

Setting an Environment Variable

([click here for detailed description](#))

We will demonstrate it on the PATH example. If you have a program in a custom location, adding it to \$PATH permanently and be able to run it under all circumstances is not an easy task.

1. Create a .sh file in /etc/profile.d.
2. the file should contain: `export PATH=$PATH:YOURPATH`
3. logout and login again to load the new added variables

Enable Variable with sudo

To enable the variable even if you use sudo, you need to edit sudo config using `sudo visudo` and:

1. exclude PATH from variable being reset when running sudo:
 - a. `Defaults env_keep += "PATH"`
2. disable the safe path mechanism for sudo, i.e., comment the line:
 - a. `Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin`
3. logout and login again to load the new added variables

Enable the Variable Outside Bash

If you need the variable outside bash, the above-mentioned approach won't work. Currently, I do not know about any general solution for variables. The solution below, unfortunately, **work only for PAM shells** (see [this SO answer](#) why).

Add the variable to /etc/environment. Note that it is not a script file, so you can use only simple variable assignments.

Enable Variable on a Device Without Root Access

Without root access, we can only edit the user config files. put the necessary config into:

- ~/.bash_profile if it already exists
- or to ~/.profile

Note that the .profile file is ignored when the .bash_profile file exists.

Enabling SSH Access to Server

1. install openssh
 - a. `sudo apt update`
 - b. `sudo apt install openssh-server`
2. configure access

- a. password:
 - i. open `/etc/ssh/sshd_config`
 - ii. set `PasswordAuthentication yes`
 - iii. Now you can log in with the user and password you use in Ubuntu
- b. keys: TODO

WSL configuration

- 1. port can be used on Windows, so change port to 2222 in `sshd_config`
- 2. when logging from Windows use 127.0.0.1 as a host

Debugging

- 1. check the ssh status with: `service ssh status`
- 2. check the ssh port with `sudo netstat -tln`

WSL

- 3. Windows firewall?

Printing used ports

`sudo lsof -i -P`

It does not work without sudo!