Install and configure a Raspberry PI and your working environment for web development

|  |  |  |
| --- | --- | --- |
| **Version** | **Author** | **Revision date** |
| 1.0 | Ian Lachance | 16-10-2016 |
| 1.1 | Ian Lachance | 20-10-2016 |

## Description

The guide will explain how to install and configure a Raspberry Pi using Ubuntu or Ubuntu Server with a non-root user (sudo privileges). Also, it will give you tools to correctly develop and access PI from your working environment (Windows).

## Precision

For the project, the only port that can be access remotely is the port for the SSH connection and for the API. We will use Bitvise to connect to the other ports. We assume that the Raspberry PI is connected to your home network.

## List of used ports on Raspberry PI for the project (RP)

|  |  |
| --- | --- |
| SSH | RP-22 |
| mongodb | RP-27017 |
| nginx | RP-80 |

## Port forwarding on your router (R)

|  |  |
| --- | --- |
| Endless-API | R-8080 to RP-80 |
| Endless-SSH | R-22 to RP-22 |

# Prerequisites

Raspberry Pi with Ubuntu or Ubuntu Server installed.

Working computer/laptop with Windows.

# Raspberry PI

|  |
| --- |
| sudo apt-get update |

## Install and configure mongodb

### Install

|  |
| --- |
| sudo apt-get install mongodb |

### Start/Stop/Restart Commands

|  |
| --- |
| sudo service mongod start  sudo service mongod stop  sudo service mongod restart |

If you encounter this problem -> "MongoDB dbpath (\data\db\) does not exist", while starting mongodb, follow these steps:

|  |
| --- |
| sudo killall -15 mongod  sudo mkdir -p /data/db/  sudo chown -R **<user>**:**<user>** /data/db  sudo mongod |

## Utilisation of mongodb in the console

Access mongodb console and show actual databases:

|  |
| --- |
| mongo  mongo> show dbs |

For other usage, refer to the mongodb cheat sheet in the server documentation (/server/mongodb-cheatsheet.pdf)

## Uninstall mongodb

|  |
| --- |
| sudo service mongod stop  sudo apt-get purge mongodb-org\*  sudo rm -r /var/log/mongodb  sudo rm -r /var/lib/mongodb |

# Working environment

## Bitvise SSH client (Connection to the server)

Install Bitvise SSH Client installer on <https://www.bitvise.com/ssh-client-download> and install it. After opening Bitvise SSH Client for the first time:

1. Write the server IP and the port for the SSH connection in ***Login > Server***.
2. Write the username and the password of your Ubuntu account in ***Login > Authentication***. Make sure the Initial method is set to **password**.
3. For optimal development, go in ***Options > On Login*** and check the **Open Terminal** and **Open SFTP** options. It will open for you a file explorer and a terminal to access and manage your server.
4. Go in ***C2S*** and, for each port not forwarded by your router that you want to use, add a new port forwarding rule (SSH Port fowarding) and enable it. Normally, we use the same port in both interface to avoid confusion and conflicts in our program (if we use the port 27017 for mongoDB, we want to use this port wherever the program is)

|  |  |  |  |
| --- | --- | --- | --- |
| **Listen Interface** | **List. Port** | **Destination Host** | **Dest. Port** |
| 127.0.0.1 | <Port on you PI> | localhost | <Port on your router> |

With Bitvise SSH Client, we see our server as 127.0.0.1, so the SSH Port forwarding works like a VPN (the server think we use the port locally, but in reality, we are connected with Bitvise).

1. Finally, you can login to the server! Each time you will open Bitvise, the only thing you’ll need to do is to click login!

## Pycharm (API development)

## Android Studio (Android development)