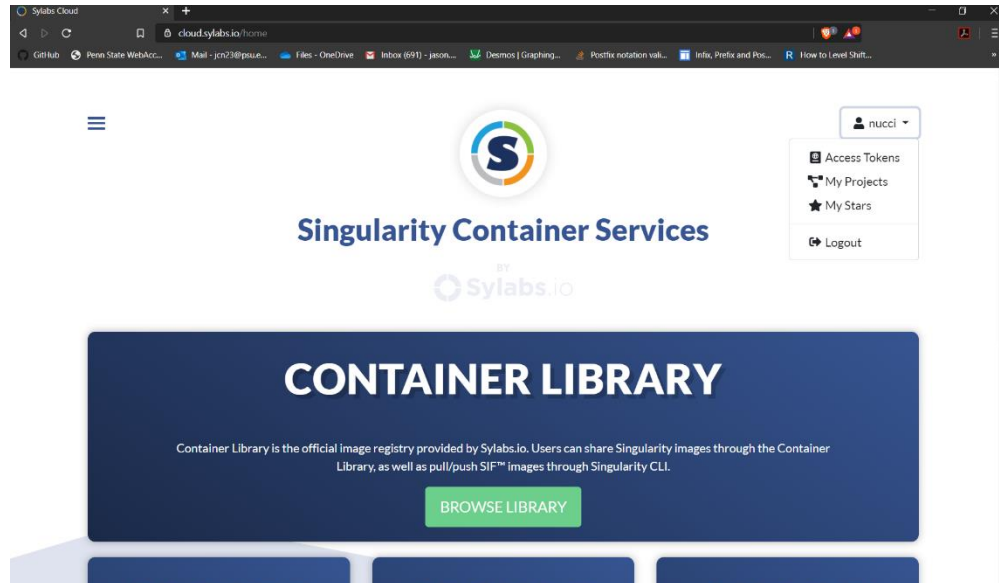


## Pushing Singularity Images from the Command Line to the Cloud

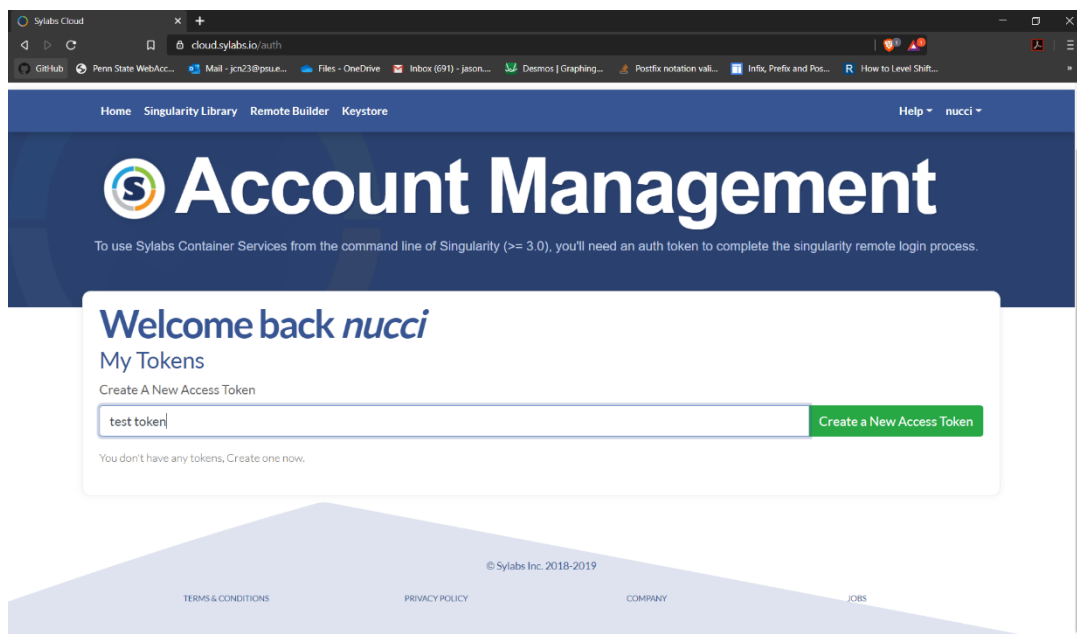
Assuming that you already have Singularity installed on your Linux machine and you have already made an account with <https://cloud.sylabs.io>, this is a step-by-step guide on how to push a Singularity image that you have made on your local machine up to the cloud.

### Setting up access on your computer:

1. First, go to <https://cloud.sylabs.io> and login
2. After logging in, click on your login id and select “Access Tokens”



3. After selecting “Access Tokens,” enter a name for your new token and then select “Create a New Access Token”



4. After creating your new token, click “Copy token to Clipboard”
5. Once you have copied the token, open Terminal and enter the command:

```
$ singularity remote login
```

After entering the command, you will be prompted to enter your access token. Right-click, select “Paste,” and then hit Enter. You are now ready to start pushing images to the cloud!

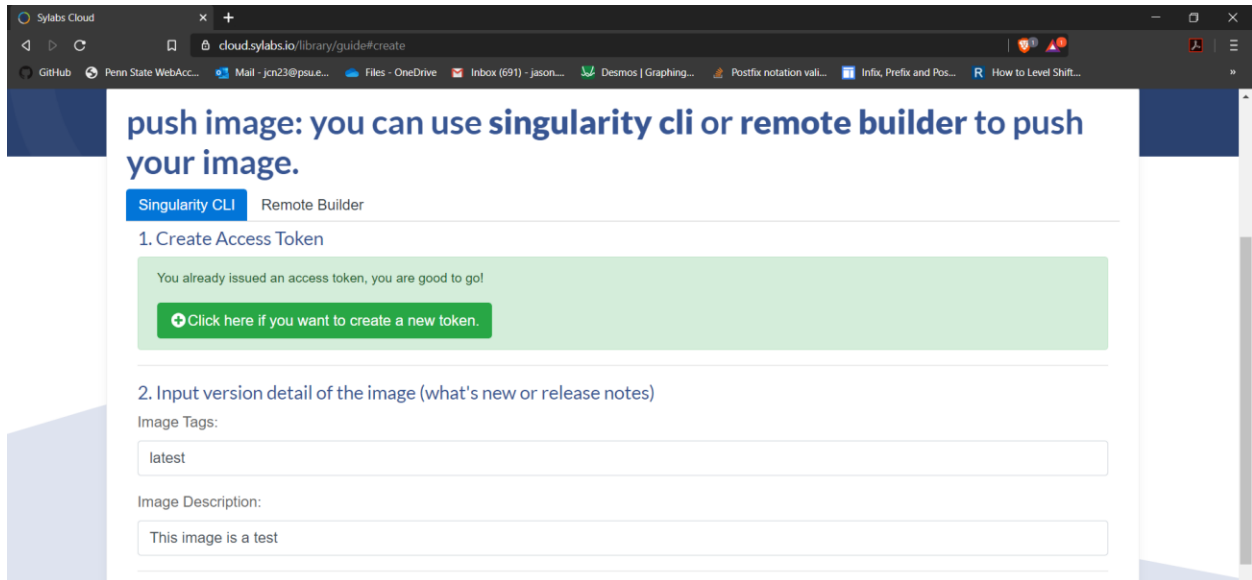
## Creating a new project

1. Go to <https://cloud.sylabs.io> and selecting “Browse Library”
2. Now, select “Create a new Project”
3. You will be taken to the “Create a Project” page. Proceed through the following steps:
  - a. Specify an owner of the project (usually yourself)
  - b. Specify a name for the project (be specific)
  - c. Write a short description about the container
  - d. Select whether you want the container to be public or private
  - e. Check or uncheck the “Initialize the project with a markdown README” (You should check this box if more than one user is going to be using this container)

The screenshot shows the 'Create a Project' interface on the Sylabs Cloud website. The form is white and centered on a dark blue background. It includes a navigation bar at the top with links to Home, Singularity Library, Remote Builder, and Keystore. The main heading is 'Create a Project' with the Sylabs logo. The form fields are: 'Owner' (a dropdown menu showing 'nucci'), 'Project name' (a text input field containing 'test\_container'), and 'Description(optional):' (a text input field containing 'This is a test'). Below these fields are two radio buttons for project visibility: 'Public' (selected) and 'Private'. At the bottom of the form is a checkbox for 'Initialize the project with a markdown README.' and a 'Next' button. The footer of the page contains copyright information '© Sylabs Inc. 2018-2019' and links for 'TERMS & CONDITIONS', 'PRIVACY POLICY', 'COMPANY', and 'JOBS'.

4. You will now be taken to the “push image” page. Proceed through the following steps:
  - a. You will be prompted to create an access token. You can ignore this step if you already made one earlier

- b. Specify a tag for your image (latest, v1.0.0, etc.)
- c. Write a short description about the image (Specify what it does, what you changed, how you configured it, etc.)



The screenshot shows a web browser window with the URL `cloud.sylabs.io/library/guide#create`. The page title is "push image: you can use singularity cli or remote builder to push your image." There are two tabs: "Singularity CLI" (selected) and "Remote Builder".

**1. Create Access Token**

You already issued an access token, you are good to go!

[Click here if you want to create a new token.](#)

**2. Input version detail of the image (what's new or release notes)**

Image Tags:

Image Description:

## Pushing image from the Command Line

1. Before you can push your image up to the cloud you need to sign the image first. This process is relatively easy, however, for a first-time user it can be difficult. Use the following steps to create a key and sign your image:
  - a. First, enter the following command:

```
$ singularity key newpair
```
  - b. You will now be prompted to create a new key. Think of a key as your way of personalizing your image; it lets other users know that you created it. Use the following steps to create your own key:
    - i. Enter your name (i.e. John Doe)
    - ii. Enter your email address
    - iii. Enter an optional comment for the key (test, devel, release, etc.)
    - iv. Create a passphrase for your key (make it something that is easier to remember as you will not be able to change it later)

- v. Push your key to the keystore (Stores your keys on the cloud so that you can access them from multiple machines. Should you choose not to push the key to the store then it will only be available on your local machine)

```
nucci@localhost:~/tmp
File Edit View Search Terminal Help
(base) [nucci@localhost tmp]$ singularity key newpair
Enter your name (e.g., John Doe) : Jason Nucciarone
Enter your email address (e.g., john.doe@example.com) : nucci.programming@gmail.com
Enter optional comment (e.g., development keys) : test keys
Enter a passphrase :
Retype your passphrase :
Would you like to push it to the keystore? [Y,n] y
Generating Entity and OpenPGP Key Pair... done
Key successfully pushed to: https://keys.sylabs.io
(base) [nucci@localhost tmp]$
```

- vi. Should you ever want to see what keys you have pushed up to the cloud, don't go to <https://keys.sylabs.io>. You will receive a 404 HTTP error. Instead, go to <https://cloud.sylabs.io/keystore>

## Search For Keys

🔍 Keys for current signed in user(can search by name or email or fingerprint) ☒ Active ☒ Expired ☒ Expiring

Keys					
Name ^	Creation Time	Expiration Time	Algorithm	Bits	Fingerprint
Jason Nucciarone (test) <nucci.programming@gmail.com>	2020-01-09 12:11:42	Never	rsa	4096	628E6FFCF854B5...
nucci <nucci.programming@gmail.com>	2019-11-07 02:37:55	Never	rsa	4096	500C83E269A9E3...

- vii. Should you ever want to see the keys that you have stored locally, simply enter the command:

\$ singularity key list

```
nucci@localhost:~/tmp
File Edit View Search Terminal Help
(base) [nucci@localhost tmp]$ singularity key list
Public key listing (/home/nucci/.singularity/syppg/pgp-public):

0) U: Jason Nucciarone (test keys) <nucci.programming@gmail.com>
   C: 2020-01-09 00:57:28 -0500 EST
   F: A1317238621A1C187A5D7950419EDB8969C3D833
   L: 4096
   -----
(base) [nucci@localhost tmp]$
```

- c. Now that you have created your new key, you will now be able to sign your images using the following command:  
`$ singularity sign myImage.sif`
- d. You will be prompted to select which key you want to use. Simply insert the integer value and hit Enter. After that, enter the passphrase for the key and hit Enter again. Congratulations! You signed your container!

```
nucci@localhost:~/tmp
File Edit View Search Terminal Help
(base) [nucci@localhost tmp]$ singularity sign example.sif
Signing image: example.sif
0) U: Jason Nucciarone (test) <jcn23@psu.edu>
   C: 2020-01-09 00:06:23 -0500 EST
   F: EC65206C9920A931DCD6434A3682FAE166B58136
   L: 4096
   -----
1) U: Jason Nucciarone (test) <nucci.programming@gmail.com>
   C: 2020-01-09 00:11:42 -0500 EST
   F: 628E6FFCF854B5126642BBFD79BFD6406FDDFF80E
   L: 4096
   -----
2) U: Jason Nucciarone (test keys) <nucci.programming@gmail.com>
   C: 2020-01-09 00:57:28 -0500 EST
   F: A1317238621A1C187A5D7950419EDB8969C3D833
   L: 4096
   -----
Enter # of private key to use : 2
Enter key passphrase :
Signature created and applied to example.sif
(base) [nucci@localhost tmp]$
```

2. Now that we have gotten signing out of the way, it is time to push your image to the cloud. Use the following command to push your image to the cloud:  
`$ singularity push myImage.sif library://path/to/project`

```
nucci@localhost:~/tmp
File Edit View Search Terminal Help
(base) [nucci@localhost tmp]$ singularity push example.sif library://nucci/default/test_project:test
INFO: Container is trusted - run 'singularity key list' to list your trusted keys
64.90 MiB / 64.90 MiB [=====] 100.00% 2.36 MiB/s 27s
(base) [nucci@localhost tmp]$
```

And here is what the final project should look like in the end!

The screenshot displays the Syllabs Cloud web interface. At the top, the browser address bar shows the URL `cloud.syllabs.io/library/nucci/default/test_project`. The main header area features the Syllabs logo, the project name **nucci/default/test\_project**, and a sub-header indicating it was created 98m 18s ago. Below this, a terminal-like box contains the command `singularity pull library://nucci/default/test_project`. A status message "This is a test" is visible, along with buttons for "Push A New Image" and "Edit".

Below the header, a dropdown menu for "Architecture" is set to "All Architectures". The main content area displays a card for the project **test\_project : test**. To the left of the card is a 3D cube icon labeled "amd64". The card includes a table of metadata:

CREATED AT:	2020-01-11 01:15:45
UNIQUE ID:	sha256:6365b438cf846baa02cb9ccd4beb5ff6fa7da503a289d5fb6c7eef61a1332896
IMAGE SIZE:	64.90 MB
ARCHITECTURE:	amd64
FINGERPRINTS:	A1317238621A1C187A5D7950419EDB8969C3D833
RELEASE NOTES:	No Description

At the bottom of the card, there are three buttons: "DOWNLOAD" (with a download icon), "SHOW PULL CMD" (with an eye icon), and "Delete this version" (with a trash icon). An "Edit" button is also present on the right side of the card.