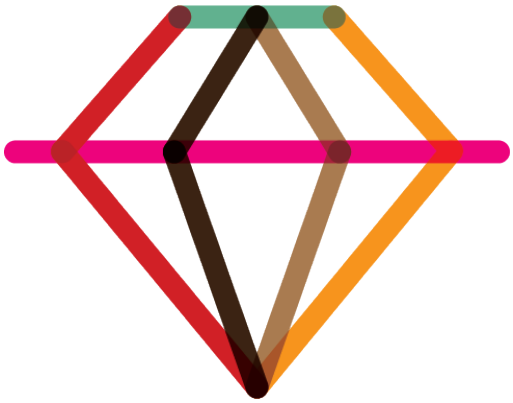


Install and maintenance guide

INDEX

From ZERO to HERO

Updates

Some considerations

This is a guide to install and maintain the App for Value Villages. To make it easier, we have different typographies and color for text like this that is an explanation

```
And text like this that will be code
```

On the code text, there will be one difference:

A) The commands to introduce in the terminal will start with $:

```
$ like this one, even is a multiline command because
it is so long, is only one command
```

And then you should press enter to execute.
Please note that the `$` symbol should be omitted, is just to highlight the fact that is one command

B) And the code that we need to write to some files with the text editor nano

```
This commands will be without $, like this one
```

Considering the network connection quality in Turkana, please make sure to have a good and stable connection in the moment to proceed with this manual.

All the code written in pink, you need to copy it literally, if you miss a single comma it won't work. There will be some wordsInBlue, that are parameters that you need to configure (just names even by you for you to make easier to follow the tutorial)

When mix the two colours, like:
Press `Control + X`
Press `Y`
Press `Enter`
Means that you need to press the `Keys` in the keyboard

Install VV software in the RaspberryPI, from ZERO to HERO

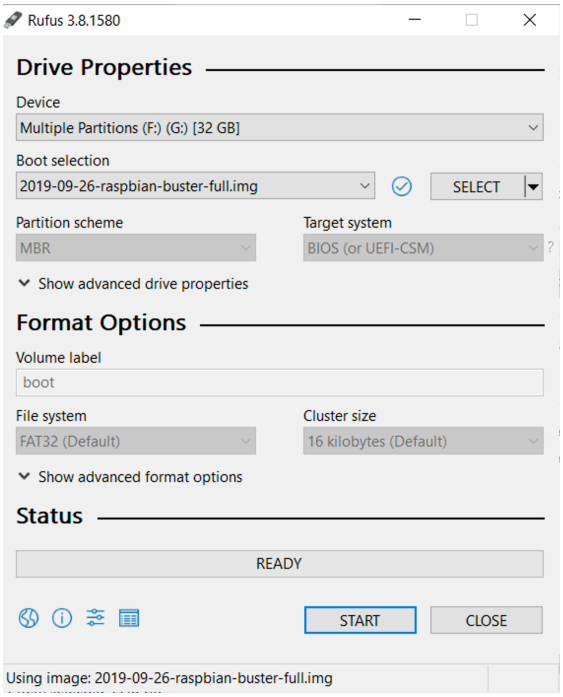
1) Set up the Hardware

Install the heatsink in the microchips of the RaspberryPI, put the thermal paste and put the clock module. Then install the RaspberryPI inside the case. Introduce the SD card.

2) Download The SO:

Make this whole step in a Windows computer.
Note: if you want to use macOS, you can use the free tool "ApplePiBacker" or "balenaEtcher"
Download the software "Rufus", is a free open-source program to make the copy the ISO image in the microSD card of the raspberryPI.
Download the option "Raspbian Buster with desktop and recommended software" in this webpage:
https://www.raspberrypi.org/downloads/raspbian/

Don't unzip it. Make an ISO image in the SD card with the ZIP downloaded:



In device select the microSDcard to install the image.

In Boot selection select the unzip ISO you have downloaded.

In File format select **FAT32**.

Whenever is finished, just press CLOSE

Introduce the SD card in the RaspberryPI and connect the micro HDMI to a Screen and a keyboard and a mouse.
The microHDMI connected to the screen is the one closest to the USB-C.

3) Initial Set up

Once you start the RaspberryPI for the first time, follow the initial instructions (just click next, next, next... and don't put any password). Choose update software.
Connect to the Wifi and reboot.
Open the terminal and we need to install:

Firmware:

```
$ sudo apt update
$ sudo apt upgrade
$ sudo reboot
```

If you think it did not update all necessary because internet connection, you can repeat the three commands again.

4) Run the app: Docker

This will run docker every time the RaspberryPI is reboot. Docker will write the data created into a folder that we are going to create on purpose in de Desktop:

To install docker, introduce:

```
$ curl -fsSL https://get.docker.com -o get-docker.sh
$ sh get-docker.sh
$ sudo usermod -aG docker $USER
$ sudo reboot
```

Now, we install the last image from the DockerHub:

```
$ docker pull dojranovski/vvimg
```

Now we create a folder in the desktop, and we linked with the docker container, so the docker container will write the information in the folder we just created.
Then execute this command, is only one long command.

```
$ docker run -d --name nameContainer  -v
pathToFolderInDesktop:/go/src/github.com/
mongolhippie/vv/local-resources -e VILLAGE=Central -e
GIN_MODE=release -e GOPATH=/go -p 8080:8080
--restart=always dojranovski/vvimg
```

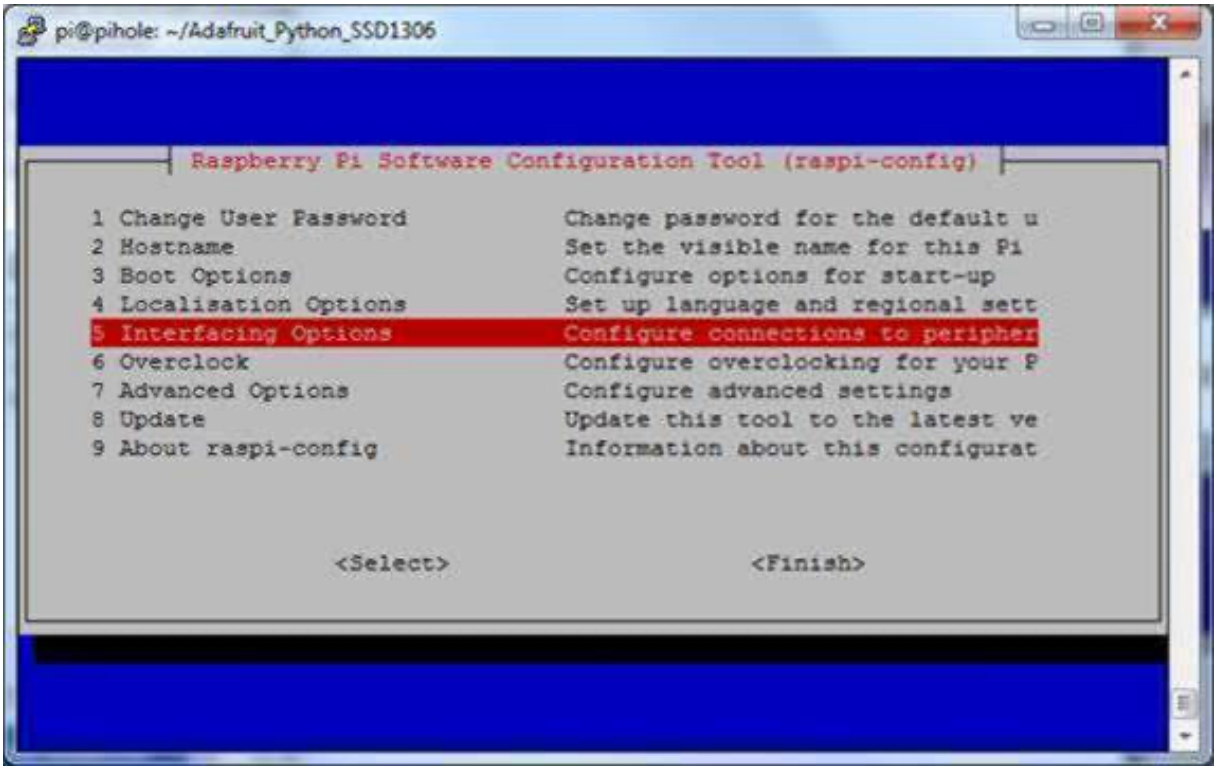
Now the app is running, you can check in your browser:
URL: localhost:8080/dashboard

5) Install Hardware Clock

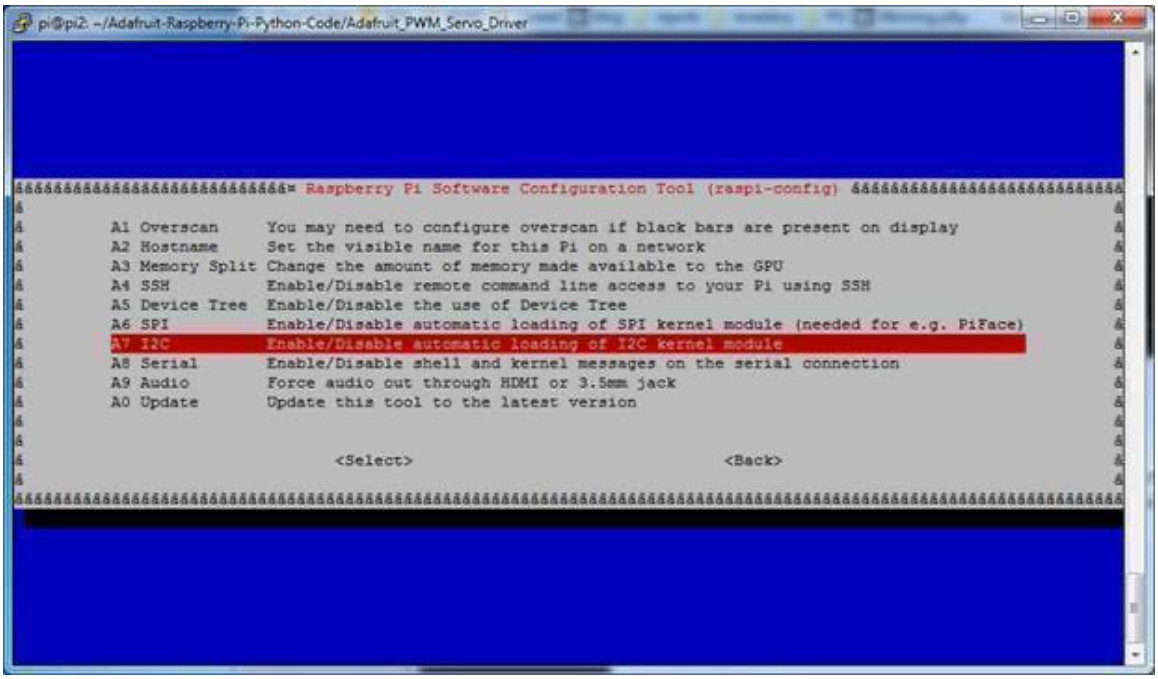
-Install kernel support

```
$ sudo raspi-config
```

Go to 5- interfacing options



Select to I2C



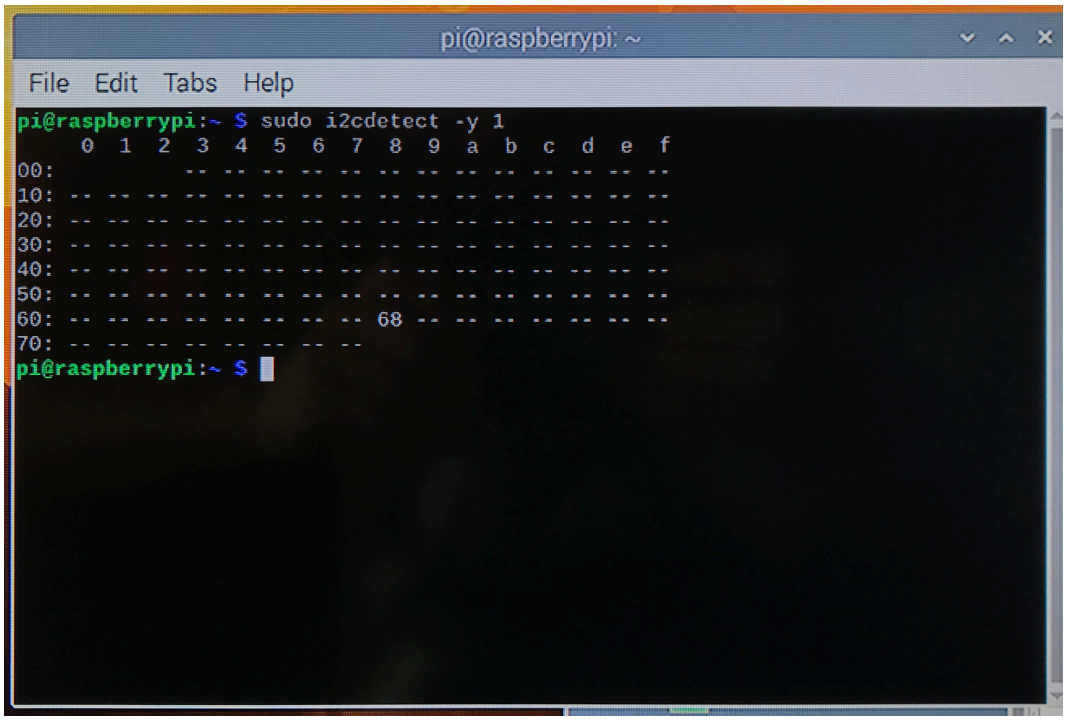
Click YES for enable


```
$ sudo reboot
```

-Test I2C

Now when you log in you can type the following command to see all the connected devices

```
$ sudo i2cdetect -y 1
```



Introduce

```
$ sudo nano /boot/config.txt
```

Add this to the end of the file

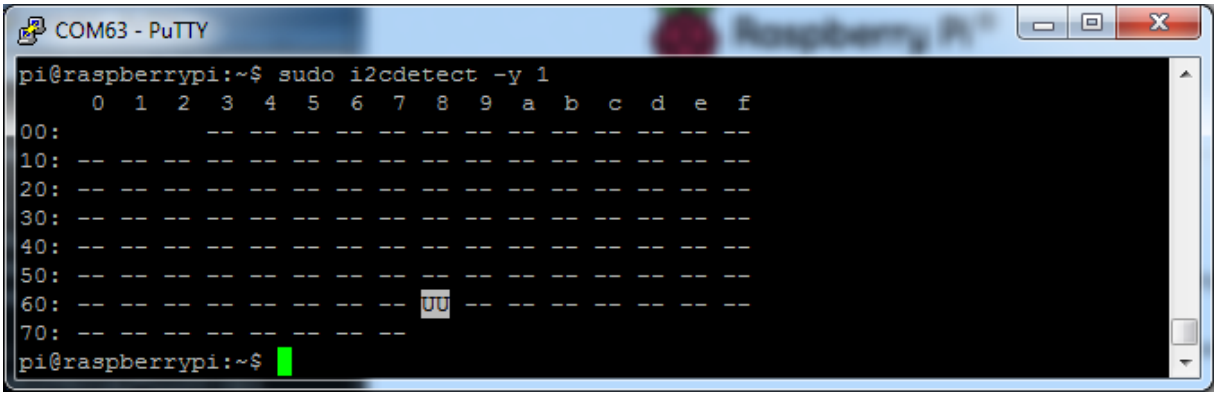
```
dtoverlay=i2c-rtc,ds3231
```

Press `Control + X`
Press `Y`
Press `Enter`

```
$ sudo reboot
```

We need to see the UU show up where 68 was before

```
$ sudo i2cdetect -y 1
```



Disable the "fake hwclock" which interferes with the 'real' hwclock

```
$ sudo apt-get -y remove fake-hwclock
$ sudo update-rc.d -f fake-hwclock remove
$ sudo systemctl disable fake-hwclock
```

Now with the fake-hw clock off, you can start the original 'hardware clock' script. Run:

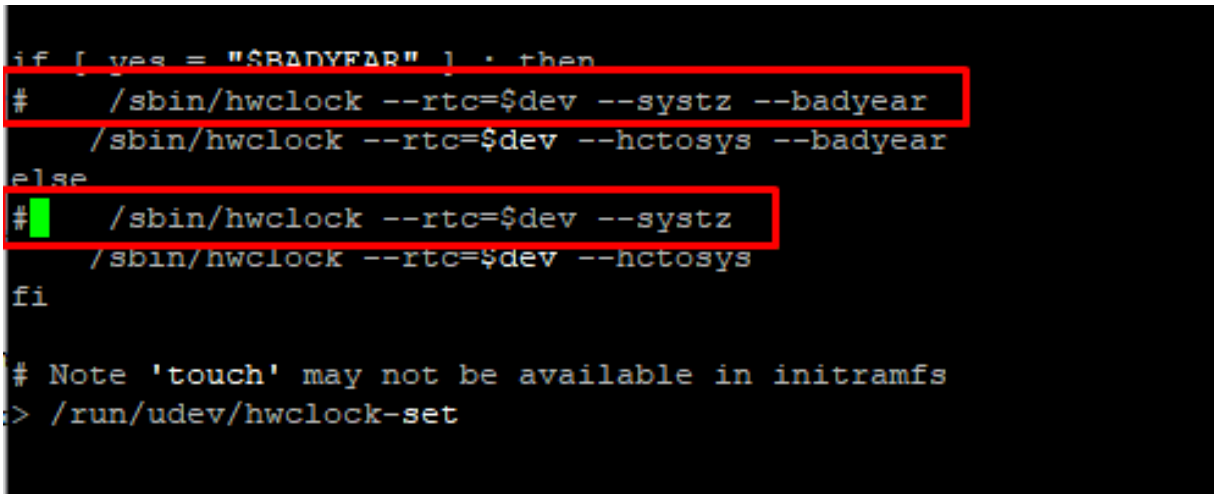
```
$ sudo nano /lib/udev/hwclock-set
```

and comment out these three lines:

```
# if [ -e /run/systemd/system ] ; then
#  exit 0
# fi
```

Also comment out the two lines

```
# /sbin/hwclock --rtc=$dev --systz --badyear
```




```
# /sbin/hwclock --rtc=$dev --systz
```

Press `Control + X`

Press `Y`

Press `Enter`

-Sync time from Pi to RTC

When you first plug in the RTC module, it's going to have the wrong time because it has to be set once. You can always read the time directly from the RTC with

```
$ sudo hwclock -D -r
```

You can see, the date at first is invalid! You can set the correct time easily, just run:

```
$ date
```

to verify the time is correct. Plug in Ethernet or WiFi to let the Pi sync the right time from the Internet. Once that's done, write the time with :

```
$ sudo hwclock -w
```

And then read the time with:

```
$ sudo hwclock -r
```

Once the time is set, make sure the clock module is inserted inside the RaspberryPI so that the time is saved. You only have to set the time **once** That's it! Next time you boot the time will automatically be synced from the RTC module

```
$ sudo reboot
```

6) Make the access point for broadcast the WIFI with the app

These are the two programs we're going to use to make your Raspberry Pi into a wireless access point. To get them, just type these lines into the terminal:

```
$ sudo apt-get install hostapd dnsmasq
```

Both times, you'll have to hit y to continue. hostapd is the package that lets us create a wireless hotspot using a Raspberry Pi, and dnsmasq is an easy-to-use DHCP and DNS server.
We're going to edit the programs' configuration files in a moment, so let's turn the programs off before we start tinkering:

```
$ sudo systemctl stop hostapd dnsmasq
```

Let's assign the IP address **192.168.0.10** to the wlan0
interface by editing the dhcpcd configuration file. Start editing with this command:

```
$ sudo nano /etc/dhcpcd.conf
```

Now that you're in the file, add the following lines at the end:

```
interface wlan0
static ip_address=10.1.1.1/24
nohook wpa_supplicant
```

Press `Control + X`
Press `Y`
Press `Enter`

```
$ sudo reboot
$ sudo service dhcpcd restart
```

We're going to use dnsmasq as our DHCP server. The idea of a DHCP server is to dynamically distribute network configuration parameters, such as IP addresses, for interfaces and services.
Let's rename the default configuration file and write a new one:

```
$ sudo mv /etc/dnsmasq.conf  /etc/dnsmasq.conf.orig
$ sudo nano /etc/dnsmasq.conf
```

You'll be editing a new file now, and with the old one renamed, this is the config file that dnsmasq will use. Type these lines into your new configuration file:

```
interface=wlan0
dhcp-range=10.1.1.100,10.1.1.200,255.255.255.0,24h
```

The lines we added mean that we're going to provide IP addresses between 10.1.1.100 and 10.1.1.200 for the wlan0 interface

Press `Control + X`
Press `Y`
Press `Enter`

```
$ sudo systemctl reload dnsmasq
```

-Configure the WIFI

```
$ sudo nano /etc/hostapd/hostapd.conf
```

This should create a brand new file. Type in this:

```
interface=wlan0
driver=nl80211
ssid=nameWifi
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=passwordWifi
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
passwordWifi needs to be with at least 8 characters
```

Press `Control + X`
Press `Y`
Press `Enter`

We still have to show the system the location of the configuration file:

```
$ sudo nano /etc/default/hostapd
```

In this file, track down the line that says #DAEMON_CONF="" – delete that # and put the path to our config file in the quotes, so that it looks like this:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Press `Control + X`
Press `Y`
Press `Enter`

Now we need to update the hostapd:

```
$ sudo systemctl unmask hostapd
$ sudo systemctl enable hostapd
$ sudo systemctl start hostapd
```

If the error given says "Job for hostpad.service failed because the control process exited with error code", it means that you have a typo in one of the three last files you have edited. Search it and correct it.

Do a quick check of their status to ensure they are active and running:

```
$ sudo systemctl status hostapd
$ sudo systemctl status dnsmasq
```

Press `q` to go out

Set up traffic forwarding

```
$ sudo nano /etc/sysctl.conf
```

Now find this line and uncomment it:

```
net.ipv4.ip_forward=1
```

Press `Control + X`
Press `Y`
Press `Enter`

Next, we're going to add IP masquerading for outbound traffic on eth0 using iptables:

```
$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

and save the new iptables rule:

```
$ sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

Install these rules on boot:

```
$ sudo nano /etc/rc.local
```

and add this just above "exit 0" to install these rules on boot.

```
iptables-restore < /etc/iptables.ipv4.nat
```

Press `Control + X`
Press `Y`
Press `Enter`
Reboot and ensure it still functions:

```
$ sudo reboot
```

Now you can connect to the Wifi, introduce your password and go to

URL: 10.1.1.1:8080/dashbboard

Install VV software in a Windows 10 PC, from ZERO to HERO

1) Install Go (Golang)

Go to golang.org/dl/ and download the Windows version.
Click next, next, next and leave everything as default.
Once install, open the terminal and check the installation

```
$ go version
```

2) Install git:

Go to git-scm.com/download/win
Download and install git (just click next, next, next...)
Restart the computer.
Check in the terminal:

```
$ git --version
```

3) Get the project and the dependencies

```
$ go get github.com/mongolhippie/vv/...
```

It won't show any update or feedback in the terminal, will appear like frozen for several minutes (maybe 20), and then will finished and all the packages will be downloaded
It may send an error when finished, but it should leave the packages in /Users/userName/go/src/github.com/mongolhippie/vv

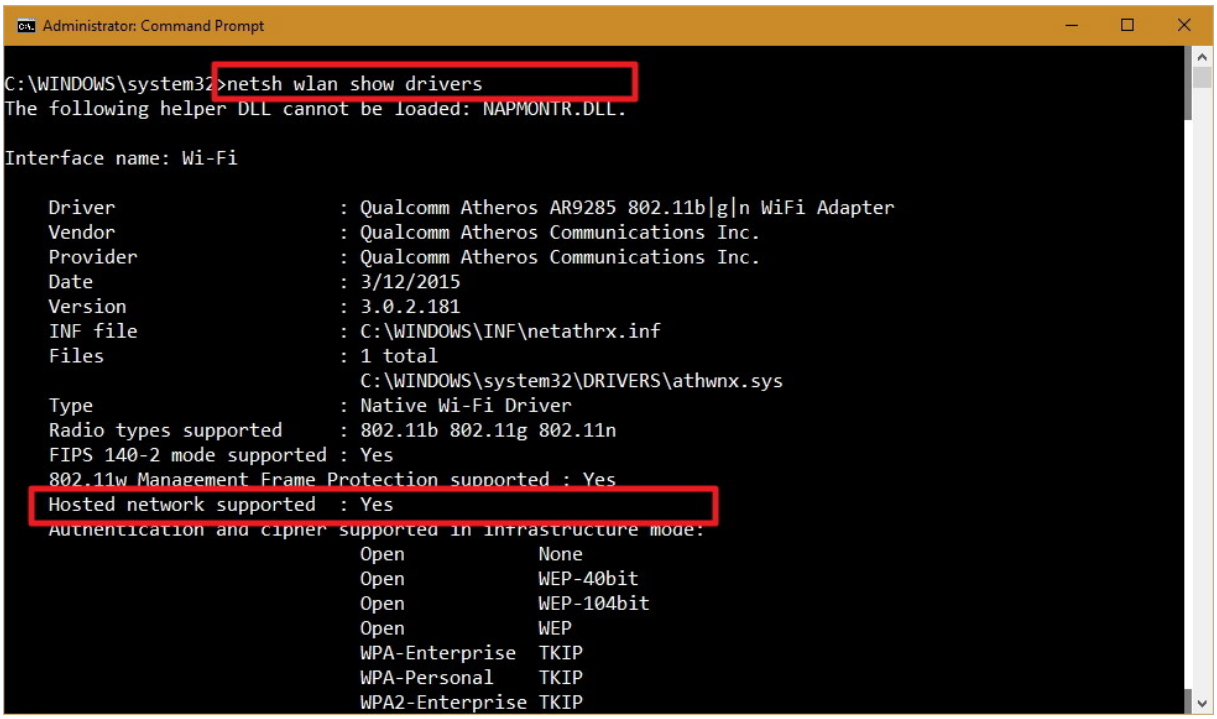
4) Install the PDF extension

Go to to wkhtmltopdf.org/downloads.html
Choose the one for Windows, download and install it.
Restart the computer
It should be installed in C:\ProrammFiles\wkhtmltopdf\bin\wkhtmltopdf.exe

5) Connect to the Windows through mobile device

While some adapters include support for Hosted Network, you will first need to verify your computer's physical wireless adapter supports this feature using the following command

```
$ NETSH WLAN show drivers
```



If is supported, go on with the tutorial. If not, get a computer that supports it.

For create the network, run the terminal as Administrator:

```
$ NETSH WLAN set hostednetwork mode=allow
ssid=WifiName key=WifiPassword
```

Remember that the passphrase has to be at least 8 characters in length

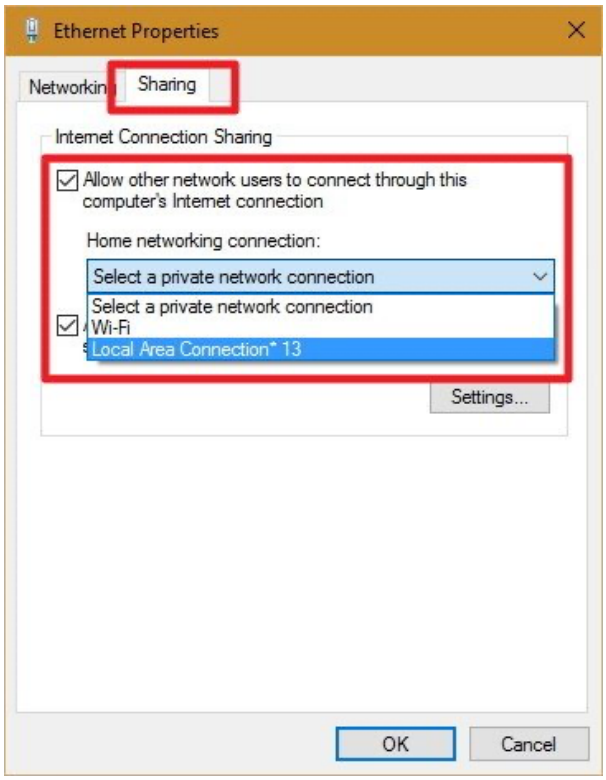
TIP: How to run it as administrator? Go to the search bar (down left in the windows screen), write cmd, and rich click on the command promp>run as administrator.

Activate the server:

```
$ NETSH WLAN start hostednetwork
```

The last thing you need to do is to share an internet connection using the "Internet Connection Sharing":

1. Use the Windows key + X keyboard shortcut to open the Power User menu, and select Network Connections. Click Change adapter options.
2. Next, right-click the network adapter with an internet connection – the wifi connected — select Properties. **Go to the tab Sharing.**
3. Check the Allow other network users to connect through this computer's Internet connection option.
4. Next, from the Home networking connection drop-down menu, select “Local Area Connection”. If there is no “Local Area Connection”, select the “WiFi” Adapter. Click OK.



If by any chance, the dropdown menu does not appear, just enable the 2 checkboxes and and press OK. The dropdown menu will appear only the very first time you make this tutorial.

6) Run the app

Here we have different options (choose only one!):

a) Run the app once

The app will be running until the computer restarts or shut down.

```
$ set VILLAGE=nameVillage
$ cd go\src\github.com\mongolhippie\vv\cmd\valuevillages
$ go run main.go
```

It may prompt and alert from Windows, you need to click Allow.

Now you can see the app in
`localhost:8080/hello`

b) Run the app continuously

This will run the app all the time the computer is on and logged into the Windows desktop. We will use docker for that. Due to some technical incompatibilities, VirtualBox it is not compatible with docker, so VirtualBox will no longer work. (It is not relevant if the computer does not have the program VirtualBox installed or if the main use of the computer is to have the app running).

Go to id.docker.com/login

You will need to create an account with a dockerID (this is just the username you want to user to log in) and verify your email address.
Once this is done, go again and log in into the docker page, from there you will be able to download docker Desktop.
Install the app and restart the computer.

It will promp an alert saying that Hyper-V and Containers features are not enabled. You need to install these and press OK in order to continue, but VirtualBox will no longer work in that computer. The computer will download the files and will restart the computer itself.

Give it few minutes until docker starts, and then introduce this command

```
$ cd go\src\github.com\mongolhippie\vv
```

Build the image where docker will start from

```
$ docker build -t nameImage .
```

And after few minutes of building the image, we need to launch it with the next command (is only one long command):

```
$ docker run -d --name nameContainer -e
VILLAGE=Central -e GIN_MODE=release -e GOPATH=$HOME/
go -p 8080:8080 --restart=always nameImage
```

7) Connect to the WiFi

Then, the Wi-Fi will be emitting signal, so with another device you just connect to the wifi, and in the Windows computer write in the terminal

```
$ ipconfig
```

Take the Auto-configuration IPv4 address (for example, 169.254.219.17) and add the 8080 port, then go to the URL. Example:

```
URL= 169.254.219.17:8080/hello
```

And then will prompt the login screen of VV app

Install VV software in macOS X, from ZERO to HERO

1) Install Go (Golang)

Go to golang.org/dl/ and download the Mac version.
Click next, next, next and leave everything as default.
Once install, restart.
Open the terminal and check the installation

```
$ go version
```

2) Install Developer tools:

If you did not have it installed in your computer, you will need to install it:

```
$ sudo xcode-select –install
```

3) Get the project and the dependencies

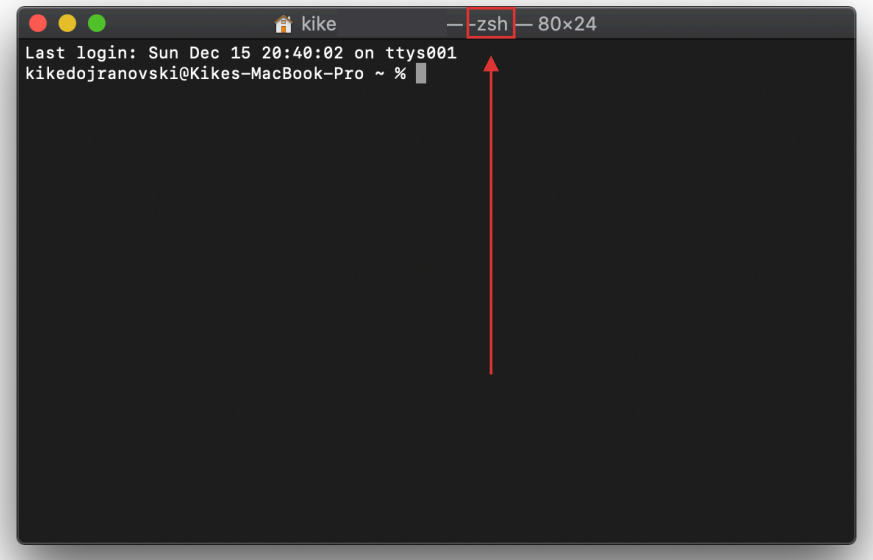
```
$ go get github.com/mongolhippie/vv/...
```

It won't show any update or feedback in the terminal, will appear like frozen for several minutes (maybe 20), and then will finished and all the packages will be downloaded

4) Setting $GOPATH

```
$ go env -w GOPATH=$HOME/go
```

Note: from macOS X 10.15 (Catalina) on, the default terminal is Zsh instead of bash. You can identify yours on the top on the terminal's window



a. Setting $GOPATH in BASH

Now we need to edit the bash, this command it will write ~/.bash_profile

```
$ export GOPATH=$HOME/go
$ source ~/.bash_profile
```

b. Setting $GOPATH in Zsh

Now we need to edit the bash, this command it will write ~/.zshrc

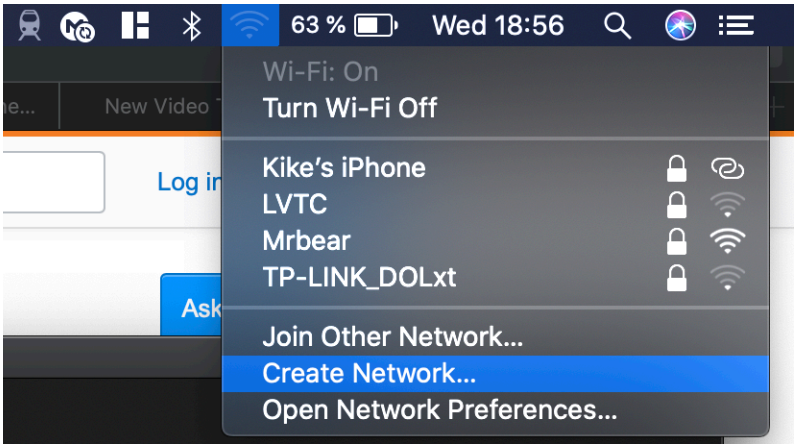
```
$ export GOPATH=$HOME/go
$ source ~/.zshrc
```

Check that worked:

```
$ echo $GOPATH
```

5) Set the access point in the Mac computer

Go to the top bar, the Wifi icon and press Create new network...



Give a name to the network and press create

6) Install wkhtmltodfp (for being able to make the pdf for the QR)

Go to to wkhtmltopdf.org/downloads.html

Choose the one for Mac, download and install it.

7) Run the app

a) Run the app once

```
$ cd go/src/github.com/mongolhippie/vv
$ VILLAGE=nameVillage go run main.go
```

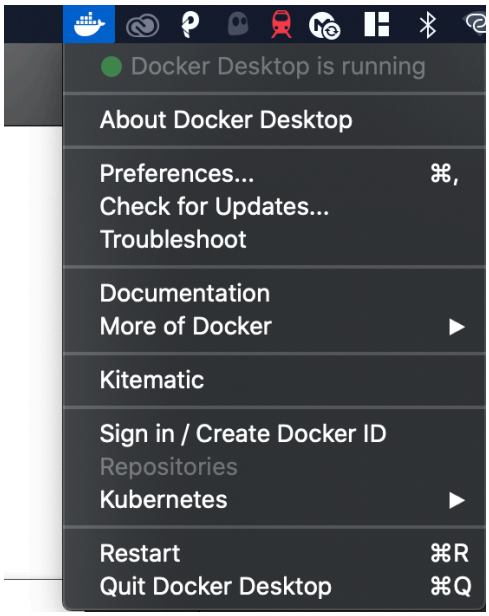
b) Run the App continuously with Docker:

Go to id.docker.com/login

You will need to create an account with a dockerID (this is just the username you want to user to log in) and verify your email address.

Once this is done, go again and log in into the docker page, from there you will be able to download docker Desktop.

Install the app and open it. Give the privileges it is asking introducing the password.




```
$ docker info
```

Type to see the info about docker, if the answer is different than “zsh: command not found: docker” then Docker is working

Now go to the main directory of the project

```
$ cd go/src/github.com/mongolhippie/vv
```

Build the image where docker will start from

```
$ docker build -t nameImage .
```

-t is for tag the build to the nameImage

And after few minutes of building the image, we need to launch it with the next command (is only one long command):

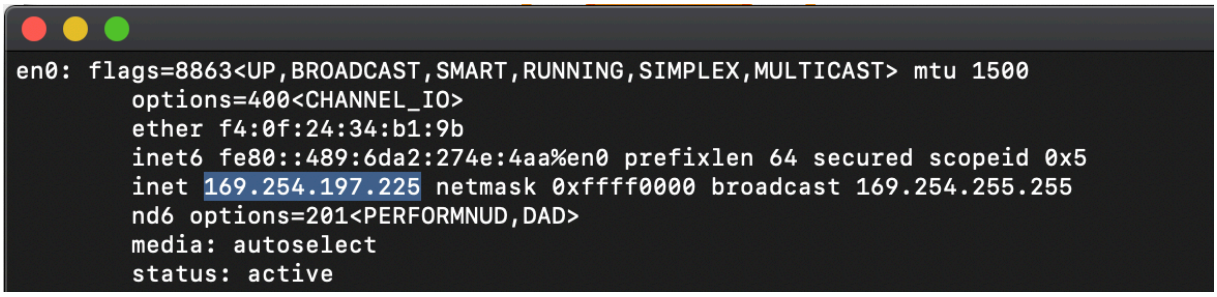
```
$ docker run --name nameContainer -d -e
VILLAGE=Central -e GIN_MODE=release -p 8080:8080 --
restart=always nameImage
```

And then we have the app running in docker in Mac!

8) Connect to the WiFi

Go to the terminal and type:

```
$ ifconfig
```



Look for either “en0" (mainly) or “en1"(rarely), and under that entry look for the "inet" listing. It will be something along the lines of "192.168.1.100". When you find that address, that's what you'll want to put in your browser's URL.

Connect your device to the network you created with your computer, and introduce in the browser the IP found followed by the server port of the app (which is 8080). So, in this example will be:

```
URL: 169.254.197.225:8080/hello
```

And then you will get the login screen in your device

TIP: if you don’t connect and use it immediately after setting up can cause problems, so you just disconnect the Wifi and stop the App, and reconnect the WIFI and start again the app.

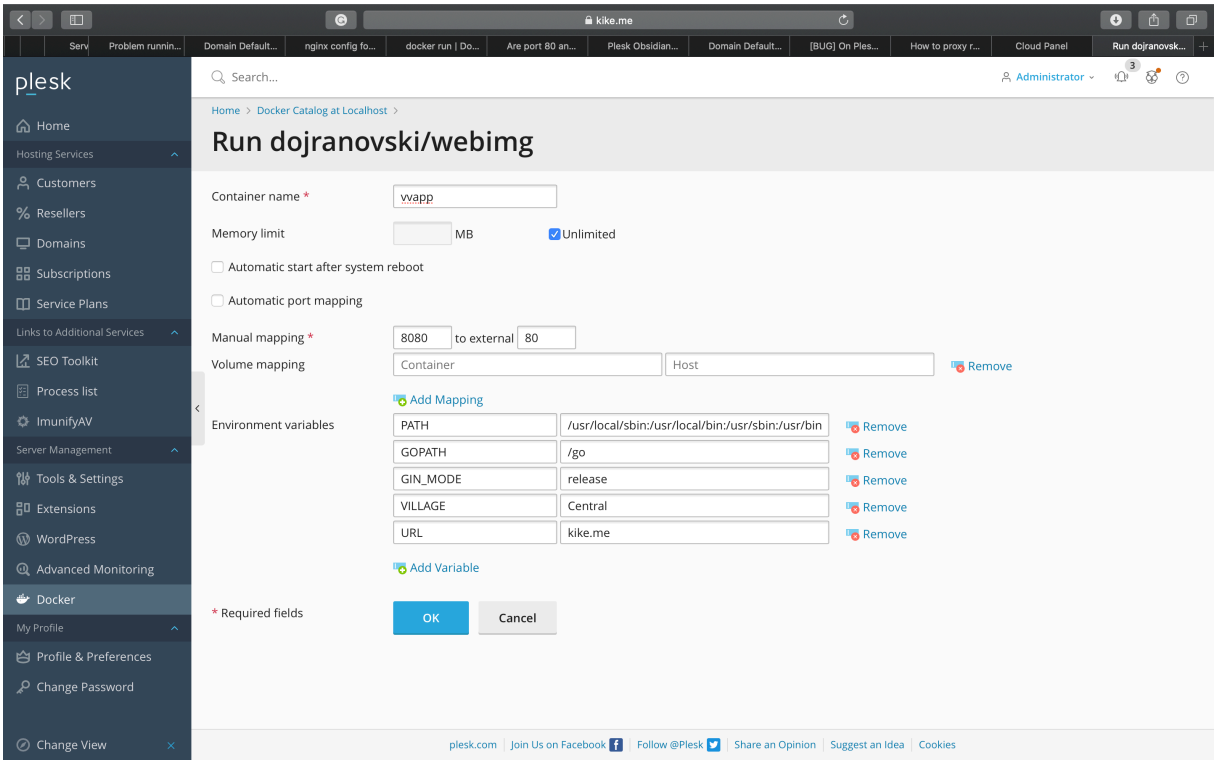
Install VV software in the IONOS server, from ZERO to HERO

Note: we are not promoted by IONOS, we just use hist services.
Here, as everywhere, we are using Docker for the deployment.
You need to have a were server of at least **2GB of RAM** and **15GB of storage**.
In this server, you need to install Plesk software running over a OS (either Ubuntu or CentOS).
Once you have your installation complete and you can access to you Plesk administrator panel, we need to search and install the docker app in extensions.

Once the docker app is running, we need to download the docker image from docker hub.

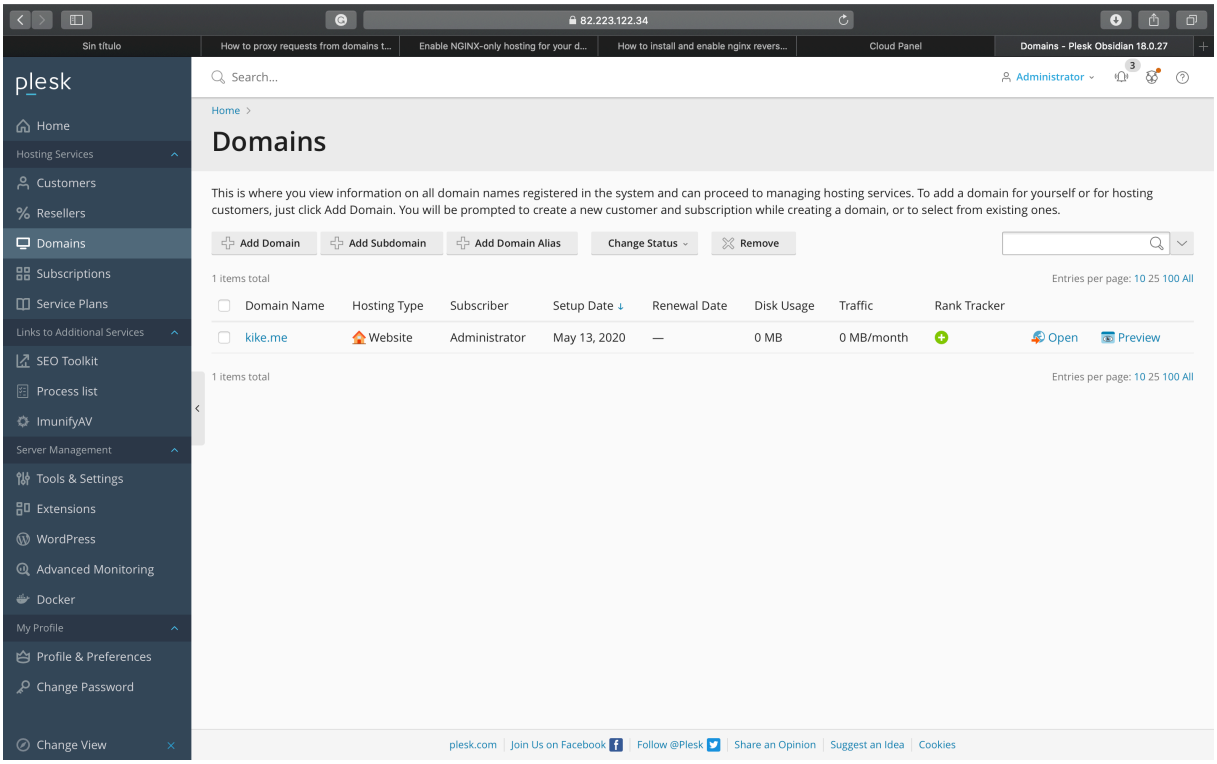
Search the docker hub image, which is “dojranovski/webimg”. Click to “Run”.

To run the container, you need to fill these environmental variables (PATH and GOPATH may appear complete by default).

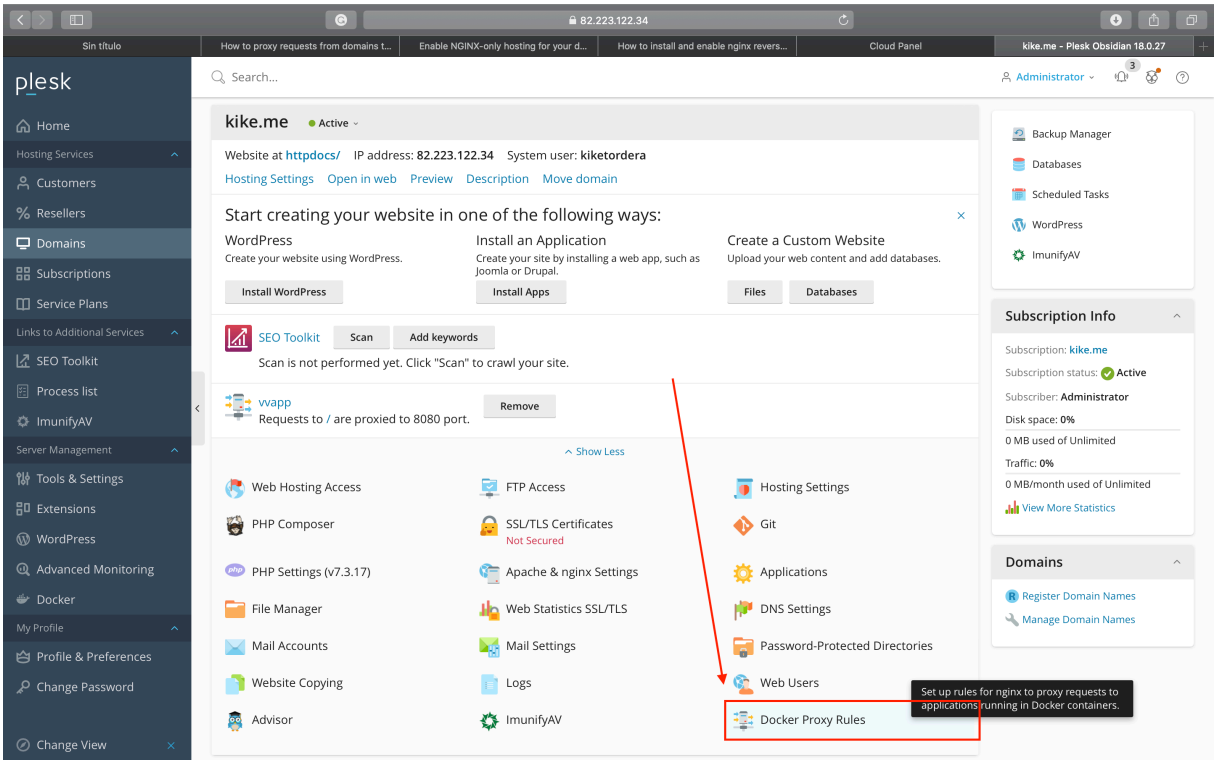


In case of problems:
Note: the VV app is working in the port 8080, and the normal routes in internet (and how Plesk workd internally) works in the port 80, so you need firt to set up a routing from the domain of the website (80) to the port of the docker container (8080).
The **Docker file MUST export** the port 8080.
Source: https://support.plesk.com/hc/en-us/articles/115004844374-How-to-proxy-requests-from-domains-to-Docker-container-in-Plesk-
Possible problem in docker container not display and can not select port: https://support.plesk.com/hc/en-us/articles/115003142213-Not-possible-to-add-Docker-Proxy-rules-no-container-is-displayed

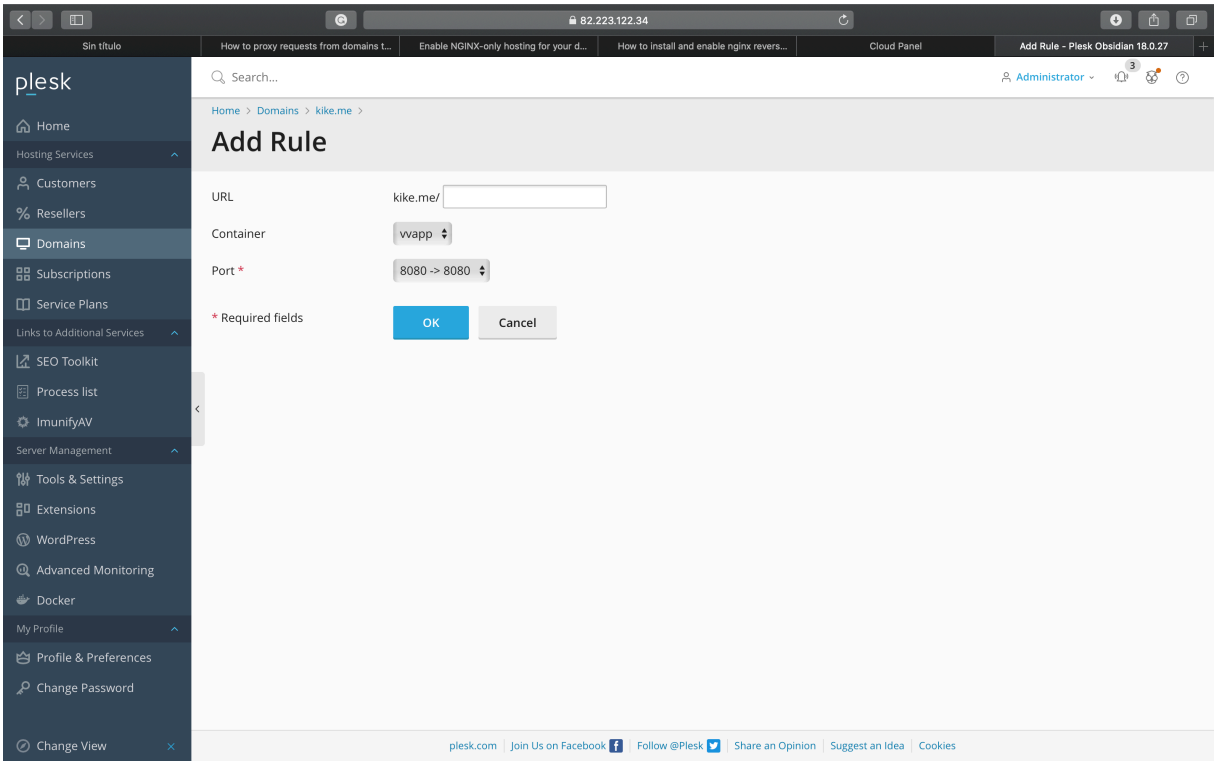
Once the container is running, we need to register our domain in the Plesk application.
If it does now show, click to add and add the domain where we want to display the app.
Once is created, it should appear listed, then we click into it.



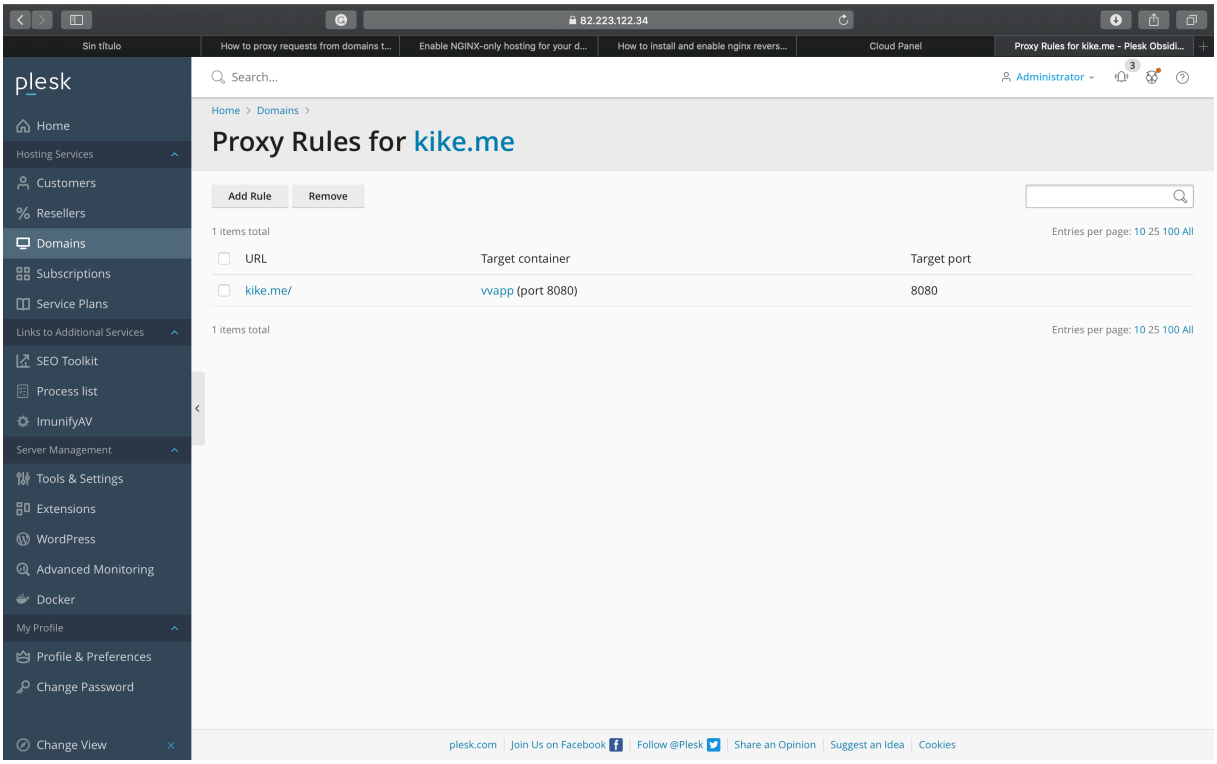
An it will display the manu with all the options. W e select “”.



Now we create a rule and we bind the port 8080 of our app to the port of the server.



Click “OK” and the new rule willl appear listed.



Give the system 10 minutes to refresh and then you will be able to access to the application in the domain selected.

Enjoy!

Update VV software in any environment

Either you are in the cloud, in your computer or in the RaspberryPI, you are running the app in docker, so the steps to update the app in any of these environments will be the same.
We need to do these instructions in order:

Locate the container you have already running

```
$ docker container ls
```

If you have any container running, it will appear in the list with his name, then stop the container and then remove it.

```
$ docker stop nameContainer
$ docker rm nameContainer
```

Now same with images: view if you have any container already running

```
$ docker image ls
```

If you have any image, it will appear in the list with his name, then remove it.

```
$ docker stop nameContainer
$ docker rm nameContainer
```

Now you need to download the latest image from docker hub

```
$ docker pull dojranovski/vvimg
```

For the OS (RaspberryPI, Windows, macOS) it will be a folder in the desktop, for the webserver, will be in /root/folder-in-your-user for the and we linked with the docker container, so the docker container will write the information in the folder we just created.
Then execute this command, is only one long command.

```
$ docker run -d --name nameContainer  -v
pathToFolderInDesktop:/go/src/github.com/
mongolhippie/vv/local-resources -e VILLAGE=Central -e
GIN_MODE=release -e GOPATH=/go -p 8080:8080
--restart=always dojranovski/vvimg
```

Important note: for implementing this on the web-server, you need to specify the image for the web server, it has specific configuration.

For that, simply change the last argument `dojranovski/vvimg` for `dojranovski/imgweb`