# Unidad 1. Final Summary. Concepts

**Web application**: When a user needs to execute an application over internet make use of web application. It has at least two parts, one in server side, where the information that the user needs are, and other is executed in the client machine. Some examples of web apps are e-mail, social networks or online shops.

**Web architecture**: is the design of a web application’s structure, including study, analysis, organization and structure of the information of web pages. The most of current web architectures are based on the client server model.

**Features of web architectures**: Scalability, Portability, Design Patterns, Responsibility Separation, User’s session management.

**Scalability**: An application is scalable when we can increase the data number and the data size. It’s important to know that efficiency and scalability is not the same. It can be Vertical, Horizontal or a server cluster.

* **Vertical Scalability**: It consists in improving the hardware to increase the performance. The advantage is that the application does not need any change, you only have to get a better machine.
* **Horizontal Scalability**: This strategy implies to add more machines, increasing in this way the capacity. The most used is shared nothing, where every machine (or node) is independent and self-sufficient. This means that the app is shared in all the system. Google use this strategy.
* **Server cluster**: a cluster consists of two o more computers working together to provide a higher level of availability and scalability. So, a server cluster is a group of independent servers working together as a single system.

**Portability**: it refers to the capacity of a web application to execute it in different platforms or architectures with the minimum modifications.

**Design patterns**: They are reusable solutions to a commonly problem in the web application development. They allow the communication between developer and analyst. The **MVC** (Model View-Controller) is a software architectural pattern for implementing user interfaces on computer.

**Responsibility separation**: It means that different responsibilities must go to different layers. The most popular used is the 3-independent layers: Presentation(Client), Business(Web server) and Persistence(Data).

User’s session management: A user session is the activity time that user spends in a web site. The number of sessions are used to calculate the traffic of data.

**Web architecture models**: There are several web architectures models, for example the W3C (World Wide Web Consortium). We focus on Client/Server Model.

**Client/Server model**: This model works like a distributed application which distinguish providers from petitioners.

**1**. From the client web browser, the user requests a web page writing the URL (Uniform Resource Locator). For example, http://www.ceedcv.org

**2**. The server receives the request.

**3**. The server looks for the web page in its management system.

**4**. The server sends the source code of the web page to the client.

**5**. The client receives the source code and the web browser shows the web page.

Nowadays this model is divided into three layers: Presentation, Business and Persistence.

Some **advantages** of this model are:

-Centralization of the information: the access, resources and data are controlled by the server avoiding that an unauthorized client can enter to the system and damage it.

-Scalability: you can increase the number of clients or servers separately.

-Easy maintenance: all the functions and responsibilities are distributed between some independent machines so is easy to replace, repair or update any server.

-There are many technologies based on this model.

And some **disadvantages** are:

-Management of the traffic congestion: if many clients send petitions to the same server, it can be down. -Resource access: if a server falls down, its resources are not available.

-Dependence of hardware and software: normally we need specific software and hardware for each case.

An example of this model is a LAMP server.

**Web server**: is a program or a set of them, which offer services over a network. Its aim is to offer necessary resources in order that two or more programs could communicate independently. **The load balancing** is implemented by an algorithm that distributes the tasks in the most efficient and equitable possible way, avoiding the bottlenecks.

**How to have a web server**: we will need:

- A stable operating system, with security and permissions control.

- A static IP address (an IP address which never changes), to allow the remote access.

- A Domain Name Server (DNS) to accede easier to our server.

- 24 Internet Connection to be accessible from another networks.

- Specific software of web server.

**Webmin:** is a program that simplifies the process of managing a Linux or Unix system. Normally you need to manually edit configuration files and run commands to create accounts, set up a web server or manage email forwarding. Webmin lets you perform these tasks through an easy to use web interface, and automatically updates all of the required configuration files for you. This makes the job of administering your system much easier.

Some of the things that you can do with Webmin are:

• Create, edit and delete Unix accounts on your system.

• Export files and directories to other systems with the NFS protocol.

• Set up Disk Quotas to control how much space users can use up with their files.

• Install, view and remove Software Packages in RPM and other formats.

• Change your system's IP address, DNS Server settings and routing configuration.

• Set up a Linux Firewall to protect your computer, or to give hosts on an internal LAN access to the Internet.

• Create and configure virtual web servers for the Apache Webserver.

• Manage databases, tables and fields in a MySQL or PostgreSQL Database Server

• Share files with Windows systems by configuring Samba Windows File Sharing.

# Unidad 1. Final Summary. Practice.

**Virtual machine**: Its an application of virtualization.

**Loopback address**: is a special IP number (127.0.0.1) that is designated for the software loopback interface of a machine. The loopback interface has no hardware associated with it, and it is not physically connected to a network.

**Ping**: ping is a computer network administration software utility used to test the reachability of a host on an Internet Protocol (IP) network. It measures the roundtrip time for messages sent from the originating host to a destination computer that are echoed back to the source.

**\* How to install a virtual machine**: First of all, we will download and install the program to create the virtual machine (in this case Virtual Box) and once installed, we install the OS.

During installation we will need to:

Give a name to the VM, select amount of RAM, create a Virtual hard disk, and choose the hard disk location and size, and select the ISO file with the OS.

**\* Configure virtual machine Network**:

go to the menu Settings > Network > Adapter 1. We are going to indicate it that it is connected to a bridge adapter. Also, we are going to change the MAC address of any virtual machine to avoid have MAC duplicate for that we will click on the refresh button.

**\*\* Configuring the virtual machines**:

We have to assign an IP address that belongs to our home network. For instance, if my home network IP address is 192.128.0.0 and the IP of my router is 192.168.0.1, I will change it in each virtual machine, I could use 192.168.0.X.

**\* Linux Configuration Commands**:

Ifconfig -a = Check the name of the network device used by the system. It will show our network interface (enp0s3) and the loopback address (lo). We can see too: IP (IPv4, IPv6), Netmask and more info.

sudo =

gedit / nano =

reboot = Reboot the system.

cd = Change directory

cat =

ping IP = Check that the virtual machines can communicate among them.

**\* Configurate Static IP Address and DNS**: Open the network configuration file (with sudo) */etc/network/interfaces* and modify it. In this file we have to write the name of the network interface, if it has a static IP address, what that address is, netmask, the IP address of the router (gateway) and the DNS.

In ubuntu 18 or higher: Search for “Networks” in “Settings of Ubuntu menu and change IP manually.

Then you have to apply changes restarting network with: sudo service networking restart or sudo /etc/init.d/networking restart

We can check it with ifconfig -a.

**\* Change Hostname or Name of machine**: We have to edit the configuration file */etc/hostname* and change the name.

**\* Associate the Hostname to the IP address 127.0.1.1 (loopback):** edit the configuration file */etc/hosts*

**\* Check that the hostname is changed and if the DNS address is correct**: in the configuration file */etc/resolv.conf*

**Configuring Windows Server**:

**Check the configuration of virtual machines with ping**: we write ping and the IP address machine which we want to check. To stop a ping, you have to type Ctrl+C. If we see 0% packet loss, it works.

**Install Webmin**: We have to run command wget with URL download link or directly from the official webpage. And then sudo dpkg –install and the file install name. Maybe we have to install the needed dependencies. **To run Webmin** we have to go to de url *https://localhost:10000* (we have to accept the certificate). The username and the password are the same as in the machine.

**If we want to connect it from other virtual machine:** we will have to write the URL: *https://IP:10000*

**Parts of Webmin:** The main window gives us the system information about our machine: hostname, OS, version, processor, etc.

Webmin has more than 100 modules and all of them can be found in the left menu grouped as: • Webmin • System • Servers • Others • Networking • Hardware • Cluster • Un-used Modules: contains all the modules not installed.

It has 6 modules: • Backup Configuration Files • Change Language and Theme • Webmin Actions Log • Webmin Configuration • Webmin Servers Index • Webmin Users

# Unidad 2. Final Summary. Theory.

DNS / FTP / SSH

**DNS Tools**:

• dig: it resolves names and IP addresses very detailed. It offers information about the records and the wait time of the DNSS.

• nslookup: it resolves names and IP addresses not detailed.

• ping: it allows to test if DNS is working

• whois: it shows the information of the domain’s owner.

• Traceroute/tracert: it shows how many jumps are needed to reach a domain.

**Forward Lookup zone:**

**Reverse lookup zone**: There is the inverse process, Reverse lookup, which uses a reverse DNS. It allows to obtain the domain names associated to an IP address. The requests follow a similar process than the one explained before, but they use a specific domain (ip\_addr.arpa for IPv4 and ip6.arpa for IPv6). The IP addresses are represented in an inverted way.

**The File Transfer Protocol (FTP)**: is designed to facilitate bi-directional transfer of files and records between hosts on a TCP/IP network. **Active mode** is the original method used by the FTP protocol for transferring data to the client application. **Passive mode**, like active mode, is initiated by the FTP client application. When requesting data from the server, the FTP client indicates it wants to access the data in passive mode and the server provides the IP address and a random.

**SSH** (or Secure SHell) is a protocol which facilitates secure communications between two systems using a client/server architecture and allows users to log into server host systems remotely. SSH encrypts the login session, making it impossible for intruders to collect unencrypted passwords.

# Unidad 2. Final Summary. Practice.

**Installing DNS Server**: First of all, we have to install the Bind packet. For that, we have to go to the group “Un-used Modules” and select Bind DNS Server. Install the modules needed and refresh modules.

Connect virtual machine to new DNS server: To do so, go to the Network Settings (Configuration → Red) and in the IPv4 tab, set the DNS server to the same IP as the server. Make sure to disable the Automatic DNS option. And then restart the network with the button.

Then go back to Webmin BIND DNS Server and make sure the DNS server is on.

**Configuring Lookup zone forward:** We will start with the forward lookup zone. and will be a master zone. For that we have to click on Create master zone button in the Existing DNS Zones section.

we will write the domain name, the email address, check that the master server is our server and click on the Create button.

Once the master zone is created, we can edit it to create **A (Address)** and **NS (Name Server)** records.

First, we are going to create the Address Records of every virtual machine. To do so we have to click Address and type the machine name and IP address. Also, we can check Yes in *Update reverse?* to create the reverse address in the reverse zone. Finally **Apply configuration**.

Now we can **check the configuration** using dig IP

**Configuring Lookup zone reverse:** It will be a master zone again. In this case we have to choose the option Reverse and write the network. Once the master zone is created, we are going to create new PTR records (Reverse address button).

If this zone had been created before we created the A records (checking Yes in Update reverse?) we will have the 3 reverse addresses already created when we created those address records. Now we have to create them, we have to click on Reverse Address. We are going to create one for each virtual machine. Then, **Apply configuration**.

Now, we can **check the configuration** using nslookup < IP-address >

**FTP Installation:** First of all, we need to install the ProFTPD module. To do so go to the Un-used Modules group and once installed refresh modules.

**Testing FTP:** For testing we need a user (all users are operating system’s users) or we can create one in group *System* -> *Users and groups.*

Now we need to start Client machine and install FTP Client software, for example **FileZilla**. Open **FileZilla** and connect with address, username and password, then click connect.

The left side shows the files and folders in the Client machine. The right side shows the files and folders in the Server machine.

This **FTP connection** allows you to transfer files from one machine to the other one very easily (drag and drop), as well as creating, renaming and deleting files and folders.

**Allow Anonymous Connections**:That means, being able to connect to the FTP Server without hacing to use a user and password (anyone can connect). For safety reasonsthis type of connections are limited by default.

To allow Anonymous Connections we have to go back to Webmin => *Servers* => *ProFTPD Server* and click on the ***Default Virtual Server*** (the world globe). Now click on the Anonymous FTP icon.

Here, we have to set the directory that anonymous users will be able to access. The access to files will be the same as the ftp user.

Now we have to click on the authentication icon.

Here we only have to change the radiobutton of Only allow login by users with valid shell? to No to allow the access to the anonymous users to the server. Then we click on Save.

**Limit user actions:** we can do it per-directory options or per-command options. In per directory we write \* (To indicate all directories) and WRITE in FTP commands.

Then, click on Access Control icon to create an access rule for WRITE in directory \* (everywhere). Under Action select Deny and All to indicate that any write action will be denied. Now click Save. Finally, we have to apply the changes.

Now we can check it connecting from Client machine, in FileZilla, with anonymous as username.

**SSH Installation**: First of all, we have to install the SSH package. To do so, we have to go to the group Un-used Modules and select SSH Server. It will probably not allow you to install it the same way as the DNS and FTP servers. For this case open terminal in Server machine and write sudo apt-get install openssh-server, and if it fails, update repositories with sudo apt-get update.

And refresh modules.

**Testing SSH**: We could use the default SSH-client but let’s try installing Putty. Write in client terminal sudo apt-get install putty

Once installed open it and try connecting to Server in port 22 clicking Open.

the server will send its key fingerprint.

To check if that fingerprint is the same as our server we can find the content of the file with the key in *Host SSH Keys* in Webmin.

and we choose the file */etc/ssh/ssh\_host\_rsa\_key.pub* (see that in the alert before the key fingerprint is rsa2)

Now, to see the key fingerprint we have to write in the terminal of the Server machine: ssh-keygen -l -E MD5 -f /etc/ssh/ssh\_host\_rsa\_key.pub

We can check that it is the same, so we click on Accept in the Putty, so the client stores the fingerprint and it will not show the alert again.

Finally, we access with the user and password.

Now you are connected to the Server machine via a shell (terminal) and you could run Linux commands as if you were there.

# Unit 3. Final Summary. Theory.

**Hypertext Transfer Protocol (HTTP)** is an application protocol for transmitting hypermedia documents, such as HTML. It was designed for communication between web browsers and web servers. It follows a classical client-server model, with a client opening a connection (port 80). The main disadvantage is that all the information sent in that connection is not encrypted.

The **HTTPS protocol** (Hypertext Transfer Protocol Secure) is the secure version of the HTTP protocol. It is not a protocol in itself, but the union of two: HTTP protocol plus a layer with the SSL or TLS protocols. It uses the port 443.

**SSL (Secure Socket Layer)** is the secure protocol that is added to HTTP. Nowadays version 3.0 is used, but there’s also a derived protocol: the **TLS** (Transport Layer Security).

**Digital certificate**: is an electronic "passport" that allows a person, computer or organization to exchange information securely over the Internet using the public key infrastructure (PKI). A digital certificate may also be referred to as a public key certificate.

**MIME Type**: The Internet standard MIME (Multipurpose Internet Mail Extensions) extends the format of email to support:

• Text in character sets other than ASCII.

• Non-text attachments: audio, video, images, application programs, etc.

• Message bodies with multiple parts (multi-part).

• Header information in non-ASCII character sets.

When a browser tries to open a file, the MIME standard helps it knowing with what type of file it is working and the program that can open it correctly. If the file does not have a MIME type specified, the program supposes that the type is the file extension. For instance, in the case of a file with a .txt extension, the program supposes it is a text file.

**Web server administration modules:** All web servers should give us the possibility to get new functionalities (or remove them) in an easy way and according to our needs. Some module examples are: SSL, PHP, LDAP, etc.

**Virtual hosts/servers blocks:** The term Virtual host is use d in Apache and Server block is used in Nginx. Both refer to the practice of running more than one web site on a single machine.

Virtual hosts and Server blocks can be "IP-based", meaning that you have a different IP address for every web site, or "name-based", meaning that you have multiple names running on each IP address.

**Authentication and authorization:** it is important to know that authentication is any process by which you verify that someone is who they claim to be, and authorization is any process by which someone is allowed to be where they want to go, or to have information that they want to have. To do so there are some methods such as **Basic** or **Digest**. In both of them the user has an id and a password to access resources. The main differences are that Basic does not encrypt the password (Digest does) and Digest works with domains (Basic doesn’t).

**Logs:** The registry files or logs allow to see the performance of the web server as well as to perform statistics or analysis. There are logs that store the errors that happen in the server or who has accessed to any resource.

# Unit 3. Final Summary. Practice.

**Installing Apache**: sudo apt-get install apache2. If the installation fails, we will update the Ubuntu repositories: sudo apt-get update and then sudo apt-get install apache2 again.

If you have problems with dpkg because it is blocked: sudo rm /var/lib/dpkg/lock

**To see apache users**: cat /etc/passwd

**The root directory of the default virtual server**: */var/www/html.*

Once installed, we can **check if the server is running** and listening in port 80/TCP. We have to write ps -ef | grep apache and then netstat -ltn

If we write: Firefox index.html we can see the default web page. */var/www/html/index.html.*

**To check the Apache configuration**: Start Client, open firefox and writhe the IP address of Server.

**List the content of apache**: ls /etc/apache2

**Checking the file** */etc/apache2/sites-available/000-default.conf* we can see:

• Inside the <VirtualHost></VirtualHost> directive, it is defined the performance of the virtual server by the default.

• The value of the **DocumentRoot** directive is /var/www/html

• The value of the **ErrorLog** directive.

**To create a new file**, we can use sudo gedit [name]

**To create a new directory,** we need to use sudo mkdir [name]

**To check de new files**, we write in Browser of client IP/page

If the server does not find the file index.html in the root directory */var/www/html* it sends the content of the directory.

To change the default file: we open the file */etc/apache2/sites-available/000-default.conf* and include the <*Directory*> directive and its content. Writing **DirectoryIndex** homepage.

**Directives**

- ServerName: indicate the name of the server

- ServerAdmin: indicate the server administrator’s email

- DocumentRoot: where are the web pages in the server - Directory: Options applied for the directory. In this case /var/www/html

+ DirectoryIndex: what is the web page by default

+ Options:

- Indexes: List the content of the directory if the DirectoryIndex does not exist

- FollowSymLinks: Allow follow symbolic links

- Multiviews: Allow the use of different languages in the answer

+ AllowOverride: indicate what access control is used

+ Order: indicate the order to allow or deny the access. (Do not write a space between them, only a coma ',') - allow from all: any machine is allowed to access to this site.

**Restart apache**: sudo /etc/init.d/apache2 restart

. When the option **Indexes** is enabled, the server lists the content of the directory but if it is disabled the server sends a forbidden message.

**Check the static modules**: sudo apache2ctl -l also we can see the list of available modules in */etc/apache2/mods-available*

**\* To configure the access control to some web pages of our server:** we have to create a new directory */var/www/html/teacher* and a file called *teacher.html*

Now, we are going to edit the file */etc/apache2/sites-available/000-default.conf* and insert a new **Directory** directive where we only allow the access from the IP we want to.

With allow from IP

\* **Change error page with message or error page**: we have to open the default configuration file of apache and edit directive ErrorDocument or we can indicate the errorpage directory.

**\* HTTP authentification**: we have to check if auth\_basic module is enabled and create user and password using sudo htpasswd -c /etc/apache2/passwd user. Then we create the directory directive with authtype, authname, authuserfile and require user

Take a look to registry files: we can see them if we write cat /var/log/apache2/error.log and cat /var/log/apache2/access.log.