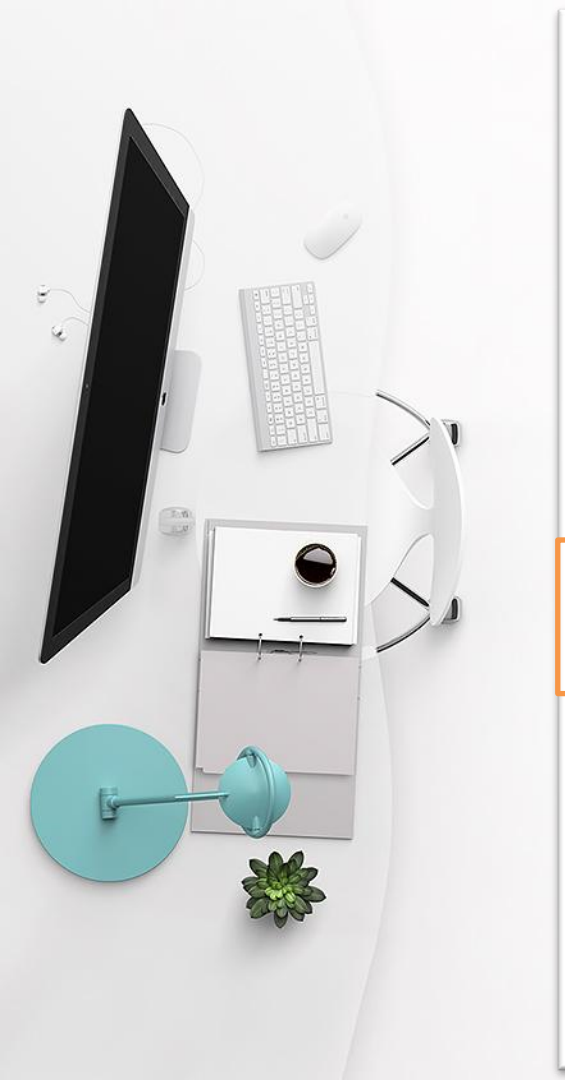
Open Source Software Development

Ung Văn Giàu
Email: giau.ung@eiu.edu.vn

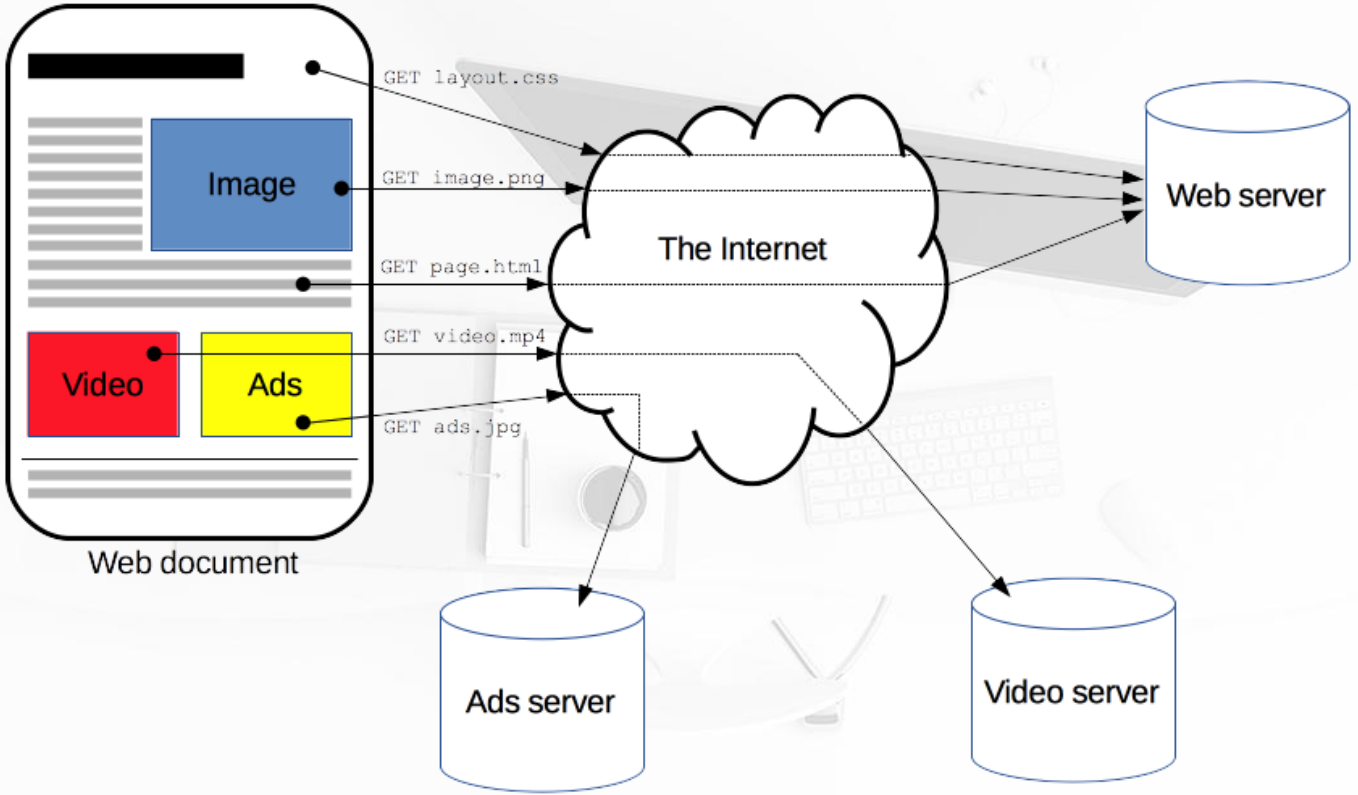
Web server

Contents



Web Server

Web server


GET layout.css
GET image.png
GET page.html
GET video.mp4
GET ads.jpg
Image
Video
Ads
Web document
The Internet
Web server
Ads server
Video server

Web server


Files
HTTP server
HTTP Request
HTTP Response
Web server
Browser

Web server

Web server can refer to hardware or software, or both working together

- On the **hardware side**
  - A web server is a **computer** that **stores** web server software and a website's component files
  - A web server **connects to the Internet** and **supports physical data interchange** with other devices connected to the web

Web server

Web server can refer to hardware or software, or both working together

- On the **software side**
  - A web server includes several parts that control how web users access hosted file
  - At a minimum, this is an **HTTP server**
    - ✓ An HTTP server is software that **understands URLs** (web addresses) and **HTTP** protocol
    - ✓ An HTTP server can **be accessed through the domain names** of the websites it stores, and it **delivers the content** of these hosted websites to the end user's device

Web server

To publish a website, you need either a static or a dynamic web server

- A **static web server** consists of a computer (hardware) with an HTTP server (software)

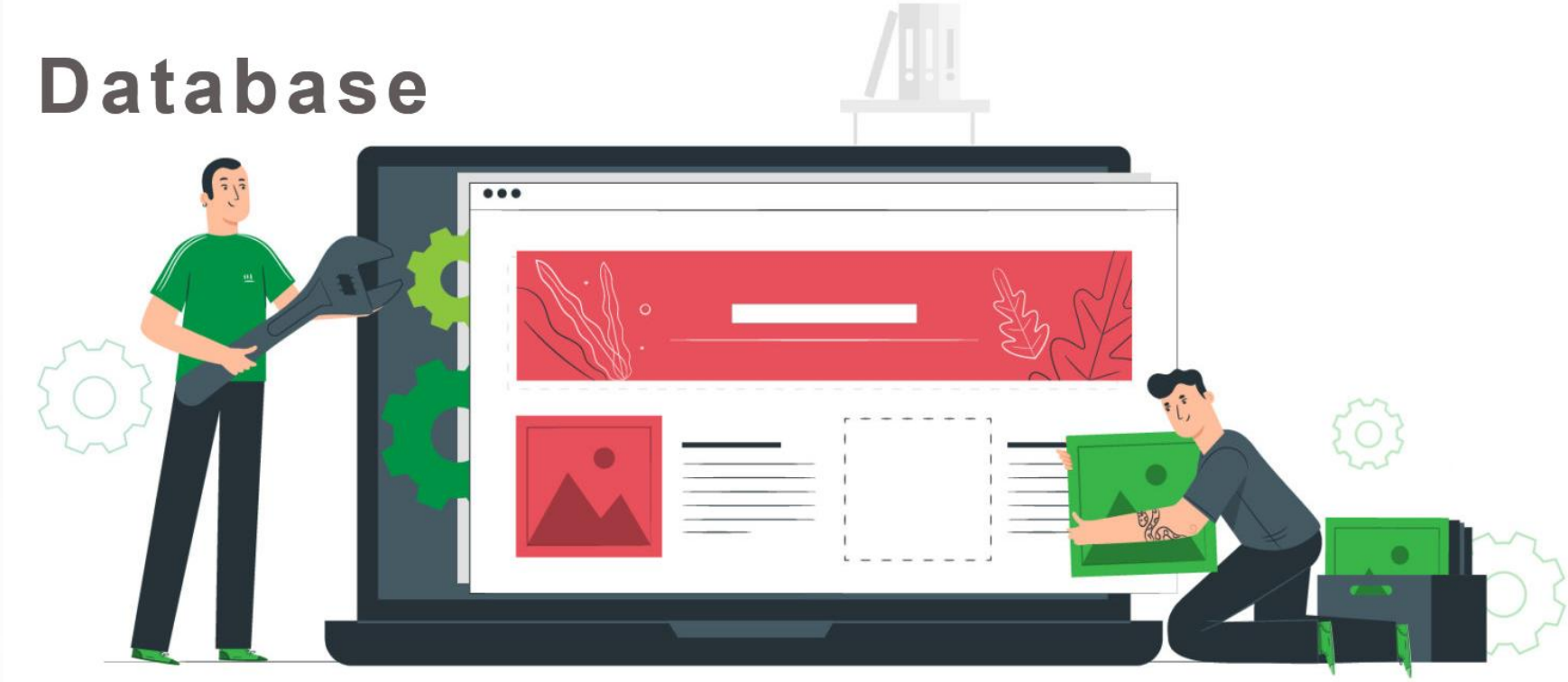
  The **server sends hosted files** as-is to your browser

- A **dynamic web server** consists of a static web server plus extra software, most commonly an application server and a database

  The **application server updates the hosted files before sending content** to your browser via the HTTP server

Web server

Dynamic web server

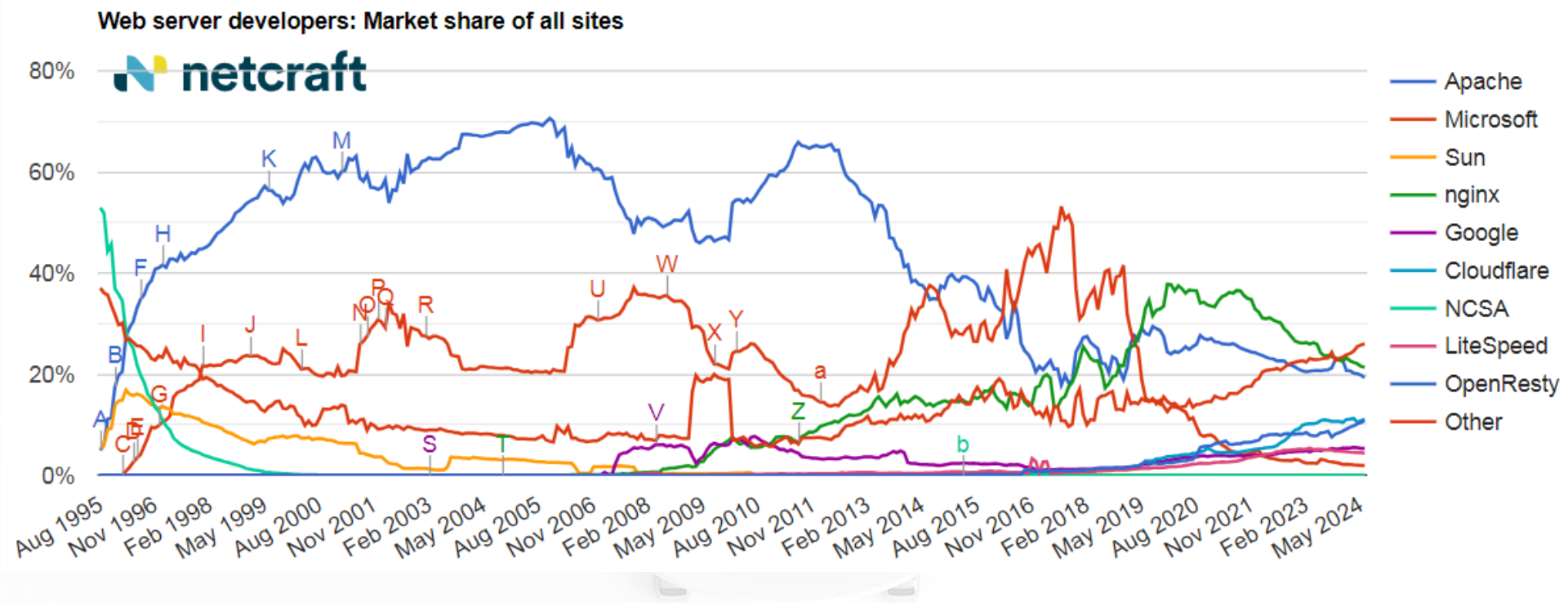


Types of Web Servers

- **Apache** (provided by Apache)
- IIS (provided by Microsoft)
- **nginx** (provided by NGINX, Inc. and pronounced like “Engine X”)
- GWS (provided by Google and short for Google Web Server)
- Lighttpd
- LiteSpeed (provided by LiteSpeed Technologies)
- Sun Java System Web Server — SJSAS

Web Servers Market Share

June 2024



Web Servers Market Share

June 2024

Developer	May 2024	Percent	June 2024	Percent
nginx	236,239,936	21.53%	235,170,823	21.35%
Apache	217,239,604	19.80%	212,402,611	19.28%
Cloudflare	118,561,124	10.80%	121,715,882	11.05%
OpenResty	114,268,616	10.41%	118,852,803	10.79%

Web Hosting

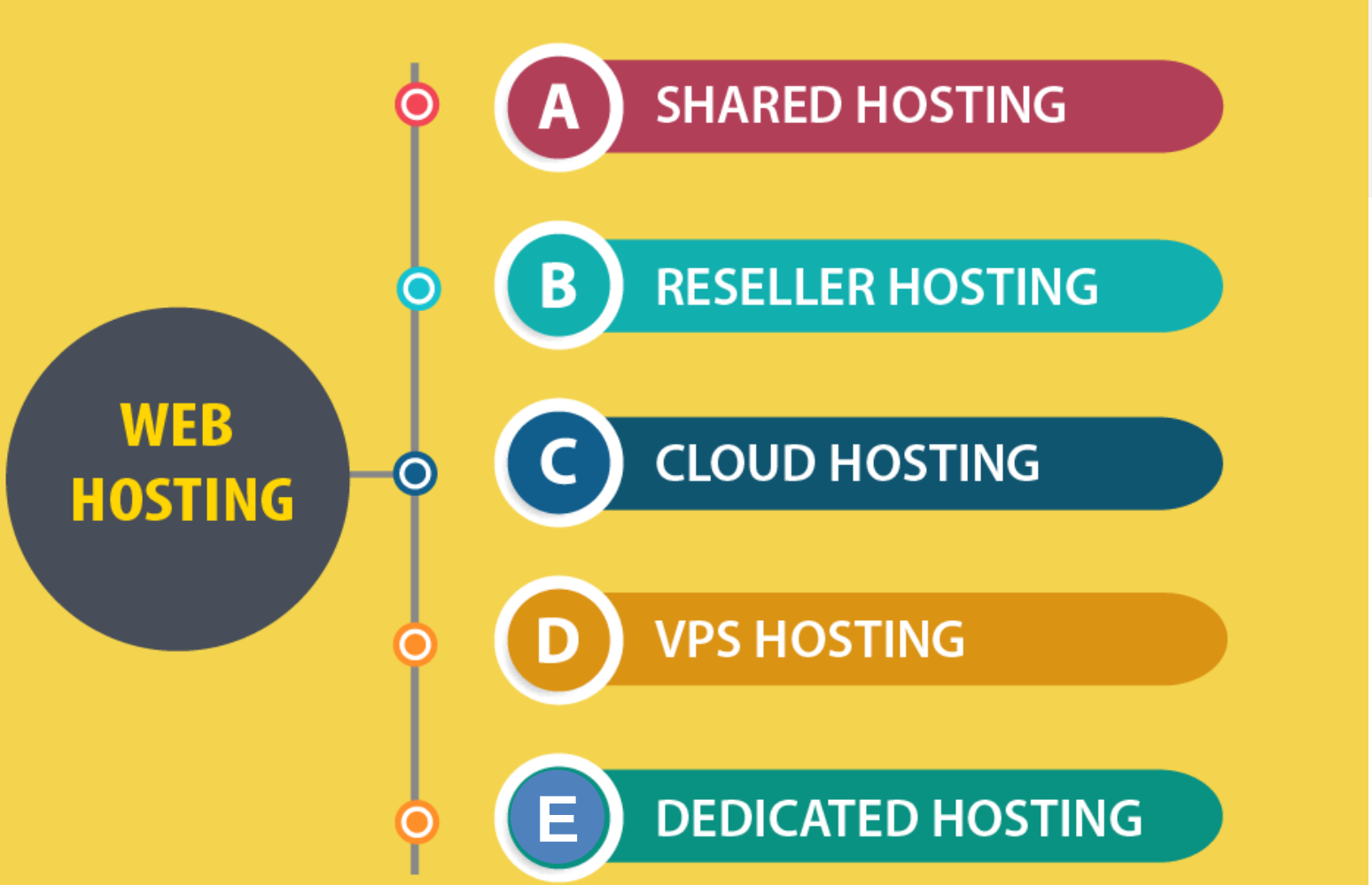
Web hosting

- To **make a website available online**, its files need to be uploaded to a web server, which is typically purchased from a **hosting provider**

→ This service is known as **web hosting**

- Web hosting services differ by:
  - how the servers are set up (i.e., the space allocated)
  - the type of access that people have to them

- Every website you've ever visited is hosted on a server


WEB HOSTING
A
SHARED HOSTING
B
RESELLER HOSTING
C
CLOUD HOSTING
D
VPS HOSTING
E
DEDICATED HOSTING

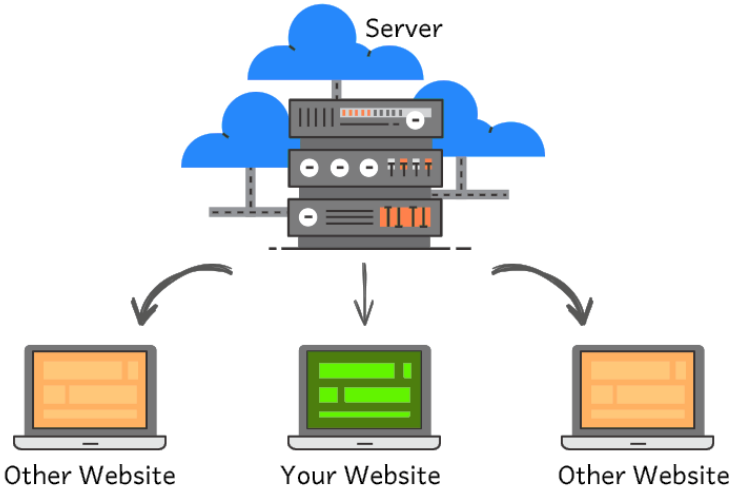
Types of Web hosting

- **Shared Hosting**

  A single physical server hosts **multiple websites**

→ costs low

  - Each customer has access to such features as databases, monthly traffic reports, disk space, email accounts, FTP accounts,…
  - System resources are shared on-demand by users on the server, where each one gets a percentage of everything from RAM and CPU as well as the single MySQL, Apache, and mail servers

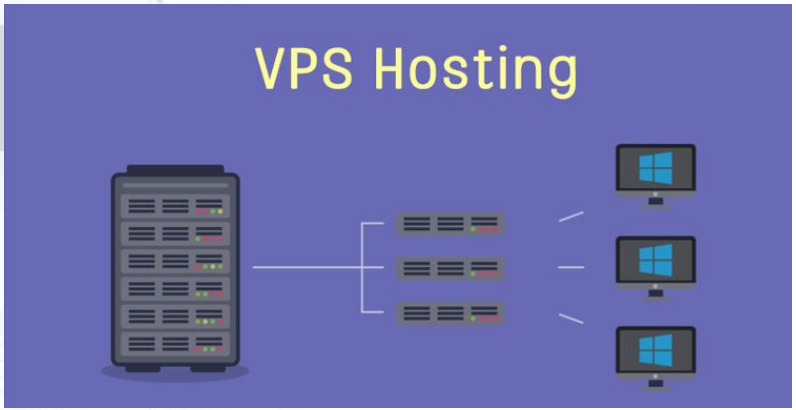


Types of Web hosting

- **Dedicated Server**
  - A **physical server** where all the resources of one machine are dedicated to a single user
  - Customer benefits from every resource available such as RAM, storage, computing power,…

Types of Web hosting

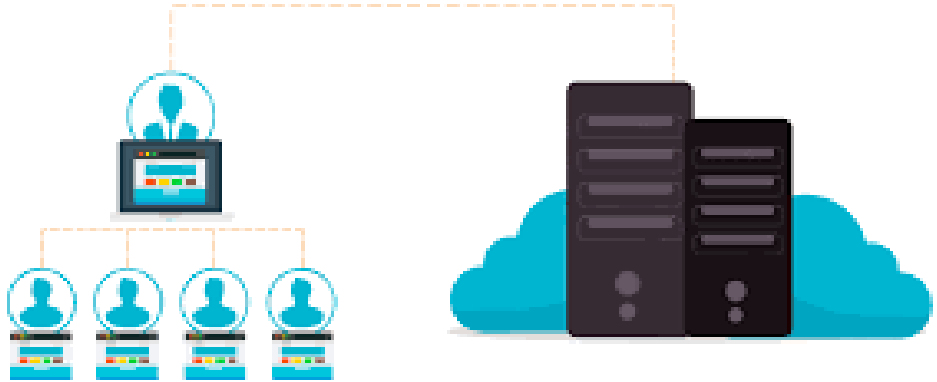
- **VPS Hosting**
  - VPS = Virtual Private Server
  - A VPS refers to the **partitioning of a physical server into multiple ones**
  - Each VPS has own Operating System (OS), it receives a specific share of resources from one physical server so they are isolated, unable to interfere, and can be separately rebooted



Types of Web hosting

- **Reseller Hosting** is a form of web hosting where the account owner has the ability to use their allotted hard drive space and bandwidth to host websites on behalf of third parties

  In other words, the reseller purchases the hosting provider’s wholesale services and then sells them to their clients/customers for a profit



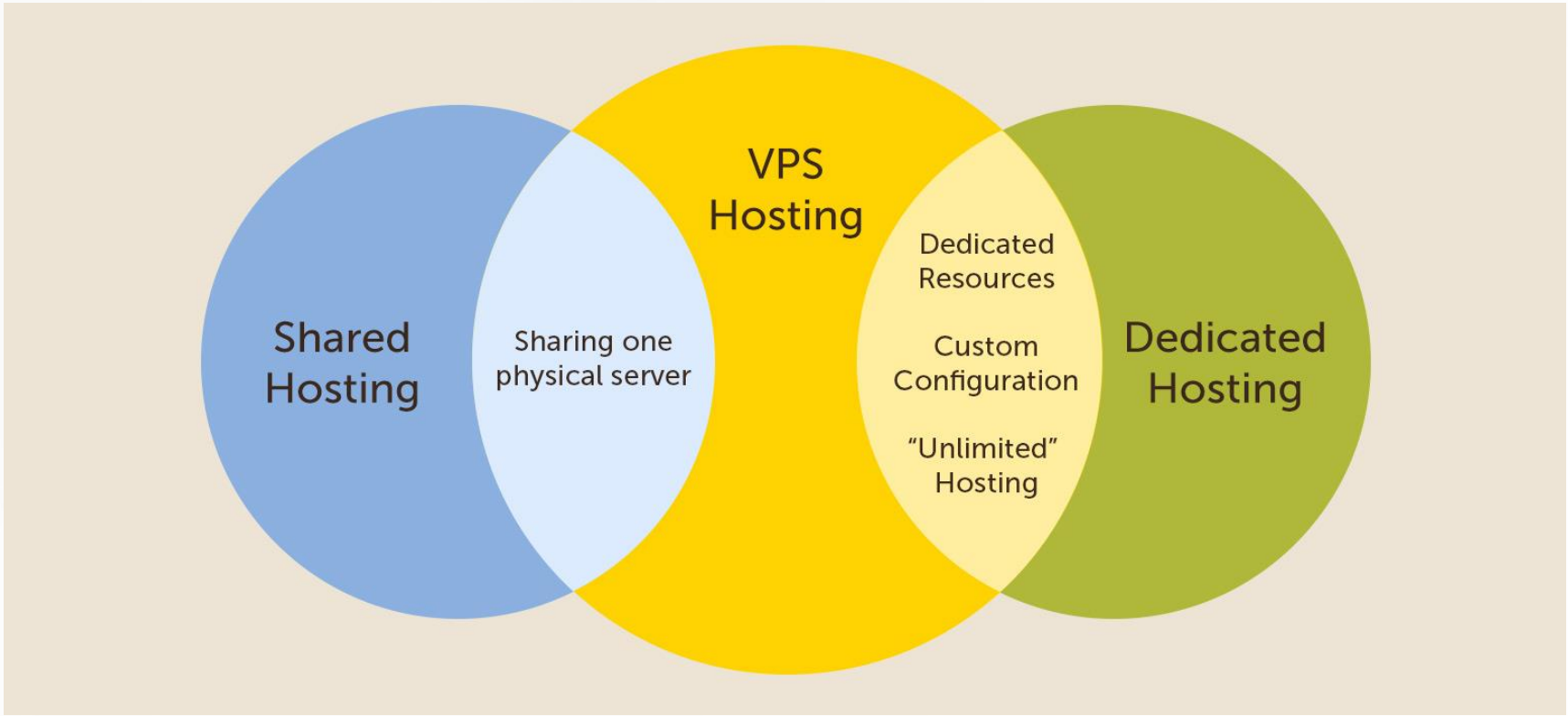
Types of Web hosting

- **Cloud hosting** is the latest hosting type to hit the market, and it's become extremely popular in recent years

  This type of hosting operates across many interconnected web servers that supply an affordable, scalable and reliable web infrastructure

Types of Web hosting

Comparation



Types of Web Hosting

Shared Hosting

A Room in a Hostel

Cloud Hosting

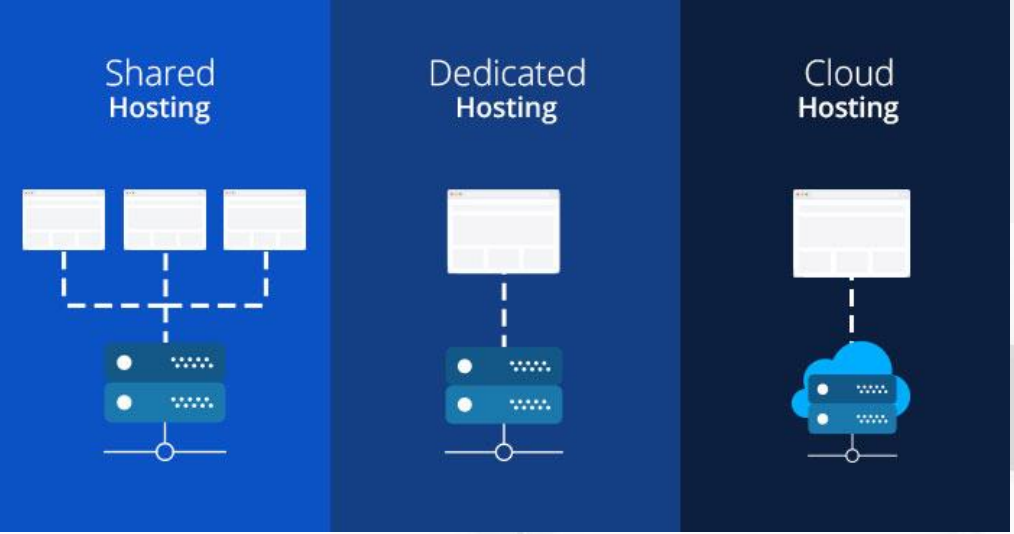
A Membership to a Chain of Hotels

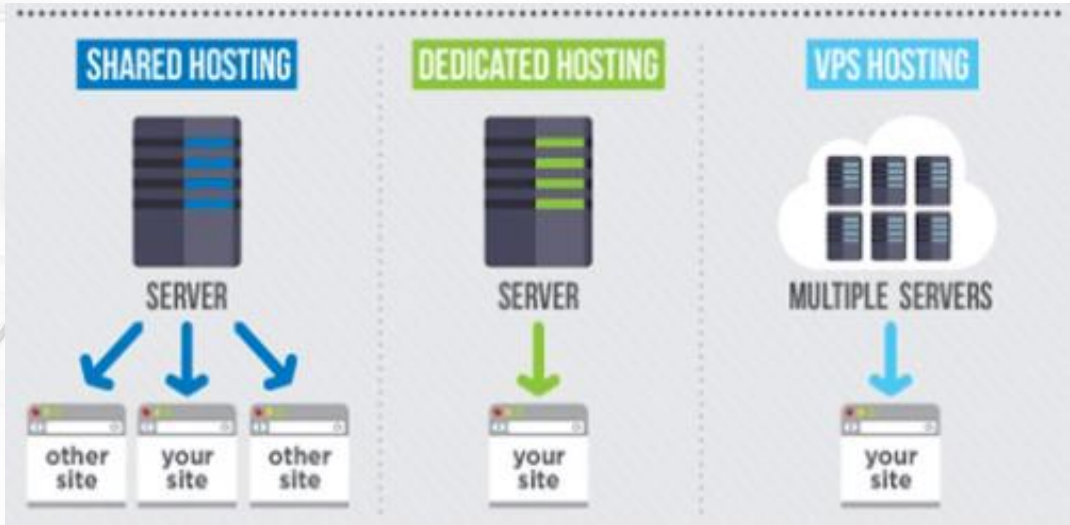
VPS Hosting

A Town House

Dedicated Hosting

Your Own Private Estate


Shared
Hosting
Dedicated
Hosting
Cloud
Hosting


SHARED HOSTING
SERVER
other site
your site
other site
DEDICATED HOSTING
SERVER
your site
VPS HOSTING
MULTIPLE SERVERS
your site

Domain name and DNS

Domain Name

- A domain name is **a string of text** (the address) that maps to a numeric IP address, used to access a website

- A domain name is used for **finding and identifying computers** on the Internet

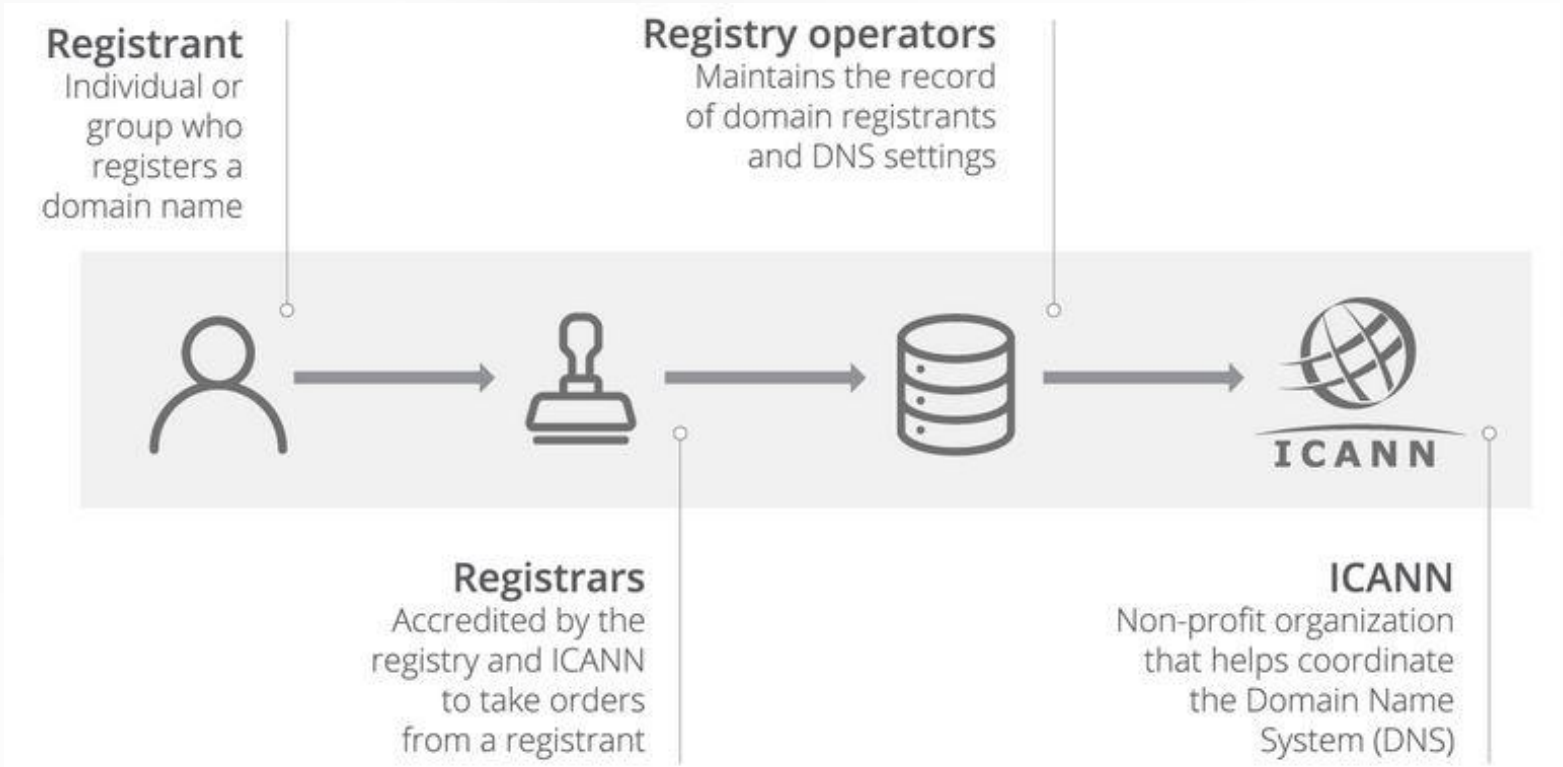
- Computers use IP addresses, which are a series of number. It is difficult for humans to remember strings of numbers → domain names were developed and used to identify entities on the Internet rather than using IP addresses

Domain Name

- A domain name can be **any combination of letters and numbers**, and it can be used in combination of the various domain name **extensions** (.com, .net,…)

- The domain name **must be registered** before using it and every domain name is **unique**

Domain Name


Registrant
Individual or group who registers a domain name
Registry operators
Maintains the record of domain registrants and DNS settings
ICANN
Registrars
Accredited by the registry and ICANN to take orders from a registrant
ICANN
Non-profit organization that helps coordinate the Domain Name System (DNS)

Domain Name System

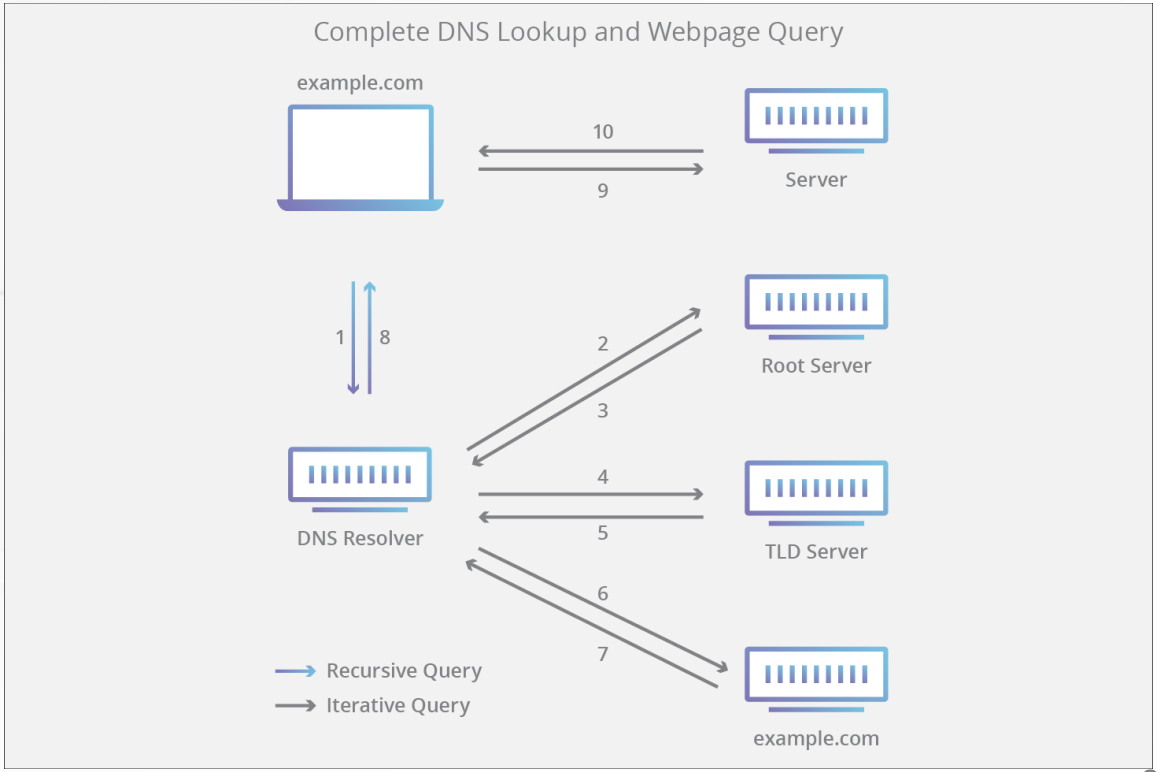
DNS

- DNS is the **phonebook of the Internet**

- DNS **translates domain names to IP addresses** so browsers can load Internet resources.

What are the steps in a DNS lookup?

- The 8 steps in a DNS lookup:



What are the steps in a DNS lookup?

1. A user types ‘example.com’ into a web browser and the query travels into the Internet and is received by a **DNS recursive resolver**
2. The resolver then queries a **DNS root nameserver** (.)
3. The root server then responds to the resolver with the address of **a Top Level Domain (TLD) DNS server** (such as .com or .net), which stores the information for its domains. When searching for example.com, our request is pointed toward the .com TLD.
4. The resolver then makes a request to the .com TLD.
5. The TLD server then responds with the **IP address** of the domain’s nameserver, example.com.

What are the steps in a DNS lookup?

6. Lastly, the recursive resolver sends a query to the domain’s nameserver
7. The IP address for example.com is then returned to the resolver from the nameserver
8. The DNS resolver then responds to the web browser with the IP address of the domain requested initially

Once the 8 steps of the DNS lookup have returned the IP address for example.com, the browser is able to make the request for the web page:

1. The browser makes a HTTP request to the IP address.
2. The server at that IP returns the webpage to be rendered in the browser (step 10)

Domain Information

Name: VITINHDUCTHANH.COM

Registry Domain ID: 2563563634_DOMAIN_COM-VRSN

Domain Status:
clientTransferProhibited

Nameservers:
ERIN.NS.CLOUDFLARE.COM
MACK.NS.CLOUDFLARE.COM

Dates

Registry Expiration: 2022-10-03 02:05:06 UTC

Created: 2020-10-03 02:05:06 UTC

Registrar Information

Name: P.A. Viet Nam Company Limited

IANA ID: 1649

Abuse contact email: abuse@pavietnam.vn

Abuse contact phone: tel:+84.2873019954

DNSSEC Information

Delegation Signed: Unsigned

Authoritative Servers

Registry Server URL: https://rdap.verisign.com/com/v1/domain/vitinhducthanh.com

Last updated from Registry RDAP DB: 2021-10-18 16:31:38 UTC

Registrar Server URL: https://rdap.pavietnam.vn/domain/VITINHDUCTHANH.COM

Hosts File

- **The typical locations**
  - Windows 10

    “C:\Windows\System32\drivers\etc\hosts”
  - Linux

    “/etc/hosts”

Hosts File

- **What is a Hosts File?**
  - The hosts file is a local plain text file that maps hostnames to IP addresses
  - It was **the original method to resolve hostnames to a specific IP address**. The hosts file is usually the **first process** in the domain name resolution procedure
  - E.g.: 127.0.0.1 localhosts #loopback
    - ✓The **first** section denotes the **IP address**
    - ✓The **second** section designates **the location**
    - ✓The **third** section specifies a **comment** for the entry
    - ✓Each entry is usually **separated by a space or a tab**

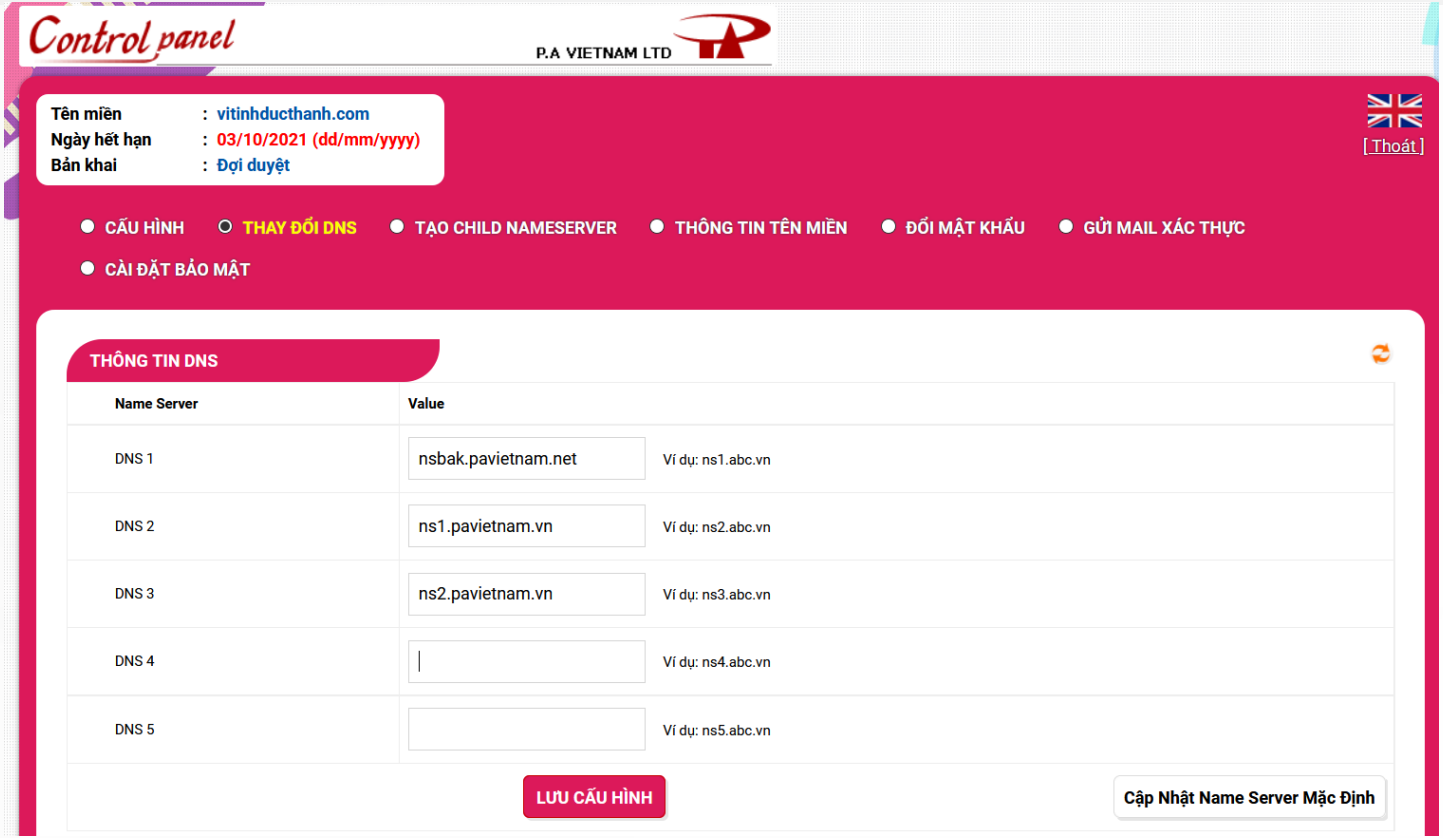
Hosts File

- **Why is a Hosts File Useful?**
  - Changing hosts file is a temporary measure to preview site
  - It is very useful for development purposes, as you can work on site on an alternate server using a hosts file

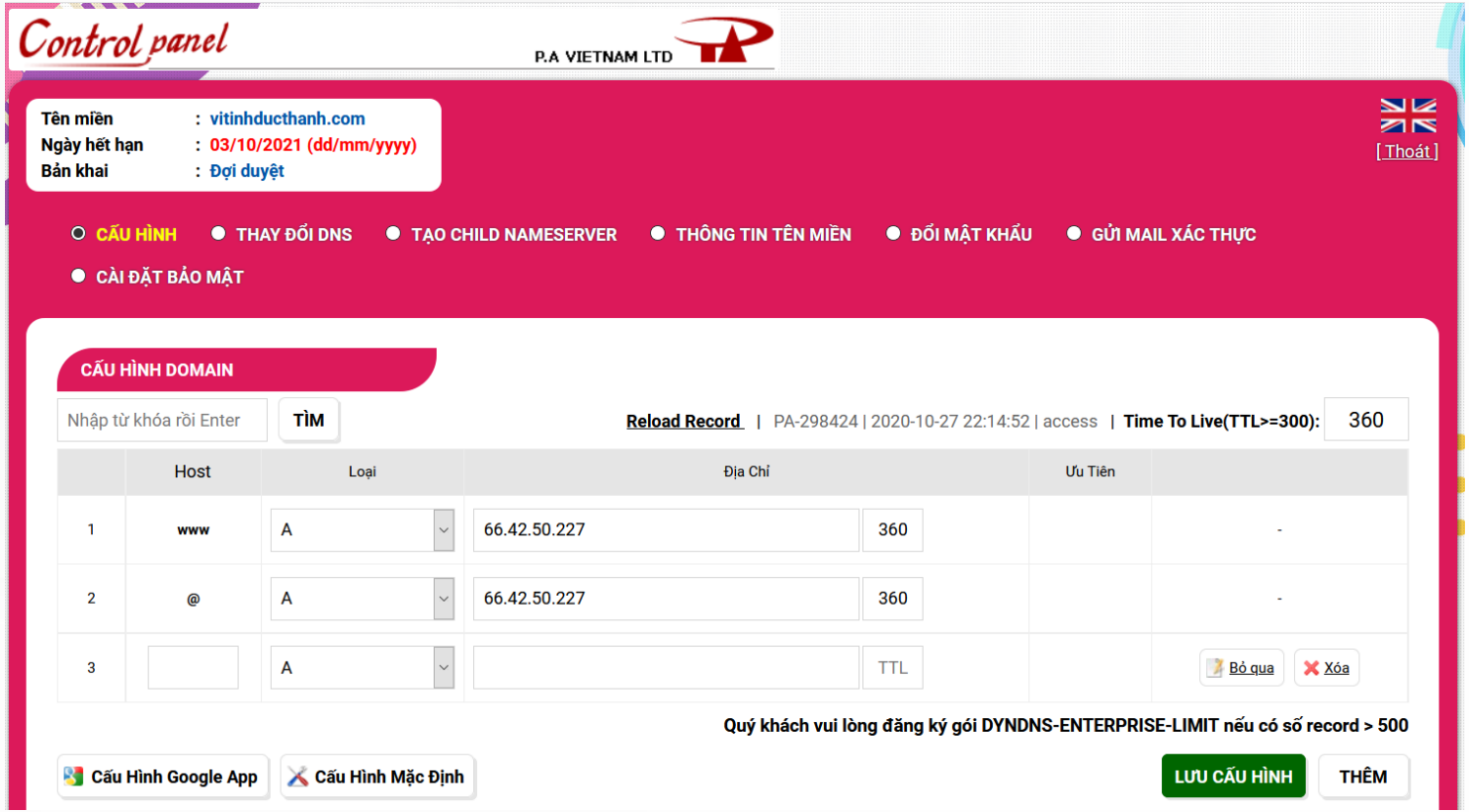
  - **Note:**

    Once you have modified hosts file, flush DNS so that the new changes can be implemented

DNS configuration



Domain name Configuration



HTTP Server

APACHE

HTTP SERVER PROJECT

Apache HTTP

- The Apache HTTP Server is **an open-source HTTP server** for modern operating systems including UNIX and Windows

- The goal is to provide **a secure, efficient and extensible server** that provides HTTP services in sync with the current HTTP standards

- The Apache HTTP Server (“httpd”) was launched in 1995 and it has been the most popular web server on the Internet since April 1996

- The Apache HTTP Server is a project of The Apache Software Foundation

Apache HTTP

- The word, Apache, has been taken from the name of the Native American tribe ‘Apache’, famous for its skills in warfare and strategy making

Apache HTTP

- It is a modular, process-based web server application that creates a new thread with each simultaneous connection

- It **supports a number of features**

  many of them are compiled as **separate modules and extend its core functionality**, and can provide everything from server-side programming language support to authentication mechanism

Important Apache Files and Directories

Linux

- **Content**

  /var/www/html

- **Server Configuration**
  - **/etc/apache2**

    The Apache configuration directory. All Apache configuration files reside here
  - **/etc/apache2/apache2.conf**

    The main Apache configuration file (global configuration). This file is responsible for loading many of the other files in the configuration directory
  - **/etc/apache2/ports.conf**

    This file specifies the ports that Apache will listen on
  - **/etc/apache2/sites-available/**

    The directory where per-site virtual hosts can be stored

Important Apache Files and Directories

Linux

- **Server Configuration**
  - **/etc/apache2/sites-enabled/**

    The directory where enabled per-site virtual hosts are stored. Typically, these are created by linking to configuration files found in the sites-available directory with the **a2ensite**

- **Server Logs**
  - **/var/log/apache2/access.log**

    By default, every request to your web server is recorded in this log file unless Apache is configured to do otherwise
  - **/var/log/apache2/error.log**

    By default, all errors are recorded in this file

NGINX

Nginx

- Nginx [engine x] is an HTTP and reverse proxy server, a mail proxy server, and a generic TCP/UDP proxy server, load balance
- The software was created by Igor Sysoev and publicly released in 2004
- Nginx is **free and open-source software**
- For a long time, it has been running on many heavily loaded Russian sites including Yandex, Mail.Ru,…
- Here are some of the success stories: Dropbox, Netflix, Wordpress.com, Adobe, Cloudflare, African Bank…

Important Nginx Files and Directories

Linux

- **Content**

  /var/www/html

- **Server Configuration**
  - **/etc/nginx**

    The Nginx configuration directory
  - **/etc/nginx/nginx.conf**

    The main Nginx configuration file (global configuration)
  - **/etc/nginx/sites-available/**

    The directory where per-site server blocks can be stored
  - **/etc/nginx/sites-enabled/**

    The directory where enabled per-site server blocks are stored. Typically, these are created by linking to configuration files found in the sites-available directory

Important Nginx Files and Directories

Linux

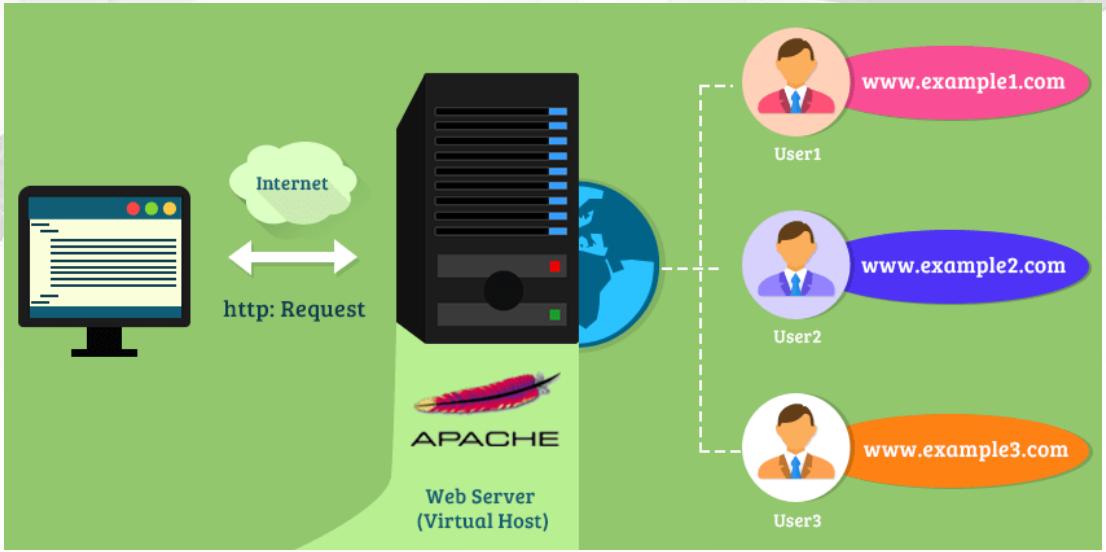
- **Server Logs**
  - **/var/log/nginx/access.log**

    Every request to your web server is recorded in this log file unless Nginx is configured to do otherwise
  - **/var/log/nginx/error.log**

    Any Nginx errors will be recorded in this log

What is Virtual Host?

- **Virtual host** is one such feature that allows a single Web Server to **serve a number of different websites**
- Any domain hosted on web server will have a separate entry in configuration file



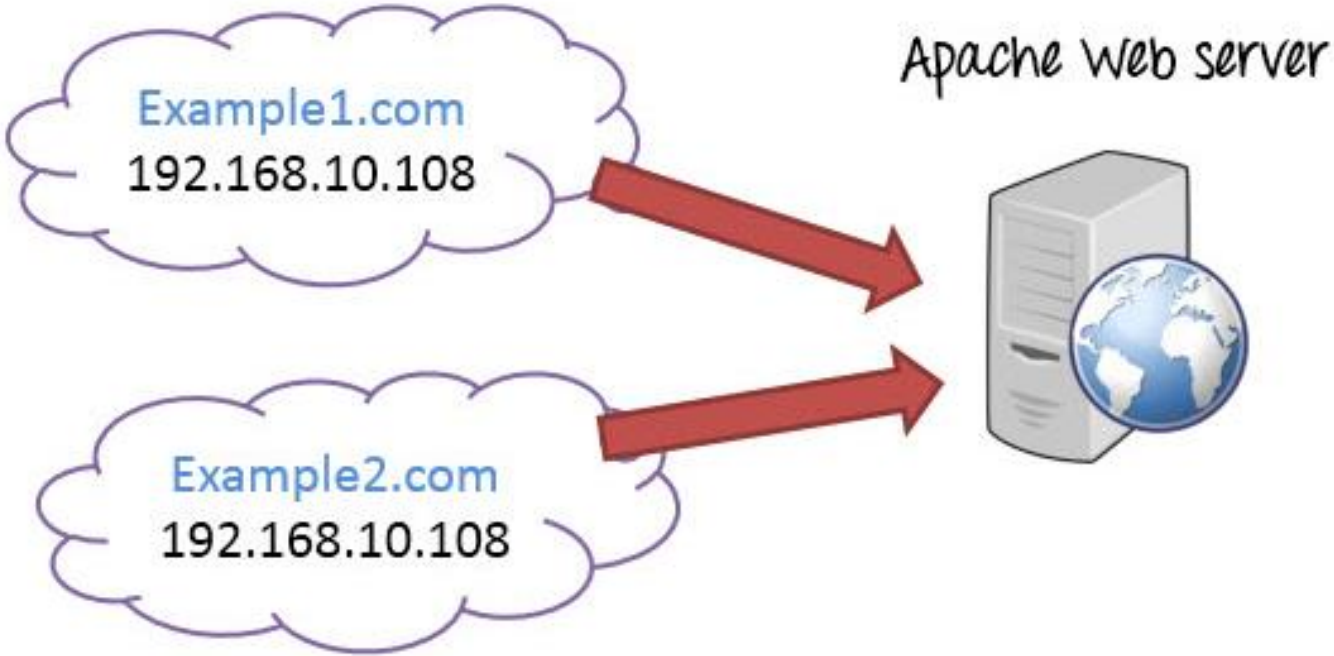
Types of Apache Virtual host

1. Name-based Virtual host
2. Address-based or IP-based virtual host

Types of Apache Virtual host

Name-based Virtual Host

- Name-based virtual host is used to host multiple virtual sites on a single IP address



Types of Apache Virtual host

Name-based Virtual Host

- Do this by **NameVirutalHost** directive within the Apache configuration (**apache2.conf** file)

Apache virtual host Example:

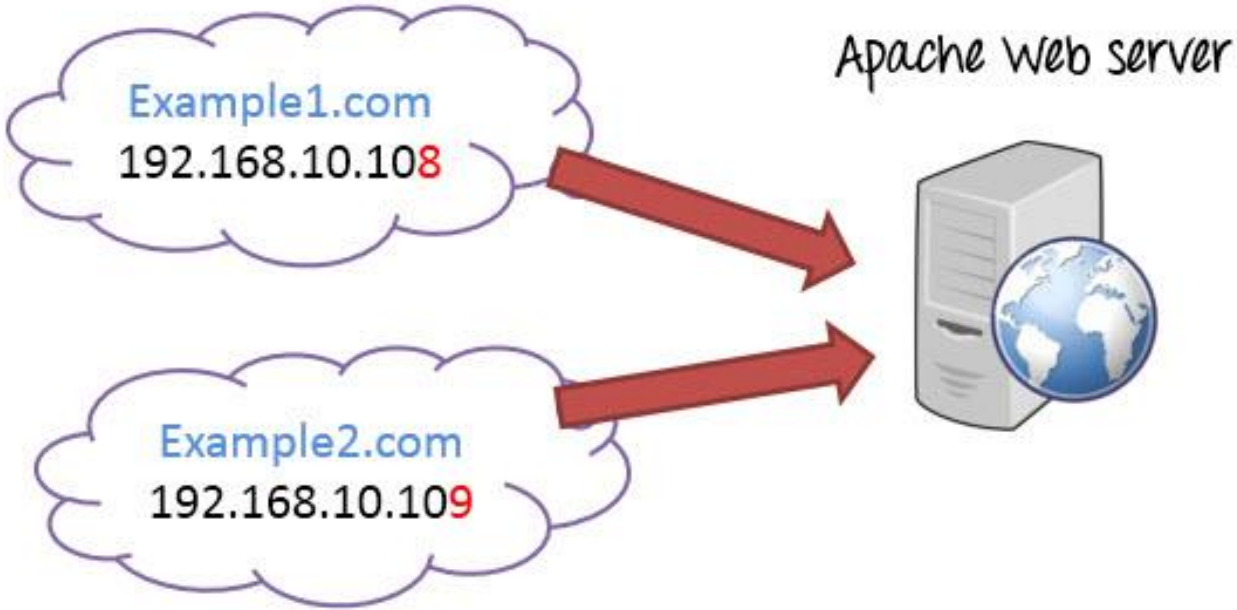
```
NameVirtualHost *:80

<VirtualHost 192.168.0.108:80>

ServerAdmin webmaster@example1.com

DocumentRoot /var/www/html/example1.com

ServerName www.example1.com

</VirtualHost>

<VirtualHost 192.168.0.108:80>

ServerAdmin admin@example2.com

DocumentRoot /var/www/html/example2.com

ServerName www.example2.com

</VirtualHost>
```

Types of Apache Virtual host

IP-based Virtual Host

- To setup IP-based virtual host, need more than one IP address configured on server

→ The number of Apache vhost will depend on number of IP address configured on server



Types of Apache Virtual host

IP-based Virtual Host

```
<VirtualHost 192.168.10.108:80>

ServerAdmin webmaster@example1.com

DocumentRoot /var/www/html/example1.com

ServerName www.example1.com

</VirtualHost>

<VirtualHost 192.168.10.109:80>

ServerAdmin admin@example2.com

DocumentRoot /var/www/html/example2.com

ServerName www.example2.com

</VirtualHost>
```

```
RewriteCond %{DOCUMENT_ROOT}/$1 !-f
RewriteCond %{HTTP_HOST} ^(admin.example.com)$
RewriteRule ^/?([a-z]+)/(.*)$ /admin.foo?page=$1&id=$2&host=%1 [PT]
```

Rewrite Engine

Rewrite engine

- A rewrite engine is a software component that **performs rewriting on Uniform Resource Locators**, modifying their appearance. This modification is called **URL rewriting**

- It is a way of implementing URL mapping or routing within a web application. The engine is typically a component of a web server or web application framework

- Rewritten URLs are used to provide **shorter and more relevant-looking links** to web pages

Rewrite engine

- The URL to a wiki page might be:

  http://example.com/w/index.php?title=Page_title

  but can be rewritten as:

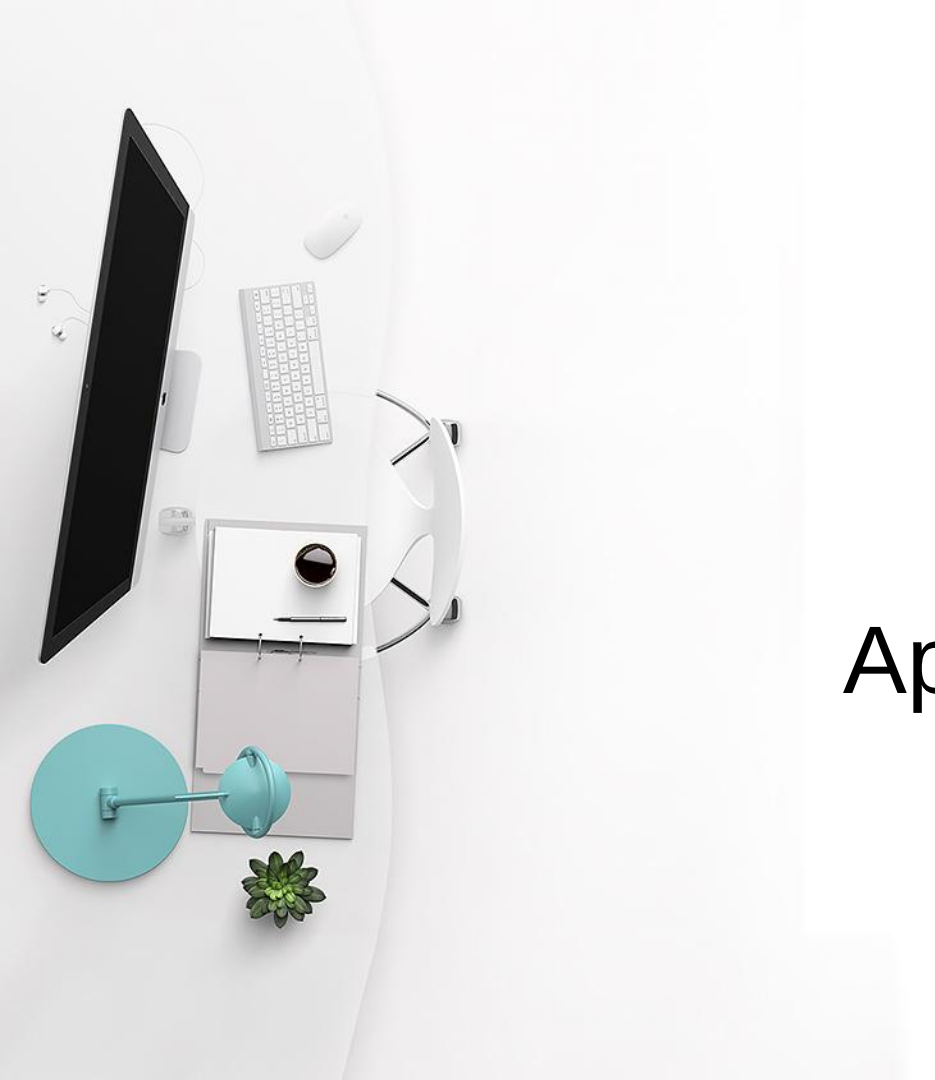
  http://example.com/wiki/Page_title

- A blog might have a URL that encodes the dates of each entry:

  http://www.example.com/Blog/Posts.php?Year=2006&Month=12&Day=19

  It can be altered like this:

  http://www.example.com/Blog/2006/12/19/



Apache HTTP Server
.htaccess files

Apache HTTP Server

.htaccess files

- To **configure the details of website** without needed to alter the server config files
- The **placement of the .htaccess file is important** because the configurations will affect everything in its directory and the directories under it

Apache HTTP Server

.htaccess files

Things to be Aware of

- **Speed**
  - The .htaccess may **slow down server** somewhat
  - Each time a page loads, the server scans its directory, and any above it until it reaches the highest directory or an .htaccess file

    This process will occur as long as the **AllowOverride allows the use of .htaccess** files

Apache HTTP Server

.htaccess files

Things to be Aware of

- **Security**
  - The .htaccess file is **much more accessible** than standard apache configuration and the **changes are made live instantly** (without the need to restart the server)
  - Granting users permission to make alterations in the .htaccess file gives them a lot of control over the server itself

Apache discourages the use of the .htaccess

Apache HTTP Server

.htaccess files

- **Manual**
  - **Activate Mod_Rewrites:** a2enmod rewrite
  - **Activate an .htaccess file**
    - ✓Edit the configuration to allow the .htaccess file to override standard website configs
    - ✓Open the apache2 default host configuration file: /etc/apache2/sites-available/default
    - ✓Find the following section, and change the line that says AllowOverride from None to All

```
<Directory /var/www/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride All
                Order allow,deny
                allow from all
</Directory>
```

    - ✓Save and exit that file, restart Apache: apache2 restart

Apache HTTP Server

.htaccess files

- **Manual**
  - **Rewrite URLS**
    - ✓ The entire URL rewriting operation takes place within the .htaccess file
    - ✓ All URL rewrite commands follow the same pattern

RewriteRule Pattern Substitution [OptionalFlags]

- ➢ **RewriteRule:** the name of the the mod_rewrite directive
- ➢ **Pattern:** This section is dedicated to interpreting the requested URL, using regular expressions
- ➢ **Substitution:** This is the actual URL of the page with the information we want to display

Apache HTTP Server

.htaccess files

- **Manual**
  - **Rewrite URLS**
    - ✓ The entire URL rewriting operation takes place within the .htaccess file
    - ✓ All URL rewrite commands follow the same pattern

RewriteRule Pattern Substitution [OptionalFlags]

- **Optional Flags:** A flag is a tag at the end of the Rewrite Rule directive that may change the behavior of the expression
  - [F]: make the URL forbidden
  - [NC]: force the rule to disregard capitalization
  - [R=301] or [R=302]: control the redirect code you want to use
  - [L]: indicate that this is the last rule in a series

Apache HTTP Server

.htaccess files

- **Examples**
  - **Example 1:** Go to Page A, find page B

    RewriteEngine on

    RewriteRule ^oranges.html$ apples.html
  - **Example 2:** The website has a parameter in its URL. How to make it look like a subdirectory

    ✓Check out this URL: http://example.com/results.php?products=apple

    ✓It would be much clearer displayed as: http://example.com/products/apple

    ✓The lines within the .htaccess file:

    RewriteEngine on

    RewriteRule ^products/([A-Za-z0-9-]+)/?$ results.php?products=$1 [NC]

Apache HTTP Server

.htaccess files

- **Examples**
  - **Example 3:** The site has an unwieldy URL. How to clean it up
    - ✓This sort of situation can arise when URLs are long and complex
    - ✓Take the URL below:

      http://example.com/results.php?products=produce&type=fruit&species=apple
    - ✓URL rewrites would allow you to convert the URL to something simpler and clearer:

      http://example.com/produce/fruit/apple
    - ✓Need the following lines in .htaccess file

      RewriteEngine on

      RewriteRule ^(meat|produce|dairy)/([^/.]+)/([^/.]+)$ results.php?products=$1&type=$2&species=$3

Apache HTTP Server

.htaccess files

- **Manual**
  - **Rewrite Conditions**
    - ✓**Example 1:** How To Prevent Hotlinking
      - **Hotlinking** is the process of using an image or object from one server on another one
      - Prevent hotlinking by redirecting all the links to an object on site to some other less pleasant image, or by forbidding the operation altogether

RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^http://(www\.)?example\.com/.$ [NC]*
RewriteRule .\.(gif|jpeg|png)$ http://www.example.com/unpleasantness.jpg [R,NC,L]*

Or (for last line)
RewriteRule .\.(gif|jpeg|png)$ - [F]*

Apache HTTP Server

.htaccess files

- **Manual**
  - **Rewrite Conditions**
    - ✓**Example 2:** How to add www to a URL
      - ➢ It’s easy for a person to see that example.com and www.example.com are the same site, search engines register them as duplicates, hurting their rankings
      - ➢ To be sure that the www is always attached:

        RewriteEngine on

        RewriteCond %{HTTP_HOST} ^example\.com$

        RewriteRule ^(.)$ http://www.example.com/$1 [R=301]*

Apache HTTP Server

.htaccess files

- **Manual**
  - **Rewrite Conditions**
    - ✓**Example 3:** Blocking a Specific IP Address

This a useful tool to prevent malicious parties at specific IP addresses from accessing a site

RewriteCond %{REMOTE_ADDR} ^(12\.34\.56\.789)$

RewriteRule (.) - [F,L]*

Apache HTTP Server

.htaccess files

- **mod_rewrite Cheat Sheet**
  - A quick reference guide for mod_rewrite, with rewrite flags, regular expression syntax and sample rules
  - Link: https://cheatography.com/davechild/cheat-sheets/mod-rewrite/

NGINX Rewrite Rules

NGINX Rewrite Rules

Rewrite rules **change part or all URL** in a client request, usually for one of two purposes:

- To inform clients that the resource they're requesting now resides at a different location
  - Changed Domain name
  - Want clients use a canonical URL format (with or without www)

  → The **return** and **rewrite** directives are suitable

- To control the flow of processing

  E.g., forward requests to an application server when content needs to be generated dynamically

  → The **try_files** directive is often used

The return Directive

- The simpler of the two general-purpose directives
- Recommend using it instead of rewrite when possible
- **Enclose** the return in a **server** or **location context** that specifies the URLs to be rewritten, and it **defines the corrected (rewritten) URL** for the client to use in future requests for the resource

```
server {
    listen 80;
    listen 443 ssl;
    server_name www.old-name.com;
    return 301 $scheme://www.new-name.com$request_uri;
}
```

The return Directive

- The **return** directive is simple to use, and **suitable when the redirect meets two conditions**:
  - the rewritten URL is appropriate for every request that matches the server or location block,
  - you can build the rewritten URL with **standard NGINX variables**

The rewrite Directive

- To test for **more complicated distinctions** between URLs, **capture elements** in the original URL that don’t have corresponding NGINX variables, or **change or add elements** in the path

- **Enclose** the rewrite directive in a **server or location context** that defines the URLs to be rewritten

- **Syntax:**

  rewrite regex URL [flag];

The rewrite Directive

```
server {
    # ...
    rewrite ^(/download/.*)/media/(\w+)\.?.*$ $1/mp3/$2.mp3 last;
    rewrite ^(/download/.*)/audio/(\w+)\.?.*$ $1/mp3/$2.ra  last;
    return  403;
    # ...
}
```

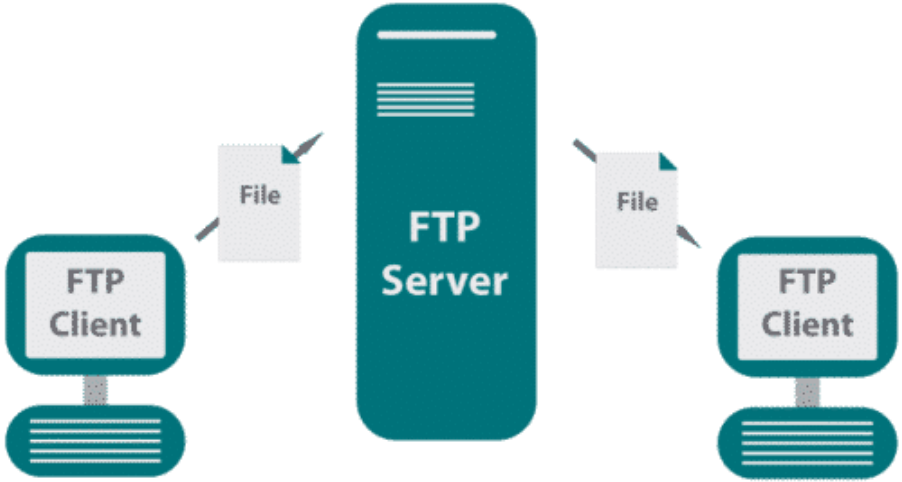
The try_files directive

- The try_files directive is placed in a **server** or **location block**
- It takes a list of one or more files and directories and a final URI

  try_files file ... uri;

  - NGINX checks for the existence of the files and directories in order, and serves the first one it finds
  - If none of the files or directories exist, NGINX performs an internal redirect to the URI defined by the final element (uri)

```
location /images/ {
    try_files $uri $uri/ /images/default.gif;
}

location = /images/default.gif {
    expires 30s;
}
```

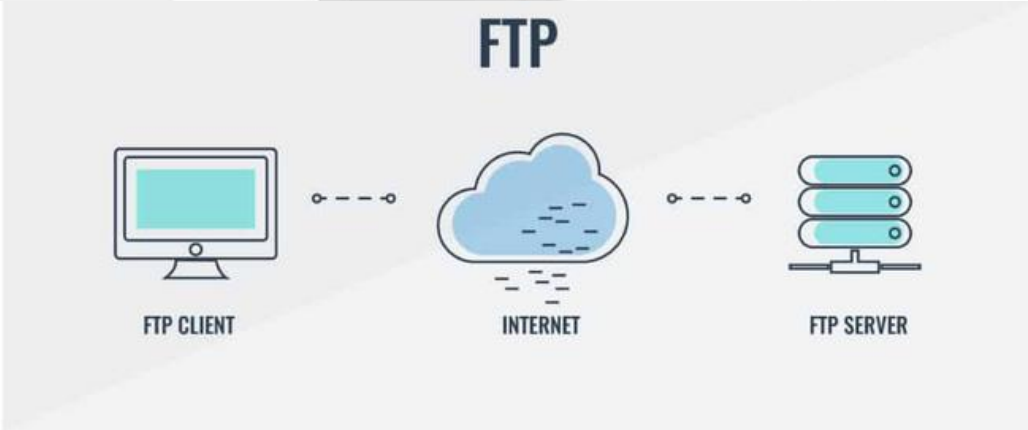


FTP, FTPS, SFTP

File Transfer Protocol

FTP

FTP is a network protocol that was once widely used for moving files between a client and server



File Transfer Protocol Secure

FTPS

- This is an extension of the popular FTP that supports TLS and SSL

- Vsftpd is probably the mostly secure and fastest FTP server for Unix-like systems

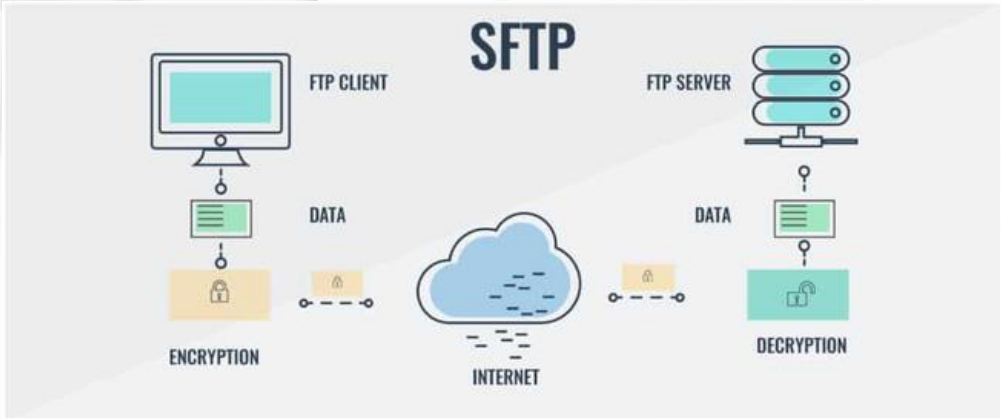
  It is optimized for security, performance, and stability, offers strong protection against many security problems found in other FTP servers

SSH Secure File Transfer Protocol

SFTP

- SFTP is a secure FTP and runs over the SSH protocol
- SFTP supports the full security and authentication functionality of SSH
- SFTP also protects against password sniffing and man-in-the-middle attacks

SFTP protects the integrity of the data using encryption and cryptographic hash functions, and authenticates both the server and the user



Exercise

Install Web server on Windows OS and Ubuntu OS


Q&A

References – Virtual Host

- **How to Configure Multiple Sites with Apache**
  https://www.liquidweb.com/kb/configure-apache-virtual-hosts-ubuntu-18-04/
- **How To Set Up Apache Virtual Hosts on Ubuntu 18.04**
  https://www.digitalocean.com/community/tutorials/how-to-set-up-apache-virtual-hosts-on-ubuntu-18-04
- **How To Install the Apache Web Server on Ubuntu 20.04**
  https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04

References

- **Apache Tutorials for Beginners**

  https://www.guru99.com/apache.html
- **HTTPD FAQ**

  https://cwiki.apache.org/confluence/display/httpd/FAQ#FAQ-WhatisApache?
- **Understanding a Web Server and Types of Web Servers**

  https://www.milesweb.com/blog/hosting/web-server-types-web-servers

References – Hosting

- **Web Hosting**

  https://www.website.com/beginnerguide/webhosting/6/1/what-is-web-hosting?.ws
- **What is Web Hosting?**

  https://www.namecheap.com/hosting/what-is-web-hosting-definition/
- **What is VPS Hosting?**

  https://www.hostinger.com/tutorials/what-is-vps-hosting
- **What Is VPS?**

  https://www.dreamhost.com/blog/beginners-guide-vps/

References – Hosting

- **What is DNS?**

  https://www.namecheap.com/dns/what-is-dns-domain-name-system-definition/
- **What Is DNS? | How DNS Works**

  https://www.cloudflare.com/learning/dns/what-is-dns/

References

- **What is a web server?**

  https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server
- **How much does it cost to do something on the Web?**

  https://developer.mozilla.org/en-US/docs/Learn/Common_questions/How_much_does_it_cost
- **Ubuntu tutorial**

  https://ubuntu.com/tutorials?topic=server&sort=difficulty-asc&page=2

References – FTP

- **How do you upload your files to a web server?**

  https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Upload_files_to_a_web_server
- **What software do I need to build a website?**

  https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_software_do_I_need
- **FTP Server**

  https://ubuntu.com/server/docs/service-ftp

References

- **Apache HTTP Server Tutorial: .htaccess files**

  https://httpd.apache.org/docs/current/howto/htaccess.html
- **Creating NGINX Rewrite Rules**

  https://www.nginx.com/blog/creating-nginx-rewrite-rules/
- **How To Create Temporary and Permanent Redirects with Nginx**

  https://www.digitalocean.com/community/tutorials/how-to-create-temporary-and-permanent-redirects-with-nginx
- **Converting Apache Rewrite Rules to NGINX Rewrite Rules**

  https://www.nginx.com/blog/converting-apache-to-nginx-rewrite-rules/

References

- **Apache Module mod_rewrite**

  https://httpd.apache.org/docs/2.4/mod/mod_rewrite.html
- **Redirecting and Remapping with mod_rewrite (Apahce)**

  https://httpd.apache.org/docs/2.4/rewrite/remapping.html

Regular Expressions Overview

What is a RegEx and what is it useful for?

Regular Expressions Overview

- A regular expression is a **set of characters that defined a pattern used to match character combinations** in strings

  Very **powerful for find/replace** type of operations

- Some examples when regular expressions are powerful:
  - Find and extract data from a document

    Extract image source from HTML, extract exceptions/errors from logs
  - Validate data provided as text:

    Passwords, emails, mobile numbers, url

Regex Syntax

- Regular expressions are an extremely powerful tool implemented in most languages
- Yet, regular expressions have their own syntax and use of special characters
  - Difficult to remember unless used frequently
  - MDN Regex reference
- Regular expressions can be tested at:
  - http://www.regexr.com/
  - https://regex101.com/

Regular Expression Flags

Regular expression have optional flags that allow for global and case insensitive searching

These flags can be used separately or together in any order

Flag	Description
g	Global search
i	Case-insensitive search
m	Multi-line search

Regex Special Characters

Groups, ranges, word boundaries,...

Special Characters in Regex

- Regular expressions have a set of special characters, that have a different behavior
  - Characters for matching multiple characters
  - Characters for matching whitespace
  - Characters for matching digits
  - Characters for matching letters
  - Etc.
- Complete list of special characters can be found here

Special Characters in Regex

Quantifiers define quantities

- ***** – The preceding character/group is matched 0 or more times
- **+** – Almost the same behaviour as * - the preceding character/group is matched 1 or more times
- **?** – The preceding character/group is matched 0 or 1 times
- **.**(dot) – matches any single character except the newline character

Special Characters in Regex

Brackets are used to find a range of characters

- **|** – Matches one pattern or the other
- **[xyz]** – Character set - Matches any of the characters
- **[x-z]** – Character set - Matches any one between the characters range
- **[^xyz]** – Inverted characters set - Matches all other characters

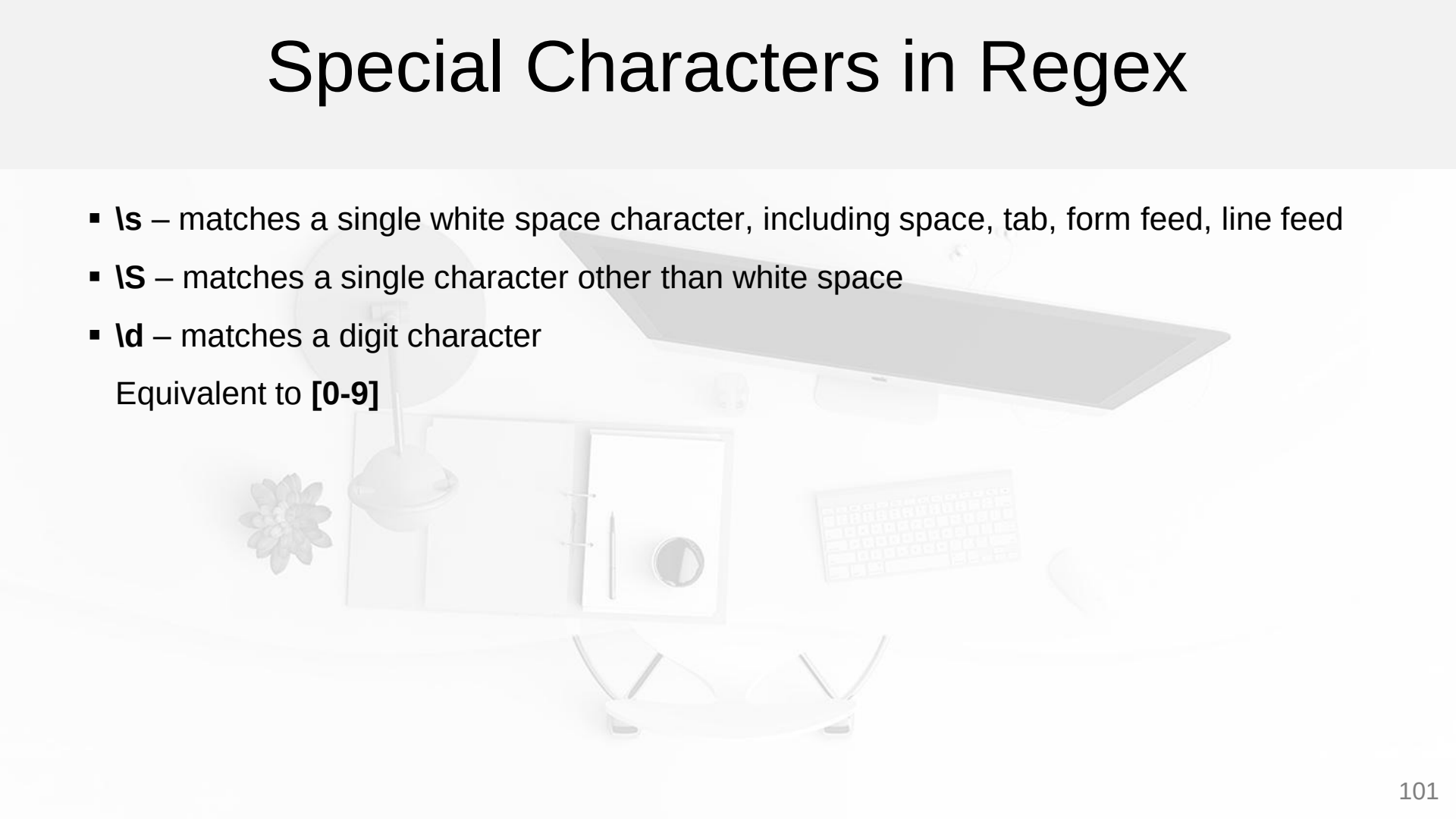
Special Characters in Regex

- **{N}** – matches exactly **N** occurrences of the preceding character/group
- **{N, M}** – matches at least **N** and at most **M** occurrences of the preceding character/group
- **^** - matches the start of the string
- **$** matches the end of the string

Special Characters in Regex

- **\s** – matches a single white space character, including space, tab, form feed, line feed
- **\S** – matches a single character other than white space
- **\d** – matches a digit character

  Equivalent to **[0-9]**



Special Characters in Regex

- **\D** – matches any non-digit character

  Equivalent to **[^0-9]**
- **\w** – matches any alphanumeric character including the underscore (_)
- **\W** – matches any non-alphanumeric or underscore character