This note describes the step-by-step instructions for setting up the OLab4 docker container on a host system. Note that there are a lot of files to download and compose in the following steps. This will take about 30-45 minutes, depending on your hardware.

### Prerequisites:

* Windows/Mac/Linux host system
* Docker CE installed on host system
* Kitematic installed on the host system (can be installed via Docker)
* Read-access to the OLab4 git repository ([contact us](https://openlabyrinth.ca/support-request/) for permission)
* GIT Bash shell installed on the host system (if Windows, part of GIT for Windows install; if Mac, use Terminal.app)
* A reasonably fast network connection

### Directions:

1. IMPORTANT: ensure that the host system does not have existing applications listening for connections on the following ports: 80, 443, 3306.  Disable/stop these programs before continuing.
2. Edit the host system 'hosts' file and add the following entry:

127.0.0.1 olab4.localhost

1. Verify you can ping olab4.localhost from a host system command line and that the name resolves to the IP address above.  
     
   **$ ping olab4.localhost**
2. Start Docker on the host system
3. Create a 'docker' directory on the host system (preferably with full permissions to the host system logged in user).

(Windows) it is recommended to use the logged in user’s ‘Documents’ directory to avoid permissions issues.

(MacOS) the recommended configuration is to create a ‘docker’ directory under the user home directory, then create a subdirectory specific to the container to create. This allows for multiple containers to be created side-by-side.

This is now the '*container root directory*'.

1. Clone the OLab4 repositories into the container root directory.  Using a host command shell, execute the following command lines (from within the container root directory):

$ **git clone --single-branch --branch 4.1dev** [**https://github.com/olab/OLab4.git**](https://github.com/olab/OLab4.git)  
$ **git clone --single-branch --branch 4.1dev https://github.com/olab/OLab4-api.git**

1. There should be two directories created:

$ ls -l  
total 8  
drwxr-xr-x 1 wirunc 197609 0 Feb 28 22:52 OLab4/  
drwxr-xr-x 1 wirunc 197609 0 Feb 28 22:53 OLab4-api/

1. Change into OLab4/docker directory below your Docker root directory.  Verify the file 'docker-compose.yml' exists in this directory.
2. Execute the following command to create the 'olab4-developer' docker container. If there is an existing container of the same name, then it is overwritten.

$ **docker-compose up -d**

Depending on the speed of the internet connection, those many, many steps may take quite some time. (If it is completed in just a few seconds, you may have an old Docker container on your computer that has not been completely removed.)

Error messages in red that appear during the package install are normal, and can be ignored. Occasonally, the composer will appear to have finished, with a “Complete!” message and nothing happening. Be patient -- wait until you are returned to the usual CLI prompt. The last message should be

Creating olab4-developer ... done

1. When the creation has completed, open Kitematic (via the Docker context menu) and verify the container is running. (You might need to View | Refresh container list) If the container creation was successful, you should see the following two log file entries that signify that the container is running:

2017-11-29 12:28:35,941 INFO success: httpd entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)

2017-11-29 12:28:35,941 INFO success: mariadb entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)

1. Open a command prompt. Execute the following command to invoke a command shell hosted WITHIN the container:

$ **docker exec -it olab4-developer bash**

(Windows) If GIT Bash was installed with an alternate command console, you may need to prefix the command as such:

$ **winpty docker exec -it olab4-developer bash**

The window should respond with a different command prompt, similar to this:

[root@aa0a1928378c /]#

1. Verify that necessary file shares to the host system are configured properly and are operational. These directories should exist and have file listings similar the the following:

[root@ aa0a1928378c /]# **ls -l /var/lib/mysql**

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:45 ./

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:18 ../

-rw-r--r-- 1 wirunc 197609 350 Feb 28 12:18 6f270241ccb5-slow.log

-rw-r--r-- 1 wirunc 197609 16384 Feb 28 12:18 aria\_log.00000001

-rw-r--r-- 1 wirunc 197609 52 Feb 28 12:18 aria\_log\_control

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:45 entrada/

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:45 entrada\_auth/

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:45 entrada\_clerkship/

-rw-r--r-- 1 wirunc 197609 5242880 Feb 28 12:48 ib\_logfile0

-rw-r--r-- 1 wirunc 197609 5242880 Feb 28 12:48 ib\_logfile1

-rw-r--r-- 1 wirunc 197609 480247808 Feb 28 12:48 ibdata1

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:18 mysql/

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:48 openlabyrinth/

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:18 performance\_schema/

drwxr-xr-x 1 wirunc 197609 0 Feb 28 12:18 test/

[root@ aa0a1928378c /]# **ls -l /var/www/vhosts/**  
total 0  
drwxrwxrwx 2 root root 0 May 16 2018 OLab

[root@ aa0a1928378c /]# **ls -l /etc/httpd/vhosts.d/**  
total 1  
-rwxr-xr-x 1 root root 625 Apr 27 2018 olab4.dev.conf

1. Run the following post-setup commands within the container to download and create the OLab4 demo databases, set up apache, and connect file shares from the host system:

[root@aa0a1928378c /]# **cd /tmp**

[root@aa0a1928378c /]# **./post-create.sh**

If, during the post-create step, you see the following message…

Cloning failed using an ssh key for authentication, enter your GitHub credentials to access private repos  
Head to … to retrieve a token. It will be stored in "/root/.composer/auth.json" for future use by Composer.  
Token (hidden):

…this means that a token is required. [Contact the OLab4 github repository manager](https://openlabyrinth.ca/support-request/) for the token.

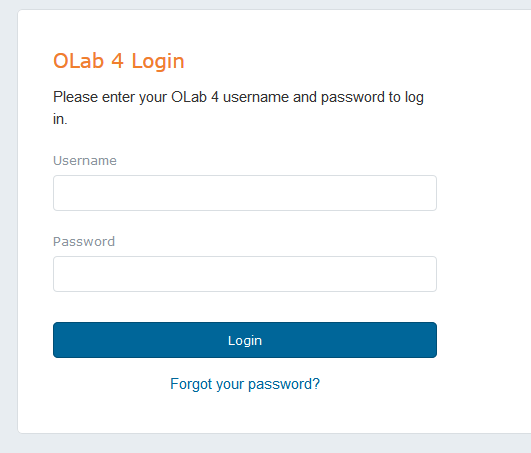
Again, many steps will run as part of this process, with some blank pauses. Be patient. If there are any serious errors, screen shot them and report them to OLab managers.

1. Open your favorite browser window and navigate to:

<https://olab4.localhost/apidev/olab>

It is safe to ignore warnings from the browser that this site is not really secure. You are only accessing it on your own machine, after all.

You should see a screen like this…



The above login dialog will appear it the container is configured properly.

1. Log into Olab4 with the ‘admin’ credentials (provided separately). Remember to remove this default account if you are creating a public installation.