LIGHTNING PROFESSOR MANUAL

Lightning 0.8.5

11/07/2018

This professor manual will explain how to create repositories of personalized scenarios and Docker images to be used with **Lightning** (https://github.com/ptoribi/lightning), the network simulator based on Docker containers.

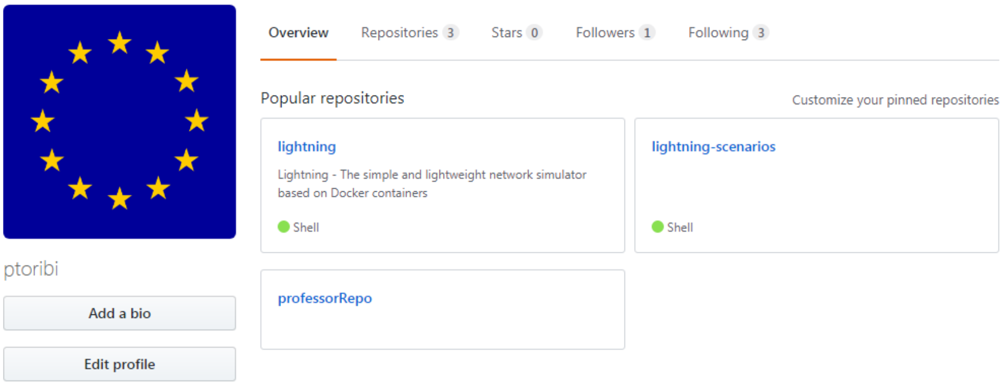
For installing the software please refer to the *"Installation Manual"*. For information about how to use this program, please take a look to the *"User Manual"*.

Use your own scenarios (stored in the cloud)

- ### Sign in and create a Git repository

  First, you will have to create an account in one of the several Git servers that offer this service, or to use a local Git server already deployed in your organization. Probably the most famous online Git repository is GitHub.

  It is up to you chose one Git server provider, create an account and start a new repository.



Lightning - The simple and lightweight network simulator based on Docker containers

Online Git server account with some repositories already created

As an example, in the next steps of this guide we will assume that the repository used by the professor is:
`https://github.com/ptoribi/professorRepo.git`

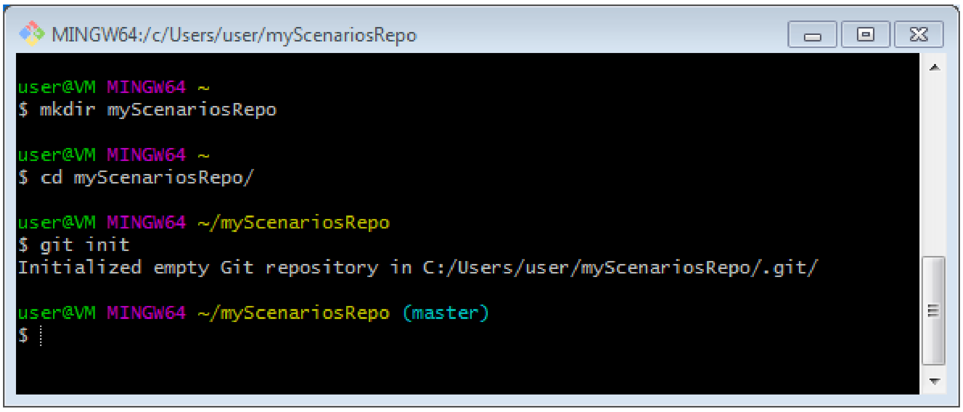
- **Install the Git version control software in your computer**

  Please download the proper package for your operating system and follow the installation instructions that you will find in the Git Downloads page: https://git-scm.com/downloads

- **Create a git local repository in your machine**

  Create a folder in your machine, go inside that folder and initialize a new Git repository:

```
$ mkdir FOLDER_NAME
$ cd FOLDER_NAME
$ git init
```



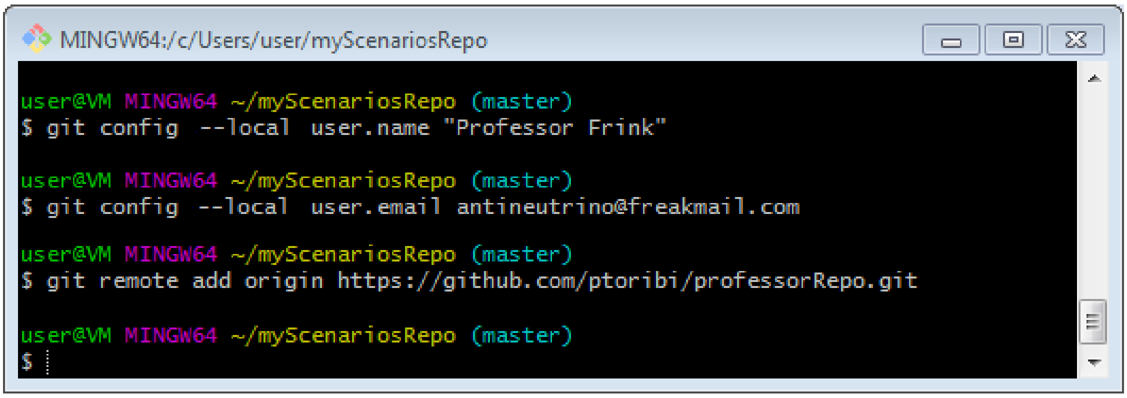
MINGW64:/c/Users/user/myScenariosRepo

```
user@VM MINGW64 ~
$ mkdir myScenariosRepo

user@VM MINGW64 ~
$ cd myScenariosRepo/

user@VM MINGW64 ~/myScenariosRepo
$ git init
Initialized empty Git repository in C:/Users/user/myScenariosRepo/.git/

user@VM MINGW64 ~/myScenariosRepo (master)
$
```

Your local Git repository needs some basic configurations before working (this step only has to be done once):

```
$ git config --local user.name "Your Name"
$ git config --local user.email yourEmail@server.com
$ git remote add origin
https://github.com/ptoribi/professorRepo.git
```

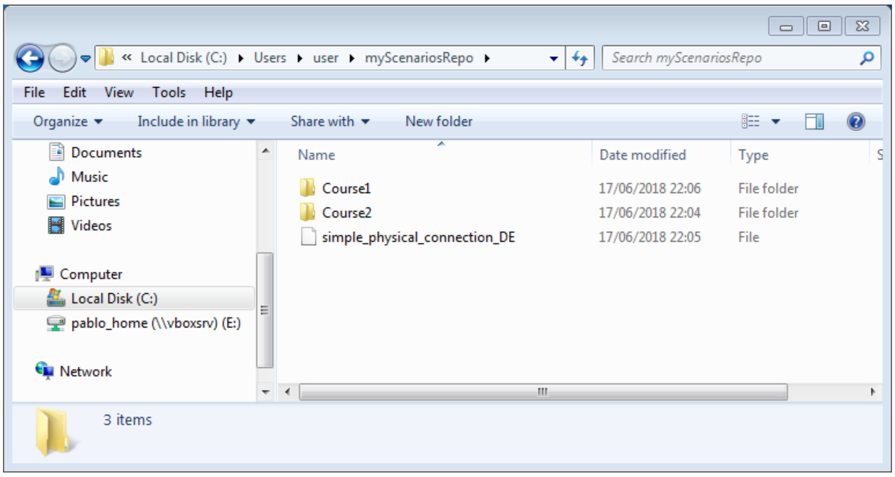
The argument `--local` means that data will only be used for that repository.



MINGW64:/c/Users/user/myScenariosRepo

```
user@VM MINGW64 ~/myScenariosRepo (master)
$ git config --local user.name "Professor Frink"

user@VM MINGW64 ~/myScenariosRepo (master)
$ git config --local user.email antineutrino@freakmail.com

user@VM MINGW64 ~/myScenariosRepo (master)
$ git remote add origin https://github.com/ptoribi/professorRepo.git

user@VM MINGW64 ~/myScenariosRepo (master)
$
```

Now is time to populate your new repository with scenario files. It is possible to classify your network scenarios in folders inside your repository. You can find some sample scenarios in the Lightning example repository: https://github.com/ptoribi/lightning-scenarios


« Local Disk (C:) ▸ Users ▸ user ▸ myScenariosRepo ▸
Search myScenariosRepo
File Edit View Tools Help
Organize ▾ Include in library ▾ Share with ▾ New folder
Documents
Music
Pictures
Videos
Computer
Local Disk (C:)
pablo_home (\\vboxsrv) (E:)
Network

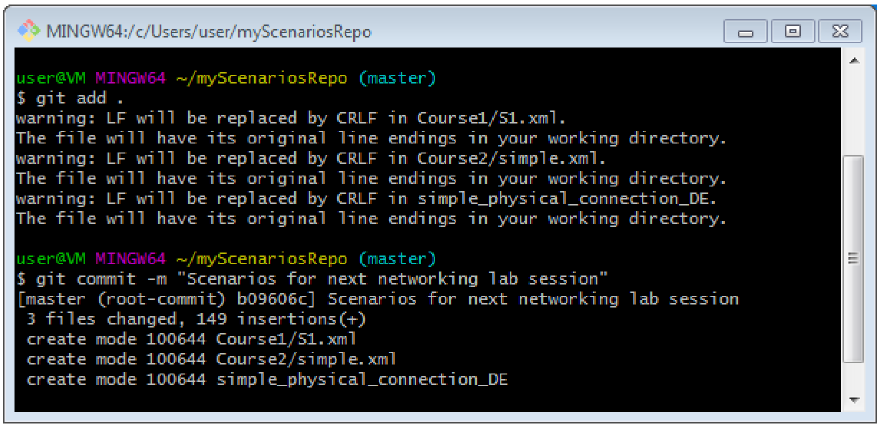
Name	Date modified	Type
Course1	17/06/2018 22:06	File folder
Course2	17/06/2018 22:04	File folder
simple_physical_connection_DE	17/06/2018 22:05	File

3 items

- **Apply changes and upload to server**

Once you have finished, it is time to add the files to the repository and commit the changes:

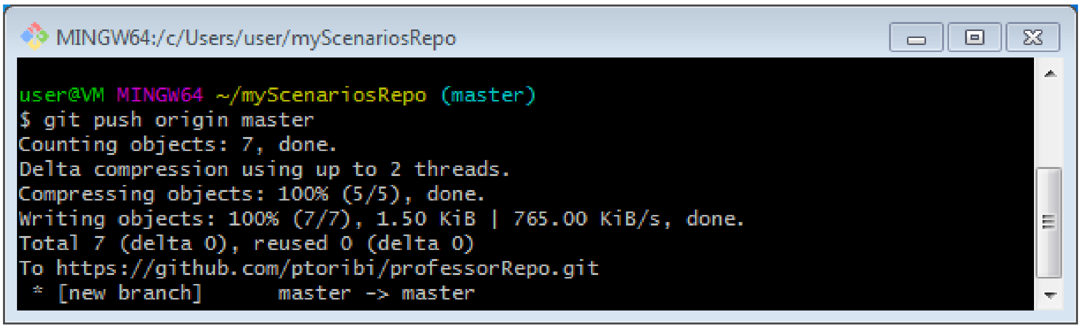
```
$ git add .
$ git commit -m "Your comment describing the changes"
```



```
user@VM MINGW64 ~/myScenariosRepo (master)
$ git add .
warning: LF will be replaced by CRLF in Course1/S1.xml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in Course2/simple.xml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in simple_physical_connection_DE.
The file will have its original line endings in your working directory.

user@VM MINGW64 ~/myScenariosRepo (master)
$ git commit -m "Scenarios for next networking lab session"
[master (root-commit) b09606c] Scenarios for next networking lab session
 3 files changed, 149 insertions(+)
 create mode 100644 Course1/S1.xml
 create mode 100644 Course2/simple.xml
 create mode 100644 simple_physical_connection_DE
```

Finally, the file changes in your repository must be uploaded:

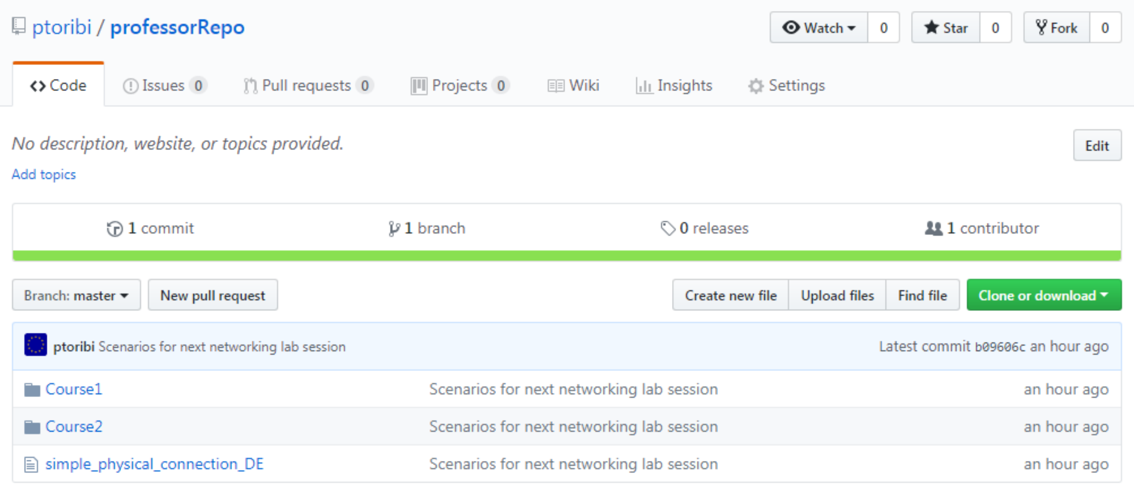
```
$ git push origin master
```



```
user@VM MINGW64 ~/myScenariosRepo (master)
$ git push origin master
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 1.50 KiB | 765.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0)
To https://github.com/ptoribi/professorRepo.git
 * [new branch]      master -> master
```

- **<u>Use your new scenarios repository with Lightning</u>**

Now you can browse your repository to check that everything worked properly:



No description, website, or topics provided.

1 commit · 1 branch · 0 releases · 1 contributor

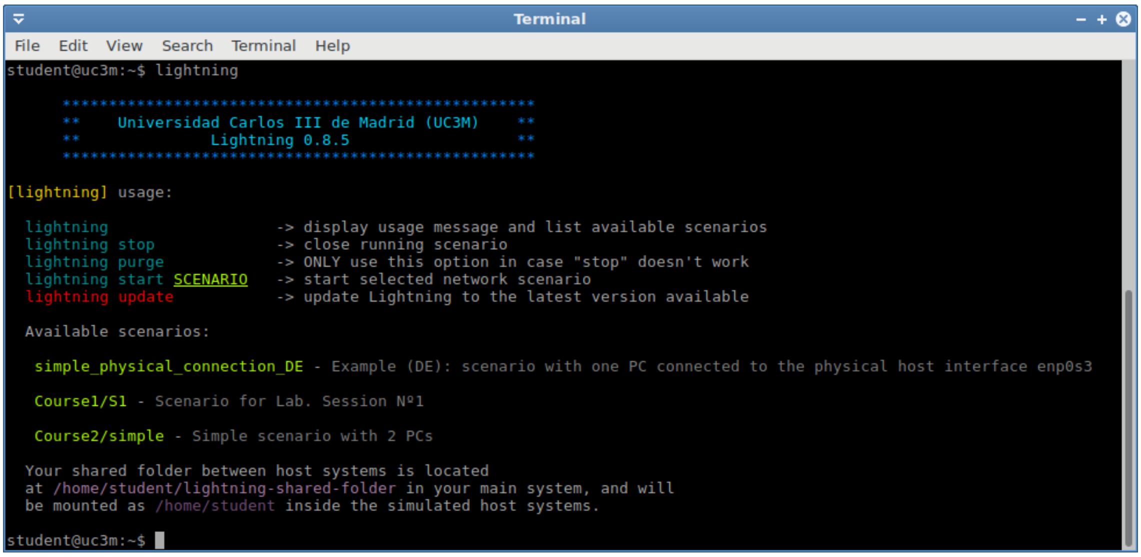
ptoribi Scenarios for next networking lab session — Latest commit b09606c an hour ago

Course1	Scenarios for next networking lab session	an hour ago
Course2	Scenarios for next networking lab session	an hour ago
simple_physical_connection_DE	Scenarios for next networking lab session	an hour ago

Please ask the system administration of your organization to set the value of your personalized repository in the `variables.conf` file of the Lightning installations:

```
# Git repository for SCENARIOS
GIT_REPO_SCENARIOS="https://github.com/ptoribi/professorRepo.git"
```

After changing the value, a "`lightning update`" has to be performed on each machine, so the new scenarios are downloaded and installed into the system. The next time Lightning will be executed, the new scenarios will be available:



```
student@uc3m:~$ lightning

      ***************************************************
      **    Universidad Carlos III de Madrid (UC3M)    **
      **              Lightning 0.8.5                  **
      ***************************************************

[lightning] usage:

  lightning                  -> display usage message and list available scenarios
  lightning stop             -> close running scenario
  lightning purge            -> ONLY use this option in case "stop" doesn't work
  lightning start SCENARIO   -> start selected network scenario
  lightning update           -> update Lightning to the latest version available

  Available scenarios:

   simple_physical_connection_DE - Example (DE): scenario with one PC connected to the physical host interface enp0s3

   Course1/S1 - Scenario for Lab. Session Nº1

   Course2/simple - Simple scenario with 2 PCs

  Your shared folder between host systems is located
  at /home/student/lightning-shared-folder in your main system, and will
  be mounted as /home/student inside the simulated host systems.

student@uc3m:~$ 
```

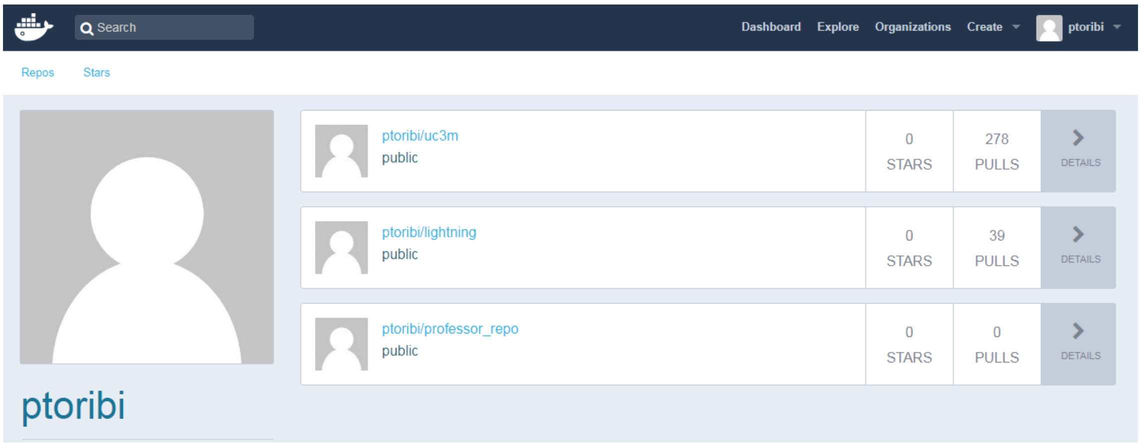
Use your own Docker images

- **Signing in and creating a Docker registry**

  A place for hosting Docker images is called a *registry*. First, you have to create an account in one of the several Docker registries that offer this service. Probably the most famous online Docker registry is Docker Hub.

  It is up to you chose one Docker registry provider, create an account and start a new repository.

  As an example, in the next steps of this guide we will assume that the repository used by the professor is: `ptoribi/professor_repo`



ptoribi/uc3m public	0 STARS	278 PULLS
ptoribi/lightning public	0 STARS	39 PULLS
ptoribi/professor_repo public	0 STARS	0 PULLS

Online Docker registry account with some repositories already created

- **Install the Docker Community Edition software in your computer**

  Please download the proper package for your operating system and follow the installation instructions that you will find in the Docker Documentation page: https://docs.docker.com/

  In the left panel go to "*Get Docker*" → "*Docker CE*" → Select your operating system and follow the guide.

- **Get the Lightning Docker base image**

  For the time being there are available two Docker images for Lightning, called "host" and "router". For this example, it will only be used the first one, but the procedure is analogous for the other.

  The first step will be to download and install the Docker image in the system:

```
# docker pull ptoribi/lightning:host
```

- **Create a container of that image**

  Now that we have the Docker image, containers based on it can be created. We will create one called `miContainertipoHost` for performing modifications:

  ```
  # docker run -it --privileged -v /home/professor:/mnt --name
  miContainerTipoHost ptoribi/lightning:host
  ```

  The `--privileged` argument will start the container in privileged mode for performing all type of changes without any restriction. The argument `-v` will mount the directory `/home/professor` of the main operating system into the `/mnt` directory of the container, this can be used for saving files into the container or installing software packages.

  Once all the modifications needed are performed, we can simply exit the container by typing “`exit`”:

  ```
  [root@container]# exit
  ```

  Now the container has stopped but its file remains in the system. For making further changes in the future it can be run again by executing:

  ```
  # docker start -i miContainerTipoHost
  ```

- **Create a new image from the modified container**

  An image can be created by packing that modified container. The name of the Docker repository has to be specified followed by a tag that names the image, in that case, “`host`”:

  ```
   # docker commit miContainerTipoHost
  ptoribi/professor_repo:host
  ```

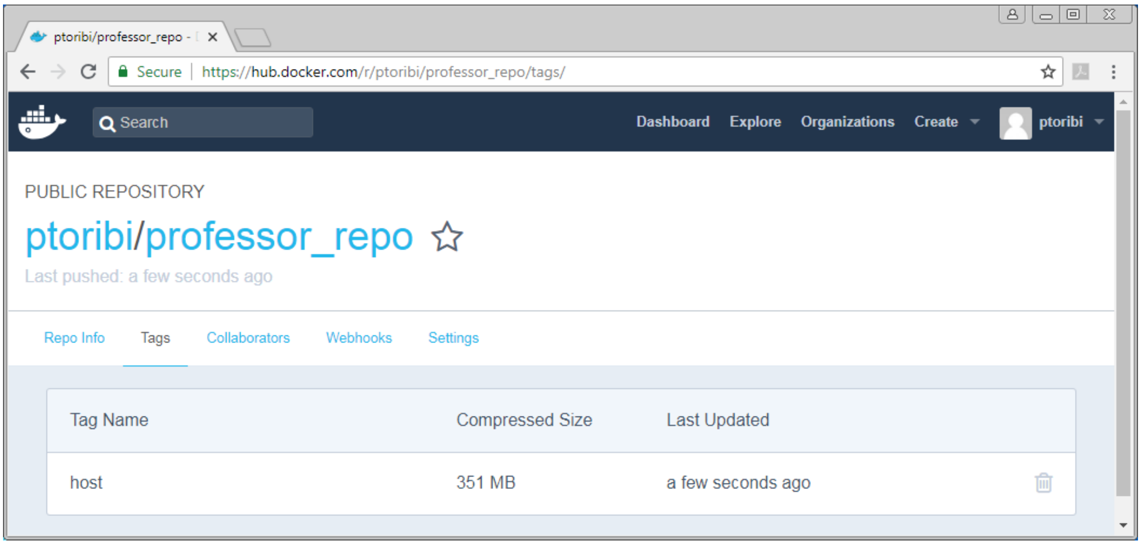
  For uploading the changes to the registry, first we have to login into the service:

  ```
  # docker login
  ```

  And finally, uploading the modified image:

  ```
  # docker push ptoribi/professor_repo:host
  ```

Now you can browse your repository to check that everything worked properly:



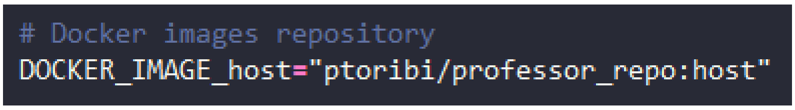
PUBLIC REPOSITORY

ptoribi/professor_repo

Last pushed: a few seconds ago

Tag Name	Compressed Size	Last Updated
host	351 MB	a few seconds ago

Please ask the system administration of your organization to set the value of your personalized repository in the `variables.conf` file of the Lightning installations:



```
# Docker images repository
DOCKER_IMAGE_host="ptoribi/professor_repo:host"
```

After changing the value, a “`lightning update`” has to be performed on each machine, so the new Docker images are downloaded and installed into the system. The next time Lightning will be executed, it will use the new filesystems.