Web, Mobile and Security

Lab: Intro to Server-Side Scripting

1. Setting up the work environment

1.1. Making sure VMWare is installed

For this course, we will be working with a Debian Linux virtual machine, running on your host operating system. This virtual machine or VM can be considered as a computer running inside your computer.

If you do not have VMWare yet (the software needed to run a VM), you can download and install it.

- Connect to https://op-nas.howest.be:5001/
- Login using `CB\firstname.lastname` and your corresponding Howest password
- Launch `File Station`
- In the folder `TI-StudentShare/TI-S0-KickOff` you will find the VMWare software for both Windows and MacOS.

1.2. Installing the virtual machine

- If you have not done so yet, first download the OVA file:
  - Connect to https://op-nas.howest.be:5001/
  - Login using `CB\firstname.lastname` and your corresponding Howest password
  - Launch `File Station`
  - In the folder `TI-StudentShare/TI-S2-WebMobileSecurity` you will find the file `TI-S2-WebMobileSecurity.ova`
  - Copy this file to your machine
- Locate the OVA file on your machine and double-click to import into VMWare.

1.3. Launching the virtual machine

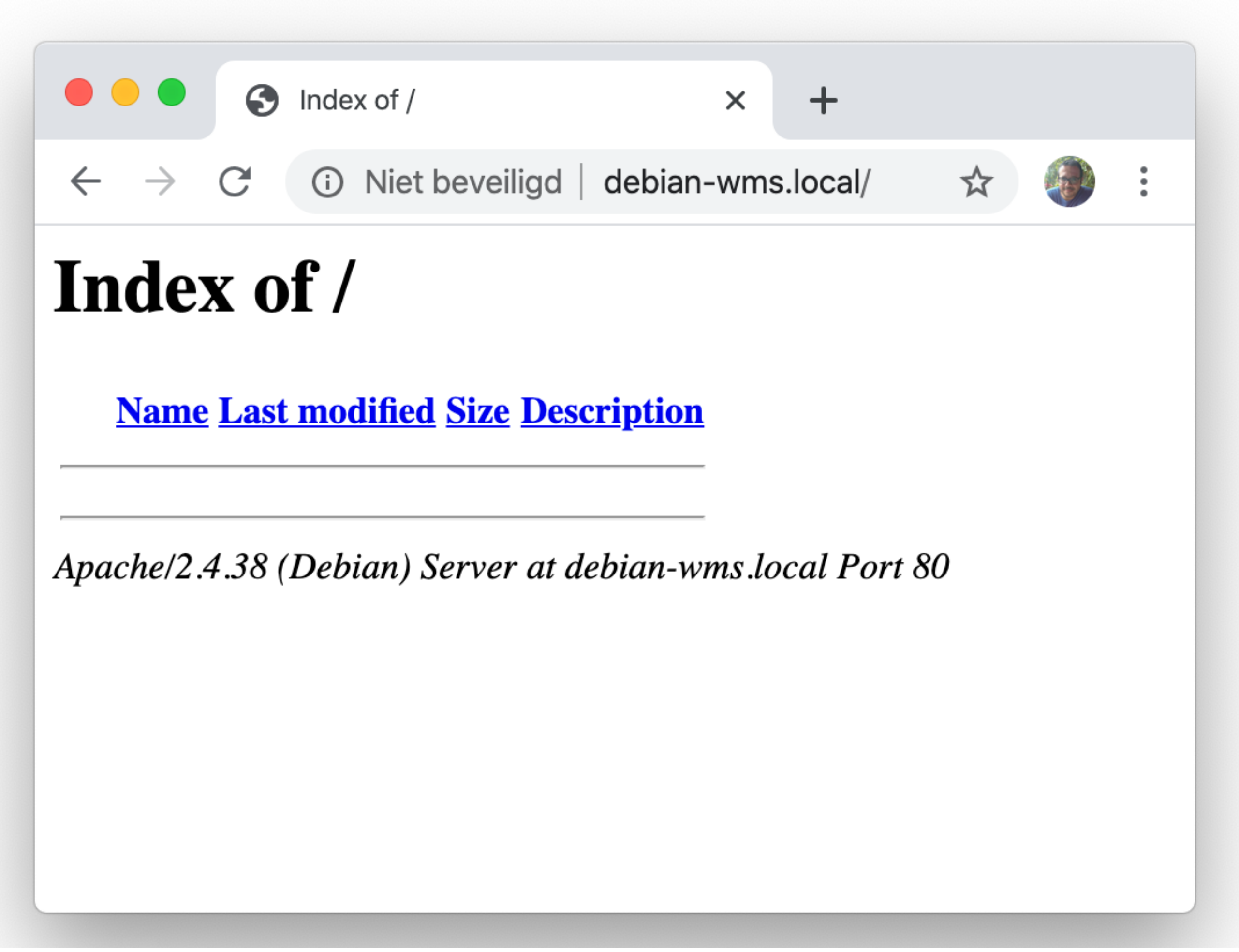
- Launch the virtual machine
- You can connect using one of the following credentials:
  - username: `user`, associated password: `user`
  - password: `root`, associated password: `root`
- Execute the command `ip addr` to obtain the IP address of your virtual machine.
- Next, on your host operating system, modify the hosts file and add an entry compared to the following:

```
192.168.215.130              debian-wms.local
```

> ‼ Make sure to put the IP address given by the `ip` command instead of the example above

> **>>** If you do not know how to modify the hosts file on your host operating system, refer to the procedure (PDF) on Leho.

- You should now be able to surf to http://debian-wms.local in your browser:



Index of /

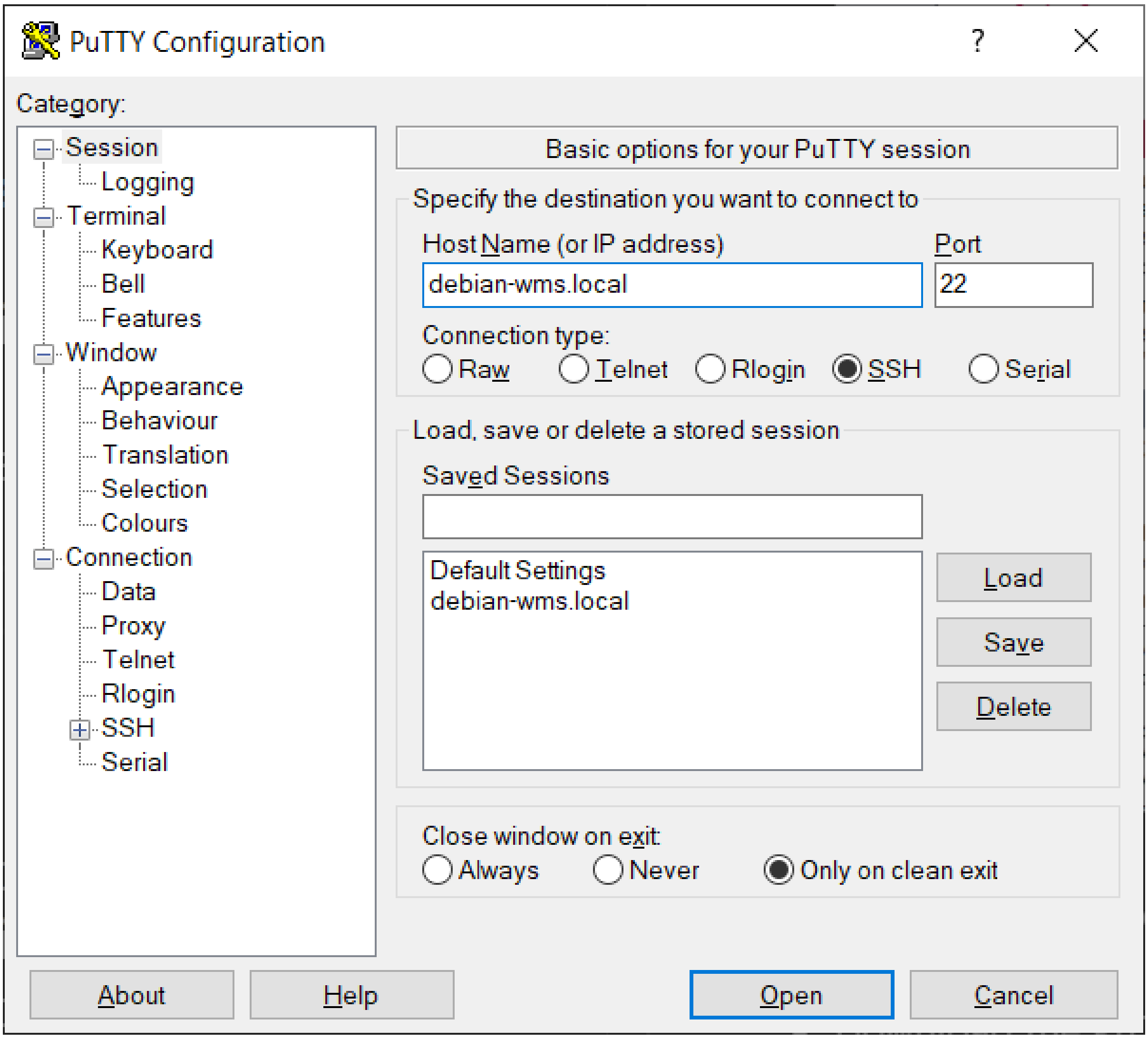
Name Last modified Size Description

Apache/2.4.38 (Debian) Server at debian-wms.local Port 80

1.4. Accessing the server using PuTTY or ssh

- Windows
  - Download the PuTTY client if you do not have it yet.
  - This application can be used to connect to the server from your Windows host.

  - You can use the following settings:



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): debian-wms.local

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions

Default Settings
debian-wms.local

Close window on exit: ○ Always ○ Never ● Only on clean exit

- MacOS
  - Use the `ssh` command from within Terminal to connect to your VM:

```
ssh -l user debian-wms.local
```

- You should be able to login as user `user` with the corresponding password `user`.

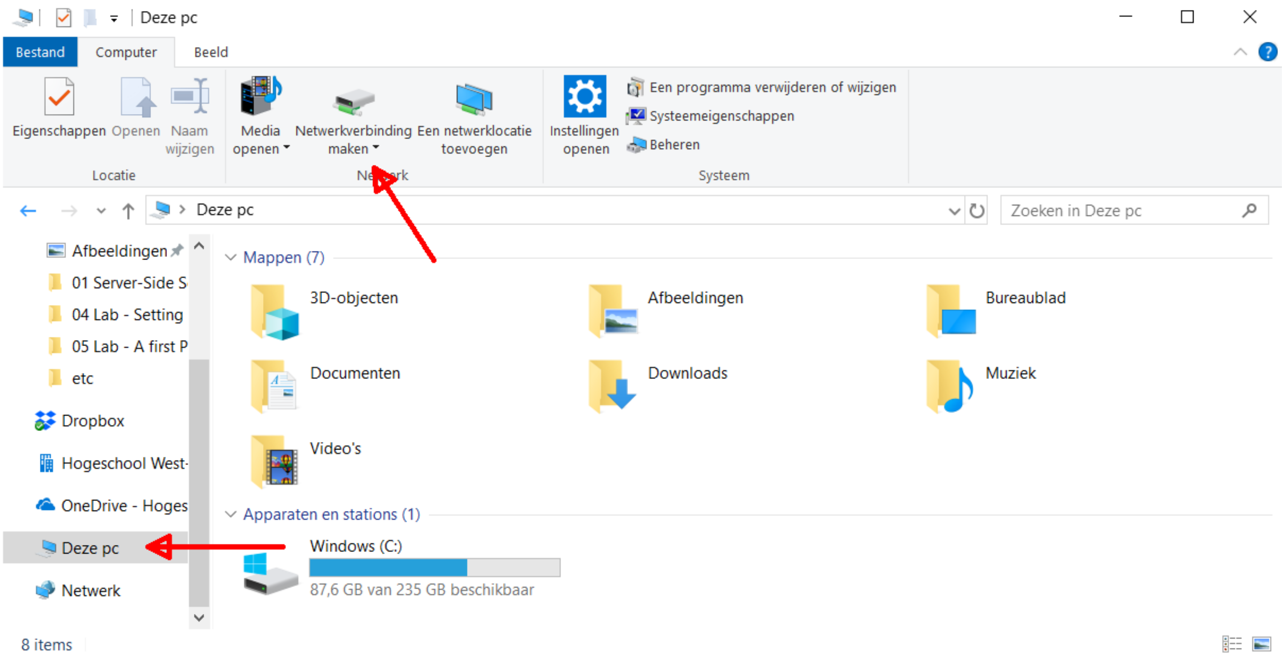
1.5. Accessing the file share on the VM

Since you will want to develop and write code from within your host operating system instead of the Linux VM, it is possible to access the required directories on the VM using a share.

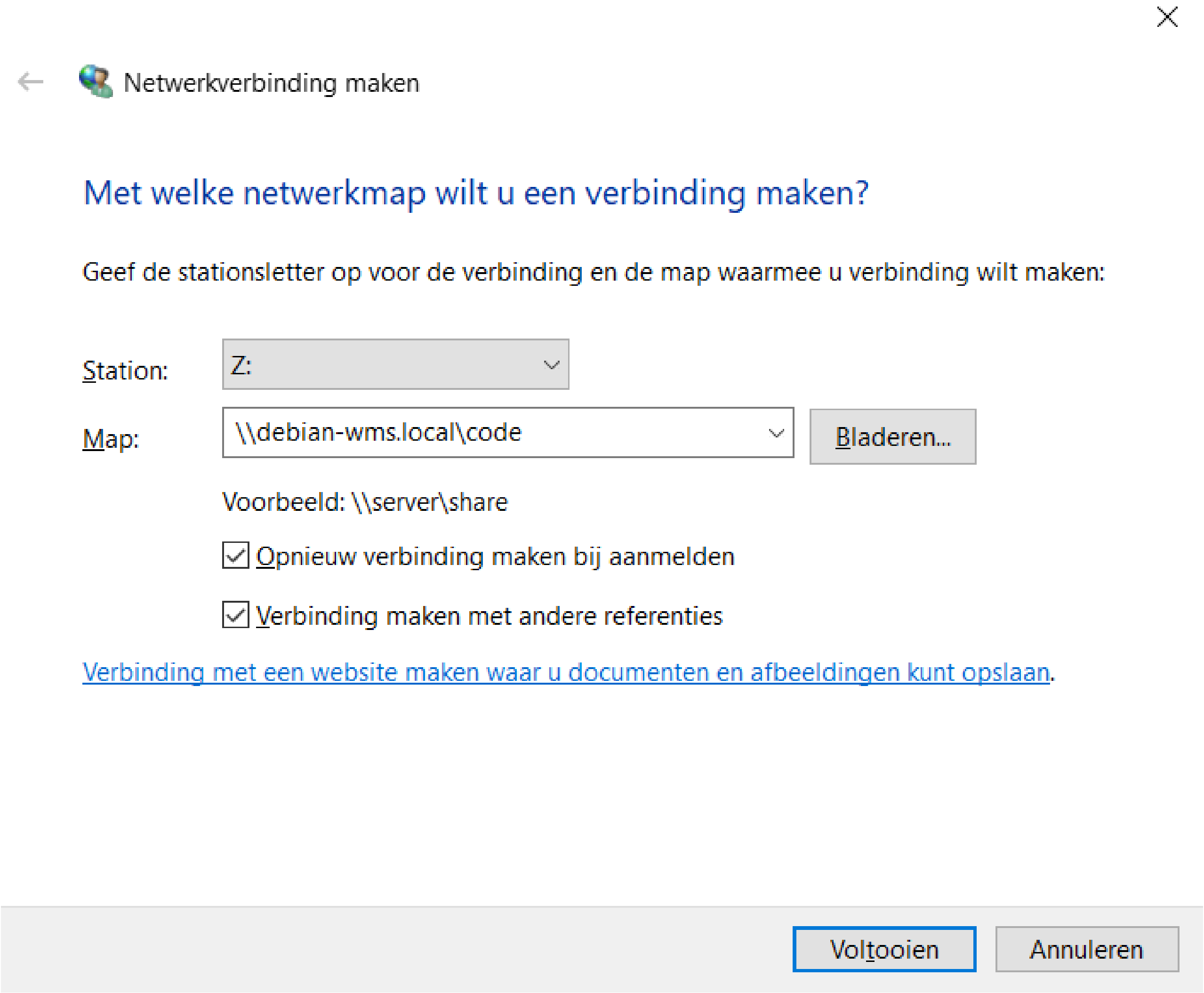
`\\debian-wms.local\code` is the name of share you need to map to a drive letter (Windows), or open using Finder (MacOS).

What follows is the procedure for Windows:

- Open Windows Explorer
- Add a new network connection



- Choose a free drive letter, specify the path to the share and make sure both checkboxes are checked.



Netwerkverbinding maken

Met welke netwerkmap wilt u een verbinding maken?

Geef de stationsletter op voor de verbinding en de map waarmee u verbinding wilt maken:

Station: Z:

Map: \\debian-wms.local\code

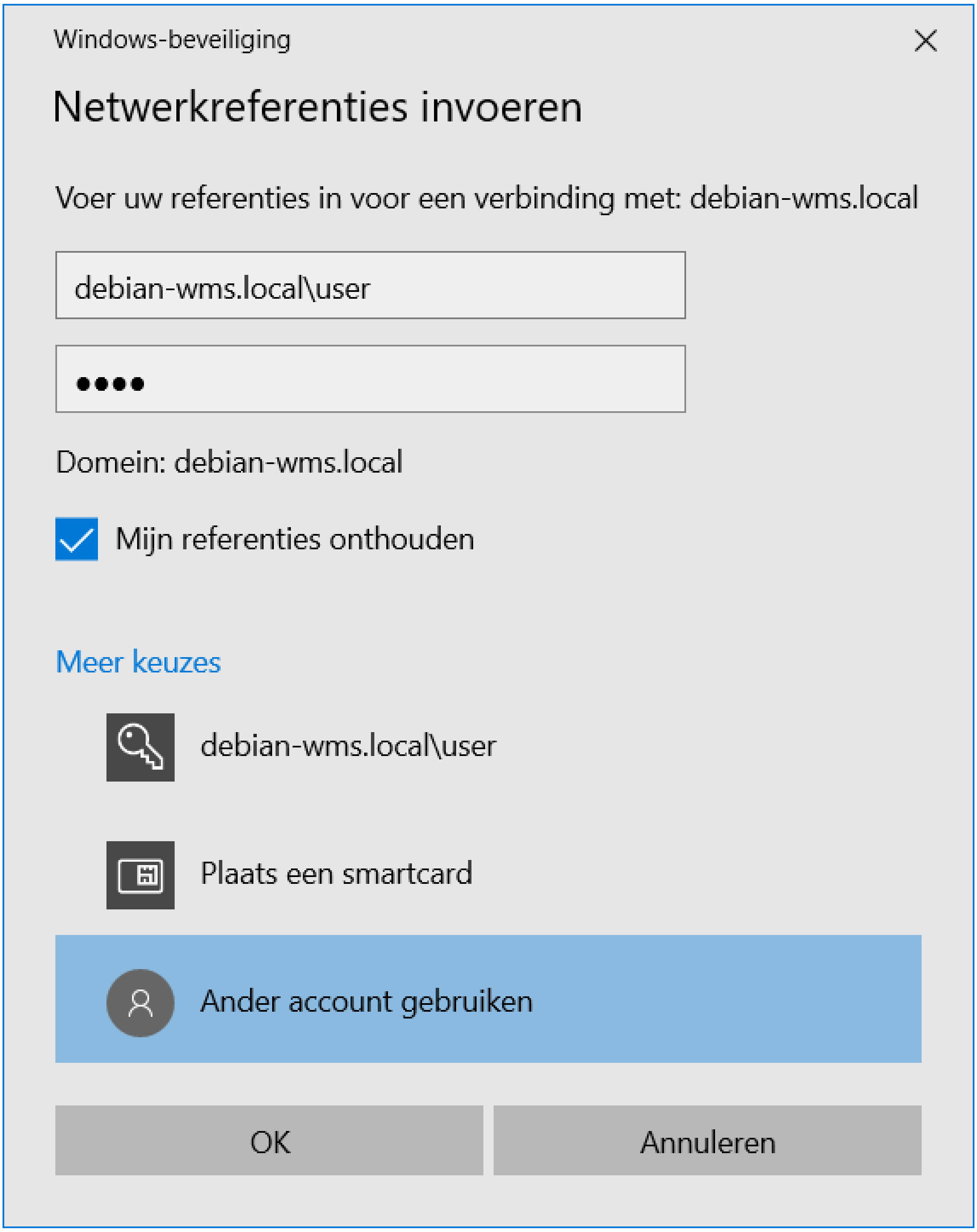
Voorbeeld: \\server\share

☑ Opnieuw verbinding maken bij aanmelden

☑ Verbinding maken met andere referenties

Verbinding met een website maken waar u documenten en afbeeldingen kunt opslaan.

- Enter the username and corresponding password (`user`), taking into account to check the box.



Windows-beveiliging

Netwerkreferenties invoeren

Voer uw referenties in voor een verbinding met: debian-wms.local

debian-wms.local\user

●●●●

Domein: debian-wms.local

☑ Mijn referenties onthouden

- You will now be able to access the share through the drive letter or directly via `\\debian-wms.local`.

2. Creating a new application

For each exercise, you can create a new application. This is done from within PuTTY/ssh by executing one of the following scripts:

- `build-laravel-project` (will be used later)
- `build-plain-project` (used in this lab)

For example, to create an application called `hello-world`, execute the following command from PuTTY:

```
./build-plain-project hello-world
```

If prompted, enter the password (`user`).

The new application is then created in the subdirectory `code`. All public files (=available through http://debian-wms.local/hello-world) are stored in the subdirectory `/home/user/code/hello-world/public`.

3. Trying out the new application

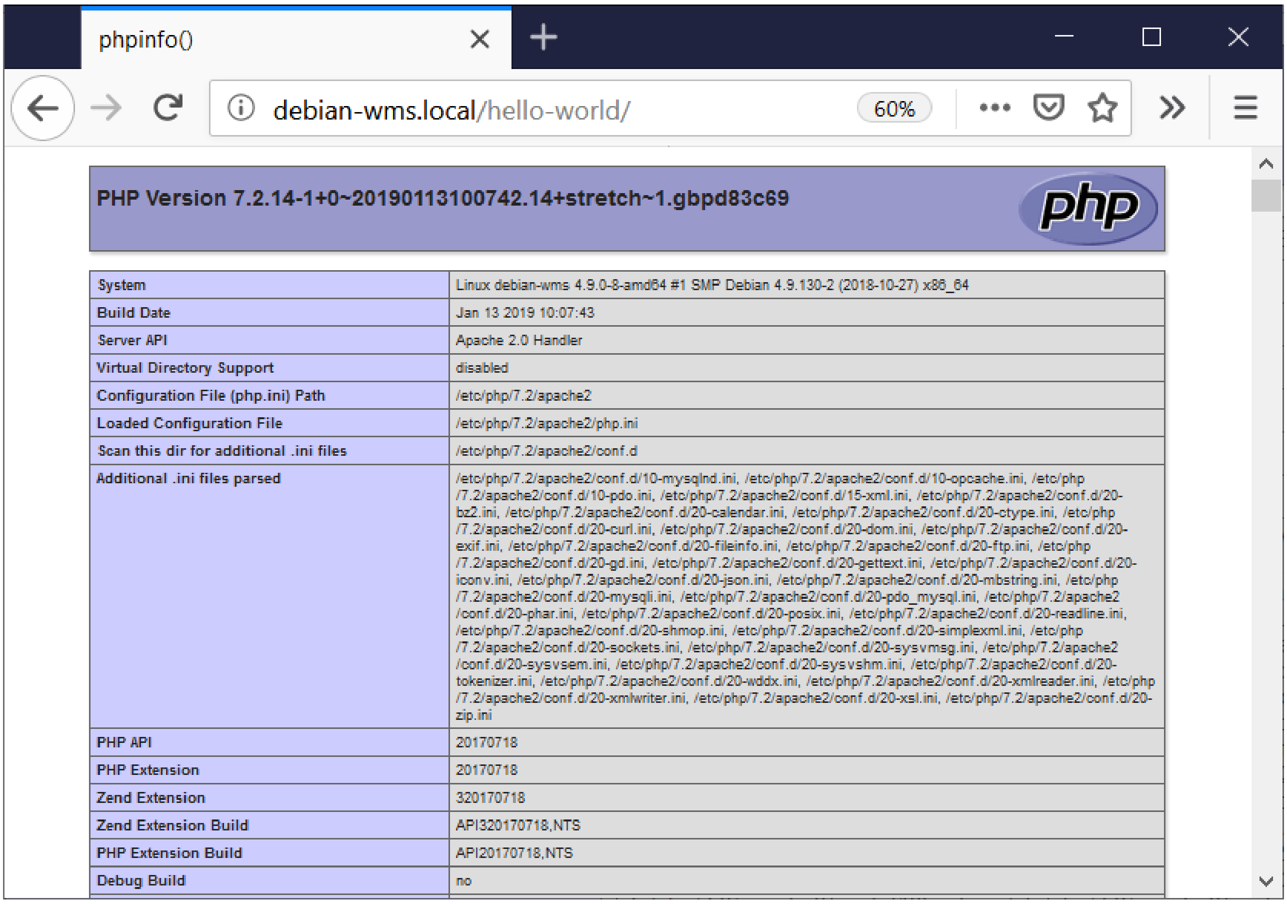
- Create a file `index.php` in the share `\\debian-wms.local\code` using your favourite editor (VSCode, PHPStorm, ...)
- In this file, put the following code:

```
<?php

    phpinfo();

?>
```

- Next, surf to http://debian-wms.local/hello-world
- If all goes well, you will see a PHP status page:



PHP Version 7.2.14-1+0~20190113100742.14+stretch~1.gbpd83c69

System	Linux debian-wms 4.9.0-8-amd64 #1 SMP Debian 4.9.130-2 (2018-10-27) x86_64
Build Date	Jan 13 2019 10:07:43
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bz2.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysqli.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-phar.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-wddx.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,NTS
PHP Extension Build	API20170718,NTS
Debug Build	no

4. Exercise: Lotto

- Create a new plain web application called `lotto`.

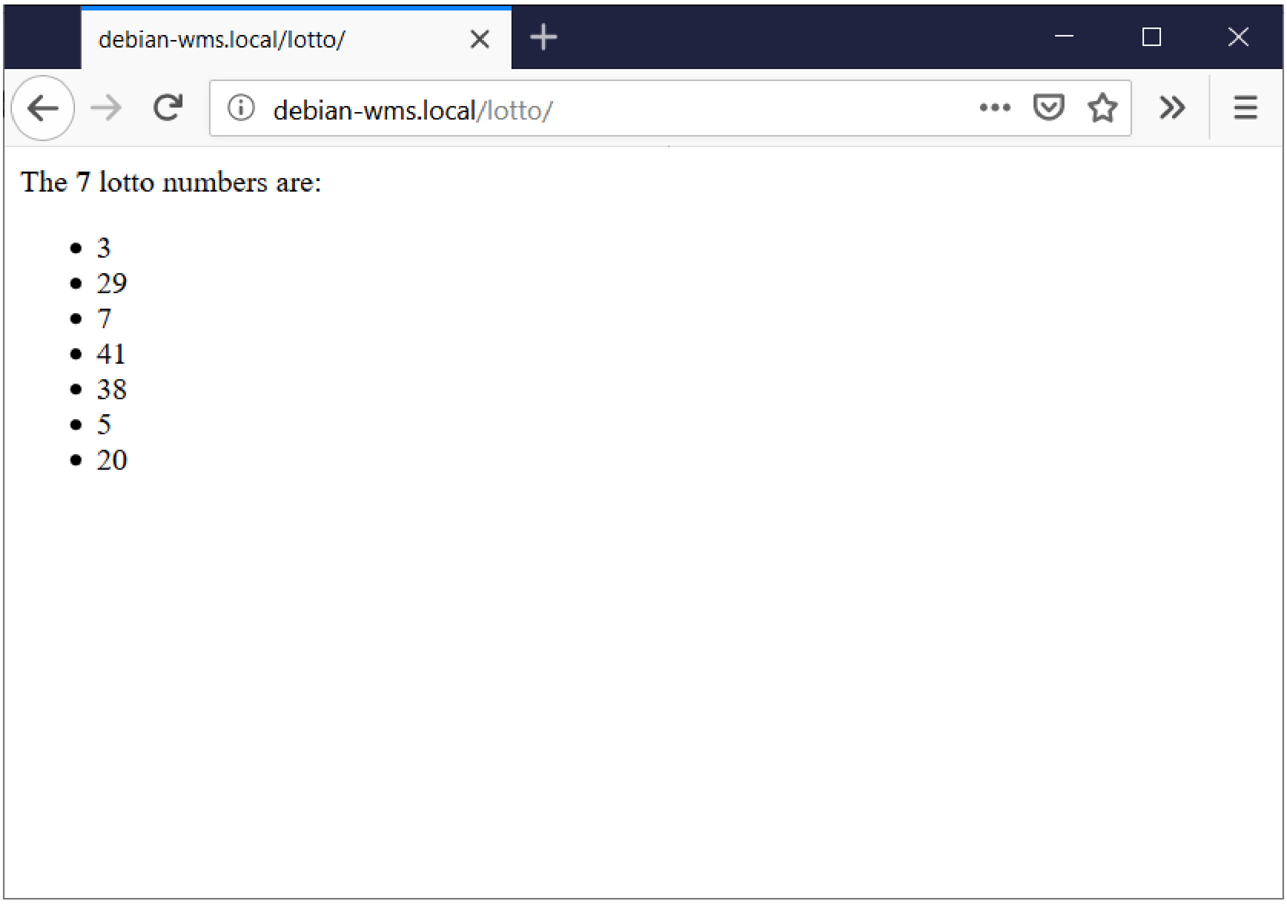
- In the `public` folder, create a file called `index.php` with the standard HTML5 template.
- Next up:
    1. Write a function `generate_lotto_numbers` which accepts one argument *n*, that generates an array of *n* unique random numbers between `1` and `42`.
    2. Call the function to obtain an array of **7** numbers and store the result in a variable `$results`.
    3. Loop over the array `$results` and write out the individual numbers one by one.
- What your code will probably look like (complete the TO DOs):

```
<?php
    function generate_lotto_numbers($n) {
        // TO DO
    }

    // store the result of the function call
    // in variable $results
    $results = generate_lotto_numbers(7);
?>

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

- The resulting page can look something like this:



The 7 lotto numbers are:

- 3
- 29
- 7
- 41
- 38
- 5
- 20

5. Uploading the Lotto exercise onto op-gitlab

1. Navigate to `https://op-gitlab.howest.be/TI/2019-2020/s2-web-mobile-and-security/students/student-firstname-lastname`, where you replace *firstname-lastname* by your own op-gitlab login name, but with dashes (-) instead of dots (.).

   This is your own dedicated group, where you can create different repositories.

2. Within the group, create a repository called `01-lotto` using the `New project` button.

3. Next, open a `Git Bash` (Windows) or a `Terminal` (macOS/Linux).

4. `cd` to the folder containing your PHP files and issue the following commands:

```
 git init
 git remote add origin https://op-gitlab.howest.be/TI/2019-2020/s2-web-mobile-and-security/students/student-firstname-lastname/01-lotto.git
 git add .
 git commit -m "Initial commit"
 git push -u origin master
```

As before, make sure to replace `firstname-lastname` by your own group name!