ZICE 17 On-boarding

Luca Mazzone & Simon Scheidegger

Jan. 24th, 2017

ZICE 17 – University of Zürich

alphacruncher

Outline

1. The compute infrastructure for ZICE
2. Accessing WLAN during ZICE
3. Access to ALPHACRUNCHER Services
4. Course management – git Classroom
5. First steps on a Linux Cluster

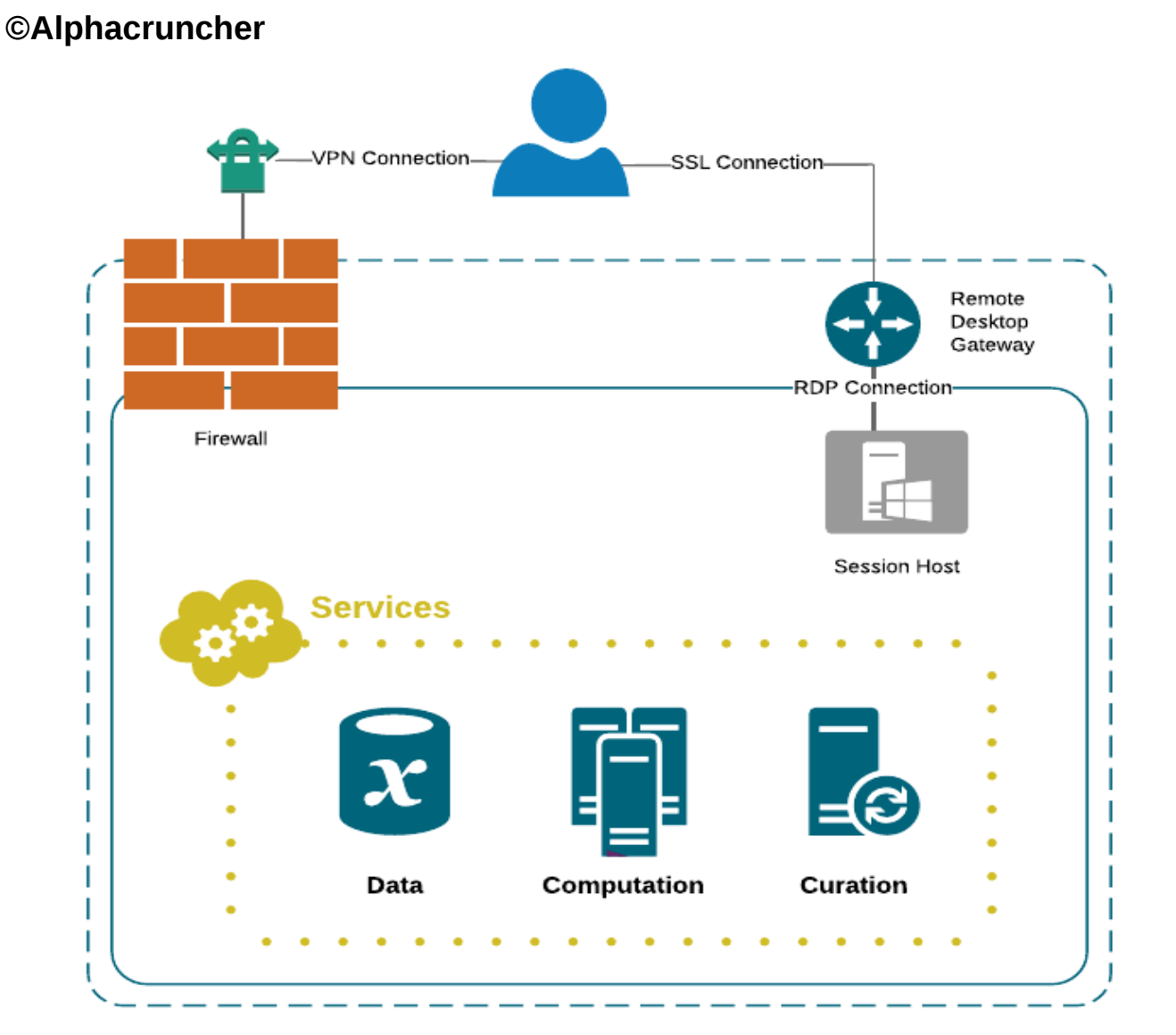
Aim

→ Every student works on a unified environment (no issues with licenses etc.)

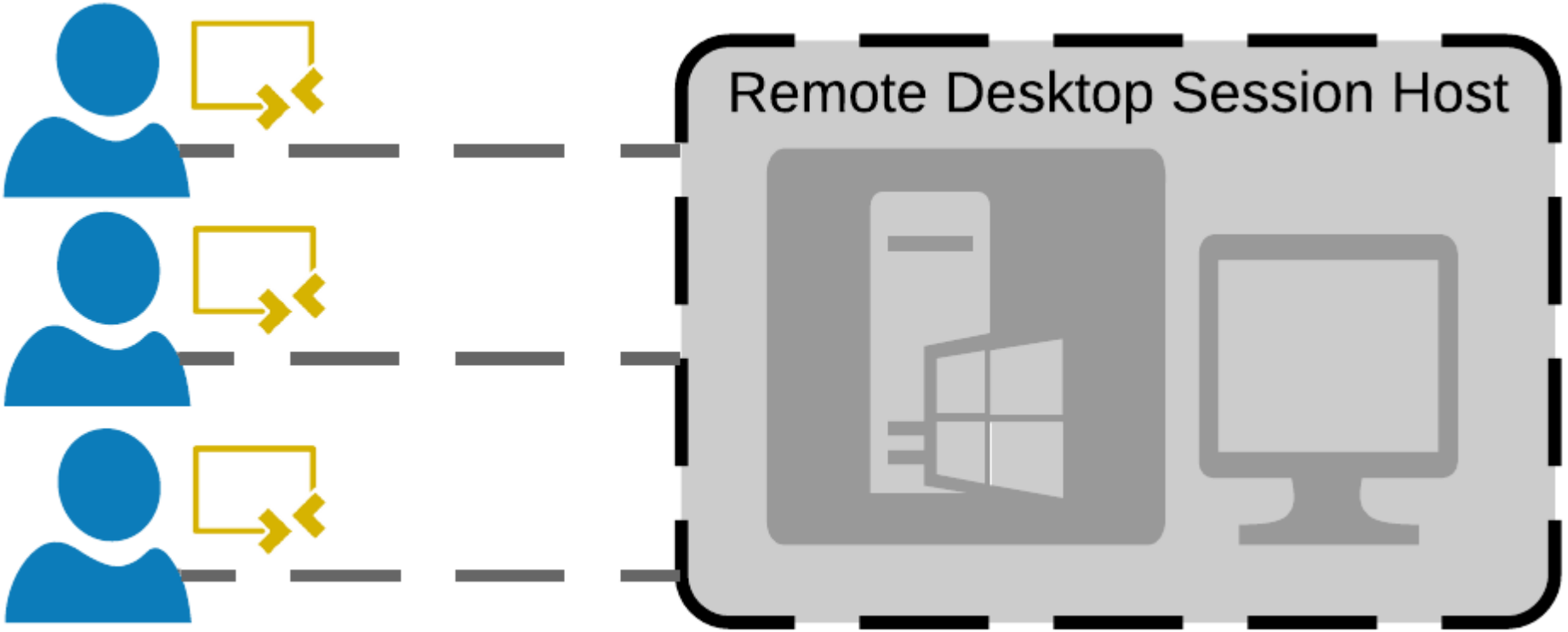
→ Service to the community: knowledge transfer

Access to resources for an extended period of time, even outside the UZH network.

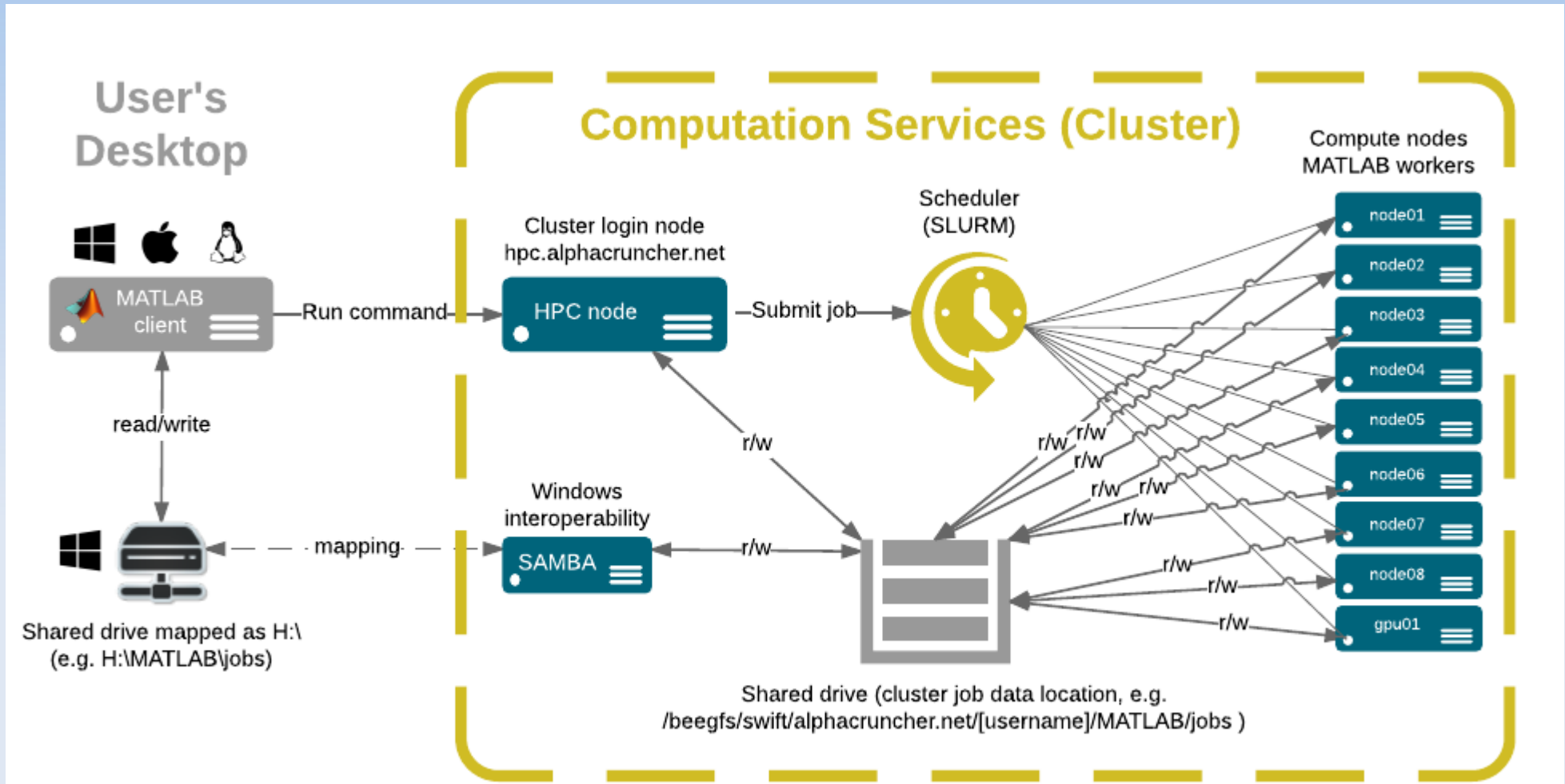
1. The compute infrastructure



Session-based desktop deployment



Compute Cluster



2. Accessing WLAN during ZICE

You all received the login credential today.

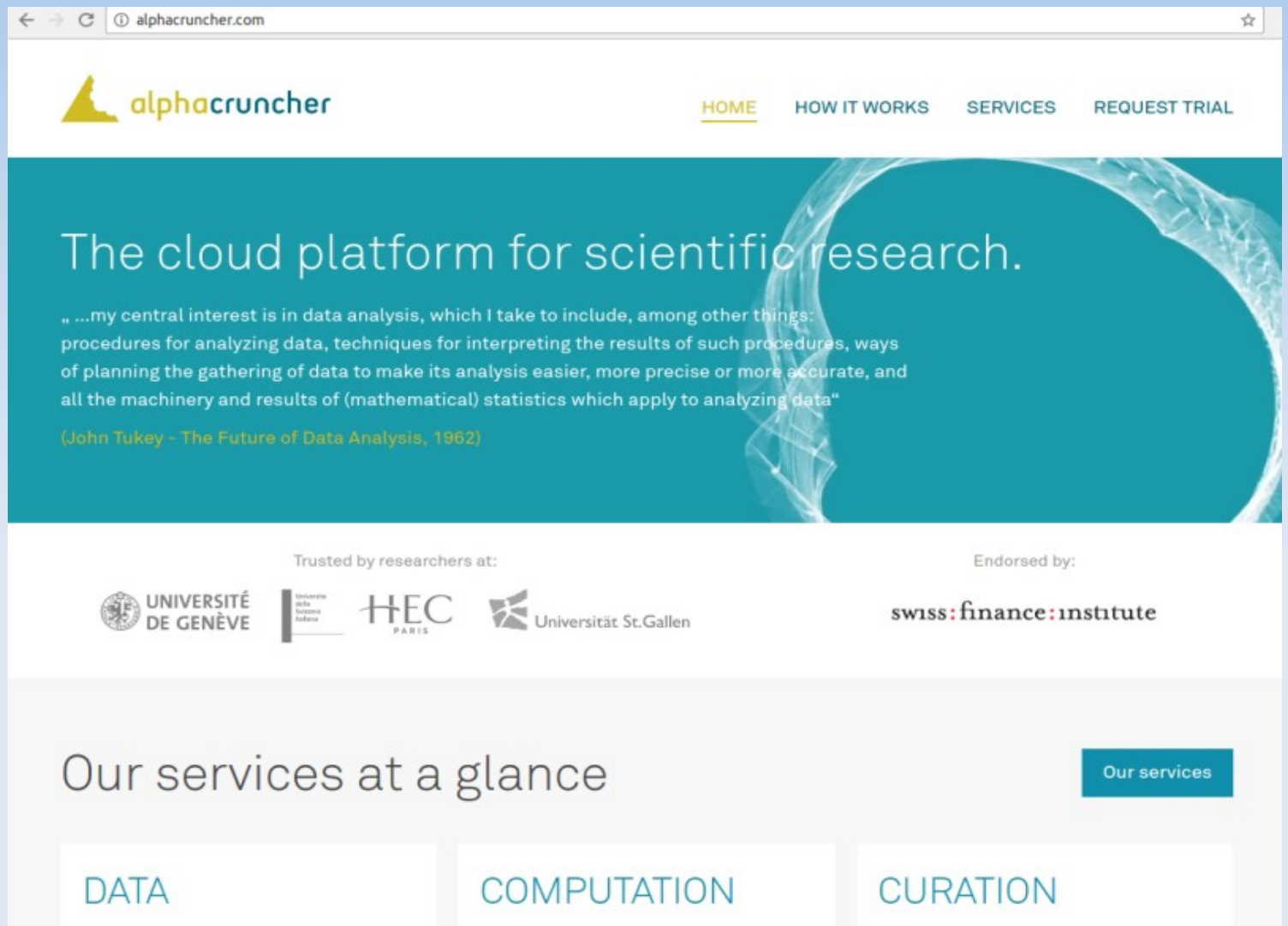
Make sure you **DO (IF POSSIBLE) NOT LOG INTO THE NETWORK WITH MULTIPLE DEVICES AT THE SAME TIME and no NETFLIX PLEASE :)**

→ we want to avoid banwith problems.

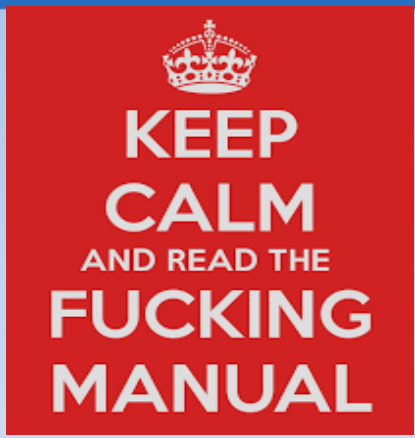
WiFi Names (SSID) – passwd: **IBF_Router**
IBF Router 1
IBF Router 2
IBF Router 3

3. Access to Alphacruncher Services

URL: alphacruncher.com



Access to the remote desktop



- You all obtained an access guide for Windows – Mac OS X – Linux
- More in the lecture notes – GIT classroom

→ **https://github.com/edualphacruncher/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY/simon/onboarding/onboarding_ZICE17.pdf**

→ **https://github.com/edualphacruncher/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY/zice17/simon/onboarding/access_guide_zice.pdf**

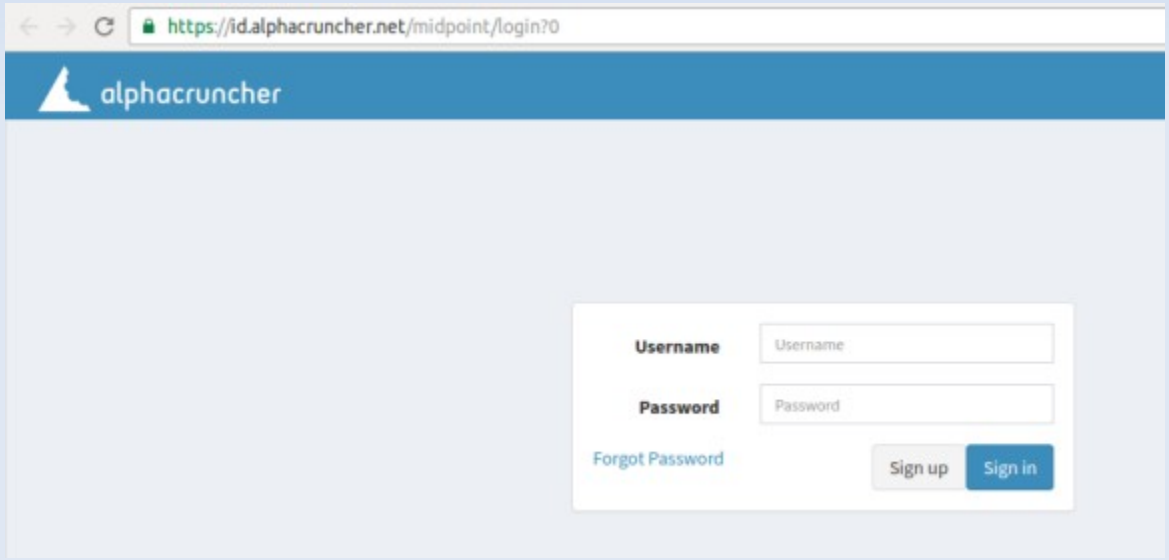
- We will step you now through Windows & Mac OS X

User name and password

You all (?) obtained an email stating

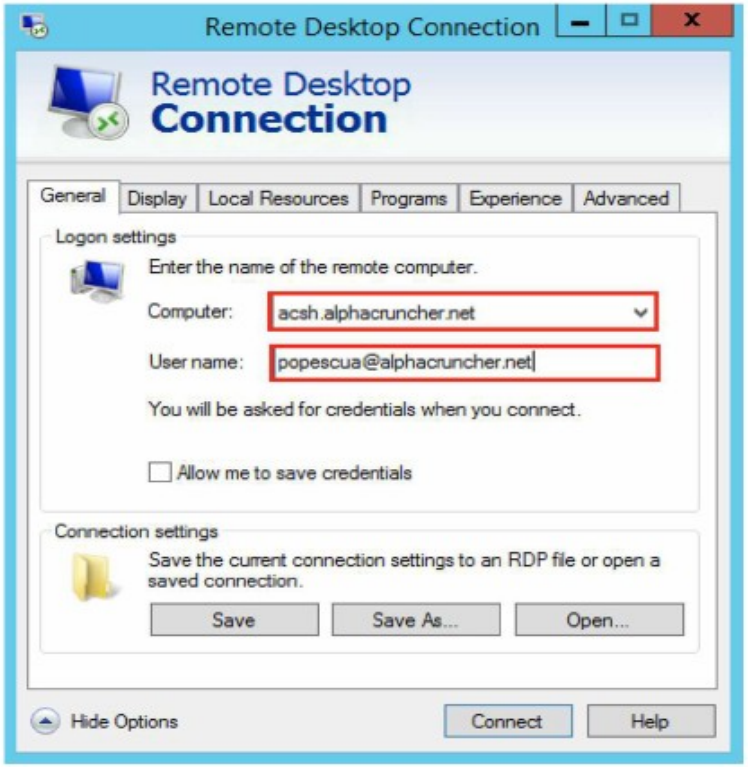
> Access to the platform is available via Remote Desktop Connection. You need to (re)set the password first before proceeding. Please point your browser to https://id.alphacruncher.net and follow the instructions (see access guide).
>
> Your username: XYZ

→ Reset your password here: **https://id.alphacruncher.net**
Click on the **Forgot Password** Link and follow the instructions



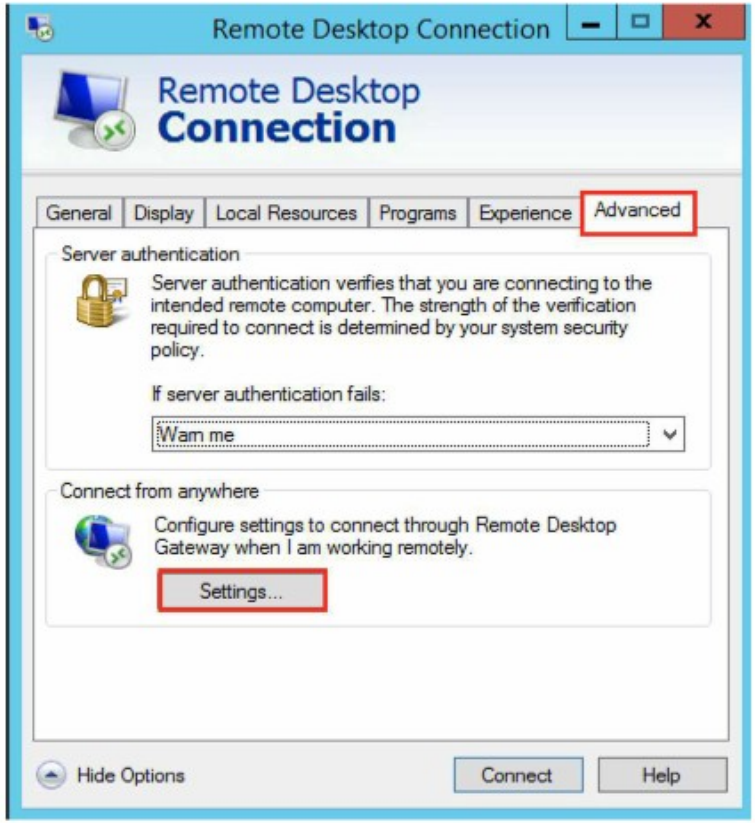
Windows (I)

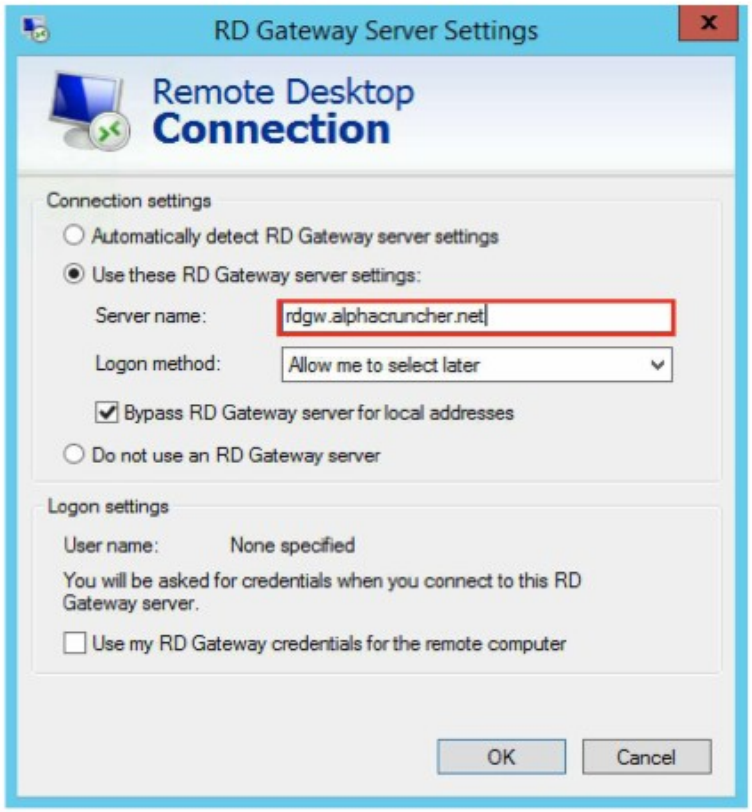
Access Remote Desktop Connection
from the **Start** button,
then click on **show options**



Computer: **eduzh.alphacruncher.net**
User Name: **your** username

Windows (II)

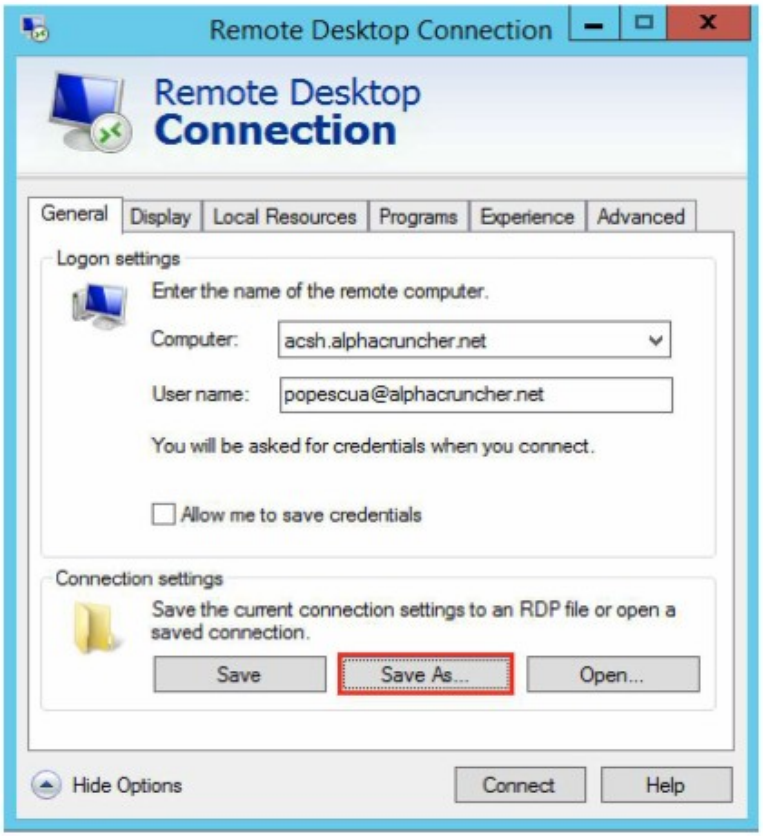


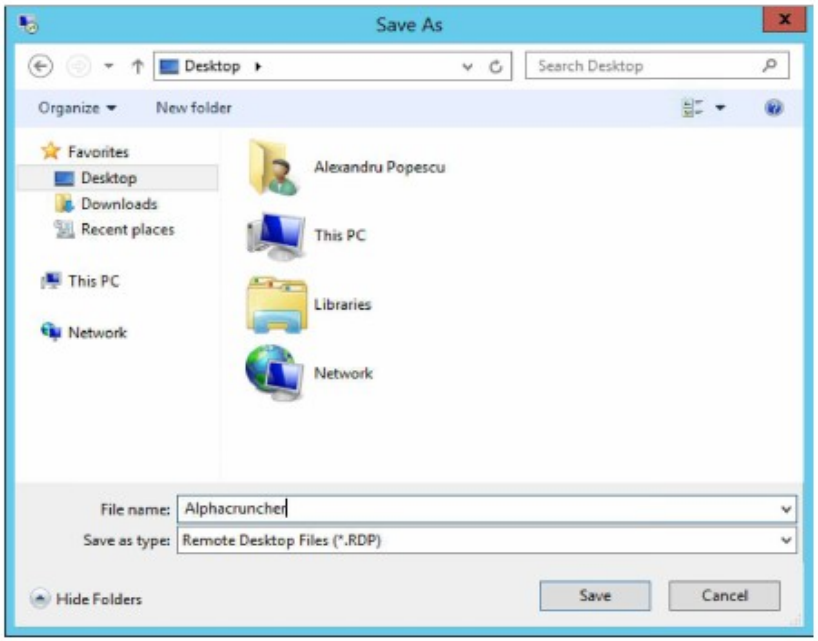


Go on "**Advanced**", click "**Settings**"

Server: **rdgw.alphacruncher.net**
Logon: *Allow me to select later*

Windows (III)

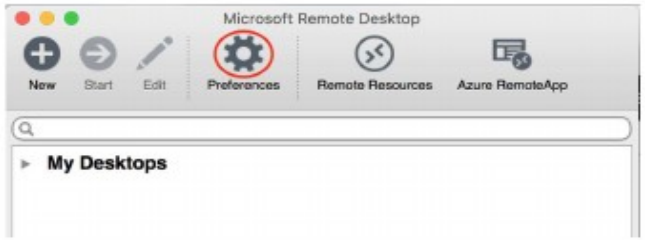


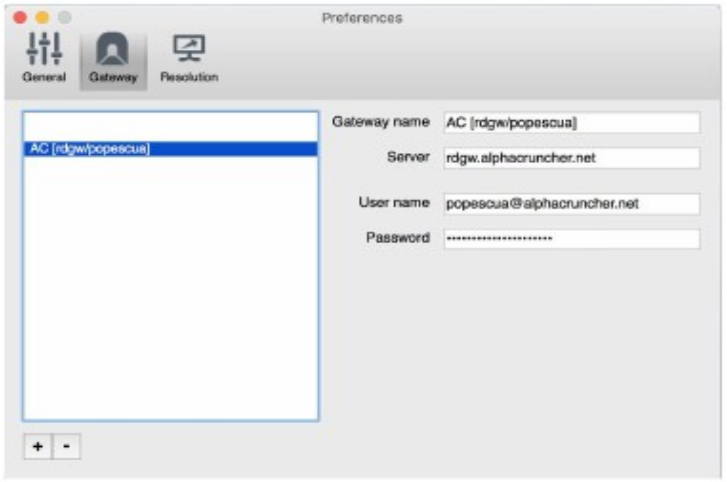


Click on "**save as**"

Save with ***.rdp** extension. Now you can connect

MAC OS X





Download the Remote Desktop Client from the Mac App Store.
Set up a New Gateway and click **Preferences**

Gateway: AC[rdgw/username]
Server: rdgw.alphacruncher.net
User: [username]@alphacruncher.net

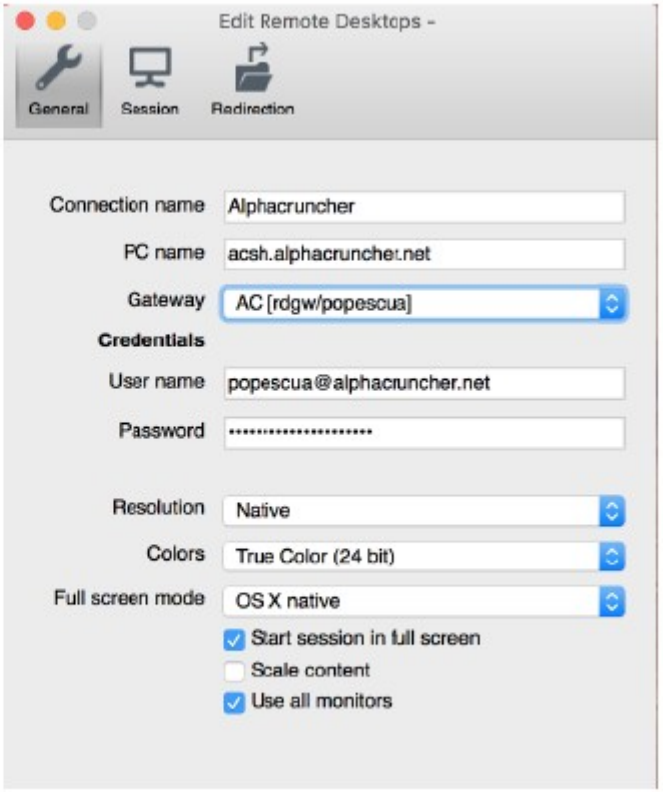
MAC OS X (II)

On the Connection Center screen, click **New**

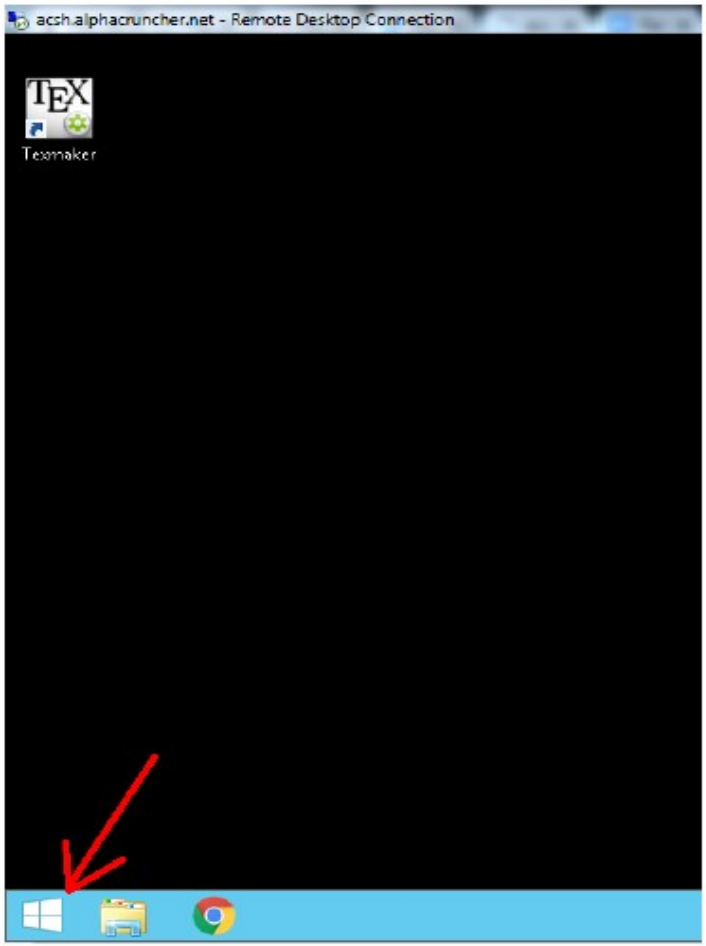
Enter the following information:

- General: Alphacruncher
- PC name: eduzh.alphacruncher.net
- Gateway: AC [rdgw/your username]
- User name: [your user-name]@alphacruncher.net

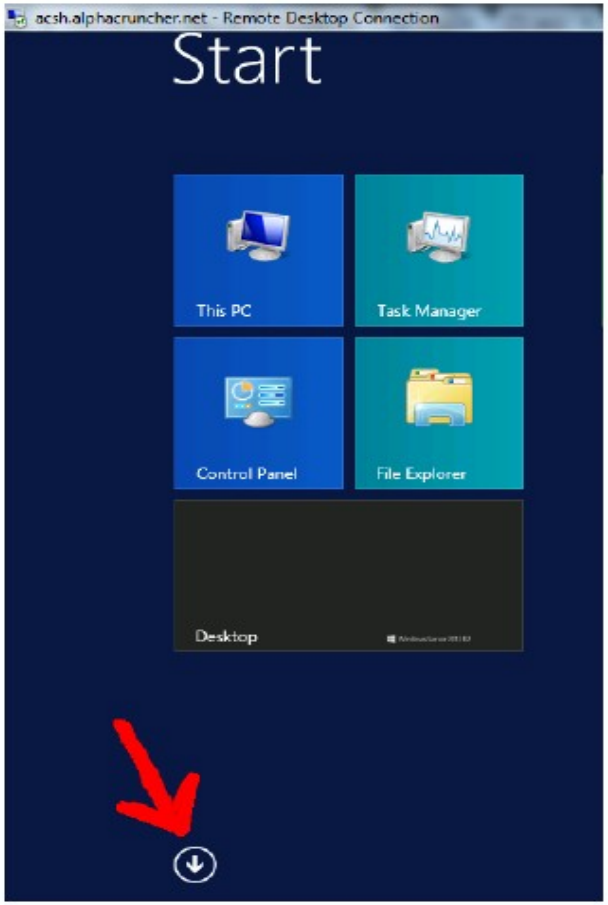
Connect!



Remote desktop

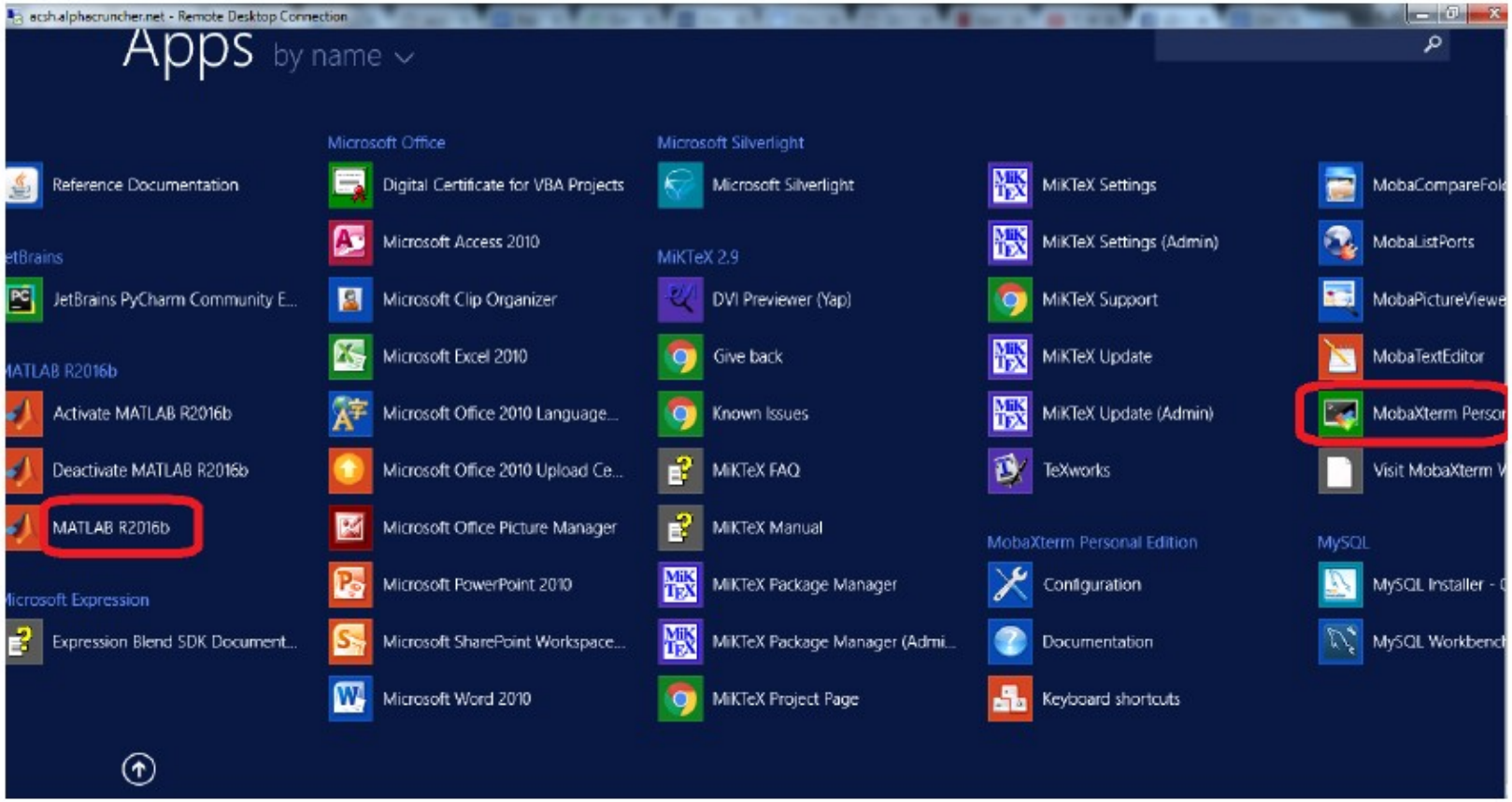


Click on the **Start** button



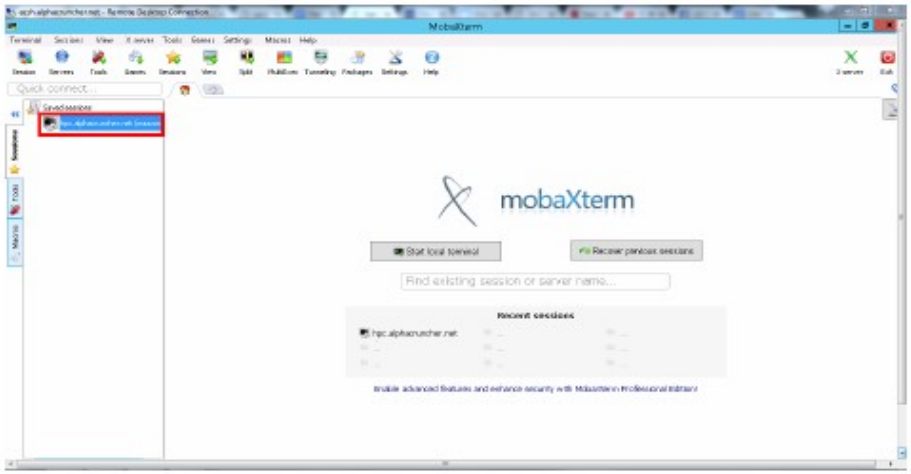
Use the lower-pointing arrow to scroll down...

Two apps you need all the time

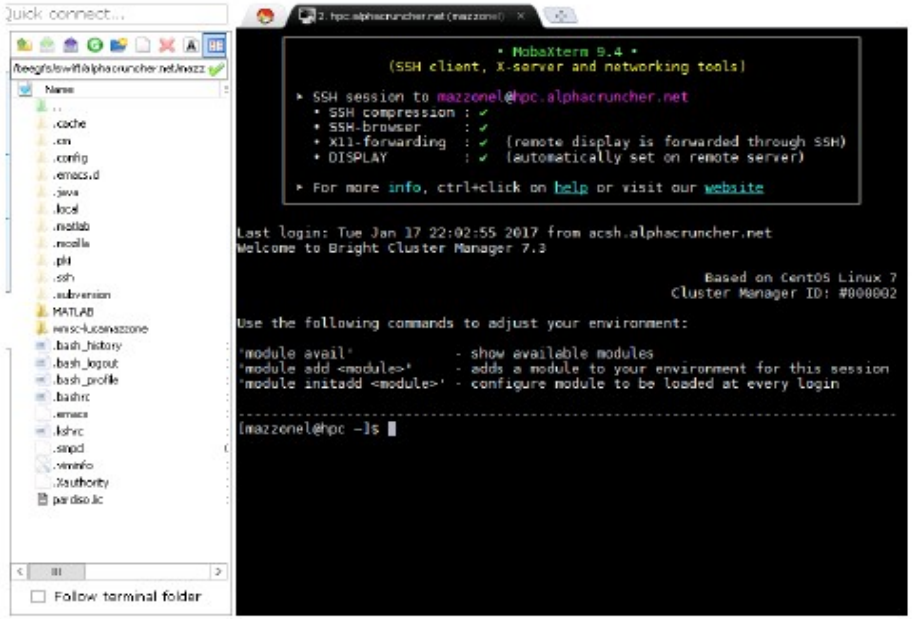


Now you can open Matlab 2016b in display mode by clicking on the icon
Click on MobaxTerm to access the command line of the cluster

Accessing the “Linux” cluster



Access by clicking on the highlighted icon on top left...



... and you’re finally on the command line of the cluster!

Help

On-sight support:

- Luca Mazzone (luca.mazzone@bf.uzh.ch)

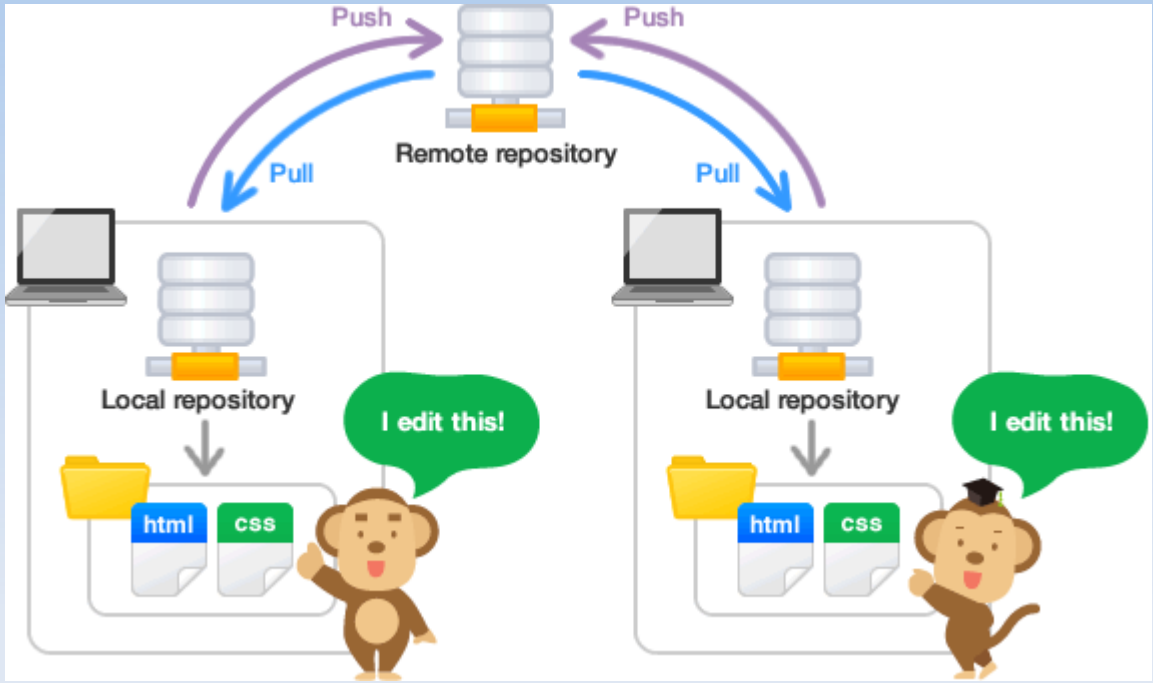
Support for nasty cases:

- Alphacruncher (support@alphacruncher.com)

Emergency support :)

http://lmgtfy.com/?q=help+on+alphacruncher

4. Course management – GIT classroom



The official website: **https://git-scm.com/**
Free git book: https://progit2.s3.amazonaws.com/en/2016-03-22-f3531/progit-en.1084.pdf

How I use git :)

https://xkcd.com/1597/

Git overview

- As you develop software and make changes, add features, fix bugs, etc. it is often useful to have a mechanism to **keep track of changes** and to ensure that **your code base and artifacts are well-protected** by being stored on a reliable server (or multiple servers).

- This allows you access to **historic versions of your application's code** in case something breaks or to "roll-back" to a previous version if a critical bug is found.

- The solution is to use a revision control system that allows you to "check-in" changes to a code base.

- It keeps track of all changes and allows you to **"branch"** a code base into a separate copy so that you can develop features or enhancements in isolation of the main code base (often called the "trunk" in keeping with the tree metaphor).

- Once a branch is completed (and well-tested and reviewed), it can then be merged back into the main trunk and it becomes part of the project.

Version control system

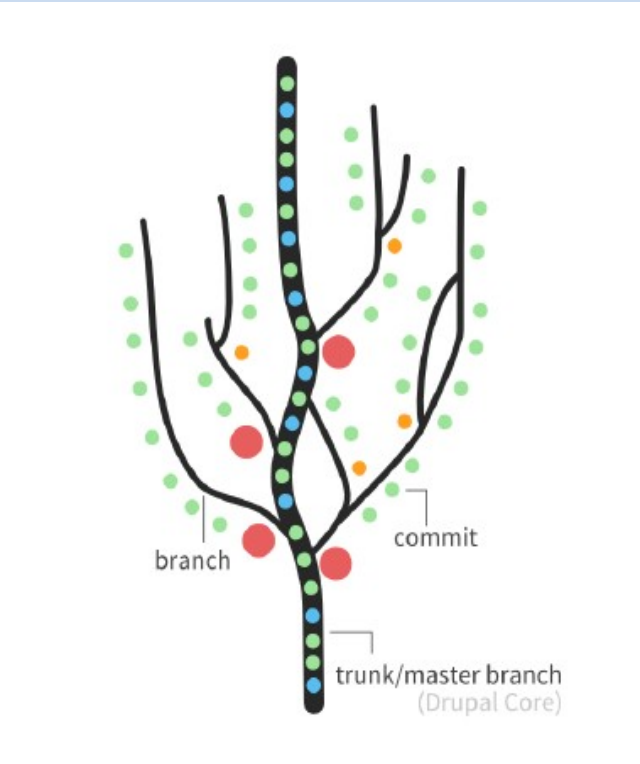
- Git essentially keeps track of all changes made to a project and allows users to work in large teams on very complex projects while minimizing the conflicts between changes.

- These systems are not only used for organizational and backup purposes, but are absolutely essential when **developing software** as part of a team.

- Each **team member can have their own working copy of the project** code without interfering with other developer's copies or the main trunk.

- Only when separate branches have to be merged into the trunk do conflicting changes have to be addressed.

- Otherwise, such a system allows multiple developers to work on a very complex project in an organized manner.



ZICE 17 Classroom

- We use git to provide you with **lecture slides/codes**

- **GitHub assignment link for class**:
  https://classroom.github.com/assignment-invitations/2dcd7145a43f8b7daba3e77fa84d88c

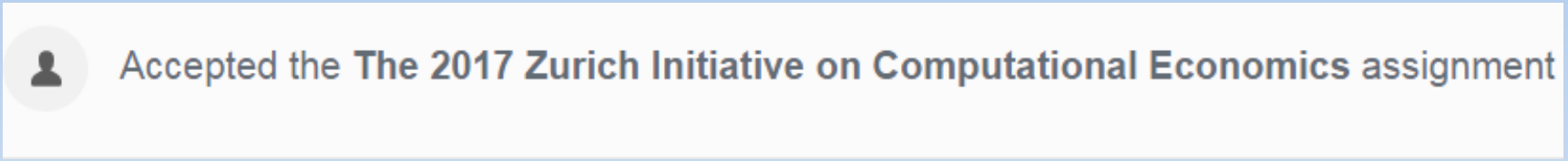
- **This is a fork of the master repository**:
  https://github.com/edualphacruncher/zice17
  (Private repository available only to speakers)

- **Keep the master repository up to date** (see below Slide 27.)

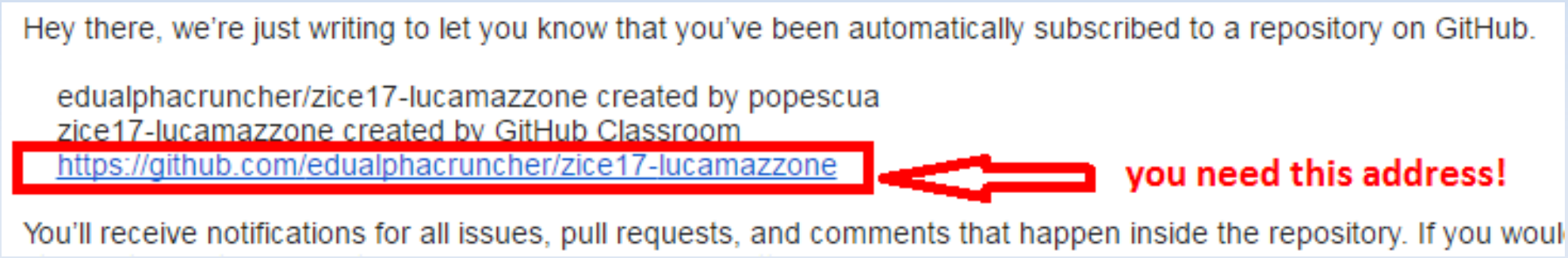
- You find github's cheat sheet here:
  .../zice17/simon/onboarding/github-git-cheat-sheet.pdf

Git on the remote desktop

You received an e-mail containing an invitation to participate in the GitHub repository. By clicking it, you will see that you have:



In the following e-mail, copy the address of your **private repository** to the clipboard – you'll need this later.



Notice: the address will be in the form https://github.com/edualphacruncher/zice17-YOUR-PRIVATE-REPOSITORY

Git on the remote desktop (II)

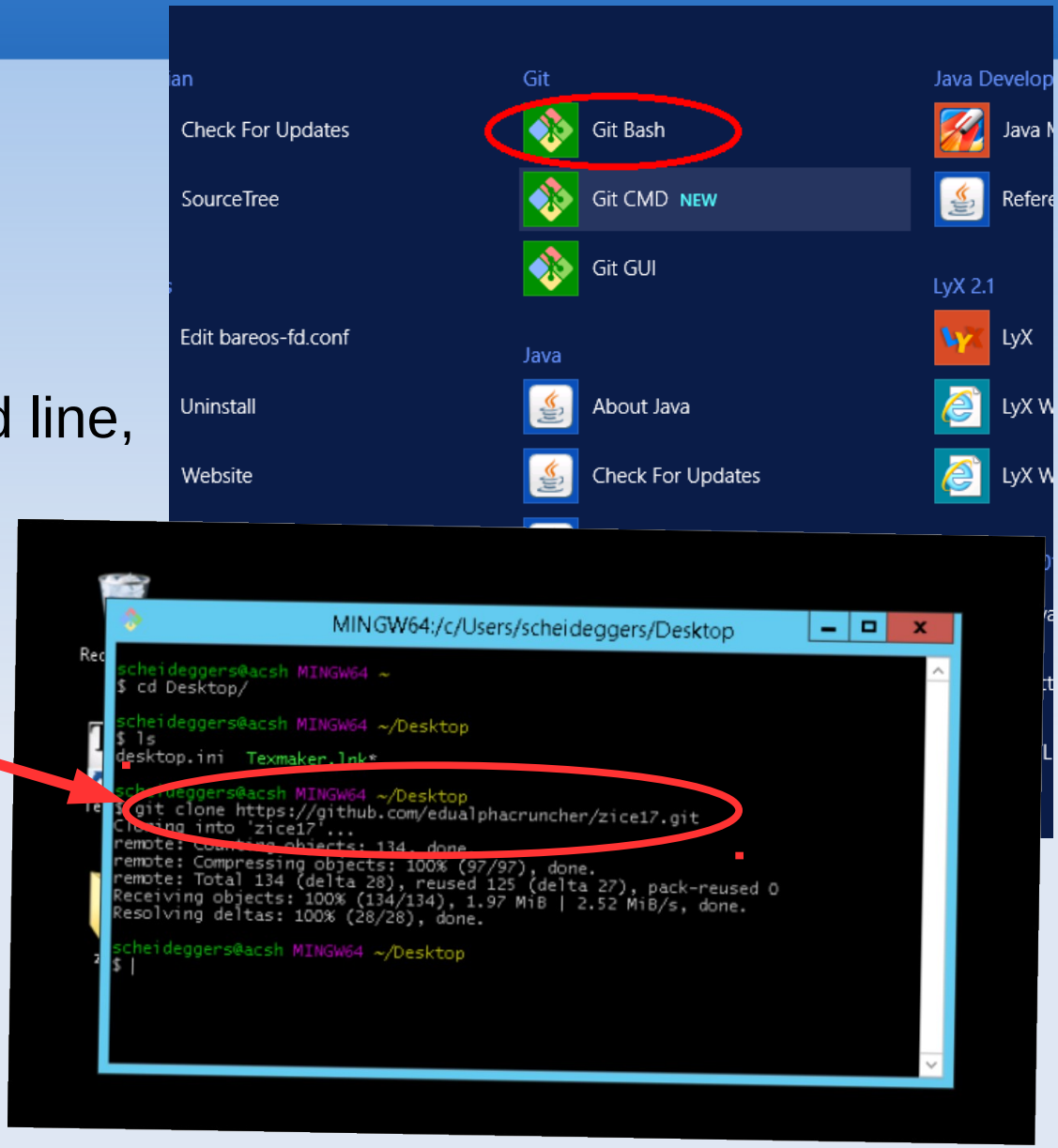
On the first day, you need to create a corresponding folder where to receive files.

On the remote computer command line, write:

$ cd Desktop (or wherever)
$ git clone *address*

Where the address is again:
https://github.com/edualphacruncher/...

zice17-YOUR-PRIVATE-REPOSITORY



Git on the remote desktop (III)

To keep your copy of the ZICE17 master repository up to date:

→ Change the current working directory to your local private repository:

```
mazzonel@eduzh MINGW64 ~/Desktop (master)
$ cd zice17/
```

→ **List the current configured remote repository for your fork**:

```
mazzonel@eduzh MINGW64 ~/Desktop/zice17 (master)
$ git remote -v
origin  git@github.com:edualphacruncher/zice17.git (fetch)
origin  git@github.com:edualphacruncher/zice17.git (push)
```

→ **Specify a new remote upstream repository that will be synchronized with your fork:**

```
mazzonel@eduzh MINGW64 ~/Desktop/zice17 (master)
$ git remote add upstream git@github.com:edualphacruncher/zice17
```

→ **Verify the new upstream repository you have specified:**

```
mazzonel@eduzh MINGW64 ~/Desktop/zice17 (master)
$ git remote -v
origin  git@github.com:edualphacruncher/zice17.git (fetch)
origin  git@github.com:edualphacruncher/zice17.git (push)
upstream        git@github.com:edualphacruncher/zice17 (fetch)
upstream        git@github.com:edualphacruncher/zice17 (push)
```

Git on the remote desktop (IV)

Once you synchronized the remote repository to your local folder, you can fetch from the remote repository whenever there is an update (this will happen throughout the workshop, and you will be instructed by lecturers about when to update your repository to get the new materials!). To fetch and update, do the following:

Fetch the branches and their respective commits from the upstream repository:

```
mazzonel@eduzh MINGW64 ~/Desktop/zice17 (master)
$ git fetch upstream
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 3), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (12/12), done.
From github.com:edualphacruncher/zice17
 * [new branch]      master     -> upstream/master
```

Finally, update your private repository!

```
mazzonel@eduzh MINGW64 ~/Desktop/zice17 (master)
$ git pull upstream master
From github.com:edualphacruncher/zice17
 * branch            master     -> FETCH_HEAD
Updating c3b0066..d1b5ef1
Fast-forward
 ZICE17_info/README.md                          |    5 +++++
 simon/onboarding/onboarding_ZICE17.pdf | Bin 1991822 -> 0 bytes
 2 files changed, 5 insertions(+)
 create mode 100644 ZICE17_info/README.md
 delete mode 100644 simon/onboarding/onboarding_ZICE17.pdf
```

Git Classroom – summary

The following instructions can be executed on any environment where you decided to clone your fork of the original repository (zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY)

1) **Clone repository zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY to your environment**
(https://help.github.com/articles/cloning-a-repository/)

$ git clone https://github.com/edualphacruncher/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY.git

2) Configuring a remote for the fork (https://help.github.com/articles/configuring-a-remote-for-a-fork/)

Change the current working directory to your local project.

$ cd ~/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY

List the current configured remote repository for your fork.

$ git remote -v

origin https://github.com/edualphacruncher/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY.git (fetch)
origin https://github.com/edualphacruncher/zice17-YOUR_GITBHUB_PRIVATE_REPOSITORY.git (push)

Specify the original repository as a new remote upstream repository for your fork.

$ git remote add upstream https://github.com/edualphacruncher/zice17.git

3) **Syncing the fork** (https://help.github.com/articles/syncing-a-fork/ and http://www.eqqon.com/index.php/Collaborative_Github_Workflow)

Git on the cluster

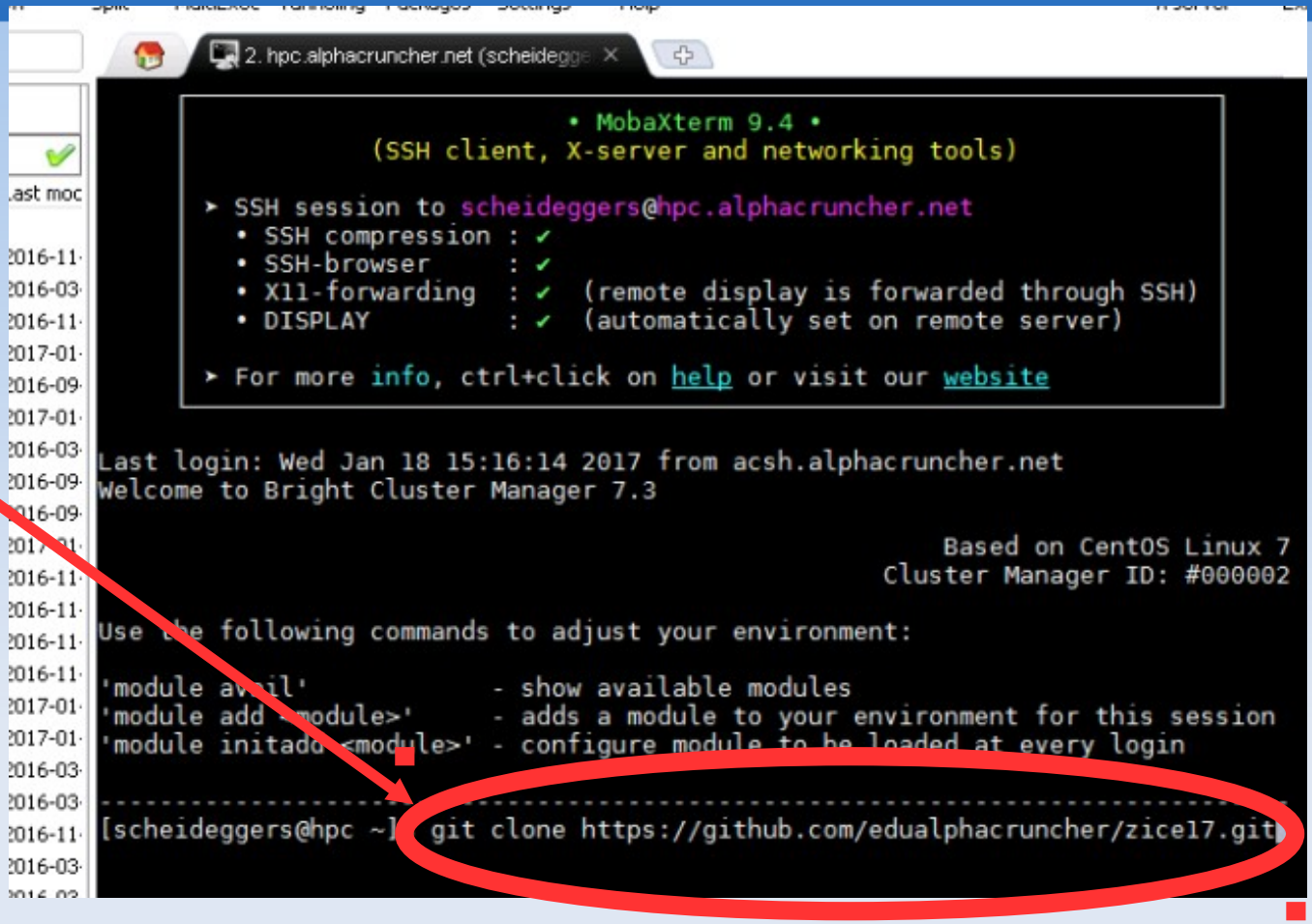
MobaXTerm, log in

clone the repository:
$ git clone *address*

To update the repository in future circumstances, go to the directory

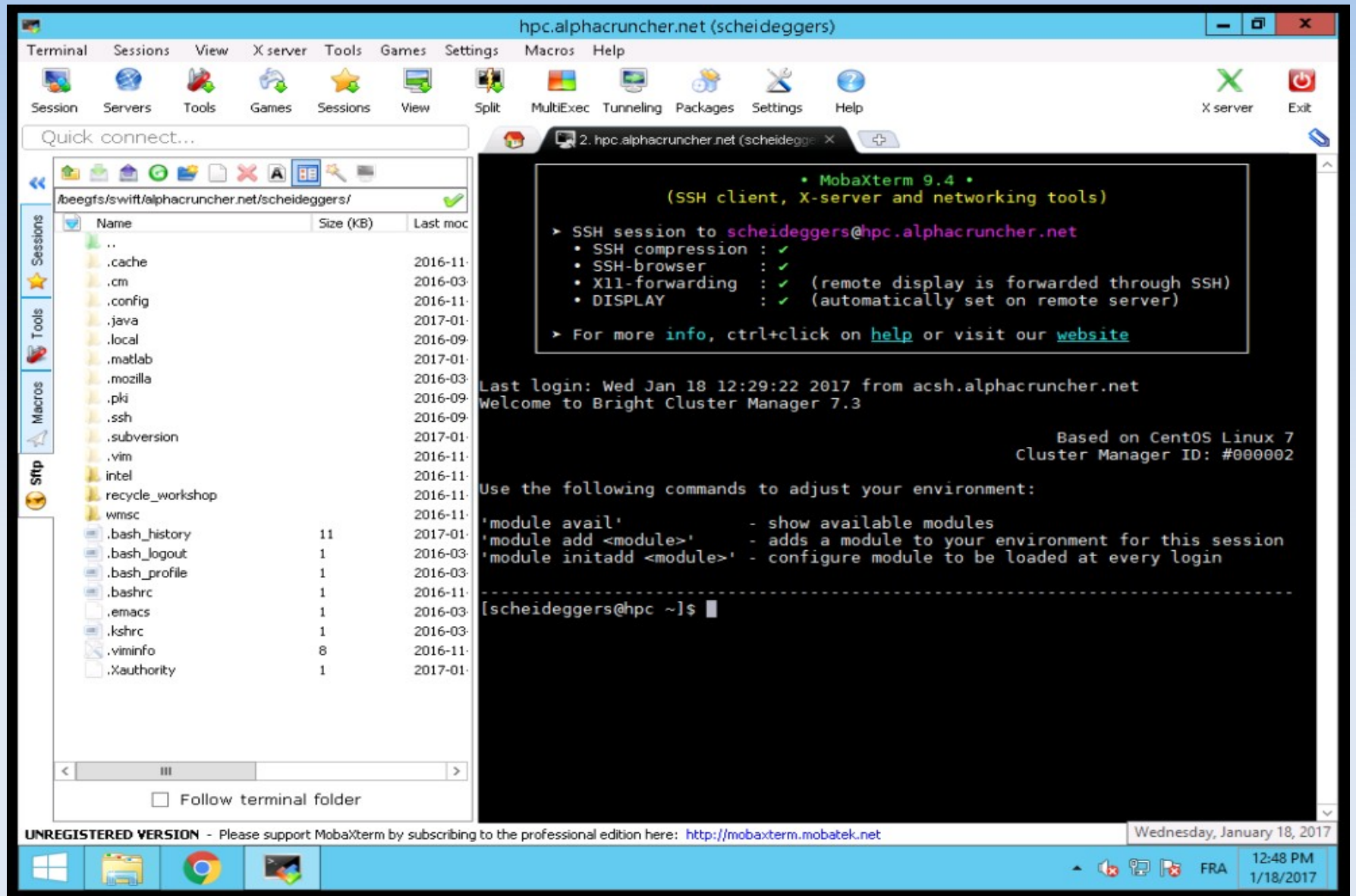
$ cd zice17-YOUR-PRIVATE-REPOSITORY

Then you can follow exactly the same directions as in the previous slides!

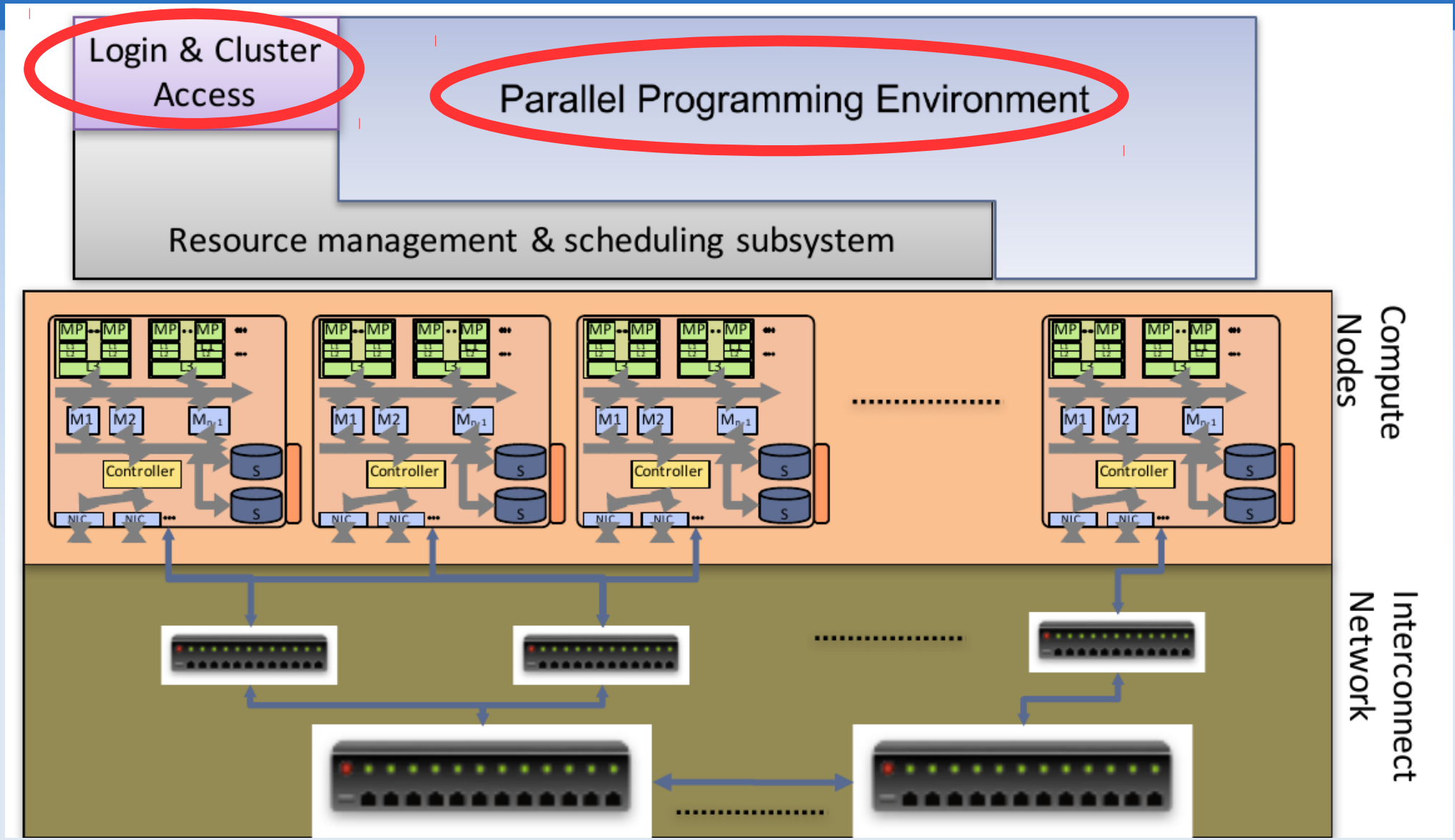


Again, *address* is: **https://github.com/edualphacruncher/zice17-YOUR-PRIVATE-REPOSITORY**

4. First steps on a Linux cluster



High-level architecture of a HPC system

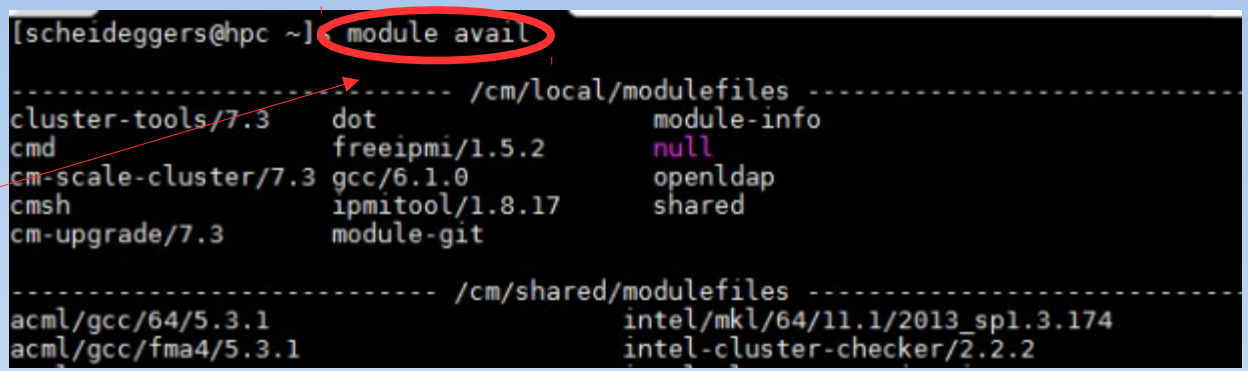


Programming environment

Supporting diverse user community requires supporting diverse tool sets (versions of compilers, debuggers, libraries, etc)

- Check available packages

$module avail

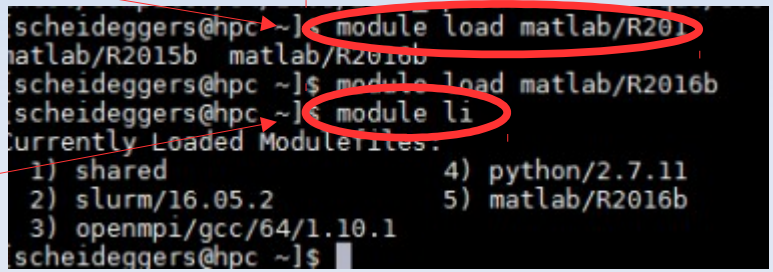


- Load a module (e.g. Matlab)

$module load matlab/R2016b

- Check what is loaded

$module list



Note: most high-performance clusters set-up in this way !

Basic Linux commands (1)

Small-scale cheat-sheet on terminal commands

Command	Description
pwd	Print name of current/working directory
cd [Directory]	Change directory (no directory → change to home)
ls [Directory]	List directory contents (no directory → list current)
cat FILE	Concatenate files and print on the standard output
mkdir DIRECTORY	Make directories
mkdir -p DIRECTORY	Make directories, make parent directories as needed
cp SOURCE... DIRECTORY	Copy files and directories
cp -r SOURCE... DIRECTORY	Copy files and directories, copy directories recursively
mv SOURCE... DIRECTORY	Move (rename) files
man COMMAND	An interface to the on-line reference manuals

Basic Linux commands (2)

Small-scale cheat-sheet on terminal commands

Command	Description
ssh -X foo@host.com	OpenSSH SSH client (remote login program), access to host.com with user foo
scp foo@host.com:/home/bar ./	Secure copy (remote file copy program), copy file bar from /home on host.com to directory
scp bar foo@host.com:/home/	Secure copy (remote file copy program), copy file bar from the local host to /home on host.com
git clone git@github.com:whatever folder-name	The stupid content tracker, Clone a repository (whatever) into a new directory (folder-name).
git checkout	Checkout a branch or paths to the working tree.

Step-by-Step – an example

```
> pwd
/beegfs/swift/alphacruncher.net/USERNAME
> mkdir -p firstFolder/secondFolder
> ls
FirstFolder
> ls firstFolder
secondFolder
> cd firstFolder
> pwd
/beegfs/swift/alphacruncher.net/USERNAME/firstFolder
> ls
secondFolder
```

KEEP

CALM

AND

GOOD

LUCK