Web hosting with Linux

CESI

Geoffrey THERY

Who am i ? who are you ?

What did you expect from this 2 days course ?

Topics covered / non covered

Agenda

Reminder : it’s an interactive session, feel free if you have questions !!
help each other if needed

Shared Cryptpad : **https://bit.ly/38zCtOD**

Internet

What is internet ?
a quick reminder

Introduction

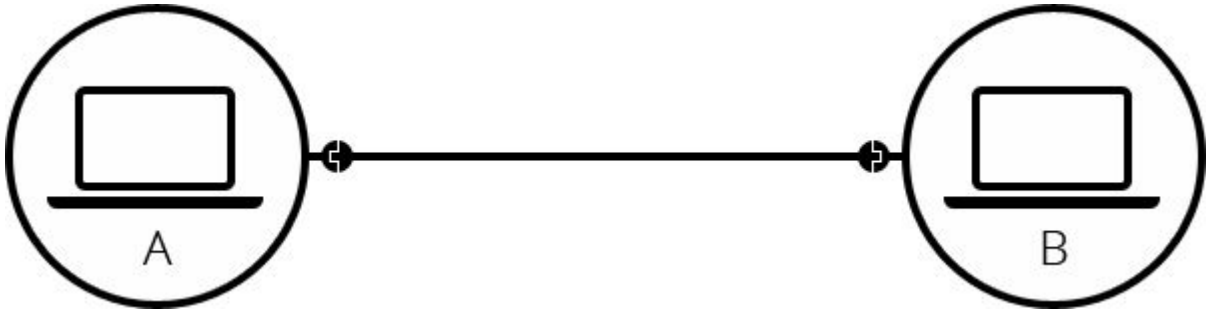
We all use web services today with a lot of devices (laptop, smartphone, Connected Objects, ...)

But, basically, how does it works ?

Internet ?

Basically, Internet is a large network of computers, servers, devices communicating together

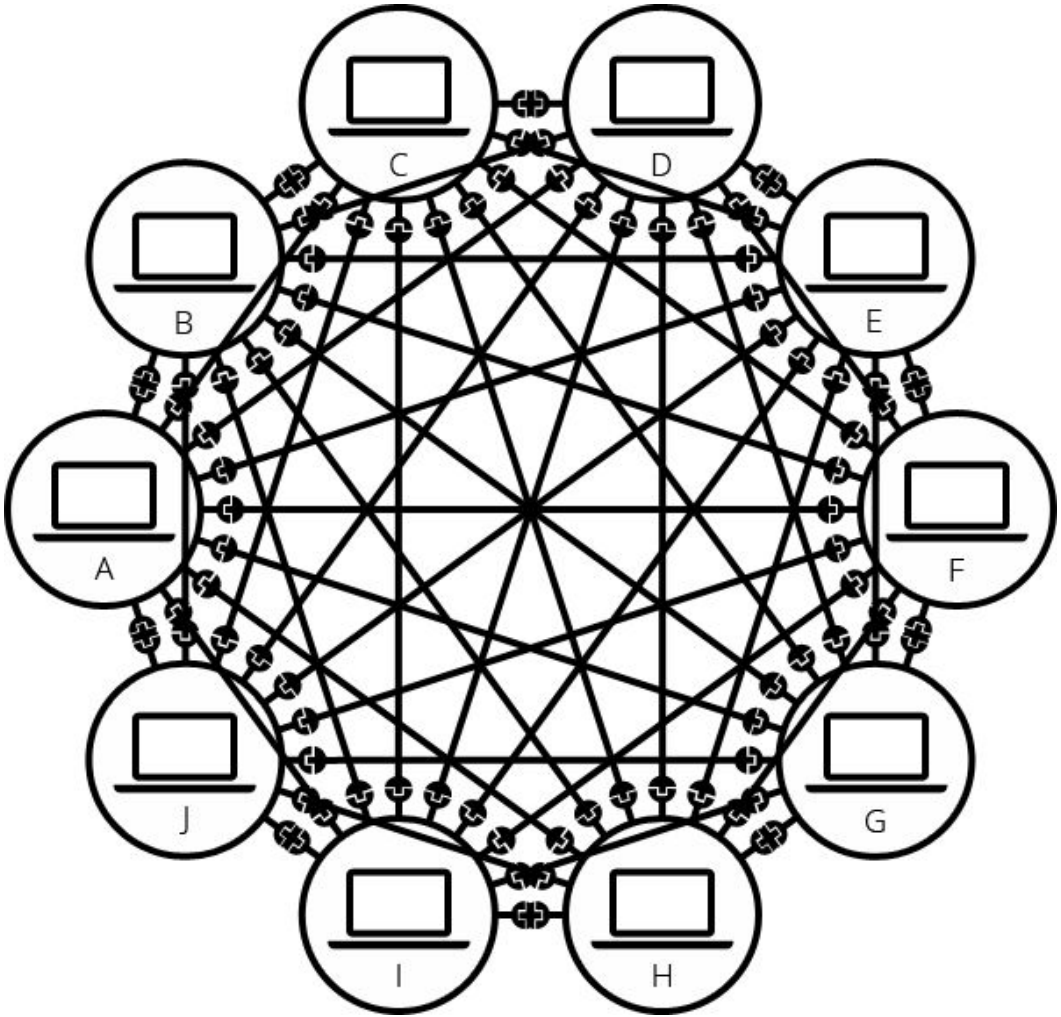
It’s easy to connect 2 computers together :



Internet ?

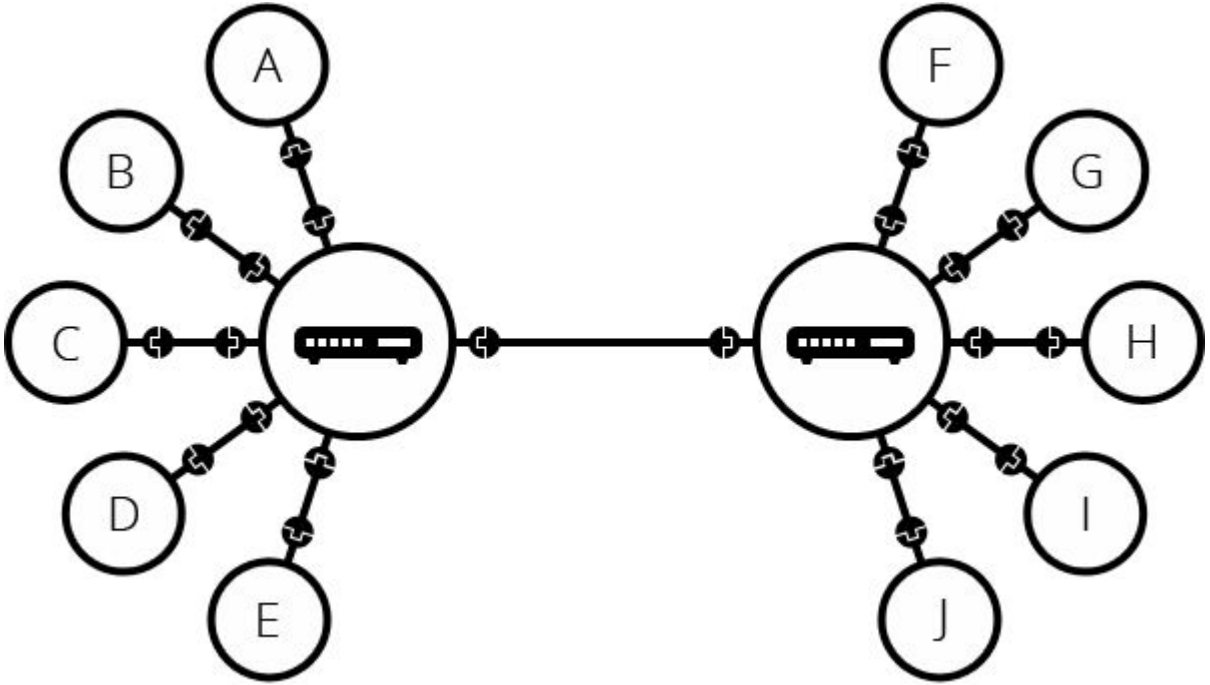
How to connect a lot of computer ?

It’s complicated to connect them directly



Internet ?

1st step : we can add routers



A **router** is hardware device designed to receive, analyze and move incoming packets to another network.

Internet ?

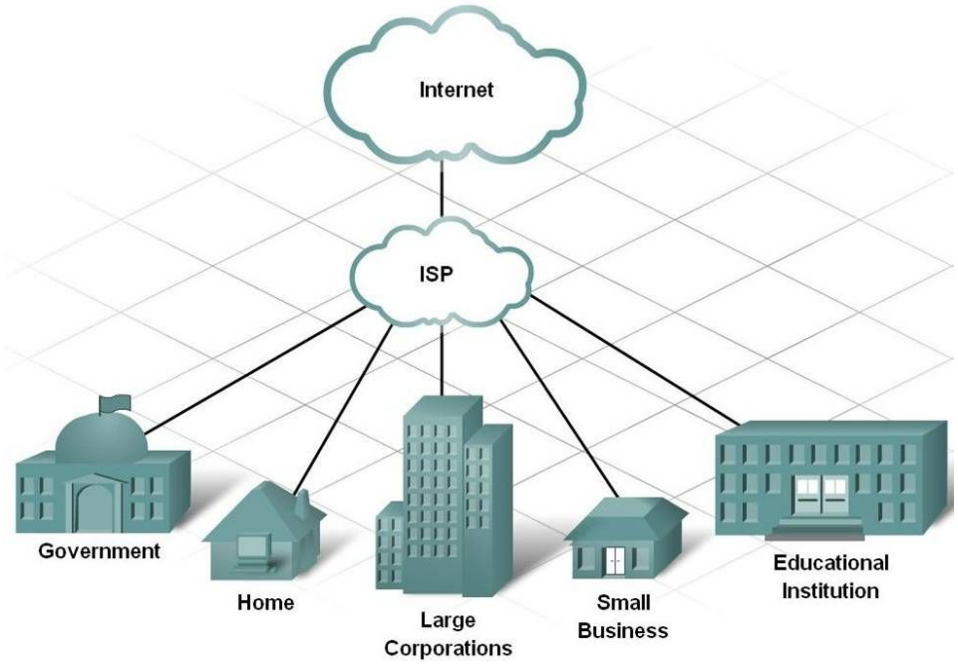
But there is still something missing :

The previous schema may be applicable for an enterprise network, but not for connecting to a different network outside the company.

We have to connect to an ISP :

Internet ?

An ISP is a company that manages some *routers* that are all linked together and can also access other ISPs' routers



Web

Internet vs Web
How the web works ?
The HTTP protocol

Internet vs Web ?

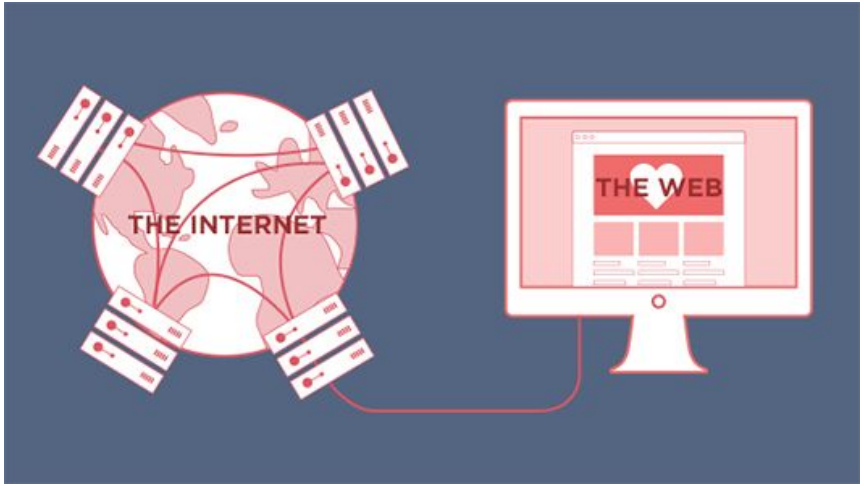
Internet is a technical infrastructure which allows billions of computers to be connected all together.

Among those computers, some computers (called *Web servers*) can send messages intelligible to web browsers.

The *Internet* is an **infrastructure**, whereas the *Web* is a **service** built on top of the infrastructure.

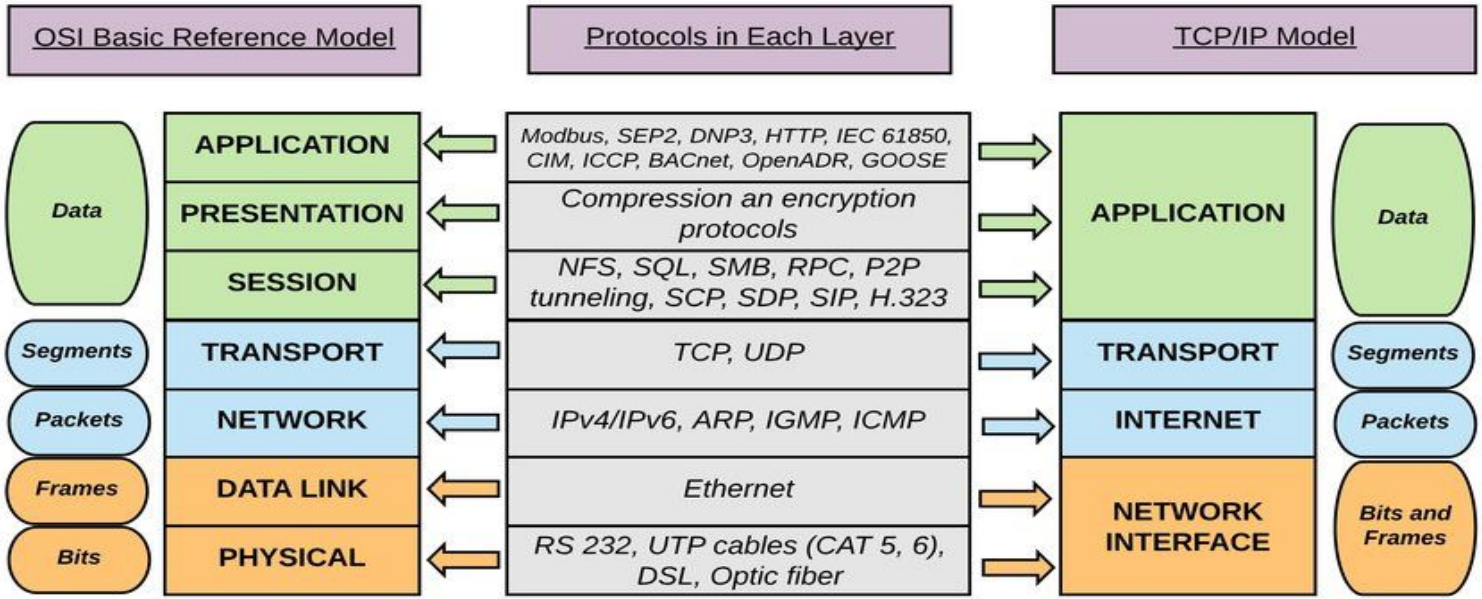
It is worth noting there are several other services

built on top of the Internet, such as email and IRC.

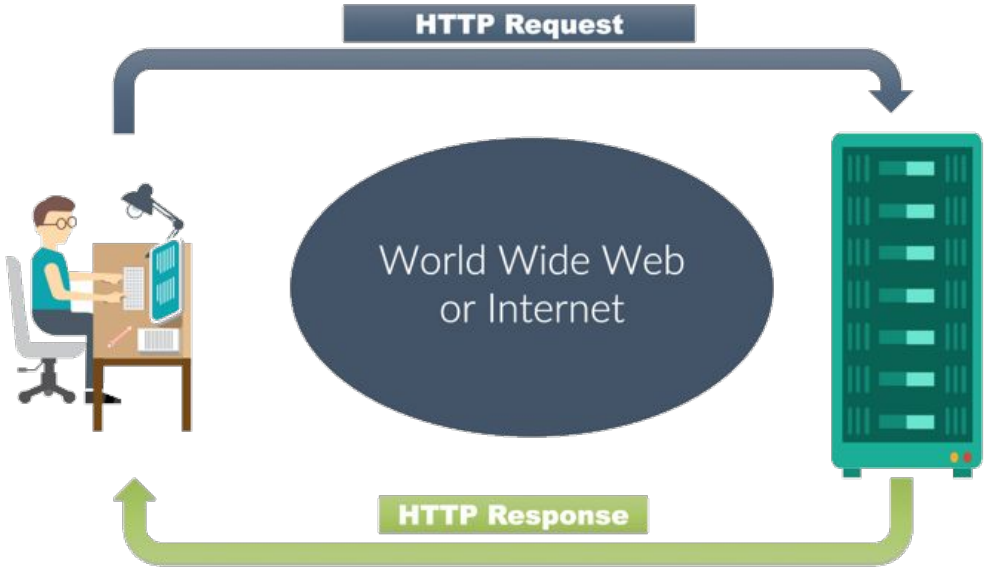


How the web works ?

Reminder : OSI and TCP/IP model

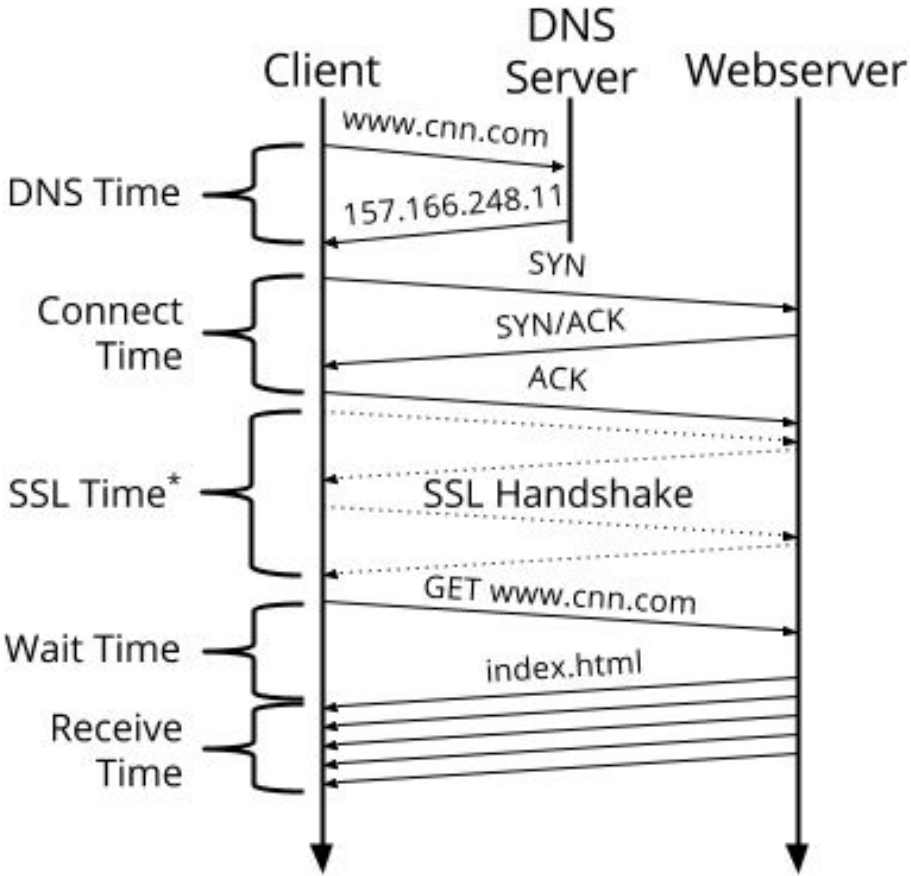


How the web works ?



What is the complete sequence of an **HTTP request** ?

Back to basics : client server model



Back to basics : client server model

Web browsers communicate with web servers using the **H**yper**T**ext **T**ransfer **P**rotocol (HTTP).

This request includes:

- A URL identifying the target server and resource (e.g. an HTML file, a particular data point on the server, or a tool to run).
- A method that defines the required action (for example, to get a file or to save or update some data) :
- GET
- POST
- PUT
- HEAD
- DELETE
- PATCH
- OPTIONS

https://www.w3schools.com/tags/ref_httpmethods.asp

Focus on HTTP

HTTP is a protocol (a client-server protocol) which allows the fetching of resources, such as HTML documents

The messages sent by the client, usually a Web browser, are called *requests* and the messages sent by the server as an answer are called *responses*.

It is an application layer protocol that is sent over TCP, or over a TLS-encrypted TCP connection

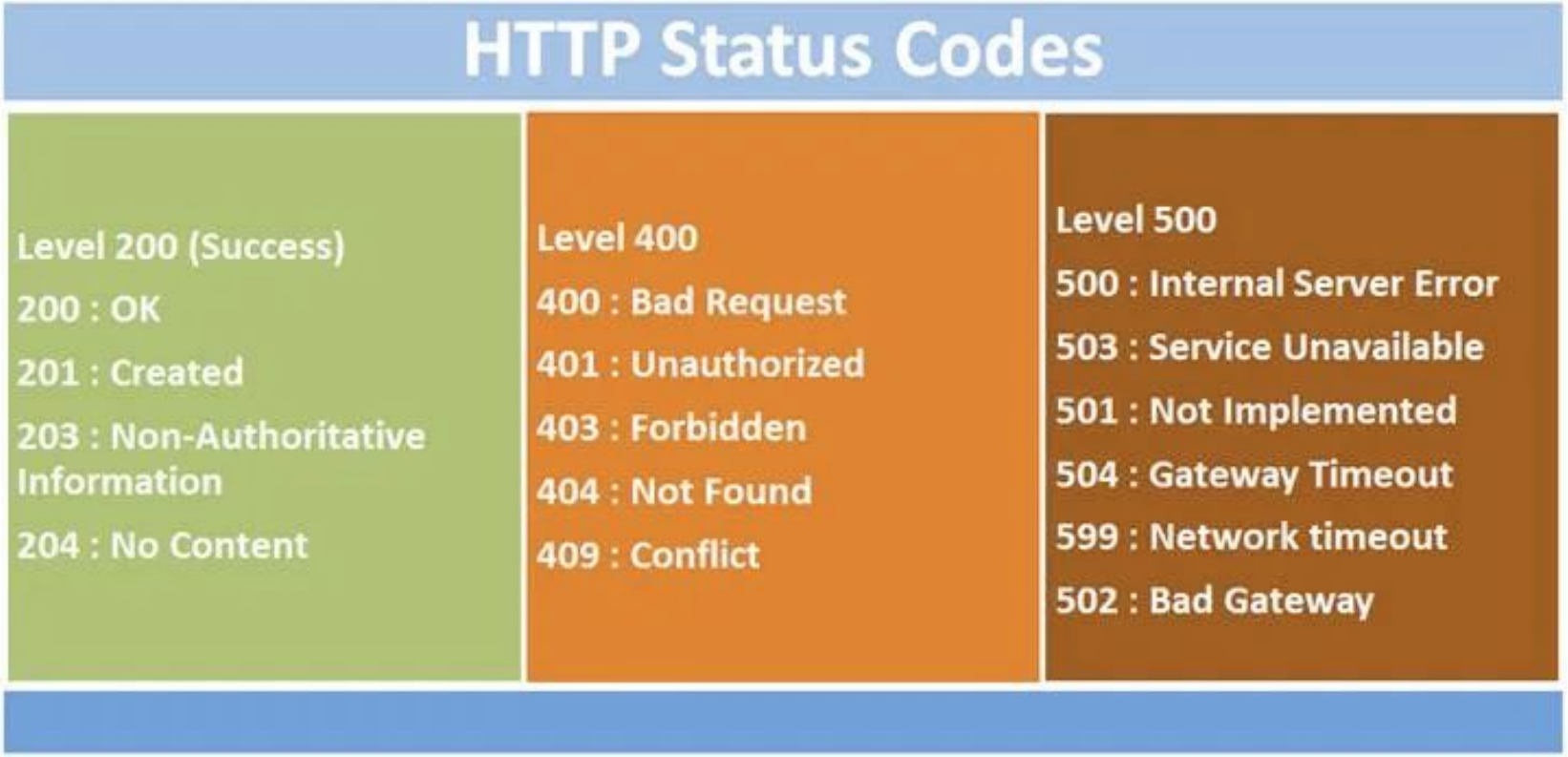
HTTP Flow

When a client wants to communicate with a server, it performs the following steps:

1. Open a TCP connection: The TCP connection is used to send a request, or several, and receive an answer.
2. Send an HTTP message: HTTP messages are human-readable. For example:
   `GET / HTTP/1.1 Accept-Language: fr`
3. Read the response sent by the server, such as:

   `HTTP/1.1 200 OK Date: Sat, 09 Oct 2019 14:28:02 GMT Server: Apache Last-Modified: Tue, 01 Dec 2018 20:18:22 GMT ETag: "51142bc1-7449-479b075b2891b" Accept-Ranges: bytes Content-Length: 29769 Content-Type: text/html <!DOCTYPE html... (here comes the 29769 bytes of the requested web page)`
4. Close or reuse the connection for further requests.

HTTP Flow

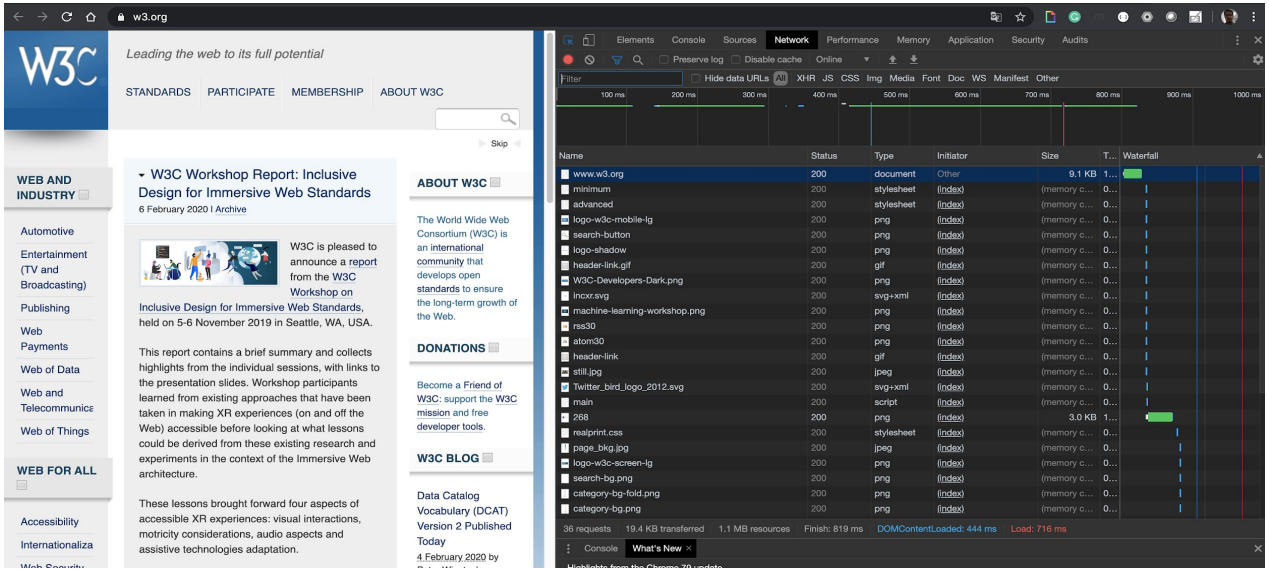


RFC 2616 : https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

HTTP Flow

Some tools to inspect http flow :

- wireshark
- developers tools on browser
- tcpdump
- ...



Web servers

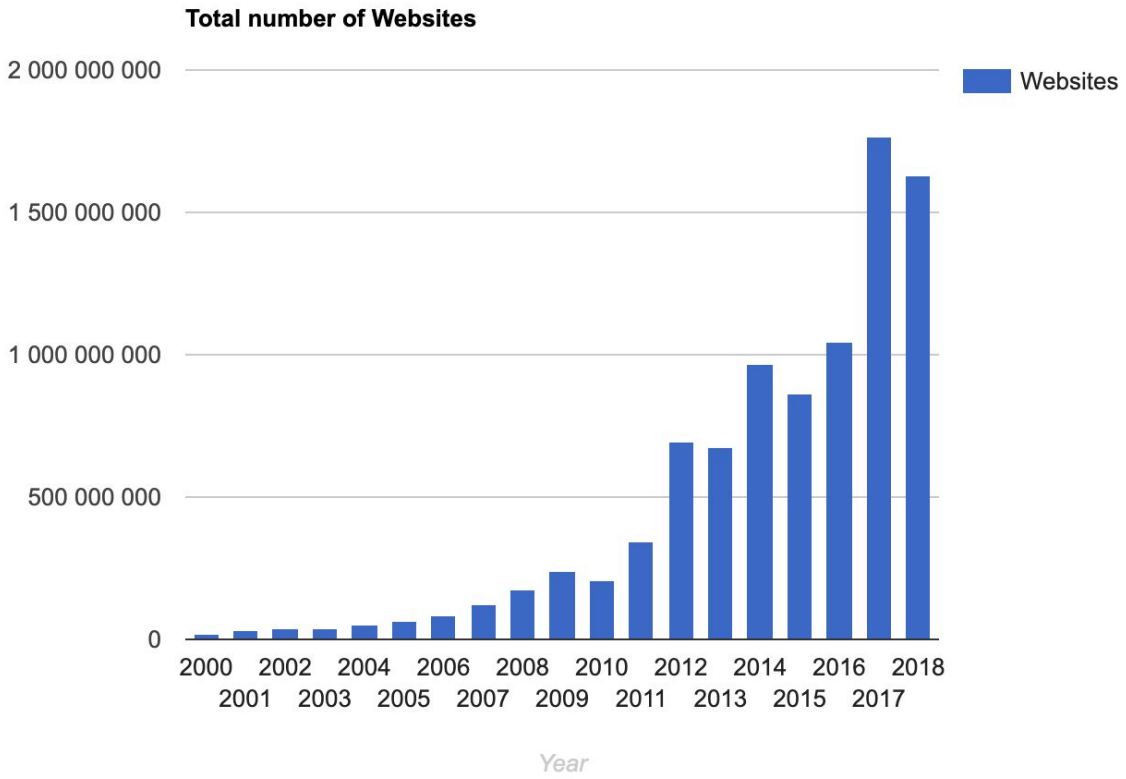
Operating Systems
Focus on Linux

Websites

A “website” means an unique hostname.

You can have :

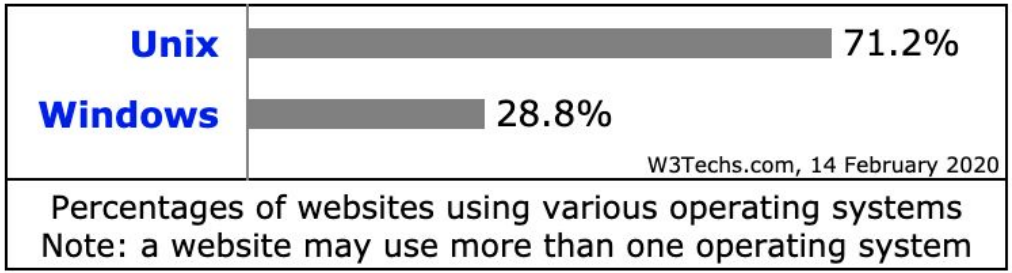
- several websites on the same servers
- a lot of servers for the same website



Web server operating system

Generally speaking, the default operating system for a server is some form of Linux/Unix.

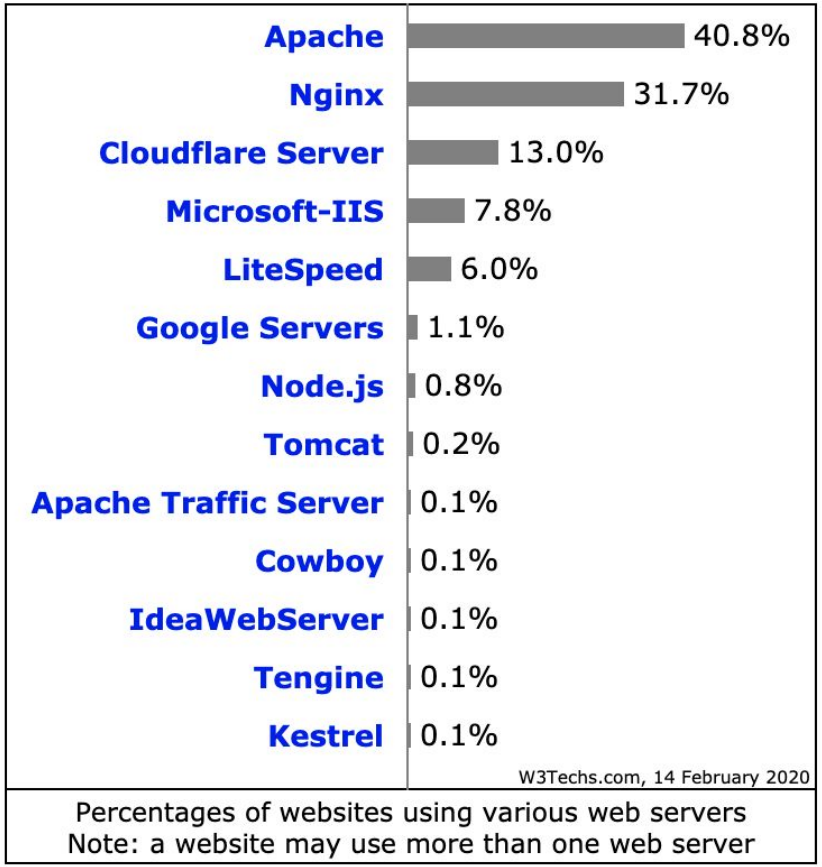
But, Windows OS are also used to host .NET framework, ASP.NET and MSSQL technologies.



source : https://w3techs.com/

Web servers

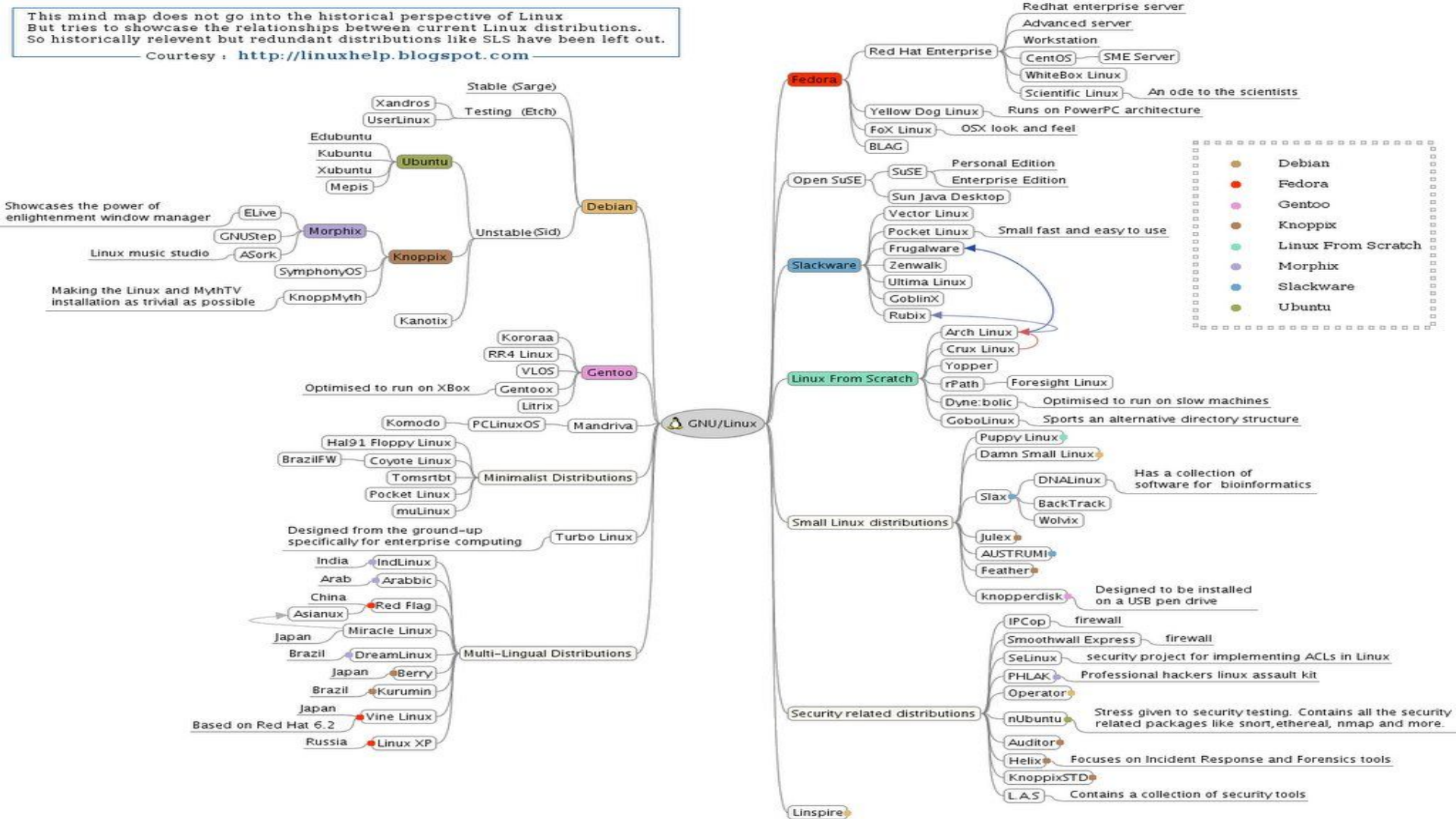

Apache
HTTP SERVER
NGINX
http server
LIGHTTPD
fly light.
Microsoft
IIS
Internet
Information
Services


Apache 40.8%
Nginx 31.7%
Cloudflare Server 13.0%
Microsoft-IIS 7.8%
LiteSpeed 6.0%
Google Servers 1.1%
Node.js 0.8%
Tomcat 0.2%
Apache Traffic Server 0.1%
Cowboy 0.1%
IdeaWebServer 0.1%
Tengine 0.1%
Kestrel 0.1%
W3Techs.com, 14 February 2020
Percentages of websites using various web servers
Note: a website may use more than one web server

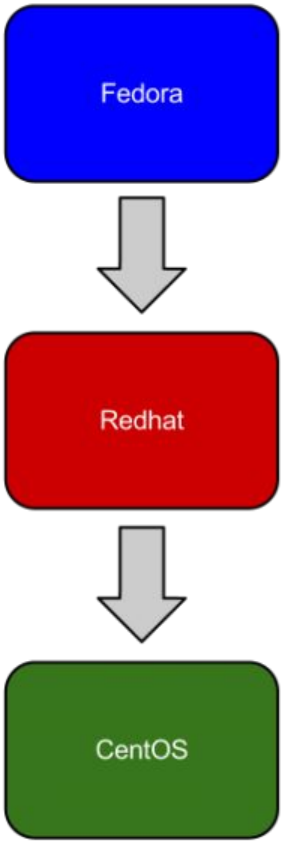
Linux for web hosting

This mind map does not go into the historical perspective of Linux
But tries to showcase the relationships between current Linux distributions.
So historically relevent but redundant distributions like SLS have been left out.

Courtesy : http://linuxhelp.blogspot.com



Centos



- Run by Redhat (company)
- Community driven
- Focused on quick releases (~6 Months)
- Stresses features and functionality
- Free

- Based on Fedora
- Run by Redhat (company)
- Released corporately by Redhat
- Focused on long releases for stability
- Stresses stability over features
- Commercial (non-free)

- Based off of commercial releases of Redhat (distro)
- Run by the community
- Basically Redhat without the cost or support

Centos / Redhat Basics

Package manager
Service management

Linux survival kit

Redhat/Centos use RPM as package manager

What an RPM contains ?

RPMs

An RPM contains :

- Files archive
- Metadata :
  - Informations to describe package (name, version, architecture)
  - Scripts
  - Files list
  - Files attributes
  - Dependencies list

```
Install a package
#rpm -ivh openssl-1.0.2k-8.el7.x86_64.rpm

Update a package
#rpm -Uhv openssl-1.0.2k-8.el7.x86_64.rpm

Uninstall a package
#rpm -evh openssl-1.0.2k-8.el7.x86_64
```

YUM

YUM is a RPM package manager

Yum manages with installation, remove, search, update of softwares

Yum downloads packages from repositories

Yum installs sotwares and their dependencies

Yum is a RPM overlay

YUM usage

search : search all the packages with “httpd” pattern

```
#yum search httpd
```

list : search the package named “httpd”

```
#yum list httpd
```

info : get informations about the httpd package

```
#yum info httpd
```

install : install httpd and its dependencies

```
#yum install httpd
```

remove : remove httpd package

```
# yum remove httpd
```

SYSTEMD

Systemd is a service manager, it replaces sys init in RedHat 7 CentOS 7 :

- first process launched by the kernel (PID)
- services initialization
- Manage linux daemons : start, stop, enable, disable
- start service in //
- ressource management (cgroups)
- manage filesystem mounts

Managing services with SystemD

Basic service management :

```
systemctl enable/disable service-name
systemctl status/stop/start/reload service-name
```

List active service :

```
systemctl list-units --type=service
```

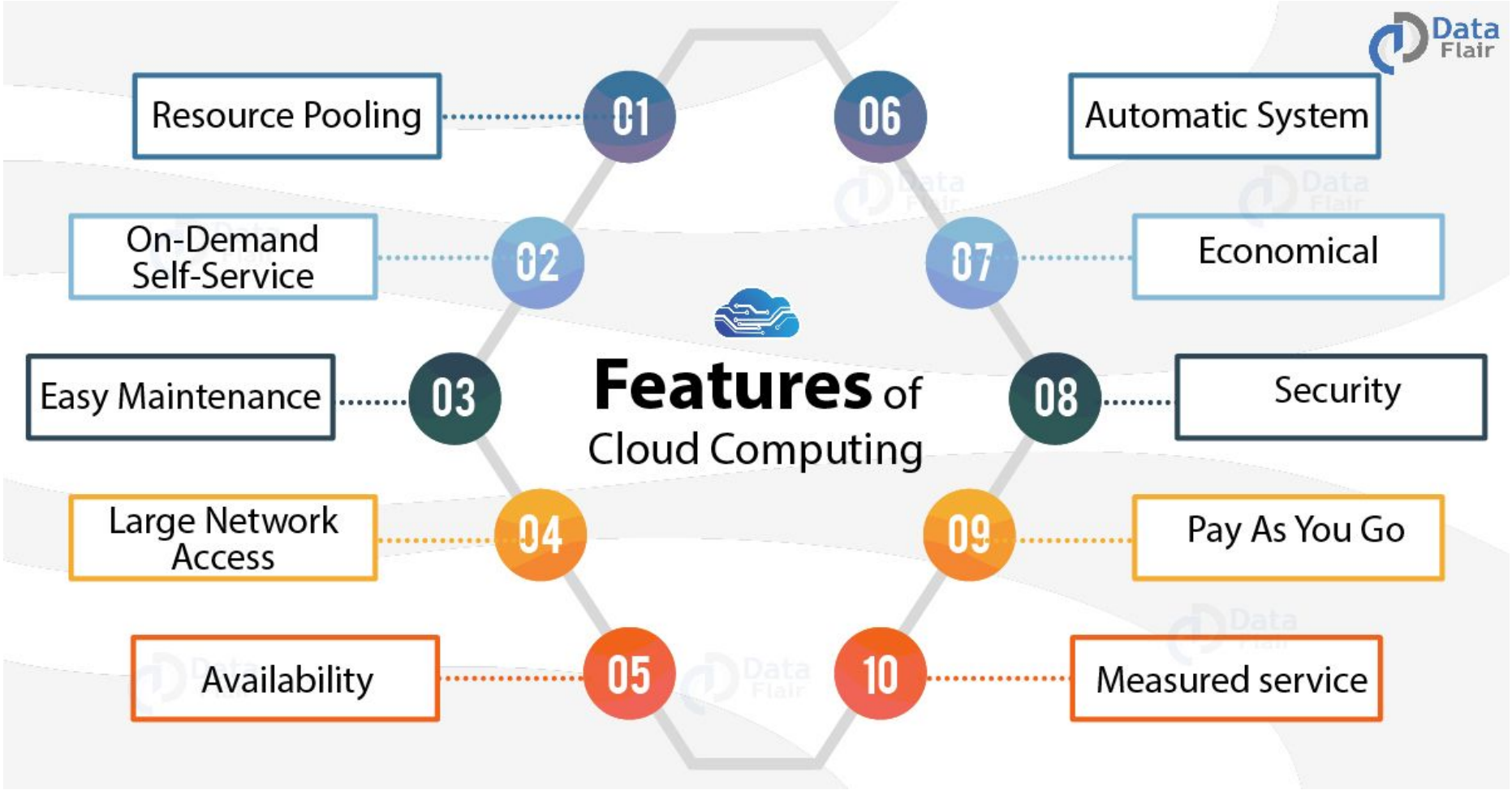
See/Edit service configuration :

```
systemctl cat/edit service-name
```

Reload when a service has changed :

```
systemctl daemon-reload
```

Cloud Platform


Data Flair
Features of Cloud Computing
01 Resource Pooling
02 On-Demand Self-Service
03 Easy Maintenance
04 Large Network Access
05 Availability
06 Automatic System
07 Economical
08 Security
09 Pay As You Go
10 Measured service

HYBRID CLOUD

- Combination of both public and private cloud
- Shared security responsibility
- Helps maintain tighter controls over sensitive data and processes

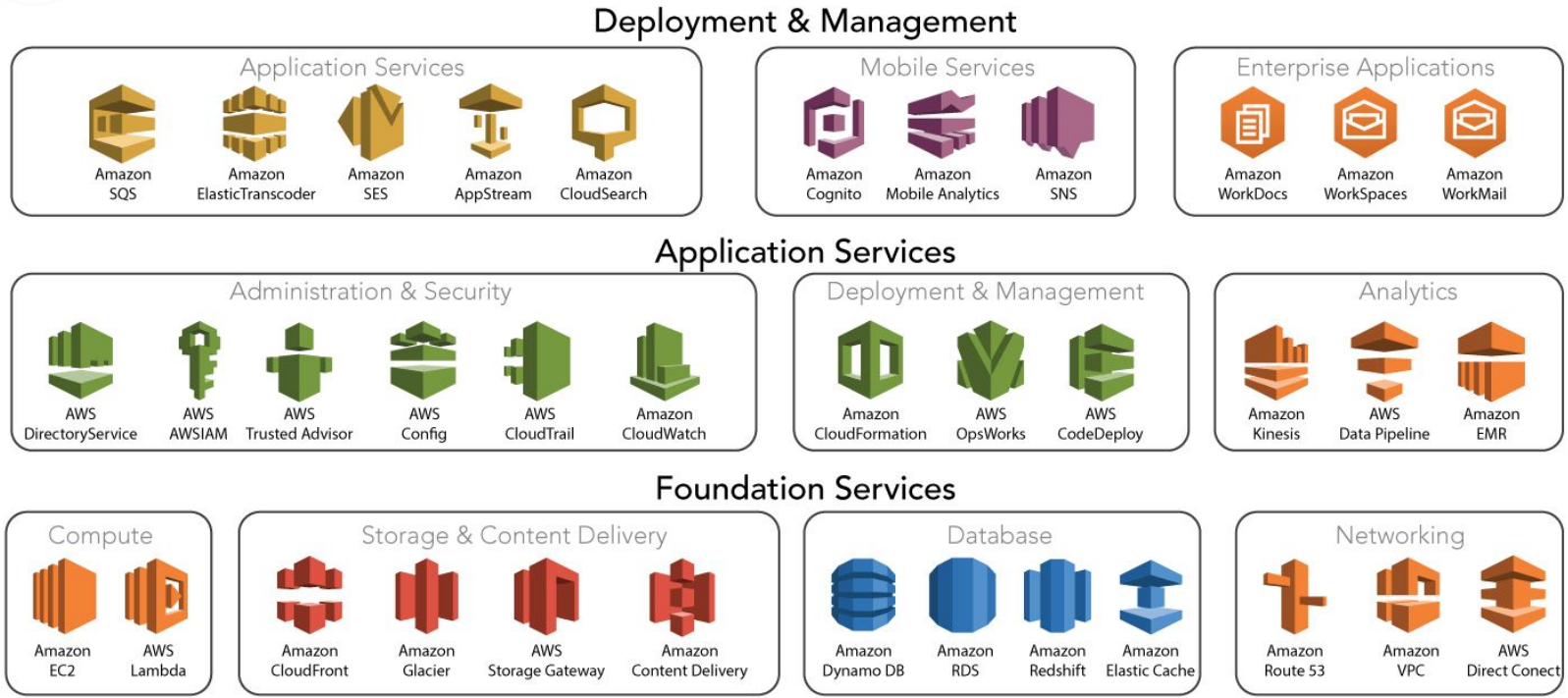
PUBLIC CLOUD

- Offered by third-party providers
- Available to anyone over the public internet
- Scales quickly and convenient

PRIVATE CLOUD

- Offered to select users over the internet or a private internal network
- Provides greater security controls
- Requires traditional datacenter staffing and maintenance

AWS Services



AWS : key words

EC2
A virtual server instance you can start and stop from the AWS console.

EBS
The “Elastic Block Storage” provides virtual harddisk that can be assigned to an EC2 instance.

Elastic IP
As the name implies, these are the IP addresses you can assign to the EC2 instance.

Security Groups
A more common term for this is “Firewall”. These firewall rule-sets can be assigned to the EC2 instance.

Key Pairs
When a new EC2 instance is created you assign a key pair to be able to login initially.

Stop talking, start playing

Lab environment setup

- Create one subnet in eu-central
- Create an Internet Gateway and attach it to the VPC.
- Create a route table; with a route 0.0.0.0/0 to the Internet Gateway, and attach it to the subnet.
- Create a key pair and download it
- Deploy a Cloud9 environment in the public subnet (it will be used to connect your server. You can also use your computer if you want)

LAB 1

- Deploy a CentOS 7 instance in your account

  type : t2.micro

  With a public IP

  Open the following ports : 22, 80

  /!\ Do not forget to add your key at the instance creation

- Connect to your server using ssh (through Cloud9 or your laptop) and install the system updates.

LAB 1

- Install Apache using yum
- Enable and start the service using systemctl
- Check the log of httpd service
- Access the public IP of your server on port 80 (web browser or curl)

Httpd config overview

On Centos, httpd configuration is located on /etc/httpd/ (/etc/apache2 on debian)

The principal configuration file is named httpd.conf

```
[root@ip-172-31-100-185 conf]# cat httpd.conf
#
# This is the main Apache HTTP server configuration file.  It contains the
# configuration directives that give the server its instructions.
# See <URL:http://httpd.apache.org/docs/2.4/> for detailed information.
# In particular, see
# <URL:http://httpd.apache.org/docs/2.4/mod/directives.html>
# for a discussion of each configuration directive.
#
# Do NOT simply read the instructions in here without understanding
# what they do.  They're here only as hints or reminders.  If you are unsure
# consult the online docs. You have been warned.
#
# Configuration and logfile names: If the filenames you specify for many
# of the server's control files begin with "/" (or "drive:/" for Win32), the
# server will use that explicit path.  If the filenames do *not* begin
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# server as '/www/log/access_log', where as '/log/access_log' will be
# interpreted as '/log/access_log'.


#
# ServerRoot: The top of the directory tree under which the server's
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path.  If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used.  If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"
```

Httpd config overview

```
#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80

#
```

```
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User apache
Group apache
```

Httpd config overview

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/var/www/html"

#
# Relax access to content within /var/www.
#
<Directory "/var/www">
    AllowOverride None
    # Allow open access:
    Require all granted
</Directory>

# Further relax access to the default document root:
<Directory "/var/www/html">
    #
    # Possible values for the Options directive are "None", "All",
    # or any combination of:
    #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
    #
    # Note that "MultiViews" must be named *explicitly* --- "Options All"
    # doesn't give it to you.
    #
    # The Options directive is both complicated and important.  Please see
    # http://httpd.apache.org/docs/2.4/mod/core.html#options
    # for more information.
    #
    Options Indexes FollowSymLinks

    #
    # AllowOverride controls what directives may be placed in .htaccess files.
    # It can be "All", "None", or any combination of the keywords:
    #   Options FileInfo AuthConfig Limit
    #
    AllowOverride None

    #
    # Controls who can get stuff from this server.
    #
    Require all granted
</Directory>
```

Lab

Have a look at the apache configuration and deploy this website on your server :

https://github.com/lassiecoder/E-CommerceWebsite.git

(use git to clone the repository directly on your server)

Vhosts

When using the Apache web server, you can use *virtual hosts* (similar to server blocks in Nginx) to encapsulate configuration details and host more than one domain from a single server

httpd on CentOS 7 has one server block enabled by default that is configured to serve documents from the /var/www/html directory.

This works well for a single site but if you want to host multiple site, you will have to use vhosts.

Vhosts

```
<VirtualHost *:80>
    ServerName www.example.com
    ServerAlias example.com
    DocumentRoot /var/www/example.com/html
    ErrorLog /var/www/example.com/log/error.log
    CustomLog /var/www/example.com/log/requests.log combined
</VirtualHost>
```

- ServerName : define the host for this vhost configuration. Only one ServerName per virtualhost
- ServerAlias : Alias for ServerName, you can have multiple ServerAlias
- DocumentRoot : root directory of your website. You will put there your website files

Lab Vhosts

Create 2 vhosts : site1 and site2

We will use xip.io DNS for our 2 websites

```
Geoffrey ~/work/git nslookup site1.3.120.193.155.xip.io
Server:         192.168.0.254
Address:        192.168.0.254#53

Non-authoritative answer:
Name:   site1.3.120.193.155.xip.io
Address: 3.120.193.155
```

Steps :

- Create 2 directories in /var/www for each website and create a simple index.html page in each website Document Root
- Create the configuration for the 2 vhosts in /etc/httpd/conf.d
- /!\ take care to split the logs in different files for the 2 websites
- the 2 websites must respond through 2 differents hostname (with xip.io)

HTML ? i want PHP

Website are not only written with static html and css files.
One common way to create dynamic website is to use php
ex : wordpress use php

There is 2 ways to execute php on your httpd server :

- integrated apache module : Apache process will be in charge to execute php program

- CGI and PHP-FPM : Apache will delegate the execution to an other process.

Lab

- Install php on your server

- replace your site 1 index.html by a file named index.php with the following content :

```
<?php
phpinfo();
?>
```

- Check your site1 with your browser

- Deploy this website into the site2 vhost :
  https://github.com/banago/simple-php-website.git

(A little bit of) Security

Keep your cards close to your chest

```
Geoffrey / telnet  3.120.193.155 80
Trying 3.120.193.155...
Connected to ec2-3-120-193-155.eu-central-1.compute.amazonaws.com.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sun, 16 Feb 2020 18:57:10 GMT
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
X-Powered-By: PHP/5.4.16
Connection: close
Content-Type: text/html; charset=UTF-8

Connection closed by foreign host.
```

Security basics

Apache » Http Server » 2.4.6 : Vulnerability Statistics

Vulnerabilities (**26**) Related Metasploit Modules (Cpe Name:*cpe:/a:apache:http_server:2.4.6*)

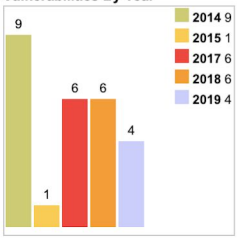
Vulnerability Feeds & Widgets

Vulnerability Trends Over Time

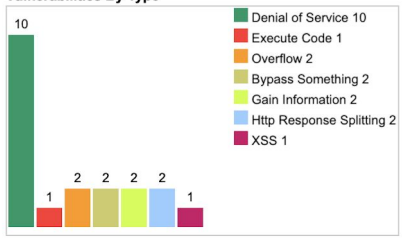
Year	# of Vulnerabilities	DoS	Code Execution	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	Http Response Splitting	Bypass something	Gain Information	Gain Privileges	CSRF	File Inclusion	# of exploits
2014	9	8	1	1						1	1				1
2015	1									1					
2017	6	1		1					1		1				
2018	6	1							1						
2019	4						1								
Total	26	10	1	2			1		2	2	2				1
% Of All		38.5	3.8	7.7	0.0	0.0	3.8	0.0	7.7	7.7	7.7	0.0	0.0	0.0	

Warning : Vulnerabilities with publish dates before 1999 are not included in this table and chart. (Because there are not many of them and they make the page look bad; and they may not be actually published in those years.)

Vulnerabilities By Year



Vulnerabilities By Type



Security basics

- Patch your server frequently

- Hide httpd infos :
  By default, Apache send in the HTTP headers the version of httpd daemon and the name of the operating system

```
ServerTokens Prod
ServerSignature Off
```

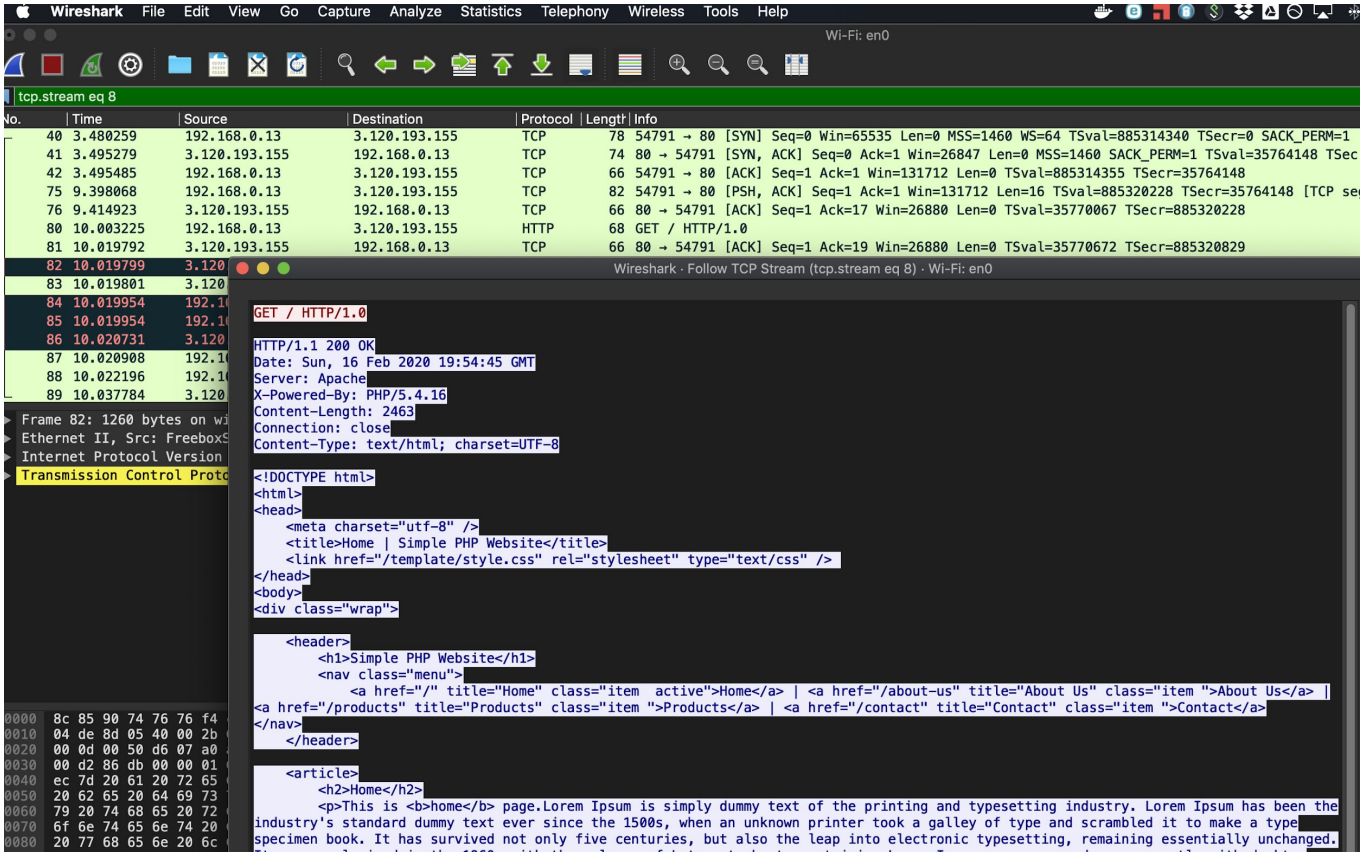
https://httpd.apache.org/docs/2.4/fr/mod/core.html#servertokens

Lab

- Send a request to your web server and analyse the HTTP headers in the response
- Configure your server to hide these informations
- Check that your web server dot not send the information anymore.

Security basics

Security basics



Add some SSL

Why SSL ?

- To secure your connection, to avoid to reveal passwords, session informations, cookies,..

- Server is identified by a certificat : the client can be sure of the real identity of the server.

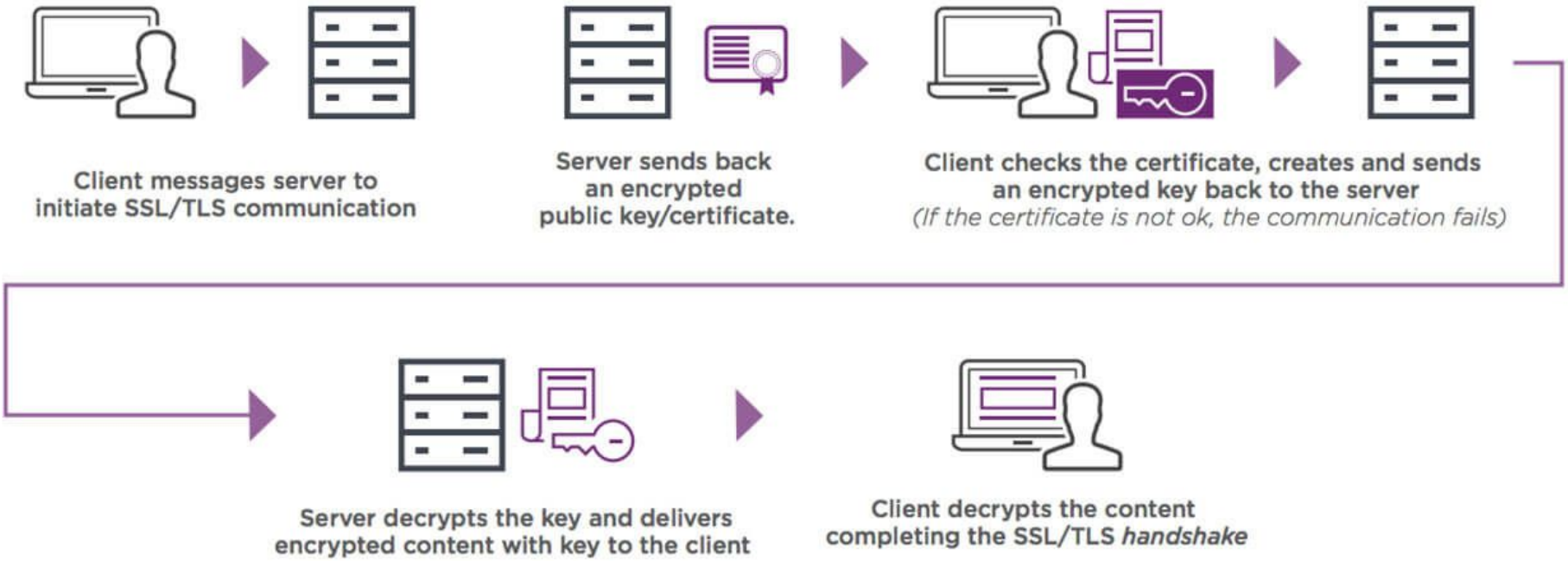
Add some SSL

Why SSL ?

- To secure your connection, to avoid to reveal passwords, session informations, cookies,..

- Server is identified by a certificat : the client can be sure of the real identity of the server.

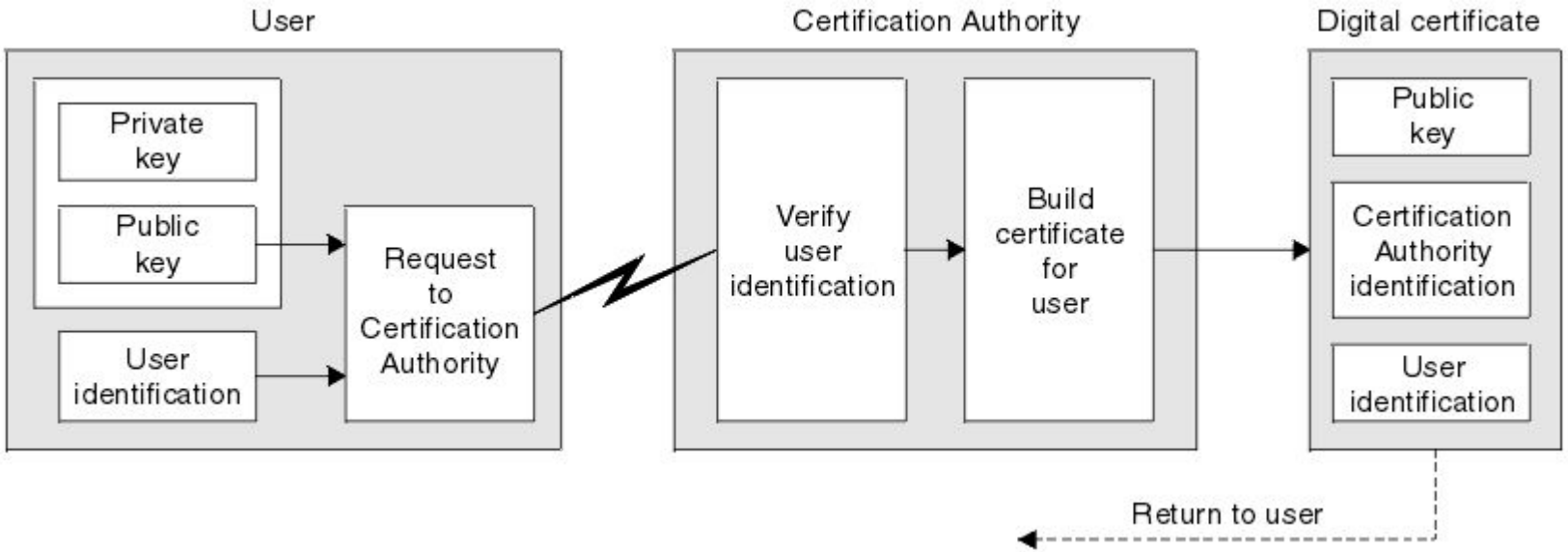
How SSL works ?




The SSL/TLS Handshake Process
Client
Server
Client send a message to a server: "I'd like to set up an encrypted session, please. Here is a list of cipher suites and the SSL/TLS versions I am willing to use."
The server responds: "Hey there! Let's use this particular cipher suite from your list. I also checked and I can use a version of TLS you use, so we're good there. Now here's my certificate, including my public key."
Client verifies the server's certificate, then extracts the public key. Client uses the public key to encrypt a new "pre-master key", then sends it to the server.
The server uses its private key to decrypt the pre-master key.
The client and server now **both** use the pre-master key to compute a shared secret key, called the "shared secret".
Client sends a message encrypted per the mutually-arrived-at specifications: "Hey, here's an encrypted message. Try to decrypt it and verify that it's up to spec. Also, everything I send from this point forward is going to be encrypted using our shared secret. Over!"
The server decrypts and verifies the message. If this works, the server now sends a message back to the client encrypted with their shared secret: "Hey there! Your encrypted message checks out - now I am sending an encrypted message, too. Decrypt it and verify the contents, please. I too am encrypting everything I send to you from here on using our shared secret. Over!"
The client and server now **both** use the shared secret encryption to protect their communication for the rest of this session.

SSL configuration

To configure your website with HTTPS, you need to create a certificate.
Everybody can generate a certificate but to ensure the server identity, Certification Authority can validate a certificate :



LAB

- Generate a certificate for your site1 :

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout site1.key -out site1.crt
```

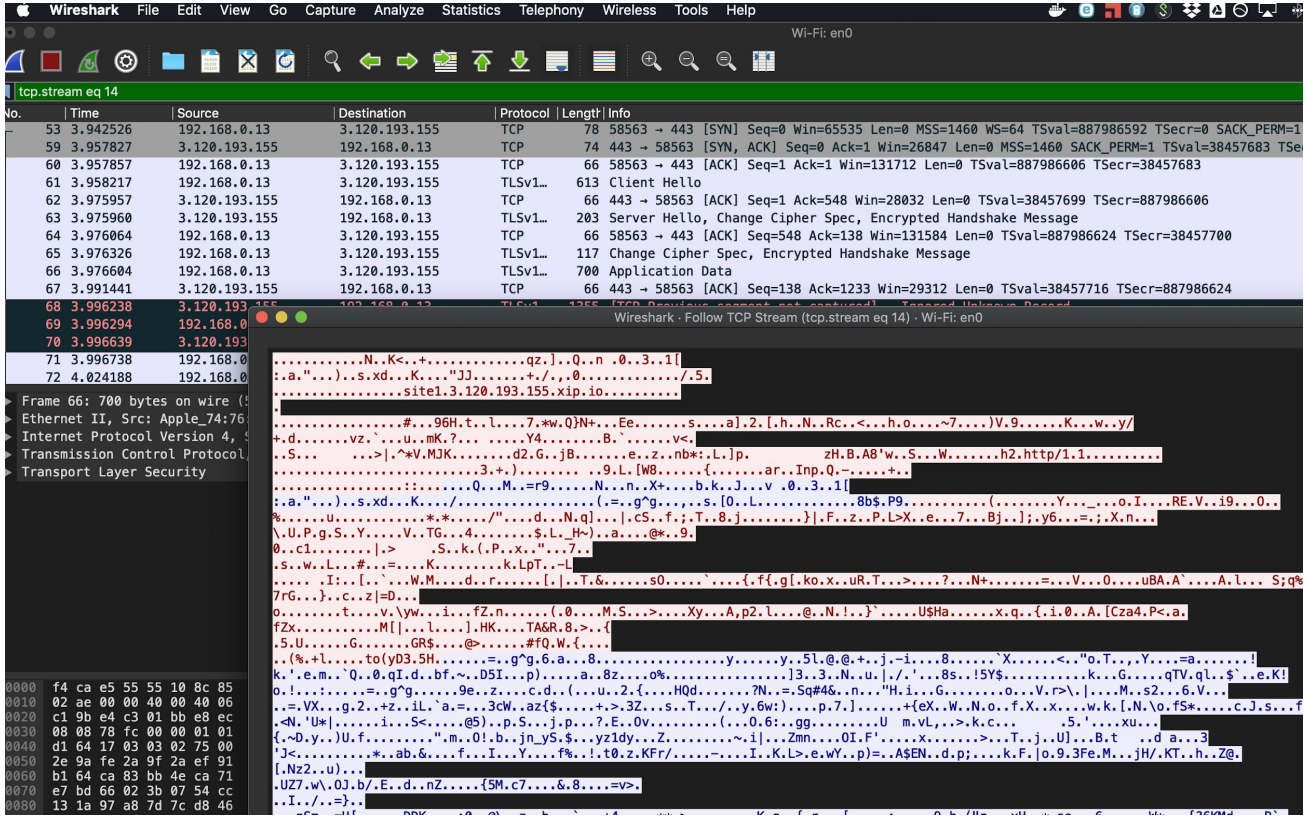
- Add a vhost for site1 and add the following configuration to enable ssl on the vhost

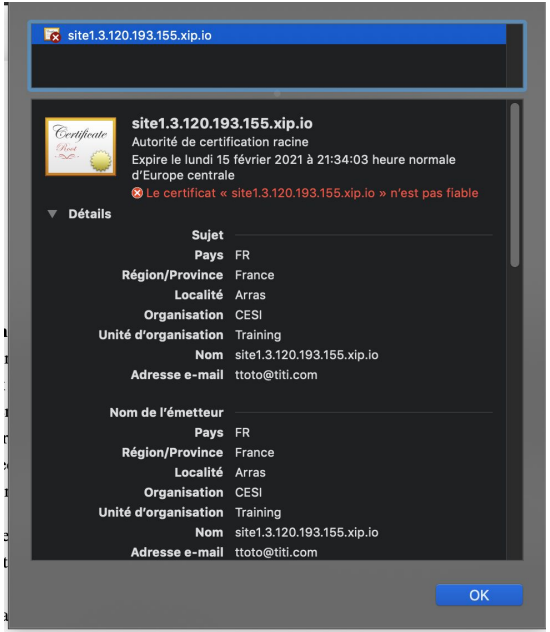
```
SSLEngine on
SSLCertificateFile /etc/httpd/ssl/site1.crt
SSLCertificateKeyFile /etc/httpd/ssl/site1.key
```

- Add the missing ssl module for apache

- Check that that ssl is active on your website

- Bonus : add a rewrite rule to redirect the client automatically from http to https (and check the status code in the http response)


Wireshark File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Wi-Fi: en0
tcp.stream eq 14
No. Time Source Destination Protocol Lengtr Info
53 3.942526 192.168.0.13 3.120.193.155 TCP 78 58563 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=887986592 TSecr=0 SACK_PERM=1
59 3.957827 3.120.193.155 192.168.0.13 TCP 74 443 → 58563 [SYN, ACK] Seq=0 Ack=1 Win=26847 Len=0 MSS=1460 SACK_PERM=1 TSval=38457683 TSe
60 3.957857 192.168.0.13 3.120.193.155 TCP 66 58563 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=887986606 TSecr=38457683
61 3.958217 192.168.0.13 3.120.193.155 TLSv1… 613 Client Hello
62 3.975957 3.120.193.155 192.168.0.13 TCP 66 443 → 58563 [ACK] Seq=1 Ack=548 Win=28032 Len=0 TSval=38457699 TSecr=887986606
63 3.975960 3.120.193.155 192.168.0.13 TLSv1… 203 Server Hello, Change Cipher Spec, Encrypted Handshake Message
64 3.976064 192.168.0.13 3.120.193.155 TCP 66 58563 → 443 [ACK] Seq=548 Ack=138 Win=131584 Len=0 TSval=887986624 TSecr=38457700
65 3.976326 192.168.0.13 3.120.193.155 TLSv1… 117 Change Cipher Spec, Encrypted Handshake Message
66 3.976604 192.168.0.13 3.120.193.155 TLSv1… 700 Application Data
67 3.991441 3.120.193.155 192.168.0.13 TCP 66 443 → 58563 [ACK] Seq=138 Ack=1233 Win=29312 Len=0 TSval=38457716 TSecr=887986624
68 3.996238 3.120.193.155
69 3.996294 192.168.0
70 3.996639 3.120.193
71 3.996738 192.168.0
72 4.024188 192.168.0
Frame 66: 700 bytes on wire (5
Ethernet II, Src: Apple_74:76
Internet Protocol Version 4, S
Transmission Control Protocol
Transport Layer Security
Wireshark · Follow TCP Stream (tcp.stream eq 14) · Wi-Fi: en0
site1.3.120.193.155.xip.io
0000 f4 ca e5 55 55 10 8c 85
0010 02 ae 00 00 40 00 40 06
0020 c1 9b e4 c3 01 bb e8 ec
0030 08 08 78 fc 00 00 01 01
0040 d1 64 17 03 03 02 75 00
0050 2e 9a fe 2a 9f 2a ef 91
0060 b1 64 ca 83 bb 4e ca 71
0070 e7 bd 66 02 3b 07 54 cc
0080 13 1a 97 a8 7d 7c d8 46


site1.3.120.193.155.xip.io
site1.3.120.193.155.xip.io
Autorité de certification racine
Expire le lundi 15 février 2021 à 21:34:03 heure normale d'Europe centrale
Le certificat « site1.3.120.193.155.xip.io » n'est pas fiable
Détails
Sujet
Pays FR
Région/Province France
Localité Arras
Organisation CESI
Unité d'organisation Training
Nom site1.3.120.193.155.xip.io
Adresse e-mail ttoto@titi.com
Nom de l'émetteur
Pays FR
Région/Province France
Localité Arras
Organisation CESI
Unité d'organisation Training
Nom site1.3.120.193.155.xip.io
Adresse e-mail ttoto@titi.com
OK

Scale your service

Scaling

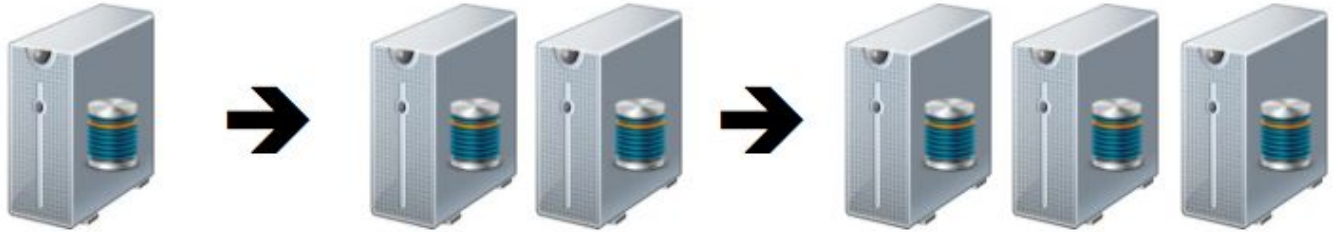
Your website is now popular and you have to face a lot of incoming connections

2 options : scale up or scale out (vertical vs horizontal)

Scaling

Scale-Up

Scale-Out



How do i choose ?

Lab

- Create an image (AMI) from your current web server
- Deploy a new instance from the image (do not forget the security group)
- Check that your website is available on your new server
- Create a loadbalancer (http/https) with the 2 servers as backend
- Check that your incoming requests are well balanced between the 2 servers

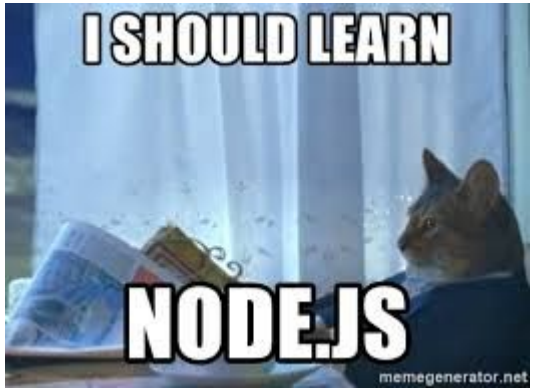
Javascript runtime example

Other hosting example with Node.JS

Node.js is a framework for writing server-side JavaScript application.

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser.

Node is often used to build back end services that communicate with client-side applications



Other hosting example with Node.JS

LAB :

Install node and npm on your server (check the instruction on cryptpad)

Start your application demoserver.js

Check your application with your browser