Welcome to RES 2015

RES, Lecture 00

Olivier Liechti
Simon Oulevay
Laurent Prévost

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Agenda

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Background**
  - Who are we?
  - What do we do at the HEIG-VD and why?
- **Course objectives**
  - Network programming
  - Application-level protocols
  - Web infrastructure
- **Tools**
  - Crash course on Git & Github

Background

Personal background

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Personal background

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

R&D at HEIG-VD

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Startups

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

NETGUARDIANS
Risk Under Control

sysmosoft

OBJECTIS

STIGNERGY
Energy Control & Optimization

novaccess

SimplicityBio
IN SILICO
BIOMARKER DISCOVERY

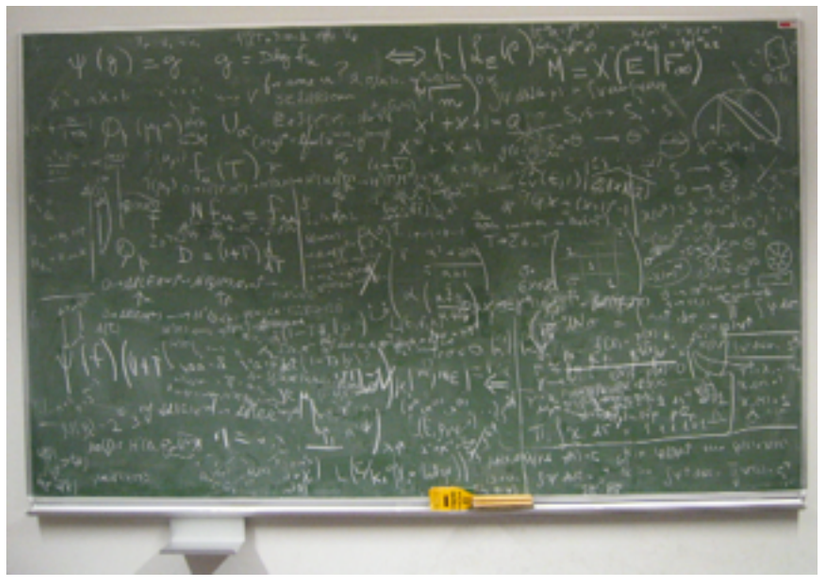
Future instruments

eBiz
Games

VIACAM
vision for safety

Why do we do applied research?

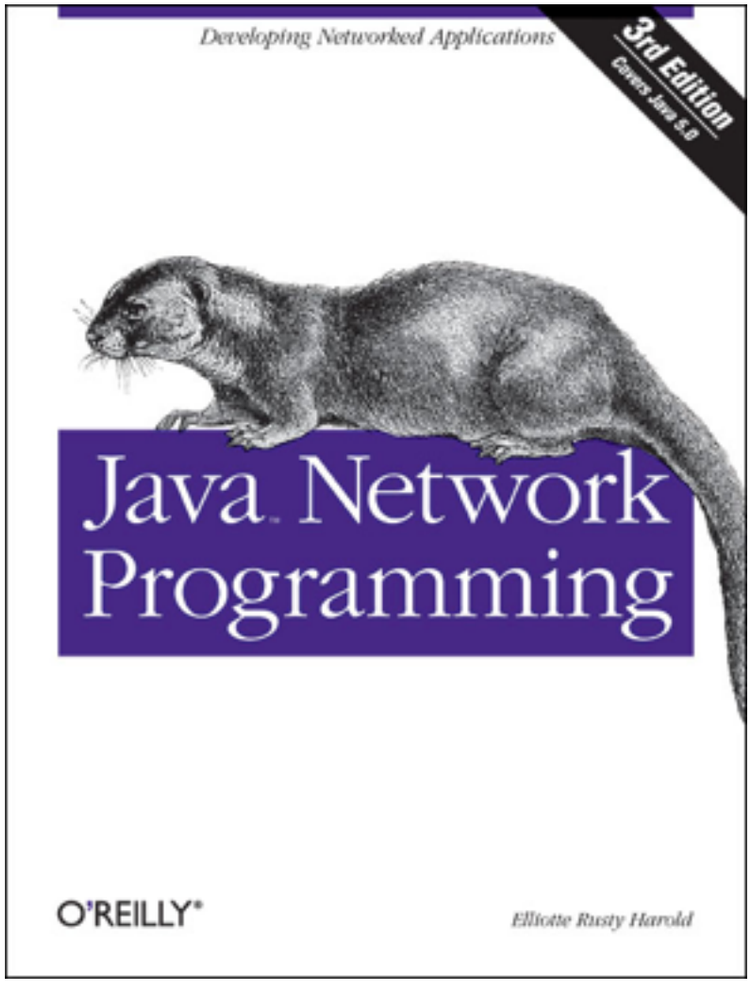
heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

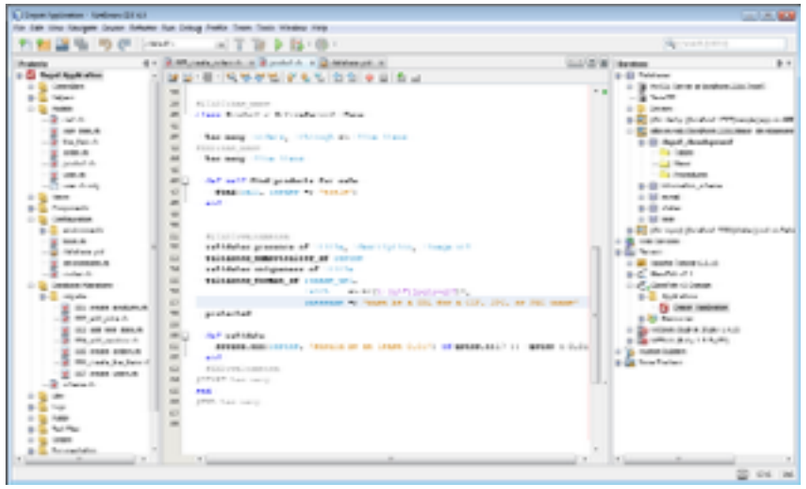


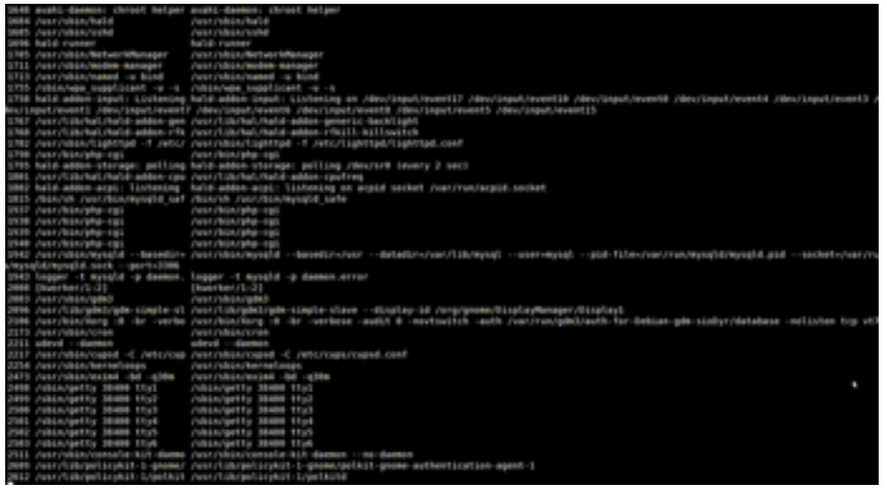
Culture of Sharing

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

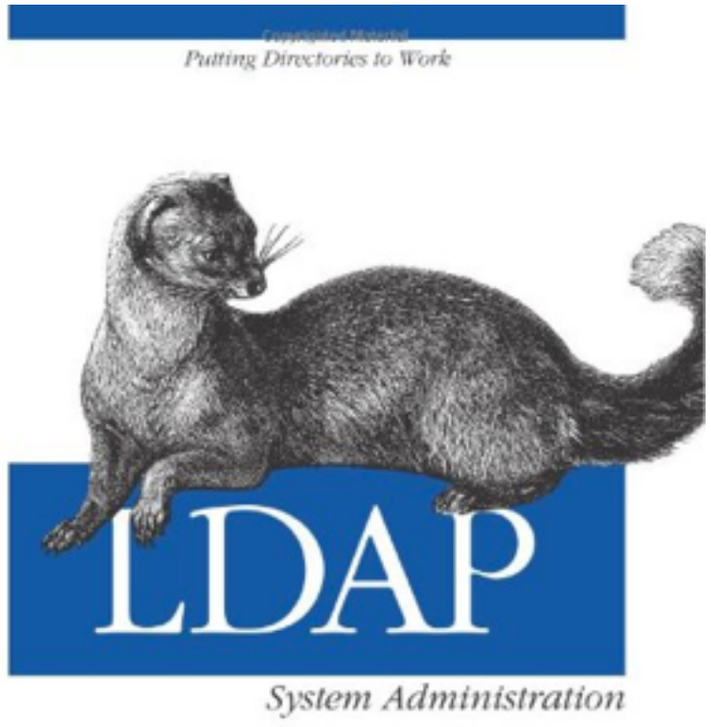
Course Objectives


Developing Networked Applications
3rd Edition
Covers Java 6.0
Java Network Programming
O'REILLY
Elliotte Rusty Harold






NGINX
Nginx as Reverse proxy for APACHE
APACHE


Putting Directories to Work
LDAP
System Administration


O'REILLY
Gerald Carter
Copyrighted Material
VAGRANT
docker

RES

2 + 2

TWEB

3 + 3

IL + TS

AMT

3 + 3

IL + TR + TS

Cyberlearn

Hes·so-CENTRE E-LEARNING

http://cyberlearn.hes-so.ch/course/view.php?id=**6971**

Semaine 1 (16 février)

Le premier objectif de la première semaine est de présenter les objectifs du cours, d'expliquer comment il prépare à deux cours conséquents de 3ème année (AMT et TWEB) et de faire un survol rapide de la matière.

Le deuxième objectif est de nous familiariser avec les outils et les processus que nous allons utiliser dans beaucoup de labos. Nous allons nous appuyer sur git et GitHub et sur des outils de génie logiciel. Il est donc important de comprendre à quoi servent ces outils et comment nous allons les utiliser.

Connaissances actuelles et attentes

Ce questionnaire va nous aider à mieux comprendre l'état de vos connaissances actuelles et d'identifier les attentes particulières que vous avez par rapport au cours.

15 minutes

Tools

Git & Github

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Step 1: install Git**
  - http://git-scm.com/downloads
  - http://git-scm.com/book/en/Getting-Started-Installing-Git
    - Check point: are you able to invoke the git command from the shell?
- **Step 2: configure Git**
- https://help.github.com/articles/set-up-git
- Step 3: configure SSH
  - http://guides.beanstalkapp.com/version-control/git-on-windows.html#installing-ssh-keys

Using Git locally

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

```
$ mkdir my-project
$ cd my-project
$ git init
$ ls -al
```

- **You do not *have to* use a server**: Git is already useful to manage versions of your files on your local machine.
- The **`git init` command** creates a **local repository**. If you look carefully, you will see **a hidden `.git` directory**, where Git keeps all of his data.
- **Important**: your **`my-project`** directory is your **working directory**. If you simply create files in it, they will not immediately be part of your repository!

Using Git locally

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

```
$ echo "text a" > a.txt
$ git status
$ git add a.txt
$ git commit -m “First version of a.txt”
$ echo “my mod on text a” > a.txt
$ git status
```

- A **commit** is a **snapshot** of your repository. Git maintains a **graph of commits** and you can always **recover the state** of a particular commit.
- When **you have modified files in your working directory**, you need to specify which ones should be **part of the next commit**.
- You use the **`git add`** command to add a file to the so-called **staging area**. It will be part of the next commit.
- You use the **`git status`** command to **check the content** of your working directory and of your staging area.

More info: http://git-scm.com/book/en/Git-Basics-Recording-Changes-to-the-Repository

Working Dir, Staging Area & Repository

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

This is a local directory, not a remote server!

`git add` is used for both

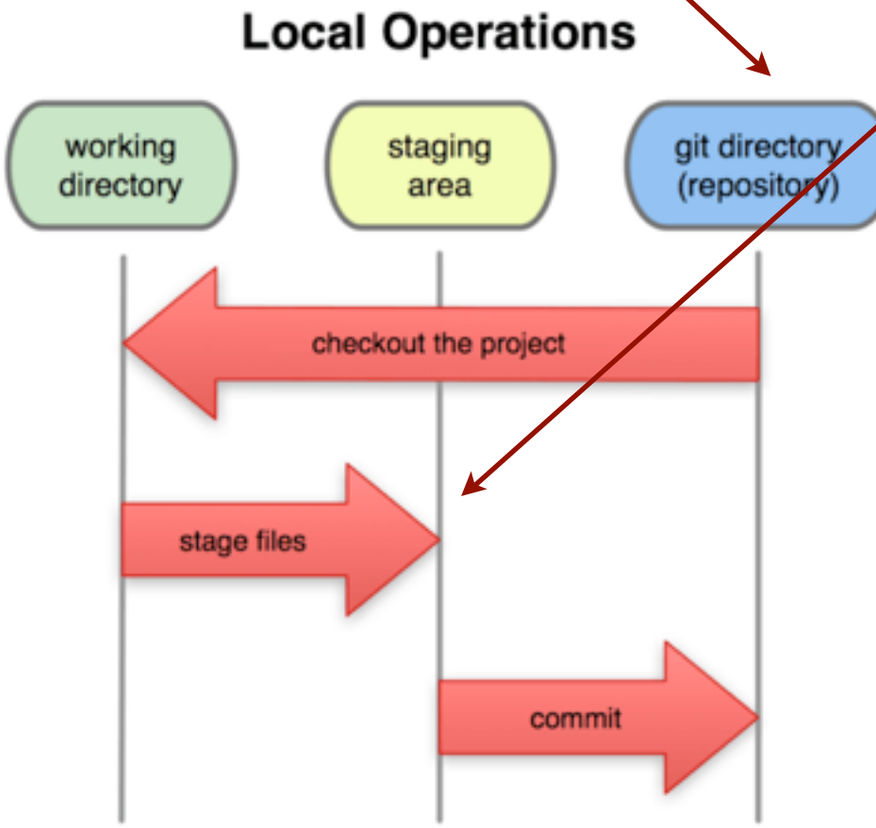


Figure 1-6. Working directory, staging area, and git directory.

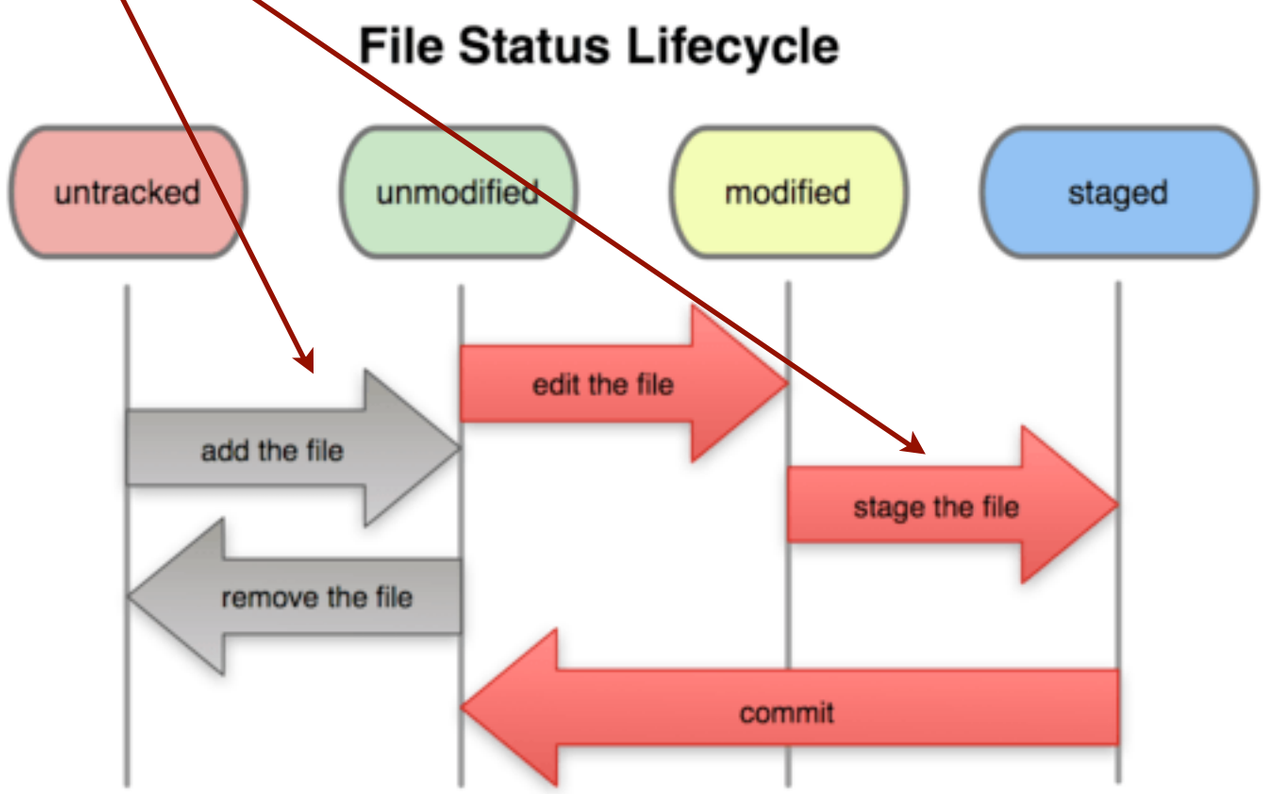
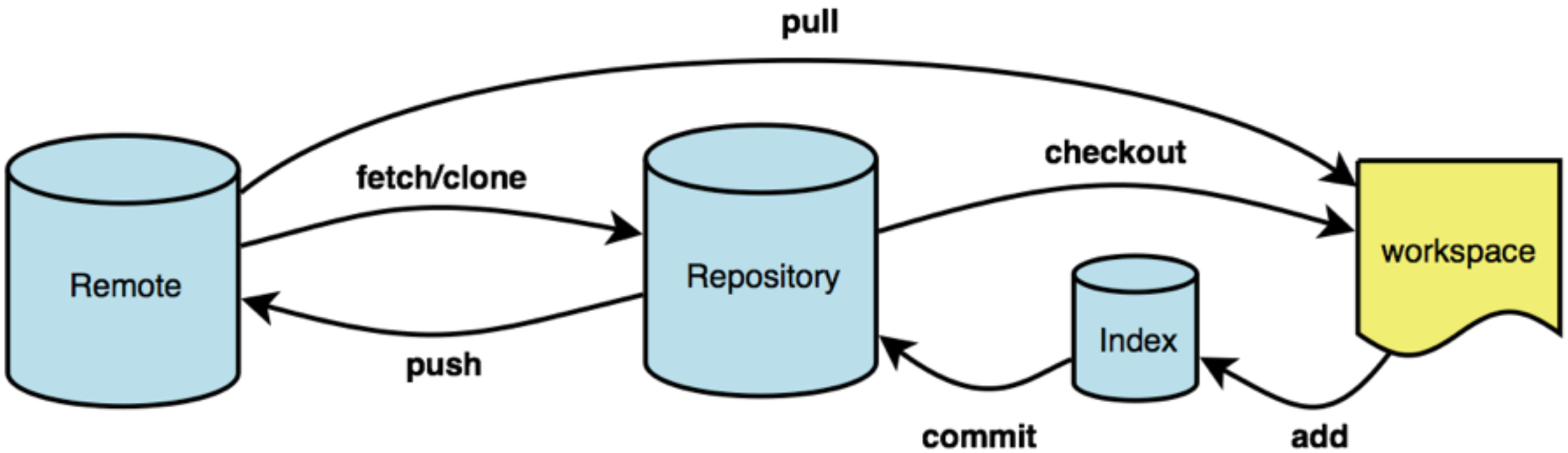


Figure 2-1. The lifecycle of the status of your files.

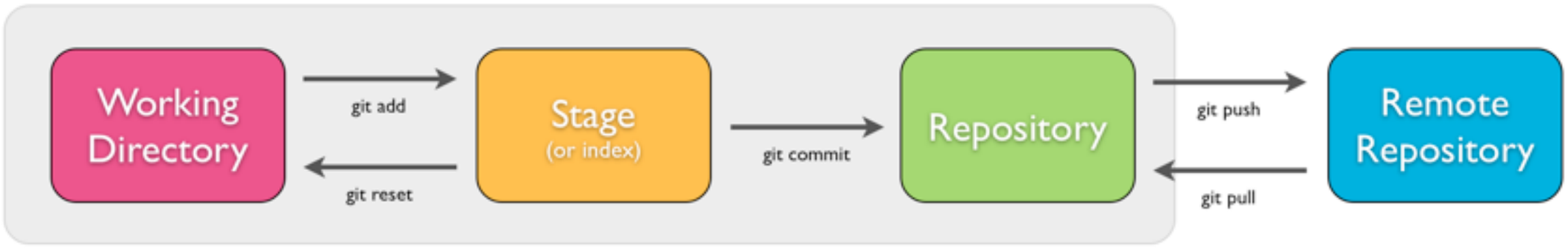
Source: http://git-scm.com/book/en/Getting-Started-Git-Basics

Git & Remote Repositories

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



Source: http://illustrated-git.readthedocs.org/en/latest/

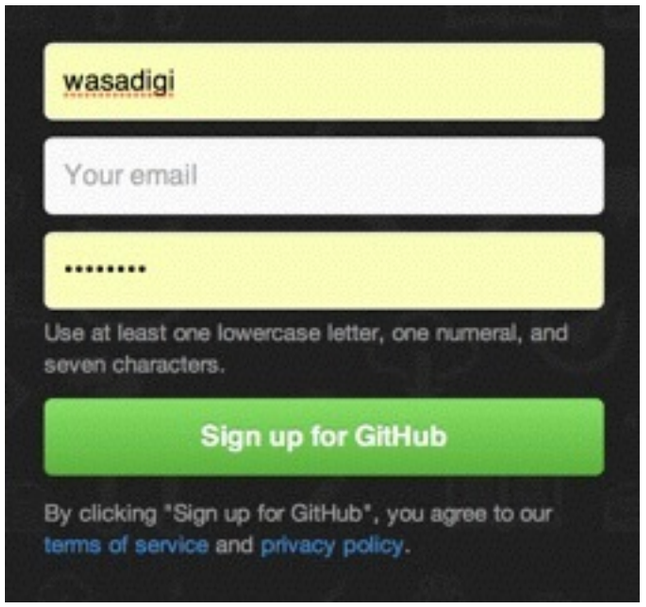


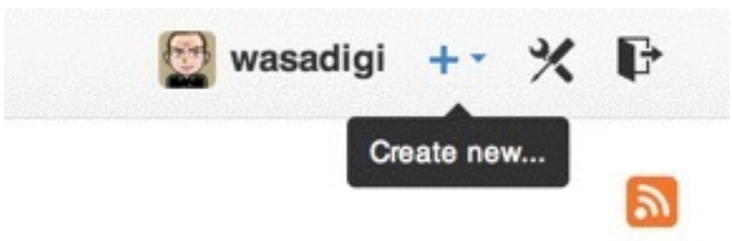
Source: http://bramus.github.io/ws2-sws-course-materials/xx.git.html#/4/1

Github Setup

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Sign up** for GitHub and get your own account:
  - Go to http://www.github.com
- Add your **SSH key**:
  - Go to your accounts settings. You will find an option to manage your SSH keys.
  - If you don't have a SSH key yet, follow the instructions in the online help.
  - If you are using windows, you will need to use Git BASH.
- **Create** your first repo, hosted on Github.
- **Copy the SSH URL** of the repo.

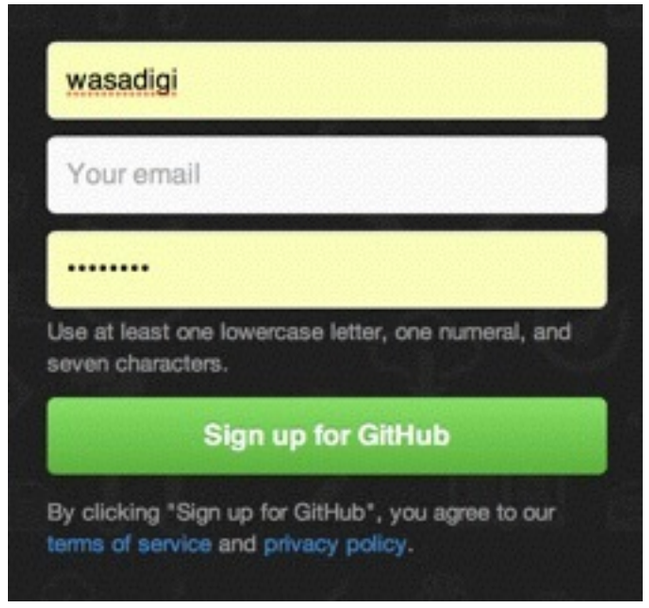


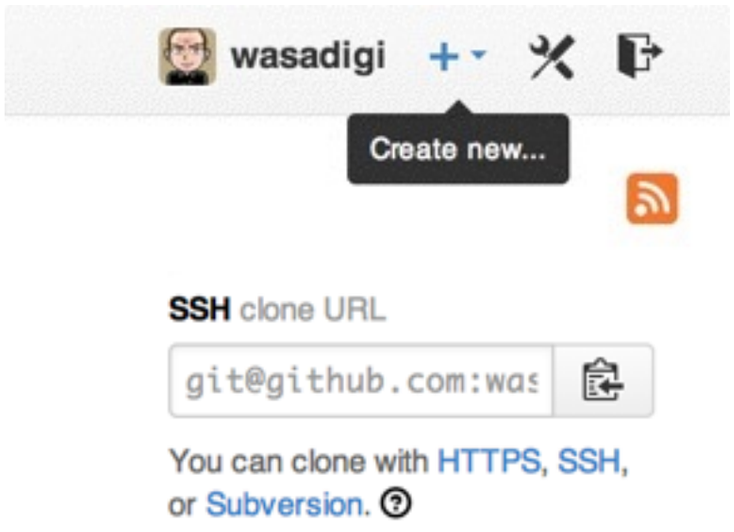


Validate your setup

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

- **Clone your repo to your laptop**
  - Open a **terminal window** (Terminal on Mac OS, Git BASH on Windows, etc.)
  - Create a **new directory** to host your clone of the repo and get into it.
  - **Clone** the repo, using the **SSH URL**.
  - **Create** a file, **add** it to the staging area, **commit** the changes and finally **push** the commit Github.



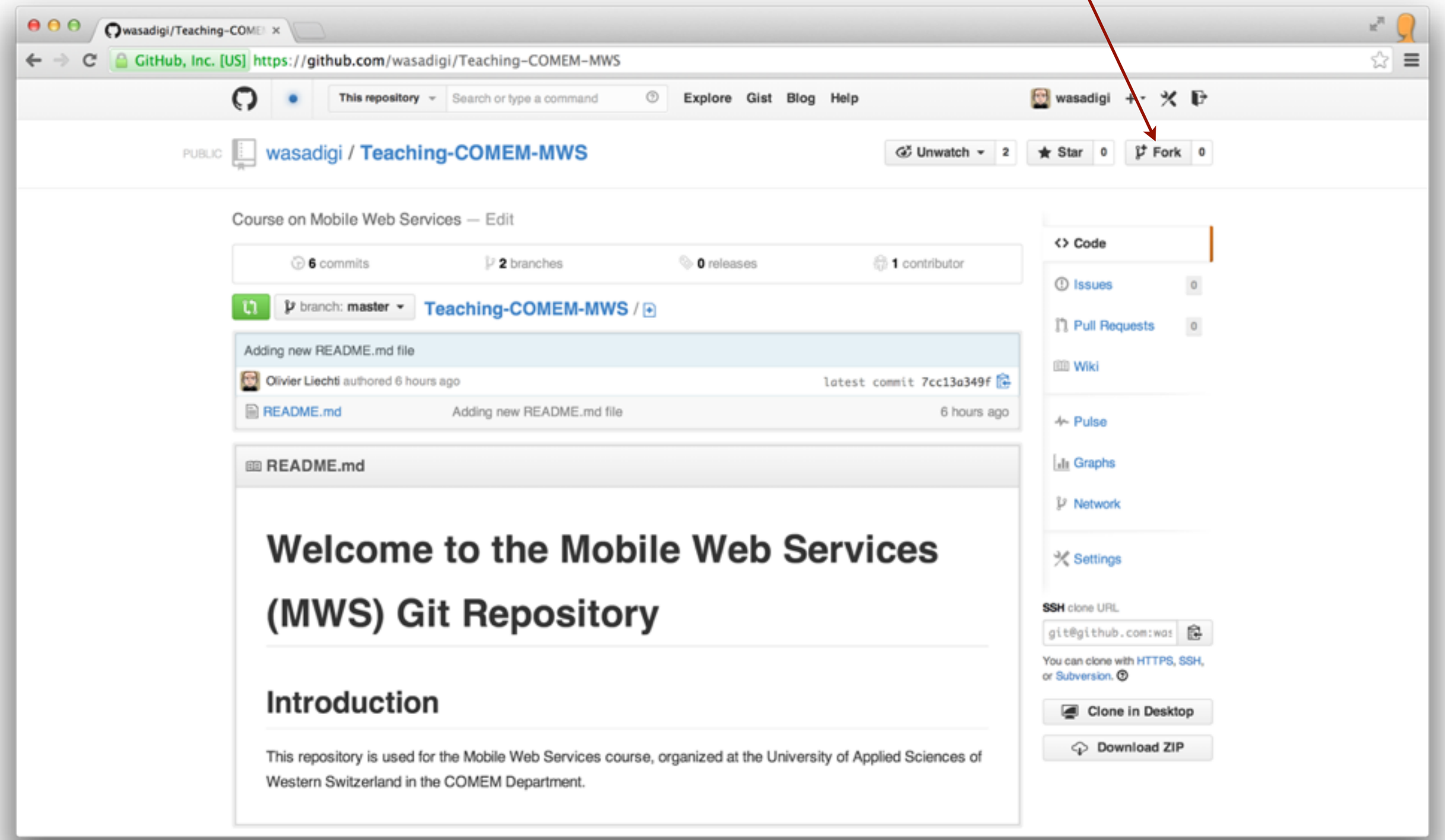



```
$ mkdir myspace
$ cd myspace
$ git clone git@github.com:UUUUU/RRRRR.git
$ git touch firstFile.txt
$ git add firstFile.txt
$ git commit -m “I have added my first file”
$ git push
```

Forks & Clones

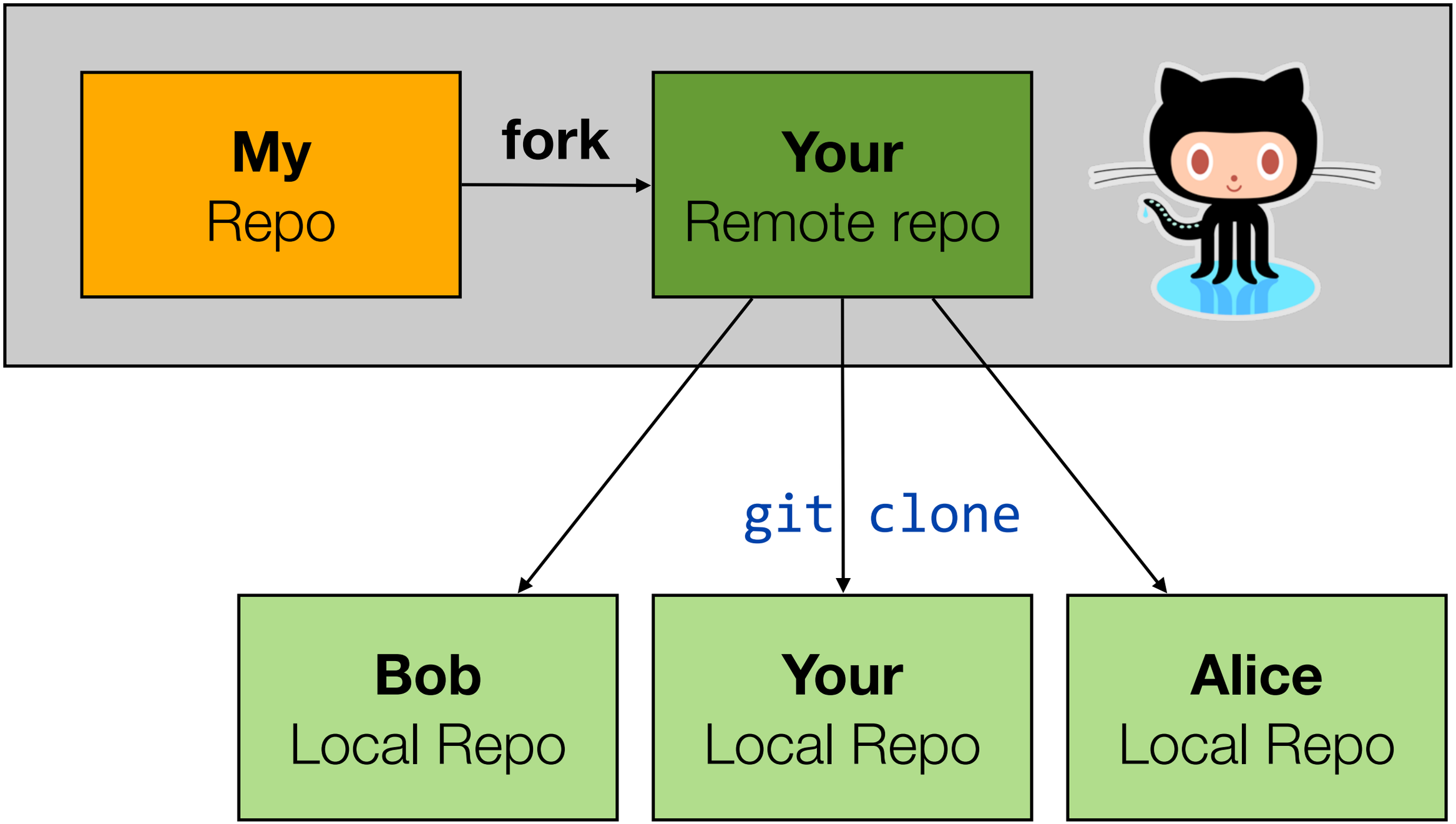
If you do this, you will have YOUR clone of MY repo hosted on Github

heig-vd
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud



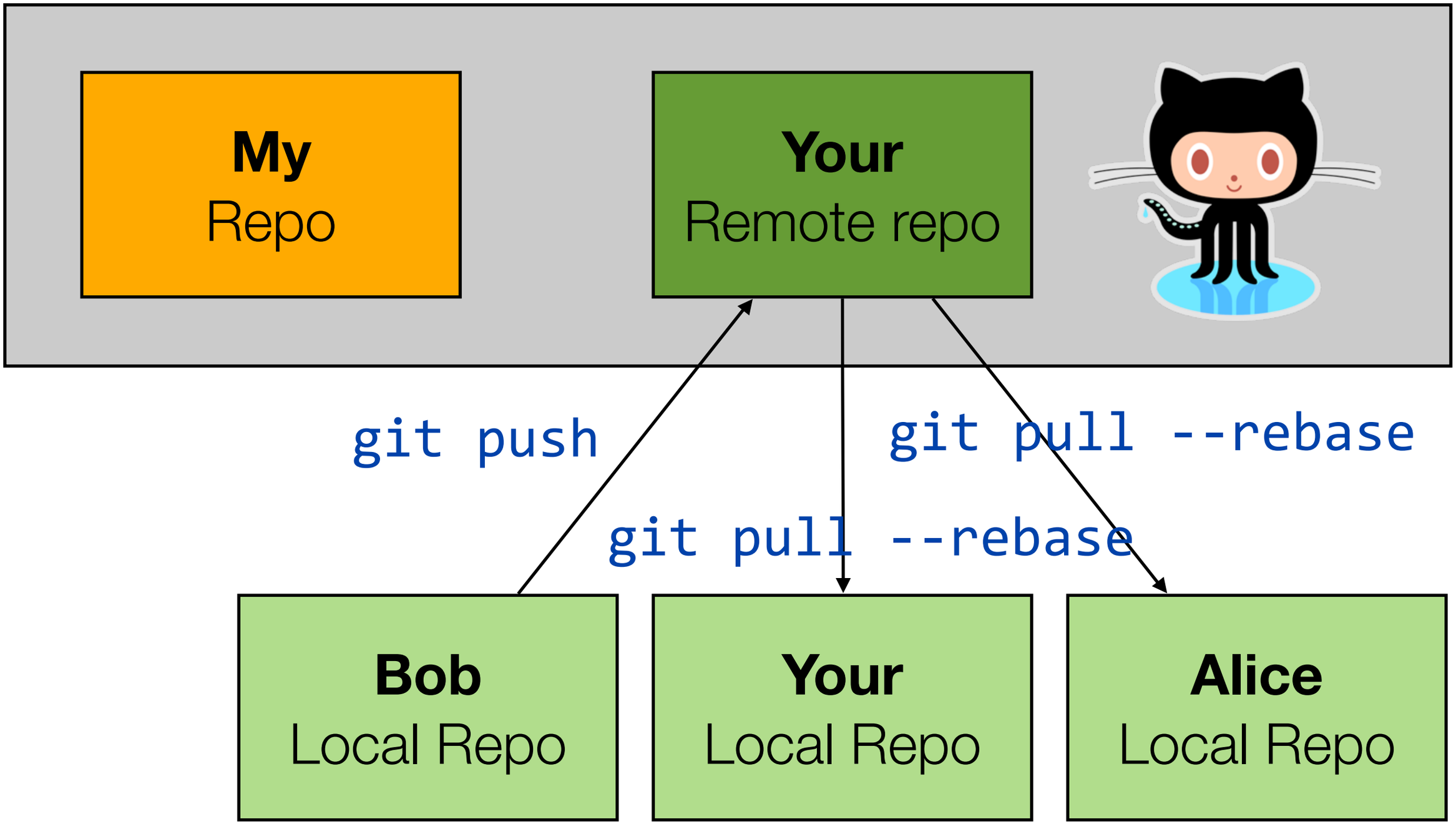
Forking My Repo on Github

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



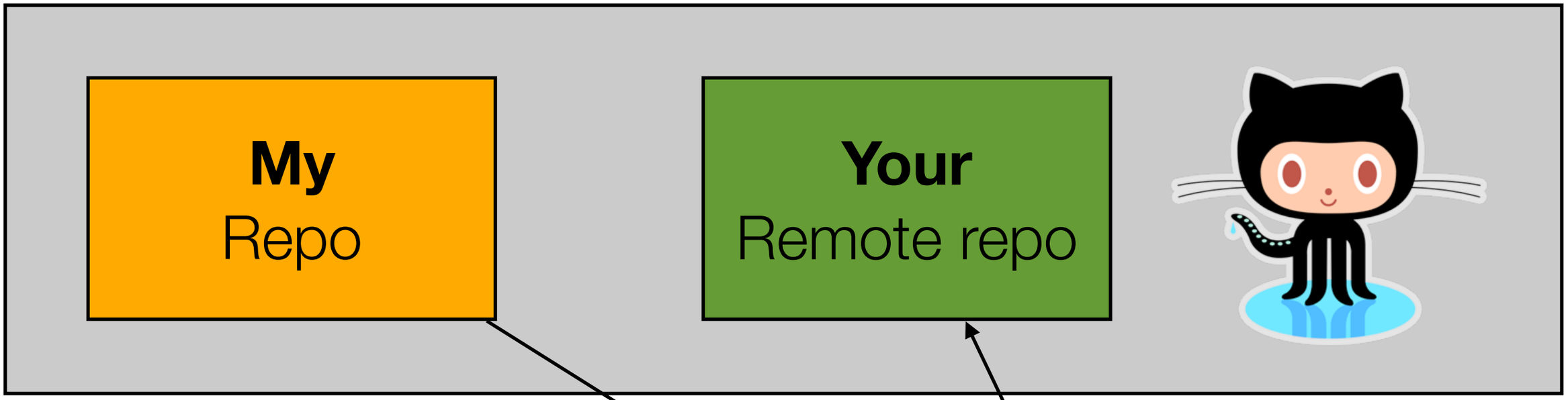
Forking My Repo on Github

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud



How Do Will You Get **My** Updates?

heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud




```
# configuration, do it once
git remote add upstream git://github.com/UU/RR.git
git remote -v
```

`git fetch upstream`
`git rebase upstream/master`

Lab Introduction