



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



# Introduction to Containers and Docker

Alberto Madonna - CSCS

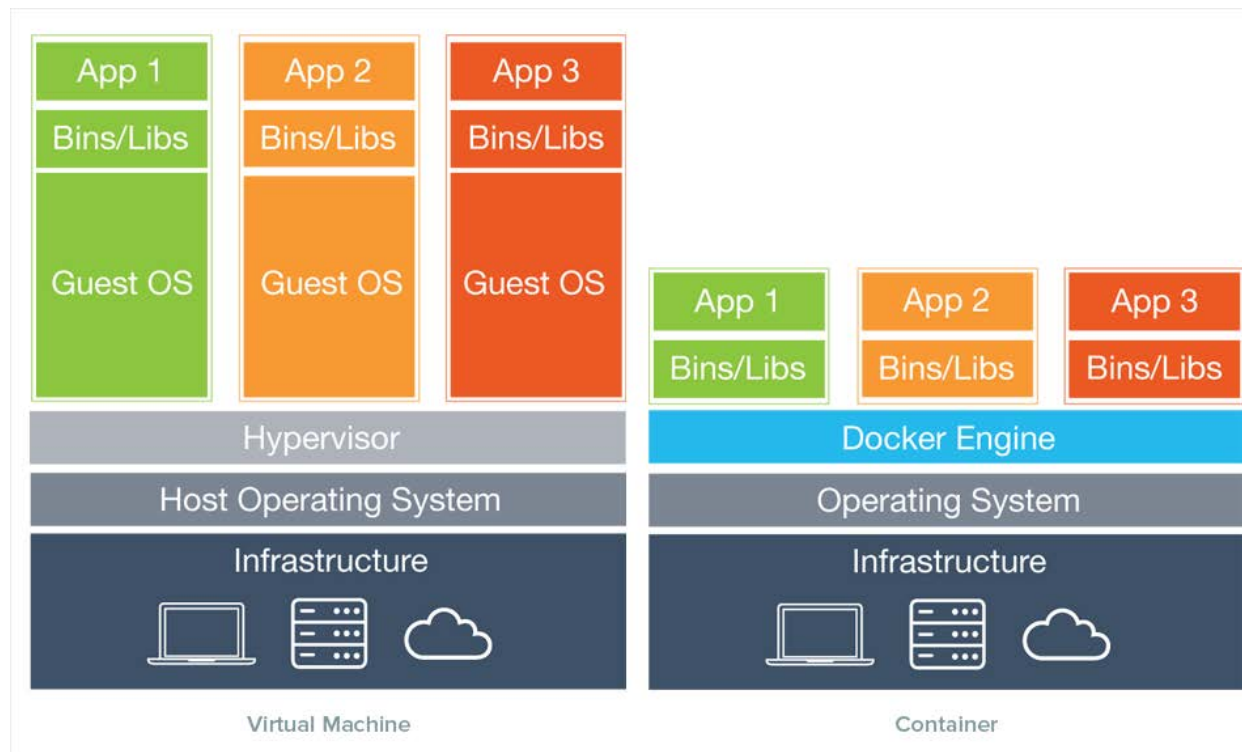
June 13<sup>th</sup>, 2018

# Table of Contents

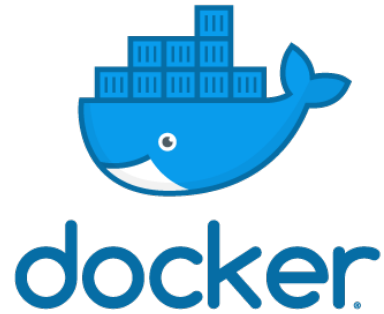
1. Introduction to containers and Docker (15 min)
  2. Live demo (30 min)
  3. Hands-on (90+ min)
- 
- Slides and code available at <https://github.com/eth-cscs/containers-hands-on>

# Containers

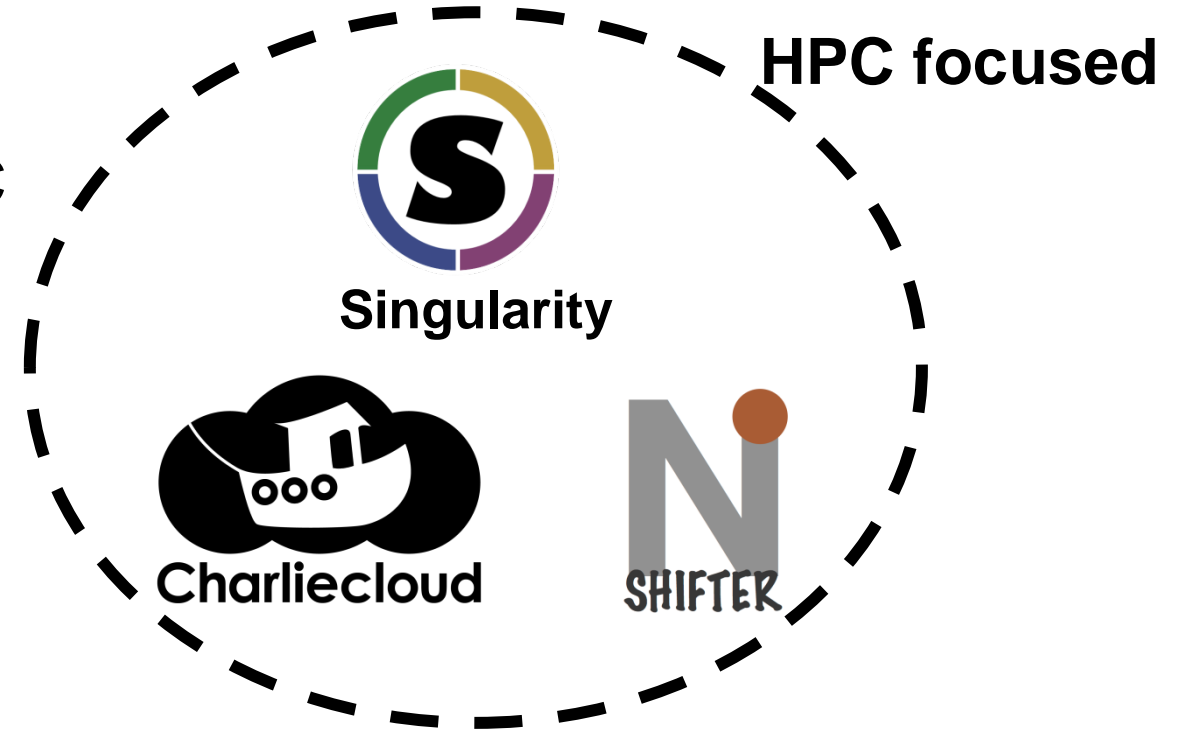
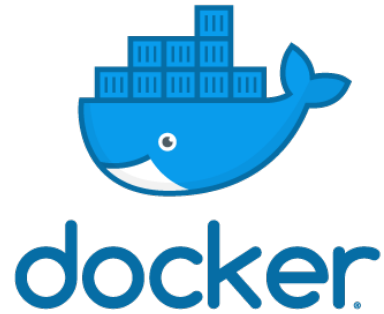
- Lightweight, isolated environments to run applications/services
- Already include all software dependencies
- Interest from HPC: a way to provide user-defined software stacks



# Container implementations

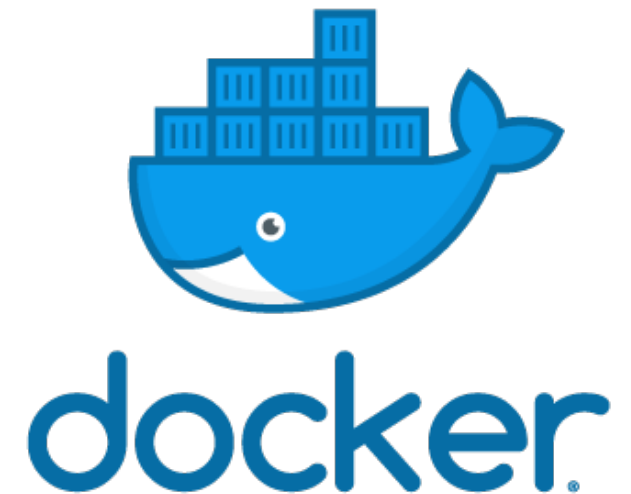


# Container implementations



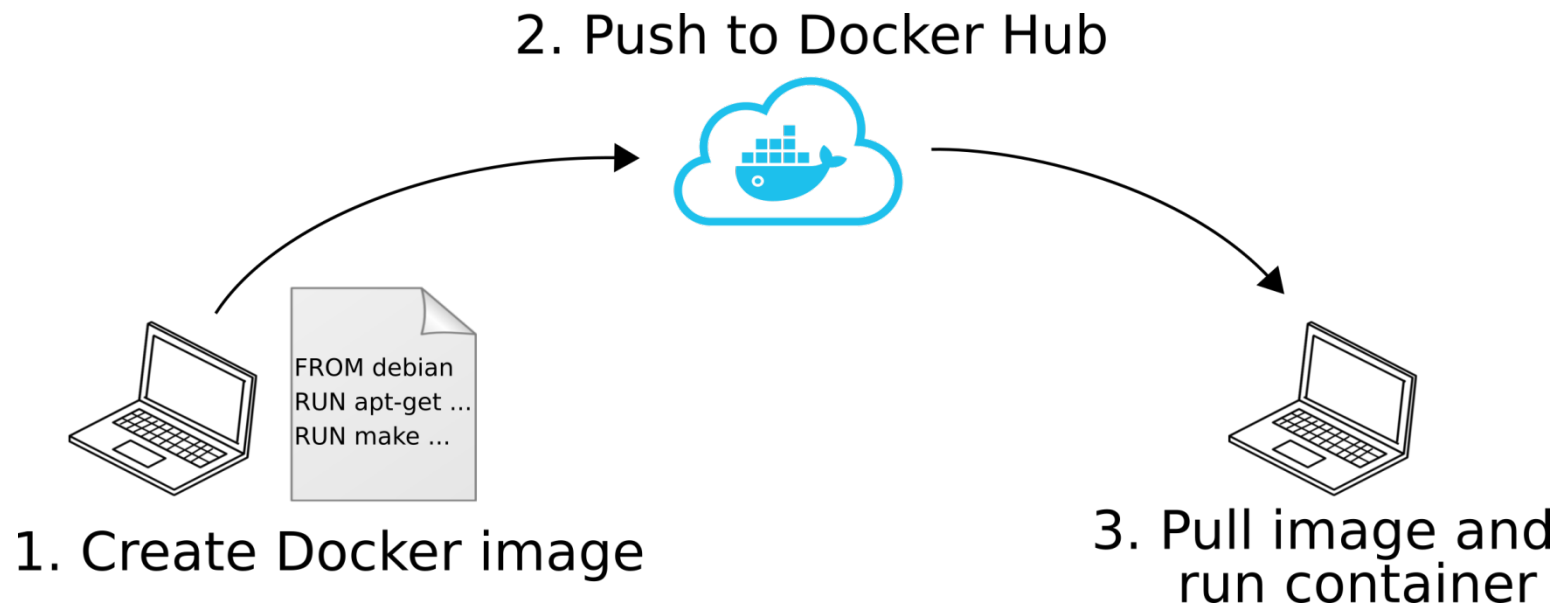
# Docker

- Extremely popular container implementation
- Easy to use authoring tools
  - Container images are created from recipe-like files
  - Images can be named, tagged and built on top of other images
- Cloud-based image distribution strategy
  - Several remote registries available (e.g. Docker Hub)
  - Client includes facilities to authenticate, push and pull images



# Docker workflow

1. An image is created locally from a Dockerfile
2. Push (i.e. upload) the image to a remote registry  
DockerHub is the public registry maintained from the Docker company
3. Pull (i.e. download) the image on a target machine and run the container



# Key terms

- **Image:** standalone, executable package that includes everything needed to run a piece of software (code, runtime libraries, environment variables, configuration files).
- **Container:** runtime *instance* of an image what the image becomes in memory when actually executed. It runs completely isolated from the host environment by default, only accessing host resources if configured to do so.



# So... how are containers useful?

- Containers give the possibility to create (scientific) applications that are:

1. Portable

2. Reproducible

3. Easy to deploy

4. Easy to test

# Live demo!

---

# Cheatsheet

Step-by-step guides: <https://github.com/eth-cscs/containers-hands-on>

```
docker pull <repo/image:tag>
```

---

```
docker run <image:tag> <command>
```

---

```
docker run -it <image:tag> bash
```

---

```
docker run <image:tag> mpiexec -n 2
```

---

```
docker images
```

---

```
docker build -t <repo/image:tag> .
```

---

```
docker login
```

---

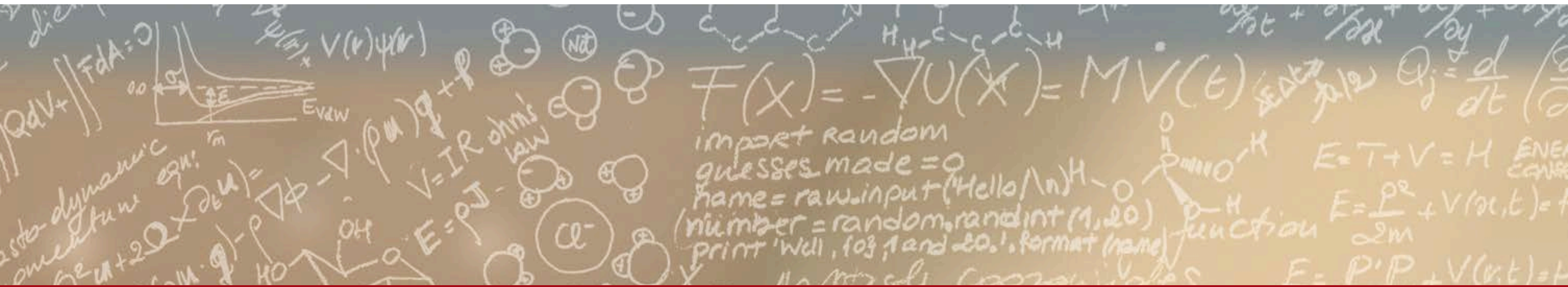
```
docker push <repo/image:tag>
```



**CSCS**

Centro Svizzero di Calcolo Scientifico  
Swiss National Supercomputing Centre

**ETH** zürich



**Thank you for your attention.**