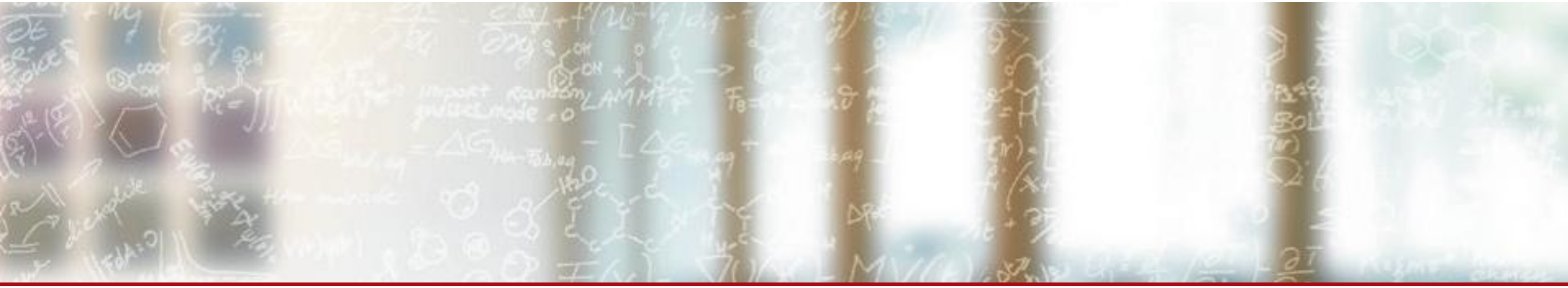
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

ETH zürich



Introduction to containers and Docker

Summer School on Effective HPC for Climate and Weather

Alberto Madonna, CSCS

August 26, 2021

Table of Contents

1. Introduction to containers and Docker (20 min)
2. Docker tutorial / live demo (50 min)
3. Overview of containers in HPC (15 min)

- Break -

4. Lab session (1h30m)

- Slides and code available at https://github.com/eth-cscs/containers-hands-on
- Lab tutorial video: https://youtu.be/dv74sFb3cVc

- *Disclaimer: This material reflects only the author's view and the EU-Commission is not responsible for any use that may be made of the information it contains*

Containers

- Isolated environments to run applications/services
- Images include all software dependencies
- Prescriptive, portable, easy to build, quick to deploy

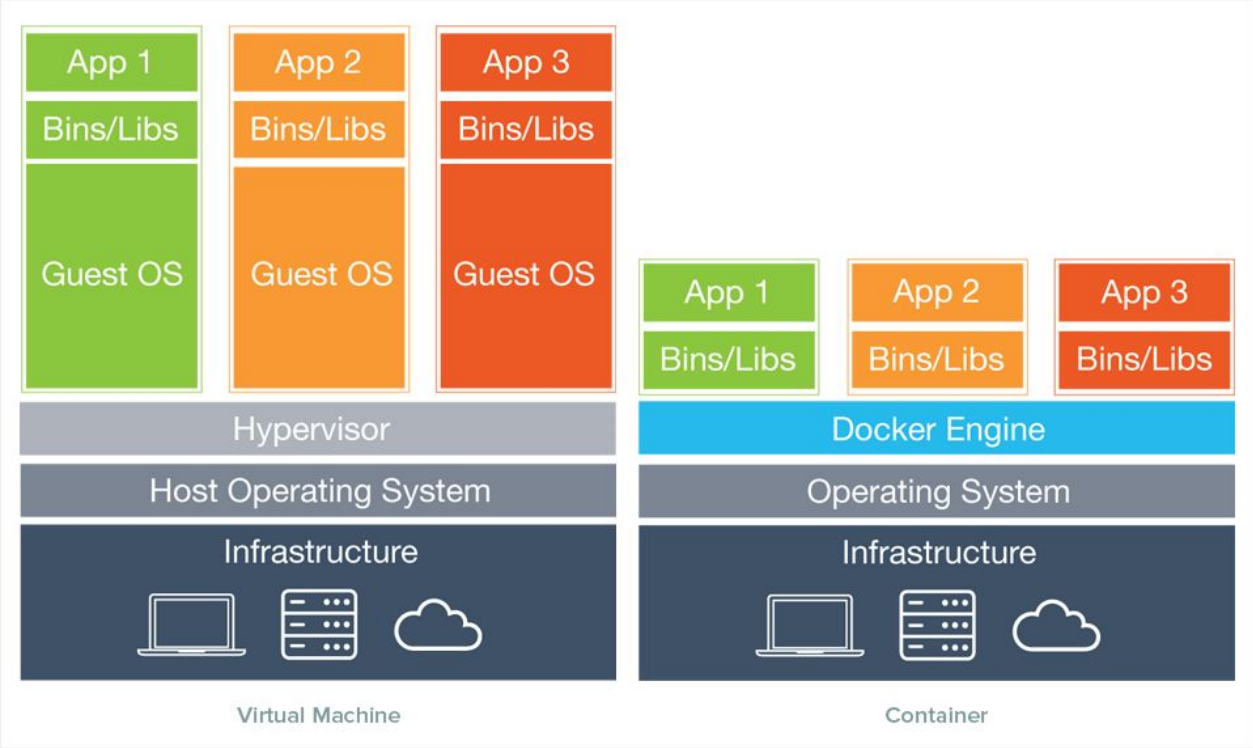


Image credit: Docker Inc.

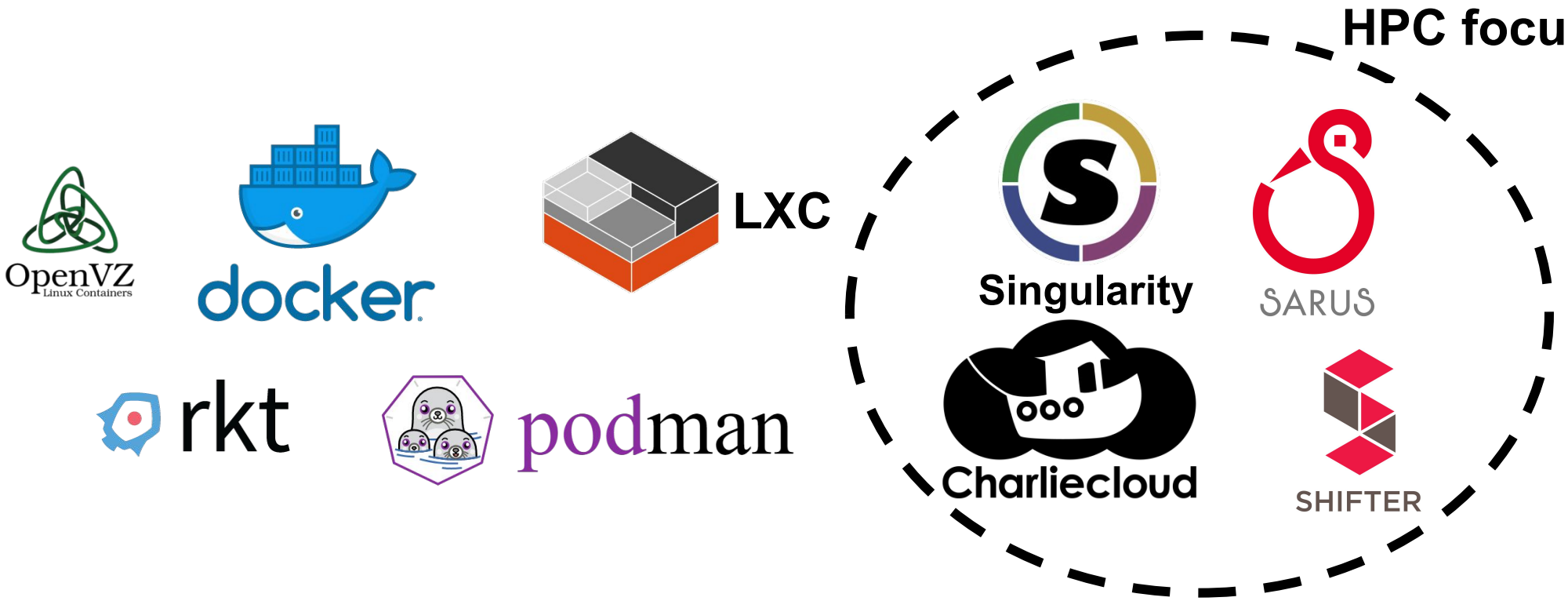
Key terms

- **Image**: standalone, executable package that includes everything needed to run a piece of software (code, runtime libraries, configuration files). Provides the filesystem and metadata (e.g. environment variables, initial working directory) for a container.

- **Container**: a process isolated from the rest of the system through abstractions created by the OS. The level of isolation can be controlled, allowing access to host resources. Its filesystem content comes from an image.
  - Can be thought as the runtime *instance* of an image: what the image becomes in memory when actually executed.

Linux containers ecosystem

- Linux containers rely on abstraction features (*namespaces*[1]) provided by the kernel
- Different design decisions and use cases gave rise to several solutions:



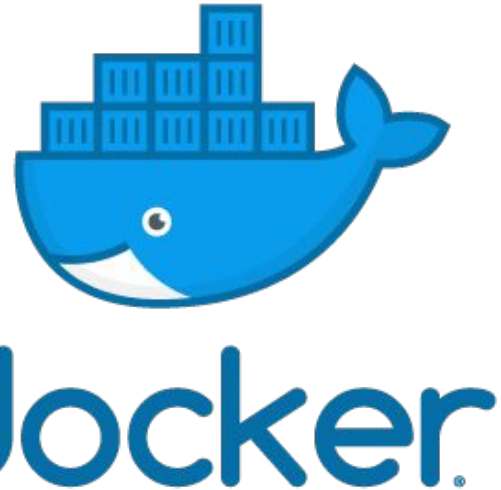
[1] "Namespaces in operation, part 1: namespaces overview" at https://lwn.net/Articles/531114/

Docker

- Extremely popular container implementation

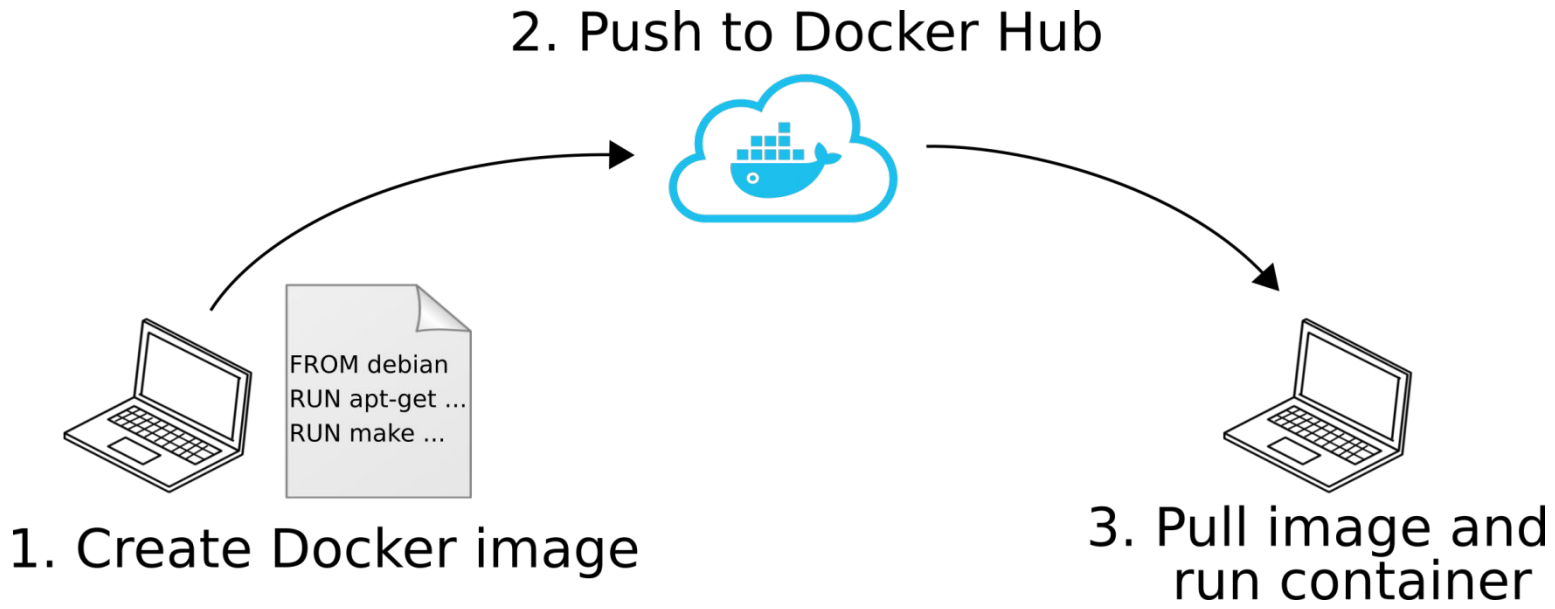
- Easy to use authoring tools
  - Container images are created from recipe-like files
  - Images can be named, tagged and built on top of other images

- 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 by the Docker company
3. Pull (i.e. download) the image on a target machine and run the container



So… how are containers useful?

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

1. Portable
2. Prescriptive
3. Easy to deploy
4. Easy to test

Live demo!

Using NVIDIA GPUs in Docker

GPU-accelerated application

- Included/built in the image, along with its runtime dependencies
- NVIDIA provides base images for CUDA, featuring compilers, runtime and accelerated libraries: https://hub.docker.com/r/nvidia/cuda
- Quickest way to get a Dockerfile going: `FROM nvidia/cuda`

GPU driver

- It is tied to the hardware: cannot be part of a portable image!
- Has to be imported upon container creation
- NVIDIA Container Toolkit to the rescue! https://github.com/NVIDIA/nvidia-docker
- Docker >= 19.03 has native support:
```
docker run --gpus all nvidia/cuda nvidia-smi
```

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

ETH zürich

Containers in HPC

Docker and HPC: not a good fit

- Security model assumes root privileges
- No integration with workload managers
- Missing support for diskless nodes
- Very limited support for kernel bypassing devices (e.g. accelerators and NICs)
- No adequate parallel storage driver

HPC-focused container software

- Enabling execution of containers complying with the peculiarities of HPC environments

Singularity

https://sylabs.io/singularity/

SARUS

https://products.cscs.ch/sarus/

Charliecloud

https://hpc.github.io/charliecloud/

SHIFTER

https://github.com/NERSC/shifter

Sarus container engine

- Combines container portability with native HPC performance
- Integrates with HPC infrastructure and software
- Customizes containers at runtime with standard plugins
- Pulls regular Docker images
- Provides a Docker-like command line interface

Further material & contact

- Slides and Lab material: https://github.com/eth-cscs/containers-hands-on
- Lab intro video: https://youtu.be/dv74sFb3cVc
- Official Docker documentation: https://docs.docker.com/
- Best practices for writing Dockerfiles: https://docs.docker.com/develop/develop-images/dockerfile_best-practices/
- Contact: alberto.madonna@cscs.ch

The ESiWACE1/2 projects have received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No **675191** and No **823988**

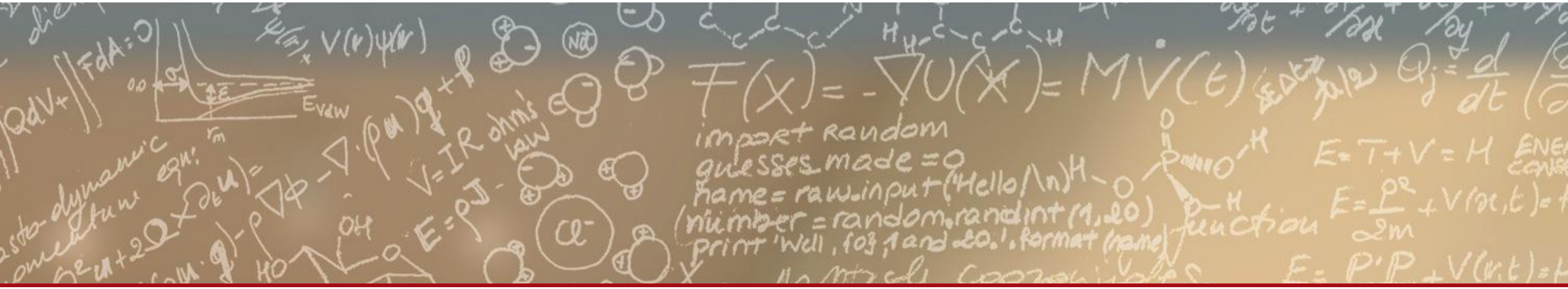
esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Disclaimer: This material reflects only the author's view and the EU-Commission is not responsible for any use that may be made of the information it contains

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

ETH zürich



Thank you for your attention.

Break time! Back at 15.30 CEST

Lab session

- Slides and material: https://github.com/eth-cscs/containers-hands-on
- Intro video: https://youtu.be/dv74sFb3cVc
- Quickstart:
  1. `git clone https://github.com/eth-cscs/containers-hands-on.git`
  2. `cd containers-hands-on/esiwace2-summer-school-2021`
  3. Open *docker-step-by-step.pdf* document

Backup slides

Docker cheatsheet

```
docker pull <user/image:tag>
docker run <image:tag> <command>
docker run –it <image:tag> bash
docker run <image:tag> mpirun –n 2
docker images
docker build –t <user/image:tag> .
docker login
docker push <user/image:tag>
```