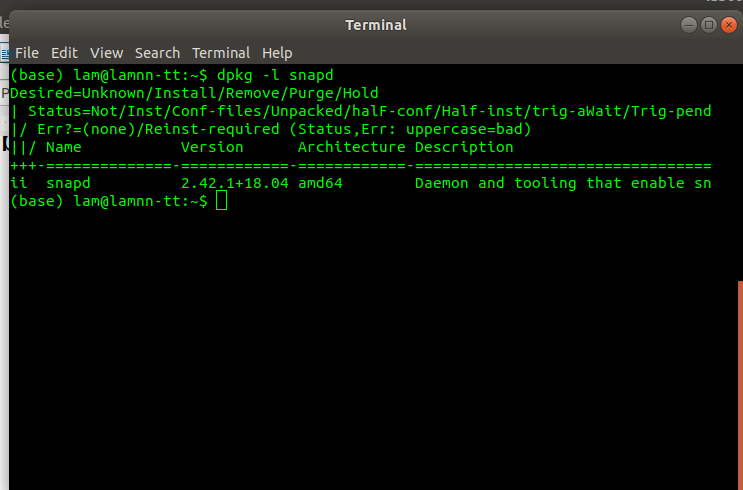
Lab 0-1 – Installation of the environment

1. Install php:
   1. For Windows:
   * I used [XAMPP](https://www.apachefriends.org/index.html) (Apache distribution containing MariaDB, PHP, and Perl)
   * Download installation file for the site above
   * Go through normal procedure of installation (high recommended to just hit next button if you do not specifically know what you want or need)
   * Find your Advanced system settings link
   * Click Environment Variables. In the section System Variables, find the PATH environment variable and select it. Click Edit. If the PATH environment variable does not exist, click New
   * Added to path variable: path\_to\_your\_xampp\_installation/php (for example C:\xampp\php)
   1. For Unix (Ubuntu 18.04): Run the following lines on terminal
   * sudo apt-get update
   * sudo apt -y install software-properties-common
   * sudo add-apt-repository ppa:ondrej/php
   * sudo apt-get update
   * sudo apt-get install -y php7.4 {bcmath,bz2,intl,gd,mbstring,mysql,zip}
   * sudo apt-get install php-cgi
2. Install phpstorm:
   1. Preparation:
   * Using the student email of the university, create a [jetbrains](https://www.jetbrains.com/) account
   * Apply for student license (educational license)
   * You should receive an email to confirm your “purchase” and be able to use the program for 1 year.
   1. Actual installation:
   * For Windows:
     + Download installation file from [PHPStorm for Windows](https://www.jetbrains.com/phpstorm/download/#section=windows) for your OS
     + Open said file, to start the installation process
     + If your computer has an anti-virus software, sometime it will pop a warning up, click okay
     + Do not install on program file or program file (x86) folder if you did not have administration rights
     + After completed the installation, open PHPStorm, read the end-user agreement (recommended), update references and user settings.
     + Enter your key (get from preparation steps) or login to your [jetbrains](https://www.jetbrains.com/) account to finish your installation. You should be able to use it now
   * For Unix(Ubuntu 18.04):
     + Since snappy is supported on ubuntu from 16.04 (Xenial Xerus), I do not need to install snappy. But if you do not have snappy on your machine yet (you can check if installed or not by using the following command: dpkg -l snapd if you have it already it should return the following message,

if not you can install it by: sudo apt update&&sudo apt install snapd)

* + - Install PHPStorm by: sudo snap install phpstorm –classic
    - Follow the last 2 steps of windows guide part to finish your installation

1. Install Xdebug:
   * Intsall php-pear: wget <http://pear.php.net/go-pear.phar> and php go-pear.phar
   * Run: pecl install xdebug
   * Put these following line to /*etc*/php/7.4/cli/php.ini:
     + [xdebug]
     + zend\_extension="<path to xdebug extension>"
     + xdebug.remote\_enable=1
     + xdebug.remote\_port="<the port for Xdebug to listen to>" (the default port is 9000)
   * Configure it in PHPStorm (port number, allowing incoming transmit)
2. Install Docker:
3. On Windows:
   * Download Docker Desktop
   * Run installation file (highly recommended to not changing the default settings if you do not know what you are doing)
4. On Ubuntu: Run the following line (${USER} is your username on your computer)
   * sudo apt update
   * sudo apt install apt-transport-https ca-certificates curl software-properties-common
   * curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
   * sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
   * sudo apt update
   * apt-cache policy docker-ce
   * sudo apt install docker-ce
   * sudo usermod -aG docker ${USER}
   * su - ${USER} (prompt password input)
5. Setup with phpStorm (on all platform)

* create a new file named docker-compose.yml to define an environment with something like:

version: '2'

services:

webserver:

image: phpstorm/php-71-apache-xdebug-26

ports:

- "80:80"

volumes:

- ./:/var/www/html

environment:

XDEBUG\_CONFIG: remote\_host=host.docker.internal

* Right-click docker-compose.yml and select Create… from the context menu to create a run/debug configuration
* Use can now run and debug code