSI Initiator
Local Security Policy
Microsoft Azure Services
ODBC Data Sources (32-bit)
ODBC Data Sources (64-bit)
Performance Monitor
Recovery Drive
Registry Editor
Resource Monitor
Services
System Configuration
System Information
Task Scheduler
Windows Defender Firewall with Advanced Security
Windows Memory Diagnostic
Windows PowerShell
Windows PowerShell (x86)
Windows Server Backup
Type here to search
10:35 AM
10/16/2022

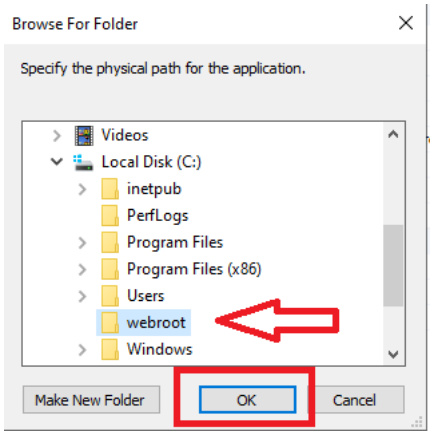


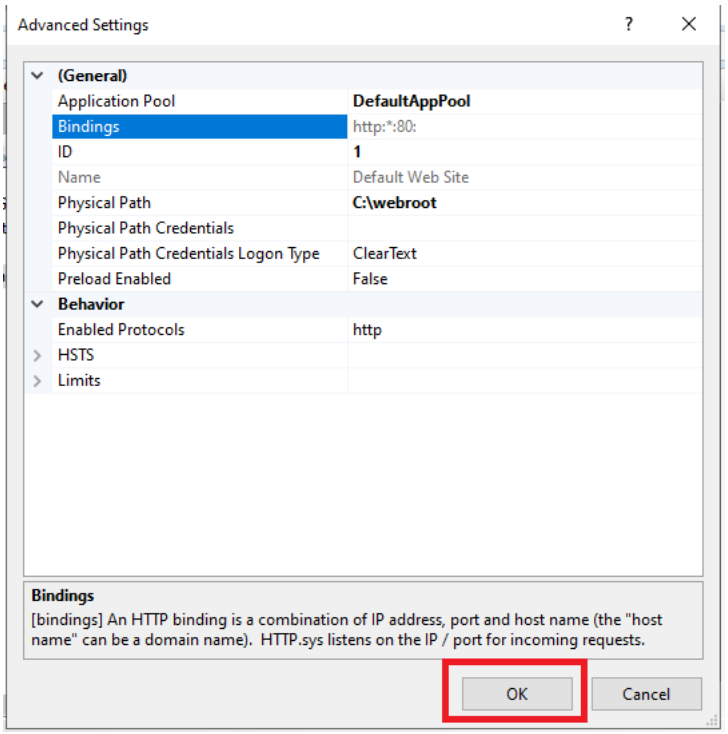


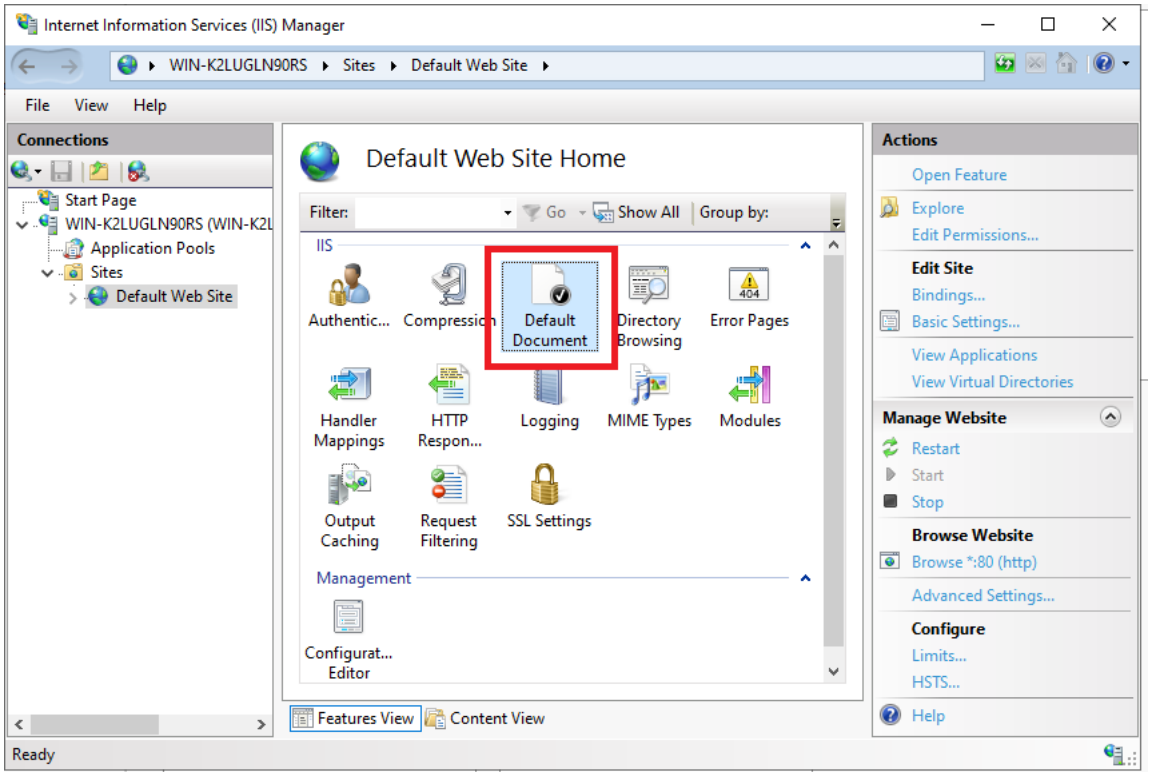
In Advanced Settings, click on Physical Path to choose a different document root.

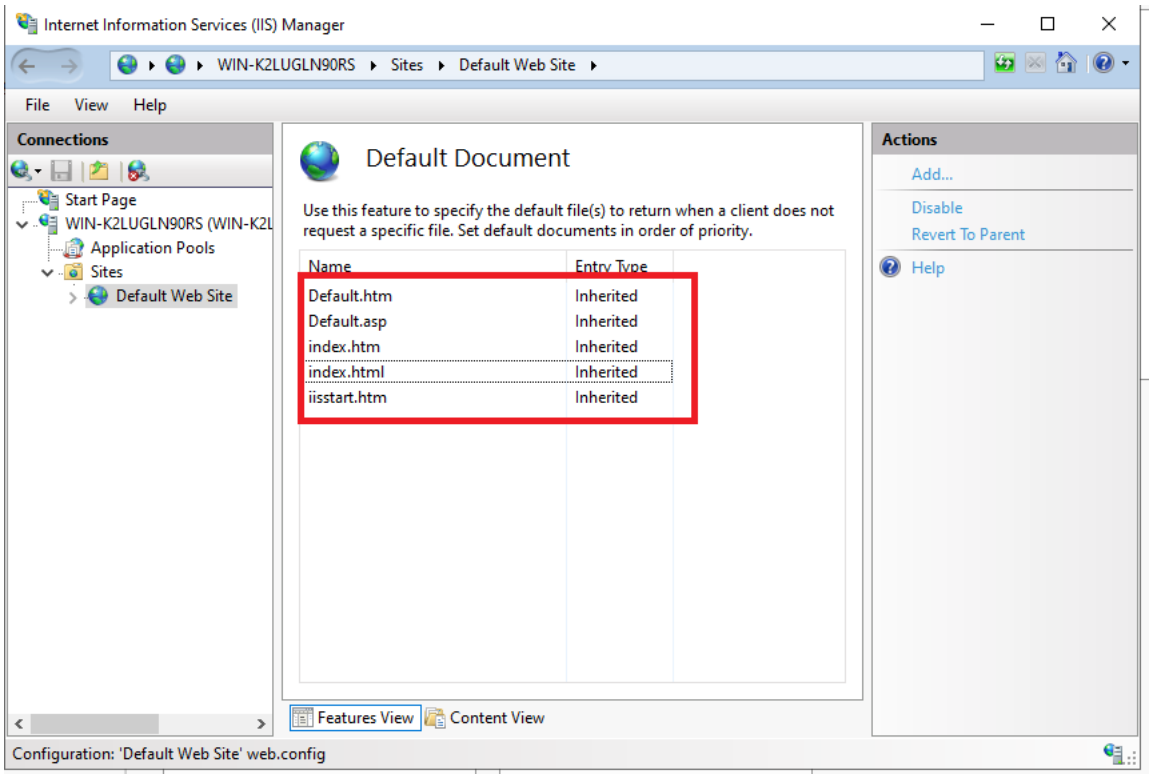


Select the document root folder. (C:\webroot)




Advanced Settings
(General)
Application Pool
DefaultAppPool
Bindings
http:*:80:
ID
1
Name
Default Web Site
Physical Path
C:\webroot
Physical Path Credentials
Physical Path Credentials Logon Type
ClearText
Preload Enabled
False
Behavior
Enabled Protocols
http
HSTS
Limits
Bindings
[bindings] An HTTP binding is a combination of IP address, port and host name (the "host name" can be a domain name). HTTP.sys listens on the IP / port for incoming requests.
OK
Cancel


Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS ▸ Sites ▸ Default Web Site ▸
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-K2L
Application Pools
Sites
Default Web Site
Default Web Site Home
Filter:
Go
Show All
Group by:
IIS
Authentic...
Compressi...
Default Document
Directory Browsing
Error Pages
Handler Mappings
HTTP Respon...
Logging
MIME Types
Modules
Output Caching
Request Filtering
SSL Settings
Management
Configurat... Editor
Features View
Content View
Actions
Open Feature
Explore
Edit Permissions...
Edit Site
Bindings...
Basic Settings...
View Applications
View Virtual Directories
Manage Website
Restart
Start
Stop
Browse Website
Browse *:80 (http)
Advanced Settings...
Configure
Limits...
HSTS...
Help
Ready

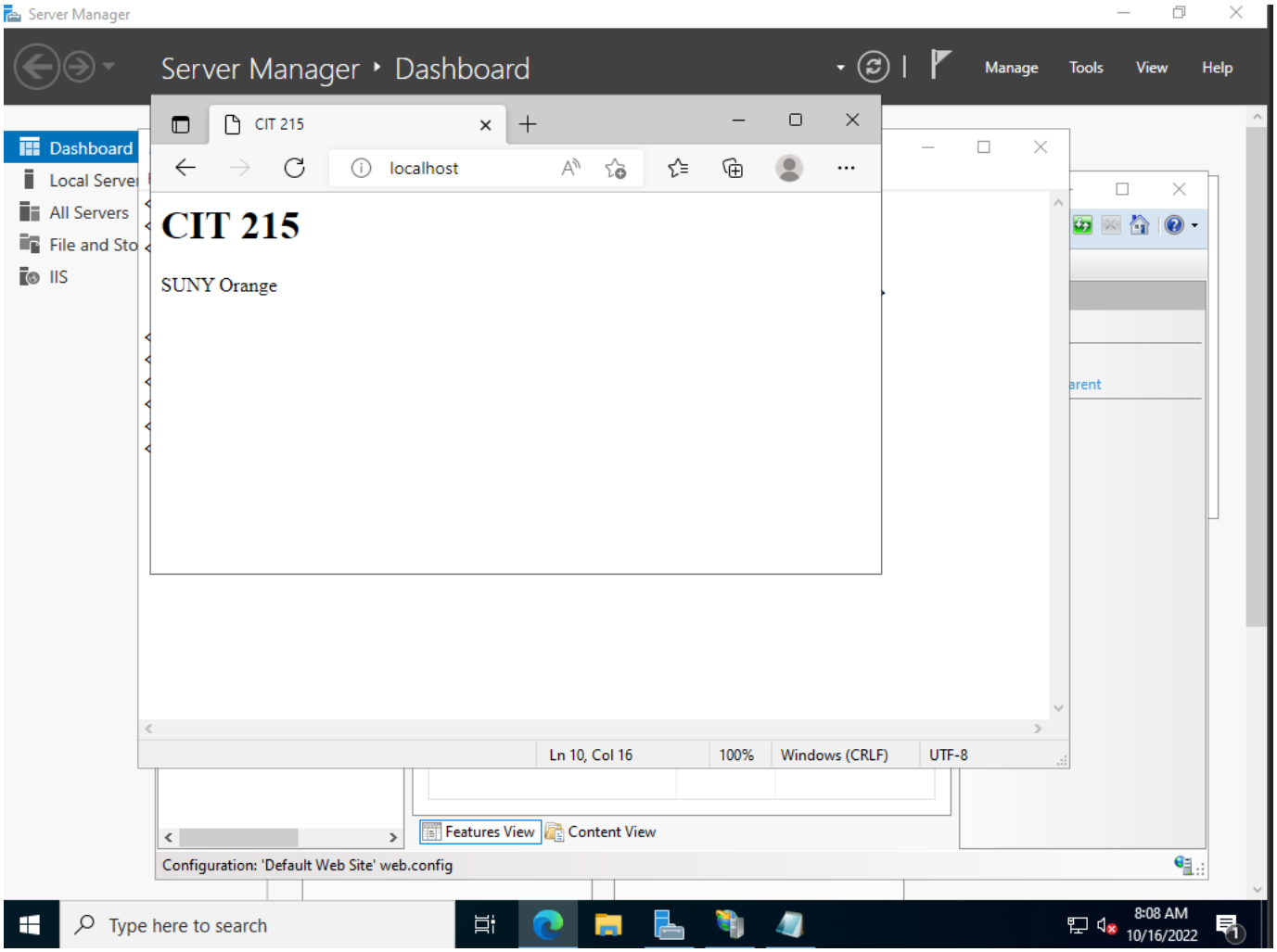


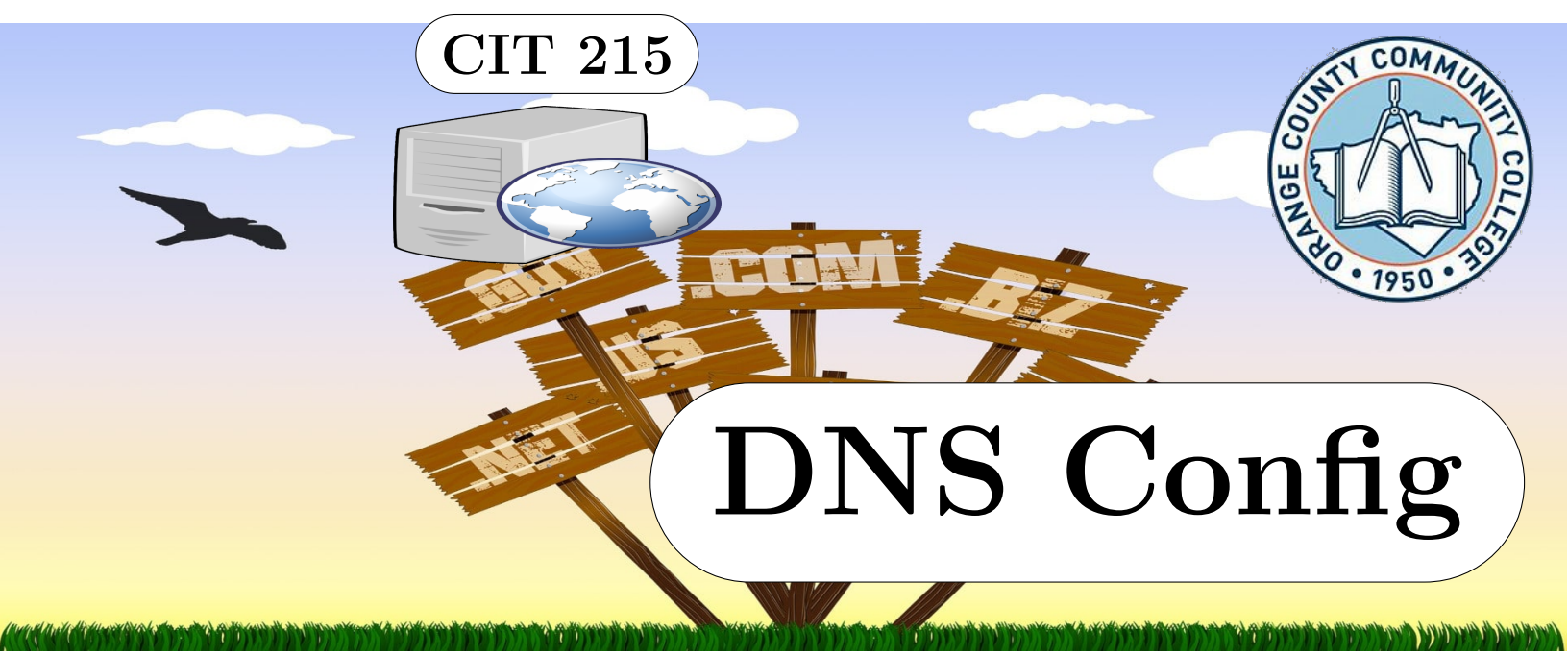
Create an index.html file in the document root and Verify access with a Web browser.

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width
    ↪  , initial-scale=1.0">
  <title>CIT 215</title>
</head>
<body>

  <h1>CIT 215</h1>
  <p>SUNY Orange</p>

</body>
</html>
```

Verify that this is our new default web page by using a web browser.


Server Manager
Server Manager ▸ Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
File and Sto
IIS
CIT 215
localhost
CIT 215
SUNY Orange
Ln 10, Col 16
100%
Windows (CRLF)
UTF-8
Features View
Content View
Configuration: 'Default Web Site' web.config
Type here to search
8:08 AM
10/16/2022



Most of the time, we will not be configuring a DNS server from scratch and doing an install. DNS servers are usually centrally administered by IT company-wide. However, we must have our own DNS server for demonstration purposes for some of the items we will cover in class.

4.0.1 What is DNS

DNS Basics

All computers on the Internet, from your smartphone or laptop to the servers that serve content for retail websites, find and communicate with one another using numbers. These numbers are known as IP addresses.

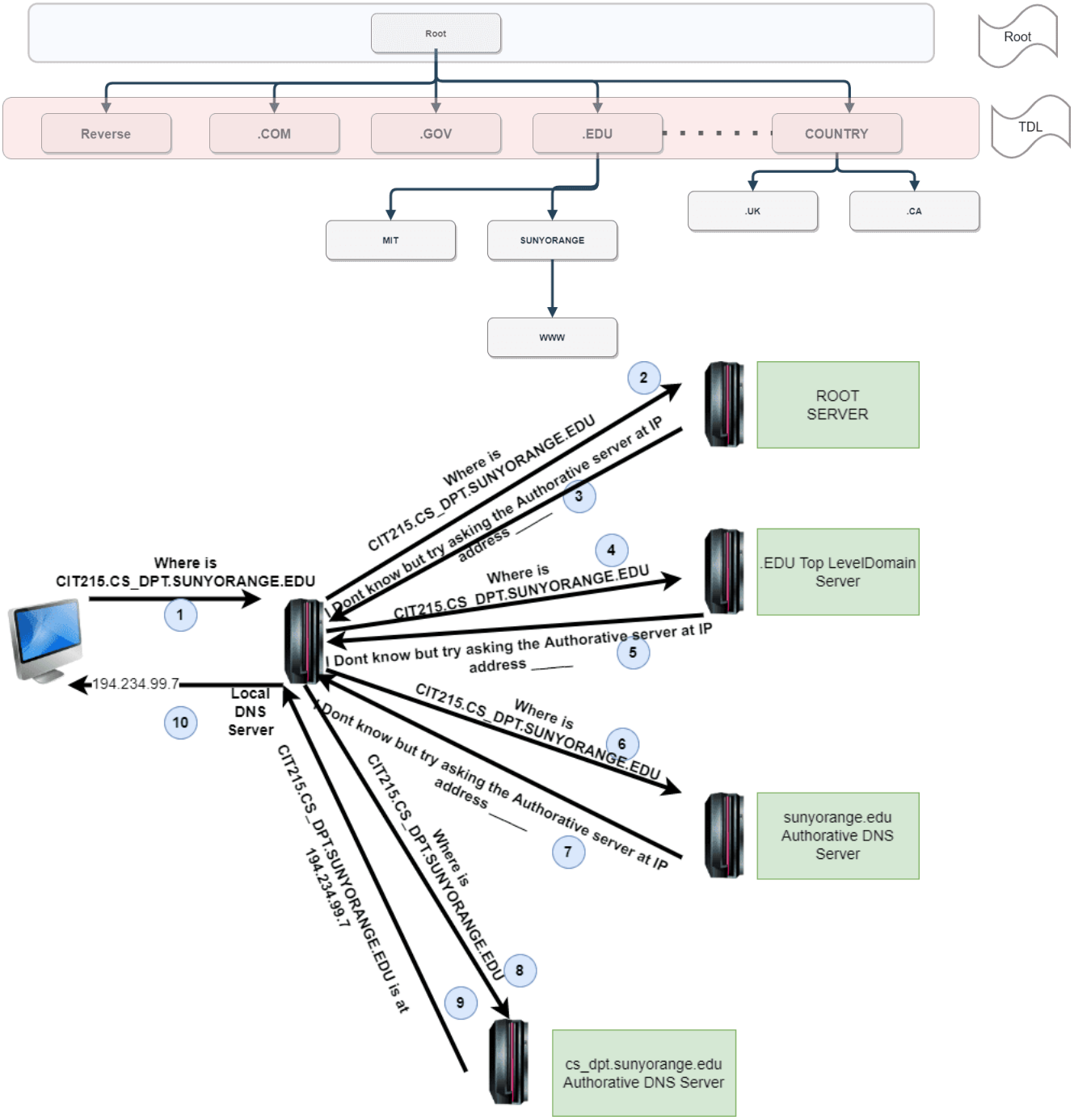
Commands

```
occc@occc-VirtualBox:~$ nslookup www.sunyorange.edu
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: www.sunyorange.edu
Address: 199.250.206.35
```

From the above, we can see that the IP address of www.sunyorange.edu is 199.250.206.35.

When you open a web browser and go to a website, you don't have to remember

and enter a long number. Instead, you can enter a domain name like www.suny-orange.edu and still end up in the right place.



List of DNS Records Types

- **A record:** The most basic type of record, also known as an address record. It provides an IPv4 address to a domain or sub-domain name. It points the domain name to an IP address.

- **AAAA record:** It maps the hostname to a 128-bits IPv6 address. For a

long time, 32-bits IPv4 addresses served the purpose of identifying a computer on the internet. But due to the shortage of IPv4, it created IPv6. The four AAAA are mnemonic to represent that IPv6 is four times larger than IPv4 in size.

- **CAA record:** Certification Authority Authorization record reflects the public policy regarding the domain's issuance of digital certificates. If no CAA record is present for your domain, any Certification Authority can issue an SSL certificate for your domain. However, by using this record, you can restrict which CA is authorized to issue your domain's digital credentials.

- **CNAME record:** Canonical Name record creates an alias of one domain name. The aliased domain or sub-domain gets all the original Domain DNS records and is commonly used to associate sub-domains with the existing main domain.

- **DS record:** Delegation Signer record consists of the unique characters of your public key and its related metadata such as Key Tag, Algorithm, Digest Type, and cryptographic hash value called Digest.

- **DNSKEY record:** It is also known as DNS Key record,containing public signing keys such as Key Signing Key (KSK) and Zone Signing Key (ZSK). The DS and DNSKEY records serve to validate the authenticity of DNS records returned by the DNS Server.

- **MX record:** Mail Exchange records tell which mail exchange servers are responsible for routing the email to the correct destination or mail server.

- **NS record:** Name Server record points to the name servers to manage and publish the domain's DNS records. These DNS servers are authoritative in handling any query related to the domain.

- **PTR record:** The Pointer record points the IPv4 or IPv6 address to its machine's hostname. It provides a reverse DNS record or rDNS record by pointing an IP address to the server's hostname.

- **SRV record:** Service record indicates which specific services the domain operates and port numbers. Some Internet protocols, such as the Presence Protocol (XMPP), Extensible Messaging, and Session Initiation Protocol (SIP), often require SRV records.

- **SOA record:** The Start of Authority record provides essential information about the domain, such as identifying the primary node of the domain authoritative name server, the email of the domain administrator, the serial number of the DNS zone, etc.

- **TXT record**: It allows the website's administrator to insert any arbitrary text in the DNS record.

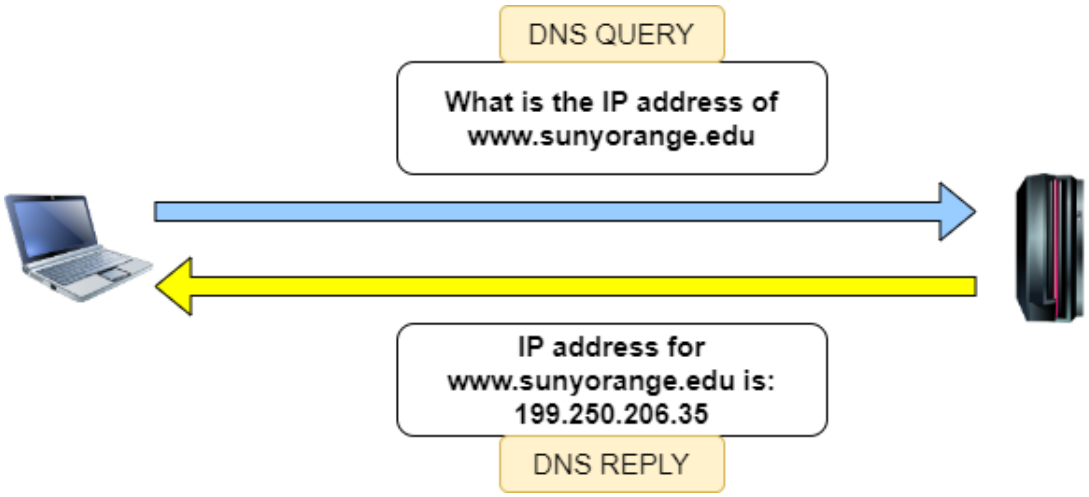
- **SSHFP record:** It is also known as SSH Public Key Fingerprint. It has a resource record for publishing SSH public host key fingerprints in the DNS System to verify the host's authenticity.

- **URI record:** It is also known as Uniform Resource Identifier. It can be used for publishing mappings from hostnames to URIs.

Ways to do DNS Lookups

Forward DNS Lookup.

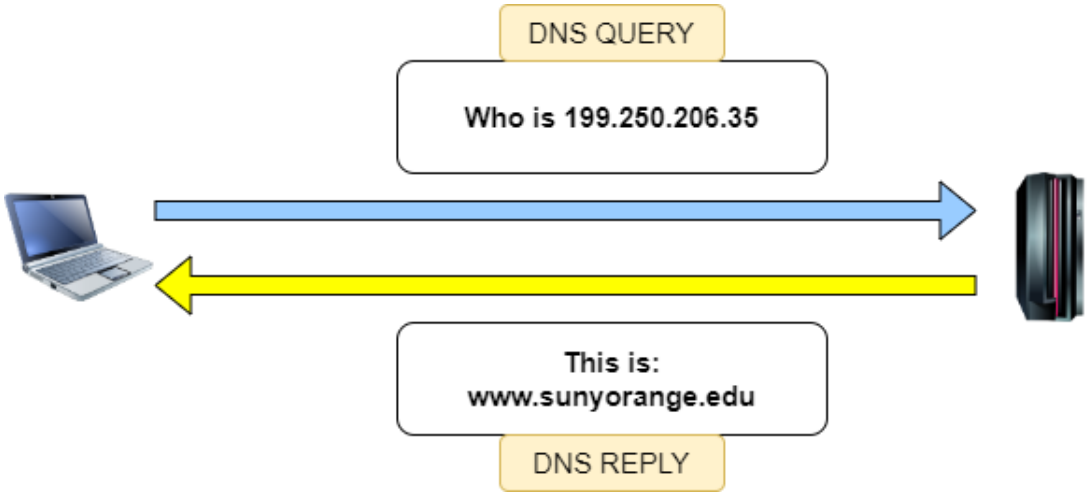
The forward lookup or simple DNS lookup is the most common approach to DNS. It is used to find out the IP address of a domain.



Reverse DNS Lookup

A Reverse DNS Lookup is just an opposite sequence of a DNS lookup. Moreover, with a normal DNS lookup, you query the DNS or hostname to get the IP address. In a Reverse DNS Lookup, you query the IP address to find the hostname. Therefore, by entering the IP address into the Reverse DNS Lookup

Tool, you can find the domain name associated with the corresponding IP.



4.0.2 Host files

You may want to preview a website with a custom domain name before the domain is publicly mapped to a live website. For example, a domain name will often be routed to a live site while development is ongoing on a separate server.

Before the website is made public, you can view a development or pre-launch website with a custom domain name by modifying the **/etc/hosts** file on your local machine to point the custom domain name to the IP address of the environment you want to test. The **/etc/hosts** file contains a mapping of IP addresses to URLs. Your browser uses entries in the **/etc/hosts** file to override the IP-address-to-URL mapping returned by a DNS server. This is useful for testing DNS (domain name system) changes and the SSL configuration before making a website live.

The /etc/hosts file affects only the local computer.

Structure of hosts files.

```
#IPAddress      Hostname          Alias
127.0.0.1       localhost         cit215.sunyorange.edu
104.127.160.68  www.mit.edu       mit-web
```

Editing your /etc/hosts file on a Mac

If you're using a Mac with OS X, to edit your /etc/hosts file, open a Terminal window and run the following command:

sudo nano /private/etc/hosts
You can now add entries to the file.
You may be asked for your password to edit the file. Enter your password.

Editing your /etc/hosts file on a Linux

sudo vim /etc/hosts
You can now add entries to the file.

Editing your /etc/hosts file using Windows

```
if you're using Windows, to edit your \etc\hosts file, open
 [SystemRoot]\system32\drivers\etc\hosts and edit the file.
 (The \etc\hosts file usually exists at
 %windir%\system32\drivers\etc\hosts.) If the directory and
 file don't exist, you can create them. Some versions of
 Windows require that users have admin privileges to
 create or make changes to this file.
```

You can now add entries to the file.

Windows 8 and up file protection

```
Windows 8 users may have trouble editing their \etc\hosts
file because Windows 8 includes file overwrite protections.
For information about how to modify your \etc\hosts file
in Windows 8 and up, google how to bypass this protection.
```

4.0.3 Ubuntu 22.04 Bind 9 install

Install Bind packages.

Commands

```
sudo apt install -y bind9 bind9utils bind9-doc dnsutils
```

Bind DNS Server Configuration

Commands

```
occc@occc-VirtualBox:/etc/bind$ cd /etc/bind
occc@occc-VirtualBox:/etc/bind$ ls -l
total 56
-rw-r--r-- 1 root root 2403 May 17 07:38 bind.keys
-rw-r--r-- 1 root root  237 Aug 25  2020 db.0
-rw-r--r-- 1 root bind  548 Jul 15 20:29 db.10.88.0.0.zone
-rw-r--r-- 1 root root  271 Aug 25  2020 db.127
-rw-r--r-- 1 root root  237 Aug 25  2020 db.255
-rw-r--r-- 1 root root  353 Aug 25  2020 db.empty
-rw-r--r-- 1 root root  270 Aug 25  2020 db.local
-rw-r--r-- 1 root bind  377 Jul 15 16:32 db.ovirt-mk.com
-rw-r--r-- 1 root bind  463 Aug 25  2020 named.conf
-rw-r--r-- 1 root bind  498 Jun 25  2021 named.conf.default-
zones
-rw-r--r-- 1 root bind  433 Jul 15 16:43 named.conf.local
-rw-r--r-- 1 root bind 1256 Jul 15 20:22 named.conf.options
-rw-r----- 1 bind bind  100 Jul 15 01:47 rndc.key
-rw-r--r-- 1 root root 1317 Aug 25  2020 zones.rfc1918
occc@occc-VirtualBox:/etc/bind$
```

The DNS main configuration directory is **/etc/bind**. It contains the zone-lookup files and other configuration files.

The global DNS conf file is located at **/etc/bind/named.conf**. This is, however, not used for local DNS configuration. **/etc/bind/named.conf.local** is used instead.
If we examine the named.conf file, we see that it includes the named.conf.local file inside it.

Commands

```
occc@occc-VirtualBox:/etc/bind$ more named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
occc@occc-VirtualBox:/etc/bind$
```

Create zones

We will do so in the /etc/bind/named.conf.local file. Use a text editor of your choice to edit the file.
We shall create the **forward** and **reverse** zones in the file. Below is a forward zone entry for cit215.sunyorange.edu.local domain. Change it to your other domain name in your configuration if you want to. Sometimes a zone that will only serve a local LAN or inside an organization will only have a .local TDL.

Commands

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "sunyorange.edu.local" IN { // Domain name

type master; // Primary DNS

file "/etc/bind/forward.sunyorange.edu.local.db"; // Forward lookup file

allow-update { none; }; // Since this is the primary DNS, it should be none.

};

zone "0.88.10.in-addr.arpa" IN {
//Reverse lookup name, should match your network in reverse order

type master; // Primary DNS

file "/etc/bind/reverse.sunyorange.edu.local.db"; //Reverse lookup file

allow-update { none; }; //Since this is the primary DNS, it should be none.

};
```

- **sunyorange.edu.local** is the zone name.
- **forward.sunyorange.edu.local.db** is the name of the forward lookup zone.
- **0.88.10.in-addr.arpa** is the zone name of reverse DNS. (If the network is 10.88.0.0, the name will be reversed as in 0.0.88.10)
- **reverse.sunyorange.edu.local.db** is the reverse DNS file.

Configure Bind DNS zone lookup files

The zone lookup files hold the DNS records for the forward and reverse zones. Copy the sample forward zone lookup file to a file called forward.sunyorange.edu.db under the /etc/bind directory:

Commands

```
sudo cp /etc/bind/db.local /etc/bind/forward.sunyorange.edu.local.db
```

Take note of the zone file syntax; domain names should end with a dot (.)

The acronyms on the file have the following description:

- **SOA** – Start of Authority
- **NS** – Name Server
- **A** – A record
- **MX** – Mail for Exchange
- **CN** – – Canonical Name

We have to edit the zone file and update the content as below.

Commands

```
sudo vi /etc/bind/forward.sunyorange.edu.db
```

When editing the conf file make sure you add the correct ip addr for the ns servers.

https://arstechnica.com/gadgets/2020/08/understanding-dns-anatomy-of-a-bind-zone-file/

Commands

```
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.sunyorange.edu.local. root.ns1.sunyorange.edu.local.  (
2         ; Serial
604800         ; Refresh
86400         ; Retry
2419200         ; Expire
604800 )       ; Negative Cache TTL
;
;@      IN      NS      localhost.
;@      IN      A       127.0.0.1
;@      IN      AAAA    ::1
;Name Server Information
@        IN      NS      ns1.sunyorange.edu.local.
;IP address of Name Server

ns1     IN      A       10.88.0.2
;Mail Exchanger

sunyorange.edu.local.   IN     MX   10   mail.sunyorange.edu.local.

;A - Record HostName To Ip Address

www     IN       A      10.88.0.2
mail    IN       A      10.88.0.2
;CNAME record

sftp     IN      CNAME   www.sunyorange.edu.local.
```

The data on a **CNAME** record must always be another DNS name - which is the whole point of a **CNAME**.

As put succinctly by RFC 1034, the data in a CNAME should be:

```
CNAME           a domain name.
```

Several things will have to change for this to work. First, the IP addresses must be changed to match your environment.

The acronyms in the reverse zone file are:

- **PTR** – Pointer
- **SOA** – Start of Authority

Commands

```
 occc@occc-VirtualBox:/etc/bind$ sudo cp db.127 reverse.sunyorange.edu.local.db
[sudo] password for occc:
occc@occc-VirtualBox:/etc/bind$
```

Commands

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     sunyorange.edu.local. root.sunyorange.edu.local. (
1         ; Serial
604800          ; Refresh
86400         ; Retry
2419200         ; Expire
604800 )       ; Negative Cache TTL
;

;Name Server Information

@       IN      NS     ns1.sunyorange.edu.local.
ns1     IN      A       10.88.0.2
;Reverse lookup for Name Server

2      IN      PTR    ns1.sunyorange.edu.local.

;PTR Record IP address to HostName

2     IN      PTR    www.sunyorange.edu.local.
2     IN      PTR    mail.sunyorange.edu.local.
```

Check BIND DNS syntax

Commands

```
 sudo named-checkconf
 sudo named-checkzone sunyorange.edu.local /etc/bind/forward.sunyorange.edu.local.db
zone sunyorange.edu.local/IN: loaded serial 2
OK
 sudo named-checkzone 0.88.10.in-addr.arpa /etc/bind/reverse.sunyorange.edu.local.db
zone 0.88.10.in-addr.arpa/IN: loaded serial 1
OK
occc@occc-VirtualBox:/etc/bind$
```

Finally restart and enable BIND service:

```
sudo systemctl restart bind9
sudo systemctl enable bind9
```

Updating Bind DNS Records

A DNS record should be updated in both the /etc/bind/forward.sunyor-

ange.edu.local.db and /etc/bind/reverse.sunyorange.edu.local.db files.

On updating the DNS record, change the serial number of the forward and reverse zone files to a number greater than the current.

Testing the DNS Server

Change any client machine's DNS server to our newly deployed server. In our case, it is 10.88.0.2.

DNS server setting varies with the operating system. In Ubuntu:

Commands

```
sudo vim /etc/resolv.conf
nameserver 10.88.0.2
```

We can also change it to local host or add it as a secondary or tertiary nameserver for internal testing.

Commands

```
root@occc-VirtualBox:/etc/bind# dig www.sunyorange.edu.local

; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> www.sunyorange.edu.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39279
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 7a2961d20a8395000101000000635597b99592cfe9e18ca5dd (good)
;; QUESTION SECTION:
;www.sunyorange.edu.local. IN A

;; ANSWER SECTION:
www.sunyorange.edu.local. 604800 IN A 10.88.0.2

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Sun Oct 23 15:36:25 EDT 2022
;; MSG SIZE  rcvd: 97

root@occc-VirtualBox:/etc/bind#
```

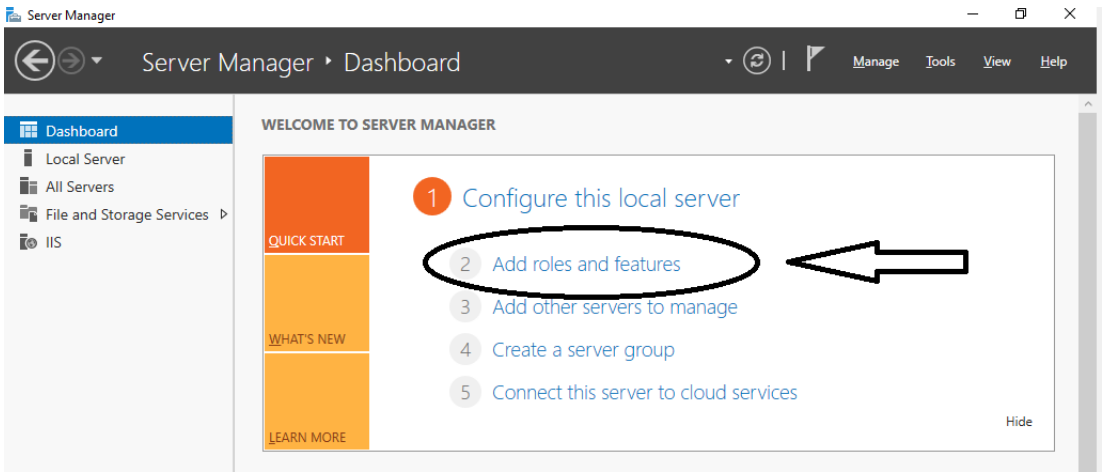
Commands

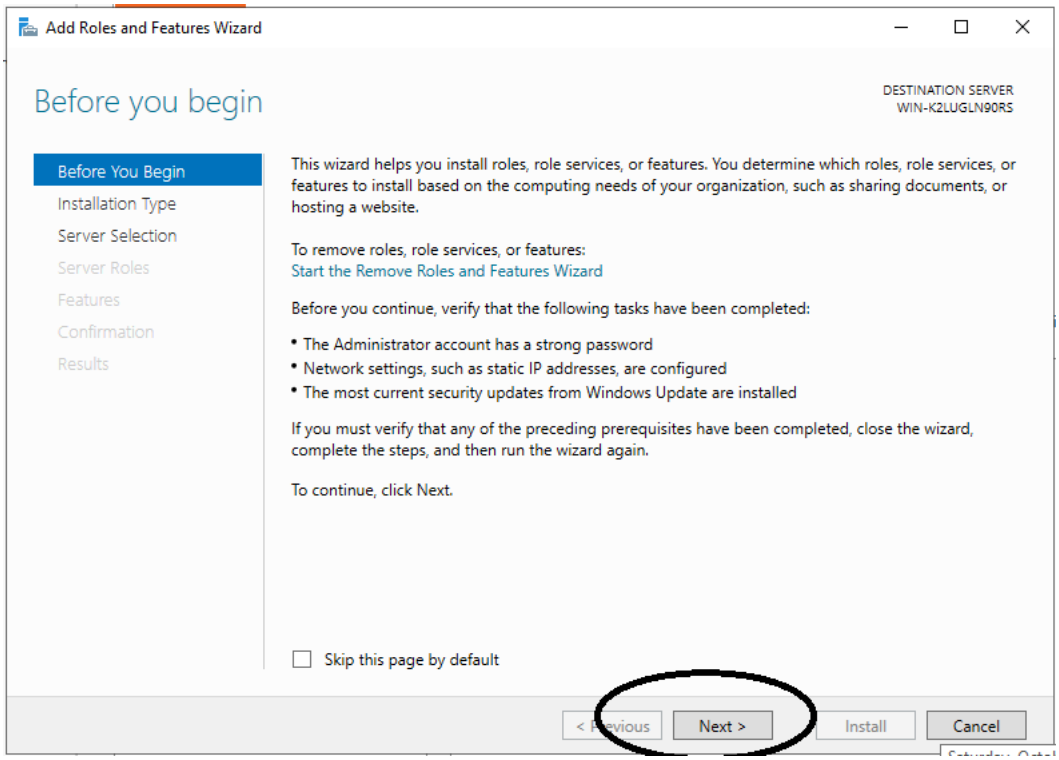
```
; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>> -x 10.88.0.2
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63556
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 40d346ade9a9657101000000 6355981ab4a78229dfc59828 (good)
;; QUESTION SECTION:
;2.0.88.10.in-addr.arpa. IN PTR

;; ANSWER SECTION:
2.0.88.10.in-addr.arpa. 604800 IN PTR mail.sunyorange.edu.local.
2.0.88.10.in-addr.arpa. 604800 IN PTR ns1.sunyorange.edu.local.
2.0.88.10.in-addr.arpa. 604800 IN PTR www.sunyorange.edu.local.

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Sun Oct 23 15:38:02 EDT 2022
;; MSG SIZE  rcvd: 154

root@occc-VirtualBox:/etc/bind#
```

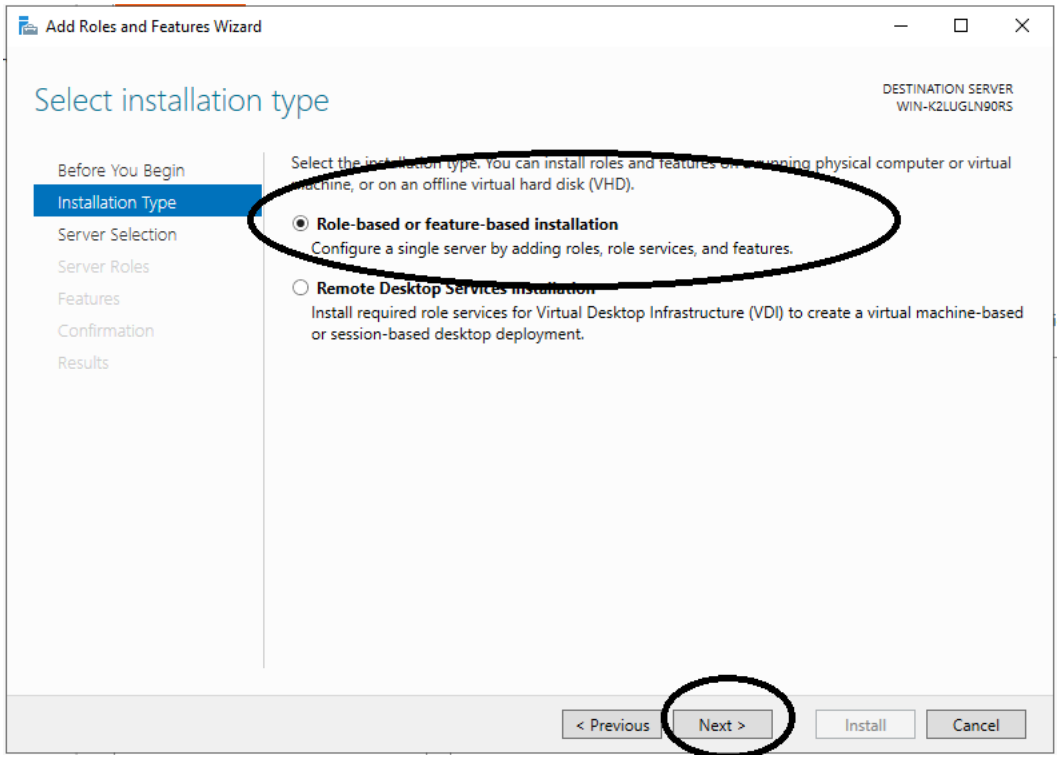
4.0.4 Windows DNS server install

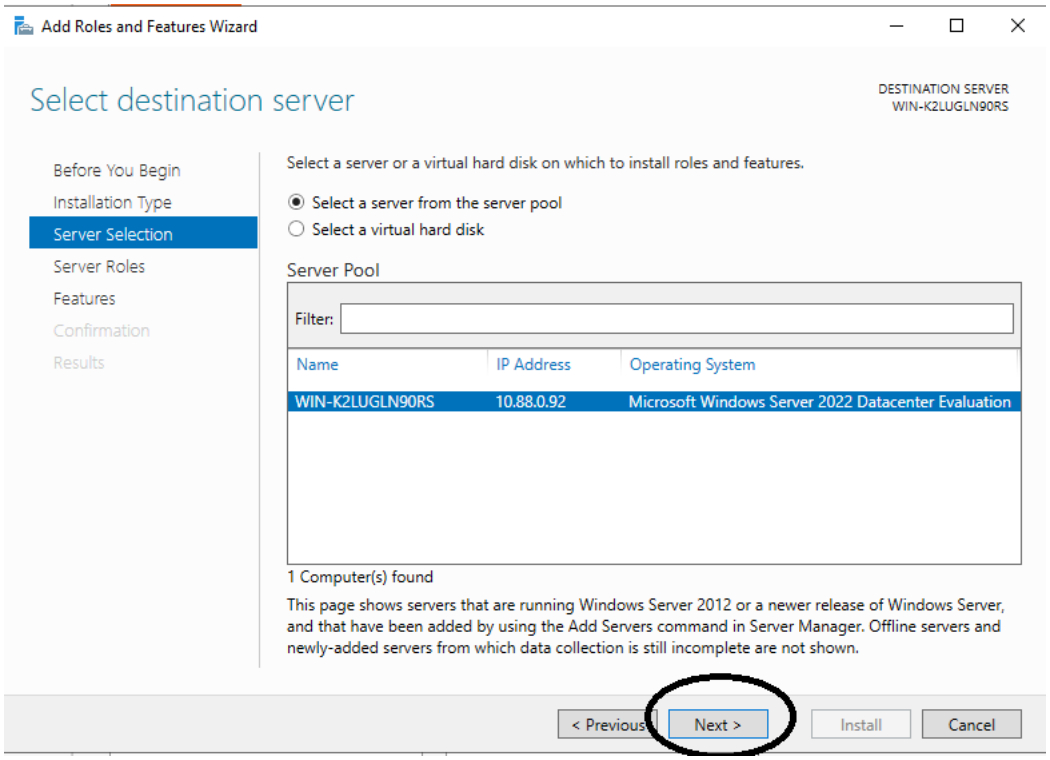
Open Server Manager. To open Server Manager, click Start, and then click Server Manager.

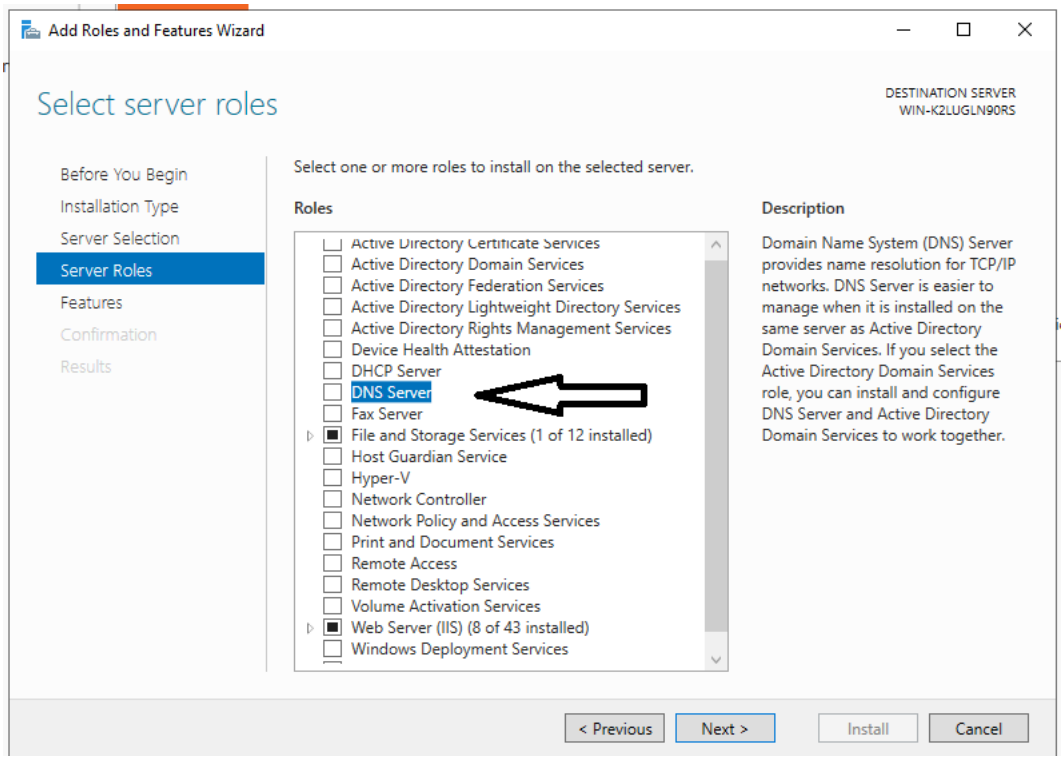
Under Roles Summary, click Add Roles.

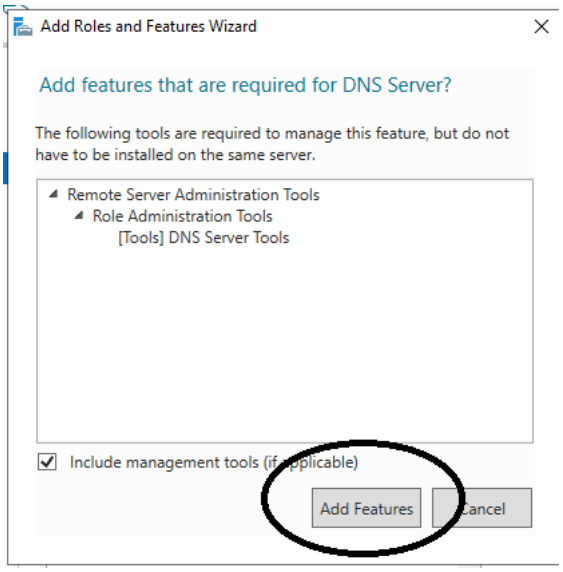


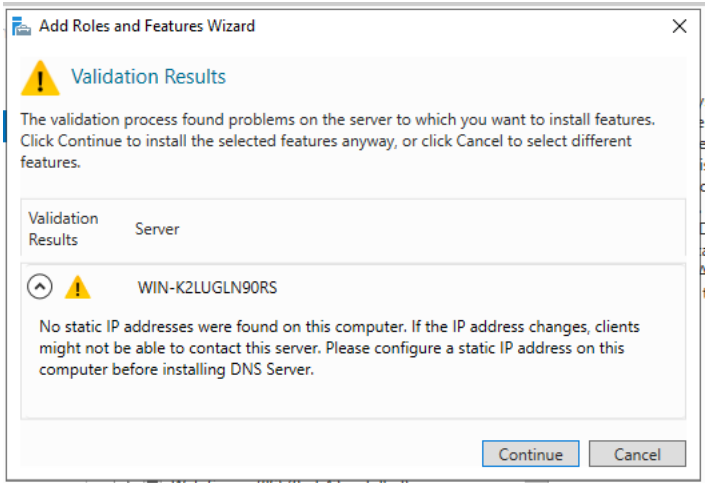

Add Roles and Features Wizard
Before you begin
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.
To remove roles, role services, or features:
Start the Remove Roles and Features Wizard
Before you continue, verify that the following tasks have been completed:
• The Administrator account has a strong password
• Network settings, such as static IP addresses, are configured
• The most current security updates from Windows Update are installed
If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.
To continue, click Next.
Skip this page by default
< Previous
Next >
Install
Cancel


Add Roles and Features Wizard
Select installation type
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Role-based or feature-based installation
Configure a single server by adding roles, role services, and features.
Remote Desktop Services installation
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.
< Previous
Next >
Install
Cancel

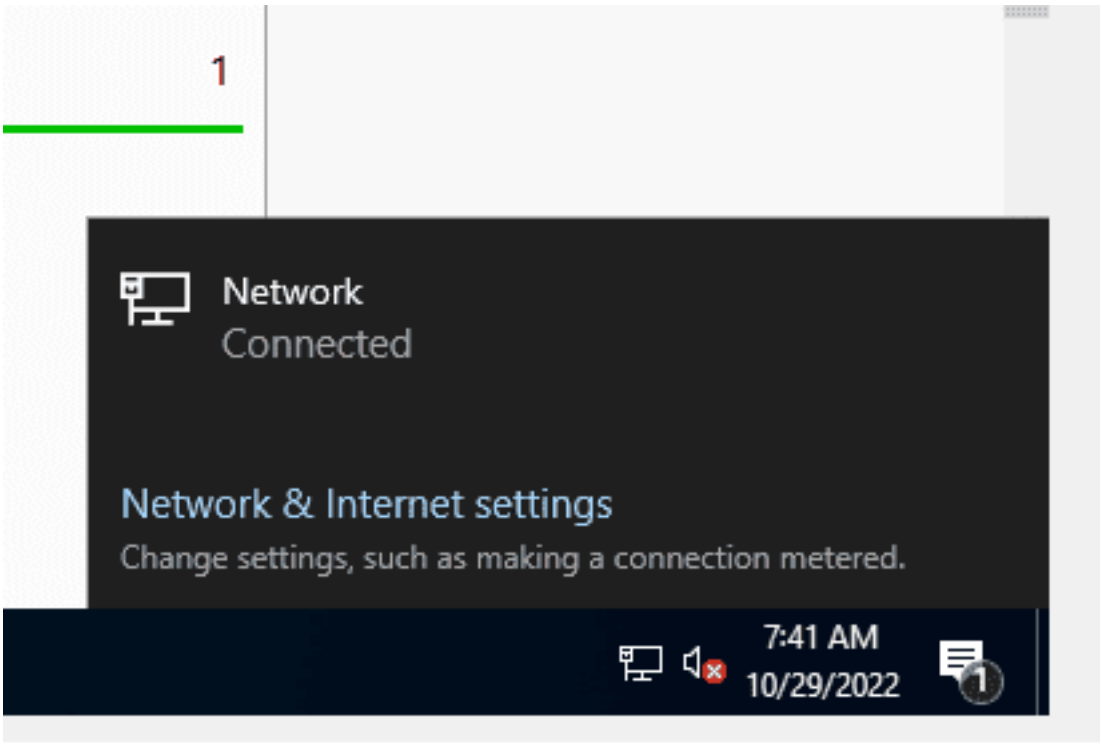

Add Roles and Features Wizard
Select destination server
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select a server or a virtual hard disk on which to install roles and features.
Select a server from the server pool
Select a virtual hard disk
Server Pool
Filter:
Name
IP Address
Operating System
WIN-K2LUGLN90RS
10.88.0.92
Microsoft Windows Server 2022 Datacenter Evaluation
1 Computer(s) found
This page shows servers that are running Windows Server 2012 or a newer release of Windows Server, and that have been added by using the Add Servers command in Server Manager. Offline servers and newly-added servers from which data collection is still incomplete are not shown.
< Previous
Next >
Install
Cancel

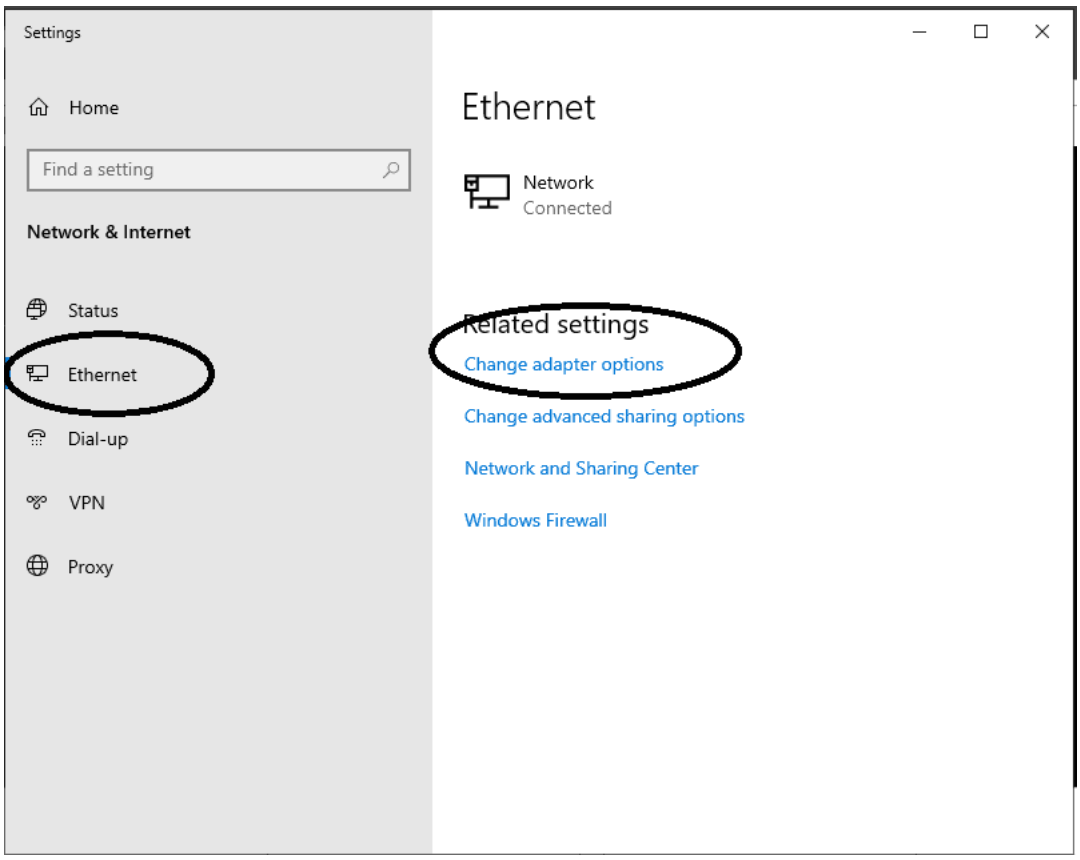

Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Active Directory Certificate Services
Active Directory Domain Services
Active Directory Federation Services
Active Directory Lightweight Directory Services
Active Directory Rights Management Services
Device Health Attestation
DHCP Server
DNS Server
Fax Server
File and Storage Services (1 of 12 installed)
Host Guardian Service
Hyper-V
Network Controller
Network Policy and Access Services
Print and Document Services
Remote Access
Remote Desktop Services
Volume Activation Services
Web Server (IIS) (8 of 43 installed)
Windows Deployment Services
Description
Domain Name System (DNS) Server provides name resolution for TCP/IP networks. DNS Server is easier to manage when it is installed on the same server as Active Directory Domain Services. If you select the Active Directory Domain Services role, you can install and configure DNS Server and Active Directory Domain Services to work together.
< Previous
Next >
Install
Cancel

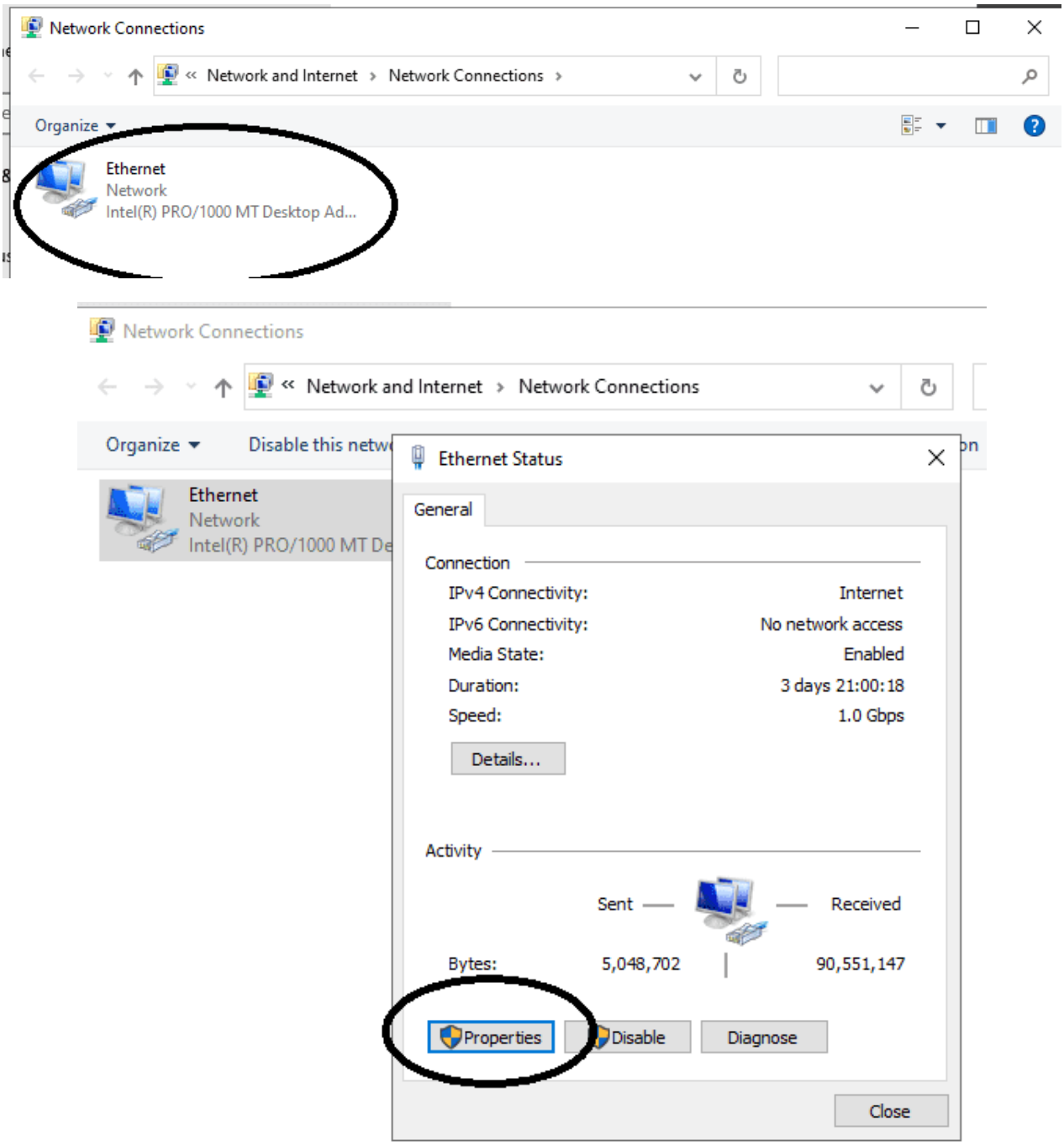


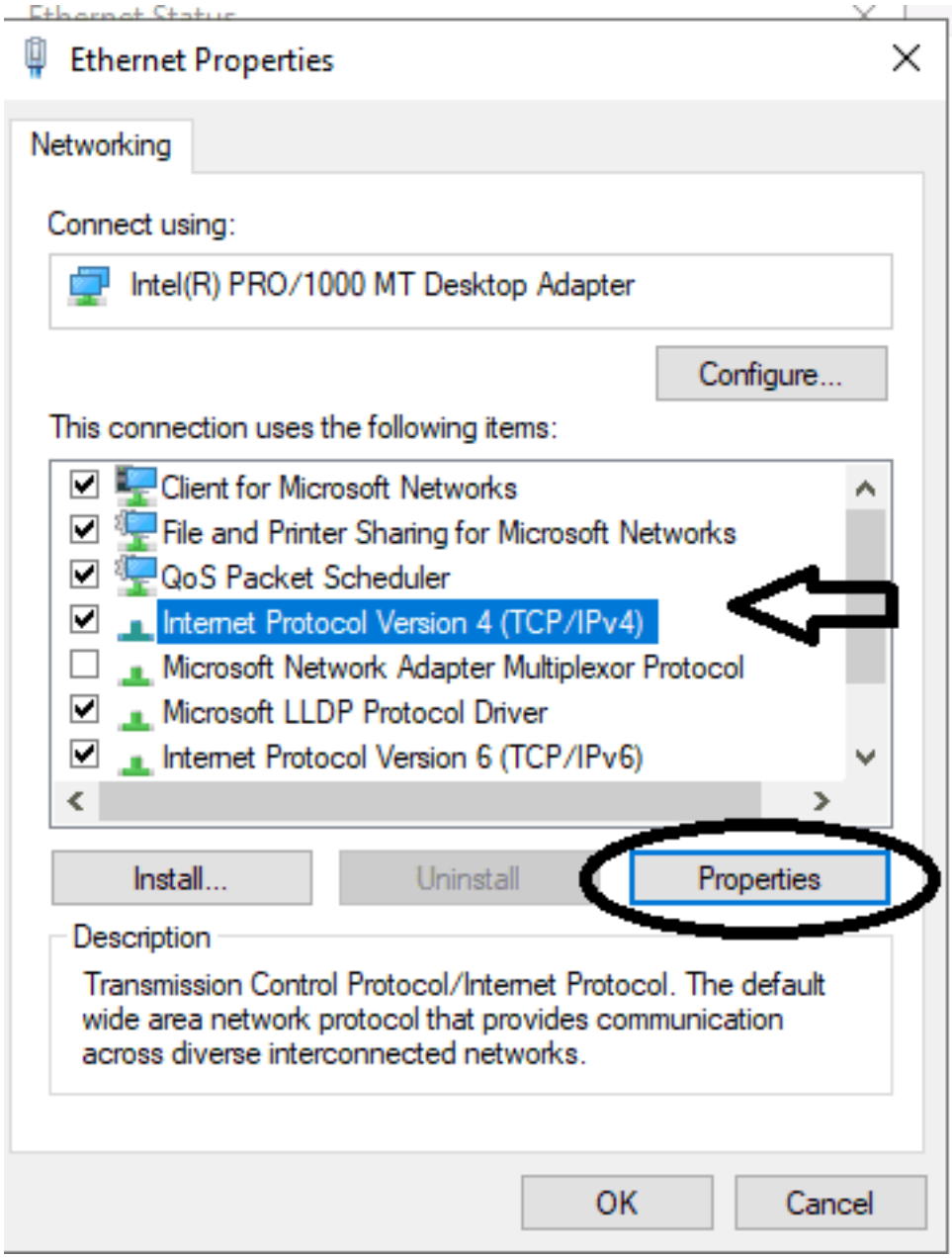


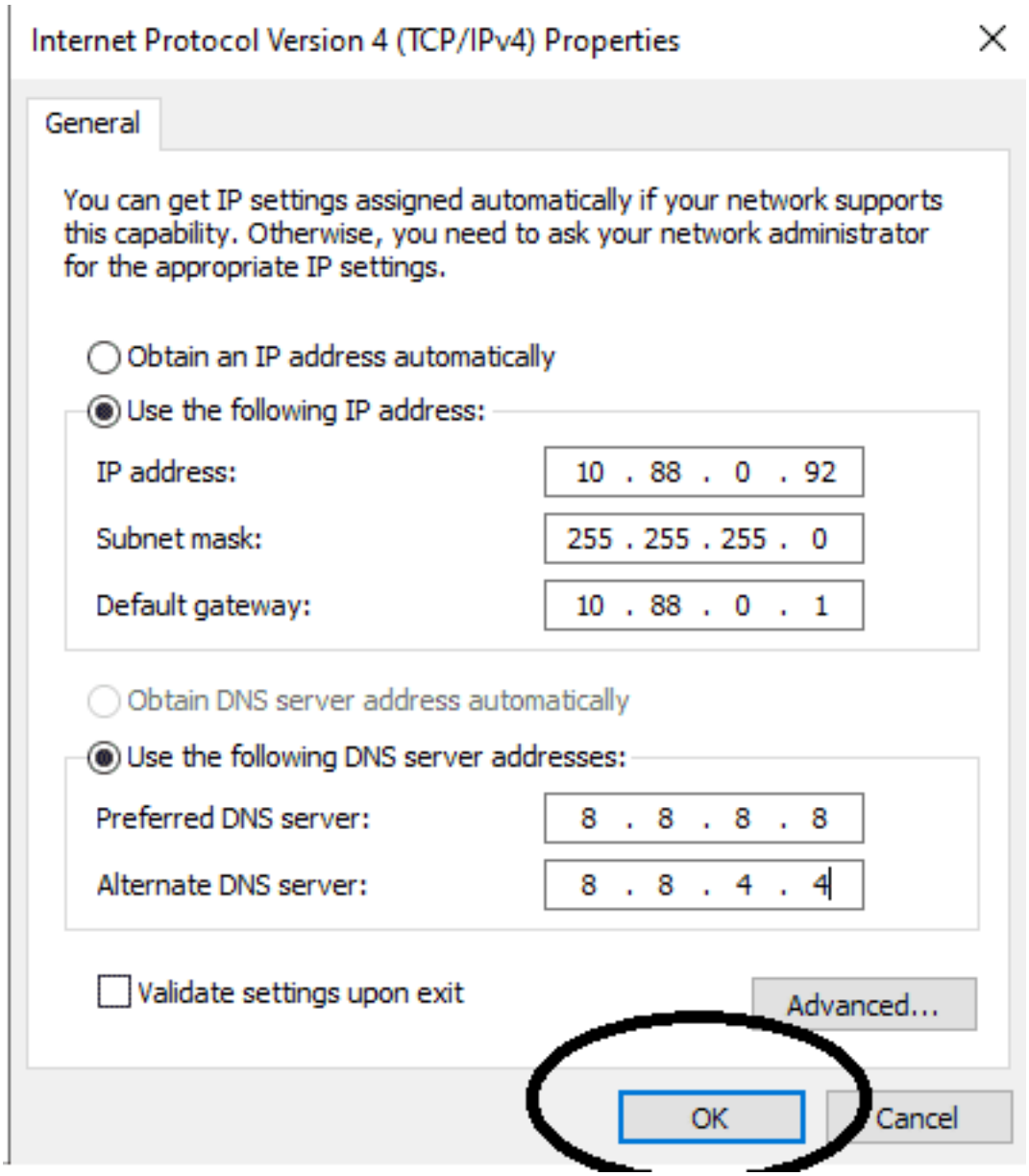
If you get this cancel and make sure you have a static ip address.


1
Network
Connected
Network & Internet settings
Change settings, such as making a connection metered.
7:41 AM
10/29/2022

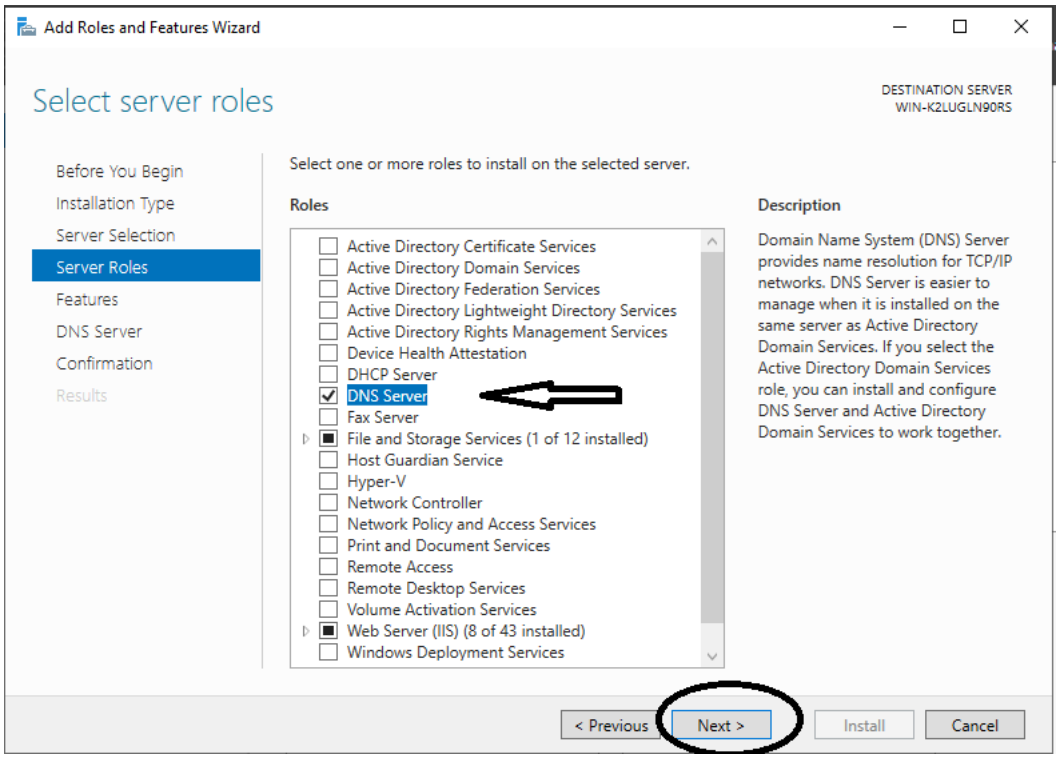

Settings
Home
Find a setting
Network & Internet
Status
Ethernet
Dial-up
VPN
Proxy
Ethernet
Network
Connected
Related settings
Change adapter options
Change advanced sharing options
Network and Sharing Center
Windows Firewall

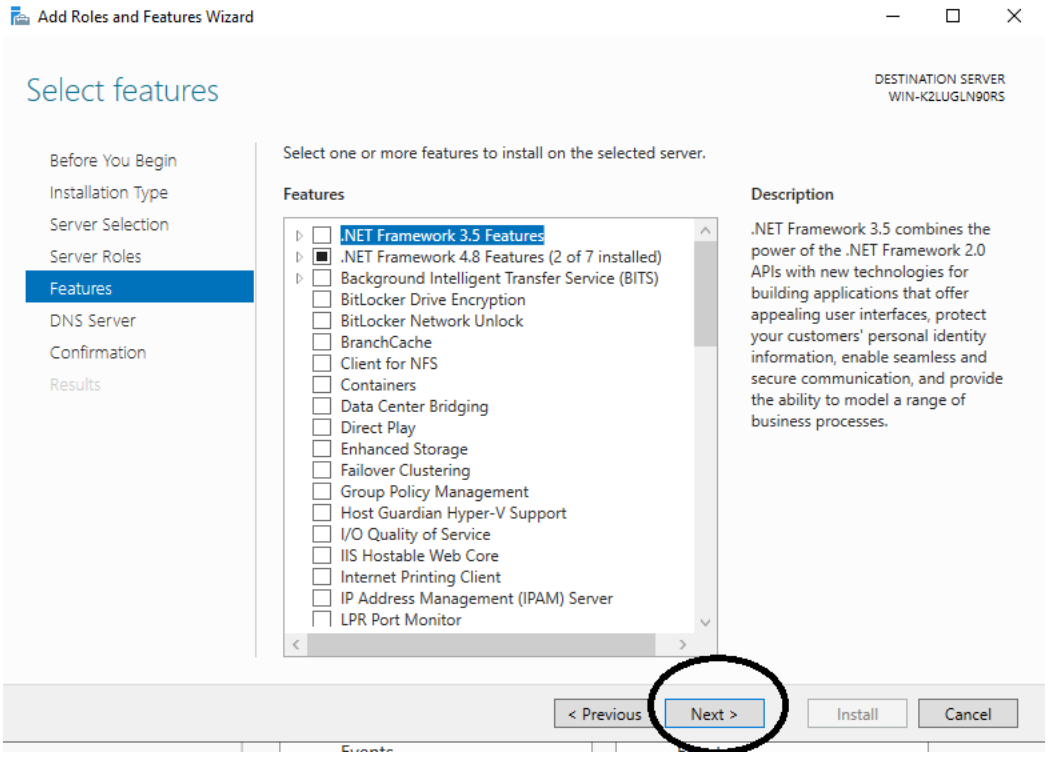

Network Connections
« Network and Internet › Network Connections ›
Organize
Ethernet
Network
Intel(R) PRO/1000 MT Desktop Ad...
Network Connections
« Network and Internet › Network Connections
Organize
Disable this netw
Ethernet
Network
Intel(R) PRO/1000 MT De
Ethernet Status
General
Connection
IPv4 Connectivity: Internet
IPv6 Connectivity: No network access
Media State: Enabled
Duration: 3 days 21:00:18
Speed: 1.0 Gbps
Details...
Activity
Sent
Received
Bytes: 5,048,702 90,551,147
Properties
Disable
Diagnose
Close

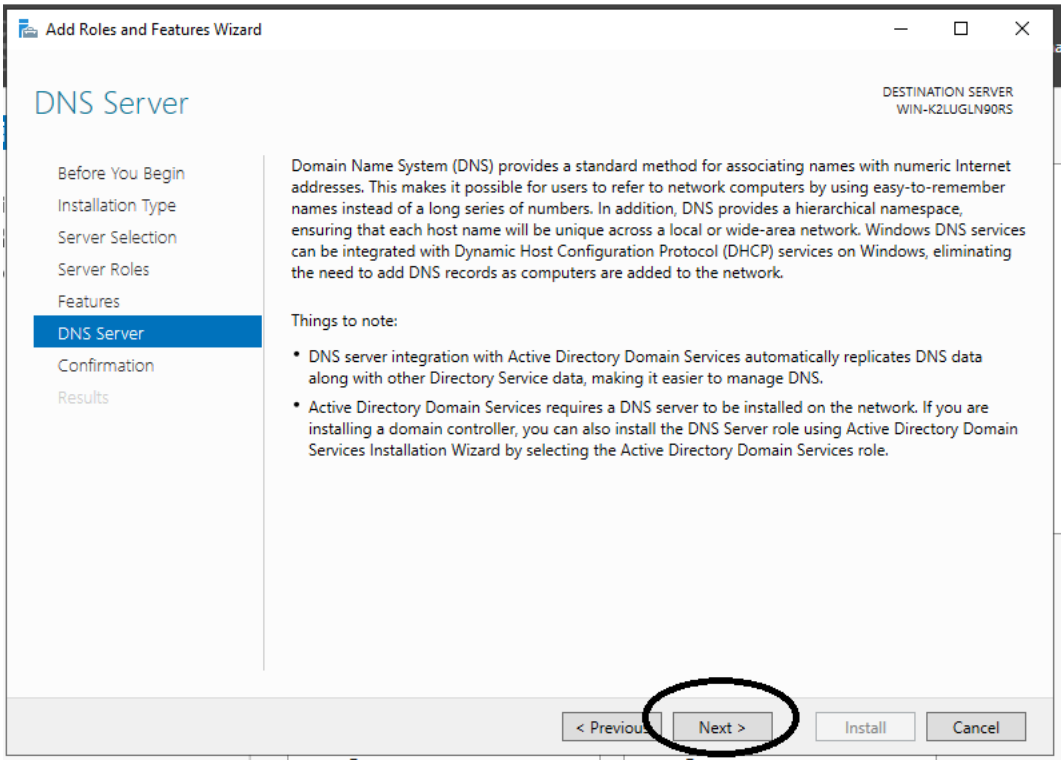

Ethernet Properties
Networking
Connect using:
Intel(R) PRO/1000 MT Desktop Adapter
Configure...
This connection uses the following items:
Client for Microsoft Networks
File and Printer Sharing for Microsoft Networks
QoS Packet Scheduler
Internet Protocol Version 4 (TCP/IPv4)
Microsoft Network Adapter Multiplexor Protocol
Microsoft LLDP Protocol Driver
Internet Protocol Version 6 (TCP/IPv6)
Install...
Uninstall
Properties
Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.
OK
Cancel

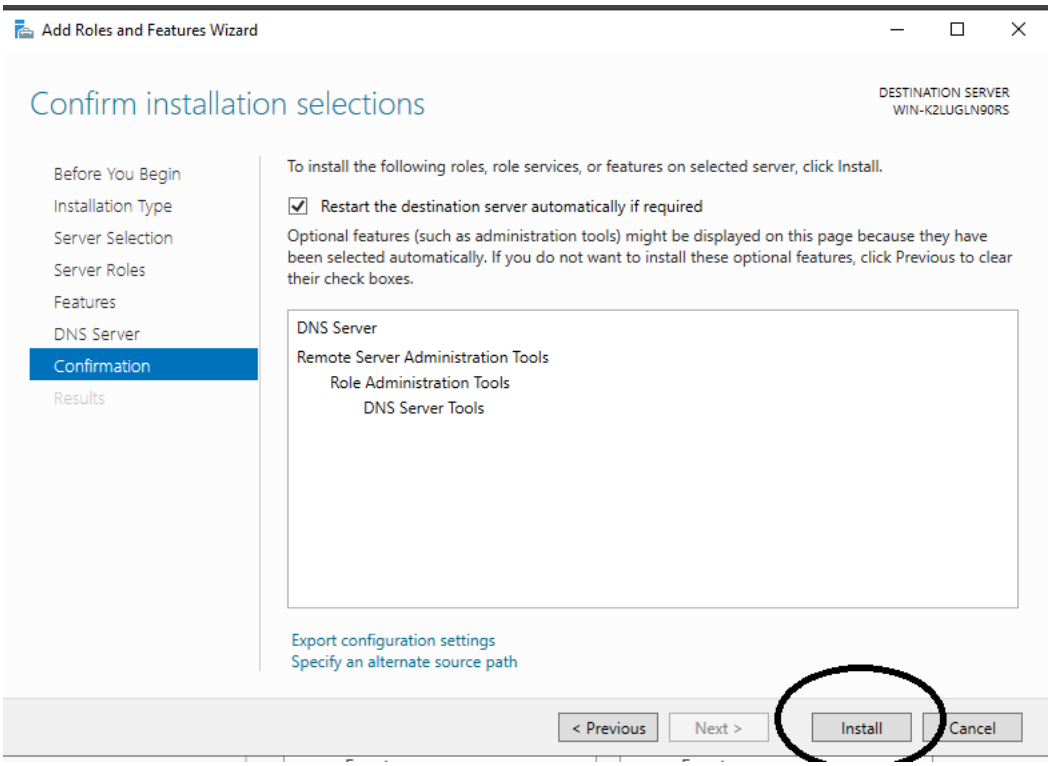


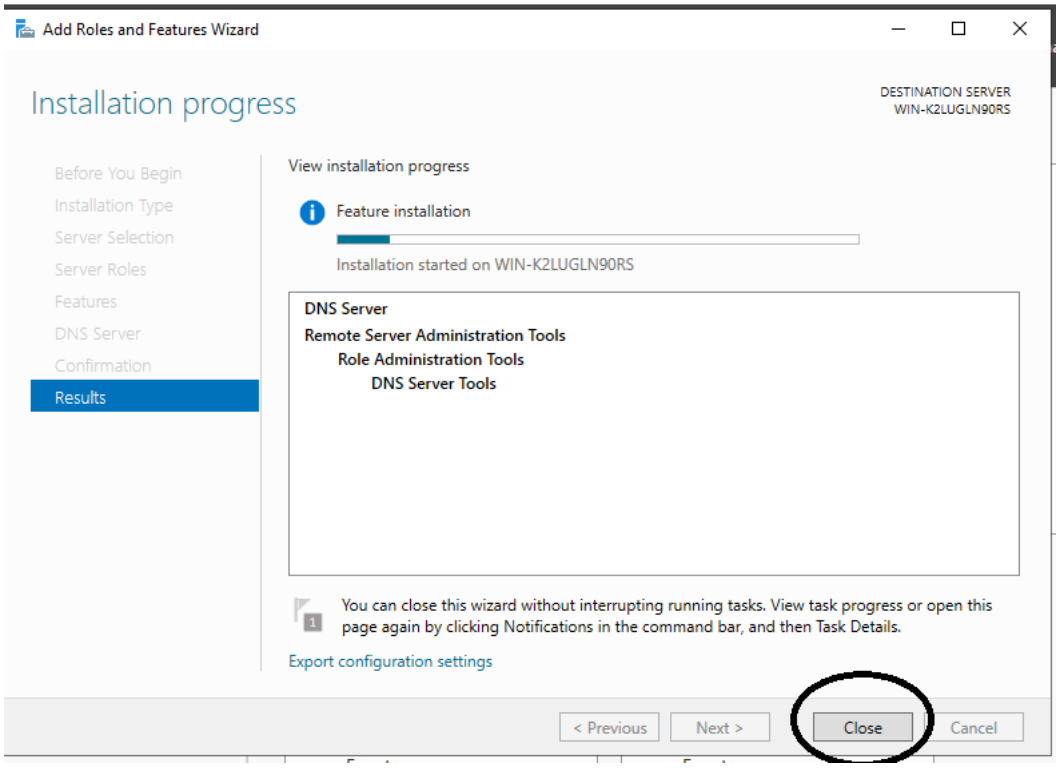
Go back to DNS install.


Add Roles and Features Wizard
Select server roles
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
DNS Server
Confirmation
Results
Select one or more roles to install on the selected server.
Roles
Active Directory Certificate Services
Active Directory Domain Services
Active Directory Federation Services
Active Directory Lightweight Directory Services
Active Directory Rights Management Services
Device Health Attestation
DHCP Server
DNS Server
Fax Server
File and Storage Services (1 of 12 installed)
Host Guardian Service
Hyper-V
Network Controller
Network Policy and Access Services
Print and Document Services
Remote Access
Remote Desktop Services
Volume Activation Services
Web Server (IIS) (8 of 43 installed)
Windows Deployment Services
Description
Domain Name System (DNS) Server provides name resolution for TCP/IP networks. DNS Server is easier to manage when it is installed on the same server as Active Directory Domain Services. If you select the Active Directory Domain Services role, you can install and configure DNS Server and Active Directory Domain Services to work together.
< Previous
Next >
Install
Cancel

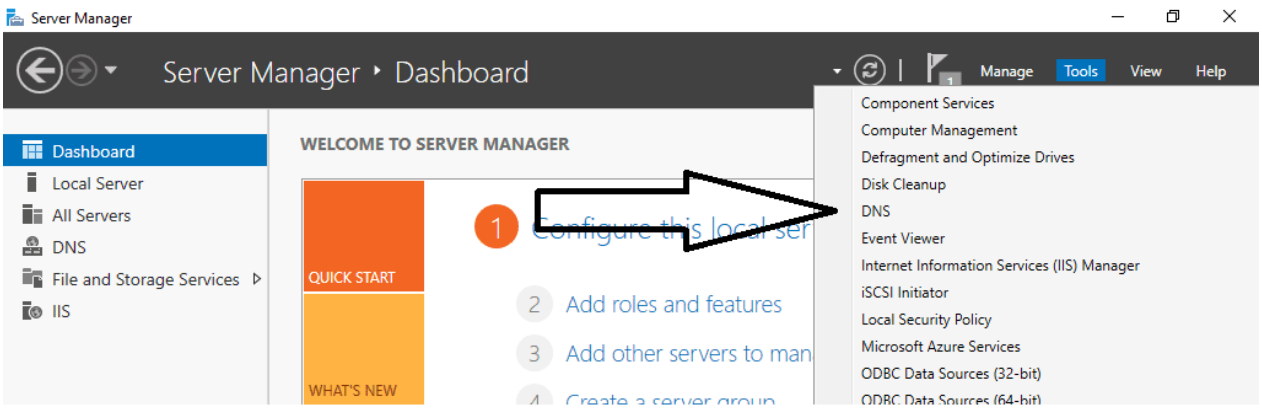

Add Roles and Features Wizard
Select features
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
DNS Server
Confirmation
Results
Select one or more features to install on the selected server.
Features
.NET Framework 3.5 Features
.NET Framework 4.8 Features (2 of 7 installed)
Background Intelligent Transfer Service (BITS)
BitLocker Drive Encryption
BitLocker Network Unlock
BranchCache
Client for NFS
Containers
Data Center Bridging
Direct Play
Enhanced Storage
Failover Clustering
Group Policy Management
Host Guardian Hyper-V Support
I/O Quality of Service
IIS Hostable Web Core
Internet Printing Client
IP Address Management (IPAM) Server
LPR Port Monitor
Description
.NET Framework 3.5 combines the power of the .NET Framework 2.0 APIs with new technologies for building applications that offer appealing user interfaces, protect your customers' personal identity information, enable seamless and secure communication, and provide the ability to model a range of business processes.
< Previous
Next >
Install
Cancel

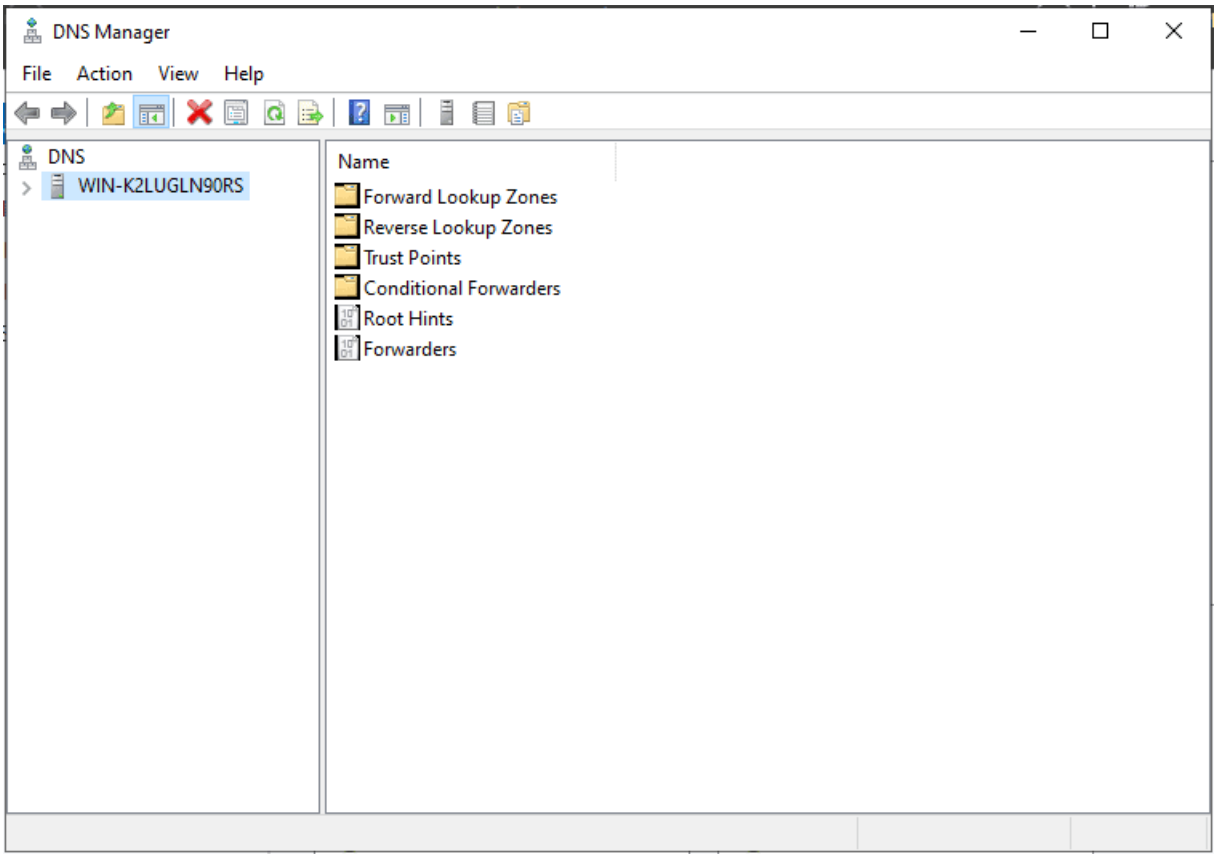

Add Roles and Features Wizard
DNS Server
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
DNS Server
Confirmation
Results
Domain Name System (DNS) provides a standard method for associating names with numeric Internet addresses. This makes it possible for users to refer to network computers by using easy-to-remember names instead of a long series of numbers. In addition, DNS provides a hierarchical namespace, ensuring that each host name will be unique across a local or wide-area network. Windows DNS services can be integrated with Dynamic Host Configuration Protocol (DHCP) services on Windows, eliminating the need to add DNS records as computers are added to the network.
Things to note:
DNS server integration with Active Directory Domain Services automatically replicates DNS data along with other Directory Service data, making it easier to manage DNS.
Active Directory Domain Services requires a DNS server to be installed on the network. If you are installing a domain controller, you can also install the DNS Server role using Active Directory Domain Services Installation Wizard by selecting the Active Directory Domain Services role.
< Previou
Next >
Install
Cancel

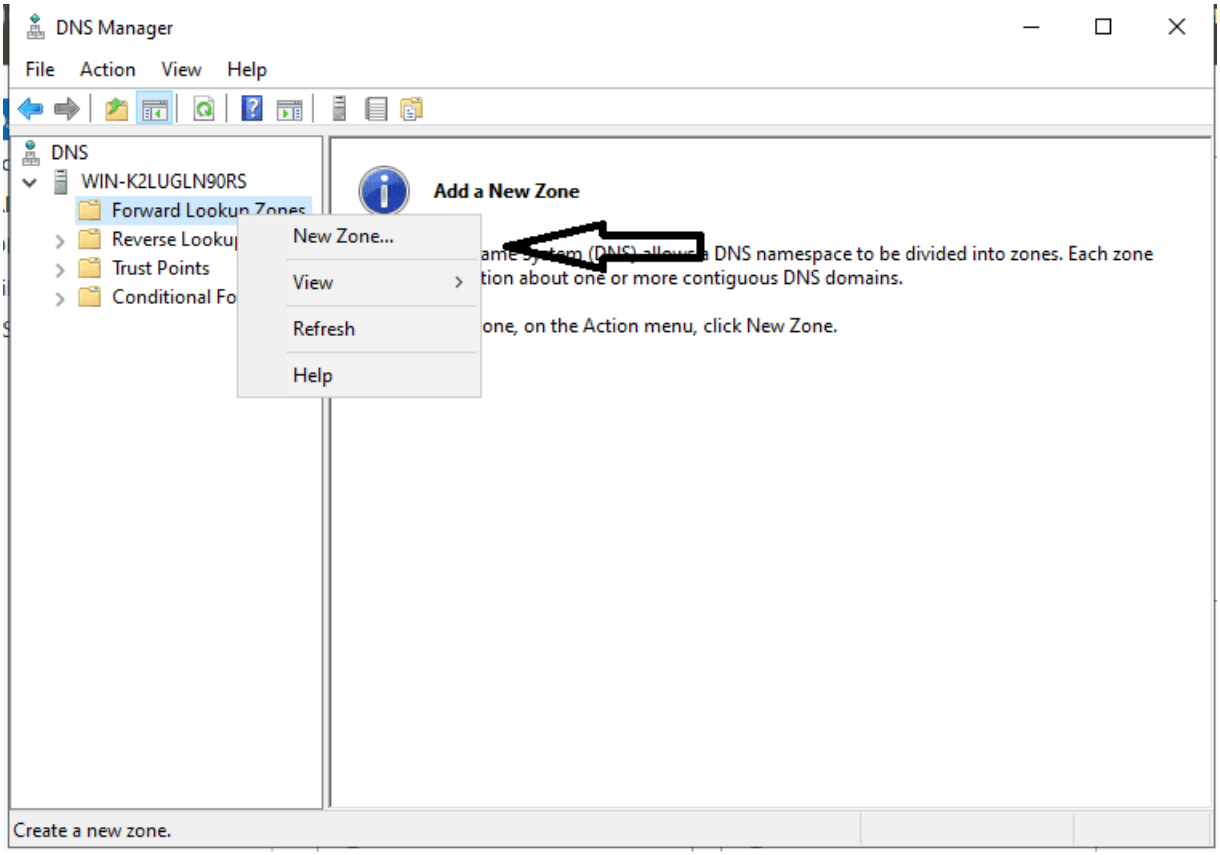

Add Roles and Features Wizard
Confirm installation selections
DESTINATION SERVER
WIN-K2LUGLN90RS
Before You Begin
Installation Type
Server Selection
Server Roles
Features
DNS Server
Confirmation
Results
To install the following roles, role services, or features on selected server, click Install.
Restart the destination server automatically if required
Optional features (such as administration tools) might be displayed on this page because they have been selected automatically. If you do not want to install these optional features, click Previous to clear their check boxes.
DNS Server
Remote Server Administration Tools
Role Administration Tools
DNS Server Tools
Export configuration settings
Specify an alternate source path
< Previous
Next >
Install
Cancel



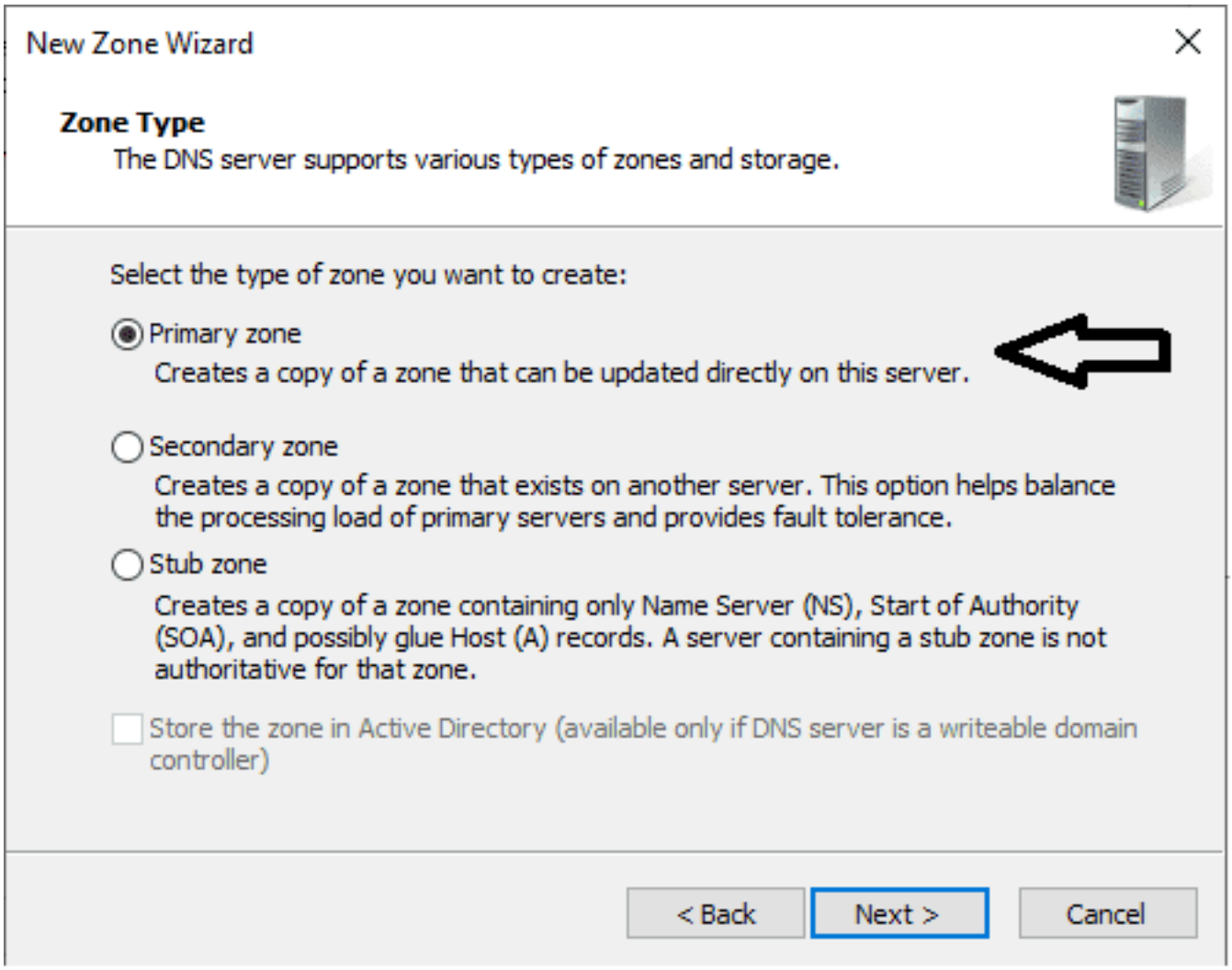
DNS Forward Lookup Zone in Windows

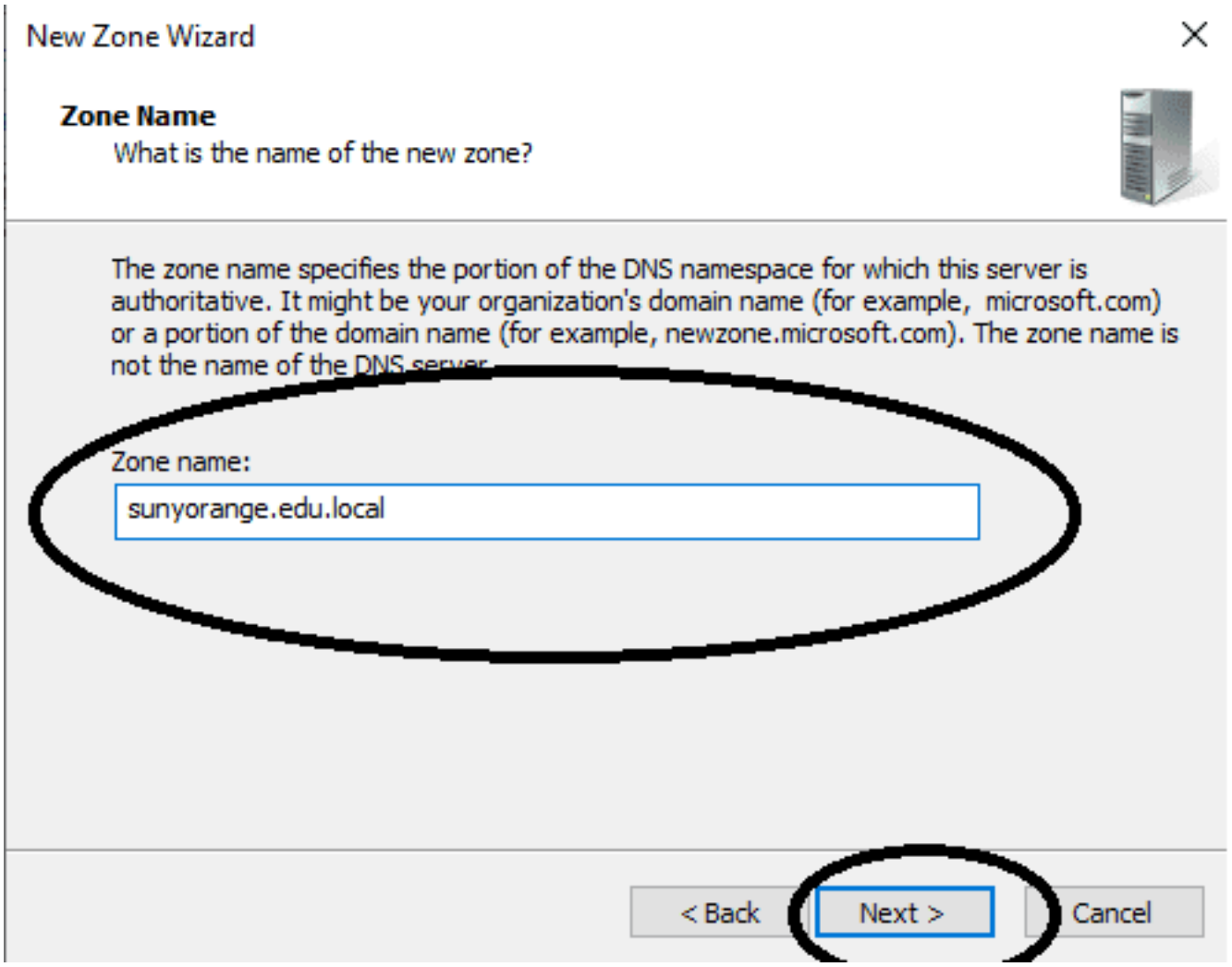


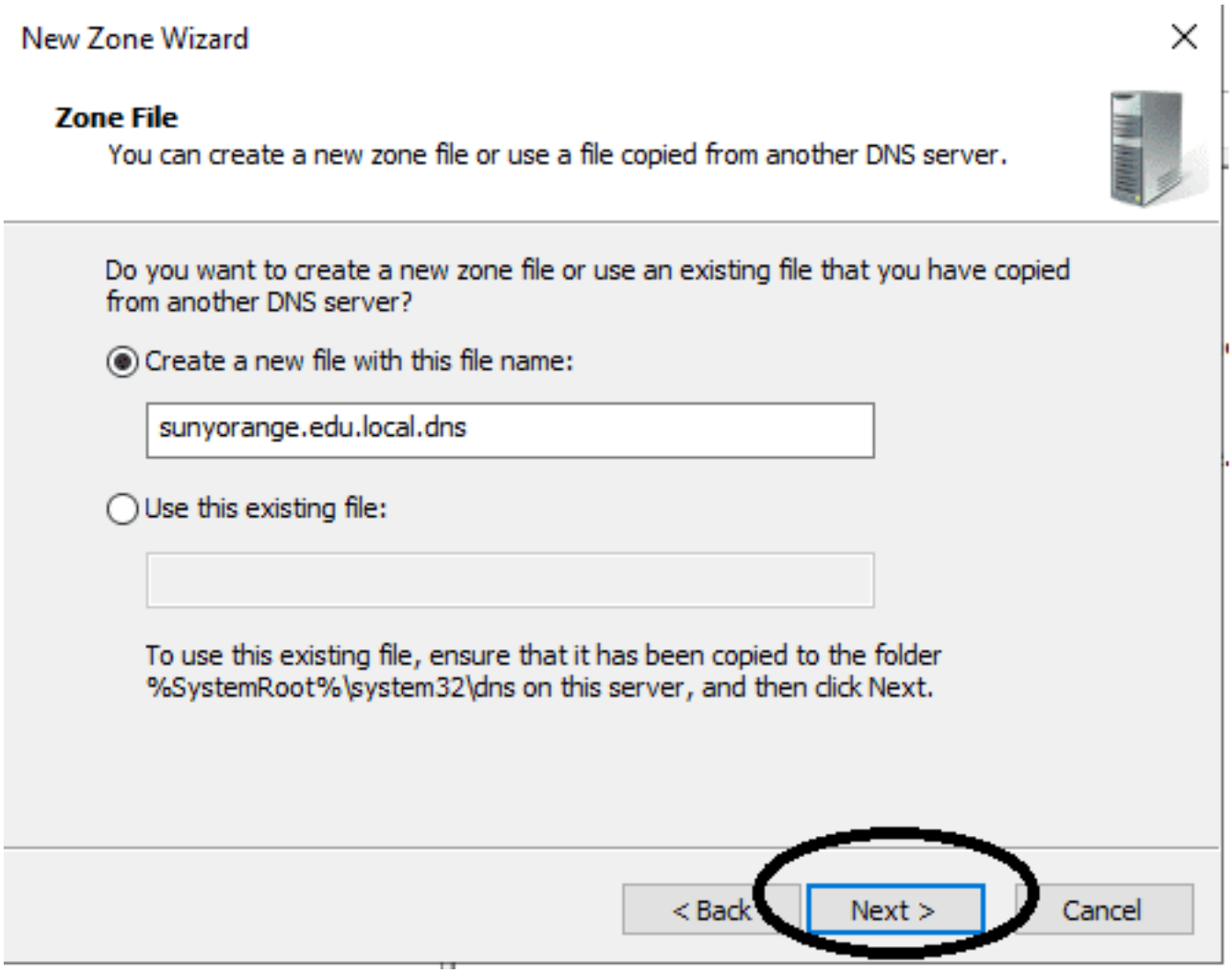

DNS Manager
File Action View Help
DNS
WIN-K2LUGLN90RS
Name
Forward Lookup Zones
Reverse Lookup Zones
Trust Points
Conditional Forwarders
Root Hints
Forwarders

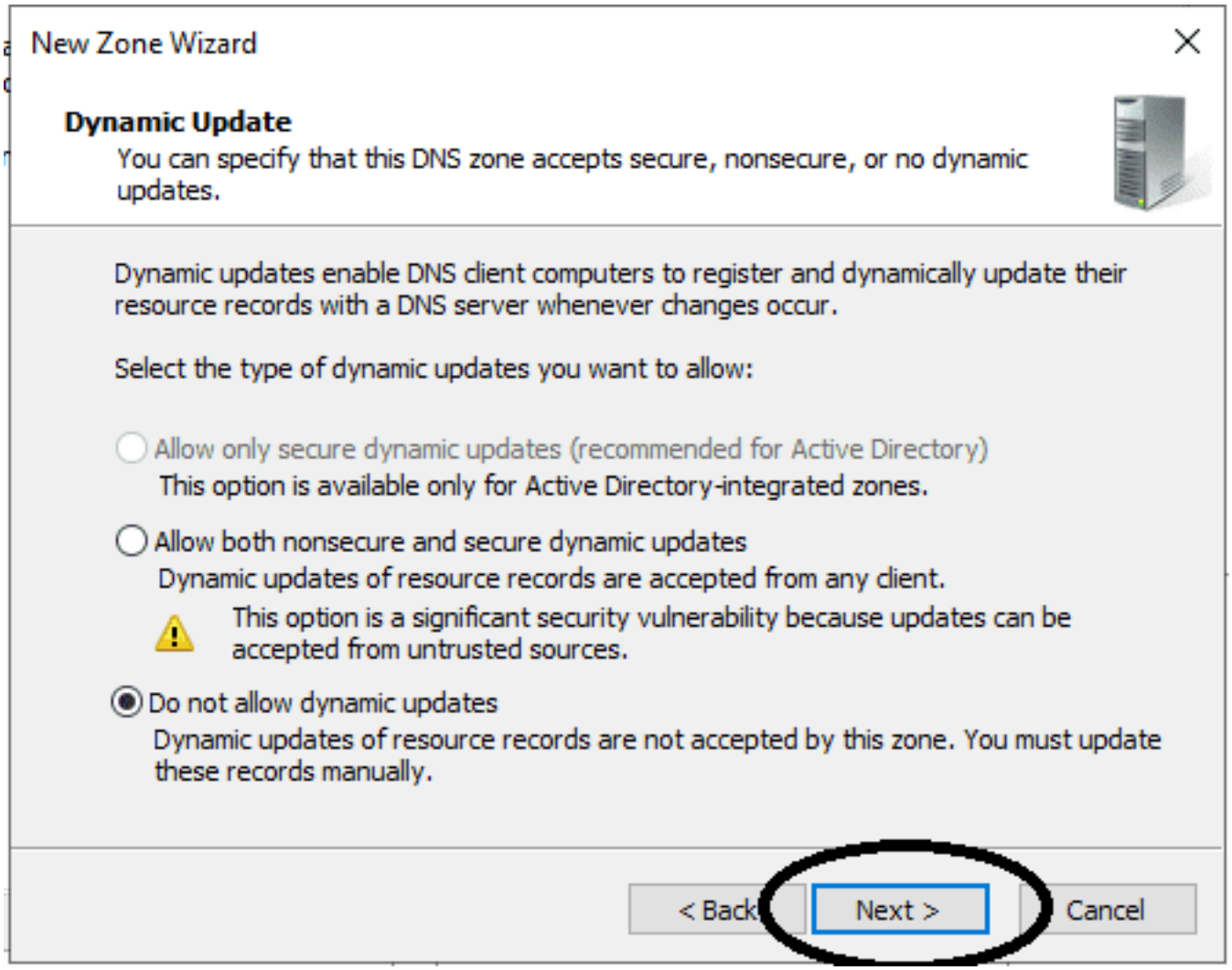

DNS Manager
File Action View Help
DNS
WIN-K2LUGLN90RS
Forward Lookup Zones
Reverse Lookup
Trust Points
Conditional Fo
New Zone...
View
Refresh
Help
Add a New Zone
DNS namespace to be divided into zones. Each zone
tion about one or more contiguous DNS domains.
one, on the Action menu, click New Zone.
Create a new zone.

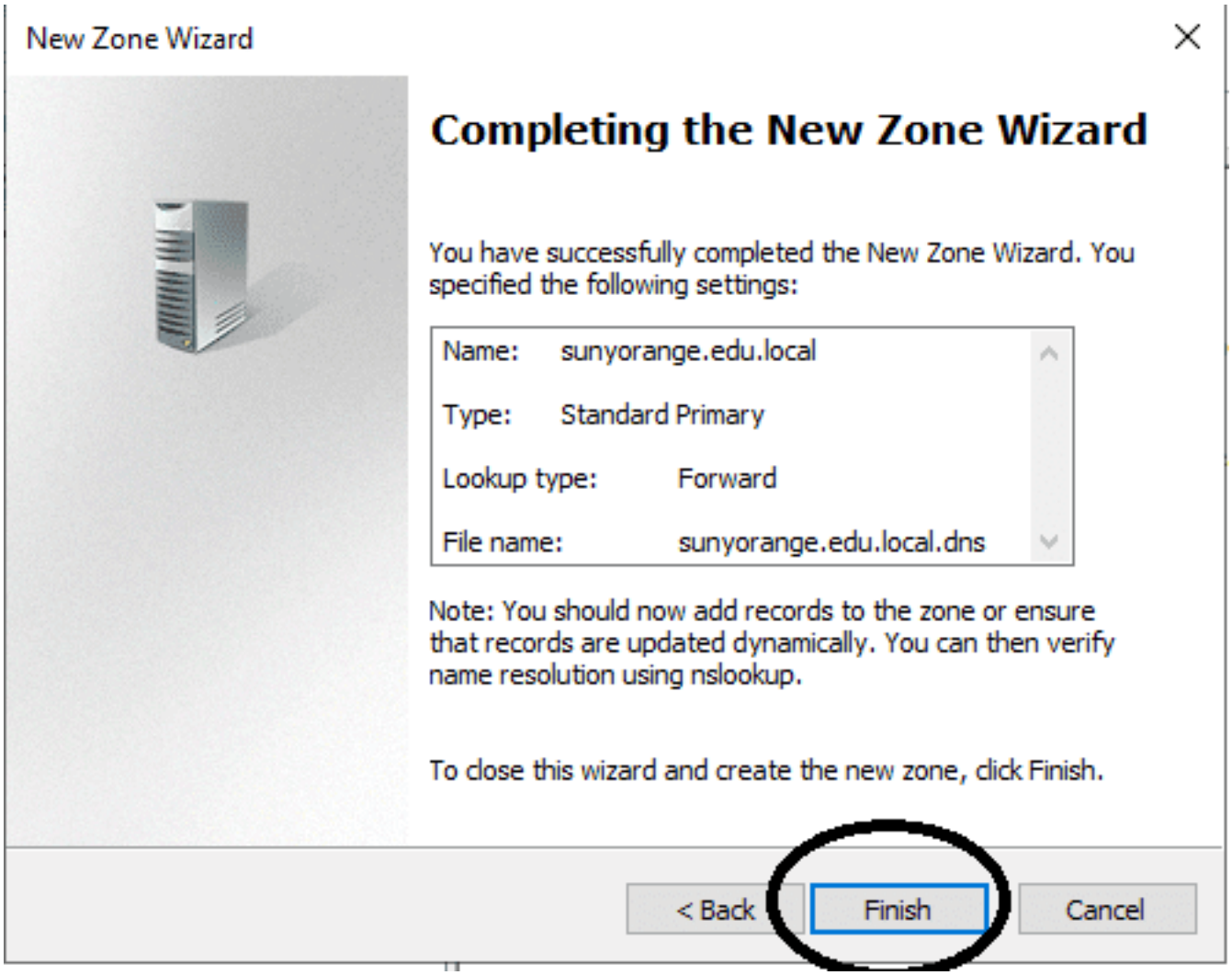

New Zone Wizard
Welcome to the New Zone Wizard
This wizard helps you create a new zone for your DNS server.
A zone translates DNS names to related data, such as IP addresses or network services.
To continue, click Next.
< Ba
Next >
Cancel

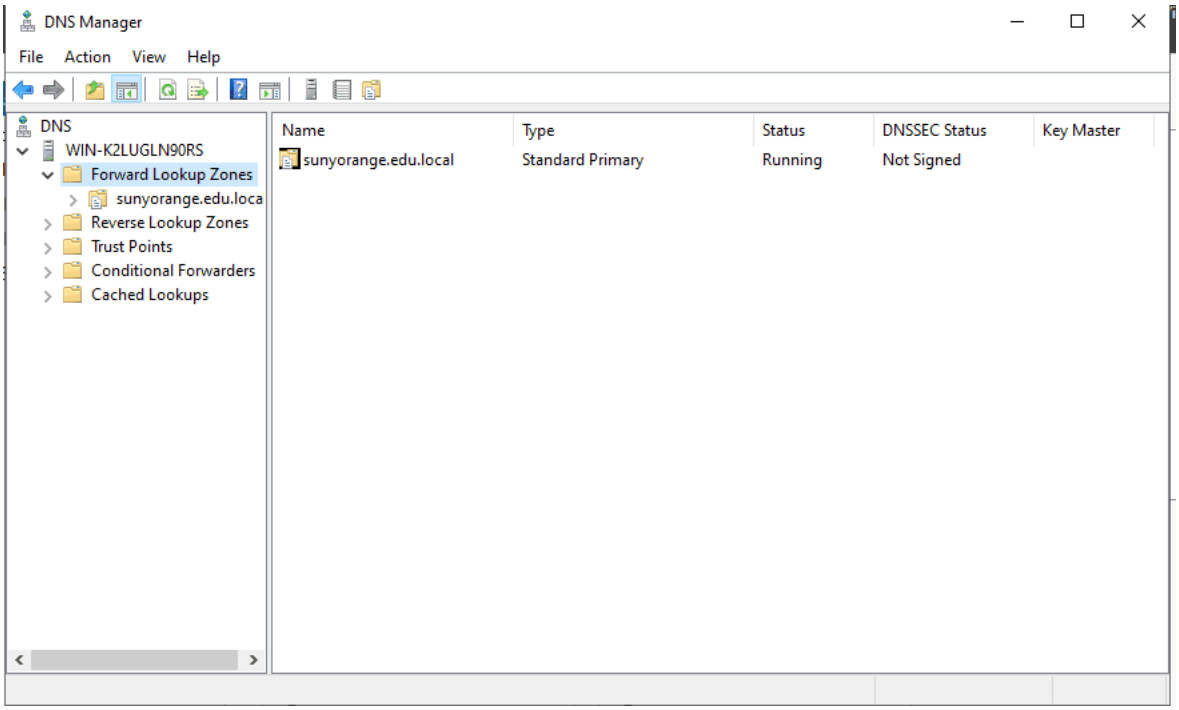

New Zone Wizard
Zone Type
The DNS server supports various types of zones and storage.
Select the type of zone you want to create:
Primary zone
Creates a copy of a zone that can be updated directly on this server.
Secondary zone
Creates a copy of a zone that exists on another server. This option helps balance the processing load of primary servers and provides fault tolerance.
Stub zone
Creates a copy of a zone containing only Name Server (NS), Start of Authority (SOA), and possibly glue Host (A) records. A server containing a stub zone is not authoritative for that zone.
Store the zone in Active Directory (available only if DNS server is a writeable domain controller)
< Back
Next >
Cancel


New Zone Wizard
Zone Name
What is the name of the new zone?
The zone name specifies the portion of the DNS namespace for which this server is authoritative. It might be your organization's domain name (for example, microsoft.com) or a portion of the domain name (for example, newzone.microsoft.com). The zone name is not the name of the DNS server.
Zone name:
sunyorange.edu.local
< Back
Next >
Cancel

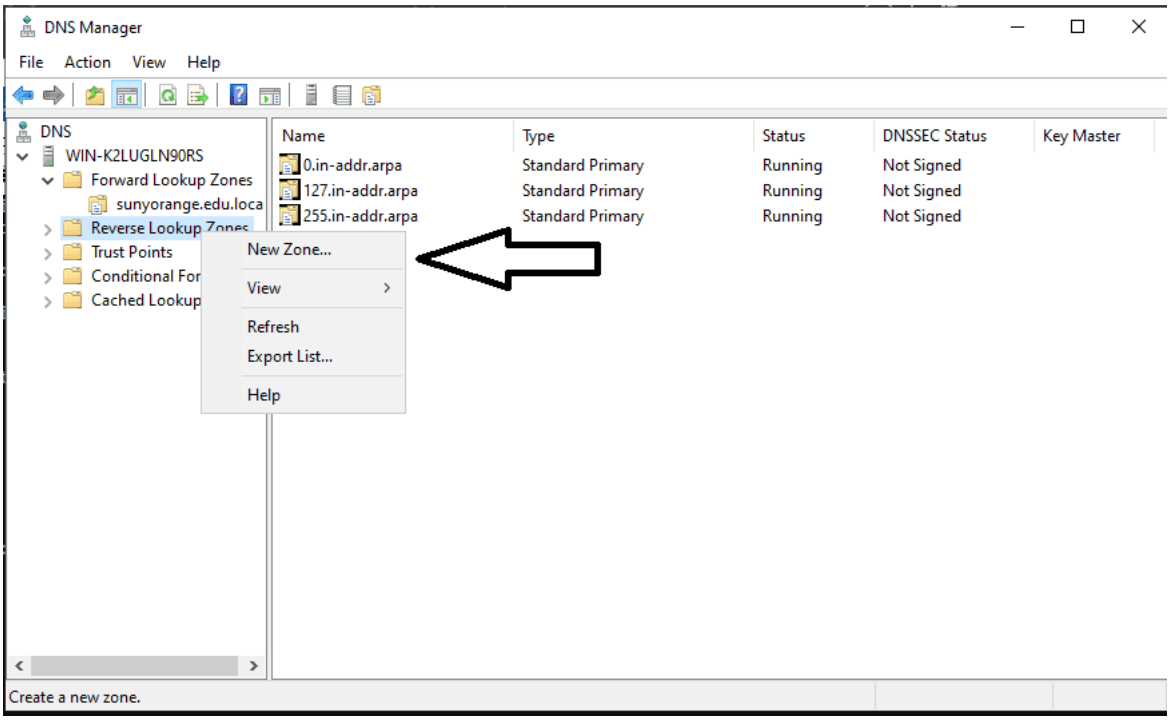

New Zone Wizard
Zone File
You can create a new zone file or use a file copied from another DNS server.
Do you want to create a new zone file or use an existing file that you have copied from another DNS server?
Create a new file with this file name:
sunyorange.edu.local.dns
Use this existing file:
To use this existing file, ensure that it has been copied to the folder %SystemRoot%\system32\dns on this server, and then click Next.
< Back
Next >
Cancel


New Zone Wizard
Dynamic Update
You can specify that this DNS zone accepts secure, nonsecure, or no dynamic updates.
Dynamic updates enable DNS client computers to register and dynamically update their resource records with a DNS server whenever changes occur.
Select the type of dynamic updates you want to allow:
Allow only secure dynamic updates (recommended for Active Directory)
This option is available only for Active Directory-integrated zones.
Allow both nonsecure and secure dynamic updates
Dynamic updates of resource records are accepted from any client.
This option is a significant security vulnerability because updates can be accepted from untrusted sources.
Do not allow dynamic updates
Dynamic updates of resource records are not accepted by this zone. You must update these records manually.
< Back
Next >
Cancel

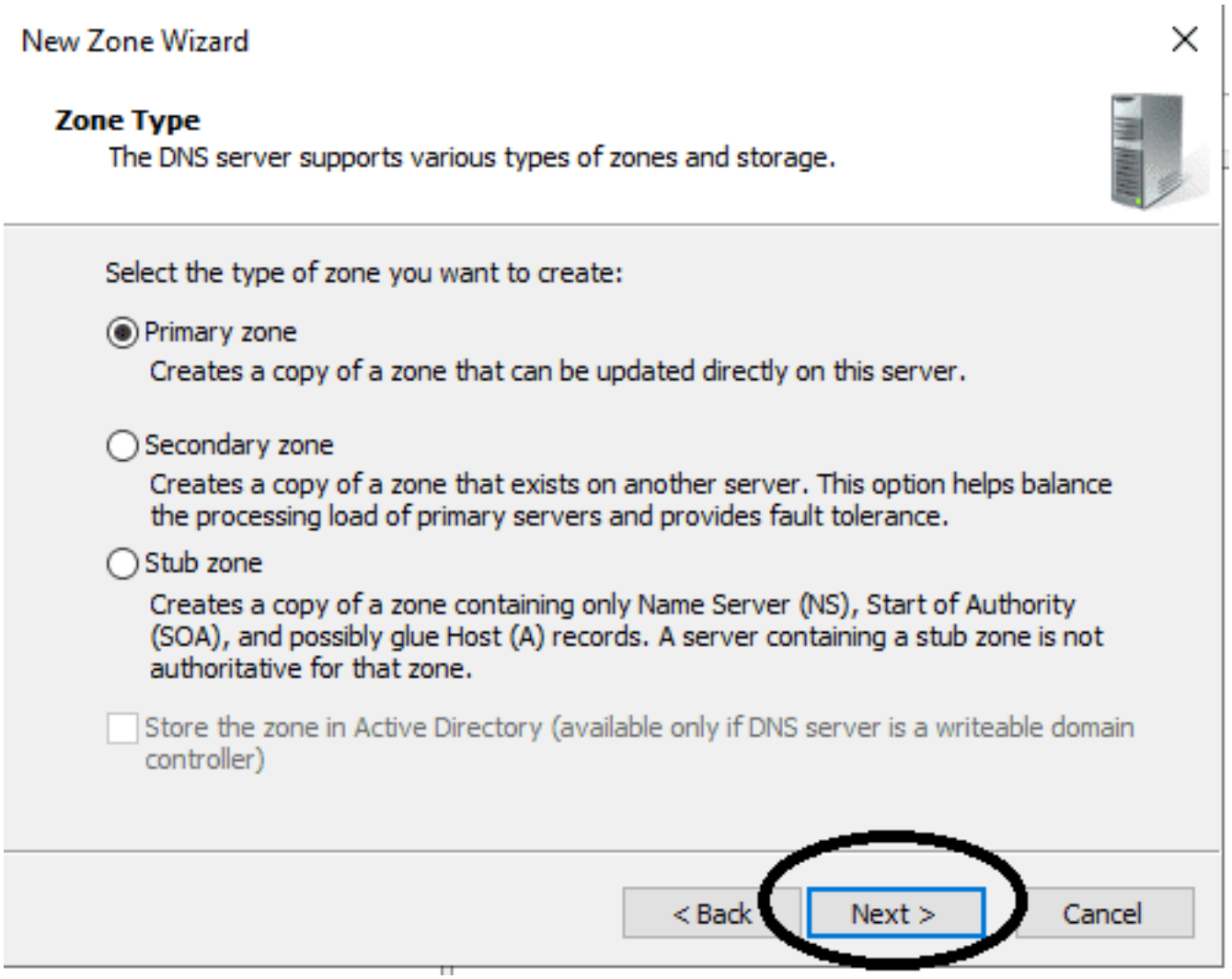


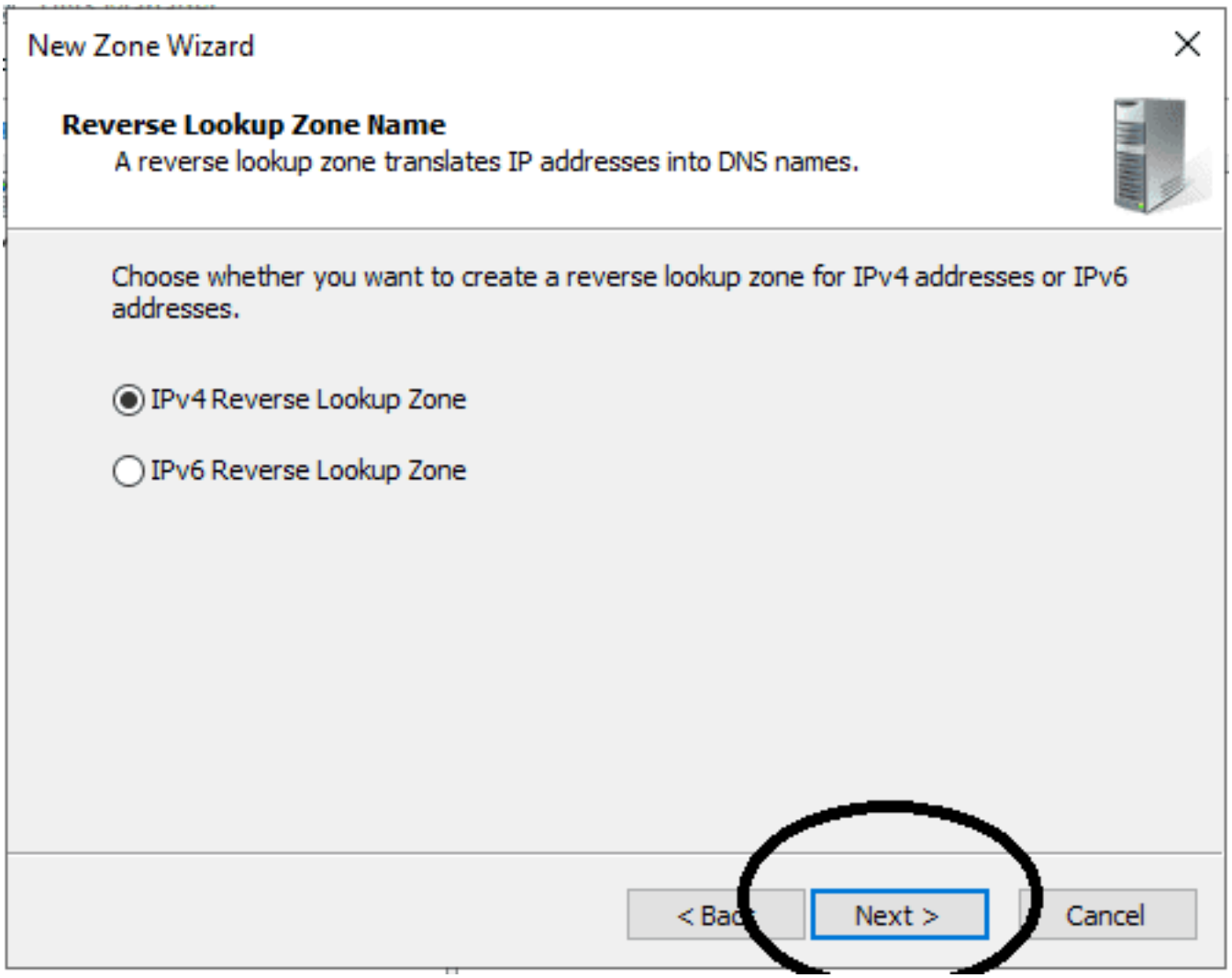


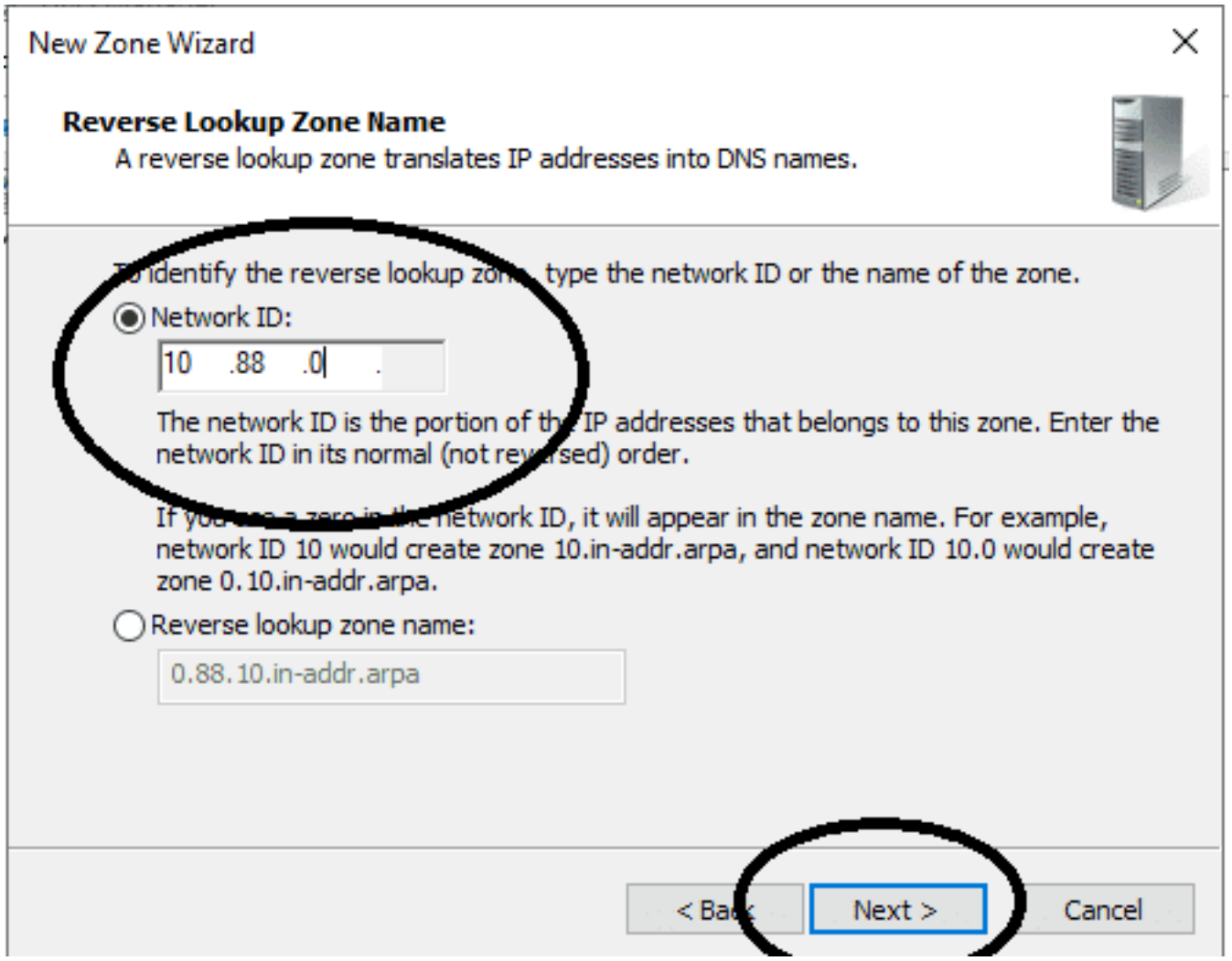
Create a reverse lookup zone Windows

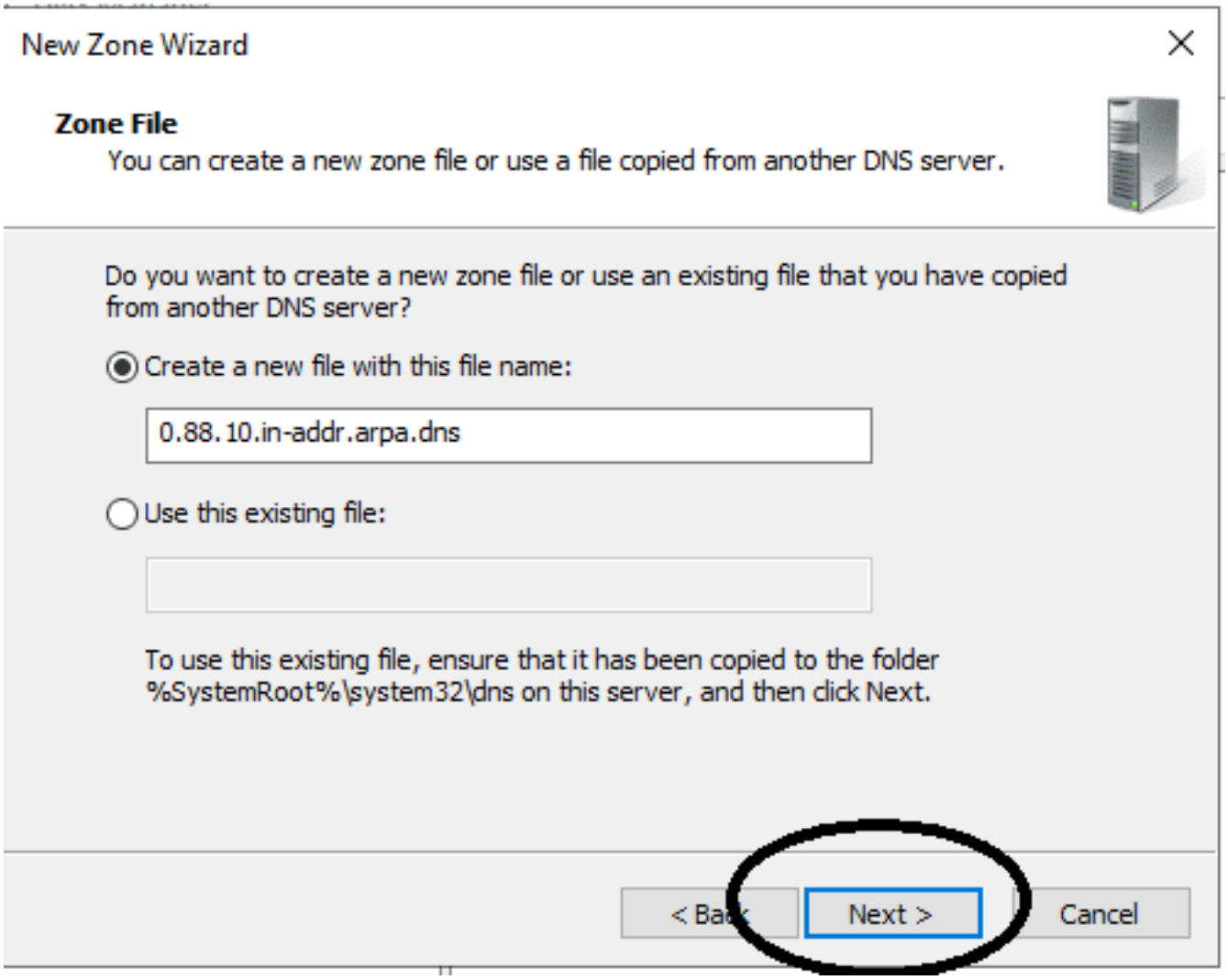

DNS Manager
File Action View Help
DNS
WIN-K2LUGLN90RS
Forward Lookup Zones
sunyorange.edu.loca
Reverse Lookup Zones
Trust Points
Conditional For
Cached Lookup
Name
Type
Status
DNSSEC Status
Key Master
0.in-addr.arpa
127.in-addr.arpa
255.in-addr.arpa
Standard Primary
Running
Not Signed
New Zone...
View
Refresh
Export List...
Help
Create a new zone.

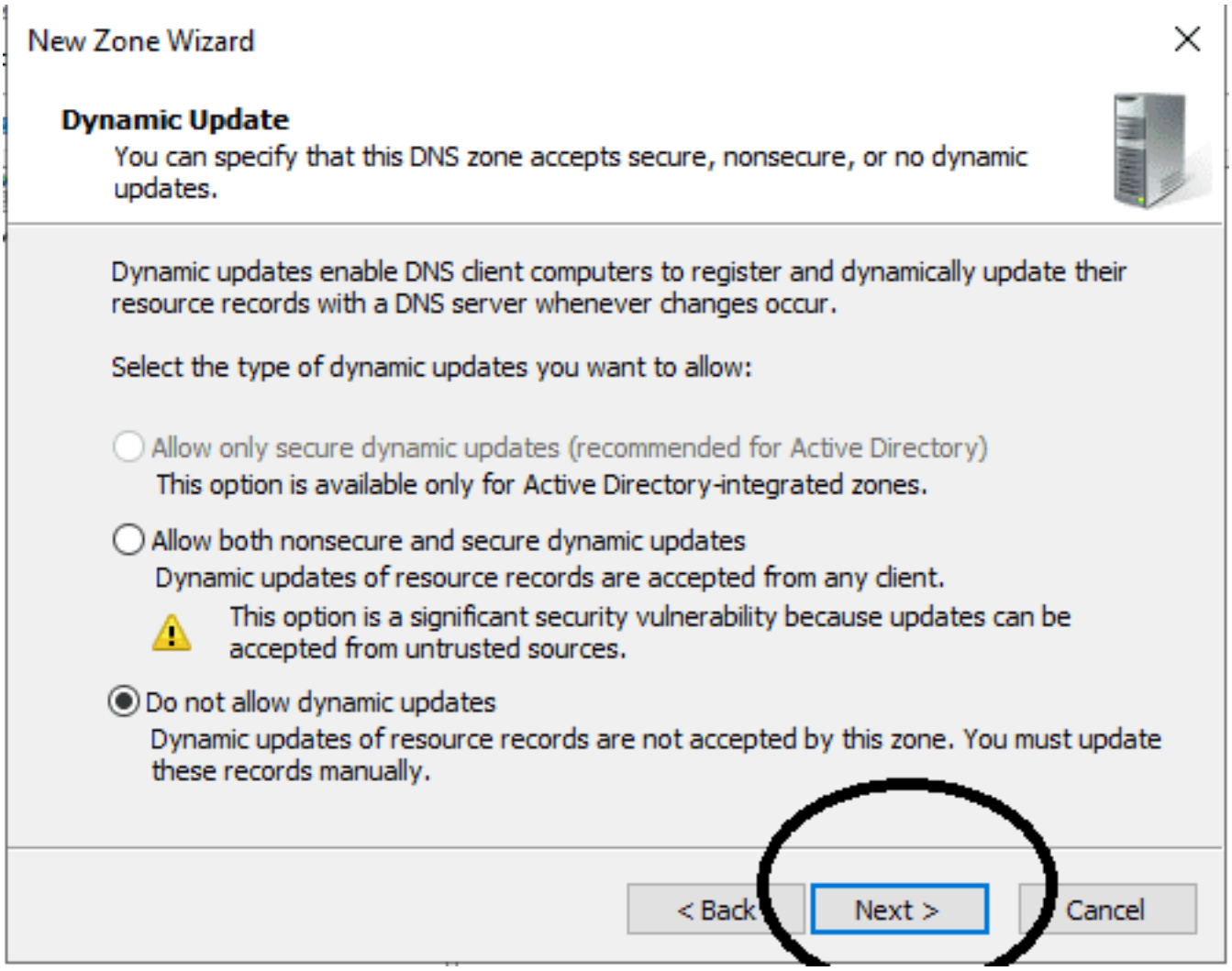

New Zone Wizard
Welcome to the New Zone Wizard
This wizard helps you create a new zone for your DNS server.
A zone translates DNS names to related data, such as IP addresses or network services.
To continue, click Next.
< Back
Next >
Cancel

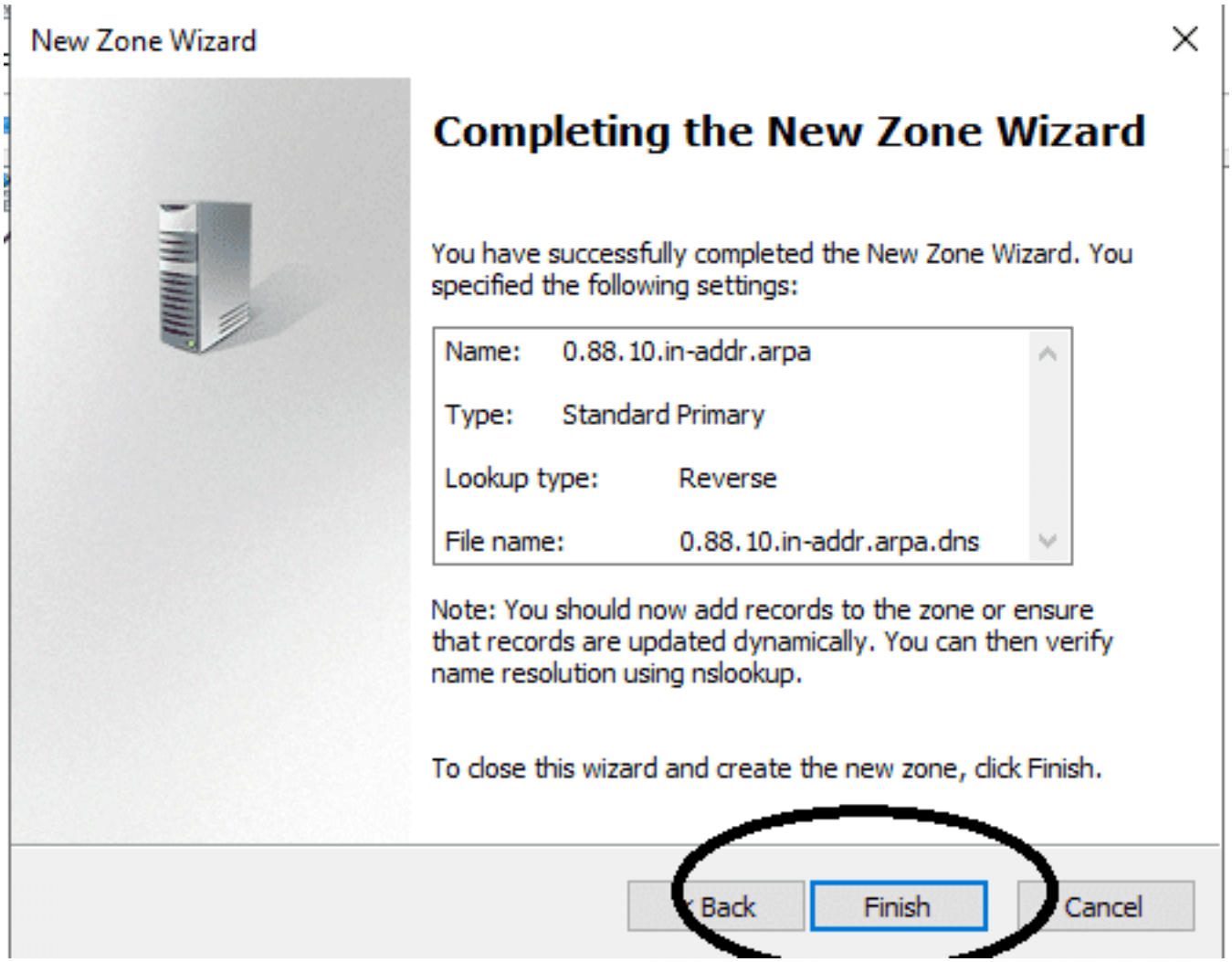

New Zone Wizard
Zone Type
The DNS server supports various types of zones and storage.
Select the type of zone you want to create:
Primary zone
Creates a copy of a zone that can be updated directly on this server.
Secondary zone
Creates a copy of a zone that exists on another server. This option helps balance the processing load of primary servers and provides fault tolerance.
Stub zone
Creates a copy of a zone containing only Name Server (NS), Start of Authority (SOA), and possibly glue Host (A) records. A server containing a stub zone is not authoritative for that zone.
Store the zone in Active Directory (available only if DNS server is a writeable domain controller)
< Back
Next >
Cancel

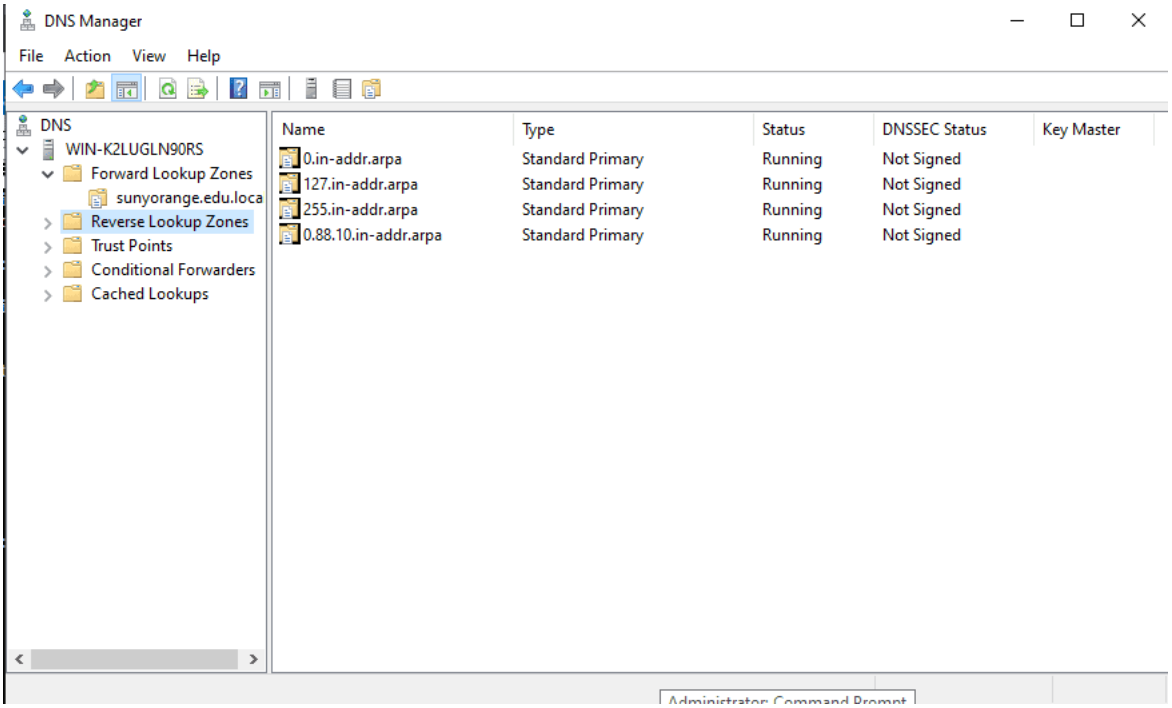

New Zone Wizard
Reverse Lookup Zone Name
A reverse lookup zone translates IP addresses into DNS names.
Choose whether you want to create a reverse lookup zone for IPv4 addresses or IPv6 addresses.
IPv4 Reverse Lookup Zone
IPv6 Reverse Lookup Zone
< Bac
Next >
Cancel


New Zone Wizard
Reverse Lookup Zone Name
A reverse lookup zone translates IP addresses into DNS names.
To identify the reverse lookup zone, type the network ID or the name of the zone.
Network ID:
10 .88 .0
The network ID is the portion of the IP addresses that belongs to this zone. Enter the network ID in its normal (not reversed) order.
If you use a zero in the network ID, it will appear in the zone name. For example, network ID 10 would create zone 10.in-addr.arpa, and network ID 10.0 would create zone 0.10.in-addr.arpa.
Reverse lookup zone name:
0.88.10.in-addr.arpa
< Back
Next >
Cancel

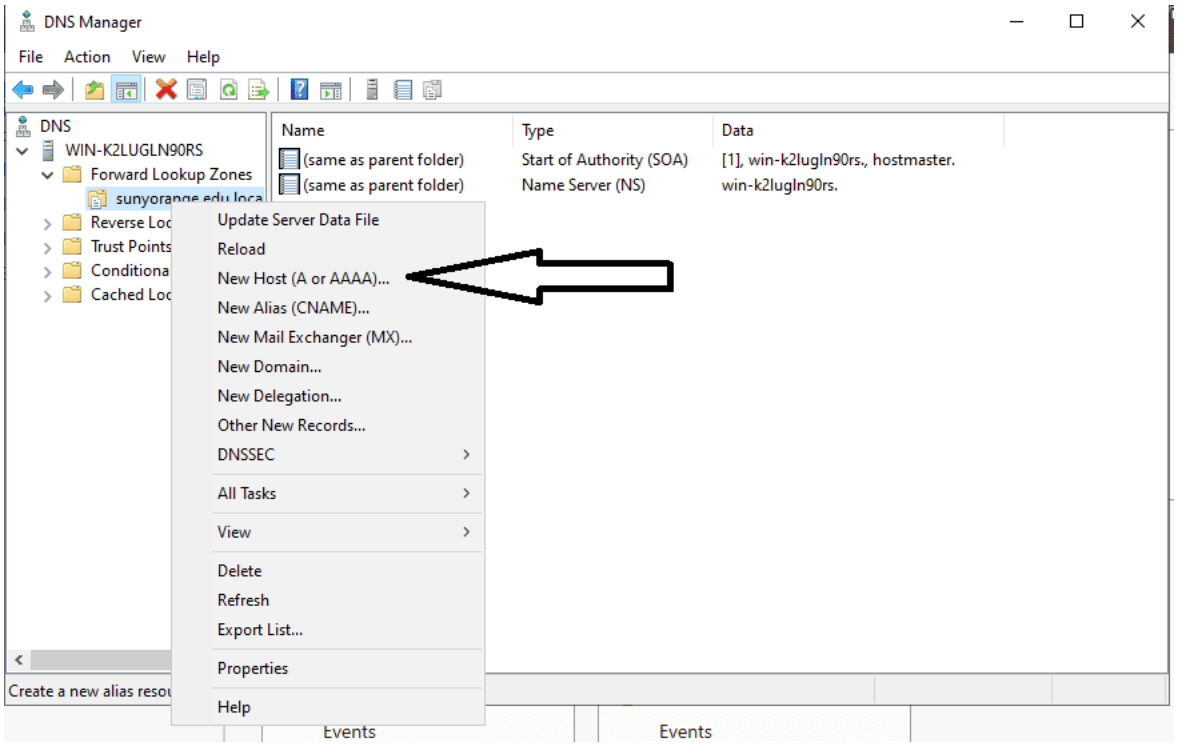

New Zone Wizard
Zone File
You can create a new zone file or use a file copied from another DNS server.
Do you want to create a new zone file or use an existing file that you have copied from another DNS server?
Create a new file with this file name:
0.88.10.in-addr.arpa.dns
Use this existing file:
To use this existing file, ensure that it has been copied to the folder %SystemRoot%\system32\dns on this server, and then click Next.
< Back
Next >
Cancel

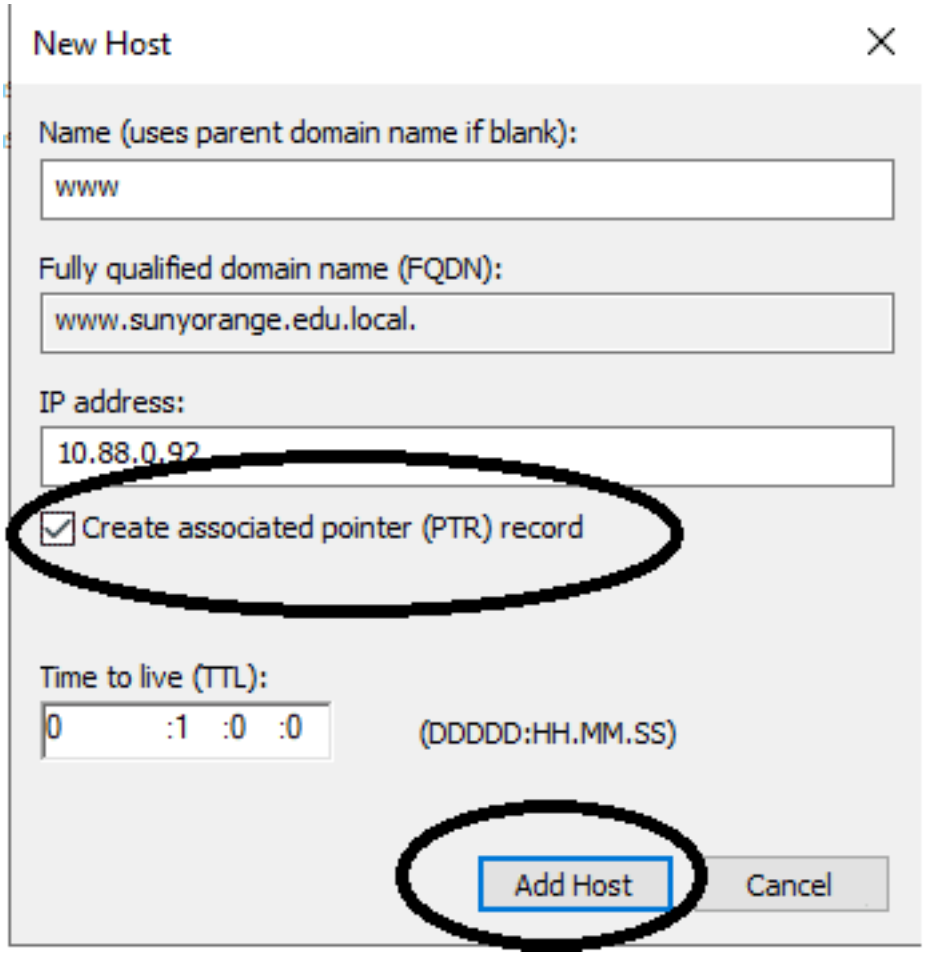

New Zone Wizard
Dynamic Update
You can specify that this DNS zone accepts secure, nonsecure, or no dynamic updates.
Dynamic updates enable DNS client computers to register and dynamically update their resource records with a DNS server whenever changes occur.
Select the type of dynamic updates you want to allow:
Allow only secure dynamic updates (recommended for Active Directory)
This option is available only for Active Directory-integrated zones.
Allow both nonsecure and secure dynamic updates
Dynamic updates of resource records are accepted from any client.
This option is a significant security vulnerability because updates can be accepted from untrusted sources.
Do not allow dynamic updates
Dynamic updates of resource records are not accepted by this zone. You must update these records manually.
< Back
Next >
Cancel





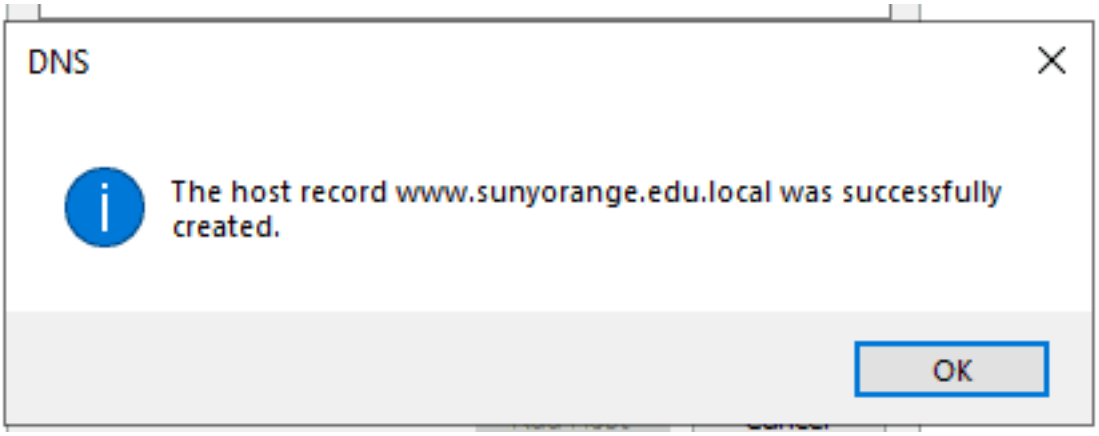
Add ANAME to our DNS Server

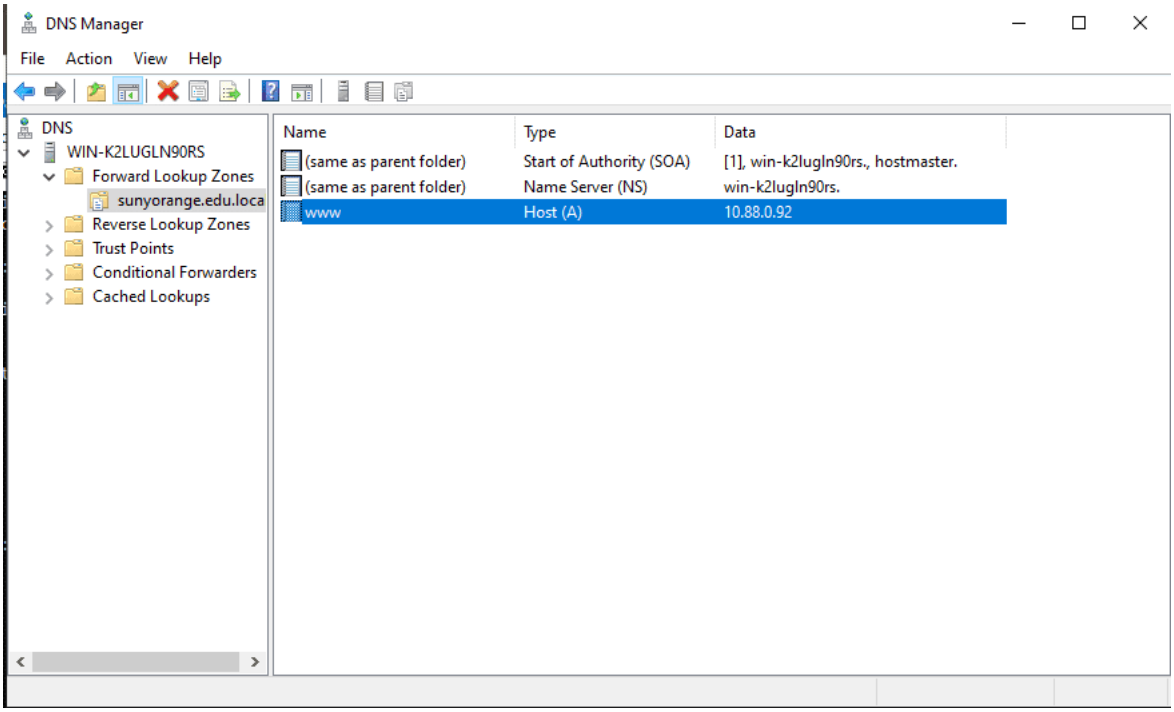


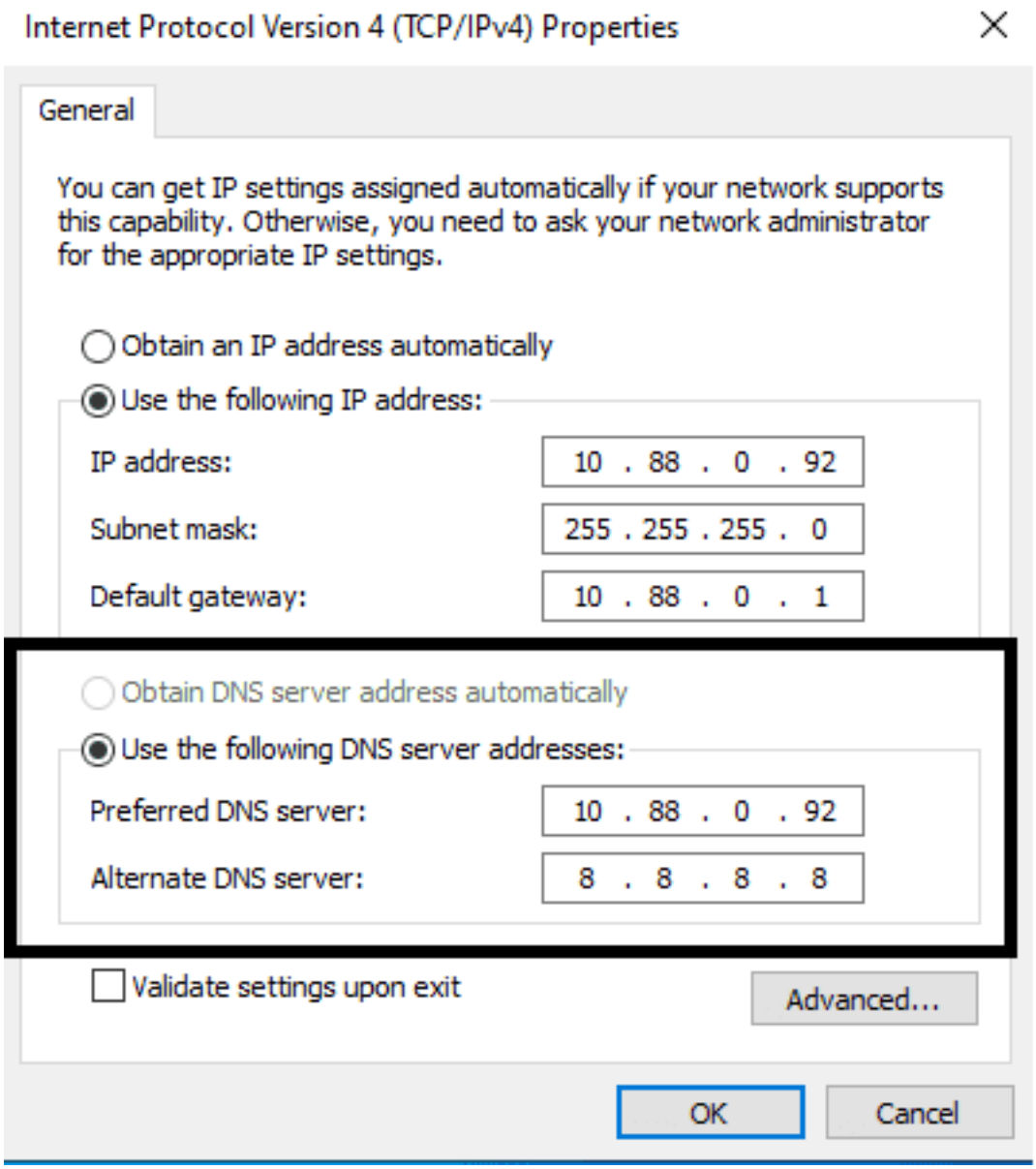


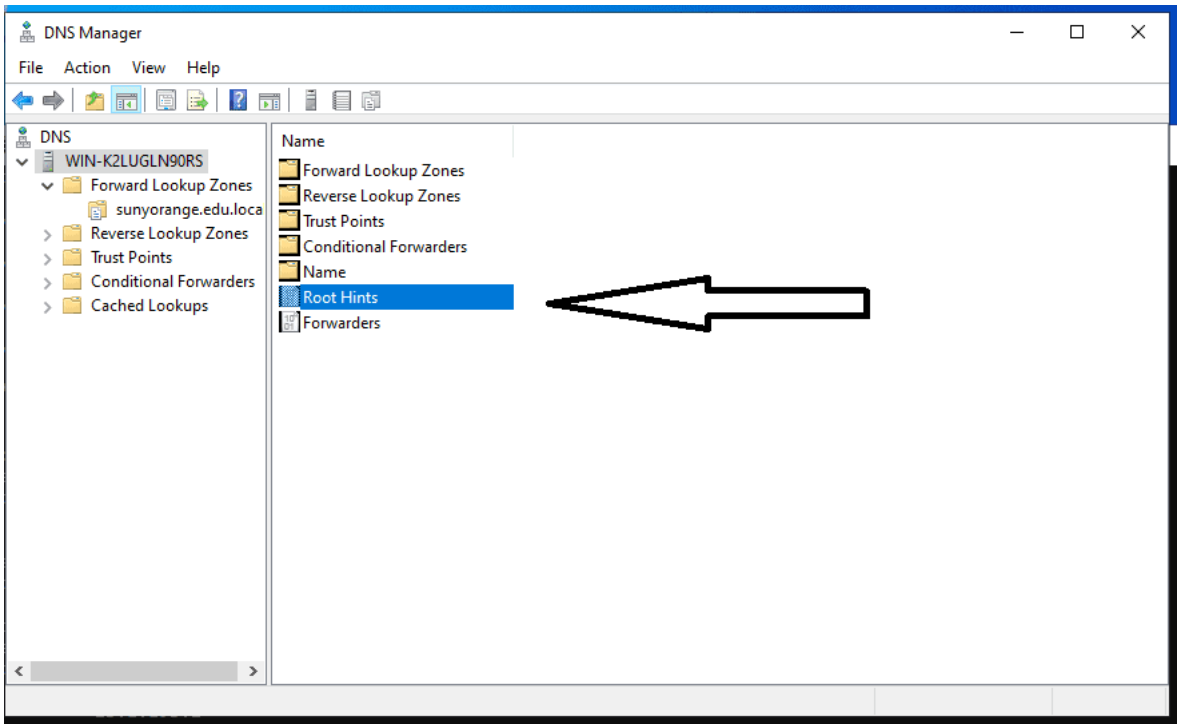
The PTR record is the data verifying that the IP address matches the domain name, and it's the reverse of the "A record," which provides the IP address associated with the domain.

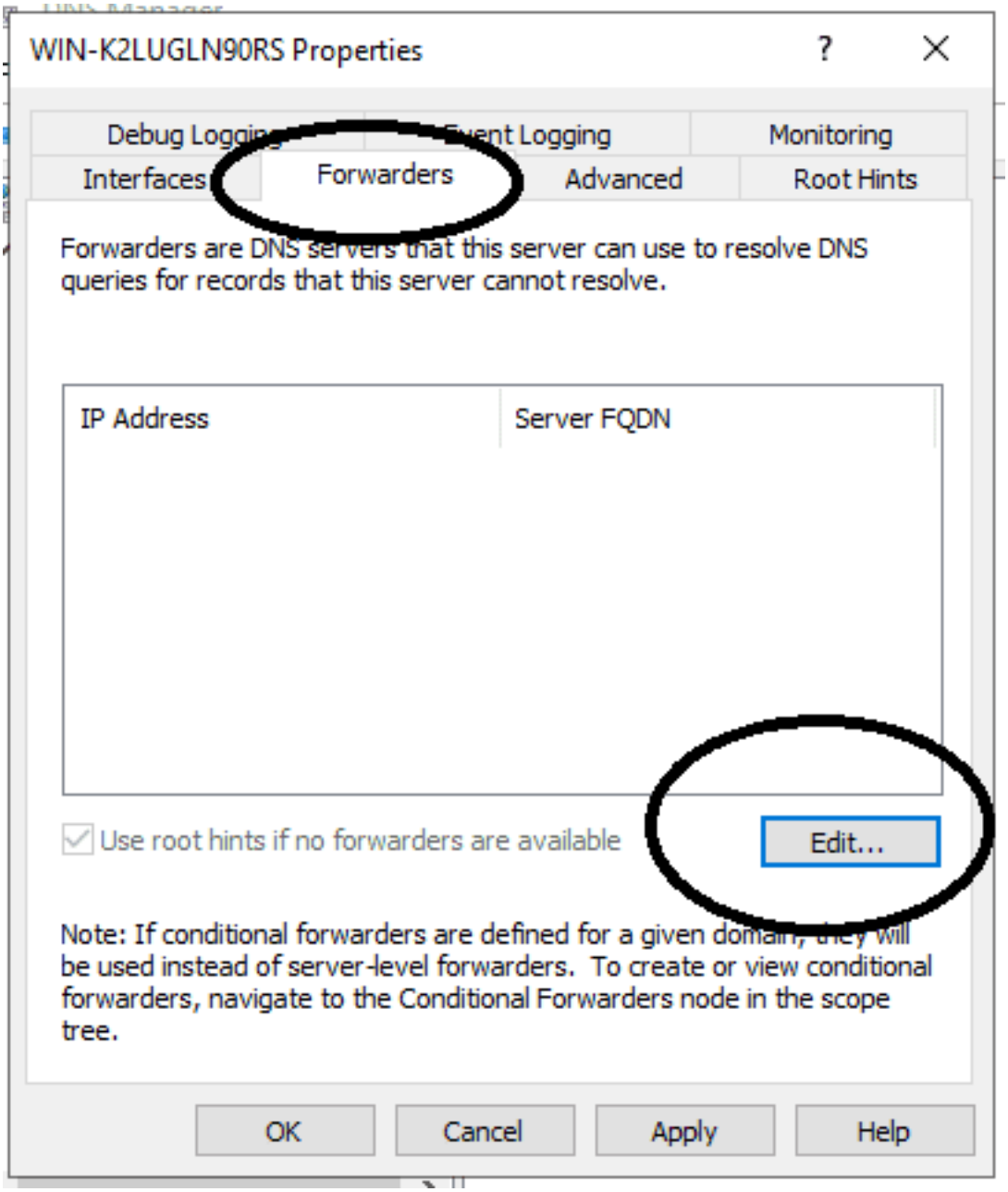
So, a PTR record is a normal DNS lookup record in reverse.

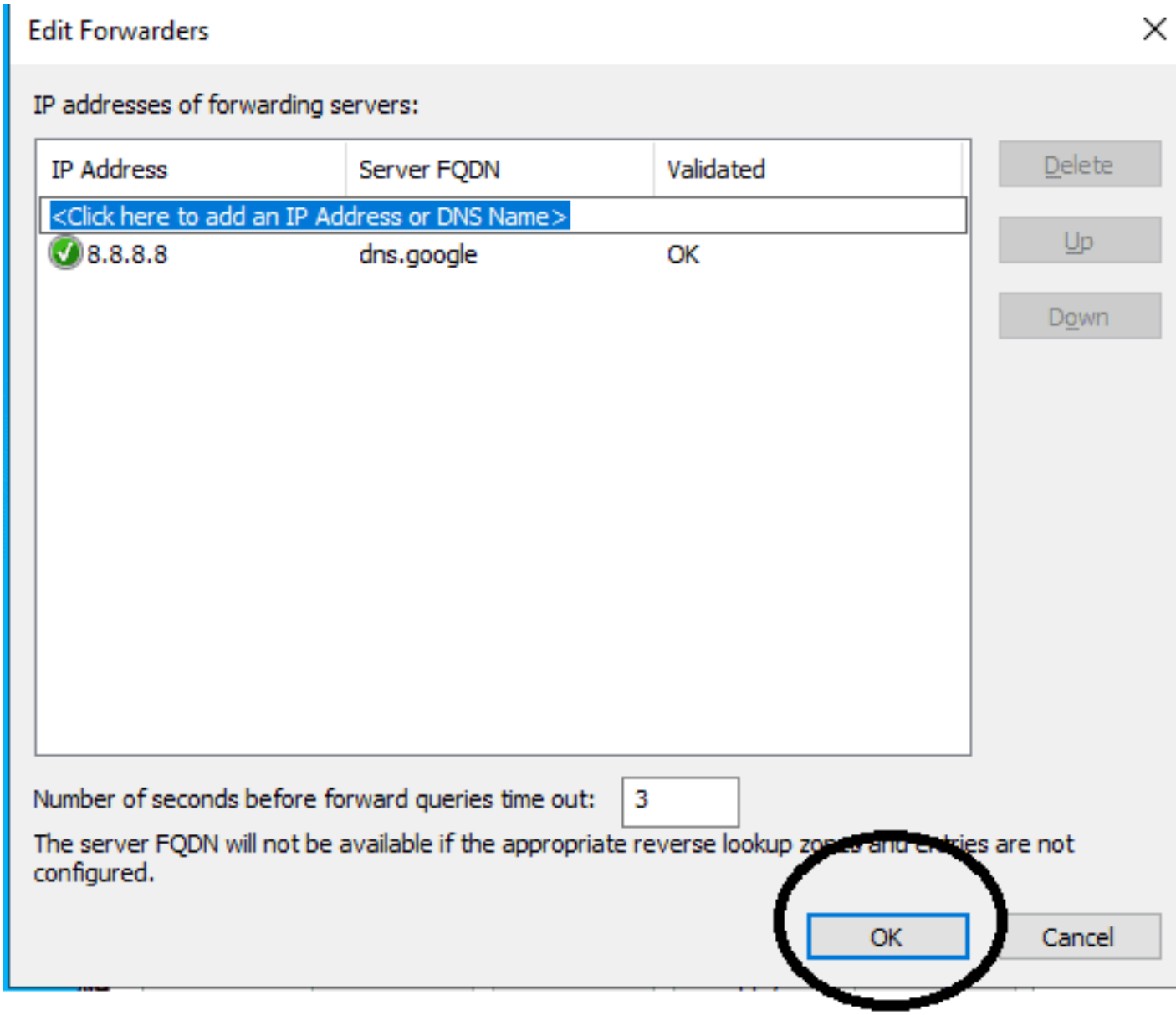


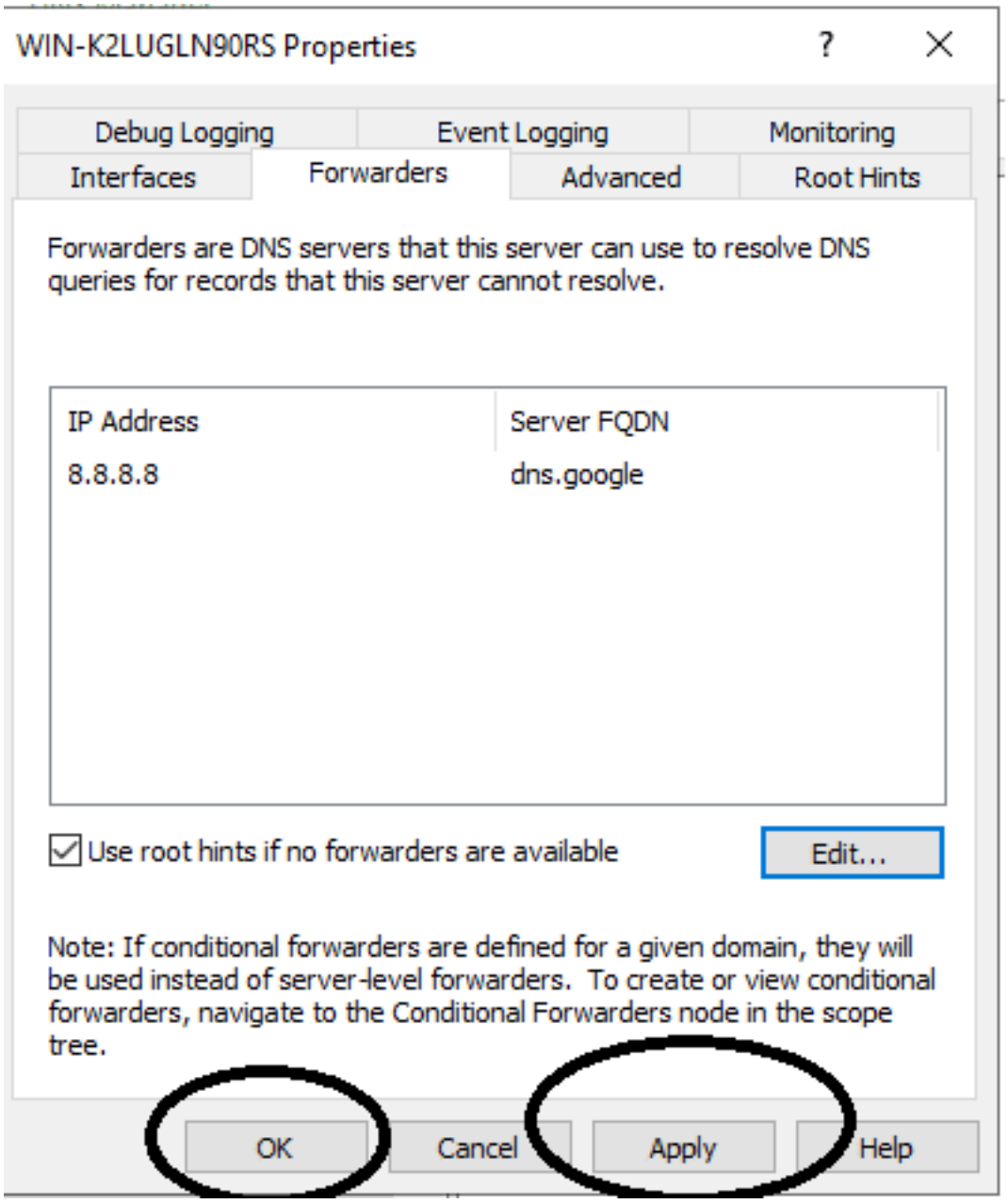


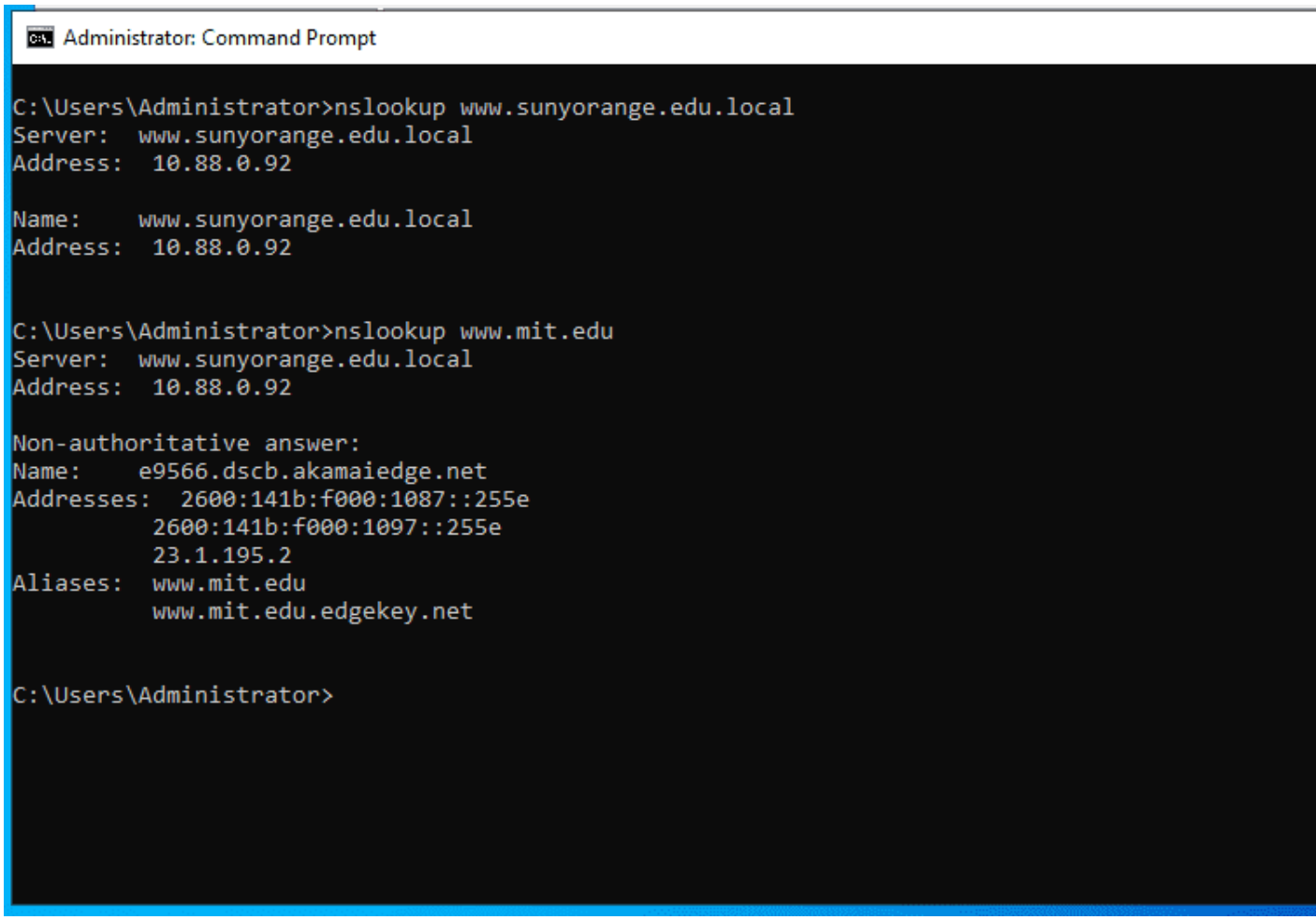

Internet Protocol Version 4 (TCP/IPv4) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 10 . 88 . 0 . 92
Subnet mask: 255 . 255 . 255 . 0
Default gateway: 10 . 88 . 0 . 1
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server: 10 . 88 . 0 . 92
Alternate DNS server: 8 . 8 . 8 . 8
Validate settings upon exit
Advanced...
OK
Cancel

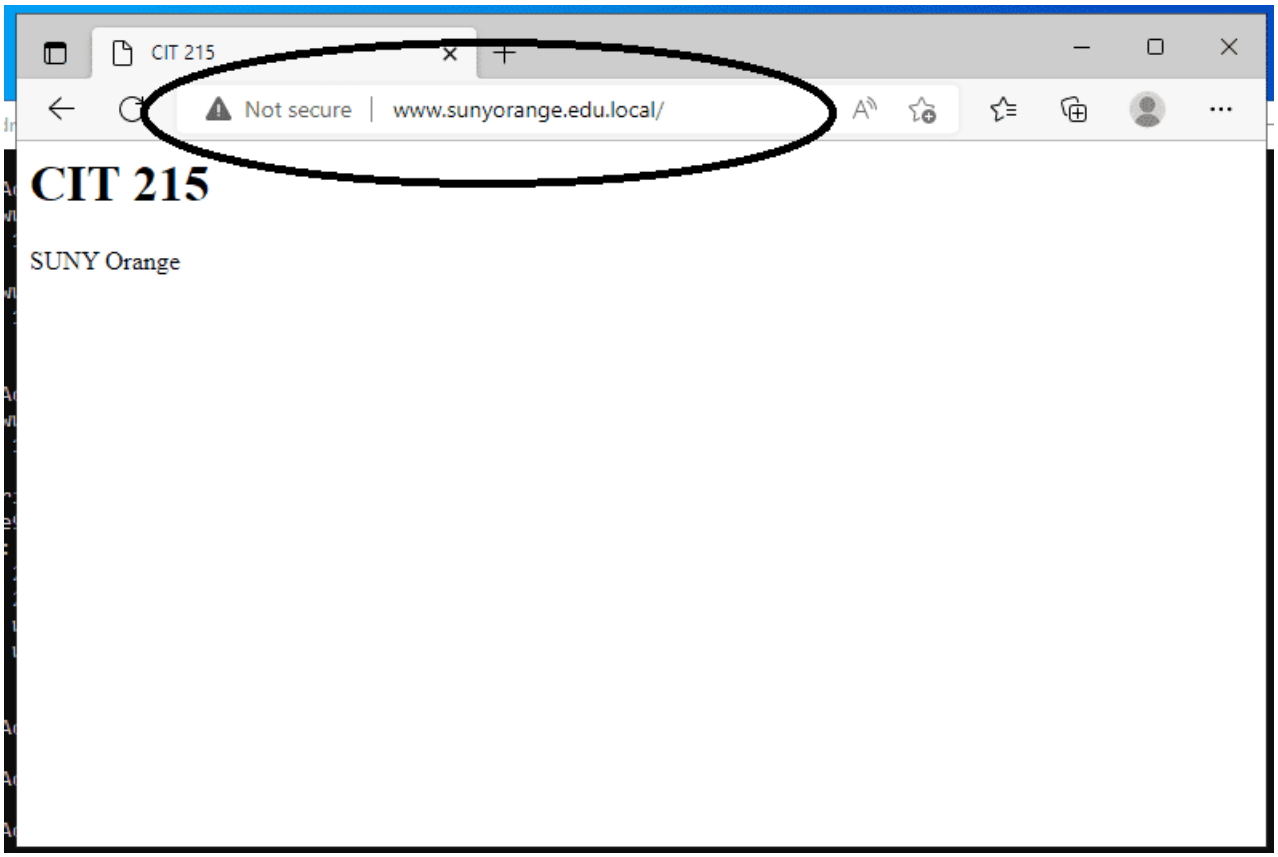

DNS Manager
File Action View Help
DNS
WIN-K2LUGLN90RS
Forward Lookup Zones
sunyorange.edu.loca
Reverse Lookup Zones
Trust Points
Conditional Forwarders
Cached Lookups
Name
Forward Lookup Zones
Reverse Lookup Zones
Trust Points
Conditional Forwarders
Name
Root Hints
Forwarders


WIN-K2LUGLN90RS Properties
Debug Logging
Event Logging
Monitoring
Interfaces
Forwarders
Advanced
Root Hints
Forwarders are DNS servers that this server can use to resolve DNS queries for records that this server cannot resolve.
IP Address
Server FQDN
Use root hints if no forwarders are available
Edit...
Note: If conditional forwarders are defined for a given domain, they will be used instead of server-level forwarders. To create or view conditional forwarders, navigate to the Conditional Forwarders node in the scope tree.
OK
Cancel
Apply
Help

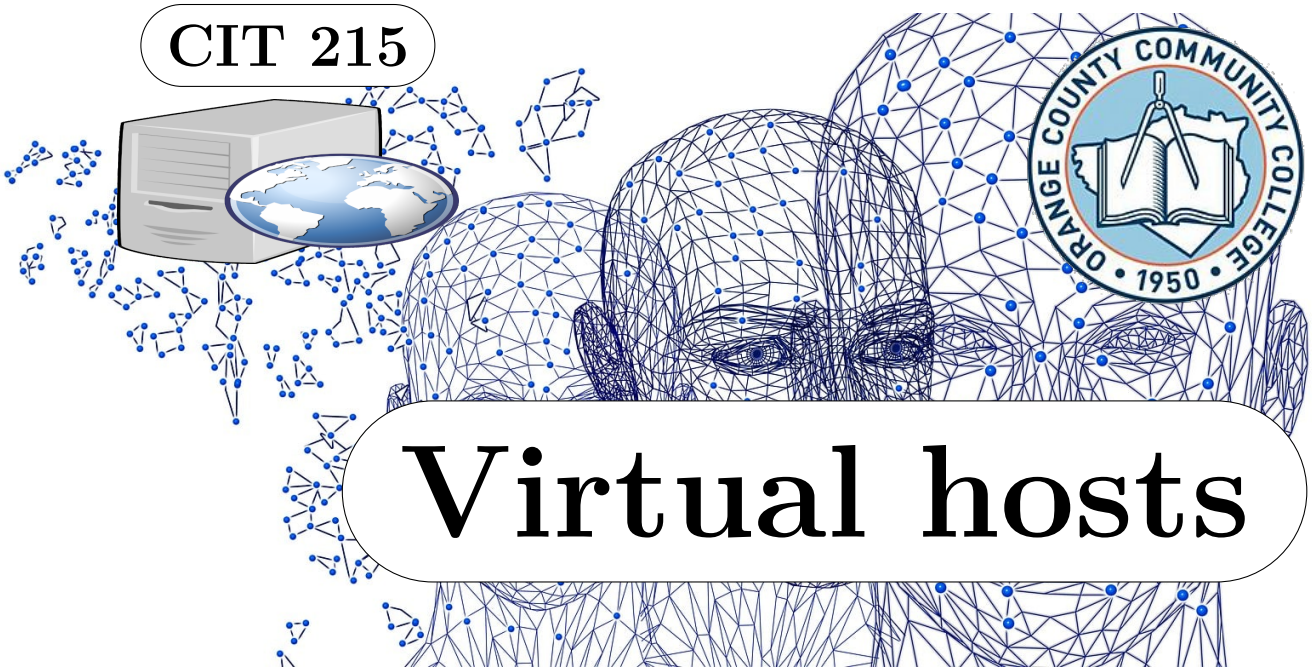

Edit Forwarders
IP addresses of forwarding servers:
IP Address
Server FQDN
Validated
<Click here to add an IP Address or DNS Name>
8.8.8.8
dns.google
OK
Delete
Up
Down
Number of seconds before forward queries time out:
3
The server FQDN will not be available if the appropriate reverse lookup zones and entries are not configured.
OK
Cancel


WIN-K2LUGLN90RS Properties
Debug Logging
Event Logging
Monitoring
Interfaces
Forwarders
Advanced
Root Hints
Forwarders are DNS servers that this server can use to resolve DNS queries for records that this server cannot resolve.
IP Address
Server FQDN
8.8.8.8
dns.google
Use root hints if no forwarders are available
Edit...
Note: If conditional forwarders are defined for a given domain, they will be used instead of server-level forwarders. To create or view conditional forwarders, navigate to the Conditional Forwarders node in the scope tree.
OK
Cancel
Apply
Help


Administrator: Command Prompt
C:\Users\Administrator>nslookup www.sunyorange.edu.local
Server: www.sunyorange.edu.local
Address: 10.88.0.92
Name: www.sunyorange.edu.local
Address: 10.88.0.92
C:\Users\Administrator>nslookup www.mit.edu
Server: www.sunyorange.edu.local
Address: 10.88.0.92
Non-authoritative answer:
Name: e9566.dscb.akamaiedge.net
Addresses: 2600:141b:f000:1087::255e
2600:141b:f000:1097::255e
23.1.195.2
Aliases: www.mit.edu
www.mit.edu.edgekey.net
C:\Users\Administrator>


CIT 215
Not secure
www.sunyorange.edu.local/
CIT 215
SUNY Orange

CIT 215



Virtual hosts

Virtual hosts allow more than one Web site on one machine or Web server. The servers are differentiated by their hostname. Visitors to the Web site are routed by hostname or IP address to the correct virtual host. Virtual hosting allows companies sharing one server to each have their own domain names. For example, **www.sunyorange.edu** and **www.sunyoccc.edu** can both be hosted on the same server.

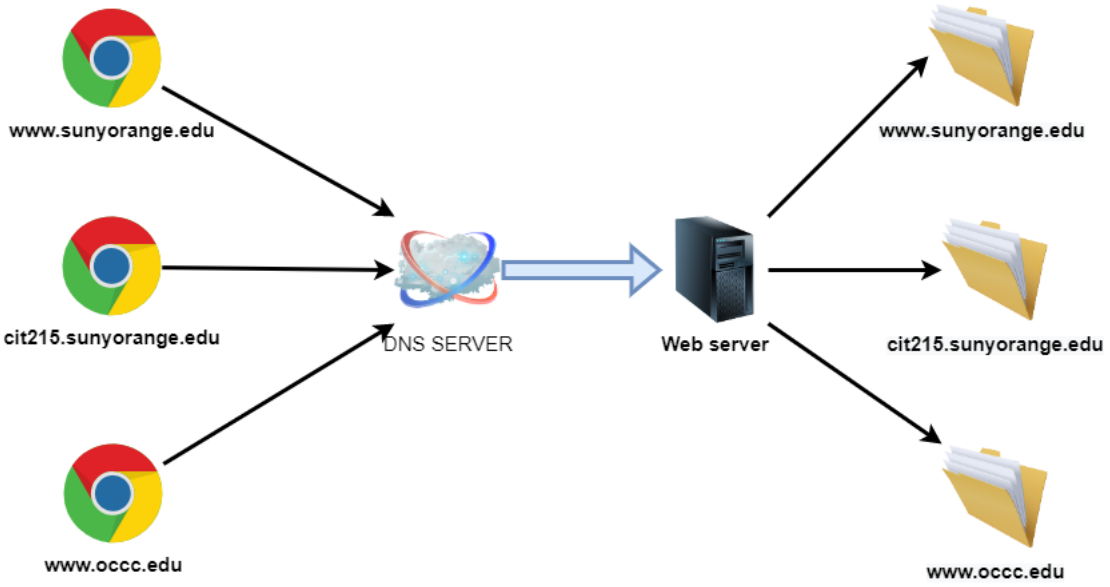
HTTP Server virtual host types

IP address-based virtual host

The IP address-based virtual host requires one IP address per Web site (hostname). This approach works very well, but requires a dedicated IP address for every virtual host.

Name-based virtual host

The name-based virtual host allows one IP address to host more than one Web site (hostname). This approach enables practically unlimited number of servers, ease of configuration and use, and requires no additional hardware or software. The main disadvantage to this approach is that the client must support HTTP 1.1 (or HTTP 1.0 with 1.1 extensions), which includes the hostname information inside the HTTP document requests.



5.0.1 Apache Virtual hosts

With name-based virtual hosting, the server relies on the web client to report the website name as part of the HTTP headers. Using this technique, many different hosts can share the same IP address.

Name-based virtual hosting is usually more straightforward, since you need only configure your DNS server to map each hostname to the correct IP address and then configure the Apache HTTP Server to recognize the different hostnames. Name-based virtual hosting also eases the demand for scarce IP addresses. Therefore you should use name-based virtual hosting unless you are using equipment that explicitly demands IP-based hosting.

Name-based virtual hosting builds off the IP-based virtual host selection algorithm, meaning that searches for the proper server name occur only between virtual hosts with the best IP-based address.

How the server selects the proper name-based virtual host

It is essential to recognize that the first step in name-based virtual host resolution is IP-based resolution. Name-based virtual host resolution only chooses the most appropriate name-based virtual host after narrowing down the candidates to the best IP-based match. Using a wildcard (*) for the IP address in all of the VirtualHost directives makes this IP-based mapping irrelevant.

When a request arrives, the server will find the best (most specific) matching **<VirtualHost>** argument based on the IP address and port used by the request. Suppose there is more than one virtual host containing this best-match address and port combination. In that case, Apache will further compare **ServerName** and **ServerAlias** directives to the server name present in the request.

If you omit the **ServerName** directive from any name-based virtual host, the server will default to a fully qualified domain name (FQDN) derived from the system hostname. This implicitly set server name can lead to counter-intuitive virtual host matching and is discouraged.

Using Name-based Virtual Hosts

The first step is to create a **<VirtualHost>** block for each different host that you would like to serve. Inside each **<VirtualHost>** block, you will need at minimum a **ServerName** directive to designate which host is served and a **DocumentRoot** directive to show where in the filesystem the content for that host lives.

Main host goes away
The global server configuration handles any request that doesn't match an existing **<VirtualHost>**, regardless of the hostname or ServerName.

When you add a name-based virtual host to an existing server, and the virtual host arguments match preexisting IP and port combinations, an explicit virtual host will now handle the requests. In this case, it's usually wise to create a default virtual host with a **ServerName** matching that of the base server. New domains on the same interface and port, but requiring separate configurations, can then be added as subsequent (non-default) virtual hosts.

For example, suppose that you are serving the domain **cit215.sunyorange.edu.local** and you wish to add the virtual host **www.sunyorange.edu.local**, which points at the same IP address. Then you simply add the following to **httpd.conf:**

Note

Creating virtual host configurations on your Apache server does not magically cause DNS entries to be created for those host names. You must have the names in DNS, resolving to your IP address, or nobody else will be able to see your web site. You can put entries in your **hosts file** for local testing, but that will work only from the machine with those hosts entries.

Virtual host config apache

```
# Ensure that Apache listens on port 80
Listen 80
<VirtualHost *:80>
    DocumentRoot "/www/main"
    ServerName www.sunyorange.edu.local

    # Other directives here
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/www/cit215"
    ServerName cit215.sunyorange.edu.local

    # Other directives here
</VirtualHost>
```

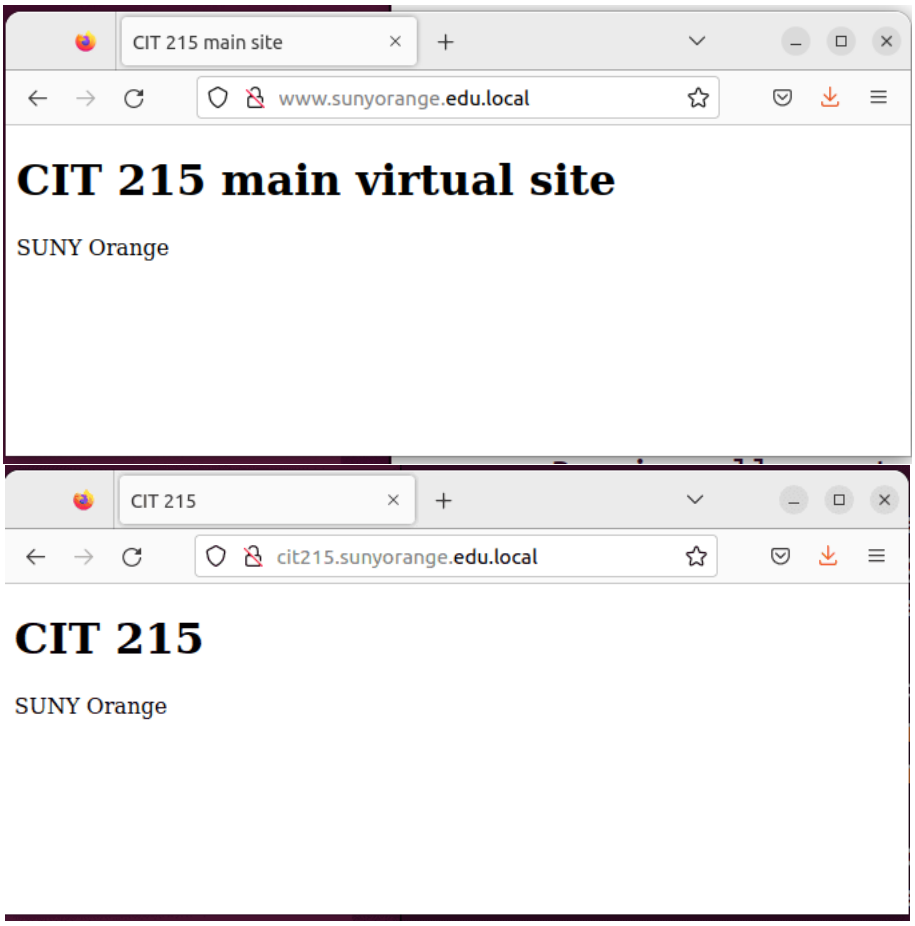
The asterisks match all addresses, so the main server serves no requests. Due to the fact that the virtual host with **ServerName www.sunyorange.edu.local** is first in the configuration file, it has the highest priority and can be seen as the default or primary server. That means that if a request is received that does not match one of the specified **ServerName** directives, it will be served by this first **<VirtualHost>**.

The above configuration is what you will want to use in almost all name-based virtual hosting situations. The only thing that this configuration will not work for, in fact, is when you are serving different content based on differing IP addresses or ports.

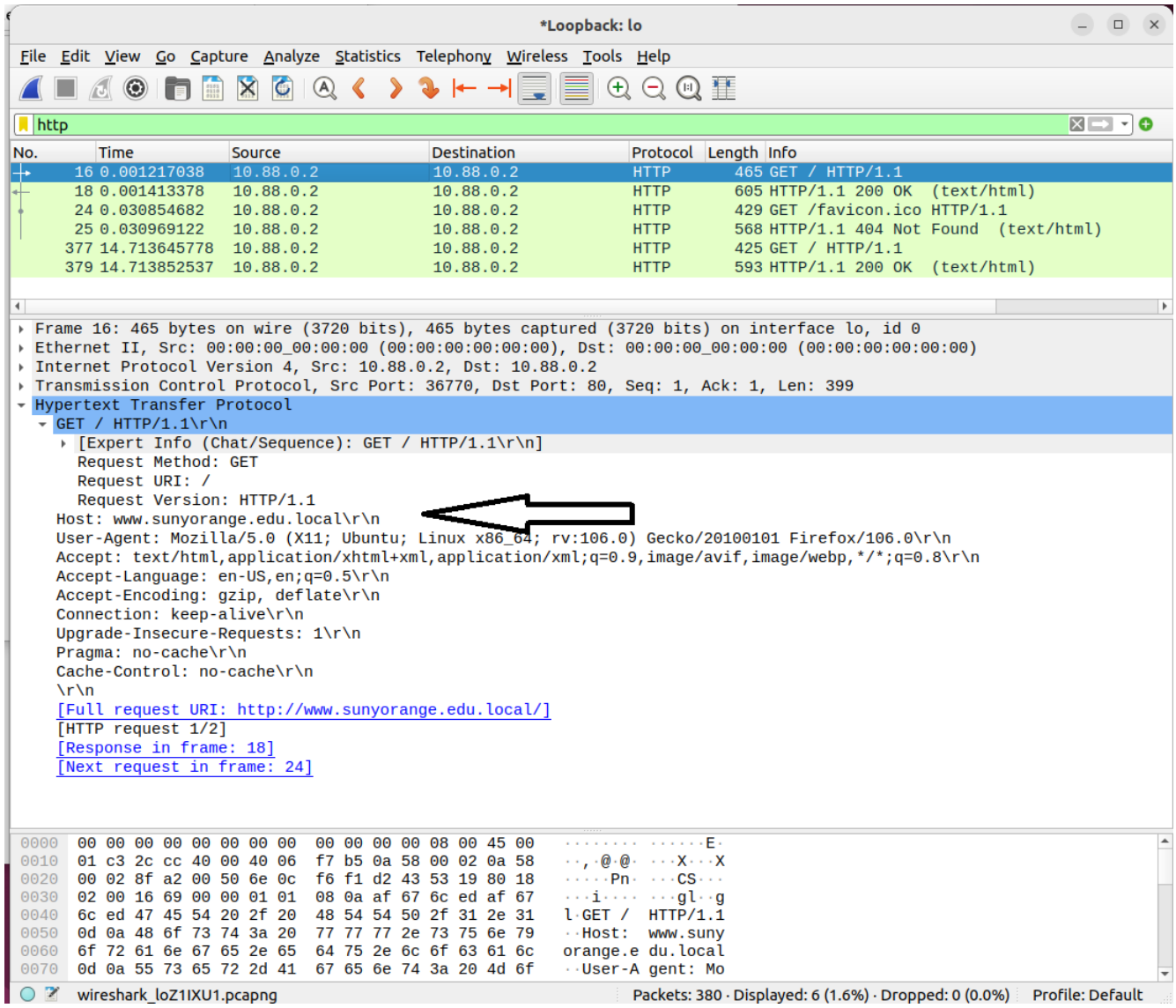
Name-based hosts on more than one IP address.

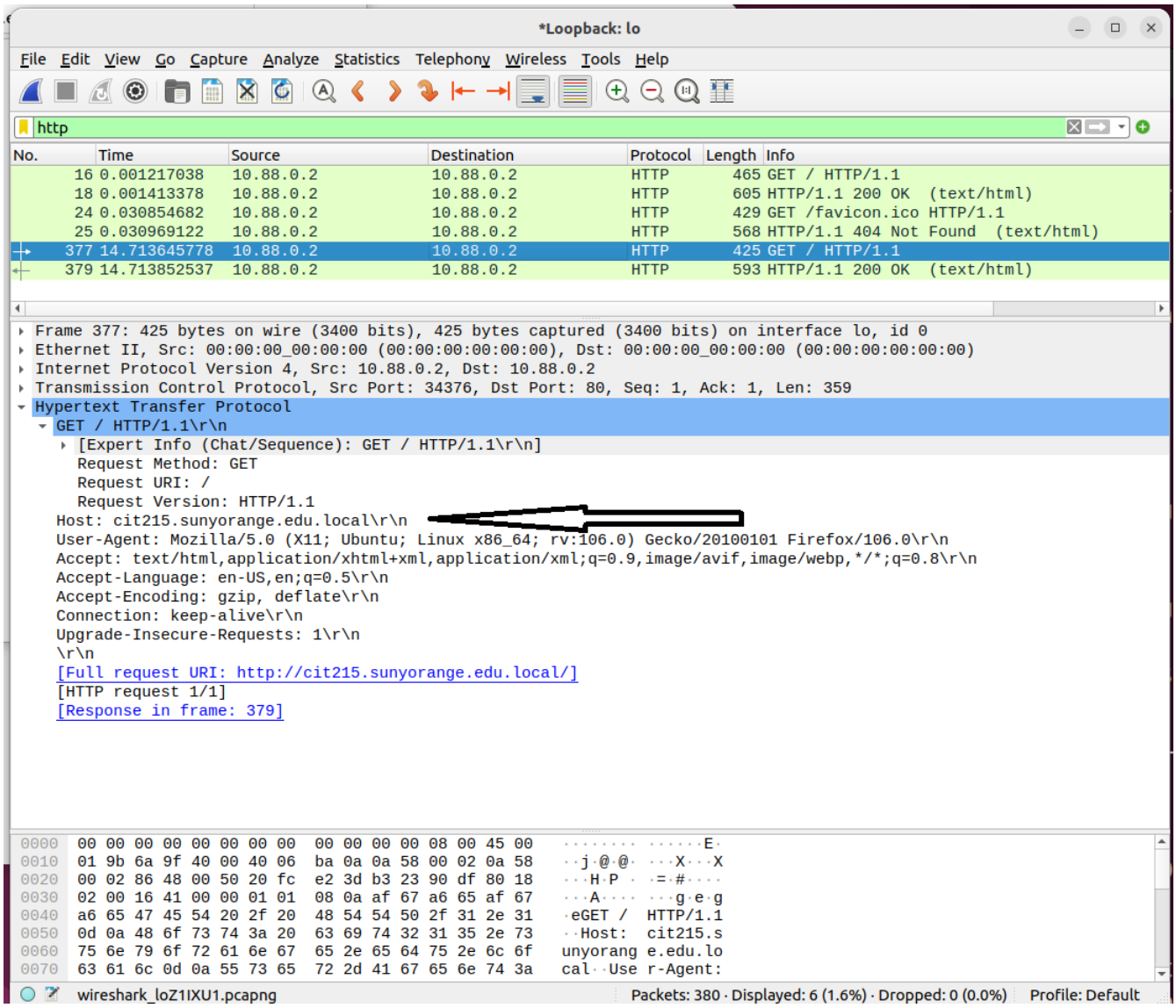
Virtual host multiple ip's config apache

```
#Listen 80

# This is the "main" server running on 10.88.0.2
#ServerName main.sunyorange.edu.local
#DocumentRoot "/www/mainserver"

<VirtualHost 10.88.0.2:80>
   DocumentRoot "/www/main"
   ServerName www.sunyorange.edu.local
   <Directory /www/main/>
      Options Indexes FollowSymLinks
      AllowOverride None
      Require all granted
   </Directory>


# Other directives here
</VirtualHost>

<VirtualHost 10.88.0.2:80>
   DocumentRoot "/www/cit215"
   ServerName cit215.sunyorange.edu.local
   <Directory /www/cit215/>
      Options Indexes FollowSymLinks
      AllowOverride None
      Require all granted
   </Directory>


# Other directives here
</VirtualHost>
```



We can see that each time we send a request to the server, the website name is included in the request, and apache is able to parse that and make a decision as to which directory and virtual site to send it to.


*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
No. Time Source Destination Protocol Length Info
16 0.001217038 10.88.0.2 10.88.0.2 HTTP 465 GET / HTTP/1.1
18 0.001413378 10.88.0.2 10.88.0.2 HTTP 605 HTTP/1.1 200 OK (text/html)
24 0.030854682 10.88.0.2 10.88.0.2 HTTP 429 GET /favicon.ico HTTP/1.1
25 0.030969122 10.88.0.2 10.88.0.2 HTTP 568 HTTP/1.1 404 Not Found (text/html)
377 14.713645778 10.88.0.2 10.88.0.2 HTTP 425 GET / HTTP/1.1
379 14.713852537 10.88.0.2 10.88.0.2 HTTP 593 HTTP/1.1 200 OK (text/html)
Frame 16: 465 bytes on wire (3720 bits), 465 bytes captured (3720 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 10.88.0.2, Dst: 10.88.0.2
Transmission Control Protocol, Src Port: 36770, Dst Port: 80, Seq: 1, Ack: 1, Len: 399
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
Host: www.sunyorange.edu.local\r\n
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0) Gecko/20100101 Firefox/106.0\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n
Accept-Language: en-US,en;q=0.5\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Upgrade-Insecure-Requests: 1\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://www.sunyorange.edu.local/]
[HTTP request 1/2]
[Response in frame: 18]
[Next request in frame: 24]
0000 00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00
0010 01 c3 2c cc 40 00 40 06 f7 b5 0a 58 00 02 0a 58
0020 00 02 8f a2 00 50 6e 0c f6 f1 d2 43 53 19 80 18
0030 02 00 16 69 00 00 01 01 08 0a af 67 6c ed af 67
0040 6c ed 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31
0050 0d 0a 48 6f 73 74 3a 20 77 77 77 2e 73 75 6e 79
0060 6f 72 61 6e 67 65 2e 65 64 75 2e 6c 6f 63 61 6c
0070 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 4d 6f
wireshark_loZ1IXU1.pcapng
Packets: 380 · Displayed: 6 (1.6%) · Dropped: 0 (0.0%)
Profile: Default



Serving the same content on different IP addresses (such as an internal and external address).

The server can be made to respond to internal and external requests with the same content, with just one **<VirtualHost>** section.

Virtual hostintranet extranet config apache

```
<VirtualHost 10.88.0.90  204.89.7.1>
     DocumentRoot "/www/main"
     ServerName www.sunyorange.edu.local
     ServerAlias server

     # Other directives here
</VirtualHost>
```

Running different sites on different ports.

Virtual host different port config apache

```
Listen 80
Listen 8080

<VirtualHost 10.88.0.90:80>
   ServerName www.sunyorange.edu.local
   DocumentRoot "/www/main-80"
</VirtualHost>

<VirtualHost 10.88.0.90:8080>
   ServerName www.sunyorange.edu.local
   DocumentRoot "/www/main-8080"
</VirtualHost>

<VirtualHost 10.88.0.90:80>
   ServerName cit215.sunyorange.edu.local
   DocumentRoot "/www/cit215-80"
</VirtualHost>

<VirtualHost 10.88.0.90:8080>
   ServerName cit215.sunyorange.edu.local
   DocumentRoot "/www/cit215-8080"
</VirtualHost>
```

Note

All of the above virtual hosts must have the proper directory stanzas configured

5.0.2 NGINX virtual hosts/server block

NGINX configuration options are known as "directives": these are arranged into groups, known interchangeably as **blocks** or **contexts**.

When a # appears before a line, these are comments and NGINX won't interpret them. Lines that contain directives should end with a semicolon (;). If not, NGINX will be unable to load the configuration properly and report an error.

/etc/nginx/nginx.conf

```
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    types_hash_max_size 2048;
    # server_tokens off;

    # server_names_hash_bucket_size 64;
    # server_name_in_redirect off;
```

```
24
25      include /etc/nginx/mime.types;
26      default_type application/octet-stream;
27
28      ##
29      # SSL Settings
30      ##
31
32      ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # ↙
          ↳  Dropping SSLv3, ref: POOD
33      LE
34      ssl_prefer_server_ciphers on;
35
36      ##
37      # Logging Settings
38      ##
39
40      access_log /var/log/nginx/access.log;
41      error_log /var/log/nginx/error.log;
42
43      ##
44      # Gzip Settings
45      ##
46
47      gzip on;
48
49      # gzip_vary on;
50      # gzip_proxied any;
51      # gzip_comp_level 6;
52      # gzip_buffers 16 8k;
53      # gzip_http_version 1.1;
54      # gzip_types text/plain text/css application/↙
          ↳  json application/javascript
55      text/xml application/xml application/xml+rss ↙
          ↳  text/javascript;
56
57      ##
58      # Virtual Host Configs
```

```
59      ##
60
61      include /etc/nginx/conf.d/*.conf;
62      include /etc/nginx/sites-enabled/*;
63  }
```

This file begins with four directives:

Global

- user Defines user and group credentials used by worker processes. If a group is omitted, a group whose name equals that of user is used.
- worker_processes Defines the number of worker processes.

  The optimal value depends on many factors, including (but not limited to) the number of CPU cores, the number of hard disk drives that store data, and the load pattern. When one is in doubt, setting it to the number of available CPU cores would be a good start (the value "auto" will try to autodetect it).
- pid Defines a file that will store the process ID of the main process.
- include Includes another file, or files matching the specified mask, into configuration. Included files should consist of syntactically correct directives and blocks.

These exist outside any particular context or block, and are said to be within the **main** context.

Additional directives are found within the events and http blocks, and these also exist within the main context.

What is the Http Block?

The http block includes directives for web traffic handling, which are generally known as **universal** . That's because they get passed on to each website configuration served by NGINX.

http block

```
1
2       http {
3
4           ##
5           # Basic Settings
6           ##
7
8           sendfile on;
9           tcp_nopush on;
10          types_hash_max_size 2048;
11          # server_tokens off;
12
13          # server_names_hash_bucket_size 64;
14          # server_name_in_redirect off;
15
16          include /etc/nginx/mime.types;
17          default_type application/octet-stream;
18
19          ##
20          # SSL Settings
21          ##
22
23          ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1↙
              ↪  .3; # Dropping SSLv3, ref: POOD
24          LE
25          ssl_prefer_server_ciphers on;
26
27          ##
28          # Logging Settings
29          ##
30
31          access_log /var/log/nginx/access.log;
32          error_log /var/log/nginx/error.log;
33
34          ##
35          # Gzip Settings
36          ##
```

```
37
38          gzip on;
39
40          # gzip_vary on;
41          # gzip_proxied any;
42          # gzip_comp_level 6;
43          # gzip_buffers 16 8k;
44          # gzip_http_version 1.1;
45          # gzip_types text/plain text/css ↙
               ↳  application/json application/↙
               ↳  javascript
46          text/xml application/xml application/xml+↙
               ↳  rss text/javascript;
47
48          ##
49          # Virtual Host Configs
50          ##
51
52          include /etc/nginx/conf.d/*.conf;
53          include /etc/nginx/sites-enabled/*;
54      }
```

All specific site configurations are imported from the following files and directories.

```
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
```

What are Server Blocks?

What is a Virtual Host? It is an Apache HTTP Server term; however, it is also commonly used by Nginx users. The proper term for Nginx is **server block**.

Examples:

Two Server Blocks, Serving Static Files

two server blocks serving static files

```
http {
    index index.html;

    server {
        server_name www.domain1.com;
        access_log logs/domain1.access.log main;

        root /var/www/domain1.com/htdocs;
    }

    server {
        server_name www.domain2.com;
        access_log  logs/domain2.access.log main;

        root /var/www/domain2.com/htdocs;
    }
}
```

https://www.nginx.com/resources/wiki/start/topics/examples/server_blocks/

Setting Up New Document Root Directories

Commands

```
occc@occc-VirtualBox:/etc/nginx$ sudo mkdir -p /webroot/cit215
[sudo] password for occc:
occc@occc-VirtualBox:/etc/nginx$ sudo mkdir -
p /webroot/main_site
occc@occc-VirtualBox:/etc/nginx$
```

Note

Depending on your needs, you might need to adjust the permissions or ownership of the folders again to allow specific access to the wwwdata(configured in nginx.conf) user. For instance, dynamic sites will often need this. Furthermore, the specific permissions and ownership requirements entirely depend on your configuration.

The permissions of our web roots should be correct already if you have not modified your **umask** value, but we can make sure by typing:

Commands

```
sudo chmod -R 755 /webroot/cit215
sudo chmod -R 755 /webroot/main_site
```

Creating Sample Pages for Each Site

Main

root@occc-VirtualBox:/webroot/cit215# more ../main_site/index.html

main site index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, ↙
     ↳  initial-scale=1.0">
6  <title>CIT 215</title>
7  </head>
8  <body>
9
10 <h1>MAIN site(www) for webserver</h1>
11 <p>SUNY Orange</p>
12
13 </body>
14 </html>
```

cit215

root@occc-VirtualBox:/webroot/cit215# pwd /webroot/cit215
root@occc-VirtualBox:/webroot/cit215# more index.html

cit215 site index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CIT 215</title>
</head>
<body>

<h1>CIT 215</h1>
<p>SUNY Orange</p>

</body>
</html>
```

Creating Server Block Files for Each Domain

Now that we have the content we wish to serve, we need to write the server blocks that will tell Nginx how to do this.

By default, Nginx contains one server block called **default** which we can use as a starting point for our own configurations. We will begin by designing our first domain's server block, which we will then copy over for our second domain and make the necessary modifications.

Creating the First Server Block File

Commands

```
sudo cp /etc/nginx/sites-available/default /etc/nginx/sites-available/cit215.edu
```

Ignoring the commented lines, the file will look somewhat like this:

first server block cit215

```
server {
    listen 80;
```

```
    listen [::]:80;

    root /webroot/cit215;

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

    server_name cit215.sunyorange.edu.local;

    location / {
        # First attempt to serve request as file, ↙
          ↳ then
        # as directory, then fall back to ↙
          ↳ displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```

First, we need to look at the listen directives. **Only one of our server blocks on the server can have the default_server option enabled**. This specifies which block should serve a request if the server_name requested does not match any of the available server blocks.
You can check that the default_server option is only enabled in a single active file by typing:

Commands

```
root@occc-VirtualBox:/etc/nginx/sites-available#
grep -R default_server /etc/nginx/sites-enabled/
/etc/nginx/sites-enabled/default: listen 80 default_server;
/etc/nginx/sites-enabled/default: listen [::]:80 default_server;
/etc/nginx/sites-enabled/default: # listen 443 ssl default_server;
/etc/nginx/sites-enabled/default: # listen [::]:443 ssl default_server
root@occc-VirtualBox:/etc/nginx/sites-available#
```

Only one file came up and that is the default file.

Creating the Second Server Block File

copy over the first block and name it as main.edu

main site server block

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

    root /webroot/main_site;

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

    server_name www.sunyorange.edu.local;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
}
```

Enabling your Server Blocks and Restart Nginx

Create symlinks for our sites.

Commands

```
sudo ln -s /etc/nginx/sites-available/main.edu  /etc/nginx/sites-
enabled/
sudo ln -s /etc/nginx/sites-available/cit215.edu  /etc/nginx/sites-
enabled/

root@occc-VirtualBox:/etc/nginx/sites-available# ls -
l ../sites-enabled/
total 0
lrwxrwxrwx 1 root root 37 Nov 19 11:59 cit215.edu -> /etc/nginx/sites-
available/cit215.edu
lrwxrwxrwx 1 root root 34 Oct 14 22:03 default -> /etc/nginx/sites-
available/default
lrwxrwxrwx 1 root root 35 Nov 19 11:59 main.edu -> /etc/nginx/sites-
available/main.edu
root@occc-VirtualBox:/etc/nginx/sites-available#
```

In order to avoid a possible hash bucket memory problem that can arise from adding additional server names, we will also adjust a single value within our /etc/nginx/nginx.conf file. Open the file and un-comment **server_names_hash_bucket_size**

server_names_hash_bucket_size

```
1  http {
2
3      ##
4      # Basic Settings
5      ##
6
7      sendfile on;
8      tcp_nopush on;
9      types_hash_max_size 2048;
10     # server_tokens off;
11
12     server_names_hash_bucket_size 64;
13     # server_name_in_redirect off;
```

Next, test to make sure that there are no syntax errors in any of your Nginx files:

Commands

```
root@occc-VirtualBox:/etc/nginx# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@occc-VirtualBox:/etc/nginx#
```

If no problems were found, restart Nginx to enable your changes:

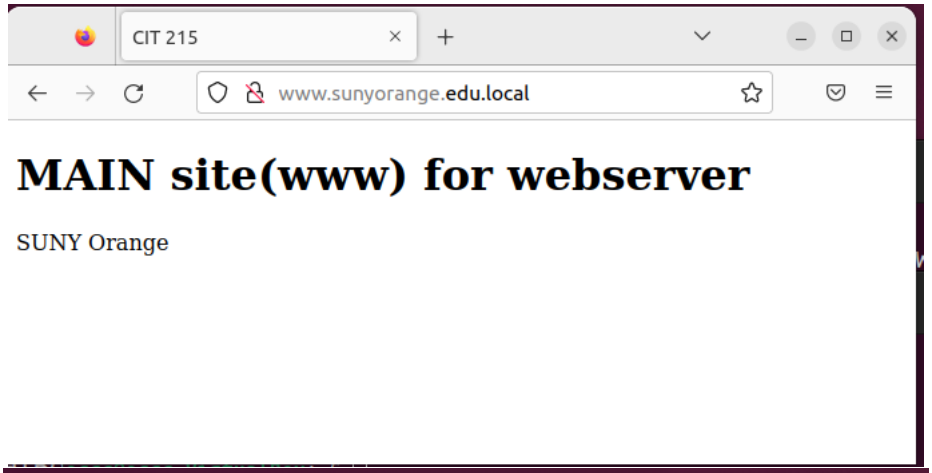
Commands

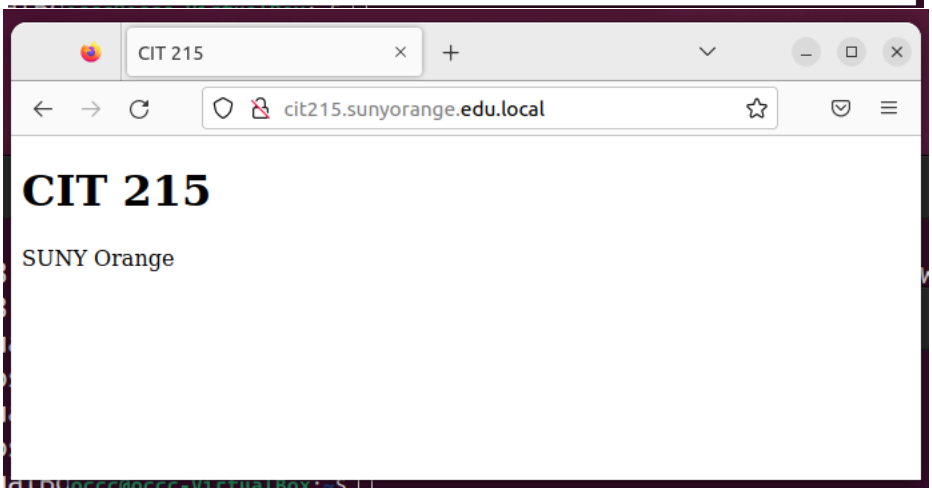
```
root@occc-VirtualBox:/etc/nginx# systemctl restart nginx
root@occc-VirtualBox:/etc/nginx# systemctl status nginx
  nginx.service - A high performance web server and a reverse proxy se
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor pre
Active: active (running) since Sat 2022-11-
19 12:11:08 EST; 11s ago
Docs: man:nginx(8)
Process: 57584 ExecStartPre=/usr/sbin/nginx -t -q -
g daemon on; master_process on; (code=exited, statu>
Process: 57585 ExecStart=/usr/sbin/nginx -
g daemon on; master_process on; (code=exited, status=0/SUCCE>
Main PID: 57586 (nginx)
Tasks: 2 (limit: 9457)
Memory: 2.6M
CPU: 11ms
CGroup: /system.slice/nginx.service
 57586 "nginx: master process /usr/sbin/nginx -
g daemon on; master_process on;"
 57587 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" ""

Nov 19 12:11:08 occc-VirtualBox systemd[1]: Starting A high performanc
Nov 19 12:11:08 occc-VirtualBox systemd[1]: Started A high performance
root@occc-VirtualBox:/etc/nginx#
```

Modifying Your Local Hosts File for Testing or make sure the domain is configured in DNS

hosts

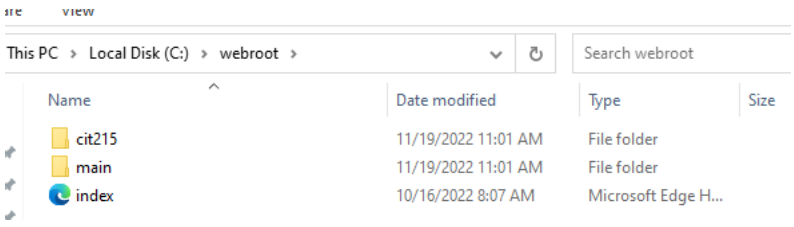
```
127.0.0.1 localhost
127.0.1.1 occc-VirtualBox
127.0.0.1 www.sunyorange.edu.local
127.0.0.1 cit215.sunyorange.edu.local

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



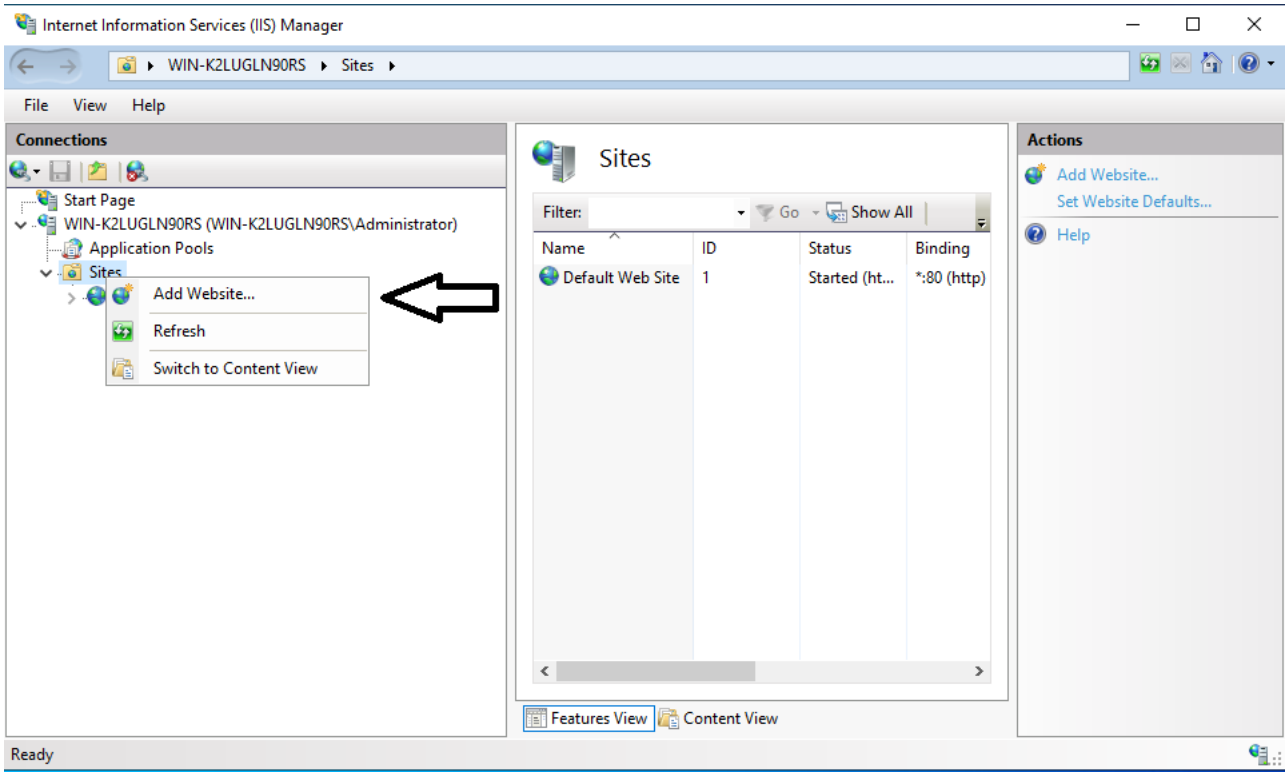


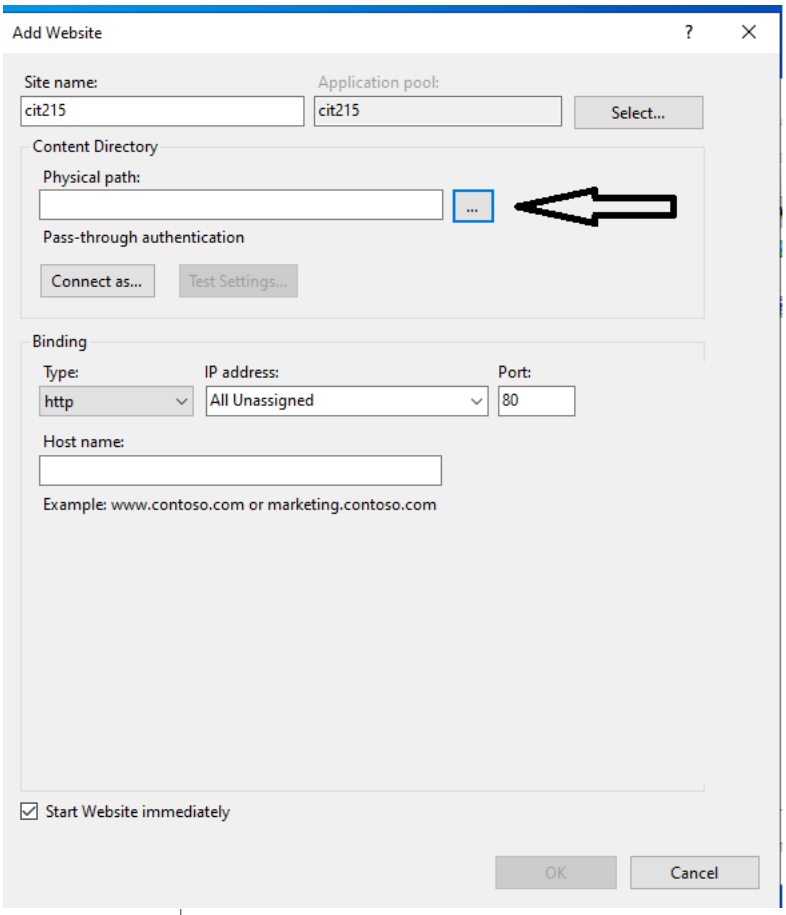
5.0.3 IIS virtual sites

Create directories for server roots

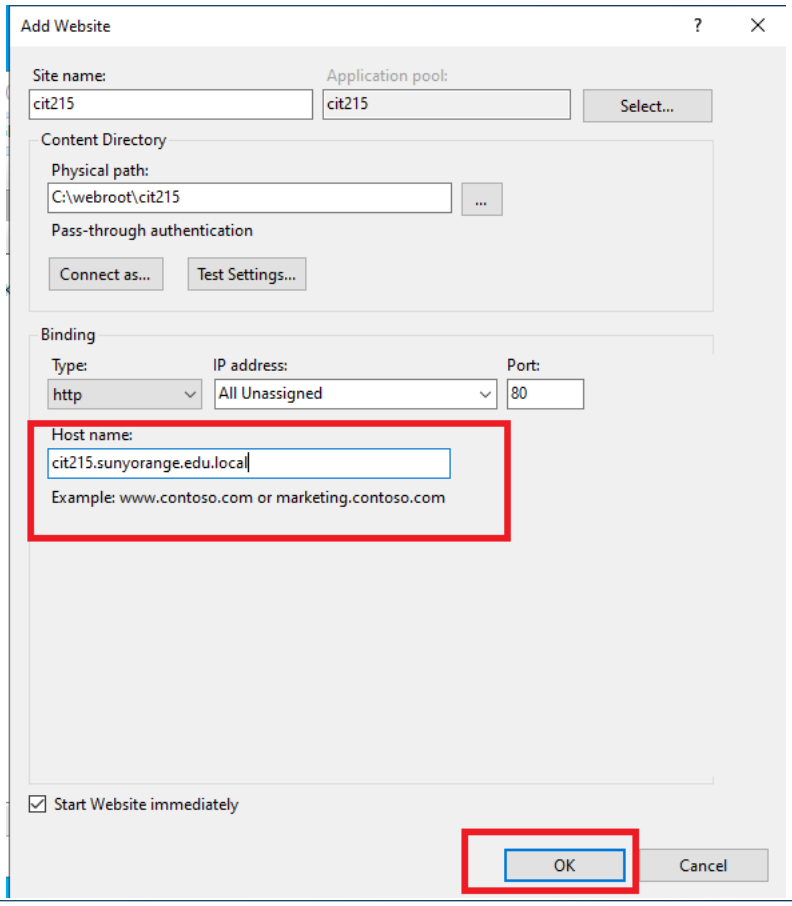


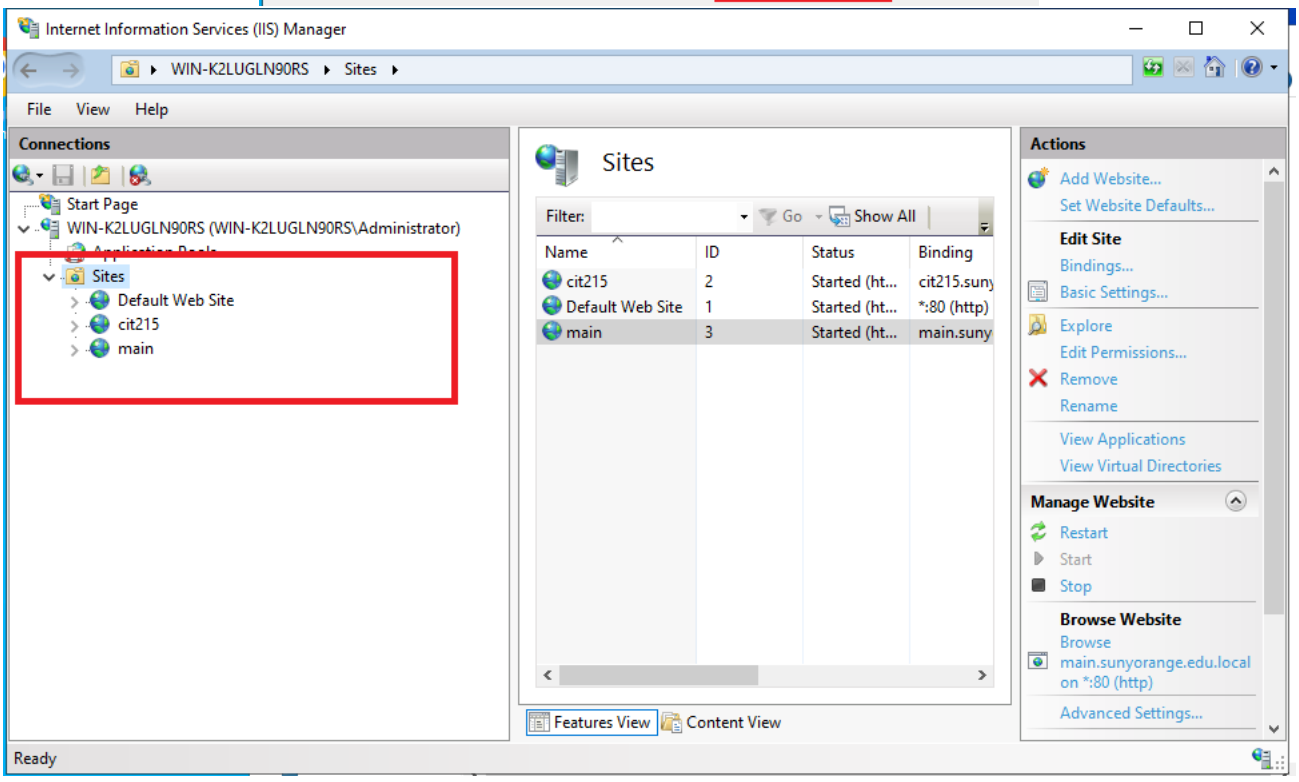
add new sites




Add Website
Site name:
cit215
Application pool:
cit215
Select...
Content Directory
Physical path:
...
Pass-through authentication
Connect as...
Test Settings...
Binding
Type:
http
IP address:
All Unassigned
Port:
80
Host name:
Example: www.contoso.com or marketing.contoso.com
Start Website immediately
OK
Cancel


Browse For Folder
Select a directory for the application.
PerfLogs
Program Files
Program Files (x86)
Users
webroot
cit215
main
Windows
CD Drive (D:) SSS_X64FREE_EN-US_DV9
Make New Folder
OK
Cancel


Add Website
Site name:
cit215
Application pool:
cit215
Select...
Content Directory
Physical path:
C:\webroot\cit215
...
Pass-through authentication
Connect as...
Test Settings...
Binding
Type:
http
IP address:
All Unassigned
Port:
80
Host name:
cit215.sunyorange.edu.local
Example: www.contoso.com or marketing.contoso.com
Start Website immediately
OK
Cancel


Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS ▸ Sites ▸
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-K2LUGLN90RS\Administrator)
Sites
Default Web Site
cit215
main
Sites
Filter:
Go
Show All
Name
ID
Status
Binding
cit215
2
Started (ht...
cit215.suny
Default Web Site
1
Started (ht...
*:80 (http)
main
3
Started (ht...
main.suny
Features View
Content View
Actions
Add Website...
Set Website Defaults...
Edit Site
Bindings...
Basic Settings...
Explore
Edit Permissions...
Remove
Rename
View Applications
View Virtual Directories
Manage Website
Restart
Start
Stop
Browse Website
Browse main.sunyorange.edu.local on *:80 (http)
Advanced Settings...
Ready

Website Log Monitoring

A log is a time-stamped, record of events over time. All network services, servers, applications, etc. produce logs, which can be grouped into three categories: plain text, structured, and binary logs. Most web servers use standardized log formats and have some categorization to help you easily sift through logs. For instance, Apache servers produce separate error logs and access logs and use plain text format for storing them. In many cases, we can customize these logs outputs and formats to capture the necessary data.

An error log gives information about a specific error with an error message. It also includes information such as the time of the error (stamped with the server's system time which should be synched to an NTP server), log level (warn, notice, info, etc.), and the client(who accessed) IP address. This information is crucial for a website administrator to keep track of errors and troubleshoot them before they cause major downtime. By correlating error logs against known threats, an web admin can block malicious IP addresses. Similarly, access logs can provide granular information about the requests processed by the web server. If the server returns the status code 200, it usually indicates a user was able to access the webpage. By looking at the 3XX, 4XX, and 5XX status codes, administrators can assess the health of their web servers and figure out if a bug is affecting their website.

Monitoring website logs allows you to get details about every request serviced by an HTTP/S connection. Plain text log entries are easy to understand, parse, and read, and they can be queried with a simple grep command. In most cases, the **tail -f** command provides significant headway in troubleshooting.

Quick detection of script errors, cron and batch jobs, failed processes and services, etc. Alerts for network outages and protocol failures Real-time awareness of network infrastructure problems Security and regulatory compliance Rich web analytics data for business decision-making

6.0.1 Log monitoring tools

There is a plethora of available web monitoring tools offering uptime monitoring and detailed visibility into the health and performance of a website using different metrics. Log management tools can help you supplement this monitoring with logs. There are two broad categories of log management solutions:

- **Open-source solutions:** the Elasticsearch, Logstash, and Kibana (ELK) stack, Graylog, LOGalyze, etc.
- **Commercial or paid solutions:** SolarWinds® Papertrail™, Logz.io, LogDNA, Sumo Logic, etc.

6.0.2 Legality of logs

European Union's General Data Protection Regulation (GDPR).

The default configuration of popular web servers including Apache Web Server and Nginx collect and store at least two of the following three types of logs:

- **Access logs**
- **Error logs (including processing-language logs like PHP)**
- **Security audit logs (e.g. ModSecurity)**

All of these logs contain personal information by default under the new regulation. IP addresses are specifically defined as personal data per Article 4, Point 1, and Recital 49. The logs can also contain usernames if your web service uses them as part of their URL structure, and even the referral information that's logged by default can contain personal information, e.g. unintended collection of

sensitive data; like being referred from a sensitive-subject website.

Children's Online Privacy Protection Act (COPPA)

COPPA is designed to protect the personal information (including log data) of children. Depending on whether or not your business is targeted to children, you will either need to:

- Include a statement in your Privacy Policy specifying that your services are not intended for children, or
- Comply with strict parental consent protocols before collecting any log data from children under the age of 13

California Online Privacy Protection Act (CalOPPA)

CalOPPA requires the following conditions to be met for anyone collecting personal data from California residents:

- Disclose the types of information you collect and who you share the information with.
- Provide an accessible method for users to review and edit their personal information.
- Notify users when the Privacy Policy is updated or changed.
- Post an effective date in the Privacy Policy.

Health Insurance Portability and Accountability Act (HIPPAA) and Health Information Technology for Economic and Clinical Health (HITECH)

You cannot log any personal identifiable information when dealing with these two laws. All Ip's must be anonymized as to not have any traceable information that can identify a person. No user names and such information can appear in any log that will identify a person.

https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification

Other laws: CMMC, DFARS, Sarbanes-Oxley,PCI DSS, FERPA, NIST 800-171

Some of these laws actually must have all information logged even if not in a manner that can be easily read but coded and be able to resolve the actual information afterwards.

6.0.3 Apache log config

On Ubuntu **apache2** logs are stored in `/var/log/apache2` directory.

File Descriptor Limits

When using a large number of Virtual Hosts, Apache may run out of available file descriptors (sometimes called file handles) if each Virtual Host specifies different log files. The total number of file descriptors used by Apache is one for each distinct error log file, one for every other log file directive, plus 10-20 for internal use. Unix operating systems limit the number of file descriptors that may be used by a process; the limit is typically 64, and may usually be increased up to a large hard-limit.

6.0.3.0 Error logs

LogLevel Directive

LogLevel adjusts the verbosity of the messages recorded in the error logs. The following levels are available, in order of decreasing significance:

Error log levels

Level	Description	Example
emerg	Emergencies - system is unusable.	”Child cannot open lock file. Exiting”
alert	Action must be taken immediately.	”getpwuid: couldn’t determine user name from uid”
crit	Critical Conditions.	”socket: Failed to get a socket, exiting child”
error	Error conditions.	”Premature end of script headers”
warn	Warning conditions.	”child process 1234 did not exit, sending another SIGHUP”
notice	Normal but significant condition.	”httpd: caught SIGBUS, attempting to dump core in ...”
info	Informational.	”Server seems busy, (you may need to increase StartServers, or Min/MaxSpareServers)...”
debug	Debug-level messages	”Opening config file ...”
trace1	Trace messages	”proxy: FTP: control connection complete”
trace2	Trace messages	”proxy: CONNECT: sending the CONNECT request to the remote proxy”
trace3	Trace messages	”openssl: Handshake: start”
trace4	Trace messages	”read from buffered SSL brigade, mode 0, 17 bytes”
trace5	Trace messages	”map lookup FAILED: map=rewritemap key=keyname”
trace6	Trace messages	”cache lookup FAILED, forcing new map lookup”
trace7	Trace messages, dumping large amounts of data	”| 0000: 02 23 44 30 13 40 ac 34 df 3d bf 9a 19 49 39 15 |”
trace8	Trace messages, dumping large amounts of data	”| 0000: 02 23 44 30 13 40 ac 34 df 3d bf 9a 19 49 39 15 |”

Using a level of at least crit is recommended.

For example:

LogLevel crit

It is also possible to change the level per directory:

loglevel per dir

```
1 LogLevel crit
2 <Directory "/webroot/htdocs/app">
3    LogLevel debug
4 </Directory>
```

We can also specify the name of the ErrorLog filename/directory location

ErrorLog /var/www/sunyorange.edu/logs/error.log

Error logs per Vhost

```
Listen 80
Listen 8080

<VirtualHost 10.88.0.90:80>
    ServerName www.sunyorange.edu.local
    DocumentRoot "/www/main-80"
    ErrorLog /var/www/80.sunyorange.edu/logs/error.log
    <Directory /www/main-80/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost 10.88.0.90:8080>
    ServerName www.sunyorange.edu.local
    DocumentRoot "/www/main-8080"
    ErrorLog /var/www/8080.sunyorange.edu/logs/error.log
    <Directory /www/main-8080/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
</VirtualHost>

<VirtualHost 10.88.0.90:80>
    ServerName cit215.sunyorange.edu.local
    DocumentRoot "/www/cit215-80"
    ErrorLog /var/www/80_cit215.sunyorange.edu/logs/error.log
    <Directory /www/cit215-80/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
    </Directory>
```

```
</VirtualHost>

<VirtualHost 10.88.0.90:8080>
    ServerName cit215.sunyorange.edu.local
    DocumentRoot "/www/cit215-8080"
    ErrorLog /var/www/8080_cit215.sunyorange.edu/
      ↳ logs/error.log
    <Directory /www/cit215-8080/>
       Options Indexes FollowSymLinks
       AllowOverride None
       Require all granted
    </Directory>
</VirtualHost>
```

6.0.3.0 Access logs

The access log contains information about requests coming into the web server. This information can include what pages people are viewing, the success status of requests, and how long the server took to respond. Here's an example of a typical access log entry:

Output

```
occc@occc-VirtualBox:/var/log/apache2$ tail access.log
127.0.0.1 - - [01/Nov/2022:19:01:57 -0400] "GET / HTTP/1.1" 200
527 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:106.0)
Gecko/20100101 Firefox/106.0"
occc@occc-VirtualBox:/var/log/apache2$
```

CustomLog Directive

Common Log Format (CLF)

```
"%h %l %u %t \"%r\" %>s %b"
```

Common Log Format with Virtual Host

```
"%v %h %l %u %t \"%r\" %>s %b"
```

NCSA extended/combined log format

```
"%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
```

Referer log format

```
"%{Referer}i -> %U"
```

Agent (Browser) log format

```
"%{User-agent}i"
```

Format String	Description
%%	The percent sign.
%a	Client IP address of the request (see the mod_remoteip module).
%{c}a	Underlying peer IP address of the connection (see the mod_remoteip module).
%A	Local IP-address.
%B	Size of response in bytes, excluding HTTP headers.
%b	Size of response in bytes, excluding HTTP headers. In CLF format, i.e. a '-' rather than a 0 when no bytes are sent.
%{VARNAME}C	The contents of cookie VARNAME in the request sent to the server. Only version 0 cookies are fully supported.
%D	The time taken to serve the request, in microseconds.
%{VARNAME}e	The contents of the environment variable VARNAME.
%f	Filename.
%h	Remote hostname. Will log the IP address if HostnameLookups is set to Off, which is the default. If it logs the hostname for only a few hosts, you probably have access control directives mentioning them by name. See the Require host documentation.
%{c}h	Like %h, but always reports on the hostname of the underlying TCP connection and not any modifications to the remote hostname by modules like mod_remoteip.
%H	The request protocol.
%{VARNAME}i	The contents of VARNAME: header line(s) in the request sent to the server. Changes made by other modules (e.g. mod_headers) affect this. If you're interested in what the request header was prior to when most modules would have modified it, use mod_setenvif to copy the header into an internal environment variable and log that value with the %{VARNAME}e described above.
%k	Number of keepalive requests handled on this connection. Interesting if KeepAlive is being used, so that, for example, a '1' means the first keepalive request after the initial one, '2' the second, etc...; otherwise this is always 0 (indicating the initial request).
%l	Remote logname (from identd, if supplied). This will return a dash unless mod_ident is present and IdentityCheck is set On.
%L	The request log ID from the error log (or '-' if nothing has been logged to the error log for this request). Look for the matching error log line to see what request caused what error.
%m	The request method.
%{VARNAME}n	The contents of note VARNAME from another module.
%{VARNAME}o	The contents of VARNAME: header line(s) in the reply.

Format String	Description
%p	The canonical port of the server serving the request.
%{format}p	The canonical port of the server serving the request, or the server's actual port, or the client's actual port. Valid formats are canonical, local, or remote.
%P	The process ID of the child that serviced the request.
%{format}P	The process ID or thread ID of the child that serviced the request. Valid formats are pid, tid, and hextid.
%q	The query string (prepended with a ? if a query string exists, otherwise an empty string).
%r	First line of request.
%R	The handler generating the response (if any).
%s	Status. For requests that have been internally redirected, this is the status of the original request. Use %>s for the final status.
%t	Time the request was received, in the format [18/Sep/2011:19:18:28 -0400]. The last number indicates the timezone offset from GMT
%{format}t	The time, in the form given by format, which should be in an extended strftime(3) format (potentially localized). If the format starts with begin: (default) the time is taken at the beginning of the request processing. If it starts with end: it is the time when the log entry gets written, close to the end of the request processing. In addition to the formats supported by strftime(3), the following format tokens are supported:
	sec number of seconds since the Epoch
	msec number of milliseconds since the Epoch
	usec number of microseconds since the Epoch
	msec_frac millisecond fraction
	usec_frac microsecond fraction
	These tokens can not be combined with each other or strftime(3) formatting in the same format string. You can use multiple %{format}t tokens instead.
%T	The time taken to serve the request, in seconds.
%{UNIT}T	The time taken to serve the request, in a time unit given by UNIT. Valid units are ms for milliseconds, us for microseconds, and s for seconds. Using s gives the same result as %T without any format; using us gives the same result as %D. Combining %T with a unit is available in 2.4.13 and later.
%u	Remote user if the request was authenticated. May be bogus if return status (%s) is 401 (unauthorized).
%U	The URL path requested, not including any query string.
%v	The canonical ServerName of the server serving the request.
%V	The server name according to the UseCanonicalName setting.
%X	Connection status when response is completed:
	"X=" Connection aborted before the response completed.
	"+=" Connection may be kept alive after the response is sent.
	"-= " Connection will be closed after the response is sent.
%I	Bytes received, including request and headers. Cannot be zero. You need to enable mod_logio to use this.
%O	Bytes sent, including headers. May be zero in rare cases such as when a request is aborted before a response is sent. You need to enable mod_logio to use this.
%S	Bytes transferred (received and sent), including request and headers, cannot be zero. This is the combination of %I and %O. You need to enable mod_logio to use this.
%{VARNAME}^ti	The contents of VARNAME: trailer line(s) in the request sent to the server.
%{VARNAME}^to	The contents of VARNAME: trailer line(s) in the response sent from the server.

/etc/apache2/apache2.conf

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-
Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

Custom Log files

```
1  Listen 80
2  <VirtualHost *:80>
3
4  ServerName www.sunyorange.edu.local
5
6  ServerAdmin webmaster@localhost
7  LogFormat "%h %l %u %t \"%r\" %>s %b" common
8  CustomLog /var/log/apache2/access_log_www common
9  DocumentRoot /webroot
10 <Directory /webroot/>
11    Options Indexes FollowSymLinks
12    AllowOverride None
13    Require all granted
14 </Directory>
15
16 ErrorLog /var/log/apache2/error.log
17
18
19 </VirtualHost>
```

output

```
root@occc-VirtualBox:/var/log/apache2# tail access_log_www
127.0.0.1 - - [02/Nov/2022:16:01:10 -0400] "GET / HTTP/1.1" 200 527
root@occc-VirtualBox:/var/log/apache2#
```

Each part of this log entry is described below.

127.0.0.1 (%h)

This is the IP address of the client (remote host) which made the request to the server.

- (%l)

The "hyphen" in the output indicates that the requested piece of information is not available. In this case, the information that is not available is the RFC

1413 identity of the client determined by identd on the clients machine.

- (%u)

This is the userid of the person requesting the document as determined by HTTP authentication.

[02/Nov/2022:16:01:10 -0400] (%t)

The time that the request was received. The format is:

[day/month/year:hour:minute:second zone]
day = 2*digit
month = 3*letter
year = 4*digit
hour = 2*digit
minute = 2*digit
second = 2*digit
zone = (‘+’ | ‘-’) 4*digit

It is possible to have the time displayed in another format by specifying %format in the log format string, where format is either as in strftime(3) from the C standard library, or one of the supported special tokens. For details see the mod_log_config format strings.

”GET / HTTP/1.1” (\”%r\”)

The request line from the client is given in double quotes. The request line contains a great deal of useful information. First, the method used by the client is GET. Second, the client requested the resource /, and third, the client used the protocol HTTP/1.1. It is also possible to log one or more parts of the request line independently. For example, the format string ”%m %U%q %H” will log the method, path, query-string, and protocol, resulting in exactly the same output as ”%r”.

200 (%>s)

This is the status code that the server sends back to the client. This information is very valuable, because it reveals whether the request resulted in a successful response (codes beginning in 2), a redirection (codes beginning in 3), an error caused by the client (codes beginning in 4), or an error in the server (codes beginning in 5). The full list of possible status codes can be found in the HTTP specification (RFC2616 section 10).

527 (%b)

The last part indicates the size of the object returned to the client, not including the response headers. If no content was returned to the client, this value will be ”-”. To log ”0” for no content, use %B instead.

mod_log_forensic

This module provides for forensic logging of client requests. Logging is done before and after processing a request, so the forensic log contains two log lines for each request. The forensic logger is very strict, which means:

The format is fixed. You cannot modify the logging format at runtime.
If it cannot write its data, the child process exits immediately and may dump core

Custom Log files

```
1  LoadModule log_forensic_module /usr/lib/apache2/↙
     ↳  modules/mod_log_forensic.so
2
3  <VirtualHost *:80>
4
5  ServerName www.sunyorange.edu.local
6
7  ServerAdmin webmaster@localhost
8  LogFormat "%h %l %u %t \"%r\" %>s %b" common
9  CustomLog /var/log/apache2/access_log_www common
10 ForensicLog  /var/log/apache2/↙
     ↳  access_log_www_forensic.log
11 DocumentRoot /webroot
12 <Directory /webroot/>
13    Options Indexes FollowSymLinks
14    AllowOverride None
15    Require all granted
16 </Directory>
17
18 ErrorLog /var/log/apache2/error.log
19
20 </VirtualHost>
```

output

```
root@occc-VirtualBox:/var/log/apache2# tail access_log_www_forensic.lo
+7120:6362d5be:0|GET / HTTP/1.1|Host:127.0.0.1|
User-Agent:Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv%3a106.0)
Gecko/20100101 Firefox/106.0|Accept:text/html,application/
xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;
q=0.8|Accept-Language:en-US,en;q=0.5|
Accept-Encoding:gzip,deflate,br|Connection:keep-alive|
Upgrade-Insecure-Requests:1|Sec-Fetch-Dest:document|
Sec-Fetch-Mode:navigate|Sec-Fetch-Site:none|Sec-Fetch-User:?1|
If-Modified-Since:Sun, 16 Oct 2022 23%3a18%3a52 GMT|
If-None-Match:"f0-5eb2f15082868-gzip"
-7120:6362d5be:0
root@occc-VirtualBox:/var/log/apache2#
```

One log or many logs?

We can have a different log per virtual server or have one log that combines them into one. Which one depends on whether the users have access to their own logs or is there only one admin that deals with the logs.
One combined log has advantages to keep it more secure and also to have logrotate.

Log rotate

Logrotate is a system utility that manages the automatic rotation and compression of log files. If log files were not rotated, compressed, and periodically pruned, they could eventually consume all available disk space on a system.

logrotate

```
/var/log/apache2/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 640 root adm
    sharedscripts
    prerotate
    if [ -d /etc/logrotate.d/httpd-prerotate ]; then
       run-parts /etc/logrotate.d/httpd-prerotate
    fi
    endscript
    postrotate
    if pgrep -f ^/usr/sbin/apache2 > /dev/null; then
       invoke-rc.d apache2 reload 2>&1 | logger -t apache2.logrotate
    fi
    endscript
}
```

From the above logrotate we see that the logs are defined to be in /var/log/apache2 directory and they also have an extension of **.log**. If we change the log names we will have to change the logrotate script as well.

6.0.4 Apache logs to rsyslog

rsyslog is the default logging program in Debian and Red Hat. It is an extension of the original syslog protocol, with additional features such as flexible configuration, rich filtering capabilities and content-based filtering. Just like syslogd, the **rsyslogd daemon** can be used to gather log messages from programs and servers and direct those messages to local log files, devices, or remote logging hosts.

6.0.5 NGINX logging

https://docs.nginx.com/nginx/admin-guide/monitoring/logging/

NGINX Log Severity Levels

NGINX supports a wide range of severity levels to make it easy to log the

information you care about. Each of these levels can be used with the error_log directive to set the minimum level at which messages are logged. Here are the supported levels in lowest to highest order, along with a guide on how they're used:

- **Debug** Debugging messages that are not useful most of the time.
- **Info** Informational messages that might be good to know.
- **Notice** Something normal but significant happened and it should be noted.
- **Warn** Something unexpected happened, however it's not a cause for concern
- **Error** Something failed.
- **Crit** A critical condition occurred.
- **Alert** Immediate action is required.
- **Emerg** The system is unusable.

Configuring error_log

The error_log directive sets up error logging to file or stderr, or syslog by specifying minimal severity level of error messages to be logged. The syntax of error_log directive is:

```
error_log log_file log_level;
```

```
http {
...
error_log /var/log/nginx/error_log crit;
...
}
```

It is also possible to record error logs for all the virtual host separately by overriding the error_log directive in the server context. The following example exactly does that by overriding error_log directive in the server context.

```
http {
...
...
error_log /var/log/nginx/error_log;
server {
listen 80;
server_name domain1.com;
error_log /var/log/nginx/domain1.error_log warn;
...
}
server {
listen 80;
server_name domain2.com;
error_log /var/log/nginx/domain2.error_log debug;
...
}
}
```

All the examples described above records the log events to a file. You can also configure the error_log directive for sending the log events to a syslog server. The following error_log directive sends the error logs to syslog server with an IP address of 10.88.0.200 in debug format.

```
error_log syslog:server=10.88.0.200 debug;
```

Setting Up the Access Log

NGINX writes information about client requests in the access log right after the request is processed. By default, the access log is located at logs/access.log, and the information is written to the log in the predefined combined format. To override the default setting, use the log_format directive to change the format of logged messages, as well as the access_log directive to specify the location of the log and its format. The log format is defined using variables.

The following examples define the log format that extends the predefined combined format with the value indicating the ratio of gzip compression of the response. The format is then applied to a virtual server that enables compression.

basic nginx conf file

```
http {
    log_format compression '$remote_addr - $remote_user [$time_local] '
    '"$request" $status $body_bytes_sent '
    '"$http_referer" "$http_user_agent" "$gzip_ratio"';

    server {
        gzip on;
        access_log /spool/logs/nginx-access.log compression;
        ...
    }
}
```

Logging to Syslog

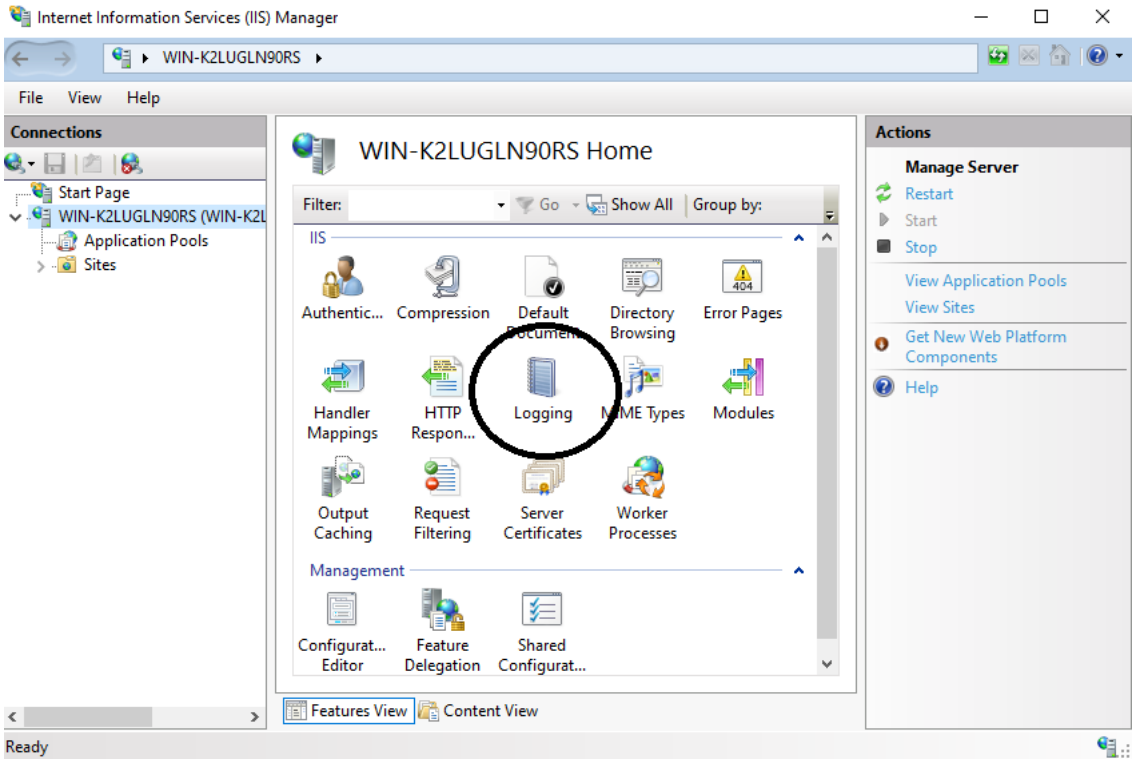
error_log syslog:server=unix:/var/log/nginx.sock debug;
access_log syslog:server=[2001:db8::1]:1234,facility=local7,tag=nginx,severity=info;

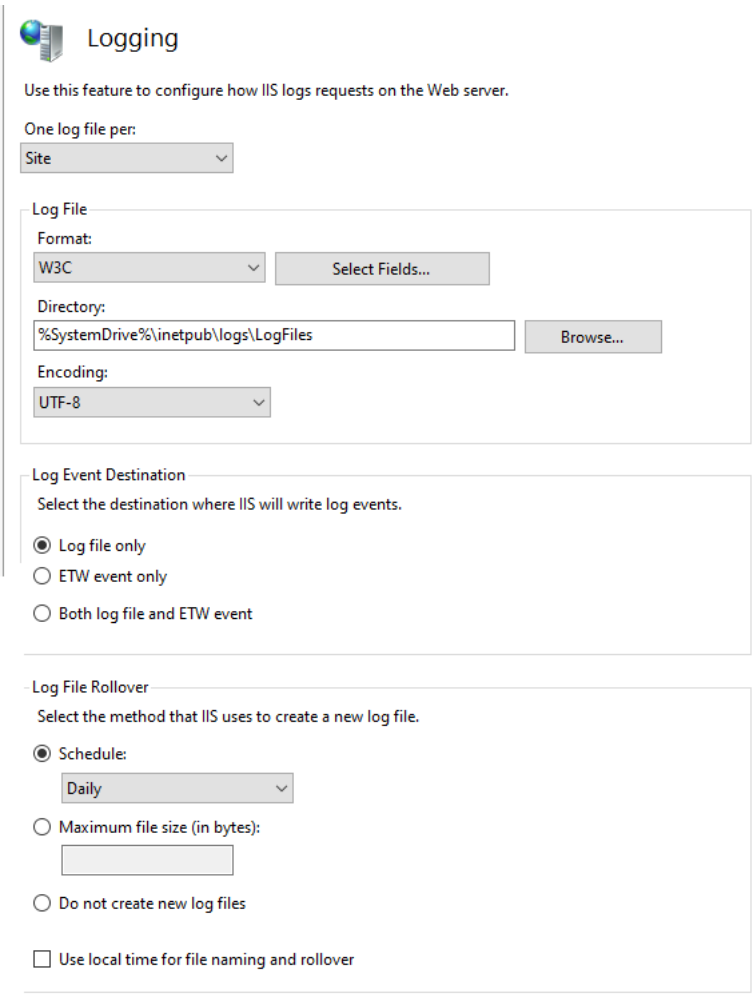
Nginx logrotate

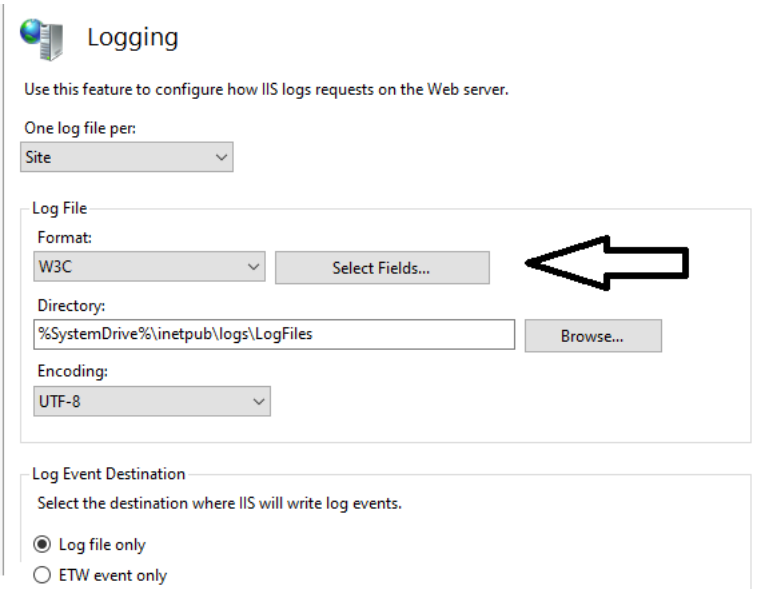
logrotate

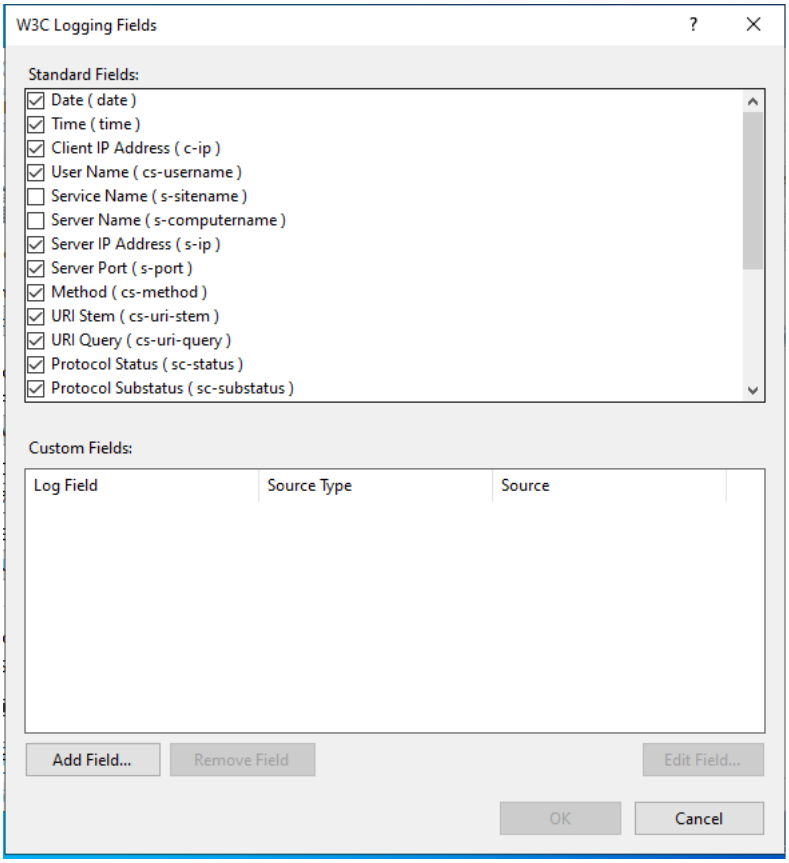
```
/var/log/nginx/*.log {
    daily
    missingok
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 www-data adm
    sharedscripts
    prerotate
    if [ -d /etc/logrotate.d/httpd-prerotate ]; then \
    run-parts /etc/logrotate.d/httpd-prerotate; \
    fi \
    endscript
    postrotate
    invoke-rc.d nginx rotate >/dev/null 2>&1
    endscript
}
```

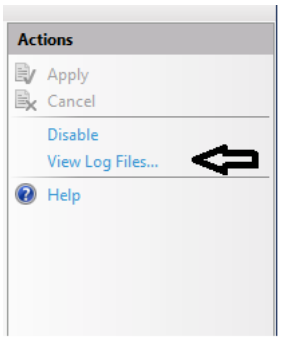
6.0.6 IIS logs

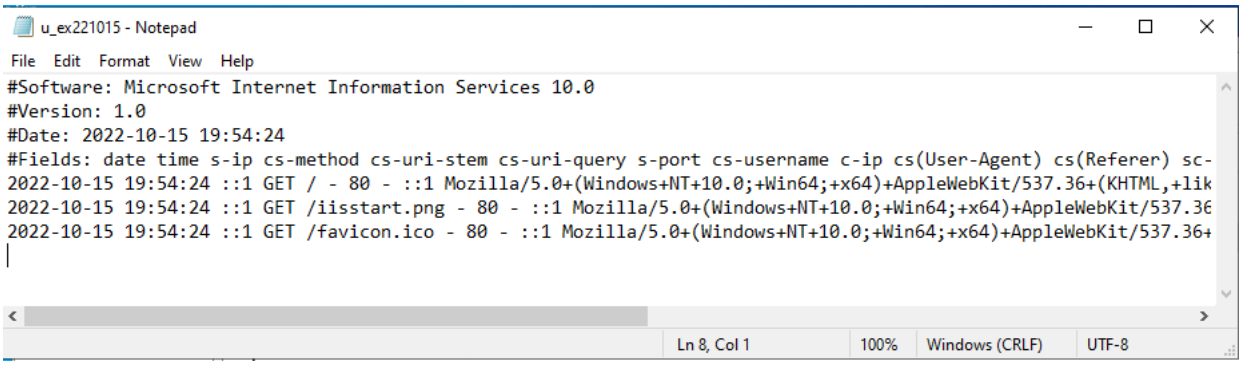

Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-K2L
Application Pools
Sites
WIN-K2LUGLN90RS Home
Filter:
Go
Show All
Group by:
IIS
Authentic...
Compression
Default Document
Directory Browsing
Error Pages
Handler Mappings
HTTP Respon...
Logging
MIME Types
Modules
Output Caching
Request Filtering
Server Certificates
Worker Processes
Management
Configurat... Editor
Feature Delegation
Shared Configurat...
Features View
Content View
Actions
Manage Server
Restart
Start
Stop
View Application Pools
View Sites
Get New Web Platform Components
Help
Ready


Logging
Use this feature to configure how IIS logs requests on the Web server.
One log file per:
Site
Log File
Format:
W3C
Select Fields...
Directory:
%SystemDrive%\inetpub\logs\LogFiles
Browse...
Encoding:
UTF-8
Log Event Destination
Select the destination where IIS will write log events.
Log file only
ETW event only
Both log file and ETW event
Log File Rollover
Select the method that IIS uses to create a new log file.
Schedule:
Daily
Maximum file size (in bytes):
Do not create new log files
Use local time for file naming and rollover


Logging
Use this feature to configure how IIS logs requests on the Web server.
One log file per:
Site
Log File
Format:
W3C
Select Fields...
Directory:
%SystemDrive%\inetpub\logs\LogFiles
Browse...
Encoding:
UTF-8
Log Event Destination
Select the destination where IIS will write log events.
Log file only
ETW event only
Both log file and ETW event
Log File Rollover
Select the method that IIS uses to create a new log file.
Schedule:
Daily
Maximum file size (in bytes):
Do not create new log files
Use local time for file naming and rollover


W3C Logging Fields
Standard Fields:
Date (date)
Time (time)
Client IP Address (c-ip)
User Name (cs-username)
Service Name (s-sitename)
Server Name (s-computername)
Server IP Address (s-ip)
Server Port (s-port)
Method (cs-method)
URI Stem (cs-uri-stem)
URI Query (cs-uri-query)
Protocol Status (sc-status)
Protocol Substatus (sc-substatus)
Custom Fields:
Log Field
Source Type
Source
Add Field...
Remove Field
Edit Field...
OK
Cancel


Actions
Apply
Cancel
Disable
View Log Files...
Help


u_ex221015 - Notepad
File Edit Format View Help
#Software: Microsoft Internet Information Services 10.0
#Version: 1.0
#Date: 2022-10-15 19:54:24
#Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-
2022-10-15 19:54:24 ::1 GET / - 80 - ::1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+lik
2022-10-15 19:54:24 ::1 GET /iisstart.png - 80 - ::1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36
2022-10-15 19:54:24 ::1 GET /favicon.ico - 80 - ::1 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+
Ln 8, Col 1
100%
Windows (CRLF)
UTF-8

Server side scripting

In general terms, a web application can be divided into two: the front-end, or the part of the application that interacts with the end-user, and the back-end, or the part of the application that handles the underlying logic.

Server-side scripts are programs that run on a web server to generate dynamic web pages, creating a unique experience for each user and limit access to proprietary data.

What are Scripting Languages?

A scripting language is a programming language that executes tasks within a special run-time environment by an interpreter instead of a compiler. **They are usually short, fast, and interpreted from source code or bytecode**.

Server-side scripting language.

The term server-side scripting language refers to those that run off a web server. Since it performs from the back-end side, the script is not visible to the visitor. Because of that, it is a more secure approach.

They are often used to create dynamic websites and platforms, handle user queries, and generate and provide data and others. A famous example of server-side scripting is the use of PHP in WordPress.

Examples: **PHP, Python, Node.js, Perl, and Ruby.**

Client-side scripting language.

Unlike the above, client-side scripting languages run off the user's browser.

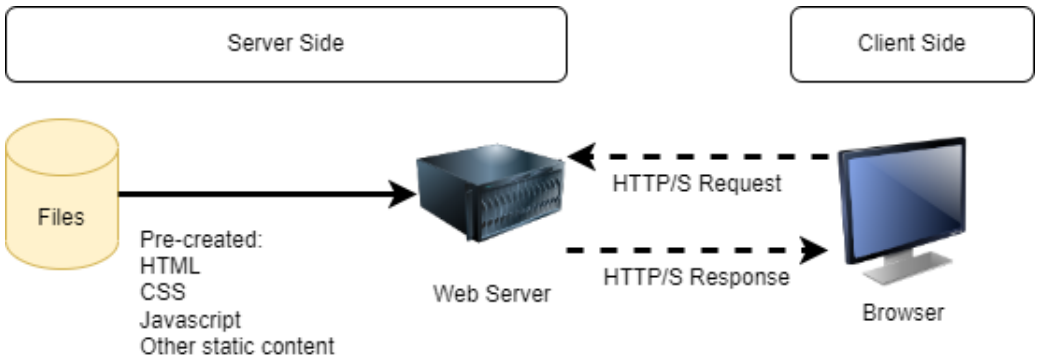
It is usually performed at the front-end, which makes it visible to visitors and makes it less vulnerable to exploits and leaks. As such, it is often used to build user interfaces and lighter functionality such as that.

Since it runs locally, they usually provide better performance and, therefore, do not strain your server.

Examples: **HTML, CSS, jQuery, and JavaScript.**

7.1 Static Sites

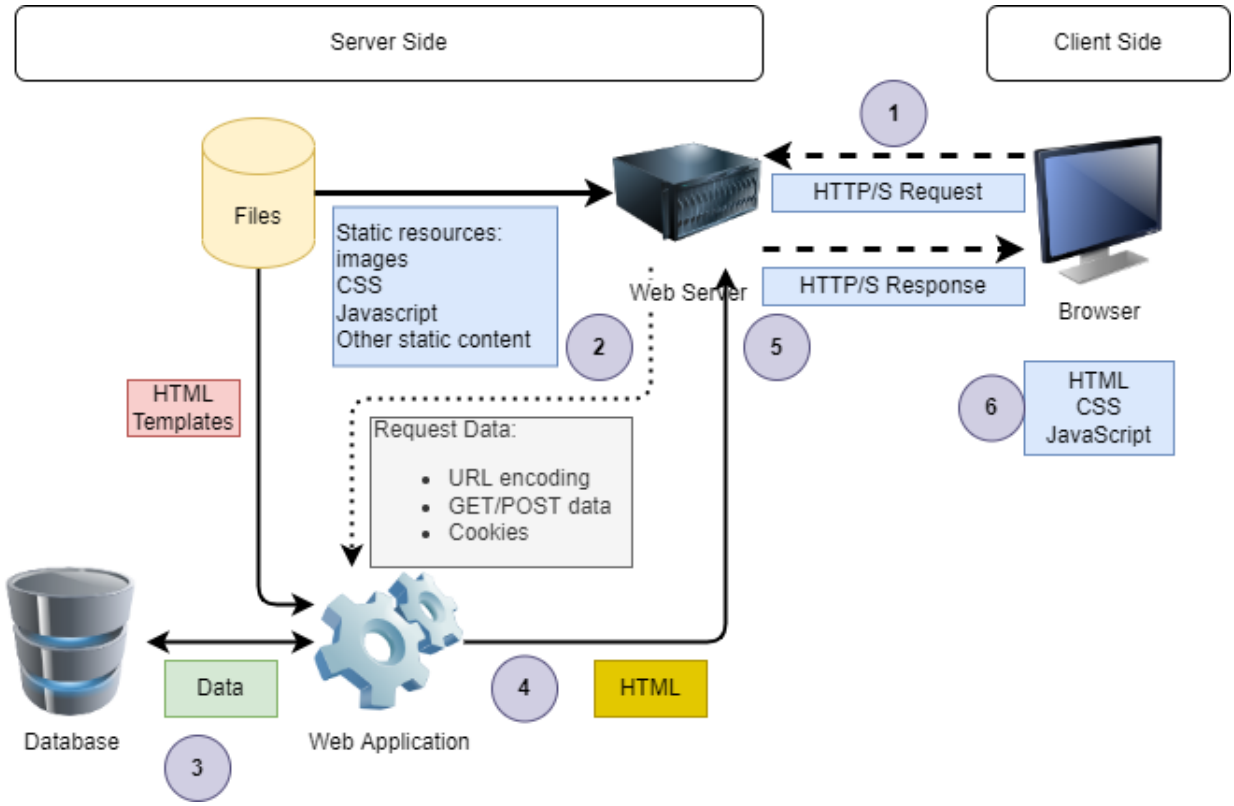
A static site is one that returns the same hard-coded content from the server whenever a particular resource is requested.The server retrieves the requested document from its file system and returns an HTTP response containing the document and a success status (usually 200 OK). If the file cannot be retrieved for some reason, an error status is returned.



7.2 Dynamic Sites

A dynamic website is one where some of the response content is generated dynamically, only when needed. On a dynamic website HTML pages are normally created by inserting data from a database into placeholders in HTML

templates. Most of the code to support a dynamic website must run on the server. Creating this code is known as **"server-side programming"** (or sometimes **"back-end scripting"**).



Requests for static data are handled in the same way as for static sites. Requests for dynamic resources are instead forwarded (2) to server-side code (Web Application). For "dynamic requests" the server interprets the request, reads required information from the database(if needed) (3), combines the retrieved data with HTML templates (4), and sends back a response containing the generated HTML (5,6).

7.2.1 server-side vs. client-side programming

Code running in the browser is known as client-side code and is primarily concerned with improving the appearance and behavior of a rendered web page. This includes selecting and styling UI components, creating layouts, navigation, form validation, etc.

By contrast, server-side website programming mostly involves choosing which content is returned to the browser in response to requests. The server-side code

handles tasks like validating submitted data and requests, using databases to store and retrieve data and sending the correct data to the client as required.

Client-side code is written using HTML, CSS, and JavaScript — it is run inside a web browser and has little or no access to the underlying operating system, This is sometimes called sandboxing.

Server-side code can be written in any number of programming languages — some of the popular server-side web languages include PHP, Python, Ruby, C#, and JavaScript (NodeJS). The server-side code has full access to the server operating system and the programmer can choose what programming language they wish to use.

7.2.2 Apache server side scripts. PHP CGI,...

7.2.2.0 PHP

https://www.php.net/
PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Commands

```
sudo apt install php libapache2-mod-php
```

Test PHP

```
root@occc-VirtualBox:~# php -version
PHP 8.1.2-1ubuntu2.6 (cli) (built: Sep 15 2022 11:30:49) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
with Zend OPcache v8.1.2-1ubuntu2.6, Copyright (c), by Zend Technologi
root@occc-VirtualBox:~#
```

Enable PHP module in apache. It may already be enabled.

Commands

```
root@occc-VirtualBox:~# a2enmod php8.1
Considering dependency mpm_prefork for php8.1:
Considering conflict mpm_event for mpm_prefork:
Considering conflict mpm_worker for mpm_prefork:
Module mpm_prefork already enabled
Considering conflict php5 for php8.1:
Module php8.1 already enabled
root@occc-VirtualBox:~#
```

restart apache.

Commands

```
sudo systemctl restart apache2
```

Edit /etc/apache2/mods-enabled/dir.conf

orig dir.conf

```
1 <IfModule mod_dir.c>
2    DirectoryIndex index.html index.cgi index.pl ↙
       ↳  index.php index.xhtml index.htm
3 </IfModule>
```

Make change so that it look like below.

new dir.conf

```
1 <IfModule mod_dir.c>
2     DirectoryIndex index.php index.html index.cgi ↙
        ↳  index.pl index.xhtml index.htm
3 </IfModule>
```

We moved the index.php into the first position so that it will serve that file first before index.html

Install PHP Modules

In order to enhance the functionality of PHP, we can optionally install some additional modules.

To see the available extra options for PHP modules and libraries, you can pipe the results of **apt-cache search** into **less**, a pager which lets you scroll through the output of other commands:

Commands

```
root@occc-VirtualBox:/etc/apache2/mods-enabled# apt-cache search php-
libnet-libidn-perl - Perl bindings for GNU Libidn
php-all-dev - package depending on all supported PHP development packages
php-cgi - server-side, HTML-embedded scripting language (CGI binary) (default)
php-cli - command-line interpreter for the PHP scripting language (default)
php-common - Common files for PHP packages
php-curl - CURL module for PHP [default]
php-dev - Files for PHP module development (default)
php-gd - GD module for PHP [default]
php-gmp - GMP module for PHP [default]
php-json - JSON module for PHP [default]
php-ldap - LDAP module for PHP [default]
php-mysql - MySQL module for PHP [default]
php-odbc - ODBC module for PHP [default]
php-pear - PEAR Base System
php-pgsql - PostgreSQL module for PHP [default]
php-pspell - pspell module for PHP [default]
php-snmp - SNMP module for PHP [default]
php-sqlite3 - SQLite3 module for PHP [default]
php-tidy - tidy module for PHP [default]
php-xml - DOM, SimpleXML, WDDX, XML, and XSL module for PHP [default]
php8.1-cgi - server-side, HTML-embedded scripting language (CGI binary)
php8.1-cli - command-line interpreter for the PHP scripting language
php8.1-common - documentation, examples and common module for PHP
php8.1-curl - CURL module for PHP
php8.1-gd - GD module for PHP
php8.1-gmp - GMP module for PHP
php8.1-ldap - LDAP module for PHP
php8.1-mbstring - MBSTRING module for PHP
php8.1-mysql - MySQL module for PHP
php8.1-odbc - ODBC module for PHP
php8.1-opcache - Zend OpCache module for PHP
php8.1-pgsql - PostgreSQL module for PHP
php8.1-pspell - pspell module for PHP
php8.1-readline - readline module for PHP
php8.1-snmp - SNMP module for PHP
php8.1-sqlite3 - SQLite3 module for PHP
php8.1-tidy - tidy module for PHP
php8.1-xml - DOM, SimpleXML, XML, and XSL module for PHP
pkg-php-tools - various packaging tools and scripts for PHP packages
bandwidthd-pgsql - Tracks usage of TCP/IP and builds html files with graphs
bluefish - advanced Gtk+ text editor for web and software development
cacti - web interface for graphing of monitoring systems
cakephp-scripts - rapid application development framework for PHP (scripts)
elpa-php-mode - PHP Mode for GNU Emacs
ganglia-webfrontend - cluster monitoring toolkit - web front-end
garagemq - AMQP message broker implemented with golang
haserl - CGI scripting program for embedded environments
```

```
icinga-php-library - Icinga PHP Library for Icinga Web 2
icinga-php-thirdparty - Icinga PHP Thirdparty libraries for Icinga Web 2
icingaweb2-module-reactbundle - 3rd party libraries php library for Icinga Web 2
kdevelop-php-l10n - localization files for KDevelop PHP plugin
libcode-tidyall-perl - your all-in-one code tidier and validator
libjs-php-date-formatter - Manipulate date/times using PHP date-time formats in javascript
libphp-adodb - ADOdb is a PHP database abstraction layer library
libphp-embed - HTML-embedded scripting language (Embedded SAPI library) (default)
libphp-jabber - Object-oriented PHP interface for the Jabber/XMPP protocol
libphp-jpgraph - Object oriented graph library for php
libphp-jpgraph-examples - Object oriented graph library for php (examples)
libphp-phpmailer - full featured email transfer class for PHP
libphp-serialization-perl - Perl module to manipulate serialized PHP data structures
libphp-simplepie - RSS and Atom feed parsing in PHP
libphp-snoopy - Snoopy is a PHP class that simulates a web browser
libphp8.1-embed - HTML-embedded scripting language (Embedded SAPI library)
mlmmj-php-web - web interface for mlmmj, written in php
mlmmj-php-web-admin - administrative web interface for mlmmj, written in php
node-babel7-standalone - Standalone build of Babel for use in browsers and other non-
Node.js environments
pdepend - design quality metrics for PHP packages
php-amqp - AMQP extension for PHP
php-amqp-all-dev - AMQP extension for PHP
php-amqplib - pure PHP implementation of the AMQP protocol
php-apcu - APC User Cache for PHP
php-apcu-all-dev - APC User Cache for PHP
php-arthurhoaro-web-thumbnailer - PHP library which will retrieve a thumbnail for any given URL
php-ast - AST extension for PHP 7
php-ast-all-dev - AST extension for PHP 7
php-async-aws-core - AsyncAws Core - AsyncAws component
php-async-aws-ses - AsyncAws Ses - AsyncAws component
php-async-aws-sns - AsyncAws Sns - AsyncAws component
php-async-aws-sqs - AsyncAws Sqs - AsyncAws component
php-auth-sasl - Abstraction of various SASL mechanism responses
php-bacon-qr-code - QR Code Generator for PHP
php-bcmath - Bcmath module for PHP [default]
php-bz2 - bzip2 module for PHP [default]
php-cache-integration-tests - Integration tests for PSR-6 and PSR-
16 cache implementations
php-cache-lite - Fast and Safe little cache system
php-cache-tag-interop - Framework interoperable interfaces for tags
php-cas - Central Authentication Service client library in php
php-codecoverage - collection, processing, and rendering for code coverage
php-codesniffer - PHP, CSS and JavaScript coding standard analyzer and checker
php-composer-ca-bundle - utility library to find a path to the system CA bundle
php-composer-metadata-minifier - Small utility library that handles metadata minification and expa
php-composer-pcre - PCRE wrapping library that offers type-safe preg_* replacements
php-composer-semver - utilities, version constraint parsing and validation
php-composer-spdx-licenses - SPDX licenses list and validation library
php-composer-xdebug-handler - Restarts a process without Xdebug
php-console-commandline - Full featured command line options and arguments parser
php-console-table - Library that makes it easy to build console style tables
php-constant-time - Constant-time Implementations of RFC 4648 Encoding (Base-64, Base-
32, Base-16)
php-crypt-gpg - PHP PEAR module for encrypting and decrypting with GnuPG
php-dapphp-radius - pure PHP RADIUS client based on the SysCo/al implementation
php-dasprid-enum - PHP 7.1 enum implementation
php-date - Generic date/time handling class for PEAR
```

```
php-db - Database Abstraction Layer
php-deepcopy - create deep copies (clones) of objects
php-directory-scanner - recursive directory scanner and filter
php-doctrine-annotations - Docblock Annotations Parser - Doctrine component
php-doctrine-cache - cache library - Doctrine component
php-doctrine-collections - Collections Abstraction library - Doctrine component
php-doctrine-common - common extensions for Doctrine
php-doctrine-data-fixtures - Data Fixtures for all Doctrine Object Managers
php-doctrine-dbal - database abstraction layer for Doctrine
php-doctrine-deprecations - Doctrine Deprecations - Doctrine component
php-doctrine-event-manager - Doctrine Event Manager component
php-doctrine-inflector - string manipulations library - Doctrine component
php-doctrine-instantiator - lightweight utility to instantiate objects in PHP
php-doctrine-lexer - base lexer library - Doctrine component
php-doctrine-orm - tool for object-relational mapping
php-doctrine-persistence - Doctrine Persistence abstractions
php-doctrine-reflection - Doctrine Reflection component
php-dompdf - HTML to PDF converter
php-dragonmantank-cron-expression - cron expression parser for PHP
php-ds - PHP extension providing efficient data structures for PHP 7
php-ds-all-dev - PHP extension providing efficient data structures for PHP 7
php-easyrdf - PHP library to consume and produce RDF
php-email-validator - A library for validating emails against several RFCs
php-enchant - Enchant module for PHP [default]
php-excimer - PHP extension that provides a non-static, non-global profiler
php-facedetect - Detect faces with PHP
php-fdomdocument - extension to PHP's standard DOM
php-fig-link-util - Common utility implementations for HTTP links
php-file-iterator - FilterIterator implementation for PHP
php-finder-facade - convenience wrapper for Symfony's Finder component
php-finder-facade-doc - convenience wrapper for Symfony's Finder component - documentation
php-font-lib - read, parse, export and make subsets of different fonts
php-fpdf - PHP class to generate PDF files
php-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary) (default)
php-fxsl - XSL wrapper and extension to XSLTProcessor
php-gearman - PHP wrapper to libgearman
php-gearman-all-dev - PHP wrapper to libgearman
php-geos - GEOS bindings for PHP
php-geshi - Generic Syntax Highlighter
php-getallheaders - A polyfill for getallheaders
php-getid3 - scripts to extract information from multimedia files
php-gettext-languages - gettext languages with plural rules
php-gmagick - Provides a wrapper to the GraphicsMagick library
php-gmagick-all-dev - Provides a wrapper to the GraphicsMagick library
php-gnupg - PHP wrapper around the gpgme library
php-gnupg-all-dev - PHP wrapper around the gpgme library
php-google-recaptcha - reCAPTCHA PHP client library
php-guestfs - guest disk image management system - PHP bindings
php-guzzlehttp-promises - Guzzle promises library
php-guzzlehttp-psr7 - PSR-7 message implementation that also provides common utility methods
php-hamcrest - This is the PHP port of Hamcrest Matchers
php-htmlawed - htmLawed PHP code to purify & filter HTML
php-htmlpurifier - Standards-compliant HTML filter
php-http - PECL HTTP module for PHP Extended HTTP Support
php-http-all-dev - PECL HTTP module for PHP Extended HTTP Support
php-http-httplug - HTTPlug, the HTTP client abstraction for PHP
php-http-interop-http-factory-tests - Unit tests for HTTP factories
php-http-message-factory - Factory interfaces for PSR-7 HTTP Message
```

```
php-http-promise - Promise used for asynchronous HTTP requests
php-http-psr7-integration-tests - Test suite for PSR7
php-http-request - Provides an easy way to perform HTTP requests
php-http-request2 - Provides an easy way to perform HTTP requests
php-http-webdav-server - WebDAV Server Baseclass
php-httpful - A Readable, Chainable, REST friendly, PHP HTTP Client
php-icinga - PHP library to communicate with and use Icinga
php-igbinary - igbinary PHP serializer
php-igbinary-all-dev - igbinary PHP serializer
php-image-text - Image_Text - Advanced text maipulations in images
php-imagick - Provides a wrapper to the ImageMagick library
php-imagick-all-dev - Provides a wrapper to the ImageMagick library
php-imap - IMAP module for PHP [default]
php-interbase - Interbase module for PHP [default]
php-intl - Internationalisation module for PHP [default]
php-invoker - Invoke callables with a timeout
php-json-schema - implementation of JSON schema
php-klogger - simple logging class
php-league-commonmark - Markdown parser based on the CommonMark JS reference implementation
php-league-flysystem - filesystem abstraction offering one API to many filesystems
php-league-html-to-markdown - An HTML-to-markdown conversion helper for PHP
php-league-mime-type-detection - generic mime-type detection interface for PHP
php-letodms-core - Document management system
php-libvirt-php - libvirt bindings for PHP
php-log - Logging Framework
php-lorenzo-pinky - A Foundation for Emails (Inky) template transpiler
php-luasandbox - PHP extension that provides a sandboxed Lua environment
php-mail - Class that provides multiple interfaces for sending emails
php-mail-mime - PHP PEAR module for creating MIME messages
php-mailparse - Email message manipulation for PHP
php-mailparse-all-dev - Email message manipulation for PHP
php-malkusch-lock - mutex library for exclusive code execution
php-mapi - Complete and feature rich groupware solution - PHP MAPI bindings
php-mariadb-mysql-kbs - Knowledge base about MariaDB and MySQL server variables
php-masterminds-html5 - An HTML5 parser and serializer
php-mbstring - MBSTRING module for PHP [default]
php-mdb2 - database abstraction layer
php-mdb2-driver-mysql - mysql MDB2 driver
php-mdb2-driver-pgsql - pgsql MDB2 driver
php-memcache - memcache extension module for PHP
php-memcache-all-dev - memcache extension module for PHP
php-memcached - memcached extension module for PHP, uses libmemcached
php-memcached-all-dev - memcached extension module for PHP, uses libmemcached
php-mf2 - Microformats2 is the simplest way to markup structured information in HTML
php-mikey179-vfsstream - Virtual file system to mock the real file system in unit tests
php-mime-type - Utility class for dealing with MIME types
php-mock - mock built-in PHP functions
php-mock-integration - integration package for PHP-Mock
php-mock-phpunit - mock built-in PHP functions with PHPUnit
php-mockery - mock object framework for PHPUnit and other testing framework
php-mockery-doc - mock object framework for PHPUnit - documentation
php-mongodb - MongoDB driver for PHP
php-mongodb-all-dev - MongoDB driver for PHP
php-monolog - send logs to various destination and web services
php-msgpack - PHP extension for interfacing with MessagePack
php-msgpack-all-dev - PHP extension for interfacing with MessagePack
php-net-dime - The Net_DIME package implements DIME encoding and decoding
php-net-dns2 - PHP Resolver library used to communicate with a DNS server
```

```
php-net-ftp - Net_FTP provides an OO interface to the PHP FTP functions plus some additions
php-net-imap - Provides an implementation of the IMAP protocol
php-net-ipv6 - Check and validate IPv6 addresses
php-net-ldap2 - Object oriented interface for searching and manipulating LDAP-entries
php-net-ldap3 - Object oriented interface for searching and manipulating LDAP entries
php-net-nntp - NNTP implementation
php-net-publicsuffix - PHP module for detecting registered domains and public suffixes
php-net-sieve - Handles talking to a sieve server
php-net-smtp - PHP PEAR module implementing SMTP protocol
php-net-socket - Network Socket Interface
php-net-url - Easy parsing of Urls
php-net-url2 - Class for parsing and handling URL
php-net-whois - PHP PEAR module for querying whois services
php-netscape-bookmark-parser - generic Netscape bookmark parser
php-nikic-fast-route - Fast request router for PHP
php-nrk-predis - transitional dummy package for php-predis
php-nyholm-psr7 - A fast PHP7 implementation of PSR-7
php-oauth - OAuth 1.0 consumer and provider extension
php-oauth-all-dev - OAuth 1.0 consumer and provider extension
php-opis-closure - serializable closures (anonymous functions) for PHP
php-parsedown - Parser for Markdown
php-parsedown-extra - Markdown Extra extension for Parsedown
php-parser - convert PHP code into abstract syntax tree
php-patchwork-utf8 - UTF-8 strings handling for PHP
php-pclzip - ZIP archive manager class for PHP
php-pcov - Code coverage driver
php-pcov-all-dev - Code coverage driver
php-pda-pheanstalk - PHP client for beanstalkd queue
php-phar-io-manifest - reading phar.io manifest information from a PHP Archive (Phar)
php-phar-io-version - handling version information and constraint
php-php-gettext - read gettext MO files directly, without requiring anything other than PHP
php-phpdbg - server-side, HTML-embedded scripting language (PHPDBG binary) (default)
php-phpdocumentor-reflection-common - Common reflection classes - phpDocumentor component
php-phpdocumentor-reflection-docblock - DocBlock parser - phpDocumentor component
php-phpdocumentor-type-resolver - TypeResolver and FqsenResolver - phpDocumentor component
php-phpmyadmin-motranslator - translation API for PHP using Gettext MO files
php-phpmyadmin-shapefile - translation API for PHP using Gettext MO files
php-phpmyadmin-sql-parser - validating SQL lexer and parser
php-phpoption - Option type for PHP
php-phpseclib - implementations of an arbitrary-precision integer arithmetic library
php-phpseclib3 - implementations of an arbitrary-precision integer arithmetic library
php-phpspec-prophecy - object mocking framework - phpspec component
php-phpspec-prophecy-phpunit - Integrating the Prophecy mocking library in PHPUnit test cases
php-phpstan-phpdoc-parser - PHPDoc parser with support for nullable, intersection and generic type
php-predis - Flexible and feature-complete Redis client for PHP and HHVM
php-proxy-manager - library providing utilities to operate with Object Proxies
php-ps - ps module for PHP
php-ps-all-dev - ps module for PHP
php-psr - PSR interfaces for PHP
php-psr-all-dev - PSR interfaces for PHP
php-psr-cache - Common interface for caching libraries
php-psr-container - Common Container Interface (PHP FIG PSR-11)
php-psr-event-dispatcher - Standard interfaces for event handling
php-psr-http-client - Common interface for HTTP clients
php-psr-http-factory - Common interfaces for PSR-7 HTTP message factories
php-psr-http-message - Common interface for HTTP messages
php-psr-link - Common interfaces for HTTP links
php-psr-log - common interface for logging libraries
```

```
php-psr-simple-cache - Common interfaces for simple caching
php-pubsubhubbub-publisher - WebSub publisher library for PHP
php-ramsey-uuid - RFC 4122 universally unique identifier (UUID) generator for PHP
php-random-compat - PHP 5.x polyfill for random_bytes() and random_int() from PHP 7
php-raphf - raphf module for PHP
php-raphf-all-dev - raphf module for PHP
php-react-promise - lightweight implementation of CommonJS Promises/A for PHP
php-readline - readline module for PHP [default]
php-redis - PHP extension for interfacing with Redis
php-redis-all-dev - PHP extension for interfacing with Redis
php-remctl - PECL module for Kerberos-authenticated command execution
php-roundcube-rtf-html-php - RTF to HTML converter in PHP
php-rrd - PHP bindings to rrd tool system
php-rrd-all-dev - PHP bindings to rrd tool system
php-sabre-dav - WebDAV Framework for PHP
php-sabre-vobject - library to parse and manipulate iCalendar and vCard objects
php-sass - PHP bindings to libsass - fast, native Sass parsing in PHP
php-seclib - implementations of an arbitrary-precision integer arithmetic library
php-services-json - PHP implementaion of json_encode/decode
php-services-weather - This class acts as an interface to various online weather-
services
php-shellcommand - An object oriented interface to shell commands
php-smbclient - PHP wrapper for libsmbclient
php-smbclient-all-dev - PHP wrapper for libsmbclient
php-soap - SOAP module for PHP [default]
php-solr - PHP extension for communicating with Apache Solr server
php-solr-all-dev - PHP extension for communicating with Apache Solr server
php-sql-formatter - a PHP SQL highlighting library
php-ssh2 - Bindings for the libssh2 library
php-ssh2-all-dev - Bindings for the libssh2 library
php-swiftmailer - Swiftmailer, free feature-rich PHP mailer
php-sybase - Sybase module for PHP [default]
php-symfony - set of reusable components and framework for web projects
php-symfony-all-my-sms-notifier - Symfony AllMySms Notifier Bridge
php-symfony-amazon-mailer - Symfony Amazon Mailer Bridge
php-symfony-amazon-sns-notifier - Symfony Amazon SNS Notifier Bridge
php-symfony-amazon-sqs-messenger - Symfony Amazon SQS extension Messenger Bridge
php-symfony-amqp-messenger - Symfony AMQP extension Messenger Bridge
php-symfony-asset - manage asset URLs
php-symfony-beanstalkd-messenger - Symfony Beanstalkd Messenger Bridge
php-symfony-browser-kit - simulate the behavior of a web browser
php-symfony-cache - provides an extended PSR-6, PSR-16 (and tags) implementation
php-symfony-cache-contracts - Generic abstractions related to caching
php-symfony-clickatell-notifier - Symfony Clickatell Notifier Bridge
php-symfony-config - load configurations from different data sources
php-symfony-console - run tasks from the command line
php-symfony-contracts - A set of abstractions extracted out of the Symfony components
php-symfony-crowdin-translation-provider - Symfony Crowdin Translation Provider Bridge
php-symfony-css-selector - convert CSS selectors to XPath expressions
php-symfony-debug-bundle - debugging tools for the Symfony framework
php-symfony-dependency-injection - standardize and centralize construction of objects
php-symfony-deprecation-contracts - A generic function and convention to trigger deprecation notic
php-symfony-discord-notifier - Symfony Discord Notifier Bridge
php-symfony-doctrine-bridge - integration for Doctrine with Symfony Components
php-symfony-doctrine-messenger - Symfony Doctrine Messenger Bridge
php-symfony-dom-crawler - ease DOM navigation for HTML and XML documents
php-symfony-dotenv - .env files parser to make environment variables accessible
php-symfony-error-handler - manage errors and ease debugging
```

```
php-symfony-esendex-notifier - Symfony Esendex Notifier Bridge
php-symfony-event-dispatcher - dispatch events and listen to them
php-symfony-event-dispatcher-contracts - Generic abstractions related to dispatching event
php-symfony-expo-notifier - Symfony Expo Notifier Bridge
php-symfony-expression-language - compile and evaluate expressions
php-symfony-fake-chat-notifier - Fake Chat (as email or log during development) Notifier Bridge
php-symfony-fake-sms-notifier - Fake SMS (as email or log during development) Notifier Bridge
php-symfony-filesystem - basic filesystem utilities
php-symfony-finder - find files and directories
php-symfony-firebase-notifier - Symfony Firebase Notifier Bridge
php-symfony-form - create HTML forms and process request data
php-symfony-framework-bundle - basic, robust and flexible MVC framework
php-symfony-free-mobile-notifier - Symfony Free Mobile Notifier Bridge
php-symfony-gateway-api-notifier - Symfony GatewayApi Notifier Bridge
php-symfony-gitter-notifier - Symfony Gitter Notifier Bridge
php-symfony-google-chat-notifier - Symfony Google Chat Notifier Bridge
php-symfony-google-mailer - Symfony Google Mailer Bridge
php-symfony-http-client - methods to fetch HTTP resources synchronously or asynchronously
php-symfony-http-client-contracts - Generic abstractions related to HTTP clients
php-symfony-http-foundation - object-oriented layer for the HTTP specification
php-symfony-http-kernel - building blocks for flexible and fast HTTP-based frameworks
php-symfony-inflector - words conversion between their singular and plural forms
php-symfony-infobip-notifier - Symfony Infobip Notifier Bridge
php-symfony-intl - limited replacement layer for the PHP extension intl
php-symfony-iqsms-notifier - Symfony Iqsms Notifier Bridge
php-symfony-ldap - abstraction layer for the PHP LDAP module
php-symfony-light-sms-notifier - Symfony LightSms Notifier Bridge
php-symfony-linked-in-notifier - Symfony LinkedIn Notifier Bridge
php-symfony-lock - create and manage locks
php-symfony-loco-translation-provider - Symfony Loco Translation Provider Bridge
php-symfony-lokalise-translation-provider - Symfony Lokalise Translation Provider Bridge
php-symfony-mailchimp-mailer - Symfony Mailchimp Mailer Bridge
php-symfony-mailer - help sending emails
php-symfony-mailgun-mailer - Symfony Mailgun Mailer Bridge
php-symfony-mailjet-mailer - Symfony Mailjet Mailer Bridge
php-symfony-mailjet-notifier - Symfony Mailjet Notifier Bridge
php-symfony-mattermost-notifier - Symfony Mattermost Notifier Bridge
php-symfony-mercure - publisher part of the Mercure Protocol
php-symfony-mercure-notifier - Symfony Mercure Notifier Bridge
php-symfony-message-bird-notifier - Symfony MessageBird Notifier Bridge
php-symfony-message-media-notifier - Symfony MessageMedia Notifier Bridge
php-symfony-messenger - send and receive messages
php-symfony-microsoft-teams-notifier - Symfony Microsoft Teams Notifier Bridge
php-symfony-mime - library to manipulate MIME messages
php-symfony-mobyt-notifier - Symfony Mobyt Notifier Bridge
php-symfony-monolog-bridge - integration for Monolog with Symfony Components
php-symfony-nexmo-notifier - Symfony Nexmo Notifier Bridge
php-symfony-notifier - Sends notifications via one or more channels (email, SMS, ...)
php-symfony-octopush-notifier - Symfony Octopush Notifier Bridge
php-symfony-oh-my-smtp-mailer - Symfony OhMySMTP Mailer Bridge
php-symfony-one-signal-notifier - Symfony OneSignal Notifier Bridge
php-symfony-options-resolver - configure objects with option arrays
php-symfony-ovh-cloud-notifier - Symfony OvhCloud Notifier Bridge
php-symfony-password-hasher - password hashing utilities - Symfony Component
php-symfony-phpunit-bridge - integration for PHPUnit with Symfony Components
php-symfony-polyfill - Symfony polyfills backporting features to lower PHP versions
php-symfony-polyfill-apcu - Symfony polyfill backporting apcu_* functions to lower PHP versions
php-symfony-polyfill-ctype - Symfony polyfill for ctype functions
```

```
php-symfony-polyfill-iconv - Symfony polyfill for the Iconv extension
php-symfony-polyfill-intl-grapheme - Symfony polyfill for intl's grapheme_* functions
php-symfony-polyfill-intl-icu - Symfony polyfill for intl's ICU-related data and classes
php-symfony-polyfill-intl-idn - Symfony polyfill for intl's idn_to_ascii and idn_to_utf8 functions
php-symfony-polyfill-intl-messageformatter - Symfony polyfill for intl's MessageFormatter class an
php-symfony-polyfill-intl-normalizer - Symfony polyfill for intl's Normalizer class and related fu
php-symfony-polyfill-mbstring - Symfony polyfill for the Mbstring extension
php-symfony-polyfill-php72 - Symfony polyfill backporting some PHP 7.2+ features to lower PHP vers
php-symfony-polyfill-php73 - Symfony polyfill backporting some PHP 7.3+ features to lower PHP vers
php-symfony-polyfill-php74 - Symfony polyfill backporting some PHP 7.4+ features to lower PHP vers
php-symfony-polyfill-php80 - Symfony polyfill backporting some PHP 8.0+ features to lower PHP vers
php-symfony-polyfill-php81 - Symfony polyfill backporting some PHP 8.1+ features to lower PHP vers
php-symfony-polyfill-util - Symfony utilities for portability of PHP codes
php-symfony-polyfill-uuid - Symfony polyfill for uuid functions
php-symfony-polyfill-xml - Symfony polyfill for xml's utf8_encode and utf8_decode functions
php-symfony-postmark-mailer - Symfony Postmark Mailer Bridge
php-symfony-process - execute commands in sub-processes
php-symfony-property-access - read from and write to an object or array
php-symfony-property-info - extract information about properties of PHP classes
php-symfony-proxy-manager-bridge - integration for ProxyManager with Symfony Components
php-symfony-rate-limiter - rate limit input and output in applications
php-symfony-redis-messenger - Symfony Redis extension Messenger Bridge
php-symfony-rocket-chat-notifier - Symfony RocketChat Notifier Bridge
php-symfony-routing - associate a request with code that generates a response
php-symfony-runtime - decouple PHP applications from global state
php-symfony-security-acl - Symfony Security Component - ACL (Access Control List)
php-symfony-security-bundle - configurable security system for the Symfony framework
php-symfony-security-core - infrastructure for authorization systems - common features
php-symfony-security-csrf - infrastructure for authorization systems - CSRF protection
php-symfony-security-guard - infrastructure for authorization systems - Guard features
php-symfony-security-http - infrastructure for authorization systems - HTTP integration
php-symfony-semaphore - provide exclusive access to a shared resource
php-symfony-sendgrid-mailer - Symfony Sendgrid Mailer Bridge
php-symfony-sendinblue-mailer - Symfony Sendinblue Mailer Bridge
php-symfony-sendinblue-notifier - Symfony Sendinblue Notifier Bridge
php-symfony-serializer - convert PHP objects into specific formats and vice versa
php-symfony-service-contracts - Generic abstractions related to writing services
php-symfony-sinch-notifier - Symfony Sinch Notifier Bridge
php-symfony-slack-notifier - Symfony Slack Notifier Bridge
php-symfony-sms-biuras-notifier - Symfony SmsBiuras Notifier Bridge
php-symfony-sms77-notifier - Symfony sms77 Notifier Bridge
php-symfony-smsapi-notifier - Symfony Smsapi Notifier Bridge
php-symfony-smsc-notifier - Symfony SMSC Notifier Bridge
php-symfony-spot-hit-notifier - Symfony Spot-Hit Notifier Bridge
php-symfony-stopwatch - profile PHP code
php-symfony-string - object-oriented API to work with strings
php-symfony-telegram-notifier - Symfony Telegram Notifier Bridge
php-symfony-telnyx-notifier - Symfony Telnyx Notifier Bridge
php-symfony-templating - tools needed to build a template system
php-symfony-translation - tools to internationalize an application
php-symfony-translation-contracts - Generic abstractions related to translation
php-symfony-turbo-sms-notifier - Symfony TurboSms Notifier Bridge
php-symfony-twig-bridge - integration for Twig with Symfony Components
php-symfony-twig-bundle - configurable integration of Twig with the Symfony framework
php-symfony-twilio-notifier - Symfony Twilio Notifier Bridge
php-symfony-uid - object-oriented API to generate and represent UIDs
php-symfony-validator - tools to validate values
php-symfony-var-dumper - mechanisms for walking through any arbitrary PHP variable
```

```
php-symfony-var-exporter - export serializable PHP data structure to plain PHP code
php-symfony-vonage-notifier - Symfony Vonage Notifier Bridge
php-symfony-web-link - manage links between resources
php-symfony-web-profiler-bundle - collect requests information for analysis and debugging
php-symfony-workflow - manage a workflow or finite state machine
php-symfony-yaml - convert YAML to PHP arrays and the other way around
php-symfony-yunpian-notifier - Symfony Yunpian Notifier Bridge
php-symfony-zulip-notifier - Symfony Zulip Notifier Bridge
php-tcpdf - PHP class for generating PDF files on-the-fly
php-text-captcha - Generation of CAPTCHAs
php-text-figlet - Engine for use FIGlet fonts to rendering text
php-text-languagedetect - Language detection class
php-text-password - Creating passwords with PHP
php-text-template - Simple template engine
php-text-wiki - transform Wiki and BBCode markup into XHTML, LaTeX or plain text markup
php-thrift - PHP language support for Thrift
php-tideways - Tideways PHP Profiler Extension
php-tijsverkoyen-css-to-inline-styles - convert HTML into HTML with inline styles
php-timer - Utility class for timing
php-token-stream - Wrapper around PHP's tokenizer extension
php-tokenizer - tokenized PHP source to XML converter
php-twig - Flexible, fast, and secure template engine for PHP
php-twig-cache-extra - A Twig extension for Symfony Cache
php-twig-cssinliner-extra - A Twig extension to allow inlining CSS
php-twig-doc - Twig template engine documentation
php-twig-extra-bundle - A Symfony bundle for extra Twig extensions
php-twig-html-extra - A Twig extension for HTML
php-twig-i18n-extension - i18n extension for the Twig template system
php-twig-inky-extra - A Twig extension for the inky email templating engine
php-twig-intl-extra - A Twig extension for Intl
php-twig-markdown-extra - A Twig extension for Markdown
php-twig-string-extra - A Twig extension for Symfony String
php-uopz - UOPZ extension for PHP 7
php-uopz-all-dev - UOPZ extension for PHP 7
php-uploadprogress - file upload progress tracking extension for PHP
php-uploadprogress-all-dev - file upload progress tracking extension for PHP
php-uuid - PHP UUID extension
php-uuid-all-dev - PHP UUID extension
php-validate - validation class
php-vlucas-phpdotenv - environment variable file loader for PHP
php-webmozart-assert - Assertions to validate method input/output with nice error messages
php-wikidiff2 - external diff engine for mediawiki
php-xajax - A library to develop Ajax applications
php-xdebug - Xdebug Module for PHP
php-xdebug-all-dev - Xdebug Module for PHP
php-xml-htmlsax3 - SAX parser for HTML and other badly formed XML documents
php-xml-rpc2 - PHP XML-RPC client/server library
php-xml-svg - XML_SVG API
php-xmlrpc - XML-RPC servers and clients functions for PHP
php-xmlrpc-all-dev - XML-RPC servers and clients functions for PHP
php-yac - YAC (Yet Another Cache) for PHP
php-yac-all-dev - YAC (Yet Another Cache) for PHP
php-yaml - YAML-1.1 parser and emitter for PHP
php-yaml-all-dev - YAML-1.1 parser and emitter for PHP
php-zend-code - Laminas Project - Code component
php-zend-eventmanager - Laminas Project - EventManager component
php-zend-stdlib - Laminas Project - Stdlib component
php-zeroc-ice - PHP extension for Ice
```

```
php-zeta-base - Zeta Components - Base package
php-zeta-console-tools - Zeta Components - ConsoleTools package
php-zeta-unit-test - Zeta Components - UnitTest package
php-zip - Zip module for PHP [default]
php-zmq - ZeroMQ messaging bindings for PHP
php-zmq-all-dev - ZeroMQ messaging bindings for PHP
php8.1-amqp - AMQP extension for PHP
php8.1-apcu - APC User Cache for PHP
php8.1-ast - AST extension for PHP 7
php8.1-bcmath - Bcmath module for PHP
php8.1-bz2 - bzip2 module for PHP
php8.1-dba - DBA module for PHP
php8.1-ds - PHP extension providing efficient data structures for PHP 7
php8.1-enchant - Enchant module for PHP
php8.1-fpm - server-side, HTML-embedded scripting language (FPM-CGI binary)
php8.1-gearman - PHP wrapper to libgearman
php8.1-gmagick - Provides a wrapper to the GraphicsMagick library
php8.1-gnupg - PHP wrapper around the gpgme library
php8.1-http - PECL HTTP module for PHP Extended HTTP Support
php8.1-igbinary - igbinary PHP serializer
php8.1-imagick - Provides a wrapper to the ImageMagick library
php8.1-imap - IMAP module for PHP
php8.1-interbase - Interbase module for PHP
php8.1-intl - Internationalisation module for PHP
php8.1-mailparse - Email message manipulation for PHP
php8.1-memcache - memcache extension module for PHP
php8.1-memcached - memcached extension module for PHP, uses libmemcached
php8.1-mongodb - MongoDB driver for PHP
php8.1-msgpack - PHP extension for interfacing with MessagePack
php8.1-oauth - OAuth 1.0 consumer and provider extension
php8.1-pcov - Code coverage driver
php8.1-phpdbg - server-side, HTML-embedded scripting language (PHPDBG binary)
php8.1-ps - ps module for PHP
php8.1-psr - PSR interfaces for PHP
php8.1-raphf - raphf module for PHP
php8.1-redis - PHP extension for interfacing with Redis
php8.1-rrd - PHP bindings to rrd tool system
php8.1-smbclient - PHP wrapper for libsmbclient
php8.1-soap - SOAP module for PHP
php8.1-solr - PHP extension for communicating with Apache Solr server
php8.1-ssh2 - Bindings for the libssh2 library
php8.1-sybase - Sybase module for PHP
php8.1-uopz - UOPZ extension for PHP 7
php8.1-uploadprogress - file upload progress tracking extension for PHP
php8.1-uuid - PHP UUID extension
php8.1-xdebug - Xdebug Module for PHP
php8.1-xmlrpc - XML-RPC servers and clients functions for PHP
php8.1-yac - YAC (Yet Another Cache) for PHP
php8.1-yaml - YAML-1.1 parser and emitter for PHP
php8.1-zip - Zip module for PHP
php8.1-zmq - ZeroMQ messaging bindings for PHP
phpab - lightweight PHP namespace aware autoload generator
phpcpd - copy and paste detector (CPD) for PHP code
phploc - tool for quickly measuring the size of a PHP project
phpmd - PHP Mess Detector
phpunit-resource-operations - provide a list of PHP built-in functions that operate on resources
rainloop - Simple, modern & fast web-based email client
tweeper - web scraper to convert supported websites like Twitter.com to RSS
```

```
php-mythtv - PHP Bindings for MythTV
root@occc-VirtualBox:/etc/apache2/mods-enabled#
```

To get more information about what each module does, you can either search the internet, or you can look at the long description of the package by typing:

Commands

```
apt-cache show package_name
```

Commands

```
root@occc-VirtualBox:/etc/apache2/mods-enabled# apt-cache show php-cli
Package: php-cli
Architecture: all
Version: 2:8.1+92ubuntu1
Priority: optional
Section: php
Source: php-defaults (92ubuntu1)
Origin: Ubuntu
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Debian PHP Maintainers <team+pkg-php@tracker.debian.org>
Bugs: https://bugs.launchpad.net/ubuntu/+filebug
Installed-Size: 25
Depends: php8.1-cli
Filename: pool/main/p/php-defaults/php-cli_8.1+92ubuntu1_all.deb
Size: 3234
MD5sum: 5fb3a24fde47119a93697d126ead9e87
SHA1: efa8a5d894ea5736c6140fdb54d4e4d45c94e57b
SHA256: 1dcc3e885ee855e23edf3bda7bcd11bd5c49cee56e71a4f30e2fd766c0fa5275
SHA512: 2c217681a37266599f94e73fbd45ab8d02a063a74e8d520875809f47ce880dc2790bc314633e653f461b212099
Description-en: command-line interpreter for the PHP scripting language (default)
This package provides the /usr/bin/php command interpreter, useful for
testing PHP scripts from a shell or performing general shell scripting tasks.
.
PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used
open source general-purpose scripting language that is especially suited
for web development and can be embedded into HTML.
.
This package is a dependency package, which depends on latest stable
PHP version (currently 8.1).
Description-md5: 232467ad58970134c89120381a7f5586

root@occc-VirtualBox:/etc/apache2/mods-enabled#
```

Locate our php.ini file.

Commands

```
root@occc-VirtualBox:/etc/apache2# php --ini | grep Loaded
Loaded Configuration File:         /etc/php/8.1/cli/php.ini
root@occc-VirtualBox:/etc/apache2#
```

At the time of PHP installation, php.ini was a special file provided as a default

configuration file.

Purpose of php.ini

- It's the main configuration file that controls what a user can or cannot do with the website.

- All the settings related to creating global variables, uploading maximum size, display log

Create a test file for PHP.

Note we will have to disable this file and the phpinfo(); command later on for security reasons.

Create a file called **phpinfo.php** within your web-server's root directory

phpinfo.php

```
1 <?php
2
3    // Show all information, defaults to INFO_ALL
4    phpinfo();
5 ?>?
```

Go to your website and test that php is working.
http://127.0.0.1/phpinfo.php

IMPORTANT!! delete the phpinfo.php file once done!!!!!!!!!!!!!

Once you've confirmed PHP is working correctly, it's important to delete phpinfo.php as it contains information that could be useful to hackers.

7.2.2.0 CGI

The CGI (Common Gateway Interface) defines a way for a web server to interact with external content-generating programs, which are often referred to as CGI programs or CGI scripts.

Configuring Apache to permit CGI

Create a cgi directory.

Commands

```
mkdir /webroot/cgi-bin
```

In **/etc/apache2/conf-available** edit the **serve-cgi-bin.conf** file.

enable cgi

```
1  <IfModule mod_alias.c>
2    <IfModule mod_cgi.c>
3      Define ENABLE_USR_LIB_CGI_BIN
4    </IfModule>
5
6    <IfModule mod_cgid.c>
7      Define ENABLE_USR_LIB_CGI_BIN
8    </IfModule>
9
10   <IfDefine ENABLE_USR_LIB_CGI_BIN>
11     ScriptAlias /cgi-bin/ /webroot/cgi-bin/
12     <Directory "/webroot/cgi-bin/">
13       AllowOverride None
14       Options +ExecCGI -MultiViews +↙
           ↪  SymLinksIfOwnerMatch
15       AddHandler cgi-script .cgi .py
16       SetHandler cgi-script
17       Require all granted
18     </Directory>
19   </IfDefine>
20 </IfModule>
```

Commands

```
root@occc-VirtualBox:/webroot/cgi-bin# a2enmod cgi
Enabling module cgi.
To activate the new configuration, you need to run:
systemctl restart apache2
root@occc-VirtualBox:/webroot/cgi-bin# systemctl restart apache2
root@occc-VirtualBox:/webroot/cgi-bin#
```

Restart webserver to have new directory read in.
In your webroot/cgi-gin directory create a small python script.

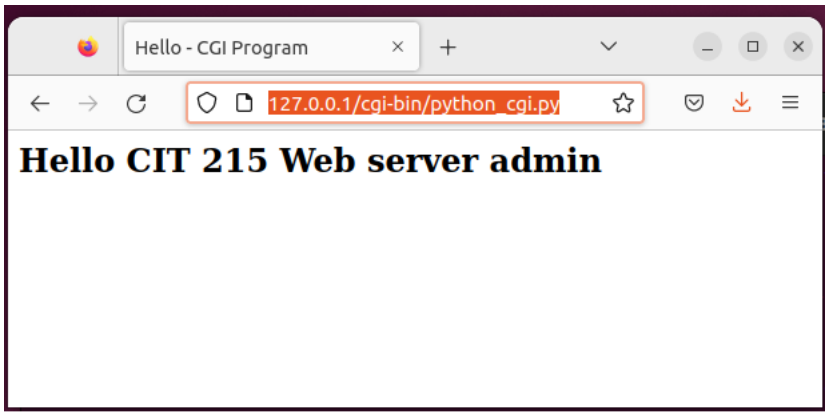
python cgi script

```
1  #!/usr/bin/python
2
3
4  print( "Content-type:text/html\r\n\r\n")
5  print( "<html>")
6  print( "<head>")
7  print( "<title>Hello -  CGI Program</title>")
8  print( "</head>")
9  print( "<body>")
10 print( "<h2>Hello {0} {1} </h2>".format("CIT 215","↙
     ↳  Web server admin"))
11 print( "</body>")
12 print( "</html>")
```

make sript executable

```
chmod 755 python_cgi.py
```

go to **http://127.0.0.1/cgi-bin/python_cgi.py**



ScriptAlias /cgi-bin/ /webroot/cgi-bin/

> ScriptAlias URLpath directory

The ScriptAlias directive converts requests for URLs starting with URLpath to execution of the CGI program found in directory.

```
AddHandler cgi-script .cgi .py
SetHandler cgi-script
```

This will add any file we get with the extension .cgi or .py to be runned as a cgi script. There will be a need to secure this directory and add other security measures to protect the server.

7.2.3 Nginx server side scripts. PHP CGI,...

PHP-FPM (FastCGI Process Manager) is an alternative to FastCGI implementation of PHP with some additional features useful for sites with high traffic. It is the preferred method of processing PHP pages with NGINX and is faster than traditional CGI based methods.

Commands

```
sudo apt install php php-cli php-fpm
```

Check that php-fpm is running.

Commands

```
systemctl status php8.1-fpm.service
```

```
root@occc-VirtualBox:/etc/nginx# systemctl status php8.1-fpm.service
● php8.1-fpm.service - The PHP 8.1 FastCGI Process Manager
     Loaded: loaded (/lib/systemd/system/php8.1-fpm.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2022-11-12 09:37:07 EST; 1 week 1 day ago
       Docs: man:php-fpm8.1(8)
   Main PID: 829 (php-fpm8.1)
     Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
      Tasks: 3 (limit: 9457)
     Memory: 12.3M
        CPU: 18.055s
     CGroup: /system.slice/php8.1-fpm.service
             ├─829 "php-fpm: master process (/etc/php/8.1/fpm/php-fpm.conf)" "" "" "" "" "" "" "" "" "" "" "
             ├─874 "php-fpm: pool www" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "
             └─875 "php-fpm: pool www" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "

Nov 12 09:37:06 occc-VirtualBox systemd[1]: Starting The PHP 8.1 FastCGI Process Manager...
Nov 12 09:37:07 occc-VirtualBox systemd[1]: Started The PHP 8.1 FastCGI Process Manager.
lines 1-16/16 (END)
```

7.2.3.0 Add PHP support to Nginx

With Nginx and PHP-FPM installed, you must edit the default Nginx config file. This will allow the PHP FastCGI Process Manager to handle requests that have a **.php** extension.

Enable PHP in Nginx's config file

Commands

```
vi /etc/nginx/sites-available/cit215.edu
```

nginx enable php

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

    root /webroot/cit215;
    # Add index.php to setup Nginx, PHP & PHP-FPM config
    index index.php index.html index.htm index.nginx-debian.html;

    server_name cit215.sunyorange.edu.local;

    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }
    # pass PHP scripts on Nginx to FastCGI (PHP-FPM) server
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;

        # Nginx php-fpm sock config:
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
        # Nginx php-cgi config :
        # Nginx PHP fastcgi_pass 127.0.0.1:9000;
    }

    # deny access to Apache .htaccess on Nginx with PHP,
    # if Apache and Nginx document roots concur
    location ~ /\.ht {
        deny all;
    }
}
```

validate an Nginx config file

Commands

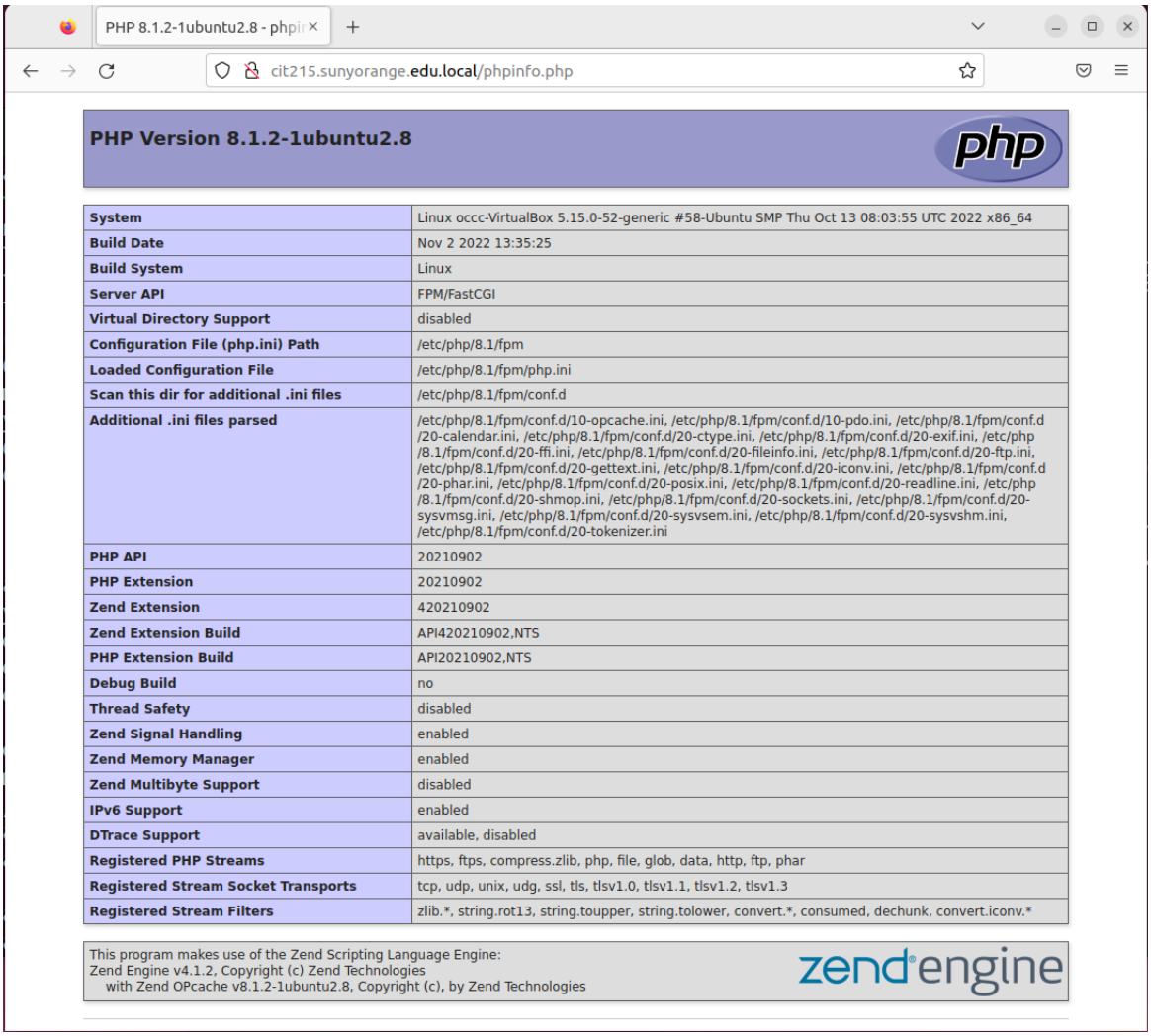
```
root@occc-VirtualBox:/etc/nginx# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@occc-VirtualBox:/etc/nginx#
```

To enable the Nginx PHP fastCGI setup, restart the server:

sudo systemctl restart nginx

```
root@occc-VirtualBox:/etc/nginx# systemctl restart nginx
root@occc-VirtualBox:/etc/nginx# systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2022-11-20 12:46:11 EST; 6s ago
       Docs: man:nginx(8)
    Process: 64718 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, statu>
    Process: 64719 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCE>
   Main PID: 64720 (nginx)
      Tasks: 2 (limit: 9457)
     Memory: 2.6M
        CPU: 10ms
     CGroup: /system.slice/nginx.service
             ├─64720 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─64721 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">

Nov 20 12:46:11 occc-VirtualBox systemd[1]: Starting A high performance web server and a reverse proxy ser>
Nov 20 12:46:11 occc-VirtualBox systemd[1]: Started A high performance web server and a reverse proxy serv>
lines 1-16/16 (END)
```

create a phpinfo web page(DELETE AFTER!!!!)



7.2.4 IIS server side scripts. PHP CGI,...

https://windows.php.net/download/

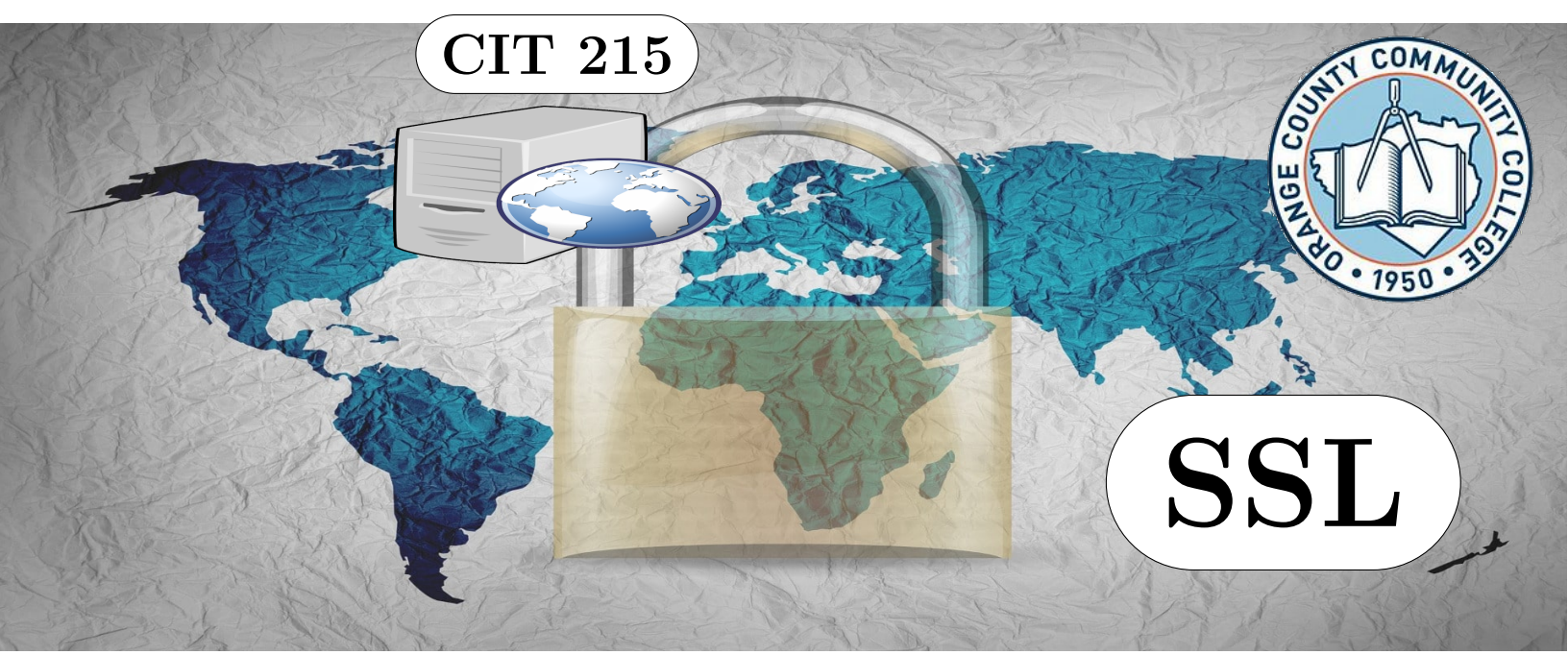
7.2.4.0 Install PHP by using Web PI

The preferred method to install PHP on a Windows or Windows Server computer is to use Web Platform Installer (Web PI).

To install PHP by using Web PI

Open a browser to the following website: Microsoft Web Platform Installer 3.0. Click Download It Now, and then click Run. At the top of the Web Platform

Installer window, click Products. Click Frameworks, and then select the current version of PHP. Click Install. The Web Platform Installation page displays the version of PHP and its dependencies that will be installed. Click I Accept. Web PI installs the PHP packages. Click Finish.



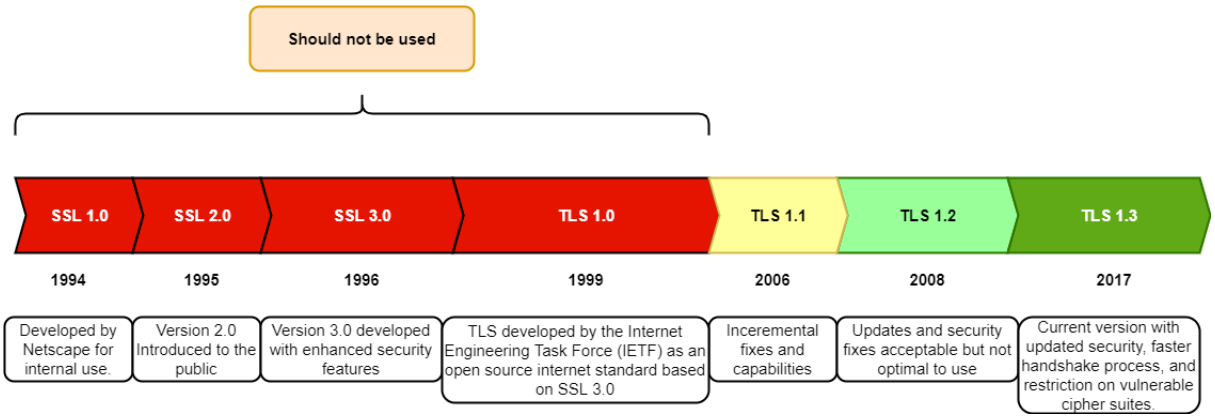
https://www.feistyduck.com/library/openssl-cookbook/online/
Other recommended books.
Bulletproof TLS and PKI, Second Edition: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications 2nd Edition

What is SSL?

SSL stands for **"secure sockets layer"** and is a cryptographic protocol used to secure data between two machines through encryption. Without SSL in place, anyone would be able to monitor all the personal information you send to a server in plain text, including passwords and credit card numbers, and easily steal it. That's why all websites today should have a digital certificate for SSL encryption installed.

TLS (Transport Layer Security)

It is just an updated, more secure, version of SSL. Most TLS certs are still refered to as SSL because it is a more commonly used term, but when you are buying SSL you are actually buying the most up to date TLS certificates with the option of ECC, RSA or DSA encryption.



What is an SSL Certificate?

SSL certificates assign a specific cryptographic key to your particular organization's details (e.g. a domain name) to secure your website when users are required to submit any sensitive information, from filling out forms to logging into their account.

When an SSL certificate is active on your server, your URL will change from HTTP to HTTPS and feature a grey padlock next to it in all web browsers, so your visitors will be able to recognize the visual cues to know your website can be trusted. Having a certificate showing on your site in the address bar is similar to registered trademarks in this way.

In addition, search engines like Google now incentive website owners to install SSL certificates by boosting their web rankings.

How Do SSL Certificates Work?

In essence, SSL certificates are data files used to verify the secure connection between your website and web server based on public key cryptography.

Your private key is known to your server and can be used to encrypt any message sent to your website. But this message can only be decrypted using the public key contained in your certificate.

SSL certificates are generated and issued by a trusted Certificate Authority (CA). There is more than one type of certificate, and all are based on their validation levels.

Here are the six types of SSL Certificates to consider:

Extended Validation Certificates (EV SSL)
Organization Validated Certificates (OV SSL)
Domain Validated Certificates (DV SSL)
Wildcard SSL Certificate
Multi-Domain SSL Certificate (MDC)
Unified Communications Certificate (UCC)
You might be already asking yourself, "What type of SSL certificate do I need?" To pick one, let's discuss the essential differences between them first.

8.0.1 Extended Validation Certificates (EV SSL)

The highest-ranking and most expensive SSL certificate type is an Extended Validation Certificate.

Setting up an EV certificate requires the website owner to go through a standardized identity verification process to confirm they have exclusive rights to their domain.

Use Cases for EV SSL Certificates

Since EV certificates are expensive and require an extended verification pro-

cess, they are mostly used by high-profile websites that require a lot of personal information from their visitors or frequently collect online payments (e.g. banks or medical providers).

An EV certificate provides the highest level of digital identity assurance by verifying the legal identity of a website owner.

According to the Guidelines for the Issuance and Management of Extended Validation Certificates the primary function of an EV certificate is to:
"Identify the legal entity that controls a Web site: Provide a reasonable assurance to the user of an Internet browser that the Web site the user is accessing is controlled by a specific legal entity identified in the EV Certificate by name, address of Place of Business, Jurisdiction of Incorporation or Registration and Registration Number or other disambiguating information."
In most cases these are no longer in use at least not widely.

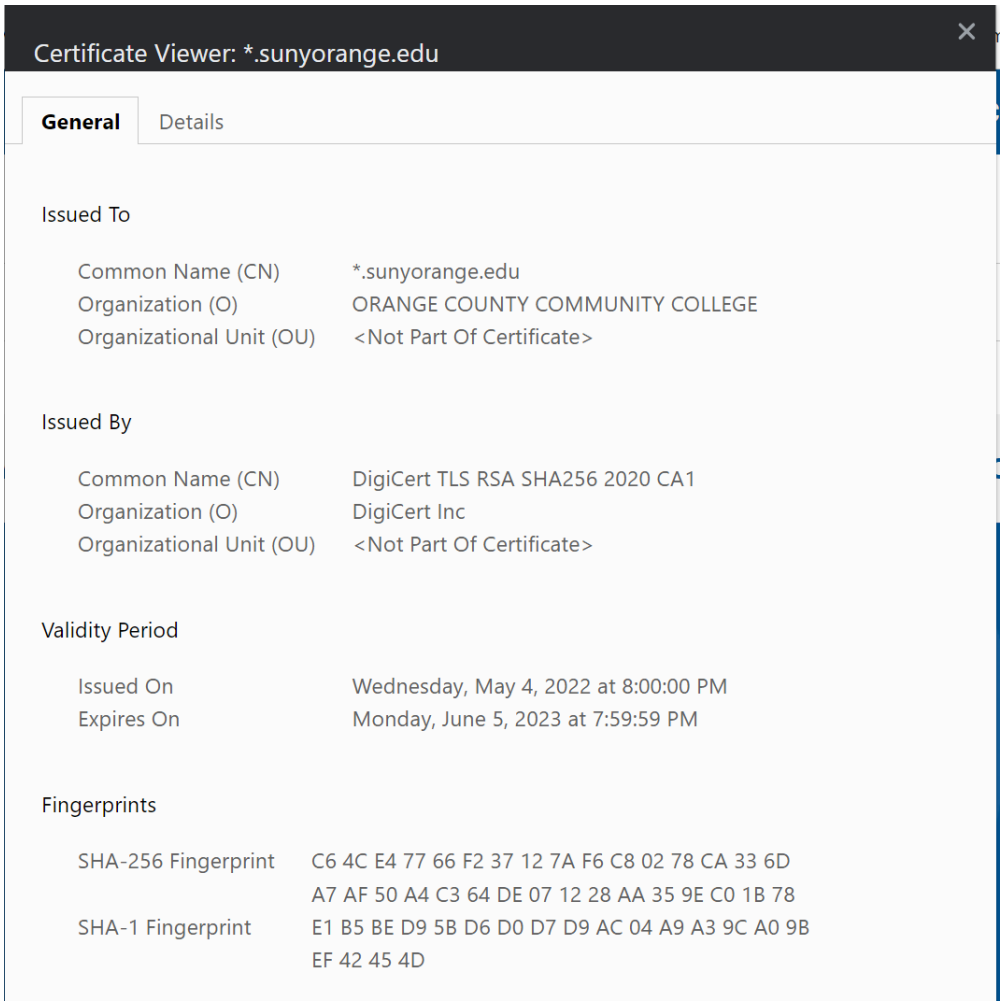
8.0.2 Organization Validated Certificates (OV SSL)

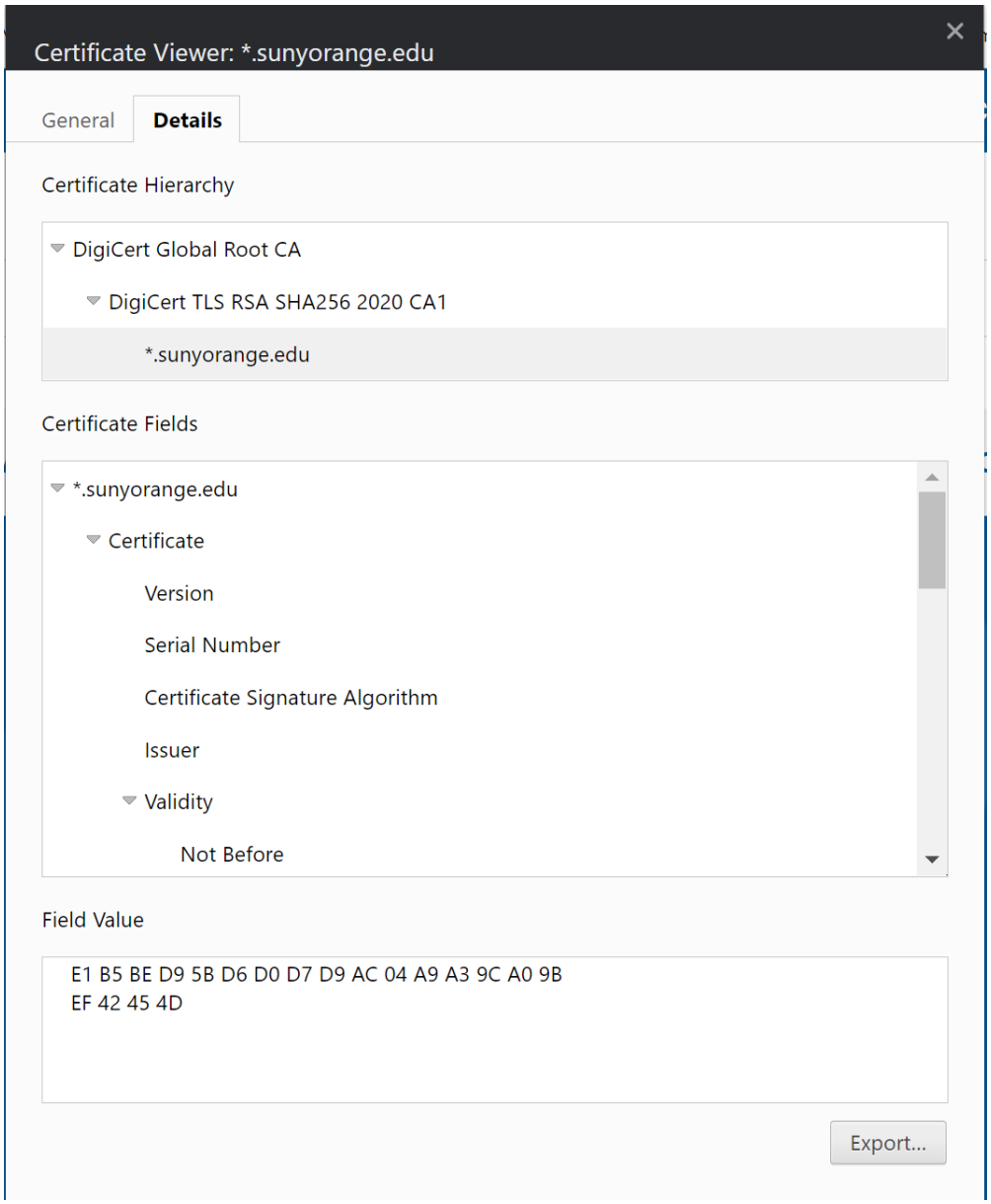
The Organization Validation SSL certificate's primary purpose is to encrypt sensitive information during transactions. The OV certificate has a high assurance, similar to the EV certificate, and is also used to validate business credibility.

OV SSL certificates are the second-highest in price. To obtain them, website owners need to complete a substantial validation process administered by a Certification Authority, which investigates the website owner to see if they have the right to their specific domain name.

Use Cases for OV SSL Certificates

OV certificates are often required for commercial and public-facing websites that collect and store their customers' information.


Certificate Viewer: *.sunyorange.edu
General
Details
Issued To
Common Name (CN) *.sunyorange.edu
Organization (O) ORANGE COUNTY COMMUNITY COLLEGE
Organizational Unit (OU) <Not Part Of Certificate>
Issued By
Common Name (CN) DigiCert TLS RSA SHA256 2020 CA1
Organization (O) DigiCert Inc
Organizational Unit (OU) <Not Part Of Certificate>
Validity Period
Issued On Wednesday, May 4, 2022 at 8:00:00 PM
Expires On Monday, June 5, 2023 at 7:59:59 PM
Fingerprints
SHA-256 Fingerprint C6 4C E4 77 66 F2 37 12 7A F6 C8 02 78 CA 33 6D A7 AF 50 A4 C3 64 DE 07 12 28 AA 35 9E C0 1B 78
SHA-1 Fingerprint E1 B5 BE D9 5B D6 D0 D7 D9 AC 04 A9 A3 9C A0 9B EF 42 45 4D



8.0.3 Domain Validated Certificates (DV SSL)

Compared to other SSLs, Domain Validation SSL certificates have low assurance and minimal encryption. Hence, the validation process to obtain this certificate type is minimal. The process only requires website owners to prove domain ownership by responding to an email or phone call.

Use Cases for DV SSL Certificates

As DV certificates are one of the least expensive and fastest types to obtain,

they are often used by blogs or informational websites that don't need to provide extra assurance to their visitors.

8.0.4 Wildcard SSL Certificates

Wildcard SSL certificates are available as both OV and DV and are used to secure a base domain and unlimited subdomains. The main benefit of purchasing a wildcard certificate is that it's cheaper than buying several single-domain certificates.

Wildcard SSL certificates have an asterisk as part of their common name. The asterisk represents any valid subdomain that has the same base domain. For example, the common name can be *.example.com, which would allow this certificate to be installed for blog.example.com and account.example.com as well.

Use Cases for Wildcard SSL Certificates

Depending on the business needs, customers can purchase either OV or DV Wildcard certificates when they need encryption for multiple subdomains. This could be valuable for blogging solutions that create different subdomains for their user accounts

8.0.5 Multi-Domain SSL Certificates

Multi-Domain SSL certificates can secure up to 100 different domain names and subdomains using a single certificate, which helps save time and money. Businesses have control of the Subject Alternative Name (SAN) field to add, change, and delete any of the SANs as needed.

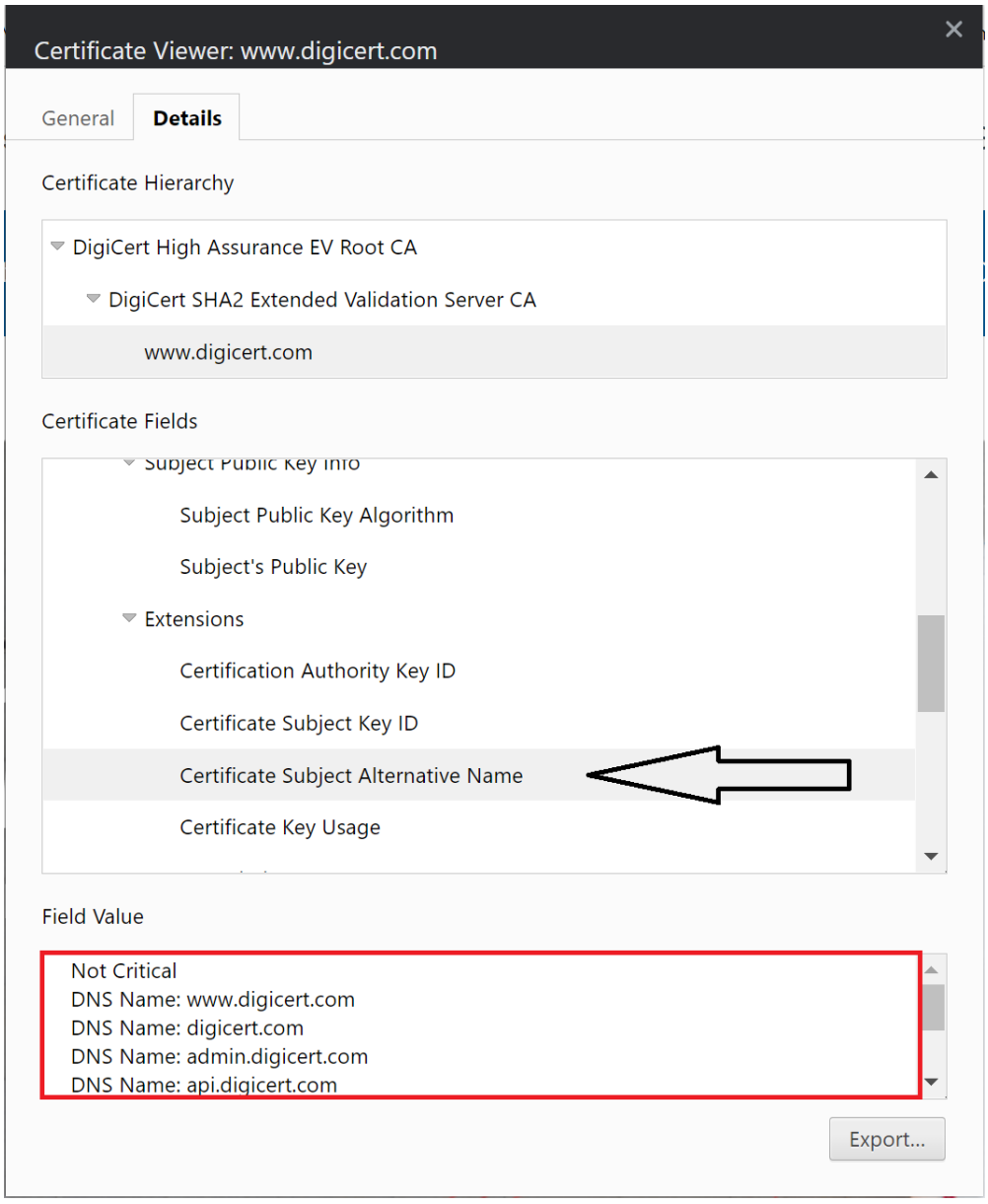
Domain Validated, Organization Validated, Extended Validated, and Wildcard certificates could be upgraded to secure multiple domains. Here are some domain name examples that can gain security with just one Multi-Domain certificate:

www.domain.com
www.domain.in
www.domain.org
domain.com

checkout.domain.com
mail.domain.com
secure.exampledomain.org
www.website.com
www.example.co.uk

Use Cases for Multi-Domain SSL Certificates

Multi-Domain SSL certificates are often used by companies that have representations in different jurisdictions, as well as international conglomerates that need to secure different top-level domain names.



8.0.6 Unified Communications Certificates (UCC)

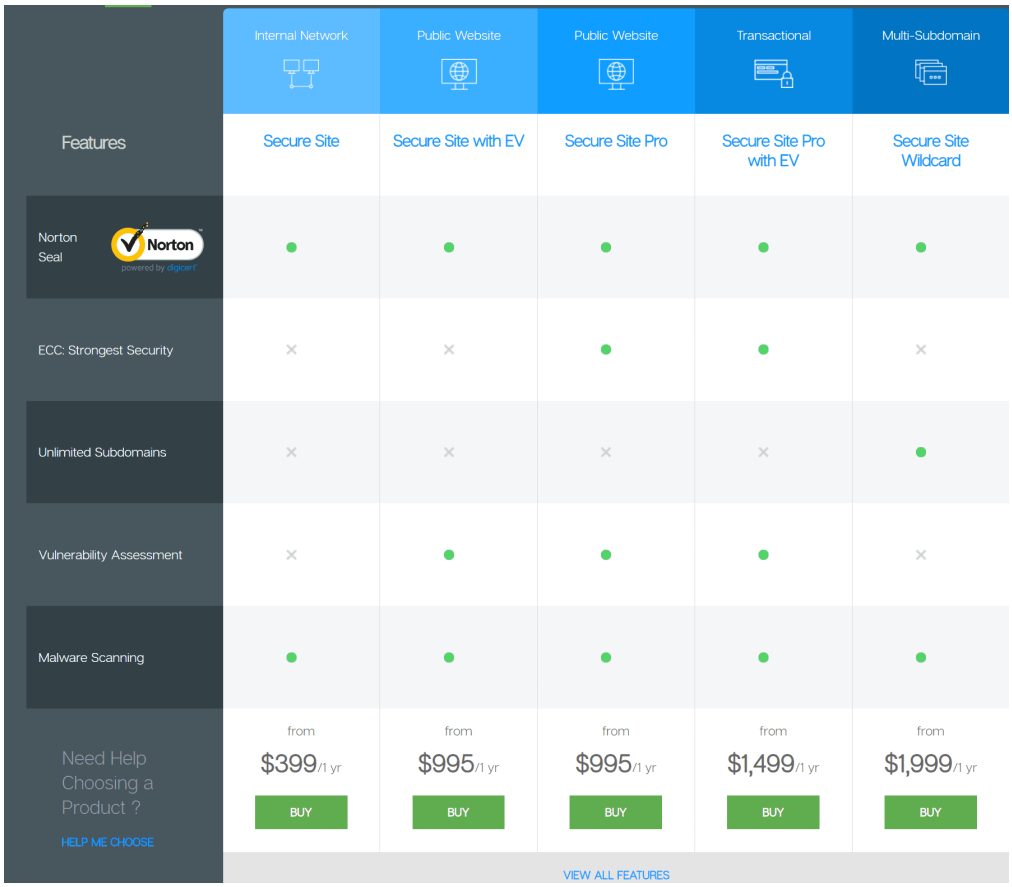
Unified Communications Certificates (UCC) are also considered Multi-Domain SSL Certificates and have the same benefits. UCCs can be used as EV SSL certificates.

Use Cases for Unified Communications Certificates

UCCs were initially designed to secure Microsoft Exchange and Live Communications servers. However, today, any website owner can use them to encrypt multiple domains with a single certificate.

8.0.7 cost of SSL certificate

Cost can range from free from companies like : https://letsencrypt.org/ to several thousand dollars a year.
These are prices from digicert.com listed on their website. (late 2022)



8.0.8 Self signed certificates

problems with self signed certs

Note: A self-signed certificate will encrypt communication between your server and any clients. However, because it is not signed by any of the trusted certificate authorities included with web browsers and operating systems, users cannot use the certificate to validate the identity of your server automatically. As a result, your users will see a security error when visiting your site.

Because of this limitation, self-signed certificates are not appropriate for a production environment serving the public. They are typically used for testing, or for securing non-critical services used by a single user or a small group of users that can establish trust in the certificate's validity through alternate communication channels.

For a more production-ready certificate solution, check out https://letsencrypt.org/, a free certificate authority.

8.0.9 Apache SSL

8.0.9.0 Self signed certificate

we can use mozilla ssl configuration generator to create a secure conf for our webserver.
https://ssl-config.mozilla.org/

Commands

```
occc@occc-VirtualBox:~$ sudo su -
[sudo] password for occc:
root@occc-VirtualBox:~# apache2 -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2022-09-30T04:09:50
root@occc-VirtualBox:~# openssl version
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
root@occc-VirtualBox:~#
```

mozilla generated file

```
1  # generated 2022-11-27, Mozilla Guideline v5.6, ↙
     ↪  Apache 2.4.52, OpenSSL 3.0.2, intermediate ↙
     ↪  configuration
2  # https://ssl-config.mozilla.org/#server=apache&↙
     ↪  version=2.4.52&config=intermediate&openssl↙
     ↪  =3.0.2&guideline=5.6
3
4  # this configuration requires mod_ssl, ↙
     ↪  mod_socache_shmcb, mod_rewrite, and ↙
     ↪  mod_headers
5  <VirtualHost *:80>
6    RewriteEngine On
7    RewriteCond %{REQUEST_URI} !^/\.well\-known/acme↙
       ↪  \-challenge/
8    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,↙
       ↪  L]
```

```
</VirtualHost>

<VirtualHost *:443>
  SSLEngine on

  # curl https://ssl-config.mozilla.org/ffdhe2048.txt >> /path/to/signed_cert_and_intermediate_certs_and_dhparams
  SSLCertificateFile      /path/to/signed_cert_and_intermediate_certs_and_dhparams
  SSLCertificateKeyFile   /path/to/private_key

  # enable HTTP/2, if available
  Protocols h2 http/1.1

  # HTTP Strict Transport Security (mod_headers is required) (63072000 seconds)
  Header always set Strict-Transport-Security "max-age=63072000"
</VirtualHost>

# intermediate configuration
SSLProtocol             all -SSLv3 -TLSv1 -TLSv1.1
SSLCipherSuite          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384
SSLHonorCipherOrder     off
SSLSessionTickets       off

SSLUseStapling On
SSLStaplingCache "shmcb:logs/ssl_stapling(32768)"
```

Make a self signed certificate for virtual host one

Commands

```
sudo openssl req -x509 -nodes -days 365 -
newkey rsa:2048 -keyout /etc/ssl/private/cit215.key -
out /etc/ssl/certs/cit215.crt
```

The openssl command explained:

- **req -x509** You're using this option to create a self-signed certificate instead of generating a CSR.
- **-nodes** By specifying this option, you're telling openssl to avoid encrypting the private key with a passphrase. This is because you don't want to keep entering a password manually every time your server restarts. Apache should read this file without your intervention.
- **-days n** This is the number of days that you want the certificate to remain valid. For instance 365 days.
- **-newkey rsa:2048** You declare an explicit key size with this option. The smallest acceptable file is 512 bits but a larger value of 2048 offers more security.
- **-keyout filename** This is the file name where you're saving your newly created private key.
- **-out filename** Here, you specify an output *.crt file to store your signed certificate.

Commands

```
root@occc-VirtualBox:~# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -
keyout /etc/ssl/private/cit215.key -out /etc/ssl/certs/cit215.crt
.....+.......+..+.+...........+.+...+................+.........
......+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++*.+....+..+.........+.+.....+....+.........+.....+....+..+
+..+.+..+.......+...........+.+.............+.+...+..+.........
+.......+...+..+.........+............+...+...........++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++*.....+.
.....+....+........+...+...+.+........+............+...........
........+..........+..+..........+...+.....+..........+.....+...
.......+......+.......+........+.+......+...+......+.....+......
..++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:Middletown
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Suny Orange
Organizational Unit Name (eg, section) []:Comp Sci
Common Name (e.g. server FQDN or YOUR name) []:cit215.sunyorange.edu.local
Email Address []:cit215@sunyorange.edu.local
root@occc-VirtualBox:~#
```

Make a self signed certificate for virtual host two

Commands

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -
keyout /etc/ssl/private/main.key -out /etc/ssl/certs/main.crt
```

Commands

```
root@occc-VirtualBox:~# sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -
keyout /etc/ssl/private/main.key -out /etc/ssl/certs/main.crt
.+...+..+.........+......+...++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++*.....+.+.........+...+.........+.....+.++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++*.+..........+.........+............+.....+.+...
......+.....+.+.....+.+.....+.+..+.......+.....+..............+..............+.
...+..+..........+...+..........+......+...+.+...+...+........+....+..........+
.......+........+....+...+.....+.........+....+.....+.......+...+......+.....+...
..........+.....+...+.........+....+......+........+..........+.....+...+...+....
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New York
Locality Name (eg, city) []:Middletown
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Suny Orange
Organizational Unit Name (eg, section) []:Comp Sci
Common Name (e.g. server FQDN or YOUR name) []:main.sunyorange.edu.local
Email Address []:root@sunyorange.edu.local
root@occc-VirtualBox:~#
```

8.0.9.0 Forward Secrecy & Diffie Hellman Ephemeral Parameters

The concept of forward secrecy is simple: client and server negotiate a key that never hits the wire, and is destroyed at the end of the session. The RSA private from the server is used to sign a Diffie- Hellman key exchange between the client and the server. The pre-master key obtained from the Diffie-Hellman handshake is then used for encryption. Since the pre-master key is specific to a connection between a client and a server, and used only for a limited amount of time, it is called Ephemeral.

With Forward Secrecy, if an attacker gets a hold of the server's private key, it will not be able to decrypt past communications. The private key is only used to sign the DH handshake, which does not reveal the pre- master key. Diffie-Hellman ensures that the pre-master keys never leave the client and the server, and cannot be intercepted by a MITM.

Apache prior to version 2.4.7 and all versions of Nginx as of 1.4.4 rely on OpenSSL for input parameters to Diffie-Hellman (DH). Unfortunately, this means

that Ephemeral Diffie-Hellman (DHE) will use OpenSSL's defaults, which include a 1024-bit key for the key-exchange. Since we're using a 2048-bit certificate, DHE clients will use a weaker key-exchange than non-ephemeral DH clients.

For Apache, there is no fix except to upgrade to 2.4.7 or later. With that version, Apache automatically selects a stronger key.

If you have Apache 2.4.8 or later and OpenSSL 1.0.2 or later, you can generate and specify your DH params file:

Generate DHparams

Commands

```
#Generate the parameters
openssl dhparam -out /etc/ssl/certsdhparam.pem 4096

# Add the following to your Apache config.
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/certsdhparam.pem"
```

Commands

```
openssl dhparam -out /etc/ssl/certsdhparam.pem 4096
Generating DH parameters, 4096 bit long safe prime
.........................+.........................
.................................................
.....................................+...........
.................................................
.....................................+.........
........+.....................................
.................................................
................................................+
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++
root@occc-VirtualBox:~#
```

make sure that the certs were created

Commands

```
root@occc-VirtualBox:/etc/ssl# ls -l private/
total 12
-rw------- 1 root root     1704 Nov 19 20:23 cit215.key
-rw------- 1 root root     1704 Nov 19 20:33 main.key
-rw-r----- 1 root ssl-cert 1704 Apr  2  2022 ssl-cert-
snakeoil.key
root@occc-VirtualBox:/etc/ssl# ls -l certs/ | grep main
-rw-r--r-- 1 root root   1529 Nov 19 20:34 main.crt
root@occc-VirtualBox:/etc/ssl# ls -l certs/ | grep cit215
-rw-r--r-- 1 root root   1537 Nov 19 20:24 cit215.crt
root@occc-VirtualBox:/etc/ssl#
```

Enable SSL on the Virtual Host File

Apache maintains a default virtual host file to handle SSL traffic under the **/etc/apache2/sites-available directory** named **default-ssl.conf**. In order for the webserver to encrypt data with your certificate, you'll make some configuration changes in this file.

default-ssl.conf

```
1  <IfModule mod_ssl.c>
2  <VirtualHost _default_:443>
3  ServerAdmin webmaster@localhost
4
5  DocumentRoot /var/www/html
6
7  # Available loglevels: trace8, ..., trace1, debug, ↙
     ↳  info, notice, warn,
8  # error, crit, alert, emerg.
9  # It is also possible to configure the loglevel for↙
     ↳   particular
10 # modules, e.g.
11 #LogLevel info ssl:warn
```

```
12
13 ErrorLog ${APACHE_LOG_DIR}/error.log
14 CustomLog ${APACHE_LOG_DIR}/access.log combined
15
16 # For most configuration files from conf-available/, which are
17 # enabled or disabled at a global level, it is possible to
18 # include a line for only one particular virtual host. For example the
19 # following line enables the CGI configuration for this host only
20 # after it has been globally disabled with "a2disconf".
21 #Include conf-available/serve-cgi-bin.conf
22
23 #   SSL Engine Switch:
24 #   Enable/Disable SSL for this virtual host.
25 SSLEngine on
26
27 #   A self-signed (snakeoil) certificate can be created by installing
28 #   the ssl-cert package. See
29 #   /usr/share/doc/apache2/README.Debian.gz for more info.
30 #   If both key and certificate are stored in the same file, only the
31 #   SSLCertificateFile directive is needed.
32 SSLCertificateFile  /etc/ssl/certs/ssl-cert-snakeoil.pem
33 SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
34
35 #   Server Certificate Chain:
36 #   Point SSLCertificateChainFile at a file containing the
37 #   concatenation of PEM encoded CA certificates which form the
```

```
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convinience.
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

#   Certificate Authority (CA):
#   Set the CA certificate verification path where to find CA
#   certificates for client authentication or alternatively one
#   huge file containing all of them (file must be PEM encoded)
#   Note: Inside SSLCACertificatePath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCACertificatePath /etc/ssl/certs/
#SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

#   Certificate Revocation Lists (CRL):
#   Set the CA revocation path where to find CA CRLs for client
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after
```

```
    ↪   changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle↙
    ↪  .crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.↙
    ↪    Types are
#   none, optional, require and optional_no_ca.   ↙
    ↪  Depth is a
#   number which specifies how deeply to verify the↙
    ↪   certificate
#   issuer chain before deciding the certificate is↙
    ↪   not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#    Translate the client X.509 into a Basic ↙
    ↪  Authorisation.  This means that
#    the standard Auth/DBMAuth methods can be used ↙
    ↪  for access control.  The
#    user name is the `one line' version of the ↙
    ↪  client's X.509 certificate.
#    Note that no password is obtained from the ↙
    ↪  user. Every entry in the user
#    file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#    This exports two additional environment ↙
    ↪  variables: SSL_CLIENT_CERT and
#    SSL_SERVER_CERT. These contain the PEM-encoded↙
    ↪   certificates of the
#    server (always existing) and the client (only ↙
    ↪  existing when client
#    authentication is used). This can be used to ↙
    ↪  import the certificates
```

```
85 #    into CGI scripts.
86 #   o StdEnvVars:
87 #    This exports the standard SSL/TLS related `SSL_*' environment variables.
88 #    Per default this exportation is switched off for performance reasons,
89 #    because the extraction step is an expensive operation and is usually
90 #    useless for serving static content. So one usually enables the
91 #    exportation for CGI and SSI requests only.
92 #   o OptRenegotiate:
93 #    This enables optimized SSL connection renegotiation handling when SSL
94 #    directives are used in per-directory context.
95 #SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
96 <FilesMatch "\.(cgi|shtml|phtml|php)$">
97 SSLOptions +StdEnvVars
98 </FilesMatch>
99 <Directory /usr/lib/cgi-bin>
100 SSLOptions +StdEnvVars
101 </Directory>
102
103 #   SSL Protocol Adjustments:
104 #   The safe and default but still SSL/TLS standard compliant shutdown
105 #   approach is that mod_ssl sends the close notify alert but doesn't wait for
106 #   the close notify alert from client. When you need a different shutdown
107 #   approach you can use one of the following variables:
108 #   o ssl-unclean-shutdown:
109 #    This forces an unclean shutdown when the connection is closed, i.e. no
110 #    SSL close notify alert is send or allowed to received.  This violates
```

```
#    the SSL/TLS standard but is needed for some brain-dead browsers. Use
#    this when you receive I/O errors because of the standard approach where
#    mod_ssl sends the close notify alert.
#  o ssl-accurate-shutdown:
#    This forces an accurate shutdown when the connection is closed, i.e. a
#    SSL close notify alert is send and mod_ssl waits for the close notify
#    alert of the client. This is 100% SSL/TLS standard compliant, but in
#    practice often causes hanging connections with brain-dead browsers. Use
#    this only for browsers where you know that their SSL implementation
#    works correctly.
#  Notice: Most problems of broken clients are also related to the HTTP
#  keep-alive facility, so you usually additionally want to disable
#  keep-alive for those clients, too. Use variable "nokeepalive" for this.
#  Similarly, one has to force some clients to use HTTP/1.0 to workaround
#  their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
#  "force-response-1.0" for this.
# BrowserMatch "MSIE [2-6]" \
#       nokeepalive ssl-unclean-shutdown \
#       downgrade-1.0 force-response-1.0

</VirtualHost>
<VirtualHost 10.88.0.2:443>
SSLEngine on
DocumentRoot "/webroot/main"
ServerName www.sunyorange.edu.local
ServerAdmin root@sunyorange.edu.local
```

```
SSLCertificateFile    /etc/ssl/certs/main.crt
SSLCertificateKeyFile /etc/ssl/private/main.key
<Directory /webroot/main/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>


# Other directives here
</VirtualHost>

<VirtualHost 10.88.0.2:443>
SSLEngine on
DocumentRoot "/webroot/cit215"
ServerName cit215.sunyorange.edu.local
ServerAdmin cit215@sunyorange.edu.local
SSLCertificateFile    /etc/ssl/certs/cit215.crt
SSLCertificateKeyFile /etc/ssl/private/cit215.key
<Directory /webroot/cit215/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>


# Other directives here
</VirtualHost>



</IfModule>
```

Use the Apache **a2enmod** command to enable the **ssl** module.

```
sudo a2enmod ssl
```

copy the default-ssl.conf to sites-enabled directory

ln -s /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-enabled/

Commands

```
root@occc-VirtualBox:/etc/apache2/sites-available# sudo a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-
signed certificates.
To activate the new configuration, you need to run:
systemctl restart apache2
root@occc-VirtualBox:/etc/apache2/sites-available#
```

Verify that config files are ok

Commands

```
root@occc-VirtualBox:/etc/apache2/sites-
available# apachectl configtest
Syntax OK
root@occc-VirtualBox:/etc/apache2/sites-available#
```

restart web server

systemctl restart apache2

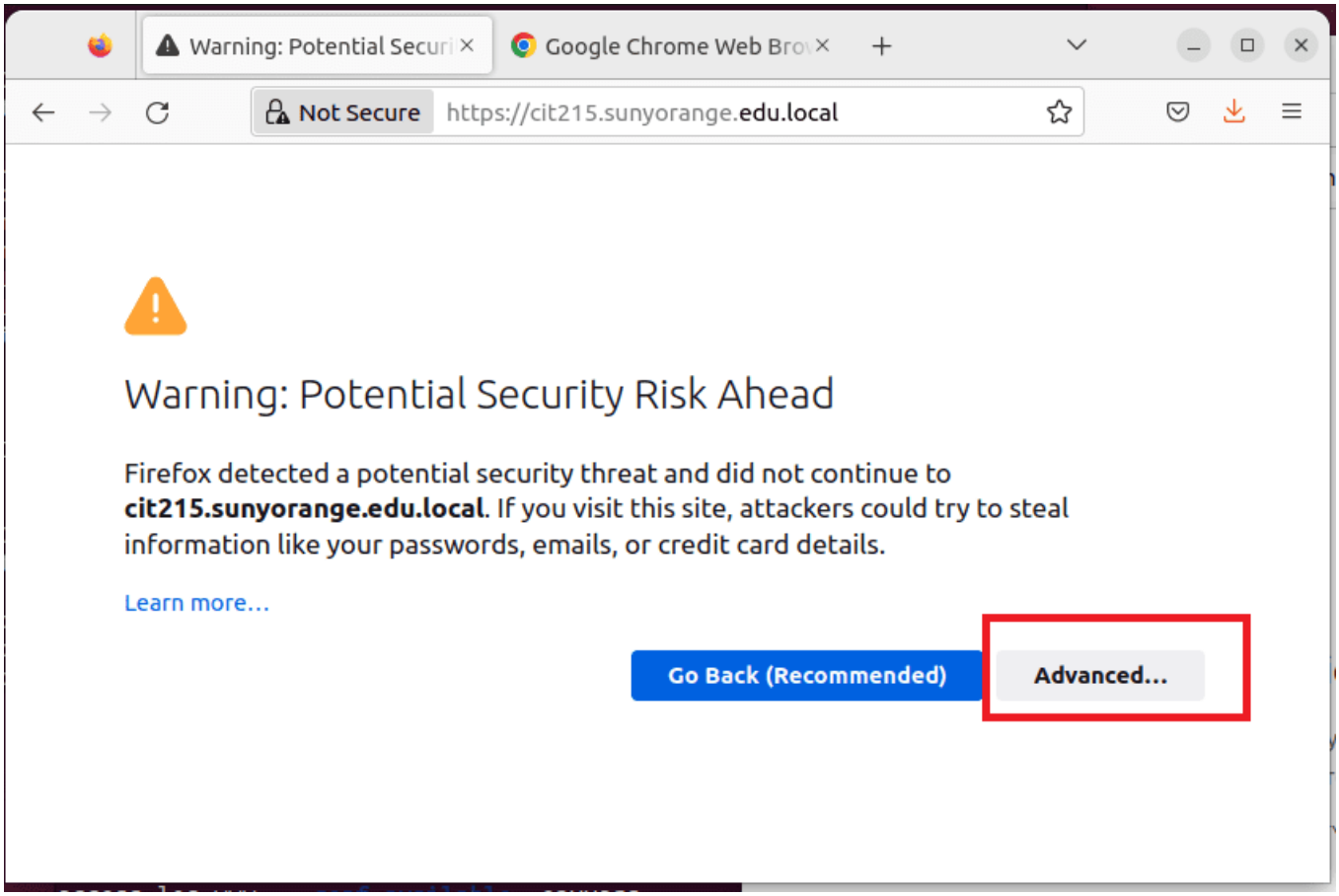
Verify that firewall is open for port 443

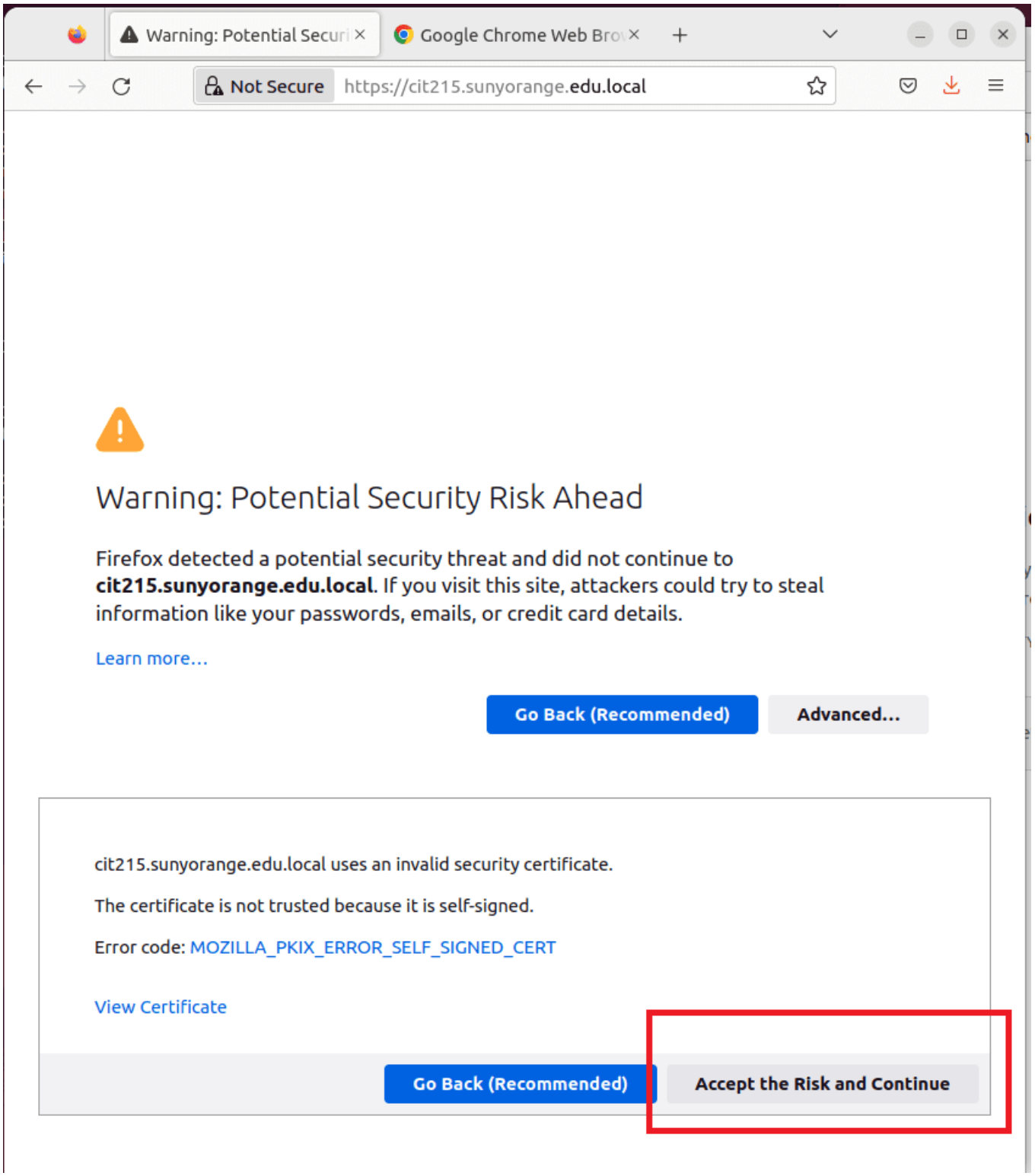
Commands

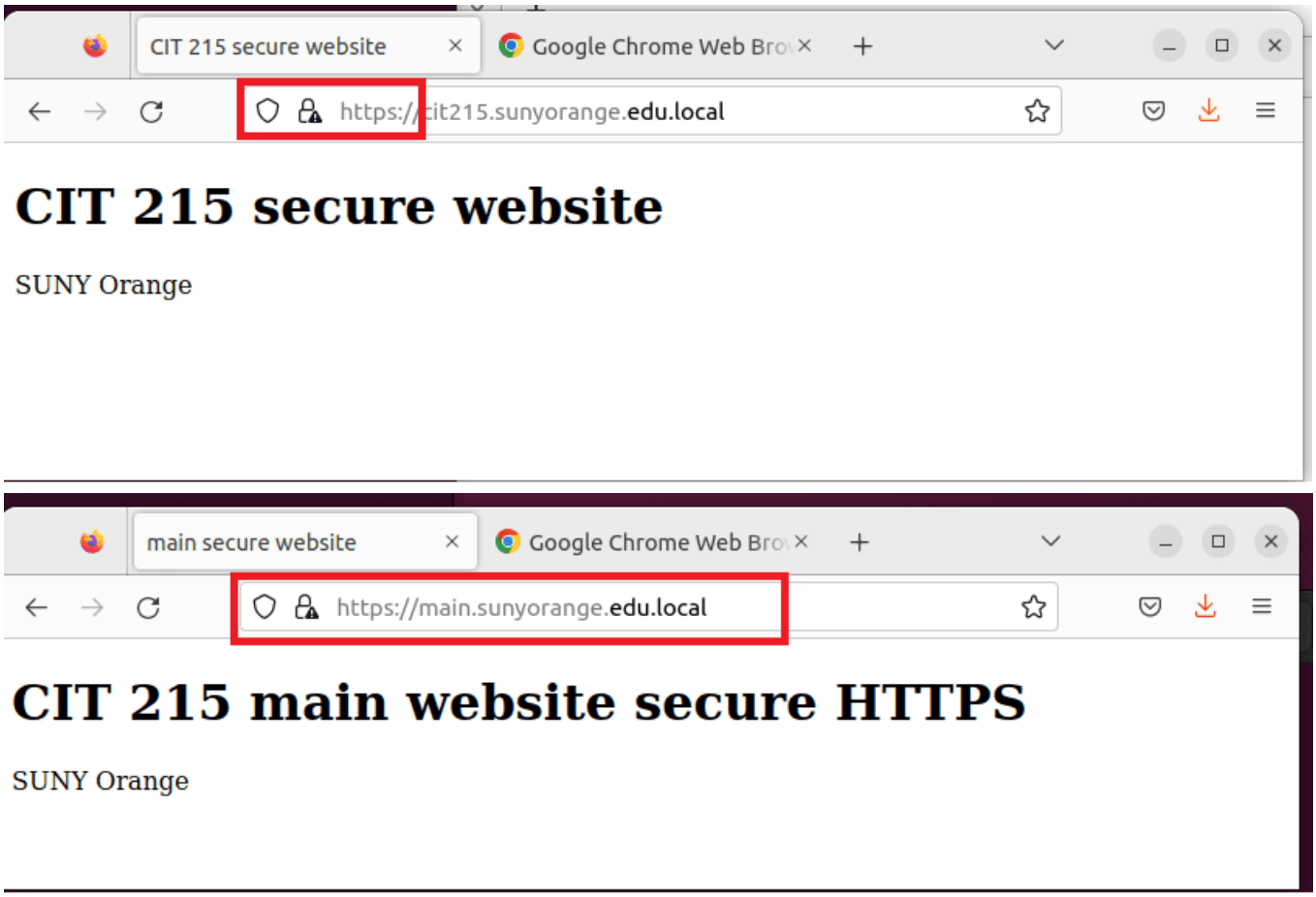
```
root@occc-VirtualBox:/etc/apache2/sites-available# ufw status
Status: active

To                         Action      From
--                         ------      ----
Bind9                      ALLOW       Anywhere
Apache Full                ALLOW       Anywhere
OpenSSH                    ALLOW       Anywhere
Bind9 (v6)                 ALLOW       Anywhere (v6)
Apache Full (v6)           ALLOW       Anywhere (v6)
OpenSSH (v6)               ALLOW       Anywhere (v6)

root@occc-VirtualBox:/etc/apache2/sites-available#
```

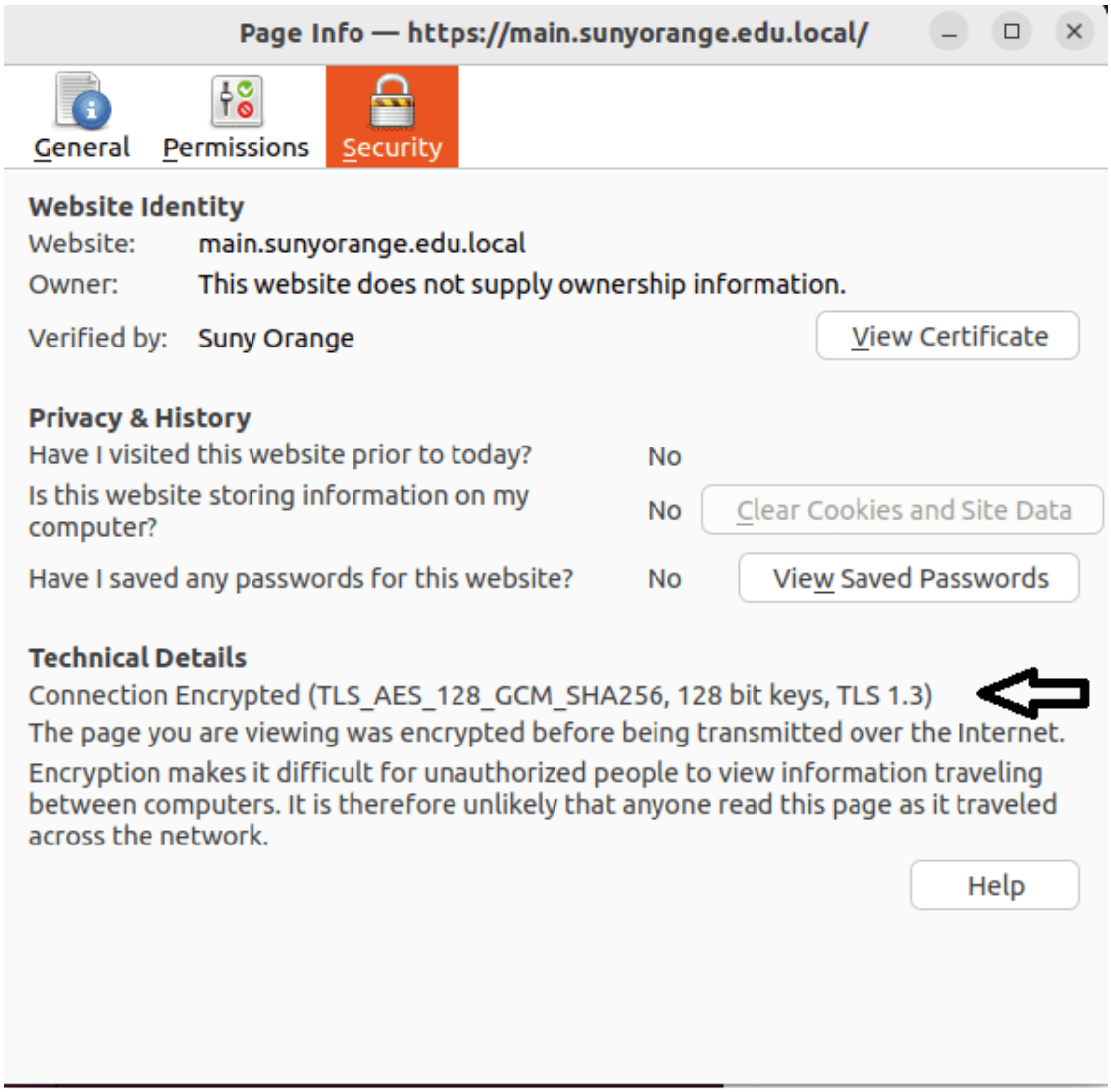
If you dont see apache full:

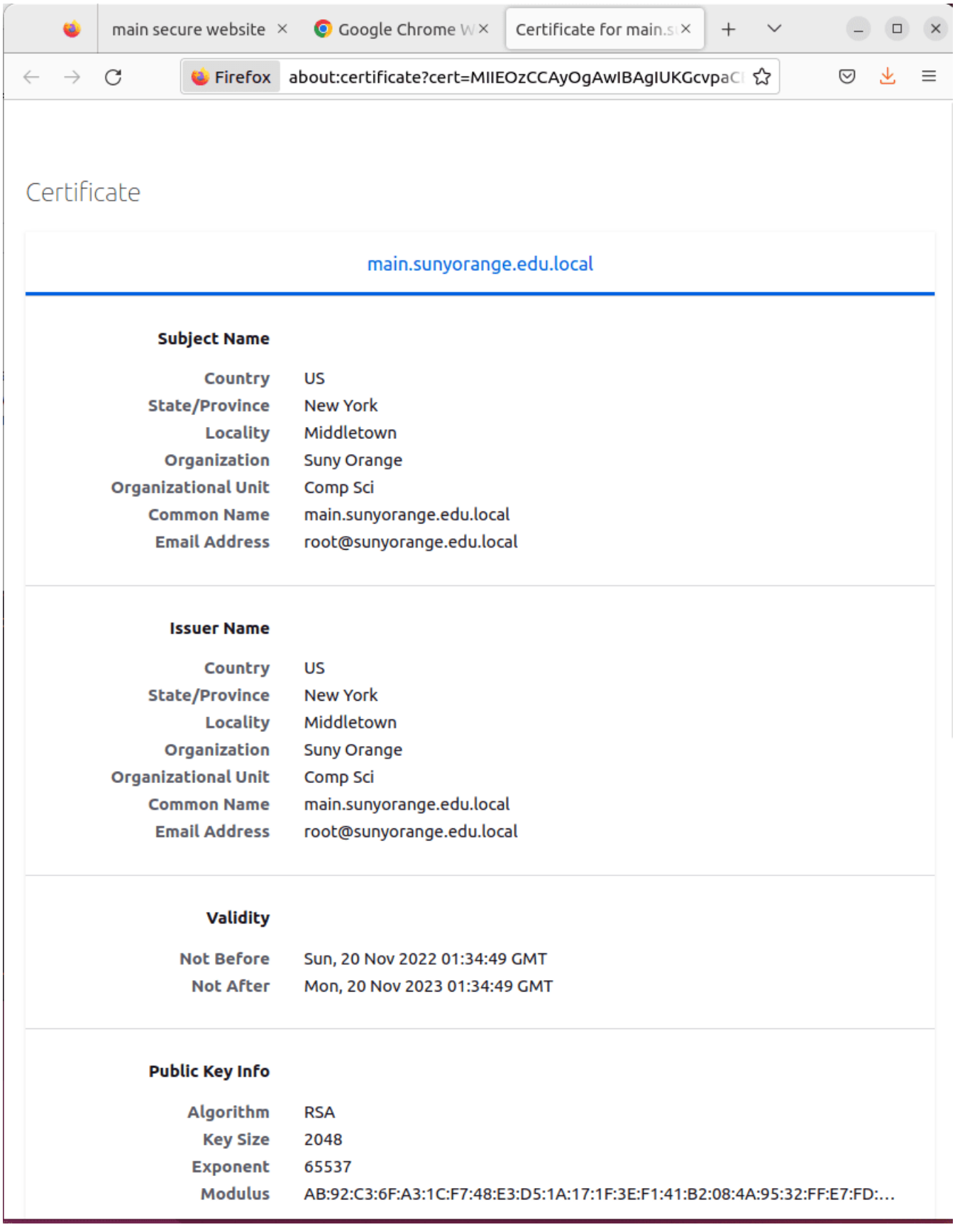
```
ufw allow 'Apache Full'
```




Warning: Potential Securi
Google Chrome Web Bro
Not Secure
https://cit215.sunyorange.edu.local
Warning: Potential Security Risk Ahead
Firefox detected a potential security threat and did not continue to cit215.sunyorange.edu.local. If you visit this site, attackers could try to steal information like your passwords, emails, or credit card details.
Learn more...
Go Back (Recommended)
Advanced...
cit215.sunyorange.edu.local uses an invalid security certificate.
The certificate is not trusted because it is self-signed.
Error code: MOZILLA_PKIX_ERROR_SELF_SIGNED_CERT
View Certificate
Go Back (Recommended)
Accept the Risk and Continue


CIT 215 secure website
Google Chrome Web Bro
https://cit215.sunyorange.edu.local
CIT 215 secure website
SUNY Orange
main secure website
Google Chrome Web Bro
https://main.sunyorange.edu.local
CIT 215 main website secure HTTPS
SUNY Orange


Page Info — https://main.sunyorange.edu.local/
General
Permissions
Security
Website Identity
Website: main.sunyorange.edu.local
Owner: This website does not supply ownership information.
Verified by: Suny Orange
View Certificate
Privacy & History
Have I visited this website prior to today? No
Is this website storing information on my computer? No
Clear Cookies and Site Data
Have I saved any passwords for this website? No
View Saved Passwords
Technical Details
Connection Encrypted (TLS_AES_128_GCM_SHA256, 128 bit keys, TLS 1.3)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.
Help



Further list os SSL options for apache.
Apache SSL options

8.0.10 lets enrypt

https://letsencrypt.org/
At this time it is not possible if ever to have any CA root certificate issued for a private server that is not accessible by the internet. This has to do with the validation of ownership of the domain. On a private network we can create our own CA authority server and import out own CA root server certs into a web browser to by pass the not secure page in logging in.

8.0.11 creating a real certificate request

https://httpd.apache.org/docs/trunk/ssl/ssl_faq.html#aboutcerts

8.0.12 self signed certificates NGINX

we can use mozilla ssl configuration generator to create a secure conf for our webserver.
https://ssl-config.mozilla.org/

Commands

```
occc@occc-VirtualBox:~$ nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
occc@occc-VirtualBox:~$ openssl version
OpenSSL 3.0.2 15 Mar 2022 (Library: OpenSSL 3.0.2 15 Mar 2022)
occc@occc-VirtualBox:~$
```

mozilla generated file

```
1 # generated 2022-11-27, Mozilla Guideline v5.6, ↙
    ↪ nginx 1.18, OpenSSL 3.0.2, intermediate ↙
    ↪ configuration
2 # https://ssl-config.mozilla.org/#server=nginx&↙
    ↪ version=1.18&config=intermediate&openssl↙
    ↪ =3.0.2&guideline=5.6
3 server {
```

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

    location / {
        return 301 https://$host$request_uri;
    }
}

server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;

    ssl_certificate /path/to/signed_cert_plus_intermediates;
    ssl_certificate_key /path/to/private_key;
    ssl_session_timeout 1d;
    ssl_session_cache shared:MozSSL:10m;  # about 40000 sessions
    ssl_session_tickets off;

    # curl https://ssl-config.mozilla.org/ffdhe2048.txt > /path/to/dhparam
    ssl_dhparam /path/to/dhparam;

    # intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
    ssl_prefer_server_ciphers off;

    # HSTS (ngx_http_headers_module is required) (63072000 seconds)
    add_header Strict-Transport-Security "max-age=63072000" always;
```

```
32
33      # OCSP stapling
34      ssl_stapling on;
35      ssl_stapling_verify on;
36
37      # verify chain of trust of OCSP response using ↙
          ↳  Root CA and Intermediate certs
38      ssl_trusted_certificate /path/to/↙
          ↳  root_CA_cert_plus_intermediates;
39
40      # replace with the IP address of your resolver
41      resolver 127.0.0.1;
42  }
```

8.0.12.0 OCSP Stapling

OCSP or **Online Certificate Status Protocol** is an internet protocol that checks the validity status of a certificate in real-time. It is an alternative to CRL or Certificate Revocation Lists. It is described in RFC 2560 - http://datatracker.ietf.org/doc/rfc2560/

OCSP is a real-time check of the status of a certificate and is fundamental in the design of Extended Validation SSL certificates.

When a user makes an https:// connection with your web server, their browser normally performs an OCSP check with the CA that issued the SSL certificate to confirm that the certificate has not been revoked. In some cases, this may create a momentary delay in the SSL handshake.

OCSP Stapling improves performance by positioning a digitally-signed and time-stamped version of the OCSP response directly on the webserver. This stapled OCSP response is then refreshed at predefined intervals set by the CA. The stapled OCSP response allows the web server to include the OCSP response within the initial SSL handshake, without the need for the user to make a separate external connection to the CA.

OCSP Stapling is outlined in RFC 6066 - http://datatracker.ietf.org/doc/rfc6066/

Note: When enabling and/or configuring OCSP Stapling on your servers, keep in mind that the OCSP request from your server to the CA must be allowed access through your firewall.

Advantages

- OCSP Stapling improves the connection speed of the SSL handshake by combining two requests into one. This cuts down on the amount of time it takes to load an encrypted webpage.

- OCSP Stapling helps maintain the privacy of the end user as no connection is made to the CRL for the OCSP request. Rather than see which websites a user has visited, the CA will only see OCSP requests from the web site and not its users.

- There are scenarios where a computer has to connect to a portal or hotspot access the internet, but it cannot verify the OCSP check (as access to the iInternet hasn't been granted yet). In these cases, OCSP Stapling helps, as the OCSP status is provided from the hotspot or portal.

Disadvantages

- Support for OCSP Stapling is not yet supported by all browsers. If either the browser or the web server do not support or have OCSP Stapling enabled, then it simply is not used and validity status lookup will automatically revert to OCSP checking directly with the CA.

HSTS

In order to achieve the best performance and be able to consume benefits of HTTP2 it is mandatory to use TLS. HSTS is a feature which allows a server to tell clients that they should only use secure protocol (HTTPS) in order to communicate with it.

When a (complying) browser receives HSTS header it will not try to contact the server using HTTP for a specified period of time.

enable hsts

```
add_header Strict-Transport-Security "max-age=31536000" always;
```

If you want to include all subdomains as well add the following line too:

enable hsts subdomains

```
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains" always;
```

Optimise SSL session cache

Creating a cache of TLS connection parameters reduces the number of handshakes and thus can improve the performance of your application. Caching is configured using ssl_session_cache directive. Default, "built-in" session cache is not optimal as it can be used by only one worker process and can cause memory fragmentation. It is much better to use shared cache.

Another parameter that effects number of handshakes that happen throughout lifetime of a server is ssl_session_timeout. By default it is set to 5 minutes. You should set it to something like 4hrs. Doing this will require you to increase the size of cache (as more information will need to be stored in it).

As a reference, a 1-MB shared cache can hold approximately 4,000 sessions.

Add the following to your nginx server config in order to set TLS session timeout to 4hrs and increase size of TLS session cache to 40MB:

ssl cache

```
ssl_session_cache shared:SSL:40m;
ssl_session_timeout 4h;
```

Enable session tickets

Session tickets are an alternative to session cache. In case of session cache information about session is stored on the server. In case of session tickets, information about session is given to the client. If a client has a session ticket, it can present it to the server and re-negotiation is not necessary. Set ssl_session_tickets directive

to on:

session tickets

ssl_session_tickets on;

Make a self signed certificate for virtual host one

Commands

```
sudo openssl req -x509 -nodes -days 365 -
newkey rsa:2048 -keyout /etc/ssl/private/cit215.key -
out /etc/ssl/certs/cit215.crt
```

Generate DHparams

Commands

```
#Generate the parameters
openssl dhparam -out /etc/ssl/certsdhparam.pem 4096

# Add the following to your Apache config.
SSLOpenSSLConfCmd DHParameters "/etc/ssl/certs/certsdhparam.pem"
```

go to our sites enabled and edit our server block. In this case the cit215.edu server block we created before.

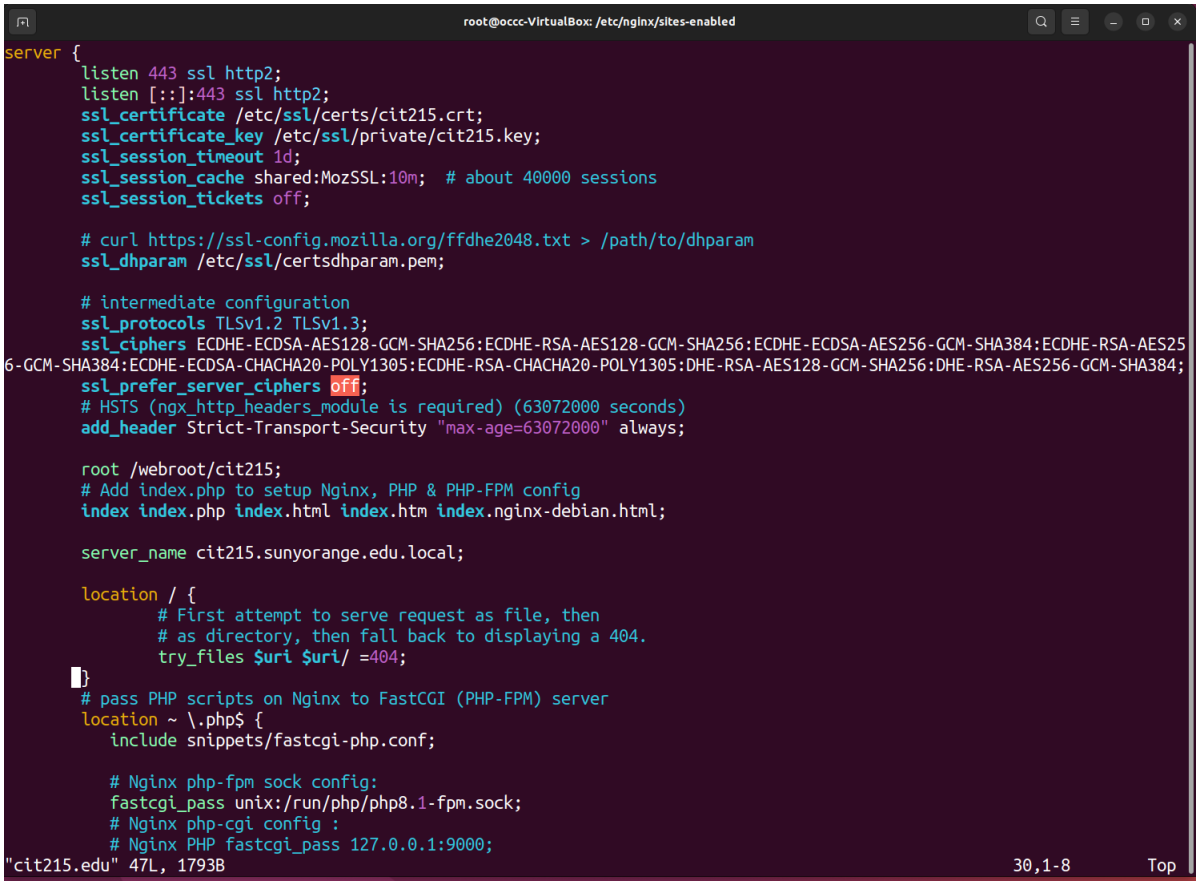
```
sudo vi cit215.edu
```

add or change the following lines.

ssl cit215

```
1 listen 443 ssl http2;
2 listen [::]:443 ssl http2;
3 ssl_certificate /etc/ssl/certs/cit215.crt;
4 ssl_certificate_key /etc/ssl/private/cit215.key;
5 ssl_session_timeout 1d;
6 ssl_session_cache shared:MozSSL:10m;  # about 40000↙
    ↳   sessions
```

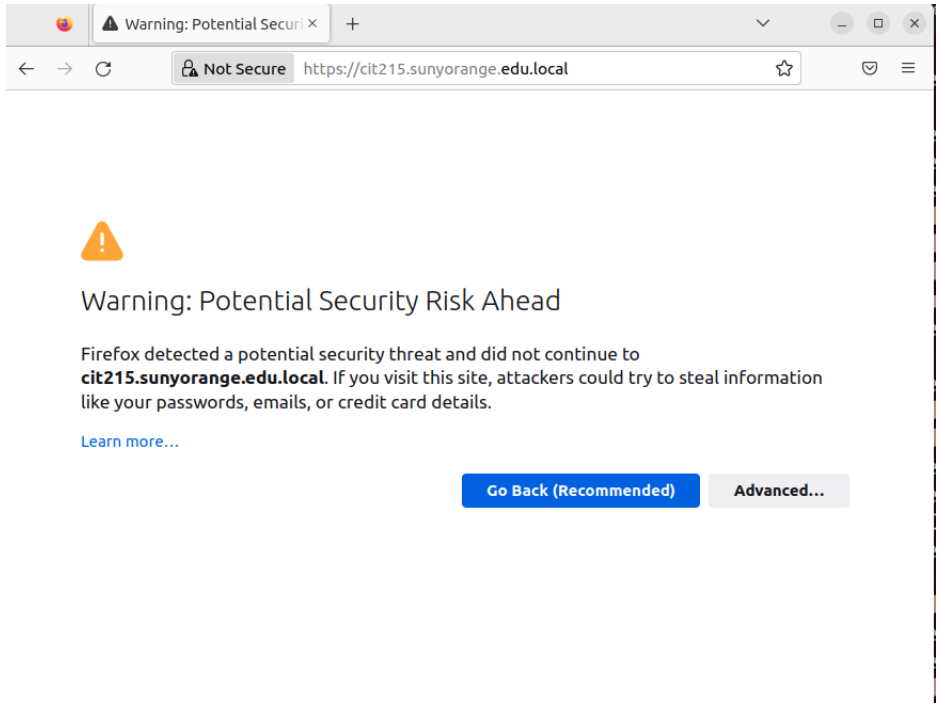
```
ssl_session_tickets off;

# curl https://ssl-config.mozilla.org/ffdhe2048.txt > /path/to/dhparam
ssl_dhparam /etc/ssl/certsdhparam.pem;

# intermediate configuration
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
ssl_prefer_server_ciphers off;
# HSTS (ngx_http_headers_module is required) (63072000 seconds)
add_header Strict-Transport-Security "max-age=63072000" always;
```

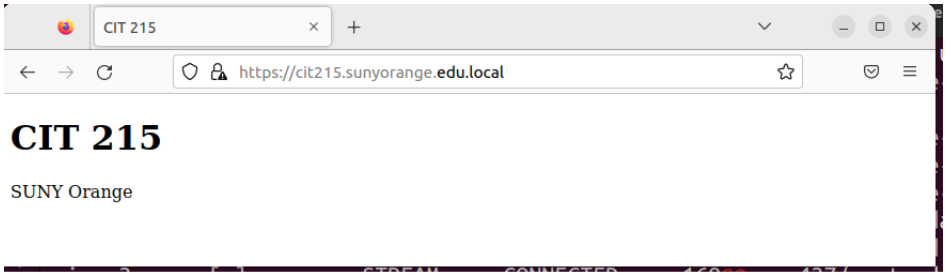


test that the config file is ok.

```
root@occc-VirtualBox:/etc/nginx/sites-enabled# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@occc-VirtualBox:/etc/nginx/sites-enabled#
```

restart nginx.

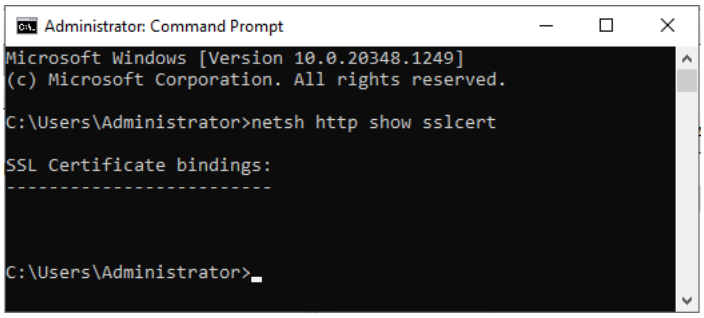




8.0.13 Windows certificate self signed certificates

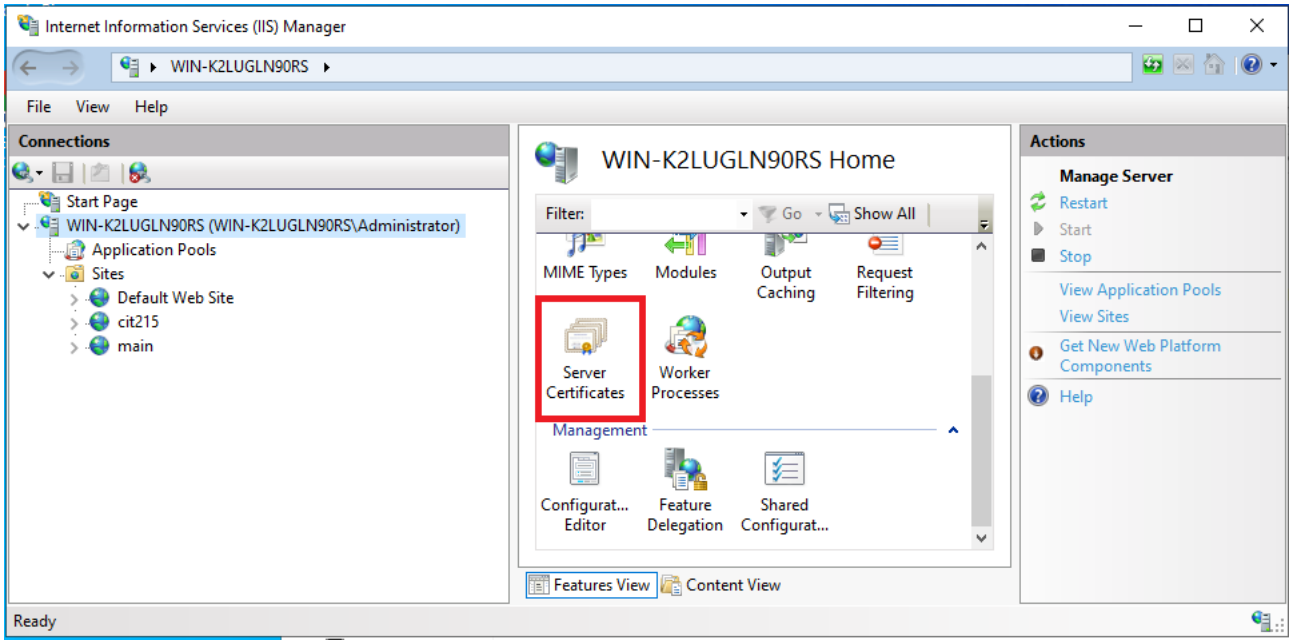
https://learn.microsoft.com/en-us/iis/manage/configuring-security/how-to-set-up-ssl-on-iis?source=recommendations
Using SSL in kernel mode requires storing SSL binding information in two places.

First, the binding is stored in %windir%\System32\inetsrv\config\applicationHost.config for your site. When the site starts, IIS sends the binding to HTTP.sys, and HTTP.sys starts listening for requests on the specified IP:Port (this works for all bindings). Second, the SSL configuration associated with the binding is stored in the HTTP.sys configuration. Use the netsh command at a command prompt to view SSL binding configuration stored in HTTP.sys as in the following example:

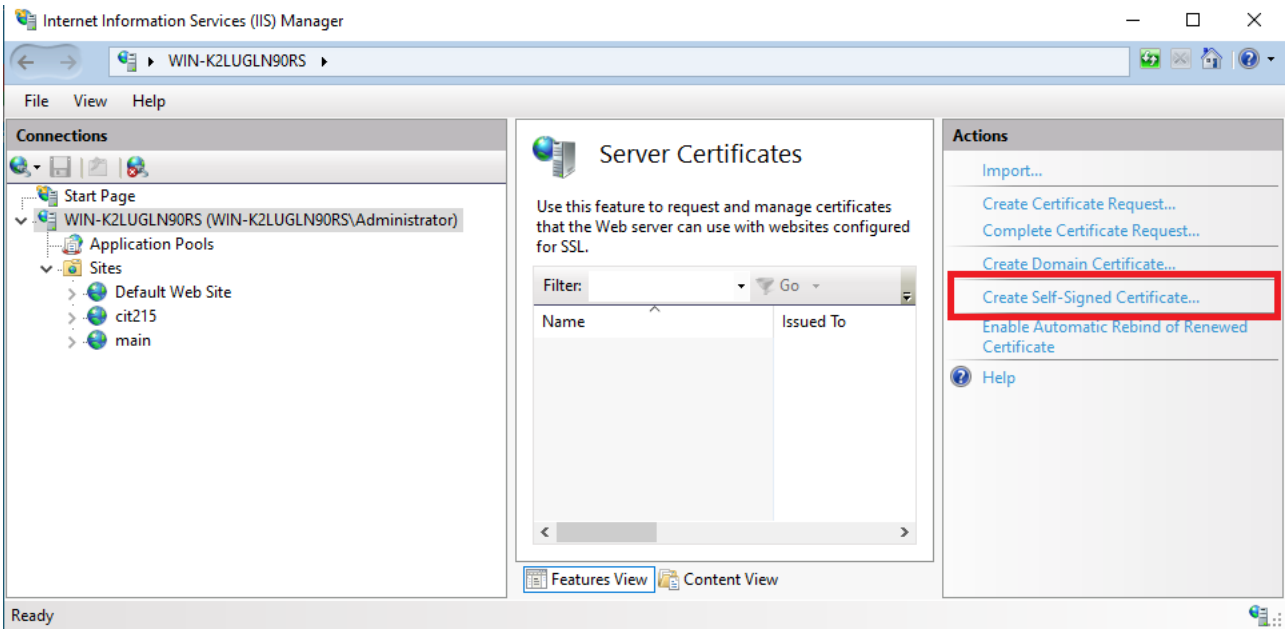


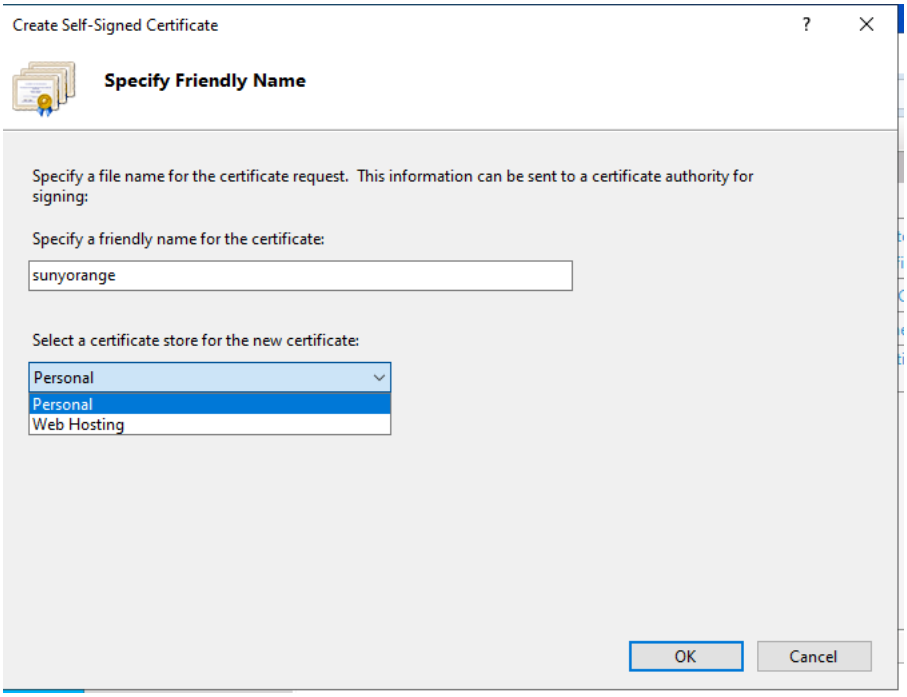
Obtain a Certificate

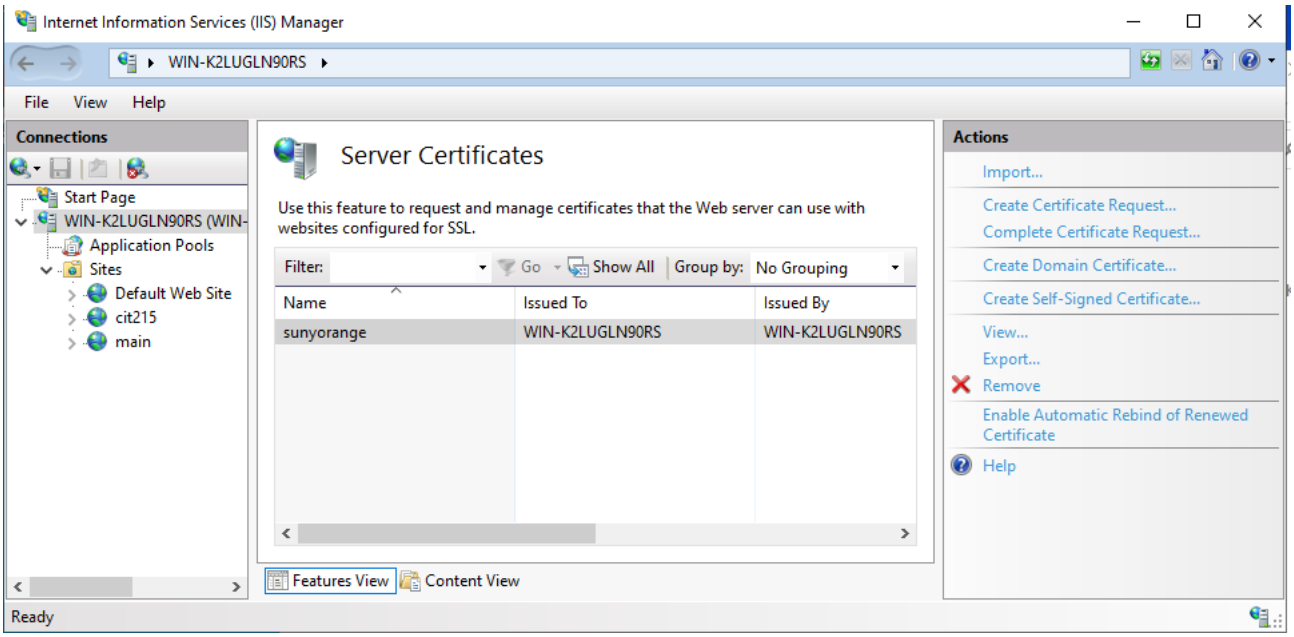
Select the server node in the treeview and double-click the Server Certificates feature in the listview:



Click Create Self-Signed Certificate... in the Actions pane.

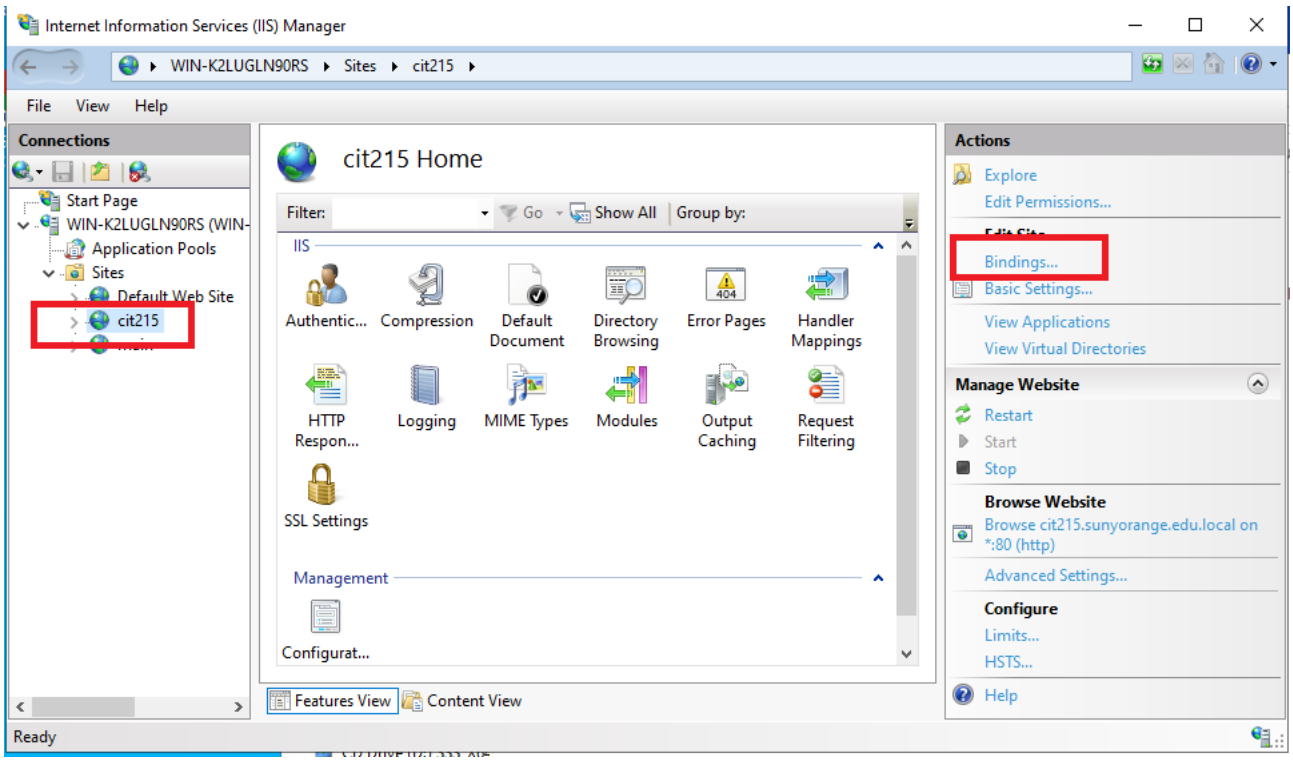

Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-K2LUGLN90RS\Administrator)
Application Pools
Sites
Default Web Site
cit215
main
Server Certificates
Use this feature to request and manage certificates that the Web server can use with websites configured for SSL.
Filter:
Go
Name
Issued To
Features View
Content View
Actions
Import...
Create Certificate Request...
Complete Certificate Request...
Create Domain Certificate...
Create Self-Signed Certificate...
Enable Automatic Rebind of Renewed Certificate
Help
Ready


Create Self-Signed Certificate
Specify Friendly Name
Specify a file name for the certificate request. This information can be sent to a certificate authority for signing:
Specify a friendly name for the certificate:
sunyorange
Select a certificate store for the new certificate:
Personal
Personal
Web Hosting
OK
Cancel

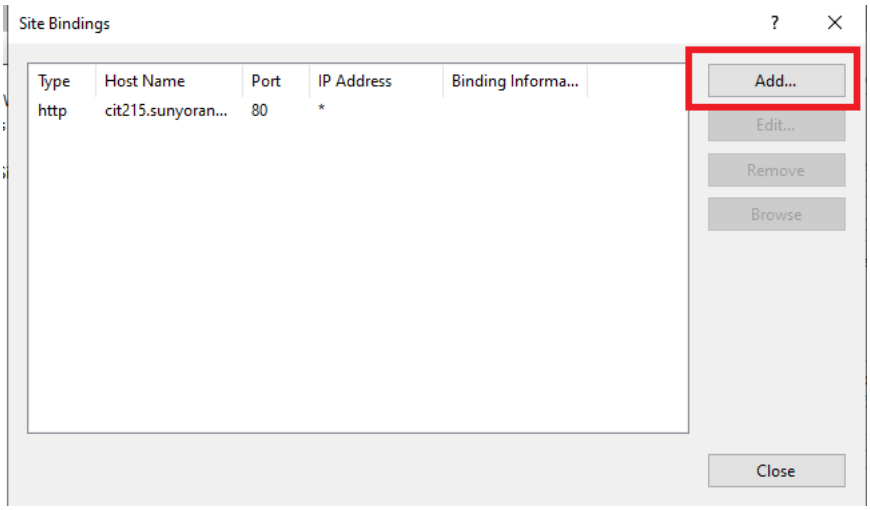


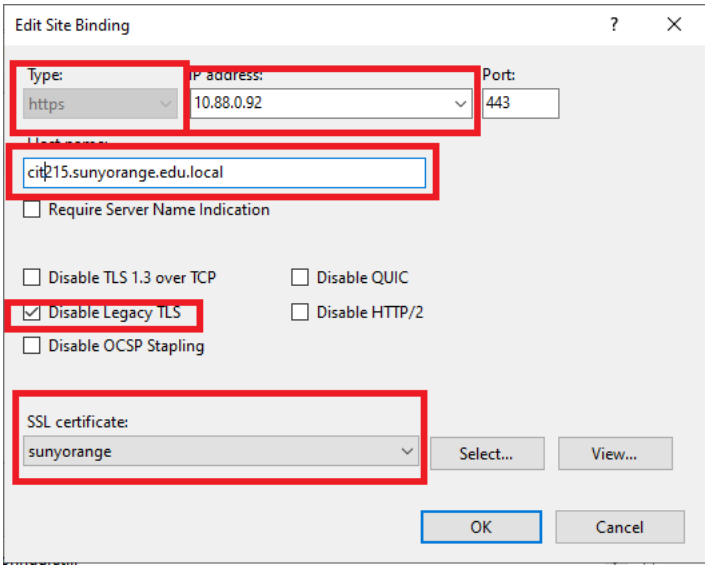
Now you have a self-signed certificate. The certificate is marked for "Server Authentication" use; that is, it uses as a server-side certificate for HTTP SSL encryption and for authenticating the identity of the server.

Create an SSL Binding

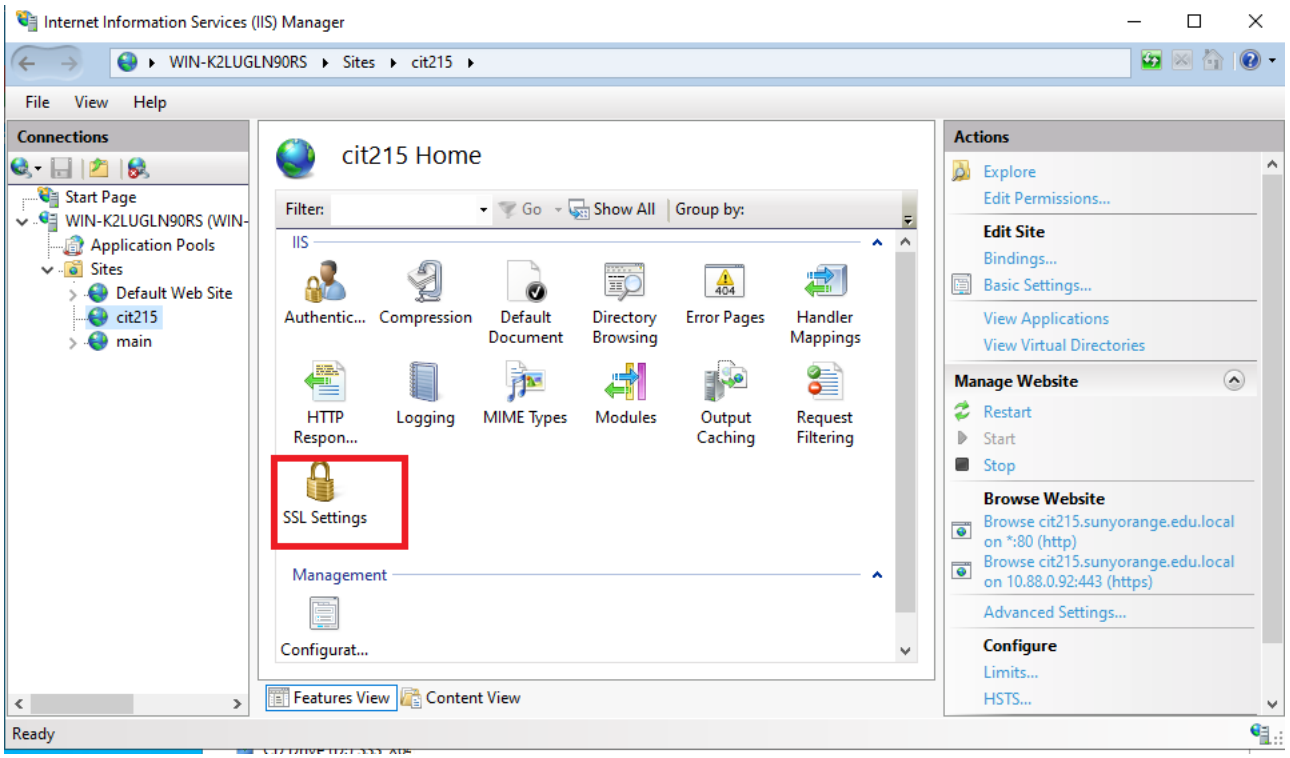


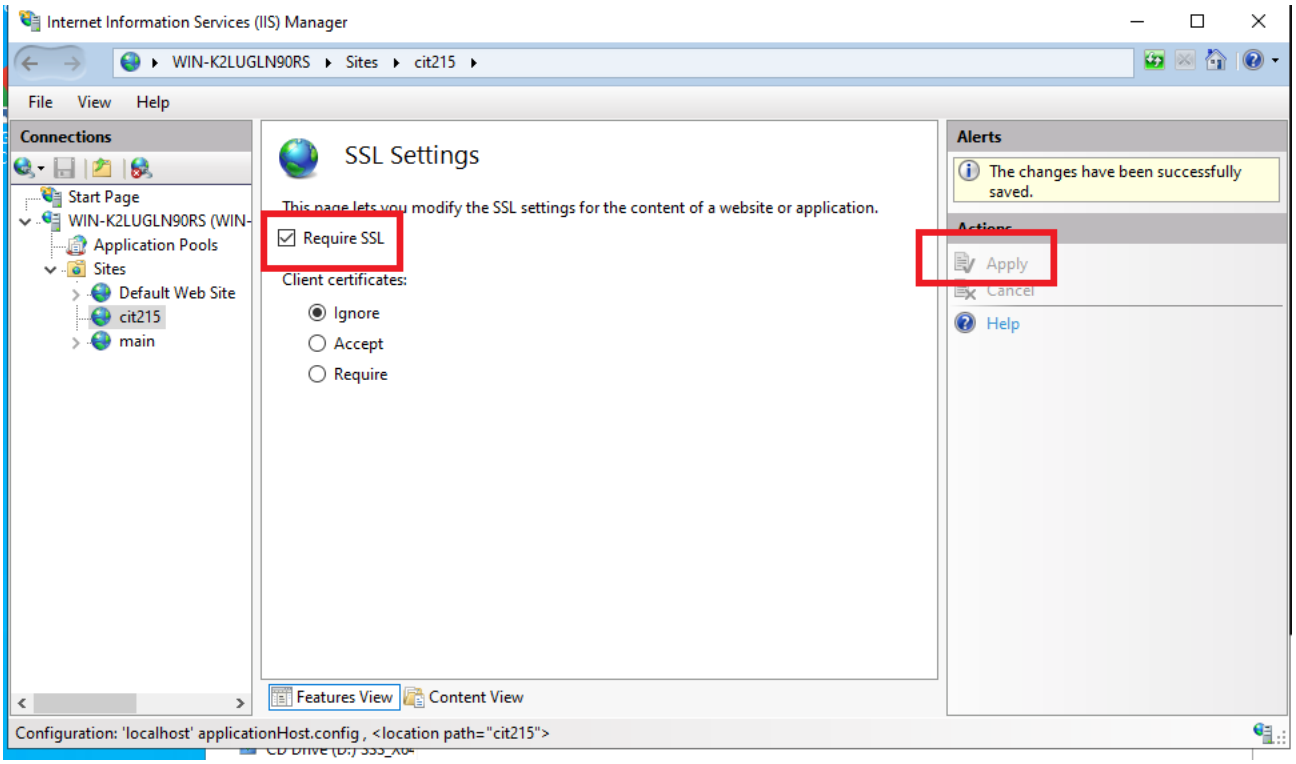
Click Add... to add your new SSL binding to the site.

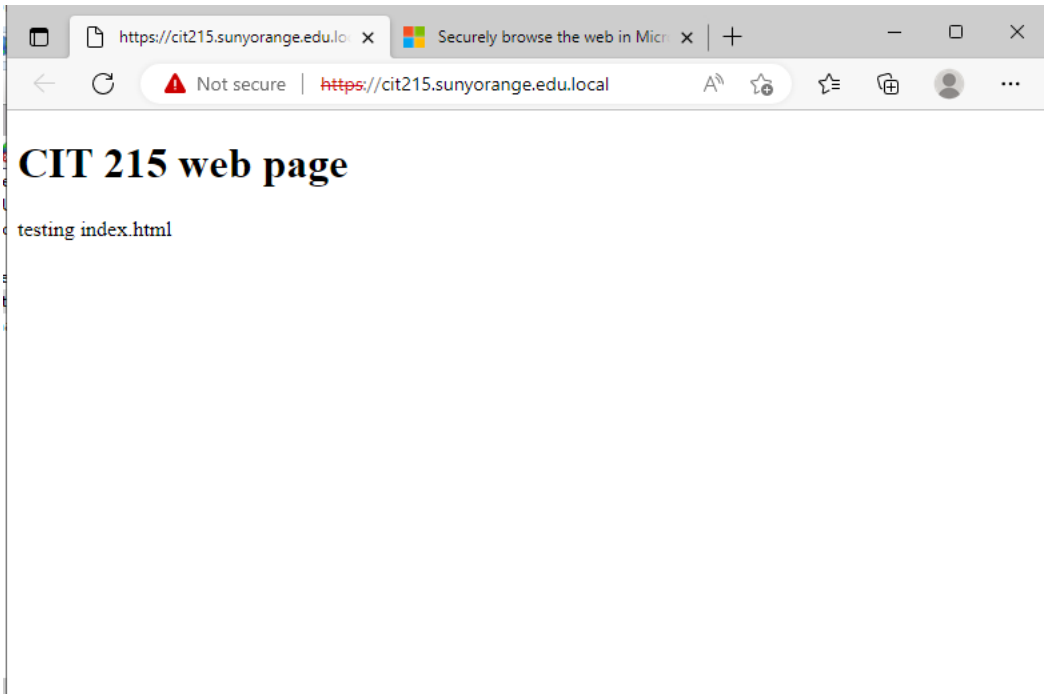


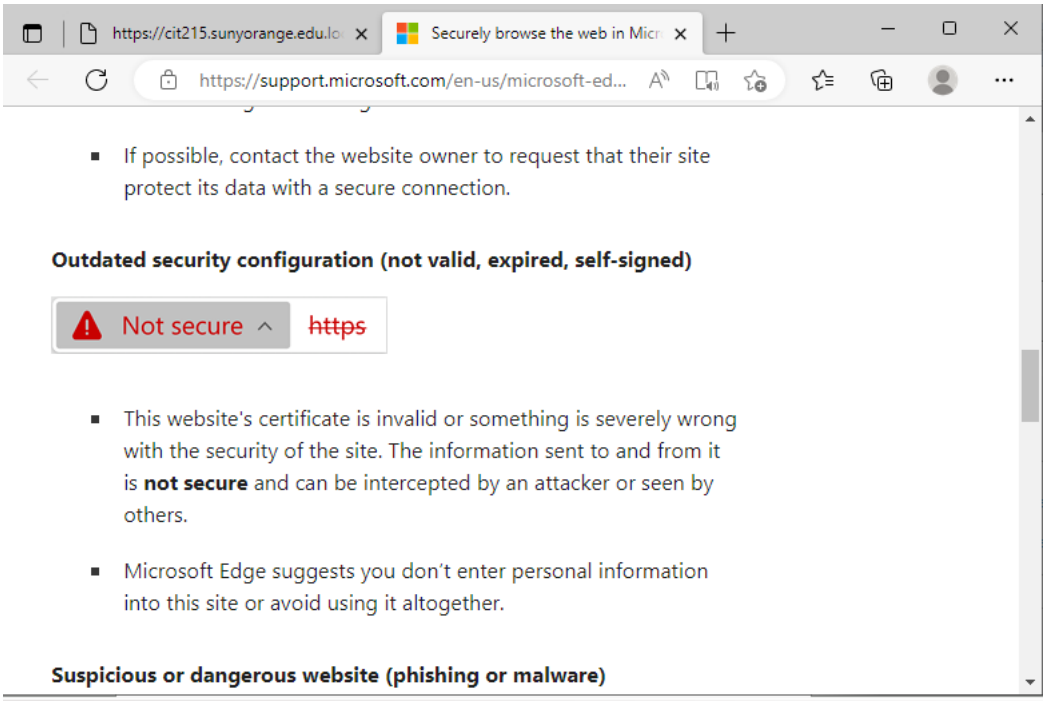


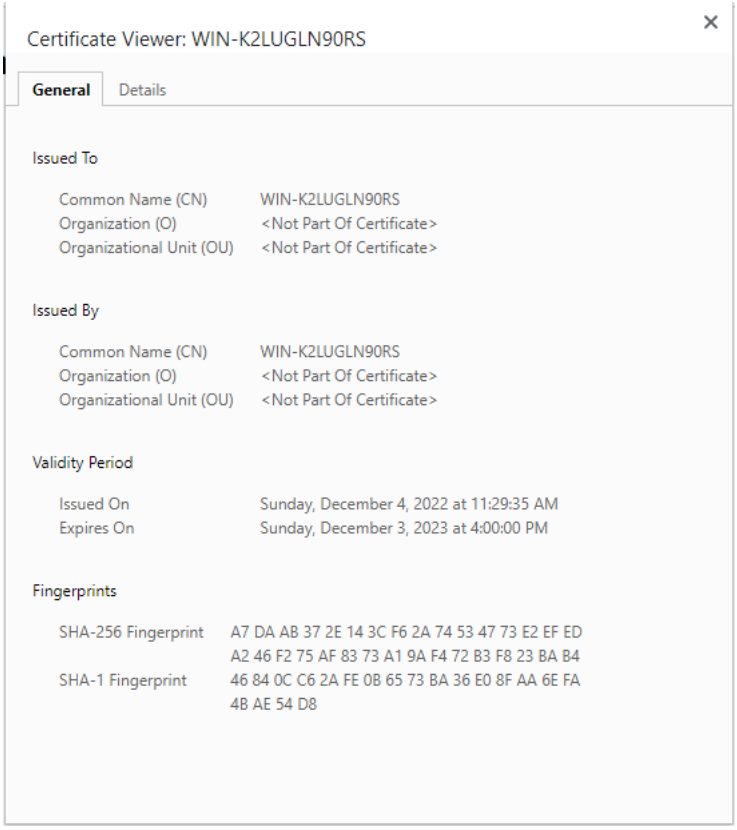
Configure SSL Settings

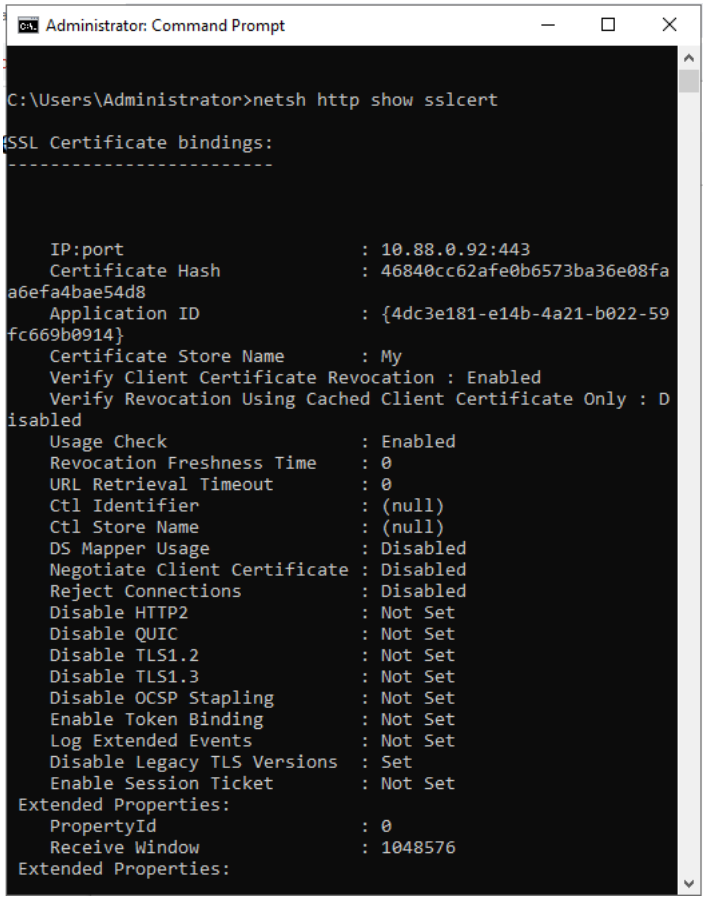

Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS ▸ Sites ▸ cit215 ▸
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-
Application Pools
Sites
Default Web Site
cit215
main
cit215 Home
Filter:
Go
Show All
Group by:
IIS
Authentic...
Compression
Default Document
Directory Browsing
Error Pages
Handler Mappings
HTTP Respon...
Logging
MIME Types
Modules
Output Caching
Request Filtering
SSL Settings
Management
Configurat...
Features View
Content View
Actions
Explore
Edit Permissions...
Edit Site
Bindings...
Basic Settings...
View Applications
View Virtual Directories
Manage Website
Restart
Start
Stop
Browse Website
Browse cit215.sunyorange.edu.local on *:80 (http)
Browse cit215.sunyorange.edu.local on 10.88.0.92:443 (https)
Advanced Settings...
Configure
Limits...
HSTS...
Ready


Internet Information Services (IIS) Manager
WIN-K2LUGLN90RS ▸ Sites ▸ cit215 ▸
File View Help
Connections
Start Page
WIN-K2LUGLN90RS (WIN-
Application Pools
Sites
Default Web Site
cit215
main
SSL Settings
This page lets you modify the SSL settings for the content of a website or application.
Require SSL
Client certificates:
Ignore
Accept
Require
Features View
Content View
Alerts
The changes have been successfully saved.
Actions
Apply
Cancel
Help
Configuration: 'localhost' applicationHost.config , <location path="cit215">


https://cit215.sunyorange.edu.lo
Securely browse the web in Micr
Not secure
https://cit215.sunyorange.edu.local
CIT 215 web page
testing index.html


https://cit215.sunyorange.edu.lo
Securely browse the web in Micr
https://support.microsoft.com/en-us/microsoft-ed...
If possible, contact the website owner to request that their site protect its data with a secure connection.
Outdated security configuration (not valid, expired, self-signed)
Not secure
https
This website's certificate is invalid or something is severely wrong with the security of the site. The information sent to and from it is not secure and can be intercepted by an attacker or seen by others.
Microsoft Edge suggests you don't enter personal information into this site or avoid using it altogether.
Suspicious or dangerous website (phishing or malware)


Certificate Viewer: WIN-K2LUGLN90RS
General
Details
Issued To
Common Name (CN) WIN-K2LUGLN90RS
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>
Issued By
Common Name (CN) WIN-K2LUGLN90RS
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>
Validity Period
Issued On Sunday, December 4, 2022 at 11:29:35 AM
Expires On Sunday, December 3, 2023 at 4:00:00 PM
Fingerprints
SHA-256 Fingerprint A7 DA AB 37 2E 14 3C F6 2A 74 53 47 73 E2 EF ED A2 46 F2 75 AF 83 73 A1 9A F4 72 B3 F8 23 BA B4
SHA-1 Fingerprint 46 84 0C C6 2A FE 0B 65 73 BA 36 E0 8F AA 6E FA 4B AE 54 D8


Administrator: Command Prompt
C:\Users\Administrator>netsh http show sslcert
SSL Certificate bindings:
IP:port : 10.88.0.92:443
Certificate Hash : 46840cc62afe0b6573ba36e08faa6efa4bae54d8
Application ID : {4dc3e181-e14b-4a21-b022-59fc669b0914}
Certificate Store Name : My
Verify Client Certificate Revocation : Enabled
Verify Revocation Using Cached Client Certificate Only : Disabled
Usage Check : Enabled
Revocation Freshness Time : 0
URL Retrieval Timeout : 0
Ctl Identifier : (null)
Ctl Store Name : (null)
DS Mapper Usage : Disabled
Negotiate Client Certificate : Disabled
Reject Connections : Disabled
Disable HTTP2 : Not Set
Disable QUIC : Not Set
Disable TLS1.2 : Not Set
Disable TLS1.3 : Not Set
Disable OCSP Stapling : Not Set
Enable Token Binding : Not Set
Log Extended Events : Not Set
Disable Legacy TLS Versions : Set
Enable Session Ticket : Not Set
Extended Properties:
PropertyId : 0
Receive Window : 1048576
Extended Properties:

Web server security refers to the tools, technologies, and processes that enable information security (IS) on a Web server. There are three main types of Web server security: physical, network and host. All network connections are protected by a firewall, a hardware or software component that prevents unauthorized access to or from a network.

Most Common Website Security Vulnerabilities

SQL Injections

SQL injection is a type of web application security vulnerability in which an attacker attempts to use application code to access or corrupt database content. If successful, this allows the attacker to create, read, update, alter, or delete data stored in the back-end database. SQL injection is one of the most prevalent types of web application security vulnerabilities.

Cross Site Scripting (XSS)

Cross-site scripting (XSS) targets an application's users by injecting code, usually a client-side script such as JavaScript, into a web application's output. The concept of XSS is to manipulate client-side scripts of a web application to execute in the manner desired by the attacker. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface websites or redirect the user to malicious sites.

Broken Authentication & Session Management)

Broken authentication and session management encompass several security issues, all of them having to do with maintaining the identity of a user. If authentication credentials and session identifiers are not protected at all times, an attacker can

hijack an active session and assume the identity of a user.

Insecure Direct Object References

Insecure direct object reference is when a web application exposes a reference to an internal implementation object. Internal implementation objects include files, database records, directories and database keys. When an application exposes a reference to one of these objects in a URL, hackers can manipulate it to gain access to a user's personal data.

Security Misconfiguration

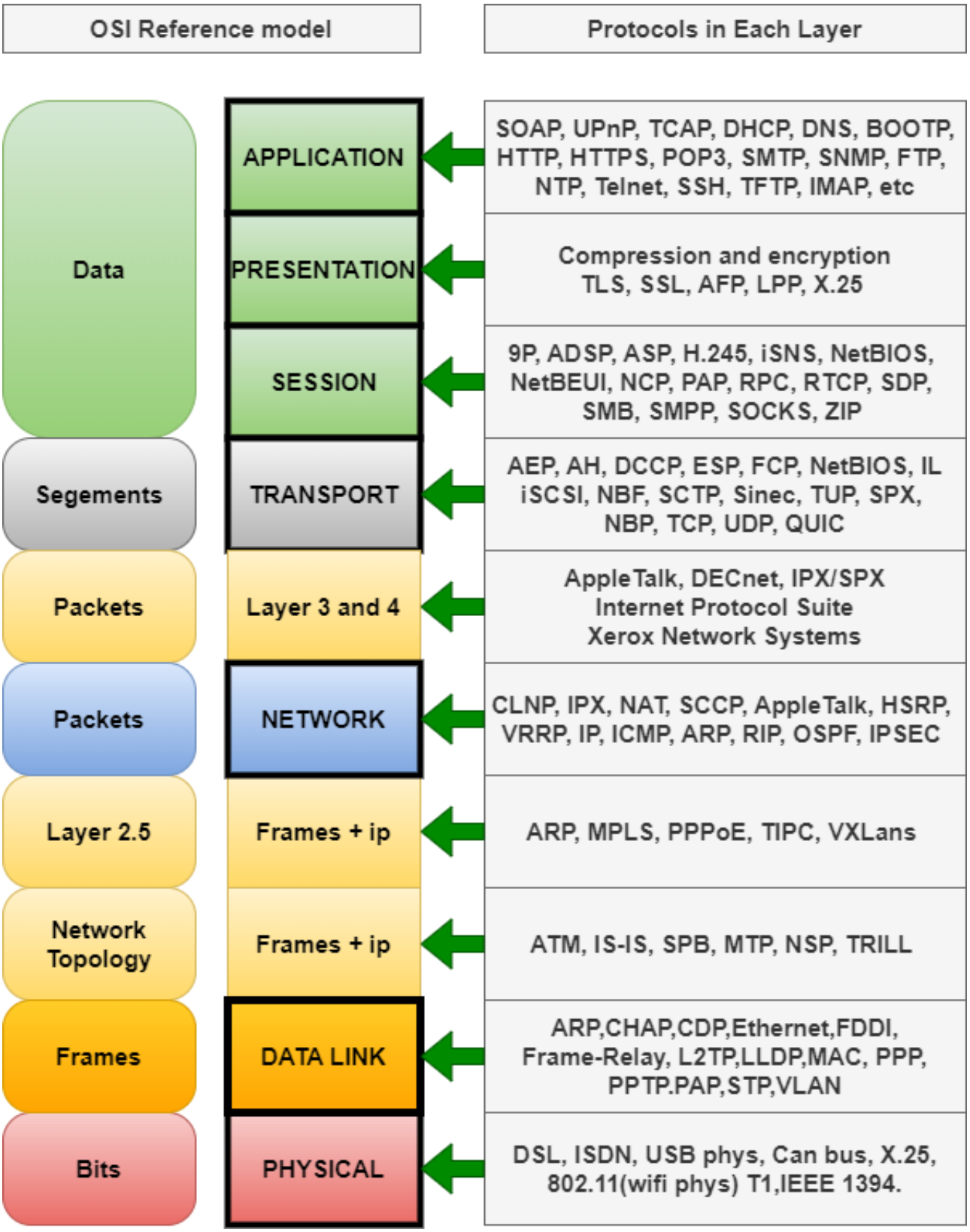
Security misconfiguration encompasses several types of vulnerabilities all centered on a lack of maintenance or a lack of attention to the web application configuration. A secure configuration must be defined and deployed for the application, frameworks, application server, web server, database server and platform. Security misconfiguration gives hackers access to private data or features and can result in a complete system compromise.

Cross-Site Request Forgery (CSRF)

Cross-Site Request Forgery (CSRF) is a malicious attack where a user is tricked into performing an action he or she didn't intend to do. A third-party website will send a request to a web application that a user is already authenticated against (e.g. their bank). The attacker can then access functionality via the victim's already authenticated browser. Targets include web applications like social media, in browser email clients, online banking, and web interfaces for network devices.

9.1 modsecurity

You probably have heard other host-based firewalls like iptables, UFW, and Firewalld, etc. The difference is that they work on layer 3 and 4 of the OSI model and take actions based on IP address and port number. ModSecurity, or web application firewalls in general(WAF), is specialized to focus on HTTP traffic (layer 7 of the OSI model) and takes action based on the content of HTTP request and response.



Mod security is a free Web Application Firewall (WAF) that works with Apache, Nginx and IIS. It supports a flexible rule engine to perform simple and complex operations and comes with a Core Rule Set (CRS) which has rules for SQL injection, cross site scripting, Trojans, bad user agents, session hijacking and a lot of other exploits.

https://coreruleset.org/

https://github.com/coreruleset/coreruleset

9.2 Apache modsecurity

ModSecurity

ModSecurity was originally designed for Apache web server. It could work with Nginx before version 3.0 but suffered from poor performance. ModSecurity 3.0 (aka libmodsecurity) was released in 2017. It's a milestone release, particularly for Nginx users, as it's the first version to work natively with Nginx. The caveat of ModSecurity 3 is that it doesn't yet have all the features as in the previous version (2.9), though each new release will add some of the missing features. Nginx users should use ModSecurity 3. However, if you use Apache, it's recommended to continue using the 2.9 branch for the time being.

Install ModSecurity with Apache on Debian/Ubuntu

The ModSecurity module for Apache is included in the default Debian/Ubuntu repository. To install it, run

```
sudo apt install libapache2-mod-security2
```

Commands

```
root@occc-VirtualBox:/etc# sudo apt install libapache2-mod-
security2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer
crda gir1.2-gnomebluetooth-1.0 libflashrom1 libftdi1-
```

```
2 libfuzzy2 libllvm12 libpython3.9-minimal libpython3.9-
stdlib ltrace python3.9
python3.9-minimal
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
liblua5.1-0 modsecurity-crs
Suggested packages:
lua geoip-database-contrib ruby python
The following NEW packages will be installed:
libapache2-mod-security2 liblua5.1-0 modsecurity-crs
0 upgraded, 3 newly installed, 0 to remove and 79 not upgraded.
Need to get 504 kB of archives.
After this operation, 2,376 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 liblua5
0 amd64 5.1.5-8.1build4 [99.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 libapac
mod-security2 amd64 2.9.5-1 [265 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy/universe amd64 modsecu
crs all 3.3.2-1 [139 kB]
Fetched 504 kB in 14s (36.0 kB/s)
Selecting previously unselected package liblua5.1-0:amd64.
(Reading database ... 217708 files and directories currently installed
Preparing to unpack .../liblua5.1-0_5.1.5-
8.1build4_amd64.deb ...
Unpacking liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Selecting previously unselected package libapache2-mod-
security2.
Preparing to unpack .../libapache2-mod-security2_2.9.5-
1_amd64.deb ...
Unpacking libapache2-mod-security2 (2.9.5-1) ...
Selecting previously unselected package modsecurity-crs.
Preparing to unpack .../modsecurity-crs_3.3.2-1_all.deb ...
Unpacking modsecurity-crs (3.3.2-1) ...
Setting up modsecurity-crs (3.3.2-1) ...
Setting up liblua5.1-0:amd64 (5.1.5-8.1build4) ...
Setting up libapache2-mod-security2 (2.9.5-1) ...
apache2_invoke: Enable module security2
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
```

```
root@occc-VirtualBox:/etc#
```

Then enable this module.

```
sudo a2enmod security2
```

Commands

```
root@occc-VirtualBox:/etc# sudo a2enmod security2
Considering dependency unique_id for security2:
Module unique_id already enabled
Module security2 already enabled
```

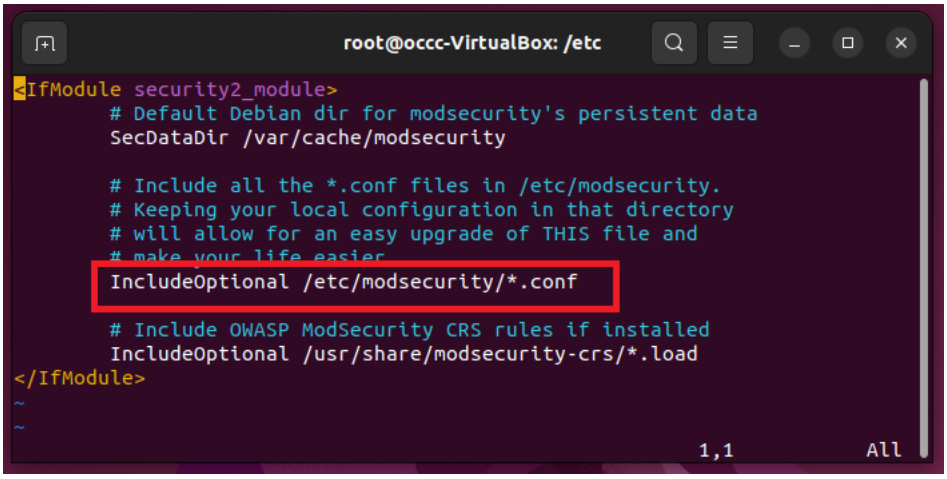
Restart Apache for the change to take effect.

```
sudo systemctl restart apache2
```

Configure ModSecurity

In the **/etc/apache2/mods-enabled/security2.conf** configuration file, you can find the following line.

```
IncludeOptional /etc/modsecurity/*.conf
```



This means Apache will include all the ***.conf** files in **/etc/modsecurity/** directory. We need to rename the **modsecurity.conf-recommended** file to

make it work.

```
sudo mv /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf
```

Then edit this file

```
sudo vi /etc/modsecurity/modsecurity.conf
```

Find the following line.

```
SecRuleEngine DetectionOnly
```

This config tells ModSecurity to log HTTP transactions, but takes no action when an attack is detected. Change it to the following, so ModSecurity will detect and block web attacks.

```
SecRuleEngine On
```

Then find the following line, which tells ModSecurity what information should be included in the audit log.

```
SecAuditLogParts ABDEFHIJZ
```

The setting should be changed to the following.

```
SecAuditLogParts ABCEFHJKZ
```

Save and close the file. Then restart Apache for the change to take effect. (Reloding the web server isn't enough.)

```
sudo systemctl restart apache2
```

Install the OWASP Core Rule Set (CRS)

To make ModSecurity protect your web applications, you need to define rules to detect malicious actors and block them. For beginners, it's a good idea to install existing rule sets, so you can get started quickly and then learn the nitty-gritty down the road. There are several free rule sets for ModSecurity. The OWASP Core Rule Set (CRS) is the standard rule set used with ModSecurity.

When installing ModSecurity from the default Debian/Ubuntu repository, the modsecurity-crs package is also installed as a dependency. This package contains the OWASP core rule set version 3.x. However, it can become out of date. If you care about security, you should use the latest version of core rule set.

```
wget https://github.com/coreruleset/coreruleset/archive/v3.3.0.tar.gz
```

Extract the file.

```
tar xvf v3.3.0.tar.gz
```

Create a directory to store CRS files.

```
sudo mkdir /etc/apache2/modsecurity-crs/
```

Move the extracted directory to /etc/apache2/modsecurity-crs/.

```
sudo mv coreruleset-3.3.0/ /etc/apache2/modsecurity-crs/
```

Go to that directory.

```
cd /etc/apache2/modsecurity-crs/coreruleset-3.3.0/
```

Rename the crs-setup.conf.example file.

```
sudo mv crs-setup.conf.example crs-setup.conf
```

Edit the /etc/apache2/mods-enabled/security2.conf file.

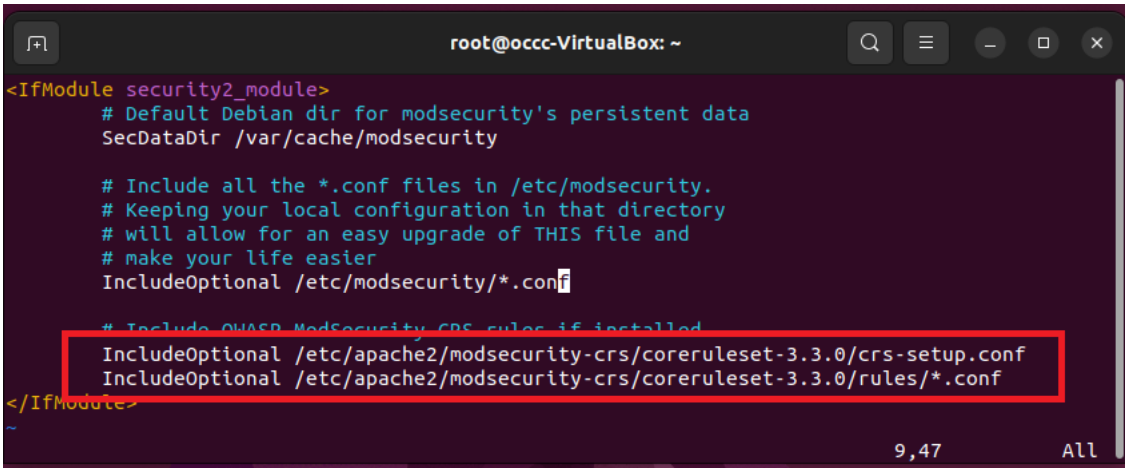
```
sudo vi /etc/apache2/mods-enabled/security2.conf
```

Find the following line, which loads the default CRS files.

```
IncludeOptional /usr/share/modsecurity-crs/*.load
```

Change it to the following, so the latest OWASP CRS will be used.

```
IncludeOptional /etc/apache2/modsecurity-crs/coreruleset-3.3.0/crs-setup.conf
IncludeOptional /etc/apache2/modsecurity-crs/coreruleset-3.3.0/rules/*.conf
```



Save and close the file. Then test Apache configuration.

```
sudo apache2ctl -t
```

NOTE if you use OWASP CRS 3.3.4 or greater then modsecurity will need to be 2.9.6 or greater.

9.2.1 Compiling mod security for apache form source

Install Required Build Tools and Dependencies

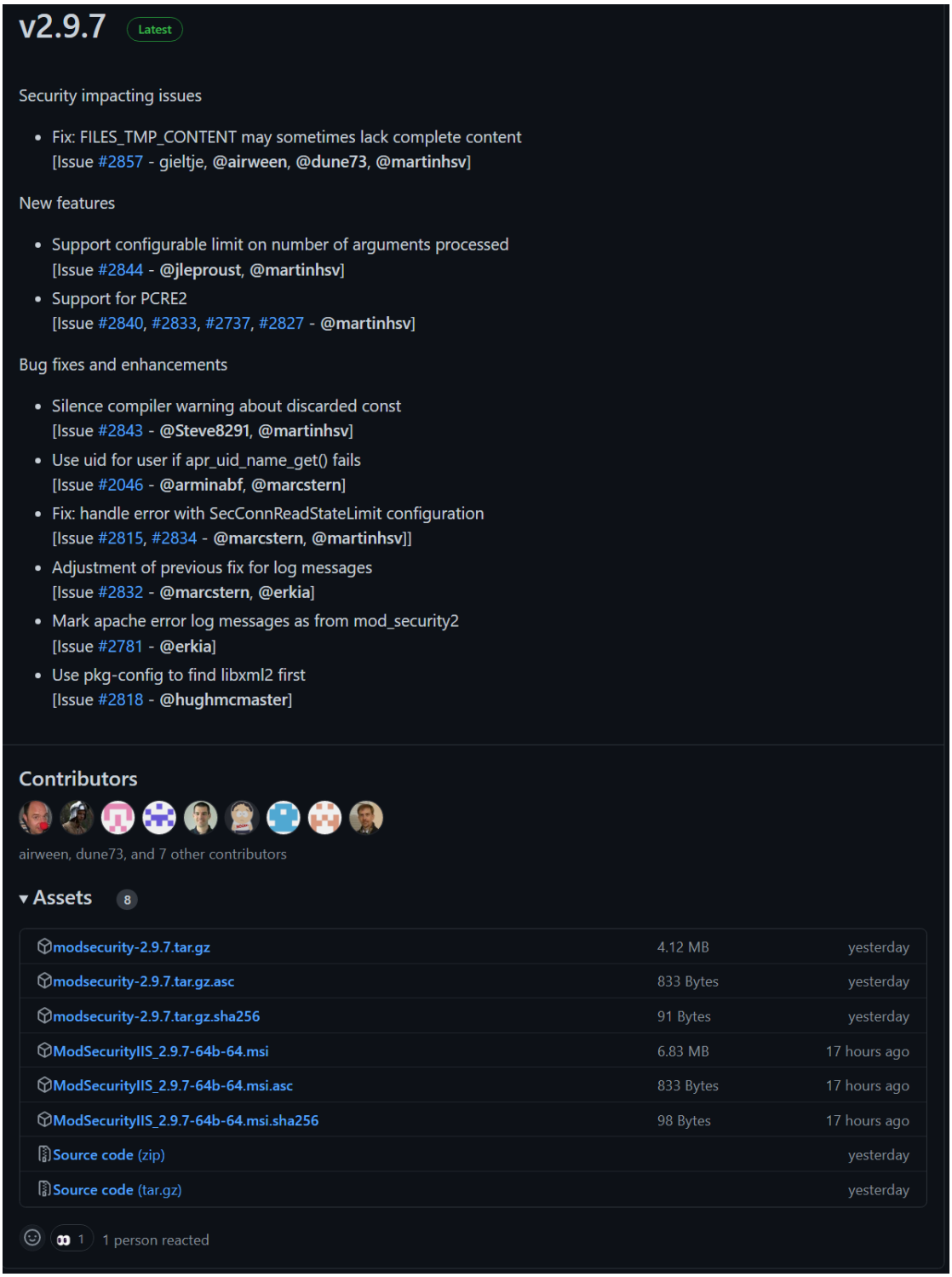
To install Libmodsecurity or Modsecurity v3 or 2.9.6+ on Ubuntu , we are going to build it from source. Hence, you need to install some required build tools and dependencies for a successful build.

```
apt install g++ flex bison curl apache2-dev doxygen libyajl-dev ssdeep
liblua5.2-dev libgeoip-dev libtool dh-autoreconf libcurl4-gnutls-dev libxml2
libpcre++-dev libxml2-dev git
```

Compile and Install LibModsecurity on Ubuntu

Download LibModsecurity Source Code

navigate to https://github.com/SpiderLabs/ModSecurity/releases/ and download ModSecurity source code. You can simply use wget to pull it.



v2.9.7 Latest

Security impacting issues

- Fix: FILES_TMP_CONTENT may sometimes lack complete content
  [Issue #2857 - gieltje, @airween, @dune73, @martinhsv]

New features

- Support configurable limit on number of arguments processed
  [Issue #2844 - @jleproust, @martinhsv]
- Support for PCRE2
  [Issue #2840, #2833, #2737, #2827 - @martinhsv]

Bug fixes and enhancements

- Silence compiler warning about discarded const
  [Issue #2843 - @Steve8291, @martinhsv]
- Use uid for user if apr_uid_name_get() fails
  [Issue #2046 - @arminabf, @marcstern]
- Fix: handle error with SecConnReadStateLimit configuration
  [Issue #2815, #2834 - @marcstern, @martinhsv]]
- Adjustment of previous fix for log messages
  [Issue #2832 - @marcstern, @erkia]
- Mark apache error log messages as from mod_security2
  [Issue #2781 - @erkia]
- Use pkg-config to find libxml2 first
  [Issue #2818 - @hughmcmaster]

Contributors

airween, dune73, and 7 other contributors

▾ Assets 8

modsecurity-2.9.7.tar.gz	4.12 MB	yesterday
modsecurity-2.9.7.tar.gz.asc	833 Bytes	yesterday
modsecurity-2.9.7.tar.gz.sha256	91 Bytes	yesterday
ModSecurityIIS_2.9.7-64b-64.msi	6.83 MB	17 hours ago
ModSecurityIIS_2.9.7-64b-64.msi.asc	833 Bytes	17 hours ago
ModSecurityIIS_2.9.7-64b-64.msi.sha256	98 Bytes	17 hours ago
Source code (zip)		yesterday
Source code (tar.gz)		yesterday

1 person reacted

wget https://github.com/SpiderLabs/ModSecurity/releases/download/v2.9.7/modsecurity-2.9.7.tar.gz

Commands

```
root@occc-VirtualBox:~# wget https://github.com/SpiderLabs/ModSecurity
2.9.7.tar.gz
```

```
--2023-01-05 15:42:31--  https://github.com/SpiderLabs/ModSecurity/rel
2.9.7.tar.gz
Resolving github.com (github.com)... 140.82.112.4
Connecting to github.com (github.com)|140.82.112.4|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-
production-release-asset-2e65be/1320594/5d2b0a79-2550-
453d-a019-a826760d09a6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230105%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20230105T204238Z&X-Amz-
Expires=300&X-Amz-Signature=73389c1717f879abec02650a19f705c18f5c8605f1
Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=1320594&response-
content-disposition=attachment%3B%20filename%3Dmodsecurity-
2.9.7.tar.gz&response-content-type=application%2Foctet-
stream [following]
--2023-01-05 15:42:38--  https://objects.githubusercontent.com/github-
production-release-asset-2e65be/1320594/5d2b0a79-2550-
453d-a019-a826760d09a6?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20230105%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20230105T204238Z&X-Amz-
Expires=300&X-Amz-Signature=73389c1717f879abec02650a19f705c18f5c8605f1
Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=1320594&response-
content-disposition=attachment%3B%20filename%3Dmodsecurity-
2.9.7.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com
Connecting to objects.githubusercontent.com (objects.githubusercontent
HTTP request sent, awaiting response... 200 OK
Length: 4320766 (4.1M) [application/octet-stream]
Saving to: ‘modsecurity-2.9.7.tar.gz’

modsecurity-2.9.7.tar. 100%[============================>]   4.12M  --
.-KB/s    in 0.1s

2023-01-05 15:42:38 (28.2 MB/s) - ‘modsecurity-
2.9.7.tar.gz’ saved [4320766/4320766]

root@occc-VirtualBox:~#
```

Extract the ModSecurity source code.

```
tar xvf modsecurity-2.9.7.tar.gz
```

Compile and Install LibModsecurity

Navigate to the LibModsecurity source directory, configure, compile and install it

```
cd modsecurity-2.9.7/
```

Configure LibModsecurity to adapt it to your system and check if any required dependency is missing.

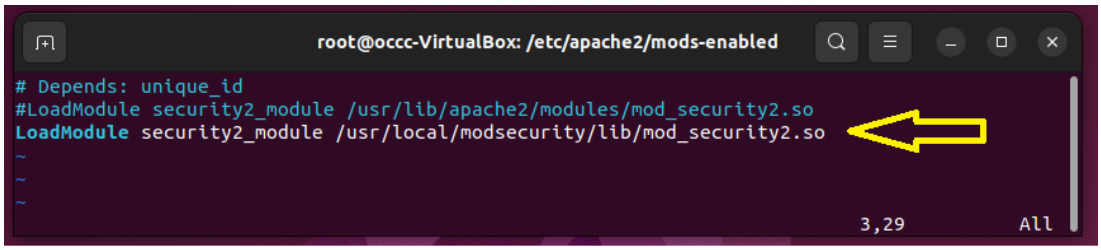
```
./configure
```

Be sure to fix any dependency issue, if any, before you can proceed to compile and install LibModsecurity with Apache on Ubuntu
If the configure script above completes with no error, proceed to compile and install LibModSecurity on Ubuntu

in /etc/apache2/mods-enabled
edit security2.load

```
vi /etc/apache2/mods-enabled/security2.load
```

point it to the new installed module we just compiled.
/usr/local/modsecurity/lib/mod_security2.so



make sure that we can load the module and have no errors.

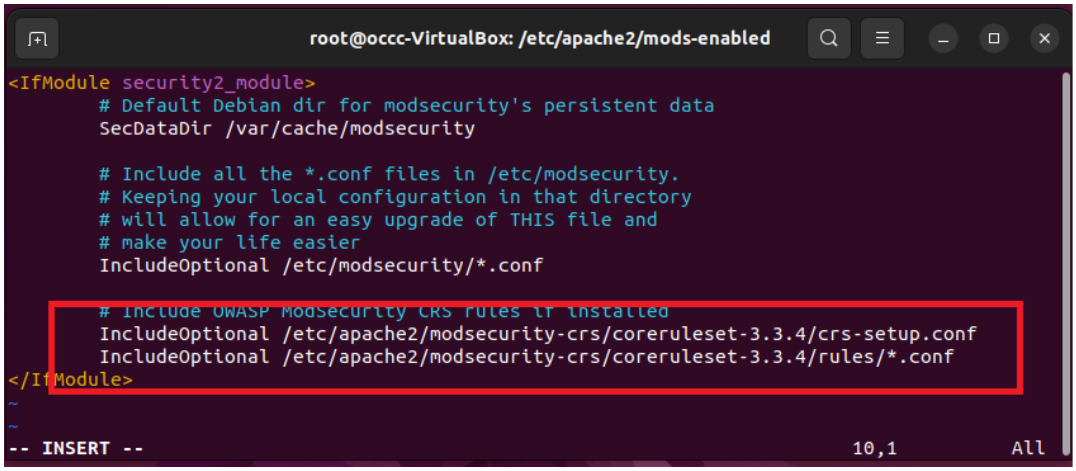
```
apache2ctl -t
```

Download the latest owasp crs

```
wget https://github.com/coreruleset/coreruleset/archive/v3.3.4.tar.gz
tar xvf v3.3.4.tar.gz
sudo mv coreruleset-3.3.4/ /etc/apache2/modsecurity-crs/
```

edit the /etc/apache2/mods-enabled/security2.conf

```
vi /etc/apache2/mods-enabled/security2.conf
```



makesure to move the setup file in the new coreruleset directory.

```
sudo mv crs-setup.conf.example crs-setup.conf
```

make sure that we can load the module and have no errors.

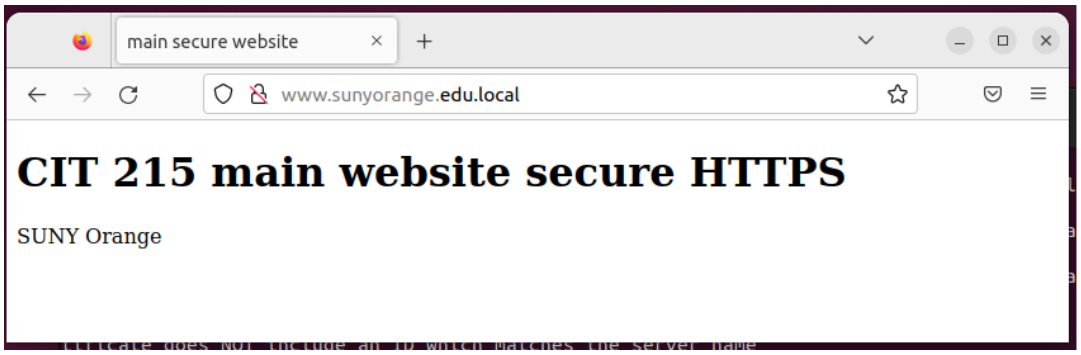
```
apache2ctl -t
```

restart webserver

```
systemctl stop apache2.service
systemctl start apache2.service
```

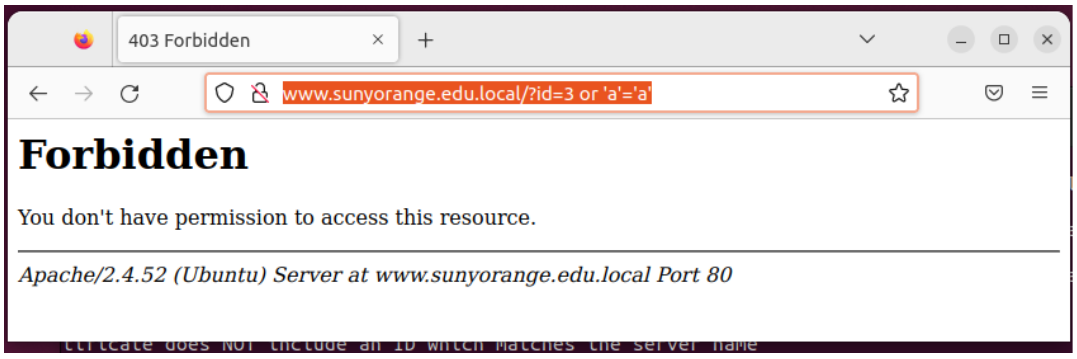
Testing

make sure our regular website is still fully functional.



test it with a possible SQL injection.

```
http://www.sunyorange.edu.local/?id=3 or 'a'='a'
```



additional security

The above picture still gives away too many important details about our server. We can see that it is running apache 2.4.52 on ubuntu operating system.
Lets change that to something less identifiable.
We're going to be dealing with the following two options:

- **ServerSignature** used to configure a footer line under the server-generated documents.

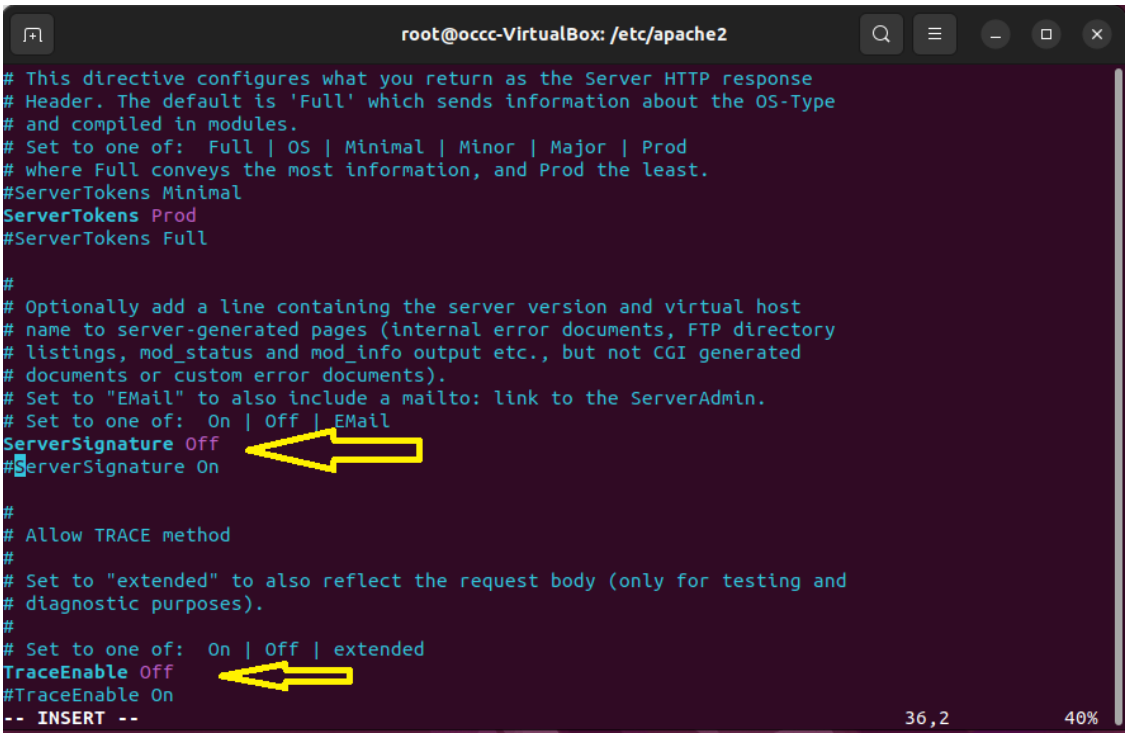
- **ServerTokens** controls the details which the server sends. The details can include OS and other complied modules.

There are six different ServerToken settings:

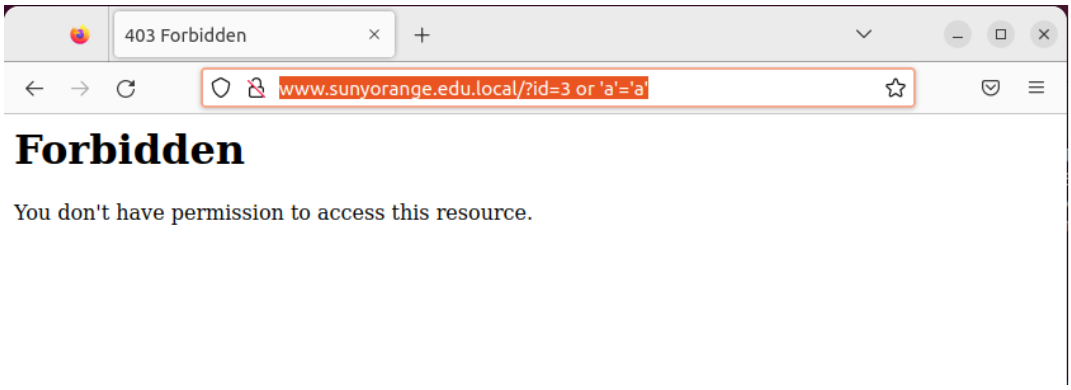
- **Full** (or not specified) – Server: Apache/2.4.52 (UNIX) PHP/7.0.25
- **Prod** (or ProductOnly) – Server: Apache
- **Major** Server: Apache/2
- **Minor** Server: Apache/2.4
- **Min** (or Minimal) – Server: Apache/2.4.52
- **OS** Server: Apache/2.4.52 (UNIX)

Lets edit our configuration.

```
vi /etc/apache2/conf-enabled/security.conf
```



Restart your apache2.



9.3 nginx modsecurity

The Nginx binary needs to be compiled with the **–with-compat** argument, which will make dynamic modules binary-compatible with your existing Nginx binary. However, not every Nginx binary shipped from the default Debian/Ubuntu repository is compiled with the –with-compat argument.

Check the configure arguments of Nginx with the following command:

Commands

```
nginx -V
```

```
occc@occc-VirtualBox:~$ sudo nginx -V
[sudo] password for occc:
nginx version: nginx/1.18.0 (Ubuntu)
built with OpenSSL 3.0.2 15 Mar 2022
TLS SNI support enabled
configure arguments: --with-cc-opt='-g -O2 -ffile-prefix-map=/build/nginx-d8gVax/nginx-1.18.0=. -flto=auto -ffat-lto-objects -flto=
auto -ffat-lto-objects -fstack-protector-strong -Wformat -Werror=format-security -fPIC -Wdate-time -D_FORTIFY_SOURCE=2' --with-ld-o
pt='-Wl,-Bsymbolic-functions -flto=auto -ffat-lto-objects -flto=auto -Wl,-z,relro -Wl,-z,now -fPIC' --prefix=/usr/share/nginx --con
f-path=/etc/nginx/nginx.conf --http-log-path=/var/log/nginx/access.log --error-log-path=/var/log/nginx/error.log --lock-path=/var/l
ock/nginx.lock --pid-path=/run/nginx.pid --modules-path=/usr/lib/nginx/modules --http-client-body-temp-path=/var/lib/nginx/body --h
ttp-fastcgi-temp-path=/var/lib/nginx/fastcgi --http-proxy-temp-path=/var/lib/nginx/proxy --http-scgi-temp-path=/var/lib/nginx/scgi
--http-uwsgi-temp-path=/var/lib/nginx/uwsgi --with-compat --with-debug --with-pcre-jit --with-http_ssl_module --with-http_stub_stat
us_module --with-http_realip_module --with-http_auth_request_module --with-http_v2_module --with-http_dav_module --with-http_slice_
module --with-threads --add-dynamic-module=/build/nginx-d8gVax/nginx-1.18.0/debian/modules/http-geoip2 --with-http_addition_module
--with-http_gunzip_module --with-http_gzip_static_module --with-http_sub_module
occc@occc-VirtualBox:~$
```

Install libmodsecurity3

libmodsecurrity is the ModSecurity library that actually does the HTTP filtering for your web applications. On Debian 10 and Ubuntu 20.04, 22.04, you can install it with **sudo apt install libmodsecurity3**
This is not recommended and will not work since other numerous dependencies need to be installed.

9.3.1 compile libmodsecurity3 from source

Modsecurity github site.
https://github.com/SpiderLabs/ModSecurity

Install git

Commands

```
root@occc-VirtualBox:/etc/nginx# sudo apt install git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer
libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
git-man liberror-perl
Suggested packages:
git-daemon-run | git-daemon-sysvinit git-doc git-email git-
gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 51 not upgraded.
Need to get 4,112 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-
perl all 0.17029-1 [26.5 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-
updates/main amd64 git-man all 1:2.34.1-1ubuntu1.5 [953 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-
updates/main amd64 git amd64 1:2.34.1-1ubuntu1.5 [3,132 kB]
Fetched 4,112 kB in 0s (12.0 MB/s)
Selecting previously unselected package liberror-perl.
(Reading database ... 205956 files and directories currently installed
Preparing to unpack .../liberror-perl_0.17029-1_all.deb ...
Unpacking liberror-perl (0.17029-1) ...
```

```
Selecting previously unselected package git-man.
Preparing to unpack .../git-man_1%3a2.34.1-
1ubuntu1.5_all.deb ...
Unpacking git-man (1:2.34.1-1ubuntu1.5) ...
Selecting previously unselected package git.
Preparing to unpack .../git_1%3a2.34.1-
1ubuntu1.5_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.5) ...
Setting up liberror-perl (0.17029-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.5) ...
Setting up git (1:2.34.1-1ubuntu1.5) ...
Processing triggers for man-db (2.10.2-1) ...
root@occc-VirtualBox:/etc/nginx#
```

Clone modsecurity git repo

git clone –depth 1 -b v3/master –single-branch https://github.com/SpiderLabs/ModSecurity /usr/local/src/ModSecurity/

Commands

```
root@occc-VirtualBox:/etc/nginx# git clone --depth 1 -
b v3/master --single-branch https://github.com/SpiderLabs/ModSecurity
Cloning into '/usr/local/src/ModSecurity'...
remote: Enumerating objects: 844, done.
remote: Counting objects: 100% (844/844), done.
remote: Compressing objects: 100% (806/806), done.
remote: Total 844 (delta 498), reused 140 (delta 31), pack-
reused 0
Receiving objects: 100% (844/844), 792.52 KiB | 9.78 MiB/s, done.
Resolving deltas: 100% (498/498), done.
root@occc-VirtualBox:/etc/nginx#
```

sudo apt install gcc g++ make build-essential autoconf automake libtool libcurl4-openssl-dev liblua5.3-dev libpcre2-dev libfuzzy-dev ssdeep gettext libpcre3 libpcre3-dev libxml2 libxml2-dev libcurl4 libgeoip-dev libyajl-dev doxygen flex bison curl apache2-dev ssdeep dh-autoreconf libpcre++-dev liblmdb-dev libpkgconf3 lmdb-doc pkgconf zlib1g-dev libssl-dev

cd /usr/local/src/ModSecurity/

Install required submodules.

git submodule init

git submodule update

Commands

```
root@occc-VirtualBox:/usr/local/src/ModSecurity# git submodule init
Submodule 'bindings/python' (https://github.com/SpiderLabs/ModSecurity
Python-bindings.git) registered for path 'bindings/python'
Submodule 'others/libinjection' (https://github.com/libinjection/libi
Submodule 'test/test-cases/secrules-language-
tests' (https://github.com/SpiderLabs/secrules-language-
tests) registered for path 'test/test-cases/secrules-language-
tests'
root@occc-VirtualBox:/usr/local/src/ModSecurity# git submodule update
Cloning into '/usr/local/src/ModSecurity/bindings/python'...
Cloning into '/usr/local/src/ModSecurity/others/libinjection'...
Cloning into '/usr/local/src/ModSecurity/test/test-
cases/secrules-language-tests'...
Submodule path 'bindings/python': checked out 'bc625d5bb0bac6a64bcce8c
Submodule path 'others/libinjection': checked out 'bfba51f5af8f1f6cf5c
Submodule path 'test/test-cases/secrules-language-
tests': checked out 'a3d4405e5a2c90488c387e589c5534974575e35b'
root@occc-VirtualBox:/usr/local/src/ModSecurity#
```

Configure the build environment.

./build.sh

./configure

Commands

```
root@occc-VirtualBox:/usr/local/src/ModSecurity# ./build.sh
libtoolize: putting auxiliary files in '.'.
libtoolize: copying file './ltmain.sh'
libtoolize: putting macros in AC_CONFIG_MACRO_DIRS, 'build'.
libtoolize: copying file 'build/libtool.m4'
libtoolize: copying file 'build/ltoptions.m4'
libtoolize: copying file 'build/ltsugar.m4'
libtoolize: copying file 'build/ltversion.m4'
libtoolize: copying file 'build/lt~obsolete.m4'
fatal: No names found, cannot describe anything.
fatal: No names found, cannot describe anything.
fatal: No names found, cannot describe anything.
...
configure.ac:307: warning: AC_PROG_LEX without either yywrap or noyywr
./lib/autoconf/programs.m4:716: _AC_PROG_LEX is expanded from...
./lib/autoconf/programs.m4:709: AC_PROG_LEX is expanded from...
configure.ac:307: the top level
root@occc-VirtualBox:/usr/local/src/ModSecurity#

root@occc-VirtualBox:/usr/local/src/ModSecurity# ./configure
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a race-free mkdir -p... /usr/bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
...
config.status: creating examples/reading_logs_via_rule_message/Makefil
config.status: creating examples/using_bodies_in_chunks/Makefile
config.status: creating src/config.h
config.status: executing depfiles commands
config.status: executing libtool commands
```

```
ModSecurity -  for Linux

Mandatory dependencies
+ libInjection                                          ....v3.9.2-46-
gbfba51f
+ SecLang tests                                        ....a3d4405

Optional dependencies
+ GeoIP/MaxMind                                        ....found
* (GeoIP) v1.6.12
-lGeoIP , -I/usr/include/
+ LibCURL                                              ....found v7.81.0
-lcurl,  -DWITH_CURL_SSLVERSION_TLSv1_2 -DWITH_CURL
+ YAJL                                                 ....found v2.1.0
-lyajl , -DWITH_YAJL -I/usr/include/yajl
+ LMDB                                                 ....disabled
+ LibXML2                                              ....found v2.9.13
-lxml2 , -I/usr/include/libxml2  -DWITH_LIBXML2
+ SSDEEP                                               ....found
-lfuzzy -L/usr/lib/x86_64-linux-gnu/, -DWITH_SSDEEP -
I/usr/include
+ LUA                                                  ....found v503
-llua5.3 -L/usr/lib/x86_64-linux-gnu/, -DWITH_LUA -
DWITH_LUA_5_3 -I/usr/include/lua5.3
+ PCRE2                                                ....disabled

Other Options
+ Test Utilities                                       ....enabled
+ SecDebugLog                                          ....enabled
+ afl fuzzer                                           ....disabled
+ library examples                                     ....enabled
+ Building parser                                      ....disabled
+ Treating pm operations as critical section           ....disabled

root@occc-VirtualBox:/usr/local/src/ModSecurity#
```

If you see the following error, you can ignore it.
fatal: No names found, cannot describe anything.

make
make install

Download the NGINX Connector for ModSecurity and Compile It as a Dynamic Module

The **ModSecurity Nginx Connector** links **libmodsecurity** to the Nginx web server. Clone the ModSecurity v3 Nginx Connector Git repository.

git clone –depth 1 https://github.com/SpiderLabs/ModSecurity-nginx.git /usr/local/src/ModSecurity-nginx/

Commands

```
root@occc-VirtualBox:/usr/local/src/ModSecurity# git clone --
depth 1 https://github.com/SpiderLabs/ModSecurity-
nginx.git /usr/local/src/ModSecurity-nginx/
Cloning into '/usr/local/src/ModSecurity-nginx'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (40/40), done.
remote: Compressing objects: 100% (38/38), done.
remote: Total 40 (delta 11), reused 11 (delta 0), pack-
reused 0
Receiving objects: 100% (40/40), 44.94 KiB | 2.25 MiB/s, done.
Resolving deltas: 100% (11/11), done.
root@occc-VirtualBox:/usr/local/src/ModSecurity#
```

Download the source code corresponding to the installed version of NGINX (the complete sources are required even though only the dynamic module is being compiled):

Commands

```
root@occc-VirtualBox:/usr/local/src/ModSecurity# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
root@occc-VirtualBox:/usr/local/src/ModSecurity#
```

Commands

```
root@occc-VirtualBox:/usr/local/src# nginx -v
nginx version: nginx/1.18.0 (Ubuntu)
root@occc-VirtualBox:src# wget http://nginx.org/download/nginx-
1.18.0.tar.gz
--2022-12-11 17:52:22--  http://nginx.org/download/nginx-
1.18.0.tar.gz
Resolving nginx.org (nginx.org)... 52.58.199.22, 3.125.197.172, 2a05:d
Connecting to nginx.org (nginx.org)|52.58.199.22|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1039530 (1015K) [application/octet-stream]
Saving to: ‘nginx-1.18.0.tar.gz’

nginx-1.18.0.tar.gz              100%[=============================

2022-12-11 17:52:23 (1.70 MB/s) - ‘nginx-
1.18.0.tar.gz’ saved [1039530/1039530]

root@occc-VirtualBox:/usr/local/src#
```

Commands

```
root@occc-VirtualBox:/usr/local/src# tar xzvf nginx-
1.18.0.tar.gz
nginx-1.18.0/
nginx-1.18.0/auto/
nginx-1.18.0/conf/
nginx-1.18.0/contrib/
nginx-1.18.0/src/
nginx-1.18.0/configure
nginx-1.18.0/LICENSE
nginx-1.18.0/README
nginx-1.18.0/html/
....
root@occc-VirtualBox:/usr/local/src# ls
ModSecurity  ModSecurity-nginx  nginx-1.18.0  nginx-
1.18.0.tar.gz
root@occc-VirtualBox:/usr/local/src# rm nginx-1.18.0.tar.gz
root@occc-VirtualBox:/usr/local/src#
```

Make sure you are in the Nginx source directory.

Commands

```
root@occc-VirtualBox:/usr/local/src# cd nginx-1.18.0/
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0#
```

Uncomment all the deb-src lines in the /etc/apt/sources.list file.

```
sudo apt update
```

Install build dependencies for Nginx.

```
sudo apt build-dep nginx
```

Commands

```
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0# apt build-
dep nginx
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer
libflashrom1 libftdi1-2
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
diffstat libbrotli-dev libdeflate-dev libfontconfig-
dev libfreetype-dev libfreetype6-dev libgd-dev libhiredis-dev
libhiredis0.14 libice-dev libjbig-dev libjpeg-dev libjpeg-
turbo8-dev libjpeg8-dev liblzma-dev libmaxminddb-dev libmhash-
dev
libpam0g-dev libperl-dev libpng-dev libpthread-stubs0-
dev libsm-dev libtiff-dev libtiffxx5 libvpx-dev libx11-
dev libxau-dev
libxcb1-dev libxdmcp-dev libxpm-dev libxslt1-dev libxt-
dev quilt x11proto-dev xorg-sgml-doctools xtrans-dev
0 upgraded, 36 newly installed, 0 to remove and 29 not upgraded.
Need to get 7,726 kB of archives.
After this operation, 29.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
Get:1 http://us.archive.ubuntu.com/ubuntu jammy/main amd64 diffstat am
1build2 [29.2 kB]
...
```

sudo apt install uuid-dev

May already be installed.

Commands

```
root@occc-VirtualBox:/usr/local/src/nginx-
1.18.0# sudo apt install uuid-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
uuid-dev is already the newest version (2.37.2-4ubuntu3).
uuid-dev set to manually installed.
The following packages were automatically installed and are no longer
libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 29 not upgraded.
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0#
```

./configure –with-compat –with-openssl=/usr/include/openssl/ –add-dynamic-module=/usr/local/src/ModSecurity-nginx

Build the **ModSecurity Nginx Connector** module.

make modules

Commands

```
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0# make modules
make -f objs/Makefile modules
make[1]: Entering directory '/usr/local/src/nginx-1.18.0'
...
objs/ngx_http_modsecurity_module_modules.o \
-Wl,-rpath,/usr/local/modsecurity/lib -
```

```
L/usr/local/modsecurity/lib -lmodsecurity \
-shared
make[1]: Leaving directory '/usr/local/src/nginx-1.18.0'
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0#
```

The module will be save as objs/ngx_http_modsecurity_module.so. Copy it to the /usr/share/nginx/modules/ directory.

Commands

```
sudo cp objs/ngx_http_modsecurity_module.so /usr/share/nginx/modules/
```

Load the ModSecurity v3 Nginx Connector Module

Edit the main Nginx configuration file.

```
sudo vi /etc/nginx/nginx.conf
```

Add the following line at the beginning of the file.

```
load_module modules/ngx_http_modsecurity_module.so;
```

Also, add the following two lines in the http ... section, so ModSecurity will be enabled for all Nginx virtual hosts.

```
modsecurity on;
modsecurity_rules_file /etc/nginx/modsec/main.conf;
```

modsecurity nginx.conf main file

```
1 user www-data;
2 worker_processes auto;
3 pid /run/nginx.pid;
4 load_module modules/ngx_http_modsecurity_module.so;
5 include /etc/nginx/modules-enabled/*.conf;
```

```
6
7  events {
8      worker_connections 768;
9      # multi_accept on;
10 }
11
12 http {
13     modsecurity on;
14     modsecurity_rules_file /etc/nginx/modsec/main.↙
          ↳  conf;
15
16     ##
17     # Basic Settings
18     ##
19
20     sendfile on;
21     tcp_nopush on;
```

Next, create the **/etc/nginx/modsec/** directory to store ModSecurity configuration.

```
sudo mkdir /etc/nginx/modsec/
```

Then copy the ModSecurity configuration file.

```
sudo cp /usr/local/src/ModSecurity/modsecurity.conf-recommended
/etc/nginx/modsec/modsecurity.conf
```

Edit the file.

```
sudo vi /etc/nginx/modsec/modsecurity.conf
```

Find the following line.

SecRuleEngine DetectionOnly

This config tells ModSecurity to log HTTP transactions, but takes no action when an attack is detected. Change it to the following, so ModSecurity will detect and block web attacks.

SecRuleEngine On

Then find the following line , which tells ModSecurity what information should be included in the audit log.

SecAuditLogParts ABIJDEFHZ

The setting should be changed to the following.

SecAuditLogParts ABCEFHJKZ

If you have a coding website, you might want to disable response body inspection, otherwise, you might get 403 forbidden errors just by loading a web page with lots of code content.

SecResponseBodyAccess Off

Save and close the file.

Next, create the /etc/nginx/modsec/main.conf file.

```
sudo vi /etc/nginx/modsec/main.conf
```

Add the following line to include the /etc/nginx/modsec/modsecurity.conf file.

```
Include /etc/nginx/modsec/modsecurity.conf
```

modsecurity nginx.conf main file

```
1 root@occc-VirtualBox:/etc/nginx/modsec# more main.conf
2 Include /etc/nginx/modsec/modsecurity.conf
3
4 root@occc-VirtualBox:/etc/nginx/modsec#
```

Save and close the file. We also need to copy the Unicode mapping file.

```
sudo cp /usr/local/src/ModSecurity/unicode.mapping /etc/nginx/modsec/
```

Then test Nginx configuration.

```
sudo nginx -t
```

Commands

```
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
root@occc-VirtualBox:/usr/local/src/nginx-1.18.0#
```

If the test is successful, restart Nginx.

```
sudo systemctl restart nginx
```

Enable OWASP Core Rule Set

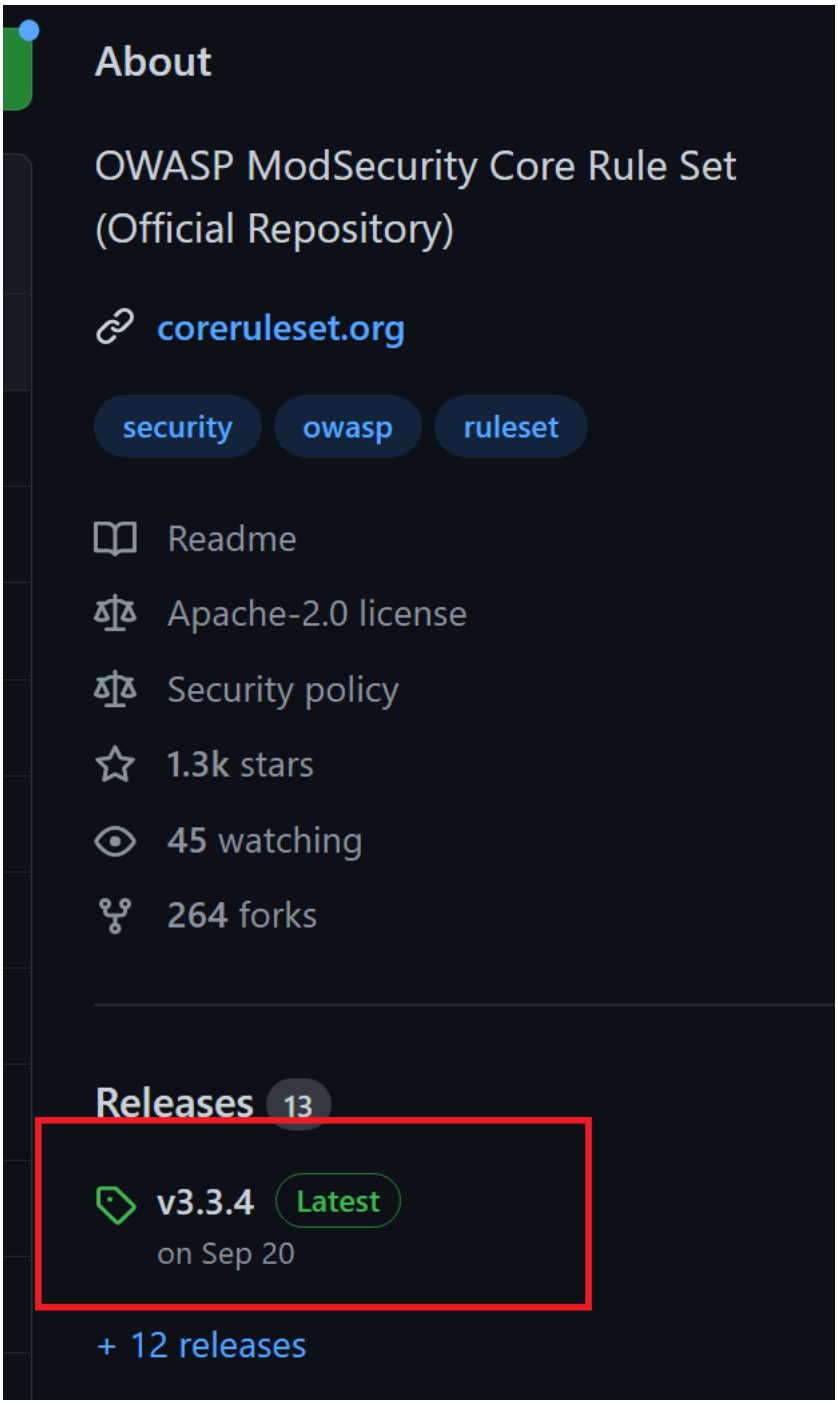
https://coreruleset.org/

To make ModSecurity protect your web applications, you need to define rules to detect malicious actors and block them. For beginners, it's a good idea to install existing rule sets, so you can get started quickly and then learn the nitty-gritty down the road. There are several free rule sets for ModSecurity. The OWASP Core Rule Set (CRS) is the standard rule set used with ModSecurity.
The CRS provides protection against many common attack categories, including:

SQL Injection (SQLi)	**HTTPoxy**
Cross Site Scripting (XSS	**Shellshock**
Local File Inclusion (LFI)	**Unix/Windows Shell Injection**
Remote File Inclusion (RFI)	**Session Fixation**
PHP Code Injection	**Scripting/Scanner/Bot Detection**
Java Code Injection	**Metadata/Error Leakages**

Download the latest OWASP CRS from GitHub.

https://github.com/coreruleset/coreruleset



Download the latest OWASP CRS from GitHub.

```
wget https://github.com/coreruleset/coreruleset/archive/v3.3.4.tar.gz
```

Extract the file.

```
tar xvf v3.3.4.tar.gz
```

Move the directory to /etc/nginx/modsec/.

```
sudo mv coreruleset-3.3.4/ /etc/nginx/modsec/
```

Rename the crs-setup.conf.example file.

```
sudo mv /etc/nginx/modsec/coreruleset-3.3.4/crs-setup.conf.example
/etc/nginx/modsec/coreruleset-3.3.4/crs-setup.conf
```

Then edit the main configuration file.

```
sudo vi /etc/nginx/modsec/main.conf
```

Add the following two lines, which will make Nginx include the CRS config file and individual rules.

```
Include /etc/nginx/modsec/coreruleset-3.3.4/crs-setup.conf
Include /etc/nginx/modsec/coreruleset-3.3.4/rules/*.conf
```

Save and close the file. Then test Nginx configuration.

```
sudo nginx -t
```

If the test is successful, restart Nginx.

```
sudo systemctl restart nginx
```

```
root@occc-VirtualBox:/etc/nginx/modsec/coreruleset-3.3.4# systemctl start nginx.service
root@occc-VirtualBox:/etc/nginx/modsec/coreruleset-3.3.4# systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-01-05 12:34:53 EST; 11s ago
       Docs: man:nginx(8)
    Process: 3478 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCE>
    Process: 3479 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3480 (nginx)
      Tasks: 2 (limit: 9457)
     Memory: 20.6M
        CPU: 135ms
     CGroup: /system.slice/nginx.service
             ├─3480 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3481 "nginx: worker process" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" ">

Jan 05 12:34:53 occc-VirtualBox systemd[1]: Starting A high performance web server and a reverse proxy server...
Jan 05 12:34:53 occc-VirtualBox systemd[1]: Started A high performance web server and a reverse proxy server.
lines 1-16/16 (END)
```

Learn How OWASP CRS Works

Let's take a look at the CRS config file, which provides you with good documentation on how CRS works.
You can see that OWASP CRS can run in two modes:

- **self-contained mode.** This is the traditional mode used in CRS v2.x. If an HTTP request matches a rule, ModSecurity will block the HTTP request immediately and stop evaluating remaining rules.

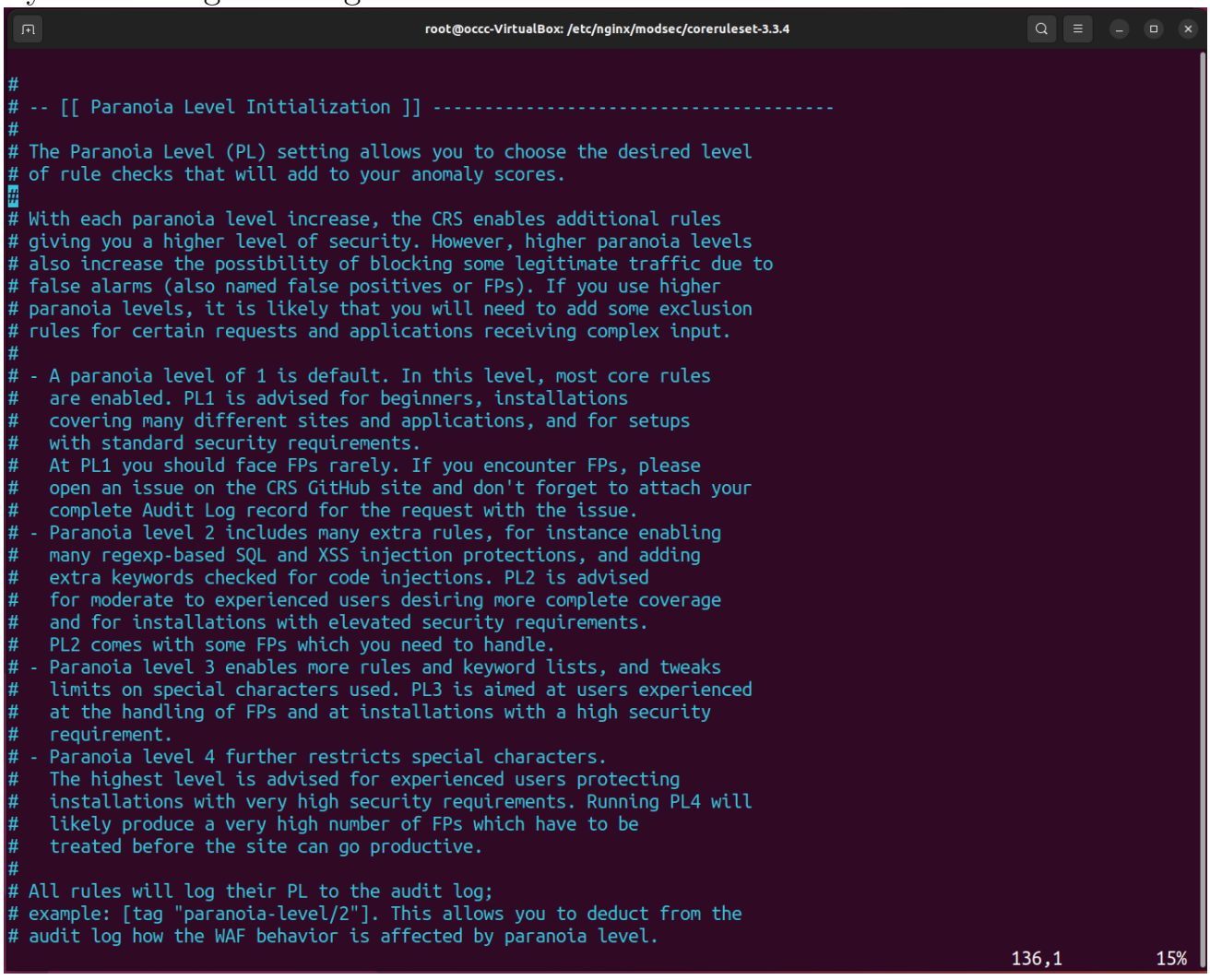
- **anomaly scoring mode.** his is the default mode used in CRS v3.x. ModSecurity will check an HTTP request against all rules, and add a score to each matching rule. If a threshold is reached, then the HTTP request is considered an attack and will be blocked. The default score for inbound requests is 5 and for outbound response is 4.

```
root@occc-VirtualBox: /etc/nginx/modsec/coreruleset-3.3.4
#
# -- [[ Mode of Operation: Anomaly Scoring vs. Self-Contained ]] ---------------
#
# The CRS can run in two modes:
#
# -- [[ Anomaly Scoring Mode (default) ]] --
# In CRS3, anomaly mode is the default and recommended mode, since it gives the
# most accurate log information and offers the most flexibility in setting your
# blocking policies. It is also called "collaborative detection mode".
# In this mode, each matching rule increases an 'anomaly score'.
# At the conclusion of the inbound rules, and again at the conclusion of the
# outbound rules, the anomaly score is checked, and the blocking evaluation
# rules apply a disruptive action, by default returning an error 403.
#
# -- [[ Self-Contained Mode ]] --
# In this mode, rules apply an action instantly. This was the CRS2 default.
# It can lower resource usage, at the cost of less flexibility in blocking policy
# and less informative audit logs (only the first detected threat is logged).
# Rules inherit the disruptive action that you specify (i.e. deny, drop, etc).
# The first rule that matches will execute this action. In most cases this will
# cause evaluation to stop after the first rule has matched, similar to how many
# IDSs function.
#
                                                                69,1          6%
```

When running in anomaly scoring mode, there are 4 paranoia levels.

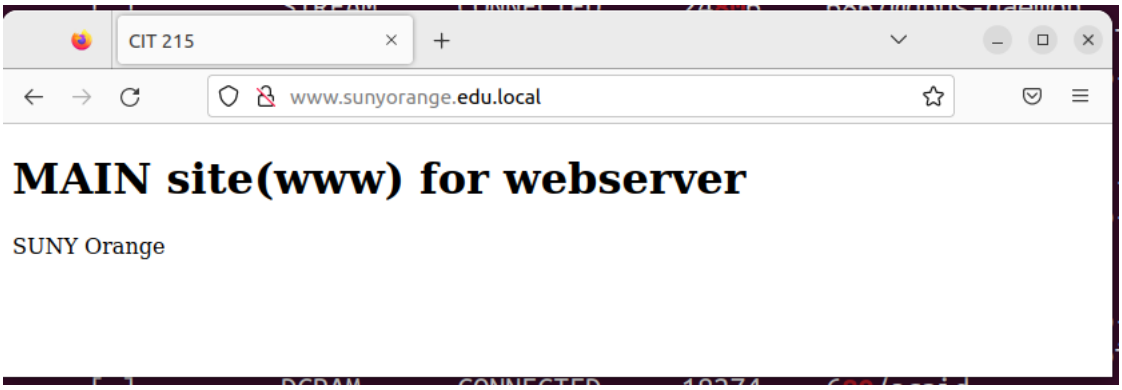
- Paranoia level 1 (default)
- Paranoia level 2
- Paranoia level 3
- Paranoia level 4

With each paranoia level increase, the CRS enables additional rules giving you a higher level of security. However, higher paranoia levels also increase the possibility of blocking some legitimate traffic due to false alarms.



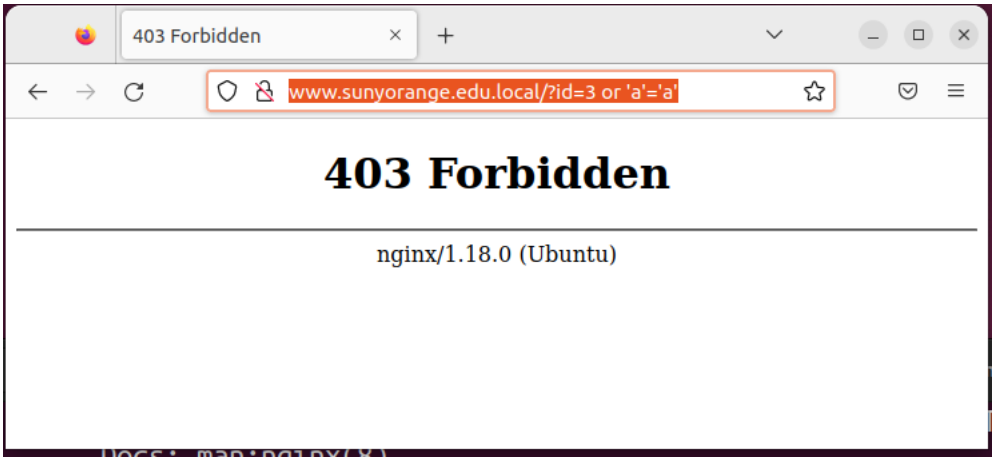
Testing

Make sure our regular website still works fine.



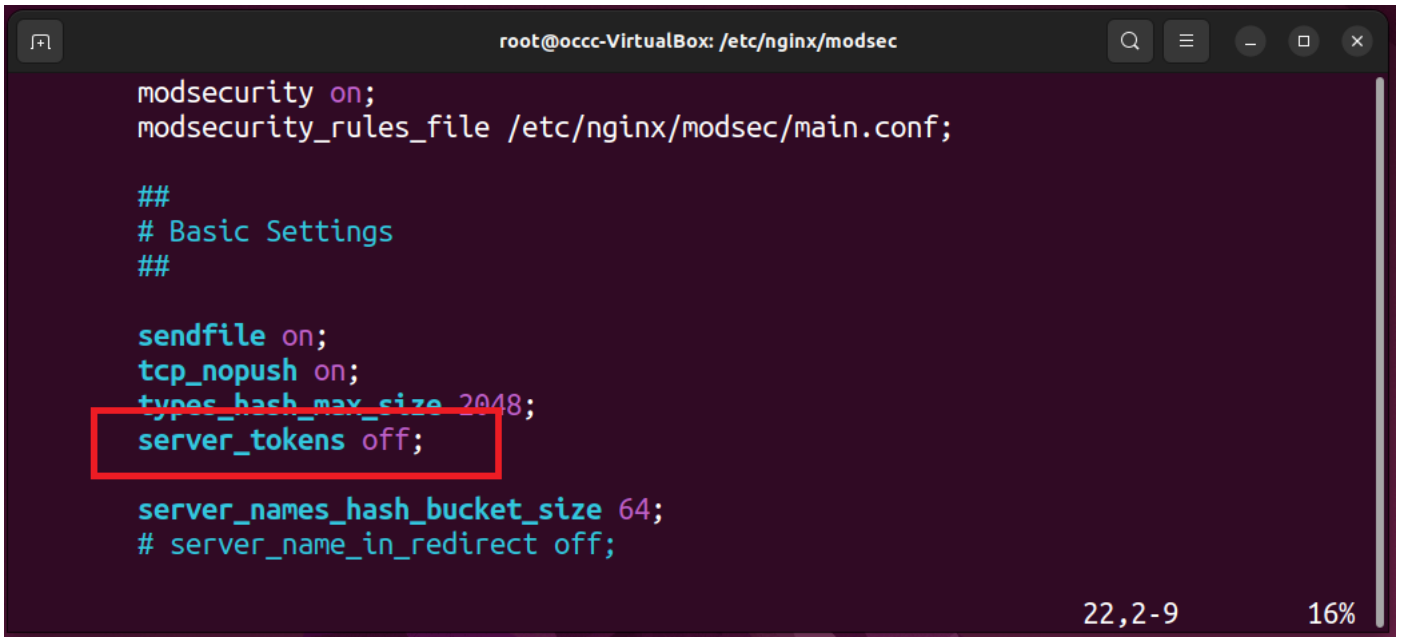
go to the url and add

http://www.sunyorange.edu.local/?id=3 or 'a'='a'



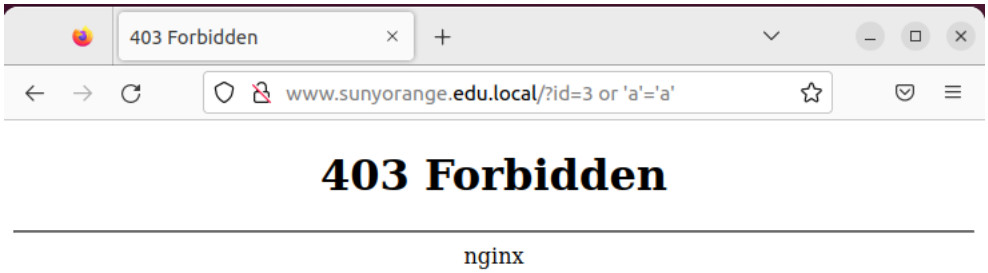
More security needs to be added

If you notice from the 403 forbidden pic from above we still have way too much information displayed.
For instance the version of our nginx webserver and the operating system that it is running on.
in /etc/nginx/nginx.conf uncomment server_tokens off;



restart nginx

```
systemctl restart nginx.service
```



Unfortunately with the free version this is as far as we can go. Commercial version allows us to add complete custom token

Remove version from server header and error pages We can change to the following values to enable or disables emitting nginx version:

- **on** Show version number.
- **off** Turn off displaying version number.
- **build** Make sure we emitt a build name along with nginx version. You must have the Nginx version 1.11.10.

- **string** Only works with commercial subscription, starting from version 1.9.13 the signature on error pages and the "Server" response header field value can be set explicitly using the string with variables. An empty string disables the emission of the "Server" field.

checking logs for modsec

Logs for mod security are in /var/logs note they are not unded the nginx directory.

modsecurity nginx.conf main file

```
1  root@occc-VirtualBox:/var/log# more modsec_audit.log
2  ---CZyd09eR---A--
3  [05/Jan/2023:12:41:35 -0500] 167294049573.090553 127.0.0.1 33266 127.0.0.1 80
4  ---CZyd09eR---B--
5  GET /?id=3%20or%20%27a%27=%27a%27 HTTP/1.1
6  Host: www.sunyorange.edu.local
7  User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/
8  108.0
9  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*
10 /*;q=0.8
11 Accept-Language: en-US,en;q=0.5
12 Accept-Encoding: gzip, deflate
13 Connection: keep-alive
14 Upgrade-Insecure-Requests: 1
15
16 ---CZyd09eR---D--
17
18 ---CZyd09eR---E--
19 <!DOCTYPE html>\x0a<html lang="en">\x0a<head>\x0a<meta charset="UTF-8">\x0a<meta name
20 ="viewport" content="width=device-width, initial-scale=1.0">\x0a<title>CIT 215</title
21 >\x0a</head>\x0a<body>\x0a\x0a<h1>MAIN site(www) for webserver</h1>\x0a<p>SUNY Orange
```

```
</p>\x0a\x09\x0a</body>\x0a</html>\x0a

---CZyd09eR---F--
HTTP/1.1 200
Server: nginx/1.18.0
Date: Thu, 05 Jan 2023 17:41:35 GMT
Content-Length: 248
Content-Type: text/html
Last-Modified: Sat, 19 Nov 2022 16:03:52 GMT
Connection: keep-alive
ETag: "6378fe68-f8"

---CZyd09eR---H--
ModSecurity: Warning. detected SQLi using libinjection. [file "/etc/nginx/modsec/core ruleset-3.3.4/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "46"] [id "940100"] [rev ""] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: 1&sos found within ARGS:id: 3 or 'a'='a'"] [severity "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [hostname "127.0.0.1"] [uri "/"] [unique_id "167294049573.090553"] [ref "v9,12"]
ModSecurity: Warning. Matched "Operator `Ge' with parameter `5' against variable `TX:ANOMALY_SCORE' (Value: `5' ) [file "/etc/nginx/modsec/coreruleset-3.3.4/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "81"] [id "949110"] [rev ""] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [data ""] [severity "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "127.0.0.1"] [uri "/"] [unique_id "167294049573.090553"] [ref ""]

```

```
---CZyd09eR---I--

---CZyd09eR---J--

---CZyd09eR---Z--

---vfts5dnW---A--
[05/Jan/2023:12:47:21 -0500] 167294084126.591432 ↙
   ↪  127.0.0.1 57688 127.0.0.1 80
---vfts5dnW---B--
GET /?id=3%20or%20%27a%27=%27a%27 HTTP/1.1
Host: www.sunyorange.edu.local
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64;↙
   ↪   rv:108.0) Gecko/20100101 Firefox/
108.0
Accept: text/html,application/xhtml+xml,application↙
   ↪  /xml;q=0.9,image/avif,image/webp,*
/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

---vfts5dnW---E--
<html>\x0d\x0a<head><title>403 Forbidden</title></↙
   ↪  head>\x0d\x0a<body>\x0d\x0a<center>
<h1>403 Forbidden</h1></center>\x0d\x0a<hr><center>↙
   ↪  nginx/1.18.0 (Ubuntu)</center>\x0d
\x0a</body>\x0d\x0a</html>\x0d\x0a

---vfts5dnW---F--
HTTP/1.1 403
Server: nginx/1.18.0
Date: Thu, 05 Jan 2023 17:47:21 GMT
Content-Length: 162
Content-Type: text/html
Connection: keep-alive

```

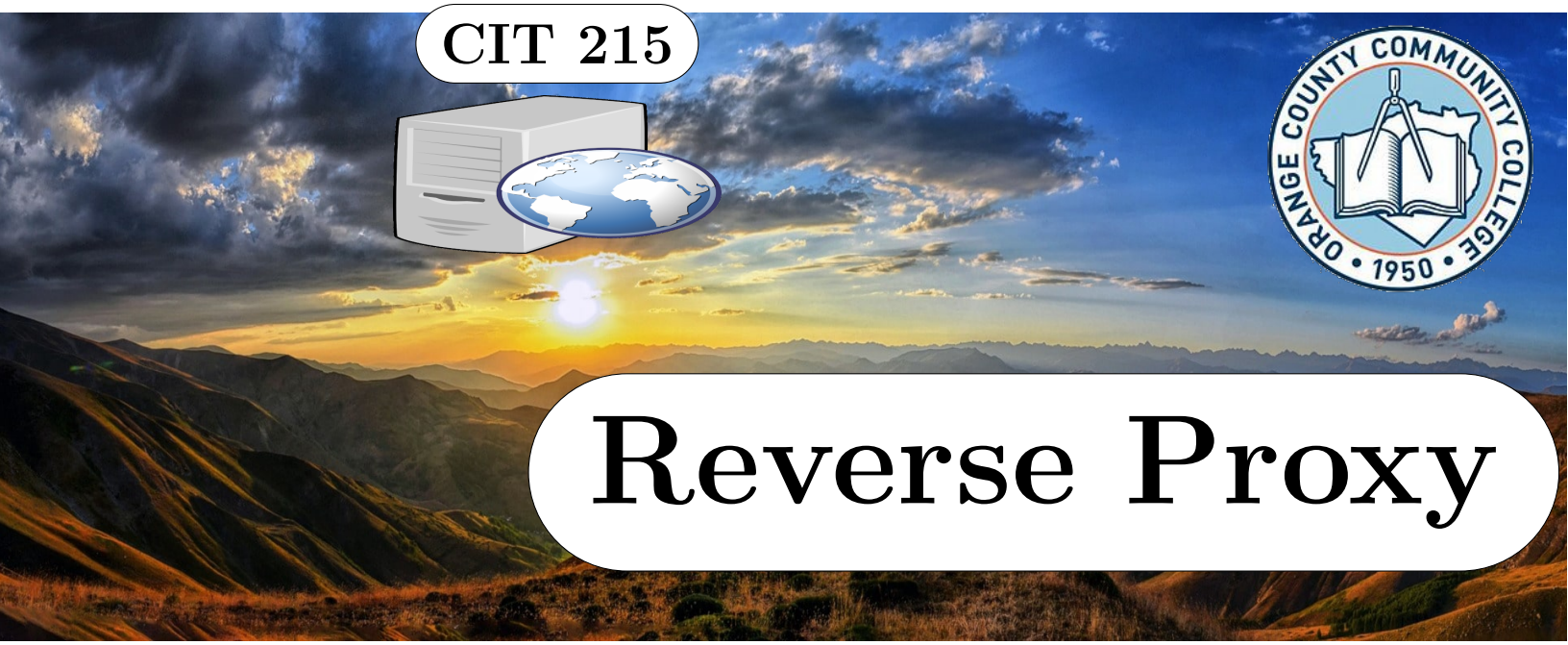
```
---vfts5dnW---H--
ModSecurity: Warning. detected SQLi using libinjection. [file "/etc/nginx/modsec/coreruleset-3.3.4/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "46"] [id "942100"] [rev ""] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: 1&sos found within ARGS:id: 3 or 'a'='a'"] [severity "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [hostname "127.0.0.1"] [uri "/"] [unique_id "167294084126.591432"] [ref "v9,12"]
ModSecurity: Access denied with code 403 (phase 2). Matched "Operator `Ge' with parameter `5' against variable `TX:ANOMALY_SCORE' (Value: `5' ) [file "/etc/nginx/modsec/coreruleset-3.3.4/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "81"] [id "949110"] [rev ""] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [data ""] [severity "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "127.0.0.1"] [uri "/"] [unique_id "167294084126.591432"] [ref ""]

---vfts5dnW---J--

---vfts5dnW---K--

---vfts5dnW---Z--

---I0wnnT8o---A--
[05/Jan/2023:12:57:31 -0500] 167294145127.787035 127.0.0.1 59990 127.0.0.1 80
---I0wnnT8o---B--
GET /?id=3%20or%20%27a%27=%27a%27 HTTP/1.1
```

```
Host: www.sunyorange.edu.local
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1

---I0wnnT8o---E--
<html>\x0d\x0a<head><title>403 Forbidden</title></head>\x0d\x0a<body>\x0d\x0a<center><h1>403 Forbidden</h1></center>\x0d\x0a<hr><center>nginx</center>\x0d\x0a</body>\x0d\x0a</html>\x0d\x0a

---I0wnnT8o---F--
HTTP/1.1 403
Server: nginx
Date: Thu, 05 Jan 2023 17:57:31 GMT
Content-Length: 146
Content-Type: text/html
Connection: keep-alive

---I0wnnT8o---H--
ModSecurity: Warning. detected SQLi using libinjection. [file "/etc/nginx/modsec/coreruleset-3.3.4/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "46"] [id "940100"] [rev ""] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: 1&sos found within ARGS:id: 3 or 'a'='a'"] [severity "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [hostname "127.0.0.1"] [uri "/"] [unique_id "167294145
```

```
127.787035"] [ref "v9,12"]
ModSecurity: Access denied with code 403 (phase 2). Matched "Operator `Ge' with param
eter `5' against variable `TX:ANOMALY_SCORE' (Value: `5' ) [file "/etc/nginx/modsec/c
oreruleset-3.3.4/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "81"] [id "949110
"] [rev ""] [msg "Inbound Anomaly Score Exceeded (Total Score: 5)"] [data ""] [severi
ty "2"] [ver "OWASP_CRS/3.3.4"] [maturity "0"] [accuracy "0"] [tag "application-multi
"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "12
7.0.0.1"] [uri "/"] [unique_id "167294145127.787035"] [ref ""]

---I0wnnT8o---J--

---I0wnnT8o---K--

---I0wnnT8o---Z--

root@occc-VirtualBox:/var/log#
```

9.3.2 PHP security

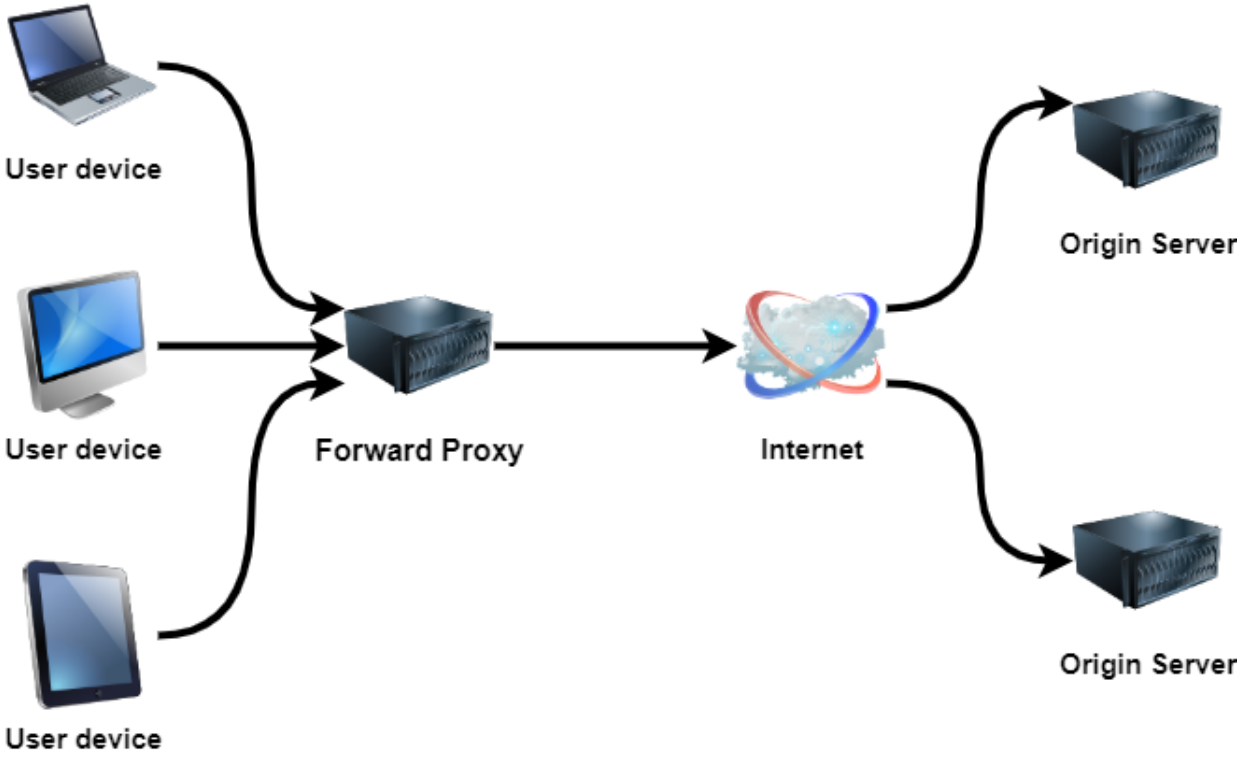
https://www.virtuozzo.com/application-platform-docs/php-security-settings/
https://cheatsheetseries.owasp.org/cheatsheets/PHP_Configuration_Cheat_Sheet.htm



Reverse Proxy

Proxy

A forward proxy, often called a proxy, proxy server, or web proxy, is a server that sits in front of a group of client machines. When those computers make requests to sites and services on the Internet, the proxy server intercepts those requests and then communicates with web servers on behalf of those clients, like a middleman.



Uses:

- **To avoid state or institutional browsing restrictions** Some governments,

schools, and other organizations use firewalls to give their users access to a limited version of the Internet. A forward proxy can be used to get around these restrictions, as they let the user connect to the proxy rather than directly to the sites they are visiting.

- **To block access to certain content** Conversely, proxies can also be set up to block a group of users from accessing certain sites. For example, a school network might be configured to connect to the web through a proxy which enables content filtering rules, refusing to forward responses from Facebook and other social media sites.

- **To protect their identity online** In some cases, regular Internet users simply desire increased anonymity online, but in other cases, Internet users live in places where the government can impose serious consequences to political dissidents. Criticizing the government in a web forum or on social media can lead to fines or imprisonment for these users. If one of these dissidents uses a forward proxy to connect to a website where they post politically sensitive comments, the IP address used to post the comments will be harder to trace back to the dissident. Only the IP address of the proxy server will be visible.

Reverse Proxy

A proxy server is a go-between or intermediary server that forwards requests for content from multiple clients to different servers across the Internet. A **reverse proxy server** is a type of proxy server that typically sits behind the firewall in a private network and directs client requests to the appropriate backend server. A reverse proxy provides an additional level of abstraction and control to ensure the smooth flow of network traffic between clients and servers.

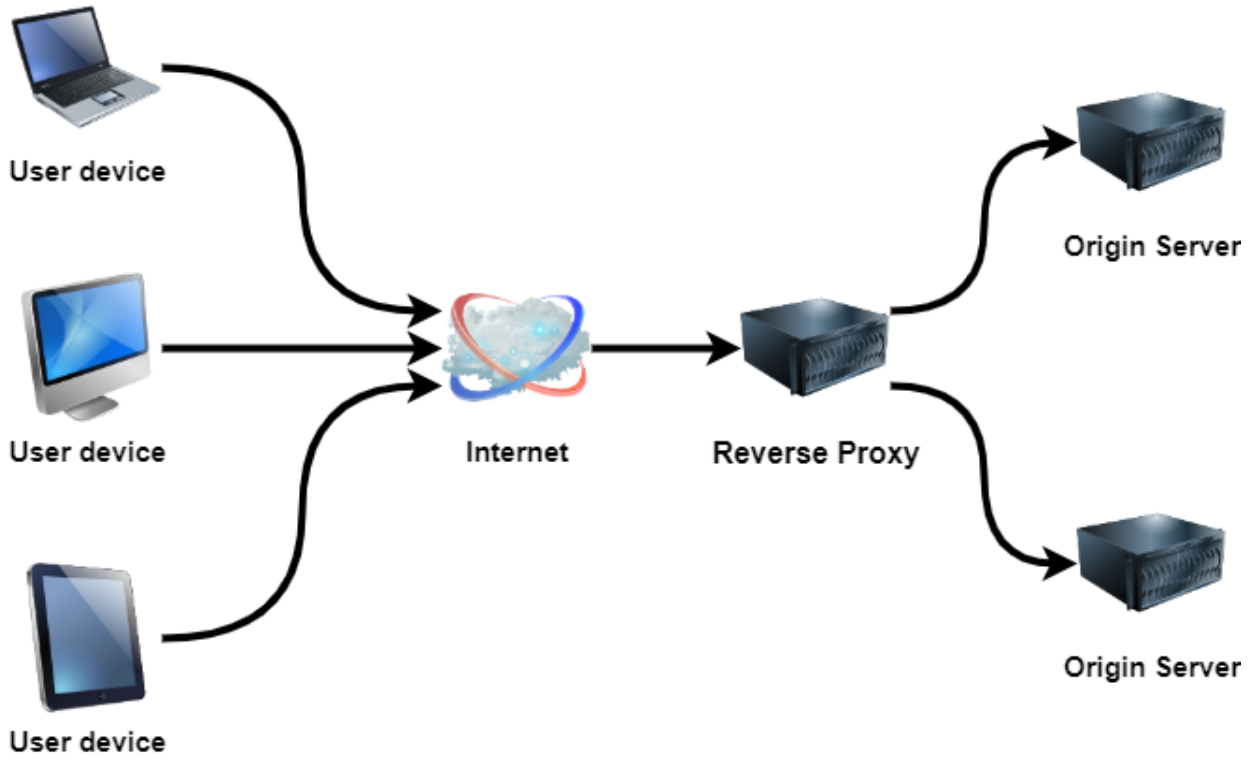
Benefits of a reverse proxy:

- **Load balancing** A popular website that gets millions of users every day may not be able to handle all of its incoming site traffic with a single origin server. Instead, the site can be distributed among a pool of different servers, all handling requests for the same site. In this case, a reverse proxy can provide a load balancing solution which will distribute the incoming traffic evenly among the different servers to prevent any single server from becoming overloaded. In the event that a server fails completely, other servers can step up to handle the traffic.

- **Protection from attacks** With a reverse proxy in place, a web site or service never needs to reveal the IP address of their origin server(s). This makes it much harder for attackers to leverage a targeted attack against them, such as a DDoS attack. Instead the attackers will only be able to target the reverse proxy, such as Cloudflare's CDN, which will have tighter security and more resources to fend off a cyber attack. Global Server Load Balancing (GSLB) - In this form of load balancing, a website can be distributed on several servers around the globe and the reverse proxy will send clients to the server that's geographically closest to them. This decreases the distances that requests and responses need to travel, minimizing load times.

- **Caching** A reverse proxy can also cache content, resulting in faster performance. For example, if a user in Paris visits a reverse-proxied website with web servers in Los Angeles, the user might actually connect to a local reverse proxy server in Paris, which will then have to communicate with an origin server in L.A. The proxy server can then cache (or temporarily save) the response data. Subsequent Parisian users who browse the site will then get the locally cached version from the Parisian reverse proxy server, resulting in much faster performance.

- **SSL encryption** Encrypting and decrypting SSL (or TLS) communications for each client can be computationally expensive for an origin server. A reverse proxy can be configured to decrypt all incoming requests and encrypt all outgoing responses, freeing up valuable resources on the origin server.



10.0.1 nginx reverse proxy

10.0.1.0 nginx with node js as origin

Setting up nginx to be the front end for a nodejs server.
Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Setting up nodejs

the easiest way to install nodejs is to use nodesource from:
https://github.com/nodesource/distributions#debinstall
installation instructions are provided at the website page.

curl -fsSL https://deb.nodesource.com/setup_19.x | sudo -E bash - && sudo apt-get install -y nodejs

Commands

```
occc@occc-VirtualBox:~$ curl -fsSL https://deb.nodesource.com/setup_19.x | sudo -
E bash - &&\
sudo apt-get install -y nodejs
[sudo] password for occc:

## Installing the NodeSource Node.js 19.x repo...


## Populating apt-get cache...

+ apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
...
Get:41 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-
11 Metadata [13.3 kB]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-
f Metadata [11.0 kB]
Fetched 7,727 kB in 2s (4,557 kB/s)
Reading package lists... Done

## Confirming "jammy" is supported...

+ curl -sLf -o /dev/null 'https://deb.nodesource.com/node_19.x/dists/jammy/Release'

## Adding the NodeSource signing key to your keyring...

+ curl -s https://deb.nodesource.com/gpgkey/nodesource.gpg.key | gpg --
dearmor | tee /usr/share/keyrings/nodesource.gpg >/dev/null
gpg: WARNING: unsafe ownership on homedir '/home/occc/.gnupg'

## Creating apt sources list file for the NodeSource Node.js 19.x repo...

+ echo 'deb [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_19.x jar
+ echo 'deb-src [signed-by=/usr/share/keyrings/nodesource.gpg] https://deb.nodesource.com/node_19.x

## Running `apt-get update` for you...

+ apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://deb.nodesource.com/node_19.x jammy InRelease [4,563 B]
Hit:5 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:6 https://deb.nodesource.com/node_19.x jammy/main amd64 Packages [773 B]
Fetched 5,336 B in 1s (7,776 B/s)
Reading package lists... Done

## Run `sudo apt-get install -y nodejs` to install Node.js 19.x and npm
## You may also need development tools to build native addons:
sudo apt-get install gcc g++ make
## To install the Yarn package manager, run:
```

```
curl -sL https://dl.yarnpkg.com/debian/pubkey.gpg | gpg --dearmor | sudo tee /usr/share/keyrings/y
echo "deb [signed-by=/usr/share/keyrings/yarnkey.gpg] https://dl.yarnpkg.com/debian stable main" |
sudo apt-get update && sudo apt-get install yarn


Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
libflashrom1 libftdi1-2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
nodejs
0 upgraded, 1 newly installed, 0 to remove and 73 not upgraded.
Need to get 28.9 MB of archives.
After this operation, 188 MB of additional disk space will be used.
Get:1 https://deb.nodesource.com/node_19.x jammy/main amd64 nodejs amd64 19.3.0-deb-
1nodesource1 [28.9 MB]
Fetched 28.9 MB in 1s (30.6 MB/s)
Selecting previously unselected package nodejs.
(Reading database ... 213985 files and directories currently installed.)
Preparing to unpack .../nodejs_19.3.0-deb-1nodesource1_amd64.deb ...
Unpacking nodejs (19.3.0-deb-1nodesource1) ...
Setting up nodejs (19.3.0-deb-1nodesource1) ...
Processing triggers for man-db (2.10.2-1) ...
occc@occc-VirtualBox:~$
```

test that node is installed

occc@occc-VirtualBox: $ node -v
v19.3.0
occc@occc-VirtualBox: $

test that npm is installed as well

occc@occc-VirtualBox: $ npm -v
9.2.0
occc@occc-VirtualBox: $

setup a basic node project

Commands

```
mkdir nginx_server_project
cd nginx_server_project
```

```
npm init -y
```

Commands

```
occc@occc-VirtualBox:/webroot/cit215$ sudo su -
[sudo] password for occc:
root@occc-VirtualBox:~# cd /webroot/cit215/
root@occc-VirtualBox:/webroot/cit215# mkdir nginx_server_project
cd nginx_server_project
npm init -y
Wrote to /webroot/cit215/nginx_server_project/package.json:

{
"name": "nginx_server_project",
"version": "1.0.0",
"description": "",
"main": "index.js",
"scripts": {
"test": "echo \"Error: no test specified\" && exit 1"
},
"keywords": [],
"author": "",
"license": "ISC"
}


root@occc-VirtualBox:/webroot/cit215/nginx_server_project#
```

The above code will create the folder nginx_server_project and change the directory into the folder. We then initialize a Node.js application with npm, using the -y flag to set yes as the default answer to all the questions.
The next step is to create the server.js file that contains the source code for our application.

server.js

```
1 const http = require("http");
2
3 const server = http.createServer((req, res) => {
4     const urlPath = req.url;
```

```
    if (urlPath === "/overview") {
        res.end('Welcome to the "cit215 overview page" of the nginX reverse proxy');
    } else if (urlPath === "/api") {
        res.writeHead(200, { "Content-Type": "application/json" });
        res.end(
        JSON.stringify({
            product_id: "sunyorange",
            product_name: "CIT 215",
        })
        );
    } else {
        res.end("Successfully started a server");
    }
});

server.listen(3000, "localhost", () => {
    console.log("Listening for request");
});
```

We created a server with a Node.js HTTP module that we imported using the require function in the above code. Within our server, we'll render two different responses, depending on our current route. The two routes are /overview and /api.

On the /overview subdomain, we'll render a plain text, while on the /api, we'll render a JSON object. The above application will be accessed on the localhost 127.0.0.1 of your virtual machine on port 3000.

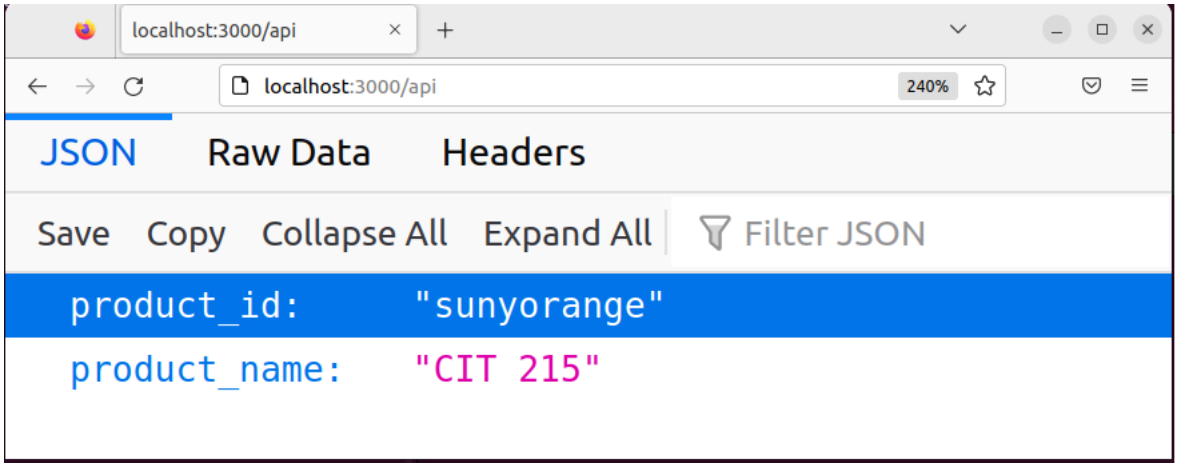
Start your Node.js server application using the following command:

```
node server.js
```

test that nodejs works

```
http://localhost:3000
```

http://localhost:3000/api



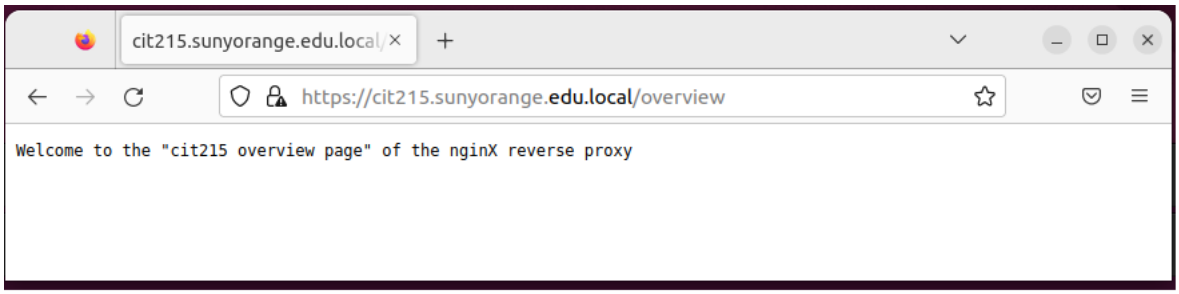
Nodejs is working but it is bound to port 3000 on the 127.0.0.1 IP. we can change the ip and port bindings but exposing the nodejs server to the internet is not advisable. We already have a WAF firewall setup for our nginx to protect it from basic malicious attacks. So lets use that.
Go back to nginx and edit the cit215.edu config file.

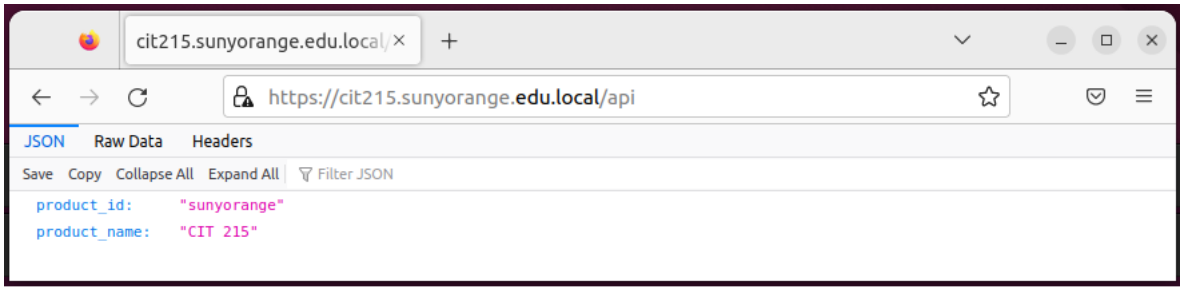
nginx reverse proxy for nodejs

```
1 server {
2     listen 443 ssl http2;
3     listen [::]:443 ssl http2;
4     ssl_certificate /etc/ssl/certs/cit215.crt;
5     ssl_certificate_key /etc/ssl/private/cit215.key
        ↪ ;
6     ssl_session_timeout 1d;
7     ssl_session_cache shared:MozSSL:10m;  # about
        ↪ 40000 sessions
8     ssl_session_tickets off;
9
```

```
    # curl https://ssl-config.mozilla.org/ffdhe2048.txt > /path/to/dhparam
    ssl_dhparam /etc/ssl/certsdhparam.pem;

    # intermediate configuration
    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA
    -AES256-GCM-SHA384:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256
    -GCM-SHA384;
    ssl_prefer_server_ciphers off;
    # HSTS (ngx_http_headers_module is required) (63072000 seconds)
    add_header Strict-Transport-Security "max-age=63072000" always;

    root /webroot/cit215;
    # Add index.php to setup Nginx, PHP & PHP-FPM config
    index index.php index.html index.htm index.nginx-debian.html;

    server_name cit215.sunyorange.edu.local;

    location / {
        proxy_pass http://localhost:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }


    # pass PHP scripts on Nginx to FastCGI (PHP-FPM
```

```
        ↪  ) server
    location ~ \.php$ {
        include snippets/fastcgi-php.conf;

        # Nginx php-fpm sock config:
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;
        # Nginx php-cgi config :
        # Nginx PHP fastcgi_pass 127.0.0.1:9000;
    }

    # deny access to Apache .htaccess on Nginx with↙
        ↪  PHP,
    # if Apache and Nginx document roots concur
    location ~ /\.ht {
        deny all;
    }
}
```





nginx reverse proxy for nodejs split locations

```
server_name cit215.sunyorange.edu.local;
location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a ↙
        ↪  404.
```

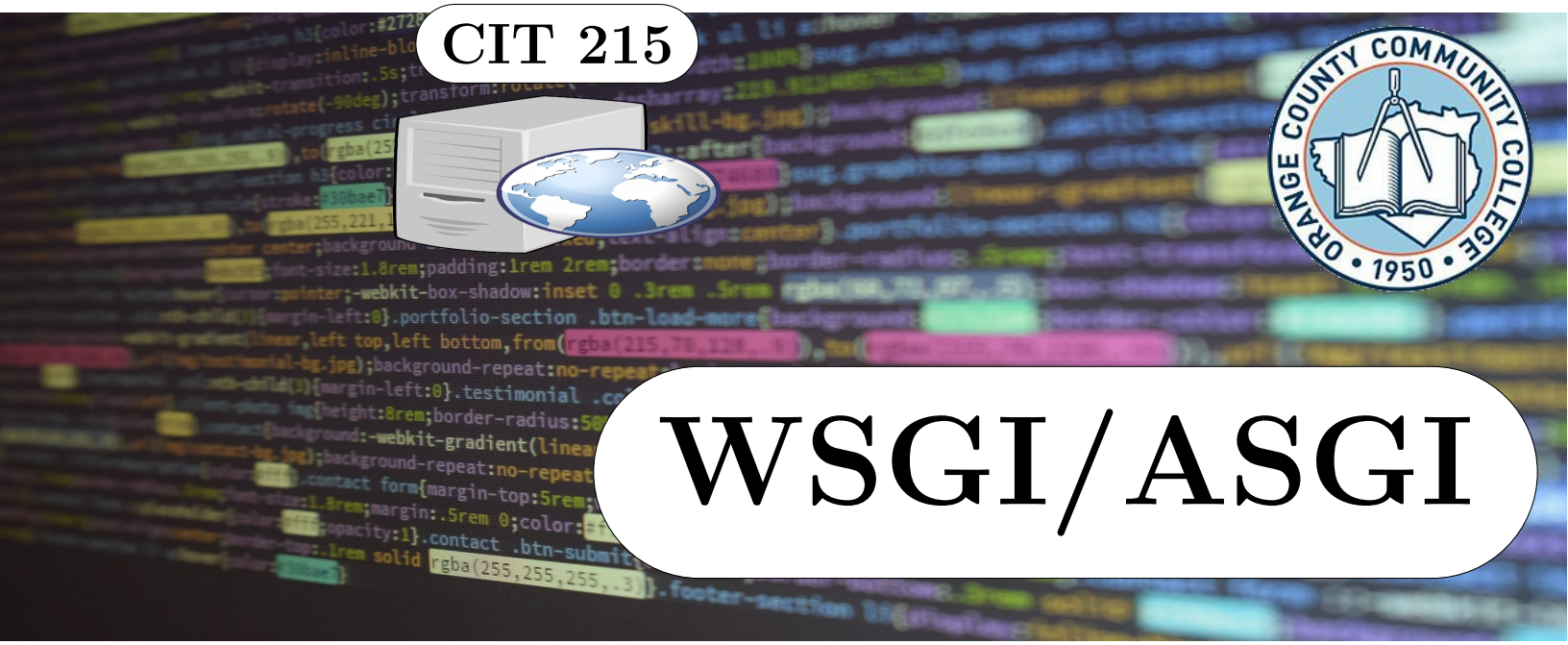
```
    try_files $uri$uri/ =404;
}

location /api/ {
    proxy_pass http://localhost:3000/api;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
location /overview/ {
    proxy_pass http://localhost:3000/overview;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
location /images/ {
    #we created a directory /webroot/images and ↙
      ↪  added a image to it
    root   /webroot/;
}
```

10.0.2 nginx Load-Balancing

https://docs.nginx.com/nginx/admin-guide/load-balancer/http-load-balancer/

10.0.3 Apache Load-Balancing

https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html



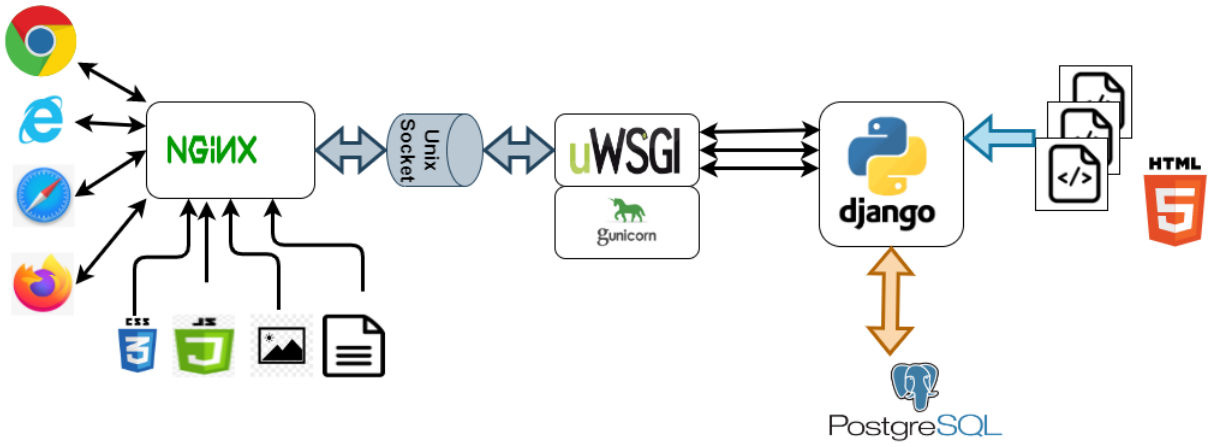
ASGI

ASGI (Asynchronous Server Gateway Interface) is a spiritual successor to WSGI, intended to provide a standard interface between async-capable Python web servers, frameworks, and applications.

Where WSGI provided a standard for synchronous Python apps, ASGI provides one for both asynchronous and synchronous apps, with a WSGI backwards-compatibility implementation and multiple servers and application frameworks.

WSGI

WSGI stands for "Web Server Gateway Interface". It is used to forward requests from a web server (such as Apache or NGINX) to a backend Python web application or framework. From there, responses are then passed back to the webserver to reply to the requestor.



https://uwsgi-docs.readthedocs.io/en/latest/tutorials/Django_and_nginx.html
https://docs.nginx.com/nginx/admin-guide/web-server/app-gateway-uwsgi-django/

Other webservers

12.1 New and emerging web protocols

12.1.1 QUIC

There are in fact two protocols that share the same name: “Google QUIC” (“gQUIC” for short), is the original protocol that was designed by Google engineers several years ago, which, after years of experimentation, has now been adopted by the IETF (Internet Engineering Task Force) for standardization.

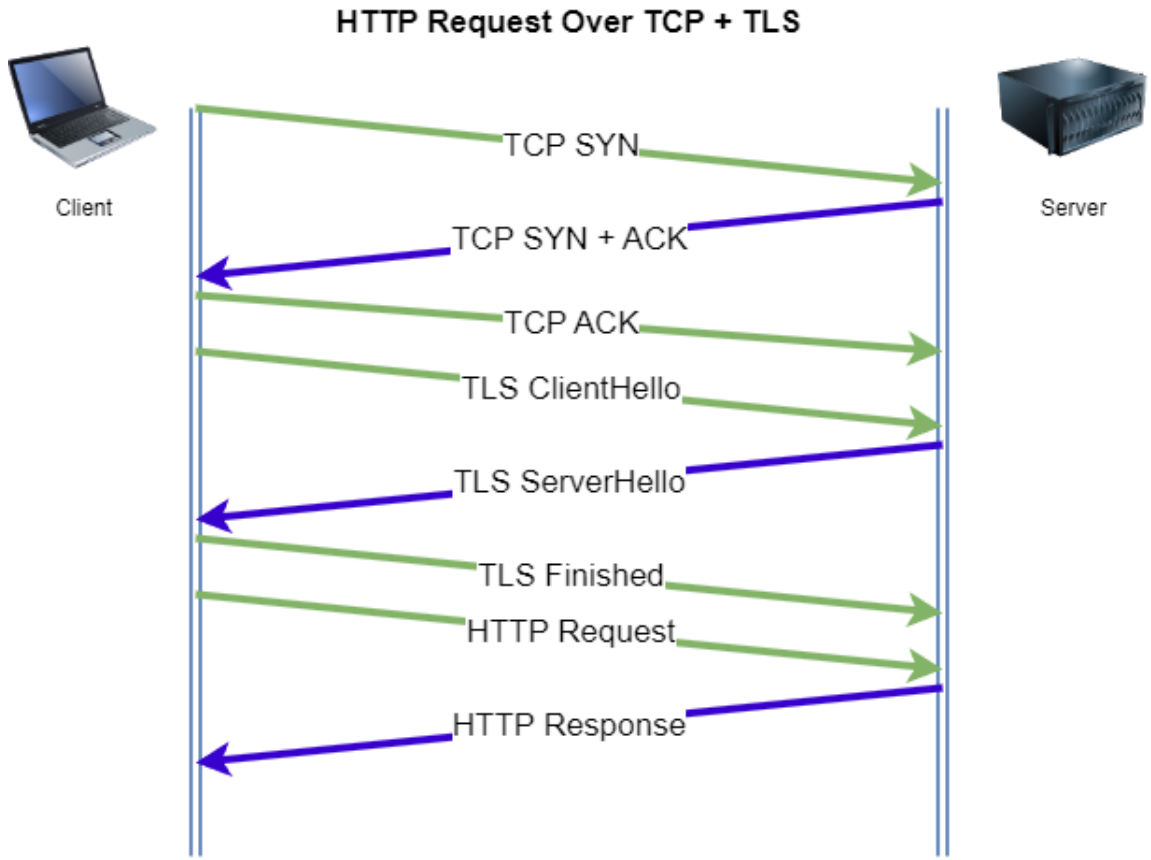
“IETF QUIC” (just “QUIC” from now on) has already diverged from gQUIC quite significantly such that it can be considered a separate protocol. From the wire format of the packets, to the handshake and the mapping of HTTP, QUIC has improved the original gQUIC design thanks to open collaboration from many organizations and individuals, with the shared goal of making the Internet faster and more secure.

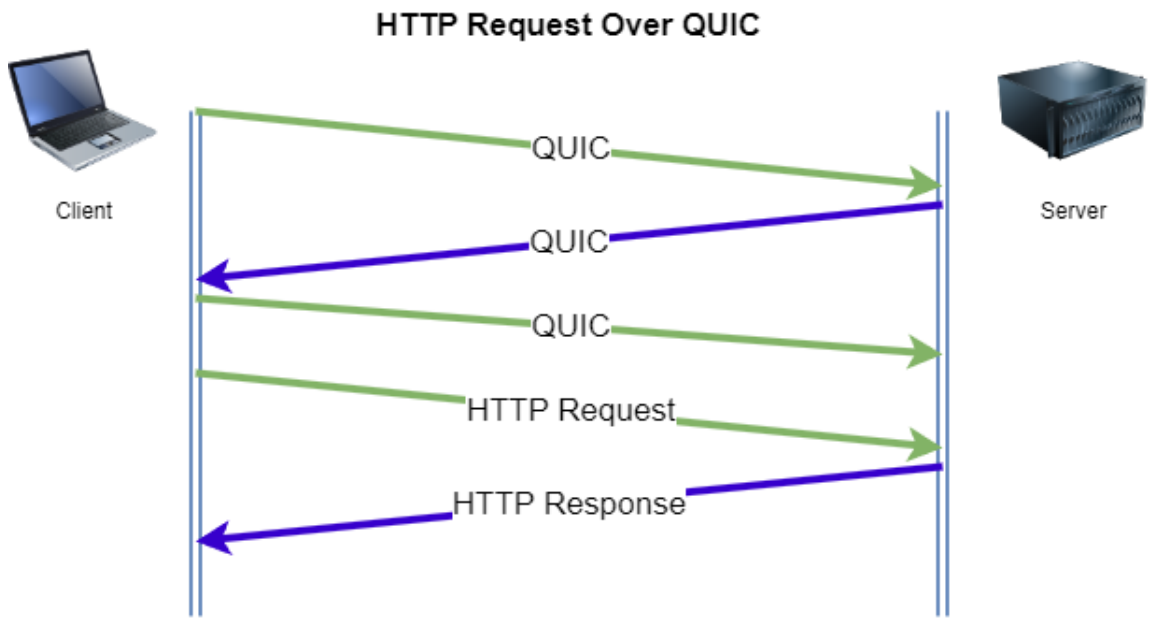
Built-in security (and performance)

One of QUIC’s more radical deviations from the now venerable TCP, is the stated design goal of providing a secure-by-default transport protocol. QUIC accomplishes this by providing security features, like authentication and encryption, that are typically handled by a higher layer protocol (like TLS), from the transport protocol itself.

The initial QUIC handshake combines the typical three-way handshake that you get with TCP, with the TLS 1.3 handshake, which provides authentication of the end-points as well as negotiation of cryptographic parameters. For those familiar with the TLS protocol, QUIC replaces the TLS record layer with its own framing format, while keeping the same TLS handshake messages.

Not only does this ensure that the connection is always authenticated and encrypted, but it also makes the initial connection establishment faster as a result: the typical QUIC handshake only takes a single round-trip between client and server to complete, compared to the two round-trips required for the TCP and TLS 1.3 handshakes combined.





But QUIC goes even further, and also encrypts additional connection metadata that could be abused by middle-boxes to interfere with connections. For example packet numbers could be used by passive on-path attackers to correlate users activity over multiple network paths when connection migration is employed (see below). By encrypting packet numbers QUIC ensures that they can't be used to correlate activity by any entity other than the end-points in the connection.

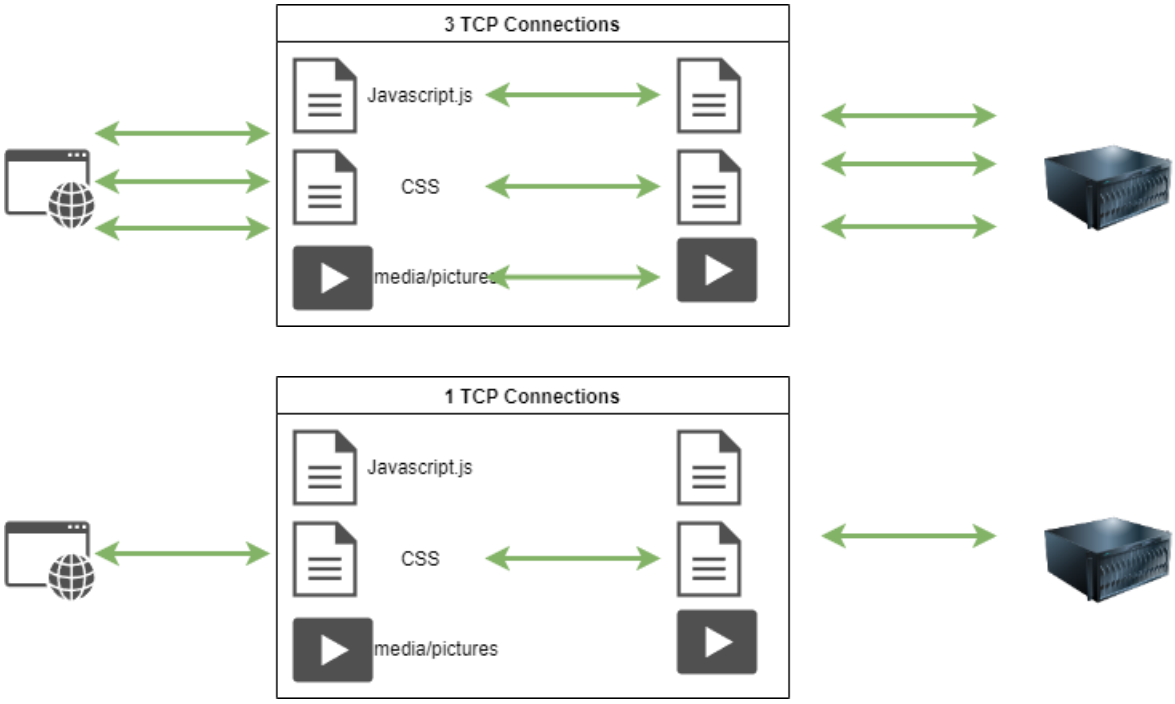
Encryption can also be an effective remedy to ossification, which makes flexibility built into a protocol (like for example being able to negotiate different versions of that protocol) impossible to use in practice due to wrong assumptions made by implementations (ossification is what delayed deployment of TLS 1.3 for so long, which was only possible after several changes, designed to prevent ossified middle-boxes from incorrectly blocking the new revision of the TLS protocol, were adopted).

Head-of-line blocking

One of the main improvements delivered by HTTP/2 was the ability to multiplex different HTTP requests onto the same TCP connection. This allows HTTP/2 applications to process requests concurrently and better utilize the network bandwidth available to them.

This was a big improvement over the then status quo, which required applications to initiate multiple TCP+TLS connections if they wanted to process multiple HTTP/1.1 requests concurrently (e.g. when a browser needs to fetch

both CSS and Javascript assets to render a web page). Creating new connections requires repeating the initial handshakes multiple times, as well as going through the initial congestion window ramp-up, which means that rendering of web pages is slowed down. Multiplexing HTTP exchanges avoids all that.

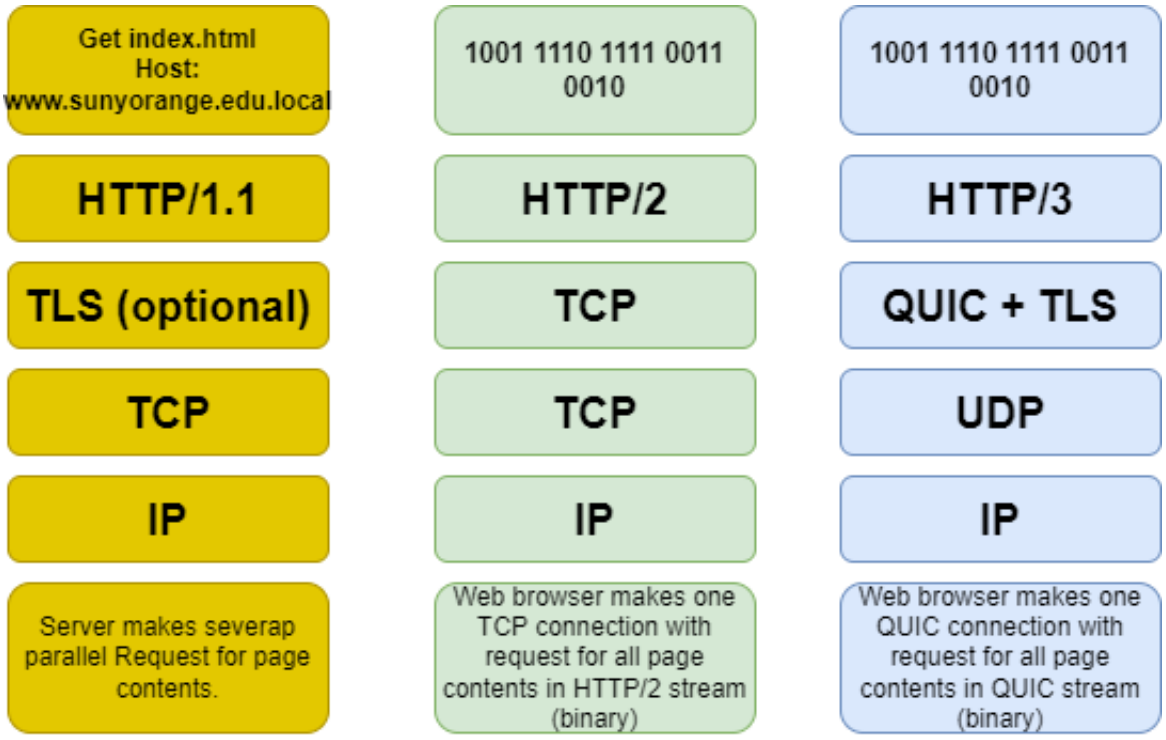


Some problems with QUIC is NAT routers. Most NAT devices do not support QUIC and have to fall back to old UDP NAT which can cause timeouts.
There might be also issue with openssl nad libressl since they do not support QUIC yet. BoringSSL from google does support it.

Web server support

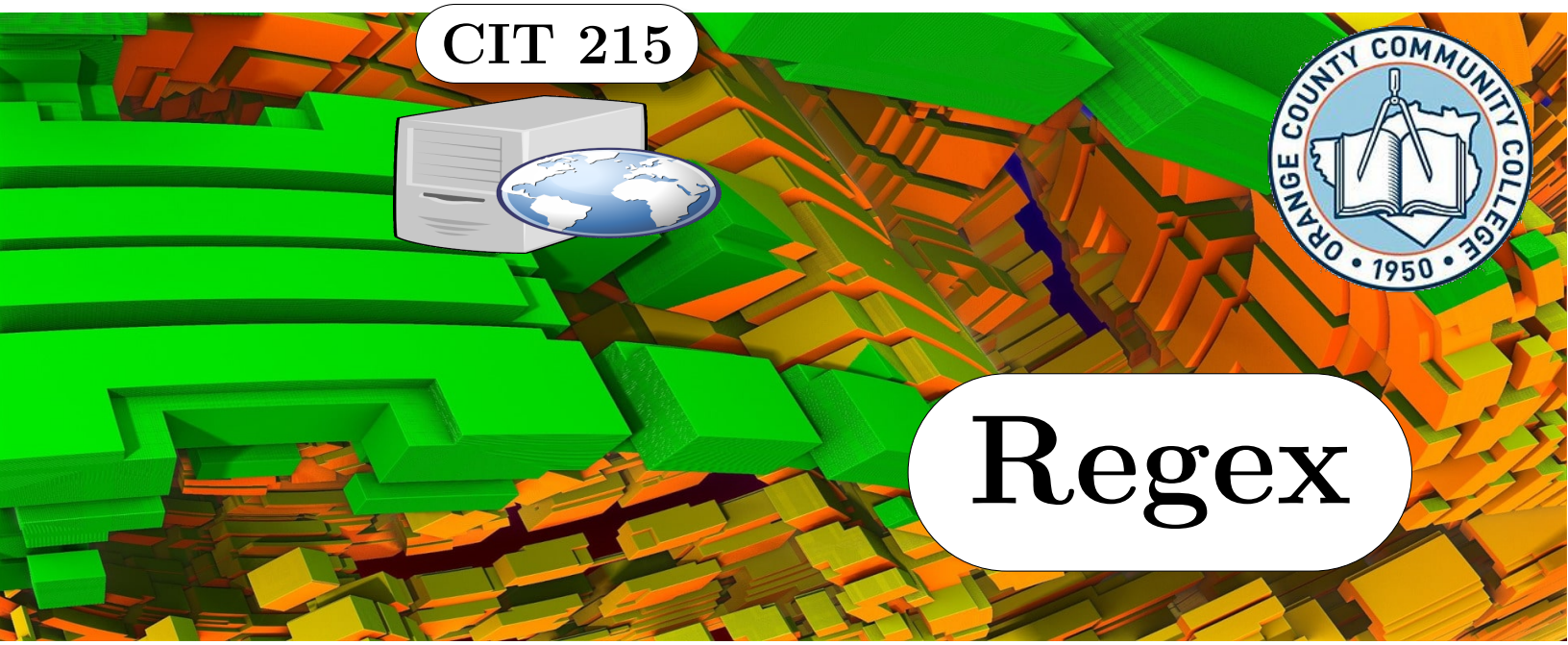
NGINX and litespeed support QUIC. Apache has only experimental not yet ready support at this time.

12.1.2 HTTP/2 and HTTP/3



12.2 litespeed

https://www.litespeedtech.com/



https://regex101.com/

13.0.1 NGINX regex

Search-Order	Modifier	Description	Match-Type	Stops-search-on-match
1st	=	The URI must match the specified pattern exactly	Simple-string	Yes
2nd	^~	The URI must begin with the specified pattern	Simple-string	Yes
3rd	(None)	The URI must begin with the specified pattern	Simple-string	No
4th	~	The URI must be a case-sensitive match to the specified Rx	Perl-Compatible-Rx	Yes (first match)
4th	~*	The URI must be a case-insensitive match to the specified Rx	Perl-Compatible-Rx	Yes (first match)
N/A	@	Defines a named location block.	Simple-string	Yes

Capturing group:

Capturing group, expression evaluation () are supported, this example **location ~ ^/(?:index|update)$** match url ending with sunyorange.edu/index and sunyorange.edu/update Note the domain name is not included in the location match and will only search on items after the domain name.

()	**Group/Capturing-group, capturing mean match and retain/output/use what matched the patern inside (). the default bracket mode is "capturing group" while (?:) is a non capturing group. example (?:a|b) match a or b in a non capturing mode**
?:	**Non capturing group**
?=	**Positive look ahead**
?!	**is for negative look ahead (do not match the following...)**
?<=	**is for positive look behind**
?<!	**is for negative look behind**

The forward slash:

Not to confuse with the regex slash `\`, In nginx the forward slash `/` is used to match any sub location including none example **`location /`**. In the context of regex support the following explanation apply

/	**It doesn't actually do anything. In Javascript, Perl and some other languages, it is used as a delimiter character explicitly for regular expressions. Some languages like PHP use it as a delimiter inside a string, with additional options passed at the end, just like Javascript and Perl. Nginx does not use delimiter, / can be escaped with \/ for code portability purpose BUT this is not required for nginx / are handled literally (don't have other meaning than /)**

The slash:

The first purpose of the regex special character is meant to escape the next character; But note that in most case followed by a character have a different meaning, a complete list is available here.
https://www.regular-expressions.info/refquick.html

Nginx does not require escaping the forward slash / it does not either deny escaping it like we could escape any other character. and thus \/ is translated/-matching /. One purpose of escaping forward slashes in the context of nginx could be for code portability.

Other regex chars:

~	Enable regex mode for location (in regex ~mean case-sensitive match)
~*	case-insensitive match
|	Or
()	Match group or evaluate the content of ()
$	the expression must be at the end of the evaluated text (no char/text after the match) $ is usually used at the end of a regex location expression.
?	Check for zero or one occurrence of the previous char ex jpe?g
^~	The match must be at the beginning of the text, note that nginx will not perform any further regular expression match even if an other match is available (check the table above); ^indicate that the match must be at the start of the uri text, while ~indicates a regular expression match mode.example (location ^~/realestate/.*)Nginx evaluation exactly this as don't check regexp locations if this location is longest prefix match.
=	Exact match, no sub folders (location = /
^	Match the beginning of the text (opposite of $). By itself, ^is a shortcut for all paths (since they all have a beginning).
.*	Match zero, one or more occurrence of any char
\	Escape the next char
.	Any char
*	Match zero, one or more occurrence of the previous char
!	Not (negative look ahead)
{}	Match a specific number of occurrence ex. [0-9]{3} match 342 but not 32 {2,4} match length of 2, 3 and 4
+	Match one or more occurrence of the previous char
[]	Match any char inside

13.0.2 Apache regex

https://httpd.apache.org/docs/current/rewrite/intro.html#regex

mod_rewrite uses the Perl Compatible Regular Expression vocabulary. For a more detailed explanation we recommend the PCRE man pages, the,Perl regular expression man page and Mastering Regular Expressions, by Jeffrey Friedl.

13.0.3 IIS regex

Rule pattern syntax

ECMAScript – Perl compatible (ECMAScript standard compliant) regular expression syntax. This is a default option for any rule.

Appendix

SSL Report: www.sunyorange.edu (199.250.206.35)

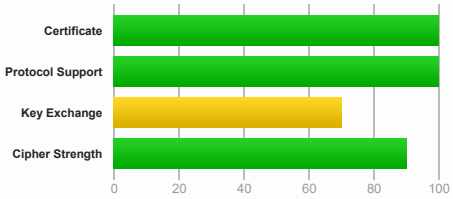
Assessed on: Sun, 27 Nov 2022 18:11:13 UTC | Hide | Clear cache

Scan Another »

Summary

Overall Rating

B



Visit our documentation page for more information, configuration guides, and books. Known issues are documented here.

This server does not support Forward Secrecy with the reference browsers. Grade capped to B. MORE INFO »

This server supports TLS 1.3.

Certificate #1: RSA 2048 bits (SHA256withRSA)

Server Key and Certificate #1

Subject	*.sunyorange.edu Fingerprint SHA256: c64ce47766f237127af6c80278ca336da7af50a4c364de071228aa359ec01b78 Pin SHA256: wD4kPqOwEstuEMS48JEIAbVAsv/CAblGLSgYItAUOV8=
Common names	*.sunyorange.edu
Alternative names	*.sunyorange.edu sunyorange.edu
Serial Number	071daea78484693f5027dfde677a92d8
Valid from	Thu, 05 May 2022 00:00:00 UTC
Valid until	Mon, 05 Jun 2023 23:59:59 UTC (expires in 6 months and 9 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	DigiCert TLS RSA SHA256 2020 CA1 AIA: http://cacerts.digicert.com/DigiCertTLSRSASHA2562020CA1-1.crt
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl3.digicert.com/DigiCertTLSRSASHA2562020CA1-4.crl OCSP: http://ocsp.digicert.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes Mozilla Apple Android Java Windows

Additional Certificates (if supplied)

Additional Certificates (if supplied)

Certificates provided	2 (2970 bytes)
Chain issues	None

#2

Subject	DigiCert TLS RSA SHA256 2020 CA1 Fingerprint SHA256: 52274c57ce4dee3b49db7a7ff708c040f771898b3be88725a86fb4430182fe14 Pin SHA256: RQeZkB42znUfsDIIFWIRiYEcKl7nHwNFwWCrnMMJbVc=
Valid until	Sun, 13 Apr 2031 23:59:59 UTC (expires in 8 years and 4 months)
Key	RSA 2048 bits (e 65537)
Issuer	DigiCert Global Root CA
Signature algorithm	SHA256withRSA

Certification Paths

Mozilla | Apple | Android | Java | Windows

Path #1: Trusted

1	Sent by server	*.sunyorange.edu Fingerprint SHA256: c64ce47766f237127af6c80278ca336da7af50a4c364de071228aa359ec01b78 Pin SHA256: wD4kPqOwEstuEMS48JElAbVAsv/CAblGLSgYltAUOV8= RSA 2048 bits (e 65537) / SHA256withRSA
2	Sent by server	DigiCert TLS RSA SHA256 2020 CA1 Fingerprint SHA256: 52274c57ce4dee3b49db7a7ff708c040f771898b3be88725a86fb4430182fe14 Pin SHA256: RQeZkB42znUfsDIIFWIRiYEcKl7nHwNFwWCrnMMJbVc= RSA 2048 bits (e 65537) / SHA256withRSA
3	In trust store	DigiCert Global Root CA Self-signed Fingerprint SHA256: 4348a0e9444c78cb265e058d5e8944b4d84f9662bd26db257f8934a443c70161 Pin SHA256: r/mIkG3eEpVdm+u/ko/cwxzOMo1bk4TyHIlByibiA5E= RSA 2048 bits (e 65537) / SHA1withRSA Weak or insecure signature, but no impact on root certificate

Configuration

Protocols

TLS 1.3	Yes
TLS 1.2	Yes
TLS 1.1	No
TLS 1.0	No
SSL 3	No
SSL 2	No

Cipher Suites

TLS 1.3 (server has no preference)

TLS_AES_128_GCM_SHA256 (0x1301) ECDH x25519 (eq. 3072 bits RSA) FS	128
TLS_AES_256_GCM_SHA384 (0x1302) ECDH x25519 (eq. 3072 bits RSA) FS	256
TLS_CHACHA20_POLY1305_SHA256 (0x1303) ECDH x25519 (eq. 3072 bits RSA) FS	256

TLS 1.2 (server has no preference)

TLS_RSA_WITH_AES_128_CBC_SHA (0x2f) **WEAK**	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33) DH 2048 bits FS **WEAK**	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) ECDH secp521r1 (eq. 15360 bits RSA) FS **WEAK**	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) **WEAK**	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (0x67) DH 2048 bits FS **WEAK**	128
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) **WEAK**	128

Cipher Suites

TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e) DH 2048 bits FS	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp521r1 (eq. 15360 bits RSA) FS **WEAK**	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp521r1 (eq. 15360 bits RSA) FS	128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35) **WEAK**	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39) DH 2048 bits FS **WEAK**	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) ECDH secp521r1 (eq. 15360 bits RSA) FS **WEAK**	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) **WEAK**	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (0x6b) DH 2048 bits FS **WEAK**	256
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) **WEAK**	256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x9f) DH 2048 bits FS	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp521r1 (eq. 15360 bits RSA) FS **WEAK**	256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp521r1 (eq. 15360 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8) ECDH secp521r1 (eq. 15360 bits RSA) FS	256

Handshake Simulation

Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Android 5.0.0	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp521r1 FS
Android 6.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Android 8.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Android 8.1	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Android 9.0	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
BingPreview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
Chrome 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Chrome 70 / Win 10	-	TLS 1.3	TLS_AES_128_GCM_SHA256 ECDH x25519 FS
Chrome 80 / Win 10 R	-	TLS 1.3	TLS_AES_128_GCM_SHA256 ECDH x25519 FS
Firefox 31.3.0 ESR / Win 7	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
Firefox 73 / Win 10 R	-	TLS 1.3	TLS_AES_128_GCM_SHA256 ECDH x25519 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH x25519 FS
IE 11 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 11 / Win 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 11 / Win Phone 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_RSA_WITH_AES_128_CBC_SHA256 No FS
IE 11 / Win Phone 8.1 Update R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH x25519 FS
Edge 16 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH x25519 FS
Edge 18 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH x25519 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Java 8u161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Java 11.0.3	-	TLS 1.3	TLS_AES_128_GCM_SHA256 ECDH secp256r1 FS
Java 12.0.1	-	TLS 1.3	TLS_AES_128_GCM_SHA256 ECDH secp256r1 FS
OpenSSL 1.0.1l R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS
OpenSSL 1.0.2s R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
OpenSSL 1.1.0k R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH x25519 FS
OpenSSL 1.1.1c R	-	TLS 1.3	TLS_AES_256_GCM_SHA384 ECDH x25519 FS
Safari 6 / iOS 6.0.1	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS

Handshake Simulation

Safari 7 / iOS 7.1 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 12.1.2 / MacOS 10.14.6 Beta R	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Safari 12.1.1 / iOS 12.3.1 R	-	TLS 1.3	TLS_CHACHA20_POLY1305_SHA256 ECDH x25519 FS
Apple ATS 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Yahoo Slurp Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp384r1 FS
YandexBot Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp521r1 FS

Not simulated clients (Protocol mismatch)

Android 2.3.7 No SNI [2]	Protocol mismatch (not simulated)
Android 4.0.4	Protocol mismatch (not simulated)
Android 4.1.1	Protocol mismatch (not simulated)
Android 4.2.2	Protocol mismatch (not simulated)
Android 4.3	Protocol mismatch (not simulated)
Baidu Jan 2015	Protocol mismatch (not simulated)
IE 6 / XP No FS [1] No SNI [2]	Protocol mismatch (not simulated)
IE 7 / Vista	Protocol mismatch (not simulated)
IE 8 / XP No FS [1] No SNI [2]	Protocol mismatch (not simulated)
IE 8-10 / Win 7 R	Protocol mismatch (not simulated)
IE 10 / Win Phone 8.0	Protocol mismatch (not simulated)
Java 6u45 No SNI [2]	Protocol mismatch (not simulated)
Java 7u25	Protocol mismatch (not simulated)
OpenSSL 0.9.8y	Protocol mismatch (not simulated)
Safari 5.1.9 / OS X 10.6.8	Protocol mismatch (not simulated)
Safari 6.0.4 / OS X 10.8.4 R	Protocol mismatch (not simulated)

(1) Clients that do not support Forward Secrecy (FS) are excluded when determining support for it.
(2) No support for virtual SSL hosting (SNI). Connects to the default site if the server uses SNI.
(3) Only first connection attempt simulated. Browsers sometimes retry with a lower protocol version.
(R) Denotes a reference browser or client, with which we expect better effective security.
(All) We use defaults, but some platforms do not use their best protocols and features (e.g., Java 6 & 7, older IE).
(All) Certificate trust is not checked in handshake simulation, we only perform TLS handshake.

Protocol Details

DROWN	No, server keys and hostname not seen elsewhere with SSLv2 **(1) For a better understanding of this test, please read this longer explanation** (2) Key usage data kindly provided by the Censys network search engine; original DROWN website here (3) Censys data is only indicative of possible key and certificate reuse; possibly out-of-date and not complete
Secure Renegotiation	**Supported**
Secure Client-Initiated Renegotiation	No
Insecure Client-Initiated Renegotiation	No
BEAST attack	Mitigated server-side (more info)
POODLE (SSLv3)	No, SSL 3 not supported (more info)
POODLE (TLS)	No (more info)
Zombie POODLE	No (more info) TLS 1.2 : 0x002f
GOLDENDOODLE	No (more info) TLS 1.2 : 0x002f
OpenSSL 0-Length	No (more info) TLS 1.2 : 0x002f
Sleeping POODLE	No (more info) TLS 1.2 : 0x002f

Protocol Details

Downgrade attack prevention	**Yes, TLS_FALLBACK_SCSV supported** (more info)
SSL/TLS compression	No
RC4	No
Heartbeat (extension)	No
Heartbleed (vulnerability)	No (more info)
Ticketbleed (vulnerability)	No (more info)
OpenSSL CCS vuln. (CVE-2014-0224)	No (more info)
OpenSSL Padding Oracle vuln. (CVE-2016-2107)	No (more info)
ROBOT (vulnerability)	No (more info)
Forward Secrecy	**With some browsers** (more info)
ALPN	Yes http/1.1
NPN	No
Session resumption (caching)	Yes
Session resumption (tickets)	Yes
OCSP stapling	**Yes**
Strict Transport Security (HSTS)	No
HSTS Preloading	Not in: Chrome Edge Firefox IE
Public Key Pinning (HPKP)	No (more info)
Public Key Pinning Report-Only	No
Public Key Pinning (Static)	No (more info)
Long handshake intolerance	No
TLS extension intolerance	No
TLS version intolerance	No
Incorrect SNI alerts	No
Uses common DH primes	No
DH public server param (Ys) reuse	No
ECDH public server param reuse	No
Supported Named Groups	secp256r1, secp384r1, secp521r1, x25519, x448 (Server has no preference)
SSL 2 handshake compatibility	Yes
0-RTT enabled	No

HTTP Requests

1 **https://www.sunyorange.edu/** (HTTP/1.1 301 Moved Permanently)

Miscellaneous

Test date	Sun, 27 Nov 2022 18:09:33 UTC
Test duration	100.150 seconds
HTTP status code	301
HTTP forwarding	https://sunyorange.edu
HTTP server signature	Apache
Server hostname	vps42027.inmotionhosting.com

SSL Report v2.1.10