



# Singularity Container Services

---

Adam Hughes  
adam@sylabs.io

---

## Introduction

---



Adam Hughes  
Software Architect

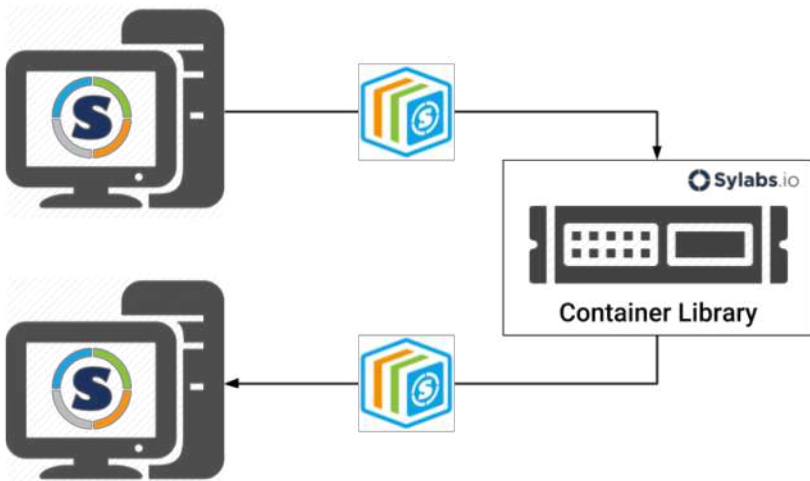
---

# Singularity Container Services Overview

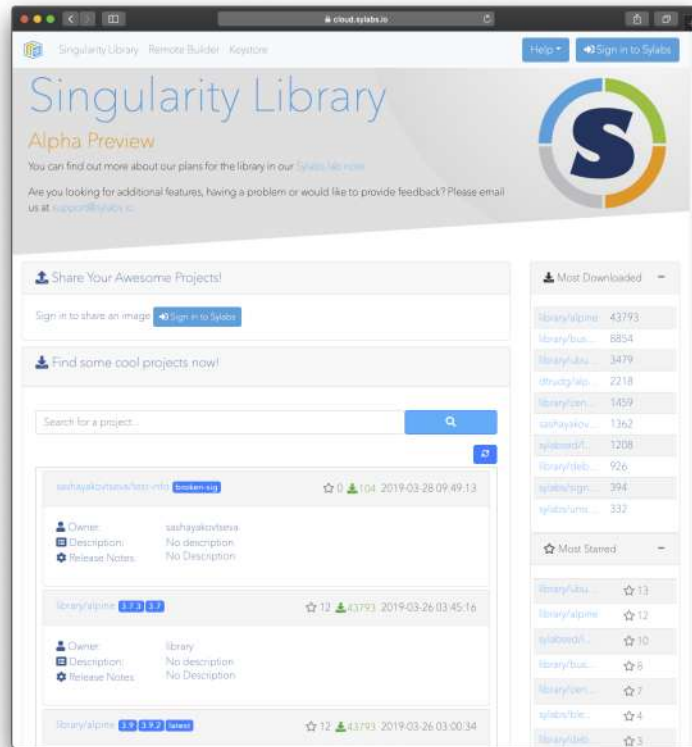
- The Singularity Container Services are a series of value added container services:
  - Container Library
  - Remote Build Service
  - Key Server
- Available in preview now
  - <https://cloud.sylabs.io>

# Singularity Container Library

```
$ sudo singularity build <local_path> <def>  
$ singularity push <local_path> library://<path>
```



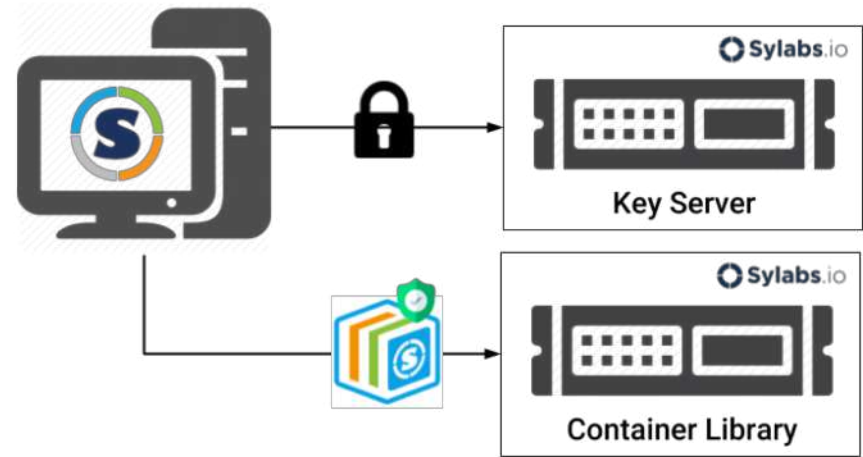
```
$ singularity pull library://<path>
```



## Signing a SIF Image

- The `sign` command adds a digital signature block to the SIF image
- The signature block ensures the integrity of the contents of the image
- The Sylabs Key Server lets you easily share your public key

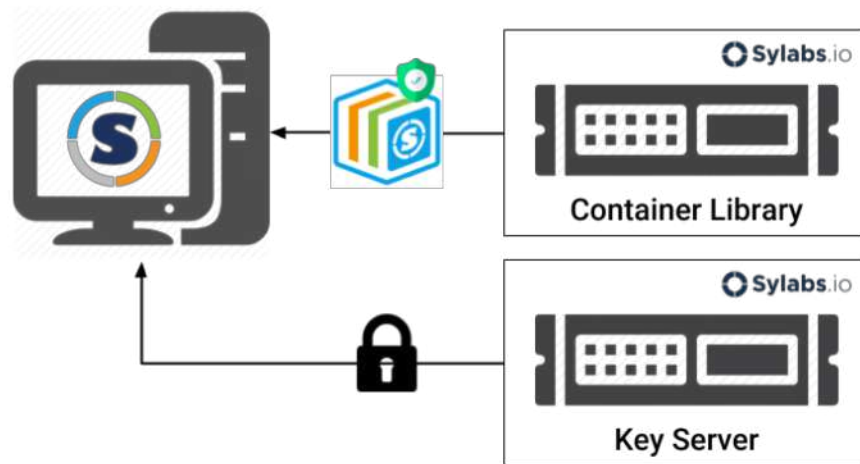
```
$ singularity sign image.sif  
$ singularity push image.sif library://<path>
```



## Verifying a SIF Image

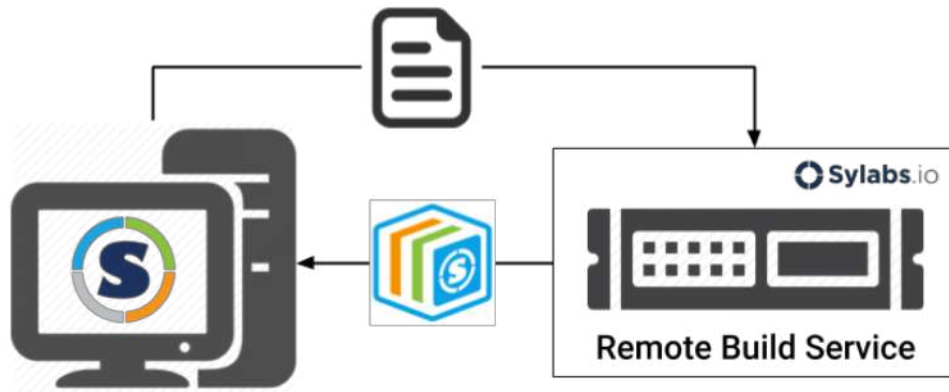
- The `verify` command cryptographically verifies the digital signature in the SIF image
- If the public key material is not in the local keyring, it fetches it from the Key Server

```
$ singularity pull alpine
$ singularity verify alpine_latest.sif
Verifying image: alpine_latest.sif
Data integrity checked, authentic and signed by:
    Sylabs Admin <support@sylabs.io>,
Fingerprint
8883491F4268F173C6E5DC49EDECE4F3F38D871E
```



- The Remote Build Service allows you to build without privilege
- Build output is live-streamed to the Singularity CLI or web browser
- When using the CLI, the image is pulled back to the workstation

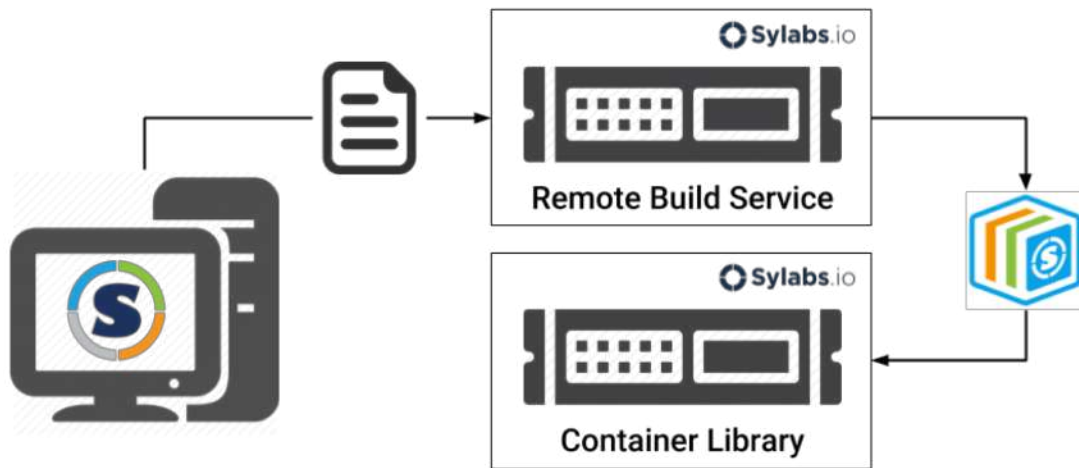
```
$ singularity build --remote <local_path> <def>
```





- The Remote Build Service can push directly to the Library
- Optionally, you can use `--detach` to start the build without streaming console output
- Build output can be retrieved later via a web browser

```
$ singularity build --remote library://<path> <def>
```



---

# SCS Front End Enhancements

# Website Internationalization

- Sylabs is a global company
- We are working hard to make our products reflect that
- We love community feedback!



The screenshot displays the Sylabs.io website interface, which is localized into Japanese. At the top, the navigation bar includes 'Singularity Library', 'Remote Builder', and 'Keystore', along with 'Help' and 'Sign in to Sylabs' buttons. The main content area features several service tiles:

- Secure**: Represented by a document icon with a checkmark and a pen. Text: '電子署名で信憑性と一致性を確保。' (Ensure reliability and consistency with electronic signatures.)
- Create [alpha preview]**: Represented by a factory icon. Text: 'クラウドでSingularity ContainerをBuildする。' (Build Singularity Container in the cloud.)
- Share [alpha preview]**: Represented by a ship icon. Text: 'Containerイメージを世界中に共用する。' (Share container images globally.)
- Container Library**: Represented by a shipping container icon. Text: 'Container LibraryがSylabs.ioから提供されたオフィシャルイメージレジストリ、ユーザがここでSingularityのイメージを検索・発表・ダウンロードができます。' (Container Library is an official image registry provided by Sylabs.io, where users can search, publish, and download Singularity images.)
- Remote Builder**: Represented by a hard hat and gears icon. Text: 'Remote Builderサービスを経由して、ユーザがクラウドで直接Singularityのイメージを作成することができます。ユーザが自分で環境などを用意されなくても問題ありません。' (Via the Remote Builder service, users can create Singularity images directly in the cloud. It's no problem even if users don't prepare their own environment.)
- Keystore Service**: Represented by a shield icon with a checkmark. Text: 'SingularityがSIFファイルフォーマットを提供して、そしたらユーザがGPGキーでイメージをサイン・検証することができます。Keystore Serviceがユーザのキーをクラウド中で管理するサービスになります。' (Singularity provides the SIF file format, so users can sign and verify images with GPG keys. Keystore Service is a service that manages users' keys in the cloud.)

At the bottom of the page, there is a footer with the text: '© Sylabs Inc. 2018-2019 Terms and Conditions Privacy Policy Company Jobs'.

# Website Recipe Editor!

- Edit Recipe Step by Step
- MultiStage support
- Built-in Stage (gcc/openmpi/etc..)
- Live documentation and editor output

The screenshot displays the Website Recipe Editor interface, divided into two main panels.

**Left Panel: Stage Configuration**

- Add New Stage:** A text input field contains 'prepare'.
- Use Built-in Stage:** A dropdown menu shows 'gcc'.
- +Add Stage:** A blue button to add a new stage.
- Stage List:** Shows 'Stage: prepare' and 'Stage: gcc', each with a red 'x' icon for deletion.
- Header:** A section for selecting the bootstrap agent, repository, and image.
- Stage gcc has 1 sections:** A blue '+Add Section' button.
- %setup:** A section with a red 'Delete Section' button.
- Section:** A dropdown menu showing '%setup'.
- Summary:** A dropdown menu showing 'install'.
- Code Editor:** A list of commands:

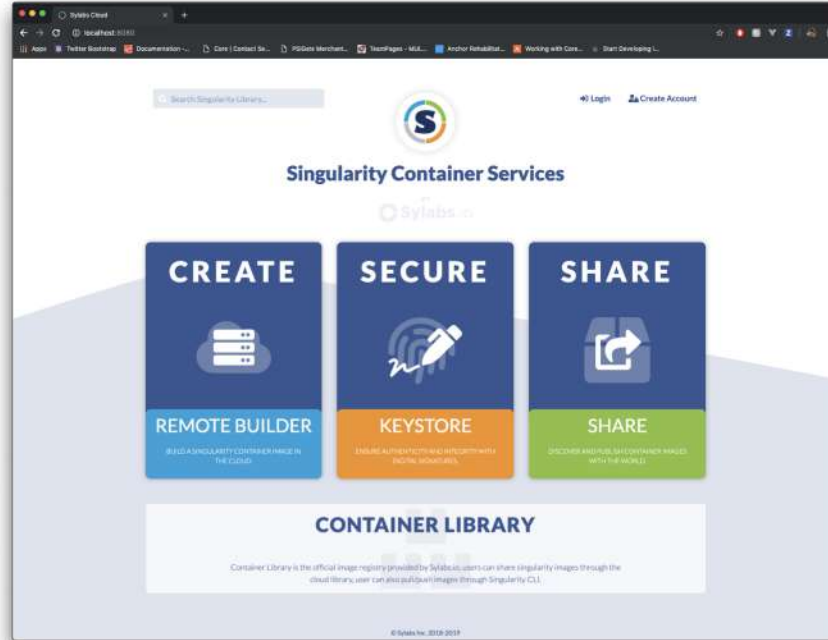
```
1 yum install -y \\  
2 gcc \\  
3 gcc-c++ \\  
4 gcc-gfortran && \\  
5 rm -rf /var/cache/yum*
```
- Help Documentation:** A section titled '%setup' with explanatory text.

**Right Panel: Build a Recipe File**

- Instructions:** 'Please attach build recipe by dragging & dropping, pasting from the clipboard or selecting them' with a blue 'selecting them' button.
- Code Editor:** A multi-line recipe file:

```
1 #####  
2 # prepare  
3 #####  
4 Bootstrap: docker  
5 From:  
6  
7  
8 #####  
9 # setup env  
10 #####  
11 %setup  
12 export PATH=$PATH:/usr/env  
13 #####  
14 # gcc  
15 #####  
16 Bootstrap: docker  
17 From: 18.04  
18  
19  
20 #####  
21 # install  
22 #####  
23 %setup  
24 yum install -y \\  
25 gcc \\  
26 gcc-c++ \\  
27 gcc-gfortran && \\  
28 rm -rf /var/cache/yum*  
29
```
- Build Recipe file is being checked:** A status message.
- Container Selection:** A dropdown menu showing 'jailpassion/default/'.
- Build:** A blue button to build the recipe.
- Search Containers:** A button with a magnifying glass icon.

# Website UX Enhancements



---

**Questions?**



# THANK YOU!

@SylabsIO

@SingularityApp

<https://www.sylabs.io>

---

Adam Hughes  
adam@sylabs.io