

# Introduction to Containers

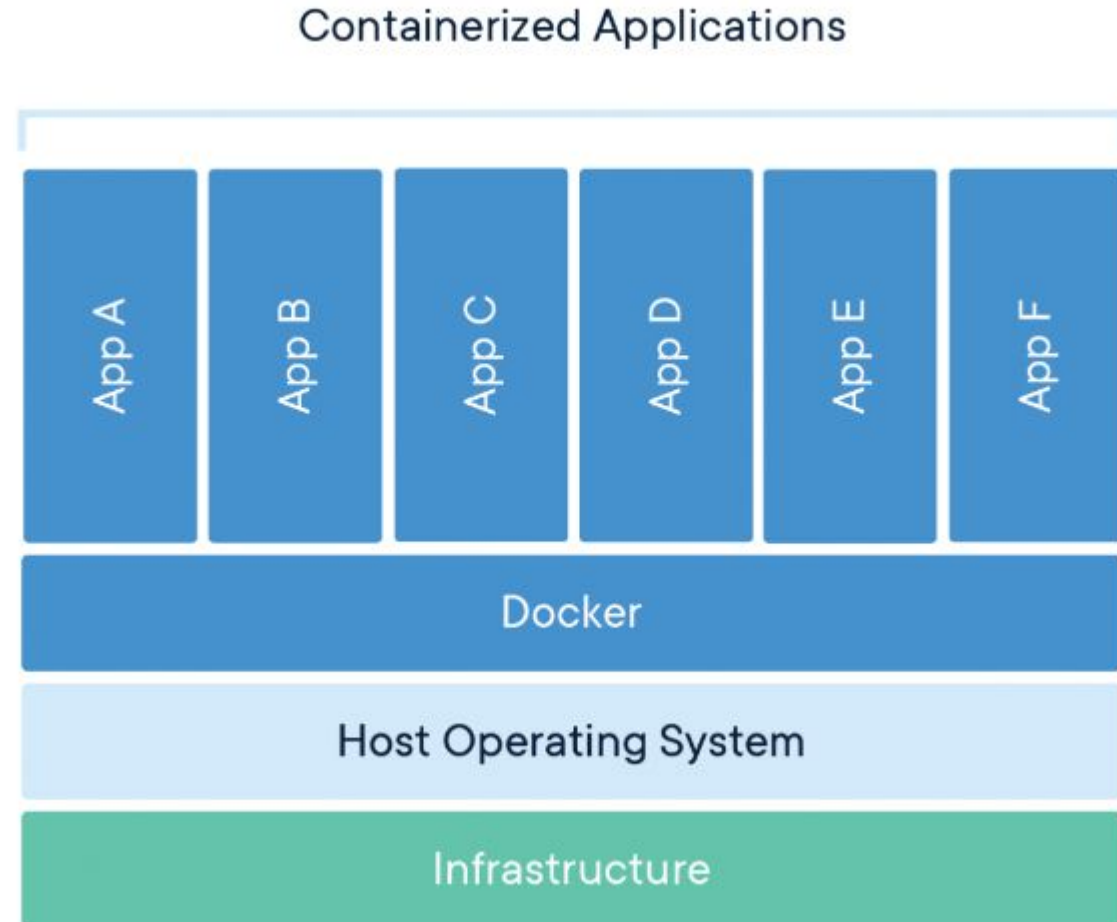
Lizel Potgieter

[lizel.potgieter@slu.se](mailto:lizel.potgieter@slu.se)

Department of Plant Breeding, SLU Alnarp

# What is a container?

- **Technical description:** A container is an object that contains an operating-system-level virtualisation
- **Practical description:** A container is a piece of software that contains an operating system, all of the libraries and dependencies, and a program (or several programs) that you want to run



# What are the advantages of containers?

- **Standardised**

- Containers will run the same way on any operating system
- Configuration profiles are consistent

- **Secure**

- Programs in containers are isolated from other programs
- Once your container is built or downloaded, it is immutable
- All containers are verifiable

- **Lightweight**

- You can stack operating systems on top of each other
- You don't need to install thousands of different dependencies (that might clash with each other)

Containers are the best way to ensure reproducible research



# Apptainer

We usually use Apptainer to run containers:

<https://apptainer.org/docs/user/main/index.html>

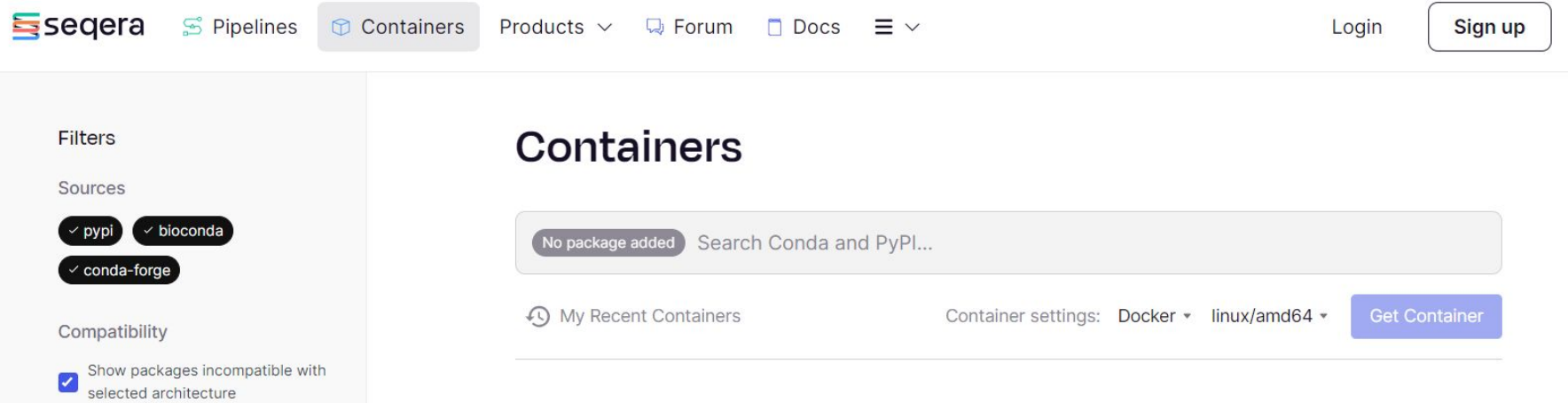
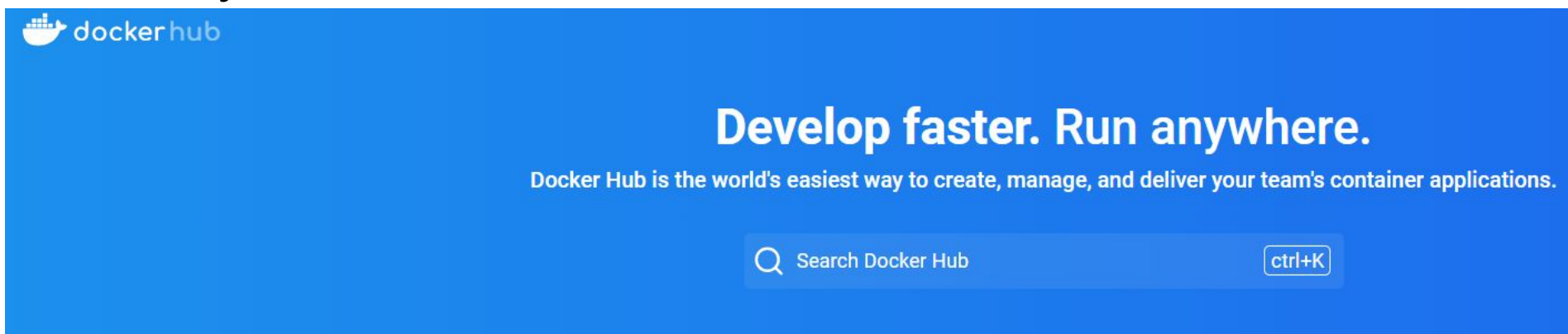
Apptainer is more secure than Docker (and is free to use!)

Basic commands we'll use

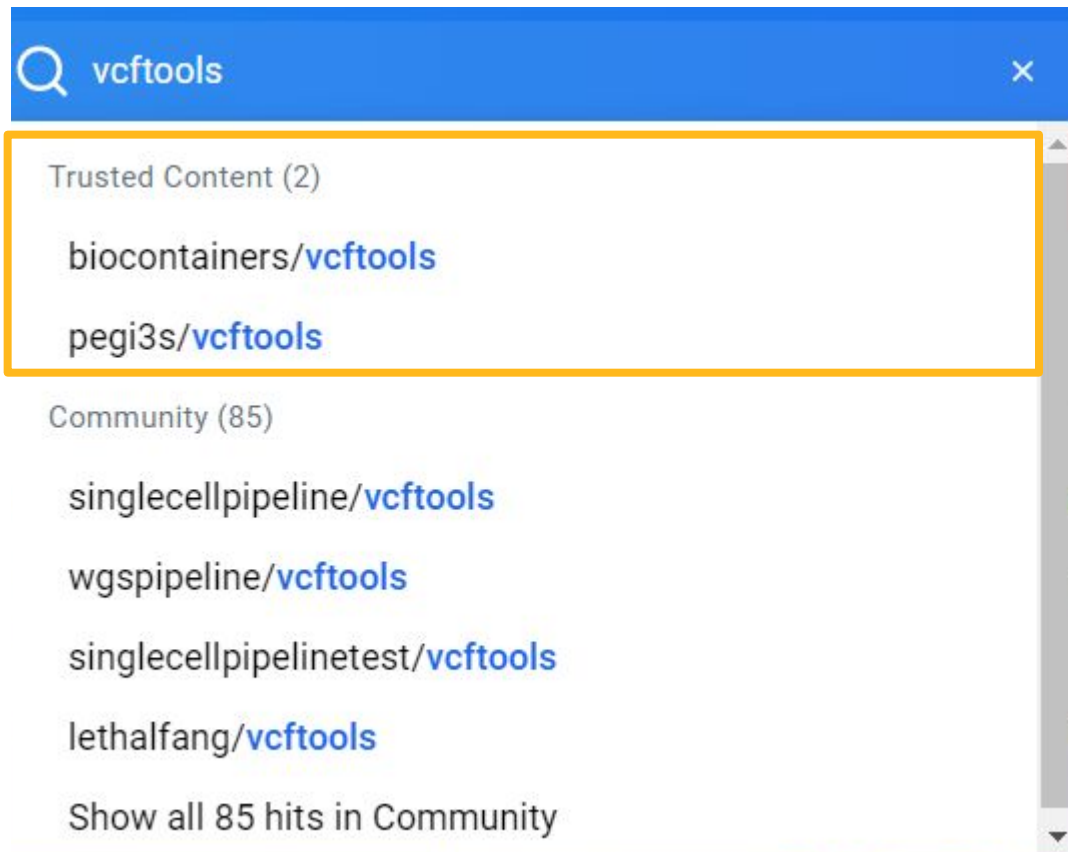
- `pull`: downloads containers from online registries
- `run`: runs containers
- `build`: build your own containers

# Where do we get containers?

- Several container registries exist
  - <https://hub.docker.com/>
  - <https://seqlera.io/containers/>
- Build your own



# Downloading from Docker Hub: VCFtools example



The main software we use are often built and verified (if possible, try to use the trusted content contributions)

biocontainers and pegi3s are the authors in this case

We will look at the version uploaded by biocontainers

# Downloading from Docker Hub: VCFtools example



**biocontainers/vcftools**  Sponsored OSS ☆5

↓ Pulls 100K+

By [biocontainers](#) · Updated almost 5 years ago

"tools written in Perl and C++ for working with VCF files"

[IMAGE](#)


DATA SCIENCE

MACHINE LEARNING & AI

**Overview**

Tags



The latest information about BioContainers is available via <https://BioContainers.pro> 

 [join chat](#)

## Containers

Repository of approved bioinformatics containers

## Links:

Web Page : <http://biocontainers.pro/> 

## Docker Pull Command

```
docker pull biocontainers/vcftools
```

[Copy](#)

pull command: `apptainer pull vcftools.sif docker://biocontainers/vcftools`





# Downloading from Docker Hub: VCFtools example

Overview

Tags

Sort by

Newest ▾

🔍 Filter Tags

TAG

[v0.1.16-1-deb\\_cv1](#)

Last pushed 5 years ago by [biocontainersci](#)

```
docker pull biocontainers/vcftools:v0.1.16-1-deb_cv1
```

Copy

Digest

OS/ARCH

Compressed Size ⓘ

[caa02f1a00f1](#)

linux/amd64

67.1 MB

TAG

[v0.1.14\\_cv2](#)

Last pushed 6 years ago by [biocontainersci](#)

```
docker pull biocontainers/vcftools:v0.1.14_cv2
```

Copy

Digest

OS/ARCH

Compressed Size ⓘ

[d533d59a2032](#)

linux/amd64

403.93 MB

TAG

[v0.1.15\\_cv2](#)

Last pushed 6 years ago by [biocontainersci](#)

```
docker pull biocontainers/vcftools:v0.1.15_cv2
```

Copy

Digest

OS/ARCH

Compressed Size ⓘ

[498473972dd3](#)

linux/amd64

449.09 MB





# Downloading from Seqera: VCFtools example

## Containers

No package added vcftools

My Recent Containers

Container settings: Docker ▾ linux/amd64 ▾

Get Container

+  
Add



bioconda::vcftools

0.1.16 ▾

A set of tools written in Perl and C++ for working with VCF files. This package only contains the C++ libraries whereas the... [Show more](#)

linux/amd64 • linux/arm64

+  
Add



bioconda::perl-vcftools-vcf

0.1.16 ▾

cpanm ready distribution of VCFtools Perl libraries

linux/amd64 • linux/arm64

+  
Add



vcftoolz

1.2.3 ▾

Tools for working with Variant Call Format files.

linux/amd64 • linux/arm64

# Downloading from Seqera: VCFtools example

## Containers

1 package bioconda::vcftools=0.1.16 X Search Conda and PyPI...

My Recent Containers Container settings: Singularity linux/amd64 Get Container

Added

Container is ready

Container image name

oras://community.wave.seqera.io/library/vcftools:0.1.16--7d05197fa40c2e4a

☒ HTTPS

[View build details](#)

+ Add

bioco cpanm ready distribution of VCFtools Perl libraries

linux/amd64 • linux/arm64

+ Add

vcftoolz 1.2.3

Tools for working with Variant Call Format files.

linux/amd64 • linux/arm64

Change Docker to Singularity

Wait until the container is ready

pull command:

```
apptainer pull vcftools.sif oras://community.wave.seqera.io/library/vcftools:0.1.16--7d05197fa40c2e4a
```



# How do we run our VCFtools container?

With the `apptainer pull vcftools.sif registry://author/tool` command we pulled an SIF image called `vcftools.sif`

To run it we use `apptainer run vcftools.sif vcftools --version` as we would if it was installed on our local system

You can run your software as you normally would!

```
VCFTools (0.1.16) containers$ apptainer run vcftools.sif vcftools --version
```



# Building your own containers

To build your own containers, you need a definition file

To build your definition file you will use

```
apptainer build container.sif library://definition_file
```

For more details: [https://apptainer.org/docs/user/main/cli/apptainer\\_build.html](https://apptainer.org/docs/user/main/cli/apptainer_build.html)



# Lolcow example: Building from scratch!

```
Bootstrap: docker
```

```
From: ubuntu:20.04
```

```
%post
```

```
apt-get -y update
```

```
apt-get -y install cowsay lolcat
```

```
%environment
```

```
export LC_ALL=C
```

```
export PATH=/usr/games:$PATH
```

```
%runscript
```

```
date | cowsay | lolcat
```

Paste this text into a file  
called lolcow.def

Then run: `apptainer build lolcow.sif lolcow.def`

```
< Fri Oct 18 11:26:00 CEST 2024 >
```

```
  \  ^__^
   \  (oo)\_______
      (__)\       )\/\
         ||----w |
         ||     ||
```

# Lolcow: Let's make it philosophical!

```
Bootstrap: docker
From: ubuntu:20.04
```

```
%post
    apt-get -y update
    apt-get -y install cowsay lolcat
```

```
%environment
    export LC_ALL=C
    export PATH=/usr/games:$PATH
```

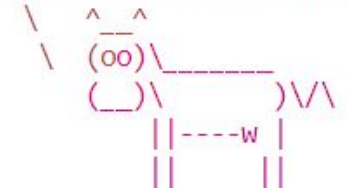
```
%runscript
    fortune | cowsay | lolcat
```

**fortune**

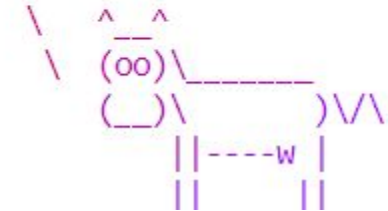
```
< Excellent day to have a rotten day. >
```



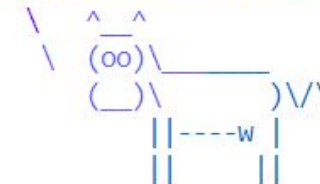
```
/ You will engage in a profitable \
\ business activity. /
```



```
/ You will have domestic happiness and \
\ faithful friends. /
```



```
/ Q: How many supply-siders does it take \
| to change a light bulb? A: None. The |
| darkness will cause the light bulb to |
\ change by itself. /
```







SCIENCE AND  
EDUCATION **FOR**  
**SUSTAINABLE**  
**LIFE**