Access to NSRC materials

November 10, 2013

Objectives

At the end of this session the participants should be able to:

- List common ways to distribute materials
- List issues with common ways of distributing materials
- Download training materials to their machines
- Download deployment materials to their Mac machines.

Collaboration and Version Control

There are lots of ways we distribute data. When we are collaborating on creating that data, we need to be able to follow different people's edits to the documents (labs, presentations) until we have a final version. Later we also need to be able to update with a newer vserion of the materials when some new information is out.

- What methods are commonly used to collaborate on creating data?
- What methods are commonly used to manage new versions?

Methods

Collaboration

Many options exist for this that are in use namely:

- "Track changes" (in say Word, Excel)
- Online office productivity tools like Google Docs
- A secretary manually merging documents.

Version Control

- Online services like Google Docs have this built-in
- Using a fixed naming scheme for files with versions/dates in the name
- Replacing the files with the "latest version" from an appointed cordinator/secretary.

What issues exist with most traditional ways to manage this problem?

Automated Systems

These have existed for a while. Internally, online services like Google Docs use similar technologies. In general these systems are referred to as SCMs - Source Control Management systems, or VCS (Version Control Systems)

Some examples include:

- cvs
- subversion written to replace CVS above
- mercurial

We use git and have locally installed an interface to help us manage our git repositories.

Git terminology

- Repositories are the top most collection of related files and folders.
- Cloning is the initial copying of a repository to your machine
- Pulling is updating your copy with new changes from whatever source you got the repository from
- Committing is creating a version that specifies changes to certain files and/or folders
- Pushing is something we won't do in class but is how you push the commits you just made in the previous step to where you got the repository from.

Accessing our repositories

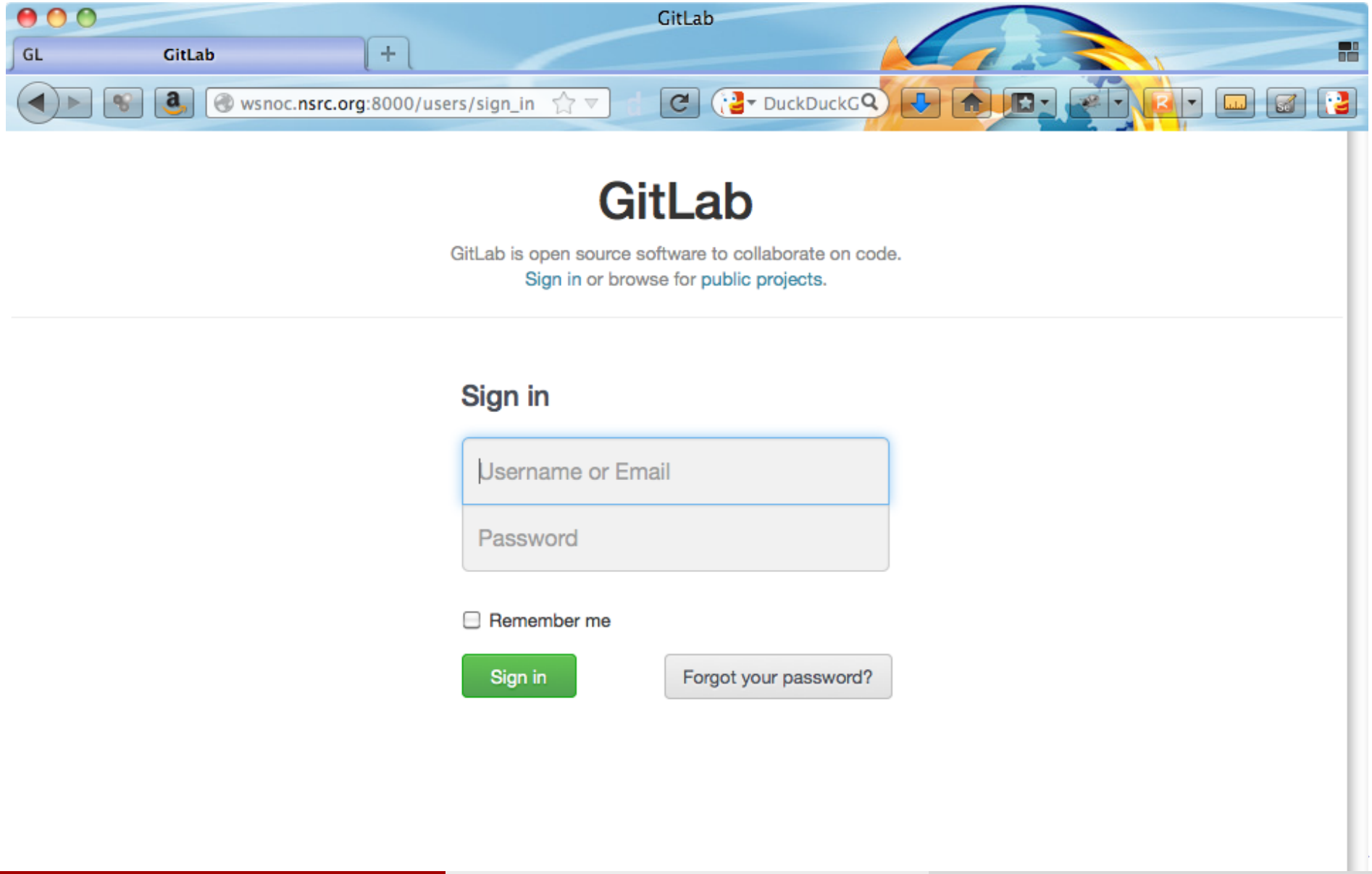
Licensing

Accessing our repositories

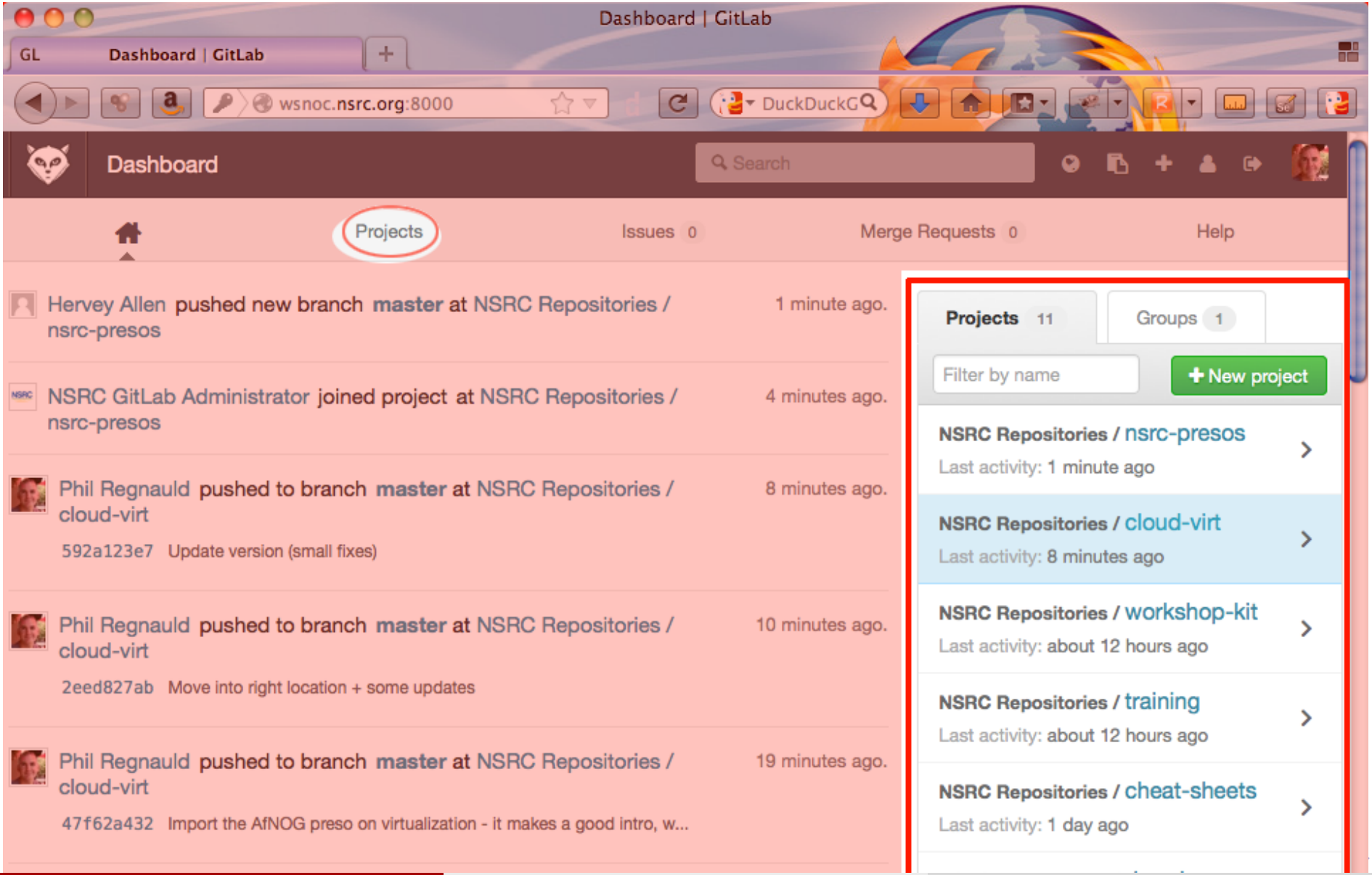
HTTP access

We have a local clone of our repositories at http://wsnoc.nsrc.org:8000/ Using gitlab. This is not publicaly available so you need a username and password that will be availed in class. You can download zip files of entire repositories to your machine using a web browser.
The disadvantage is each time there's a change say during the week you'd have to download the entire zip and re-uncompress it. Also you don't have the ability to swap between versions.

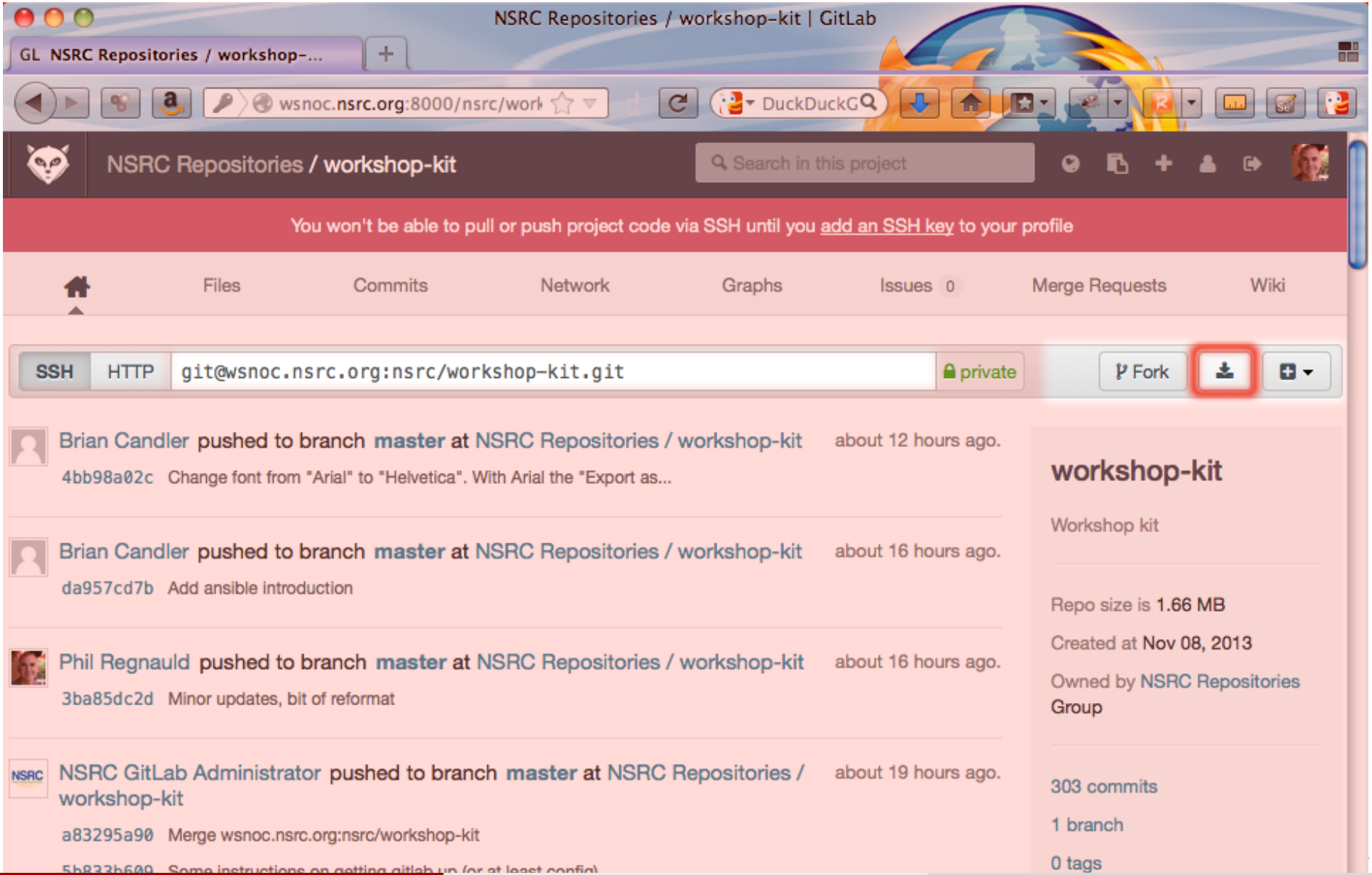
Accessing our repositories



Accessing our repositories



Accessing our repositories



Accessing our repositories

Git access

You can use a git client to clone a repository to your machine. UNIX machines can install command-line git clients and many pre-install them (e.g OS X) does. The next session on ansible shows how to do this with Ubuntu on your Mac minis.

On your laptops you can either install the command line utilities or use graphical tools like SourceTree which has free OS X and Windows versions. We have put installers for this on the wiki under the Agenda page.

Once you have cloned a repository you can just "pull" to update it to the latest version.

You can also make commits and push if you have write access, but for the time being we have given you read only access to our repositories.

Accessing our repositories

Git access

Once you have your git client you can clone a repository. You will have to know the URI to use to clone the repo. That URI is listed in the website so you can copy and paste it. You can clone using the HTTP(S) or SSH [a]. We have used HTTP in this class for simplicity but remember it will send your password in clear text. If you install gitlab on your own please add an SSL cert and use HTTPS.
The actual method depends on the client. For the GUI just click on add new repo. For the command line clients you will use something similar to the following.

```
$ git clone http://wsnoc.nsrc.org:8000/nsrc/workshop-kit.git
$ cd workshop-kit
```

[a] To use SSH you need to have uploaded an SSH key to the account distributed in class

Accessing our repositories

Long term access

We will be using Gitlab and will distribute during the week details and credentials (username & password) on how to access the latest versions of our materials. You can use either method (Web or using Git) to either download the latest zip or pull to update your local copy.

A word on file formats

We try to stick to open standards when developing materials

We primarily use two file formats

- Presentations: ODP[a]
  - Open/LibreOffice's presentation format (PowerPoint-like) Works across all platforms (Linux, Windows, OS X) Easier to store/version/compare in a VCS like Git
- Worksheets for labs: MarkDown
  - Actually just plain text with some simple conventions for formatting (lists/bullets, indentation, section headers, links, text styling). If you want to build our content from markdown we use the pandoc variant from http://johnmacfarlane.net/pandoc/ which also requires that you have LaTeX http://johnmacfarlane.net/pandoc/[b]

[a] We also use PPT (PowerPoint) but we're trying to stay away from it.

[b] Mac TeX for OS X http://www.tug.org/mactex/

A word on file formats (cont'd)

Presentations can also be written in MarkDown (this one is)

For example
bold text and *italic* and clickable link
written as

```
**bold** text and *italic* and
clickable [link](http://eu.org/)
```

New slide:

```
# Slide title

## Heading

* Item
```

Slide title

Heading

- Item

. . . and some text.

Future

Automation

In the coming weeks/months, we'll build tools to automatically convert presentations and worksheets to PDF/HTML, and place them online for trainers to dowload - then you won't have to create the formatted versions yourself (unless you want to make changes, of course)