Cloud Computing - lab 2

Cloud control and monitoring for a parking system using Nokia Internet of Things

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| Author | Cosmin Ivan, Cristian Cosariu |
| Owner | Nokia Timisoara, Universitatea de Vest Timisoara |
| Organization | Nokia Romania |
| Approver | Florin Ciocan |
| Document ID |  |
| Document location | public |
|  | |  |
|  | | |  |

Contents

[1 Generate teams 3](#_Toc463367001)

[2 My virtual machine (MobaXterm) 3](#_Toc463367002)

[2.1 Accessing the machine trough ssh 3](#_Toc463367003)

[2.2 Useful links 4](#_Toc463367004)

[3 GIT - version control system 5](#_Toc463367005)

[3.1 What is GIT 5](#_Toc463367006)

[3.2 My first project in GIT 6](#_Toc463367007)

[3.3 Git branches 9](#_Toc463367008)

[3.4 Useful links 9](#_Toc463367009)

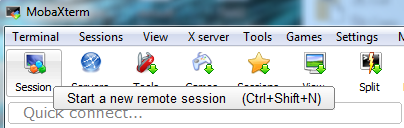
# Generate teams

15 mins to generate the teams. Provide also a name for the team.

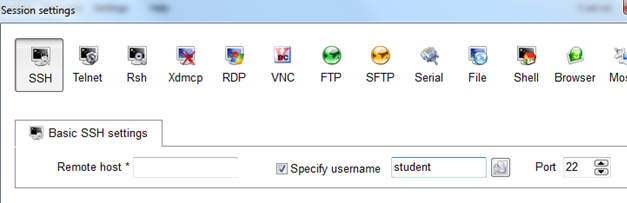
# My virtual machine (MobaXterm)

## Accessing the machine trough ssh

1. Open MobaXterm from the desktop (in case it is missing you can download it from the link in the section “[1.2 useful links](#_Useful_links)”)
2. Open a new session

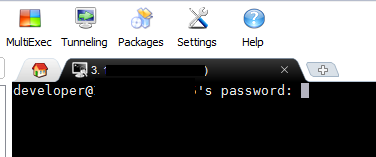


1. On the SSH tab add your virtual machine IP Address, click on specify username and add your username. The required information can be found on the paper that you received.

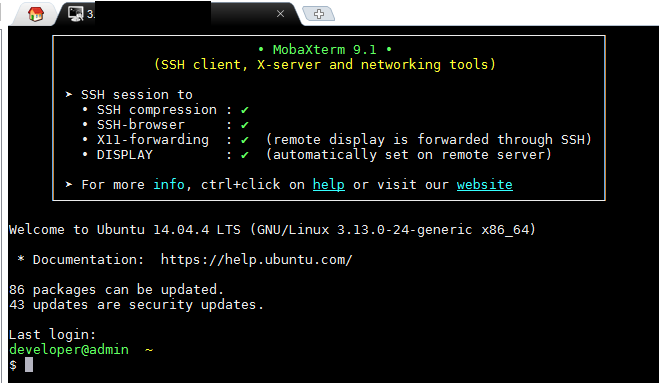


The ip address must be something like: 10.11.8.xx

1. After you click ok a terminal will open and you will be asked for your password



1. You are now connected to your virtual machine



## Useful links

<http://mobaxterm.mobatek.net/demo.html>

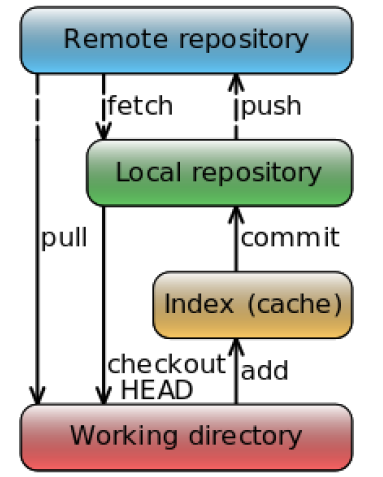
<http://mobaxterm.mobatek.net/download.html>

# GIT - version control system

## What is GIT

Git is a [free and open source](https://git-scm.com/about/free-and-open-source) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git workflow:



## GitLab and first project in GIT

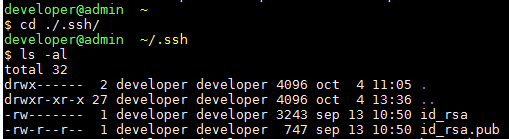
1. Login to <http://10.11.8.14> with the user provided for your team.
2. At first login you have to change the generic password (**student111**)
3. Create a project and name it **cloud\_[TEAM\_NAME]**. Leave the settings to Private.
4. Click on the project created, notice that there’s an URL for this project (SSH or HTTP)
5. Open a terminal with MobaXterm. Create a connection to your assigned workstation, using the user for your team. Change the generic password (**student111**) using **passwd** command.
6. First we need to generate a ssh key. Run the command bellow (accept the default file location and don’t set a password):

**ssh-keygen -t rsa -b 4096 -C "*your\_email@example.com*"**

1. Now go to the home directory to see if the file has been generated:

**cd ~/.ssh/**

**ls -al**

****

You should have id\_rsa and id\_rsa.pub in the folder.

1. In order to see the key run the following command:

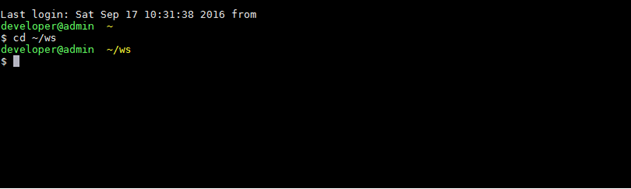
**cat id\_rsa.pub**

1. Now let’s add the key to git lab. Access the url <http://10.11.8.1>4 using firefox or another browser.
2. Go to settings -> profile settings -> SSH Keys and copy your ssh key here (notice: to copy something from the terminal you just have to select is with the mouse)
3. Now let’s create a working directory and go inside of it. In the terminal run the commands:

**mkdir ~/ws**

1. Go to the folder ws using the command line:

**cd ~/ws**

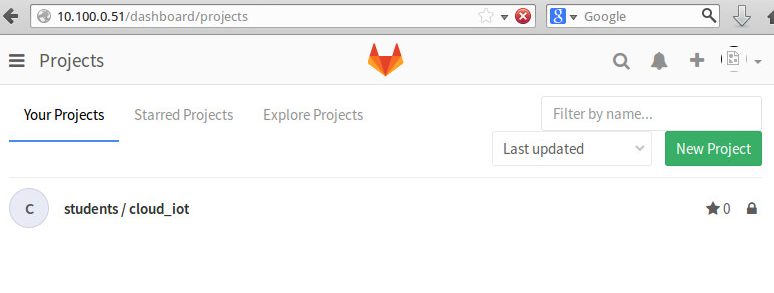
****

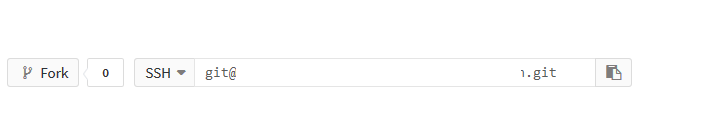
1. Inside the folder clone your project

**git clone [url taken from gitlab]**

**Notice: if the dns for gitlab machine is not set you will need to change cloud-git with the machines ip 10.11.8.14**

To copy the git remote url go to the project students/cloud\_iot in the browser on gitlab -> select ssh and copy the url



****

1. A folder with the repository has been created. Enter it with cd command:

**cd cloud\_[TEAM\_NAME]**

1. Now let’s try to do some changes. To open atom editor just write atom in the terminal and execute it:

**atom**

1. What happened? An instance for atom has been started on your virtual machine. Thanks to mobaXterm the visual output from the virtual machine is redirected to your pc.
2. The next step is to open your new project in atom. **File -> Add project folder**. The full path for the project is: **/home/student/ws/cloud\_iot**
3. In the project folder create a new file with your name and the extension .txt : [your\_name].txt. Add some text inside and save it.
4. In order to see the files that you have modified just execute the git status command:

**git status**

1. The next step is to add the modified files to the staged status:

**git add [your\_name].txt**

In case you want to add all files just use the “.” instead of the file name ( **git add .** )

1. In order to send our changes to the server we need to create a commit:

**git commit –m “[Your name]: [a specific message for what has been changed]”**

1. You received an error because your user is not set up. Add the two commands bellow:

**git config –-global user.email “[yourmail]”**

**git config –-global user.name “[your teams name]”**

1. Now we are ready to send the changes to the server but first we need to check if we have the latest version of the project and if someone else has done other changes on the same file. To get all the changes from the server use the command:

**git pull**

Do not worry if you forget since git will notice you that your project is not up to date.

1. To send the changes to the server just use the command:

**git push -u origin master**

1. Now all the changes are on the server and other developers from the project will be able to see them.
2. Let’s make again the steps and create a new file with the name team\_info.txt and add inside the name of your team. Don’t forget: add-> commit->pull -> push

## Git branches

1. To be easier to know on with branch you are working we need to modify the “.profile” file to be able to see the branch in the terminal. Another solution would be to sue git status.
2. Use the command below to edit the profile file:

**nano ~/.profile**

1. Add a new line in the file with this content:

**export PS1="\[\033[32m\]\u@\h \[\033[35m\]$MSYSTEM \[\033[33m\]\w\[\033[36m\]\$(\_\_git\_ps1)\[\033[0m\]\n$ "**

1. Use (Ctrl+X to save the file)
2. Run the command below to reload the bash profile:

**source ~/.profile**

1. Create a new branch with your team name

**git checkout -b [team\_name]**

1. Now you should see the name of the working branch in the terminal. Also if you run git status you should be noticed that your current branch is [team\_name]
2. Add one new file with a name that you desire and do a commit for it (don’t push it yet).
3. After you do the commit add another file and do the another commit with it
4. Push the changes on the server and check what happened in gitLab

**git push -u origin [team\_name]**

## Useful links

<https://git-scm.com/>

<https://tortoisegit.org/>

<http://learngitbranching.js.org/>