

Guide to using the Git version control system, services GitHub and GitHubClassroom for completing labs

Step 1. Install Git

You must install the Git version control software on the operating system on which you will be performing the lab. For example, if you are performing the lab on a Linux OS that is installed on a virtual machine running on a Windows OS, you must install the Git software on the Linux guest OS, not on the main (host) Windows OS.

On Ubuntu, you can install Git using the **apt-get install git command**. Commands for installing Git for other Linux distributions, installers for Windows and Mac OS can be found on the official website of the Git version control system: <https://git-scm.com>.

Step 2. Register on github.com

If the user has not previously created an account on GitHub, they must go to <https://github.com> and register.

Step 3. Open the link with an individual task

To access the individual repository with the task, follow the link to GitHub provided by the teacher. If the user is not authorized on the site, the authorization window shown in Figure 1 will open. In this window, you must log in (enter your login and password), or go to the previous step and register on GitHub, if you have not done so yet, by clicking on the "Create an account" link.

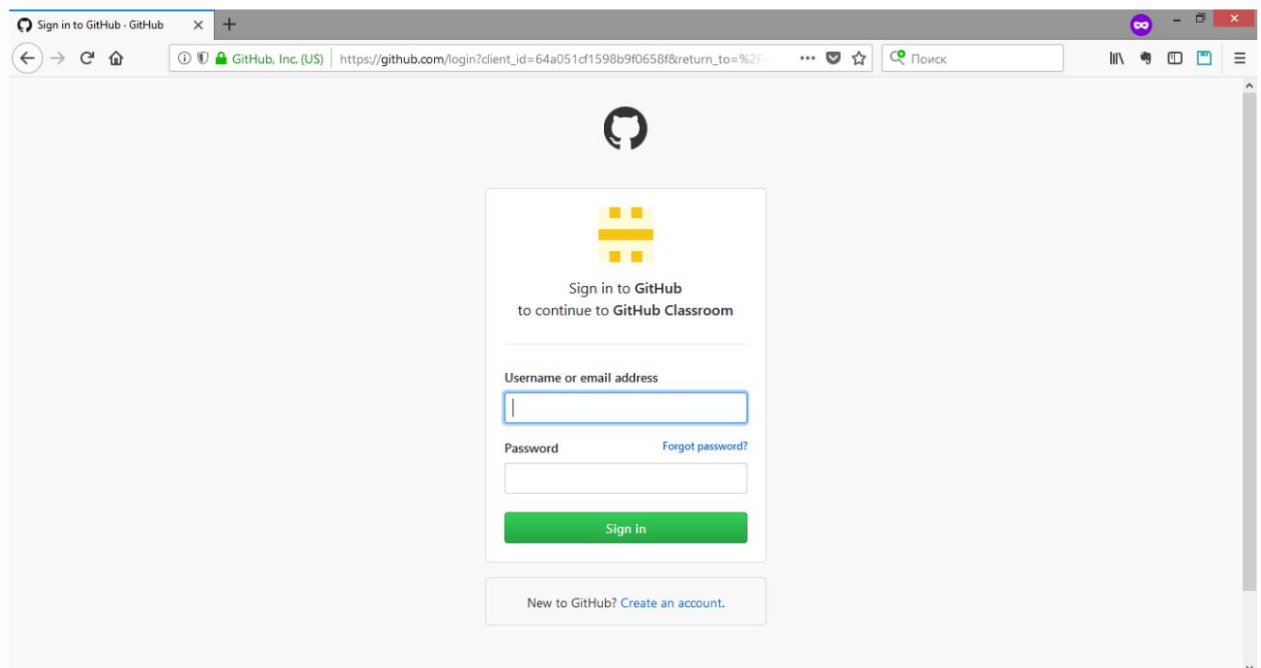


Figure 1. Authorization window

If the user is already logged in to GitHub, after clicking on the GitHub link provided by the teacher, the user should proceed to the next step.

Step 4. Initialize the repository

After successful authorization, GitHub will create an individual repository with the task for the authorized user. The window shown in Figure 2 will appear. Next, you need to follow the link to the repository with the task, located after the words "Your assignment has been created here».

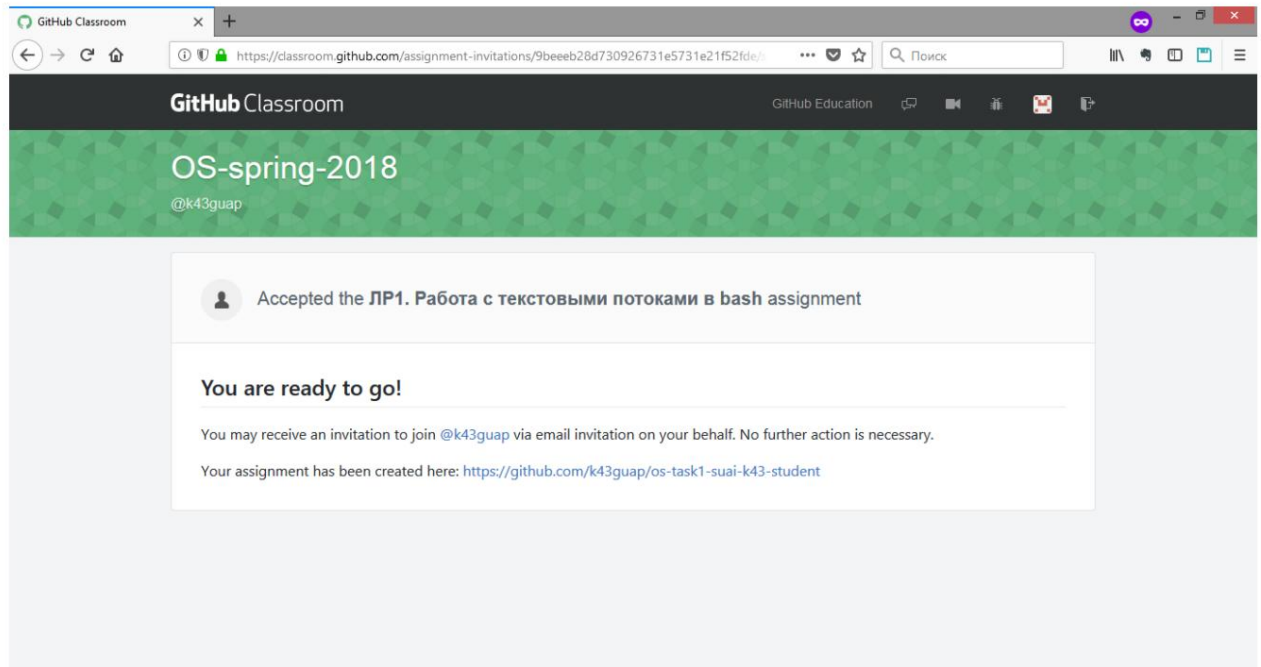


Figure 2. The repository initialization process is complete

Step 5. Connecting the automatic testing system

The build and tests will be run automatically with each new commit to the repository. However, you should not abuse this process and rely only on automated tests. Before each commit, you should run the tests yourself on your machine to ensure that the code you are adding to the repository is fully functional.

It should also be borne in mind that in some laboratories, automated testing may do not connect to the repository immediately. If the tests do not work within 24 hours after creating the repository, you should contact the teacher by email or in person. In case When sending email requests, please include your full name, group number, and repository link.

Step 6. Cloning the repository

The window with an open repository on GitHub is shown in Figure 3. To complete the lab assignment, the repository must be cloned (create a local working copy) into the user's operating system. Further work will be done with this local working copy of the repository.

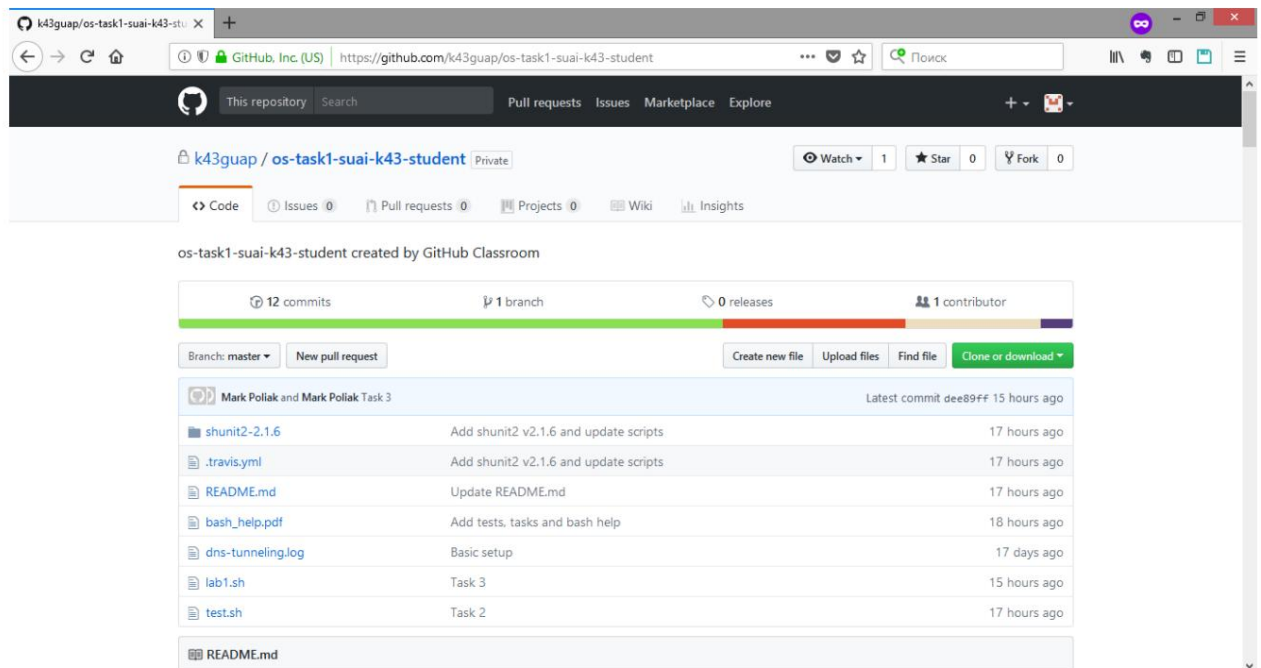


Figure 3. Repository with the task for the laboratory work

To clone a repository, you need to click the green “Clone or download” button on the main page of the repository on GitHub, as shown in Figure 4. In the pop-up window that opens, you need to copy the link by clicking the “Copy to clipboard” button.

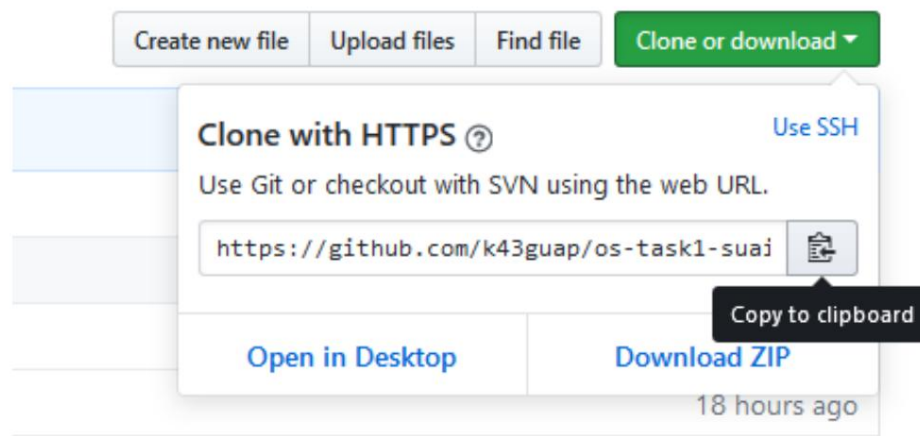


Figure 4. Getting a link to clone a repository

Next, you need to open a terminal (command interpreter) in the OS where you will create a copy of the repository. In the terminal, you should change the current directory to the one where the repository copy will be located, and then enter the command **git clone <repository_link>**. Instead of **<repository_link>**, you should paste the link you just copied from the repository page on GitHub. When executing the repository cloning command,

Depending on your settings, you may be asked for your username and password for your account on GitHub, after which the files will start copying. An example of executing the specified command in the terminal window of the Ubuntu OS is shown in Figure 5.

```

Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-97-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
markpolyak@ds:~$ mkdir lab1
markpolyak@ds:~$ cd lab1
markpolyak@ds:~/lab1$ git clone https://github.com/k43guap/os-task1-suai-k43-student.git
Cloning into 'os-task1-suai-k43-student'...
Username for 'https://github.com': m.polyak@guap.ru
Password for 'https://m.polyak@guap.ru@github.com':
remote: Counting objects: 85, done.
remote: Compressing objects: 100% (66/66), done.
remote: Total 85 (delta 23), reused 79 (delta 17), pack-reused 0
Unpacking objects: 100% (85/85), done.
Checking connectivity... done.
markpolyak@ds:~/lab1$

```

Figure 5. Cloning a repository

Step 7. Working with a local copy of the repository

After the repository has been cloned to the local machine (a local copy of the repository has been created), a folder with the name of the repository will appear in the current directory. This is where the local copy of the repository is located. You need to go to this folder by running the command **cd**

<repository_name>.

Next, you should complete the lab assignment by editing the files in your local copy of the repository and adding new ones if necessary. Once you have completed the assignment, you should save the changes you made to your local copy.

repository by running the commands:

git add <space-separated_list_of_files>

git commit

The first command, **git add ...**, specifies which files to save changes to in the repository. The second command, **git commit**, saves changes (creates a “restore point”) in all previously specified files. The **git status** command can be used to check which files have been changed, deleted, or added since the last **git commit command**. When executing the **git commit** command, you will be asked to enter a text message describing the changes made to the files. The message should be short but informative. An example of executing the commands is shown in Figure 6.

```

markpolyak@ds:~/lab1$ cd os-task1-suai-k43-student/
markpolyak@ds:~/lab1/os-task1-suai-k43-student$ nano lab1.sh
markpolyak@ds:~/lab1/os-task1-suai-k43-student$ git add lab1.sh
markpolyak@ds:~/lab1/os-task1-suai-k43-student$ git commit
[master 19ddfdb] Update lab1.sh
Committer: Mark <m.polyak@guap.ru>
1 file changed, 2 insertions(+), 2 deletions(-)

```

Figure 6. Working with a local copy of the repository

Step 8. Synchronize the local copy of the repository with GitHub.

After successfully executing the **git commit** command, you need to synchronize your local copy of the repository with the main repository on GitHub. To do this, use the **git push** command. When you execute it, you may again be asked to enter your GitHub account username and password. An example is shown in Figure 7.

```
markpolyak@ds:~/lab1/os-task1-suai-k43-student$ git push
Username for 'https://github.com': m.polyak@guap.ru
Password for 'https://m.polyak@guap.ru@github.com':
Counting objects: 3, done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/k43guap/os-task1-suai-k43-student.git
   dee89ff..19ddfdb  master -> master
markpolyak@ds:~/lab1/os-task1-suai-k43-student$
```

Figure 7. Synchronizing a local copy of a repository with GitHub

Step 9. Automatic assembly and testing

After synchronizing the local version of the repository with GitHub, the tests should run automatically. In the main window of the repository on GitHub, you need to open the "commits" section, as shown in Figure 8.

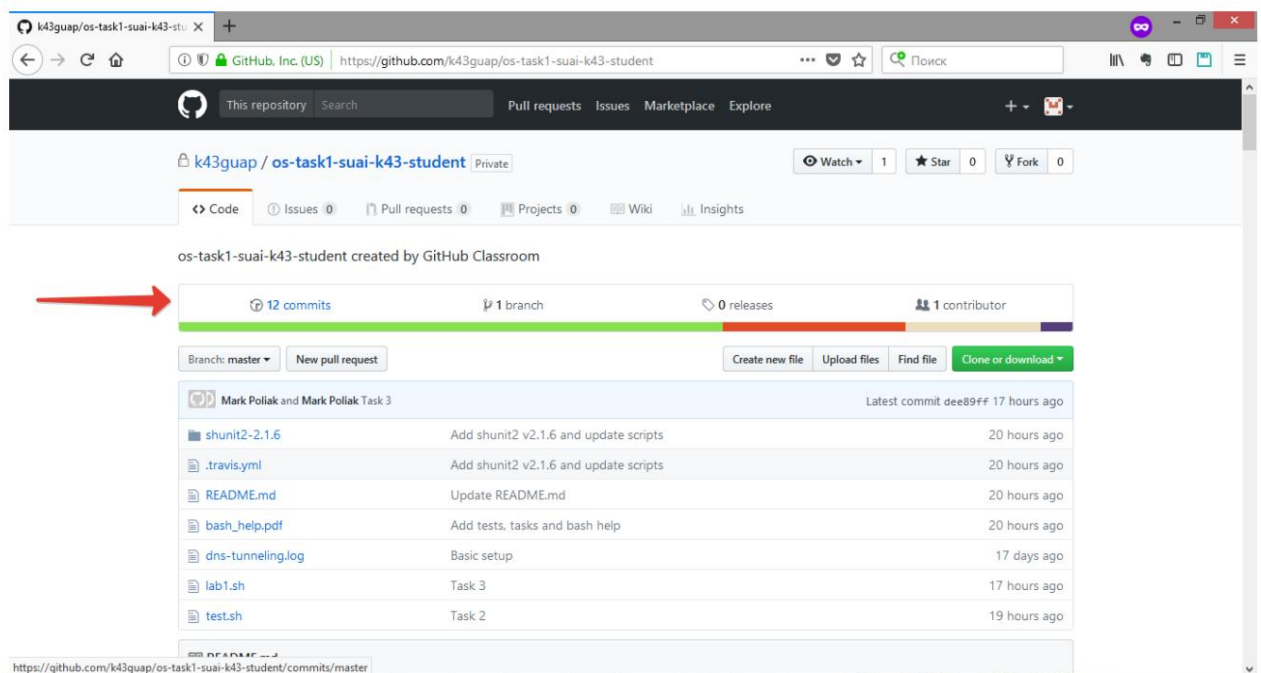


Figure 8. The "commits" section

The "commits" section of a repository on the GitHub website displays all commits ever made to the repository by any user. The most recent commits are displayed at the top. Figure 9 shows the test results for three commits. The red cross under the commit text description means that the code in this commit failed the tests. The green check mark there means that the tests passed successfully. In this case, you can start preparing the lab report. The yellow circle means that the tests have been started and are still running, in which case you should wait. Finally, if within five

minutes opposite the text description of the last commit, none of the three graphic symbols listed above appeared (including the yellow circle), this means that automatic tests are not connected to the repository. In this case, you should wait, and if there is a long delay in connecting automatic testing, you should contact the teacher in accordance with the algorithm from step 5.

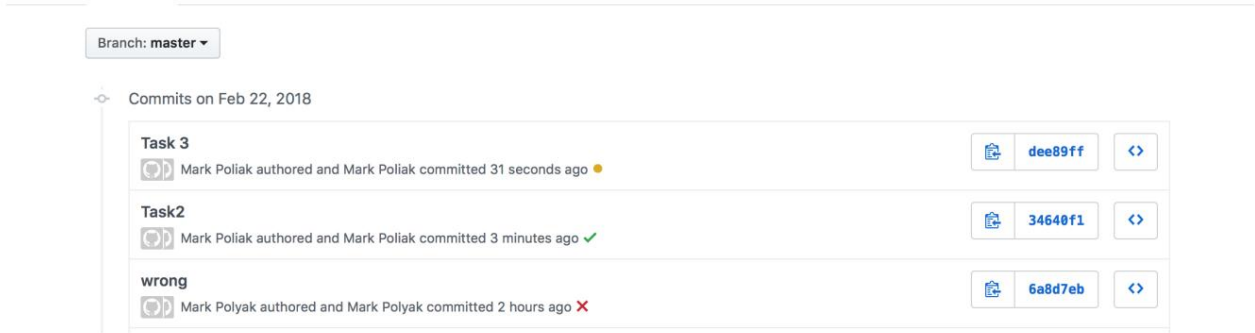


Figure 9. Automatic test execution