Code & Go

Web server administration

Day 1

Teaching managers
pedagowac@epitech.eu

Table of contents

Administrative details

- This project is to be done alone.
- There is **nothing to submit in your repository for today**.

Introduction

Today, you will discover how to administer a web server to run your various projects.

Tip: since there is nothing to submit, you are strongly advised to spend this day along with your assistants so that you can ask questions at the right time.

Definitions and explanations

We will now focus on the various terms you may encounter while you create and manage your web server.

The clients/servers protocol

The client-server environment defines a communication mode between several programs on the network. The client program sends requests and the server waits for these requests and then proceeds to answer them.

The server

- Waits for an incoming connection on one or several ports.
- When a client connects on the listening port, a communication socket is opened.
- The server and the client use this socket to communicate.

The client

- Establishes a connection with the server on the server's listening port.
- Once a connection is established, it communicates via the created socket and respects the communication protocol defined by the server.

The main types of client/server configuration

- Peer-to-peer
- 2-tier architecture
- 3-tier architecture
- N-tier architecture

The WEB server

Web servers are computers with powerful configuration (most of the time ☺), sitting in a secure datacenter. The HTTP server software sitting on those machines is also called a web server.

When you enter any **Uniform Resource Locator (URL)** here is what happens internally:

1. Your browser will first send a name request to a name server, which translate that domain name to the server's IP address.
2. Then the browser sends an HTTP/GET request to port 80 of that IP address.
3. Web server receive the GET request and **serves** the file (usually an HTML) to the browser and the browser renders it to your display. This happens every time you send a request

This whole operation is performed by the HTTP and TCP/IP protocol. TCP/IP does the network-level task, while HTTP mainly deals with serving request/response.

The main HTTP servers are:

- Apache
- IIS (Internet Information Services)
- Lighthttpd
- Nginx
- NodeJS

Communication between two machines

IP address and port

An IP address (**IP** stands **for Internet Protocol**) is an identification number given permanently or temporarily to each device connected to a computer network using the Internet Protocol. The IP address is at the core of the transit system (routing) of messages on the net.

There are two versions:
- IPv4 addresses (on 32 bits, i.e. 4 bytes)
- IPv6 addresses (on 128 bits, i.e. 16 bytes).

Version 4 is the most used today: it is generally displayed in decimal form with four numbers between 0 and 255, separated by dots, for example: 212.85.150.134. Today, the ranges of IPv4 addresses are close to saturation, and operators are now encouraging the transition from IPv4 to IPv61.

The notion of software porting, which corresponds to the transport layer of the OSI model, gives the possibility, on a given computer, to identify various interlocutors. These interlocutors are computer programs that can listen or send information on these ports. A port is designated by its number.

Application

Obtain the information about your network interfaces

```
$> ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:02:35:48
          inet addr:172.16.146.129  Bcast:172.16.146.255
Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe02:3548/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:282851 errors:0 dropped:0 overruns:0 frame:0
          TX packets:142028 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:401298164 (382.7 MiB)  TX bytes:11182098 (10.6 MiB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:339 errors:0 dropped:0 overruns:0 frame:0
          TX packets:339 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:58292 (56.9 KiB)  TX bytes:58292 (56.9 KiB)
```

The highlighted area is your current IP address.

Create a server that listens on port 4242 of your machine

```
$> nc -l -p 4242
```

Create a client which connects to the server created above.

```
$> nc myIp 4242
```

The main communication protocols of the WEB

HTTP

HyperText Transfer Protocol, aka **HTTP**, is a client-server communication protocol designed for the World Wide Web. **HTTPS** (with an S for secured) is a variation of HTTP secured by the use of SSL or TLS protocols.
HTTP is a protocol of the application layer. It can function on any reliable connection, and, in practice, we use the TCP protocol as a transport layer. By default, a HTTP server uses port 80 (443 for HTTPS).

The most renowned HTTP are Web browsers which grant a user access to a server containing data. There are also systems that automatically retrieve the content of a site such as website copiers or web crawlers.

HTTPS

HyperText Transfer Protocol Secure, commonly abbreviated as **HTTPS** is the combination of HTTP and an encryption layer such as SSL or TLS.

HTTPS lets the visitor verify the identity of the web site thanks to an authentication certificate sent by a third party with a reliable reputation (which also belongs to the white list of internet browsers). In theory, it guarantees the confidentiality and integrity of the data sent by the user (such as the data entered in forms) and received from the server. It can be used to verify the identity of the visitor, if the latter uses a client authentication certificate.

HTTPS is commonly used in online financial transactions: ecommerce, online banking, online brokerage, etc. It is also used for consulting private data, such as emails, for example.
Since 2010, HTTPS has become common on social networks.
By default, HTTPS servers are connected to the port TCP 443.

HTTP operations and their use

HTTP provides various request types such as **GET**, **POST**, **PUT**, **DELETE** and **HEAD**. Each request type is designed to provide a certain kind of operation. You may have heard of the **create**, **read**, **update** and **delete** (**CRUD**) operations for various sites and programming blogs. Let's map theses CRUD operations to our HTTP requests types.

Create :
This operation generates some new data at the server's end. As the POST HTTP operation is meant to be used when we are generating data, whenever you want to create something (e.g. an user or a post), you should use a POST.

Read :
This operation is directly mapped to the GET operation of HTTP. You should always use the GET request type to read anything from the server.

Update :
To update any existing data on the server, you should use the PUT http operation. PUT is quite similar to the POST operation; however its internal working is different.

Delete :
This operation is directly mapped to the DELETE HTTP operation.

FTP

File Transfer Protocol, or **FTP**, is a communication protocol designed to exchange computer files on a TCP/IP network. From a computer, FTP lets you copy files to another computer in the network, or delete or modify files on this computer. This copy operation is commonly used to feed a web site hosted by a third party.

The FTP variation protected by SSL or TLS protocols (SSL being the predecessor of TLS) is called FTPS. FTP follows a client-server model. This means that one of the two parties, the client, sends requests to which the other – the server – reacts. In practice, the server is a computer on which is executed a software called FTP server. The latter makes public a file tree similar to a UNIX file system. To access a FTP server, we use a client FTP software (via a graphic interface or a command line).

By convention, two ports are reserved (well known ports) for FTP connections: port 21 for commands and port 20 for data. For the implicit FTPS, the conventional port is 990.
This protocol can function with IPv4 and IPv6.

Reminder on APT-GET

Advanced Packaging Tool - APT – is a package manager used by Debian GNU/Linux and its derivatives.

APT simplifies installing, updating and uninstalling software by automating retrieval of packages from APT sources (Internet, local network, CD-ROM, etc.). It handles managing dependencies and sometimes compiling. When packages are installed, updated or uninstalled, the package management program can display packages dependencies. As the administrator, it can ask if recommended or suggested dependencies for newly installed packages are also to be installed, and automatically resolve these dependencies.

Useful commands

```
$> apt-get install <package> <package>
$> apt-cache search <words>
$> apt-get remove <package> <package>
$> apt-get update
$> apt-get upgrade
```

SSH

Definition

Secure Shell (SSH) is both a computer program and a secured communication protocol.

The connection protocol imposes an exchange of encryption keys at the start of the connection. Then, all TCP segments are authenticated and encrypted. It becomes impossible to use a sniffer to see what the user is doing.

Installation

```
$> apt-get install ssh openssh-server
```

Connection test

```
$>ssh your_login@localhost

The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is 11:7f:e6:31:96:46:1f:26:a2:14:bf:00:db:d4:20:71.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known
hosts.
your_login@localhost's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Thu Oct  8 13:56:54 2015 from localhost
```

Generation of SSH keys (RSA 4096)

```
$> ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/your_login/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/your_login/.ssh/id_rsa.
Your public key has been saved in /home/your_login/.ssh/id_rsa.pub.
The key fingerprint is:
98:5c:bb:c8:ca:da:9c:c2:ab:20:57:78:6d:ca:2c:65 root@debian8-vm
The key's randomart image is:
+---[RSA 4096]----+
|                 |
|                 |
|        .        |
|   . o + .       |
|  . E * S        |
|   B + . .       |
|o.o + o .        |
|o.o= o           |
|..o+*            |
+-----------------+
```

Tests

Create a user named jerry

```
$> adduser jerry
Adding user `jerry' ...
Adding new group `jerry' (1001) ...
Adding new user `jerry' (1001) with group `jerry' ...
Creating home directory `/home/jerry' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for jerry
Enter the new value, or press ENTER for the default
     Full Name []: Jerry Doe
     Room Number []:
     Work Phone []:
     Home Phone []:
     Other []:
Is the information correct? [Y/n] Y
```

Add the public key of your neighbor to the authorized keys of user jerry

Use SSH to connect with jerry's account on the computer of your neighbor by using your private key.

SCP

Definition

Secure copy (**SCP**) designates a secured file transfer between two computers using the SSH communication protocol. The term SCP designates both the scp program and the scp protocol.

The SCP protocol is based on the BSD RCP protocol. It differs from the latter because the data is encrypted during transfer to avoid data extraction coming from sniffed data packages. While the protocol does not implement the authentication and the necessary security for the transfer, it relies on the underlying protocol SSH for these processes.

SCP can ask the user any password necessary for a remote connection, while RCP fails in such situation.
The SCP protocol only implements file transfer. A connection is established by using SSH, then an SCP server is launched. The server program is the same as the client program.

During an upload, the client sends the files to upload to the server eventually including their basic attributes (permissions, timestamp). This an advantage over the FTP protocol, which cannot include the attributes during the upload.
During a download, the client sends a request to obtain the files and folder to download. The server then feeds the client by including sub-folders. Since the download is steered by the server, there is security risk once connected to a malevolent server.

In most applications, the SCP protocol is superseded by the FTP protocol which is also based on SSH.

Test

Try this out to create a file named testSCP and containing “Hello World!”

```
$> echo 'Hello World !' > testSCP
```

Try this out to send the file testSCP by using jerry’s account in the /tmp folder

```
$> scp testSCP jerry@localhost:/tmp
```

cURL

Client URL Request Library is a command line tool for transferring data through different protocol.

Installation

Use this command to Install cURL

```
$> apt-get install curl
```

Test your installation

```
$> curl -v google.com
* Rebuilt URL to: google.com/
* Hostname was NOT found in DNS cache
*   Trying 173.194.45.41...
* Connected to google.com (173.194.45.41) port 80 (#0)
> GET / HTTP/1.1
> User-Agent: curl/7.38.0
> Host: google.com
> Accept: */*
>
< HTTP/1.1 302 Found
< Cache-Control: private
< Content-Type: text/html; charset=UTF-8
< Location: http://www.google.fr/?gfe_rd=cr&ei=22gWVuyzBsHI8gf5x6igCA
< Content-Length: 258
< Date: Thu, 08 Oct 2015 13:00:11 GMT
* Server GFE/2.0 is not blacklisted
< Server: GFE/2.0
<
<HTML><HEAD><meta http-equiv="content-type"
content="text/html;charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A
HREF="http://www.google.fr/?gfe_rd=cr&amp;ei=22gWVuyzBsHI8gf5x6igCA">here
</A>.
</BODY></HTML>
* Connection #0 to host google.com left intact
```

Try a cURL Request on google.fr
Try a cURL Request on epitech.eu

Apache

Why Apache?

Apache is the most common HTTP server. It is designed to accept many modules which give it additional functionalities (Perl, PHP, Python, Ruby, CGI, SSI, URL rewriting, content negotiation, etc.).

Apache installation

```
$> apt-get install apache2 apache2-doc
```

Configuration
All Apache configuration files are store in Apache **/etc/apache2/** :
apache2.conf : General configuration (formerly httpd.conf)
mods-available/ : Available modules
mods-enabled/ : Enabled modules
sites-available/ : Available sites (contains default for the site hosted by default)
sites-enabled/ : Enabled sites
In order to enable the modules, use **a2enmod** (Apache 2 Enable Module) followed by the name of the module. We enable here mod_rewrite for rewriting URLs.

```
$> a2enmod rewrite
```

In order to activate a site with its configuration stored in *sites-available*, use **a2ensite** (Apache 2 Enable Site):

```
$> a2ensite <siteName>
```

To disable a site, use a2dissite. To disable a module, use a2dismod.

Note: These actions on modules and sites create a symbolic link in the folder enabled towards the file contained in available, for example:

```
$> ln -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/
```

and delete it to disable it.

Test

Display the Web page of the neighbor's Apache server.

PHP

PHP is a server-side programming language.

Installation

```
$> apt-get install php5-common libapache2-mod-php5 php5-cli php5-curl
```

The PHP configuration used in command line is stored in /etc/php5/cli/php.ini. This is not to be confused with the Apache configuration which is stored in **/etc/php5/apache2/php.ini**. Here are some suggestions for modifications:

```
max_execution_time = 30
max_input_time = 60
memory_limit = 64M
upload_max_filesize = 10M
register_globals = Off
expose_php = Off
```

Please refer to the documentation for a complete description.

Don't forget to restart your Apache server after any modification of the PHP configuration.

```
$> service apache2 reload
```

As a matter of fact, this configuration is loaded on startup.

Test

In order to test your installation of PHP, we are going to create a PHP file that we will launch in the CLI (command line) and with Apache.

A default site is created when installing Apache, it is located in */var/www/html*

Create a file named *index.php* with the following content:

```
$> emacs /var/www/html/index.php
```

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

Now if you go on the address `localhost/index.php` you should see the following page:

PHP Version 5.6.13-0+deb8u1

System	Linux jolivad-debian8-vm 3.16.0-4-686-pae #1 SMP Debian 3.16.7-ckt11-1+deb8u4 (2015-09-19) i686
Build Date	Sep 8 2015 12:02:11
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API220131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	enabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk

This program makes use of the Zend Scripting Language Engine:
Zend Engine v2.6.0, Copyright (c) 1998-2015 Zend Technologies
with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2015, by Zend Technologies

The *`phpinfo()`* function is commonly used to test if php works fine.

MySQL

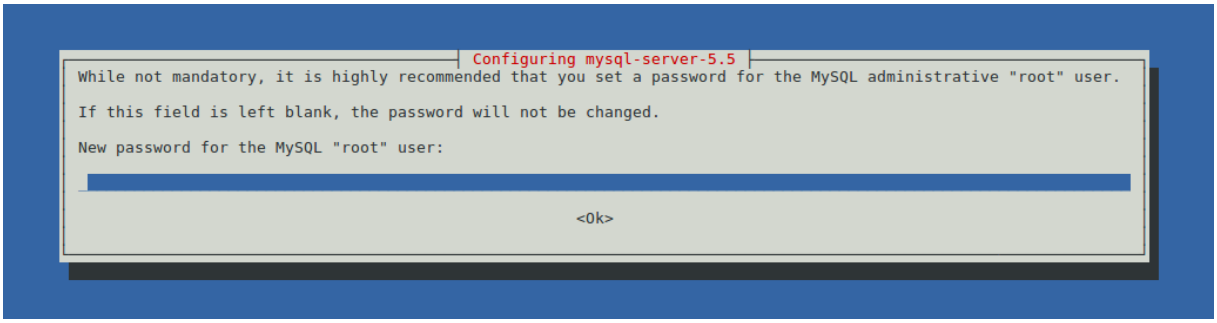
MySQL is a database management system.

Installation

Use this command line to install mysql on your linux

```
$> apt-get install mysql-server php5-mysql
```

During the installation process you should see the following screen asking for a new root password:
Set the password as you want. But remember it.



Test

Run the following command:

```
$> mysql -u root -p
```

Then type the password you entered before. You should have a new prompt:

```
$> mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 44
Server version: 5.5.44-0+deb8u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> exit
Bye
```