Getting a Grip on Docker

IoT Workshop

Jorg Liebeherr

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

Objectives

- Getting a handle on terminology:

  Docker, Dockerhub, container, image, registry, containerd

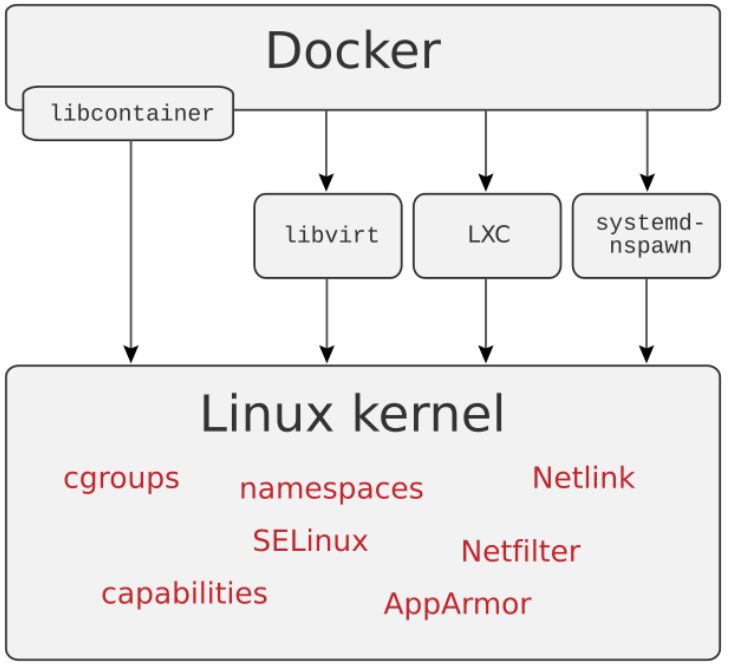
- Basic concepts:
  - Running an existing Docker image
  - Creating a custom Docker image
  - Push a custom Docker image to a registry
  - Distributing a custom Docker image

- Workshop

Docker

- **Docker** is a platform for containerized applications
  - First released in 2013 by Docker, Inc.
  - Software that hosts the containers is called **Docker Engine**
  - Docker is based on Linux containers and namespaces

- Reasons for Docker's popularity:
  - Runs on Linux, Windows, MacOS
  - Convenient tools (e.g., Docker Desktop)
  - Easy to build custom containers
  - Free repository (Dockerhub)



Docker in context

- Docker is a very (the most?) popular platform for running containers
- Fact: There are alternatives to Docker

  Podman, LXC/LXD (Linux Containers), CRI-O, containerd

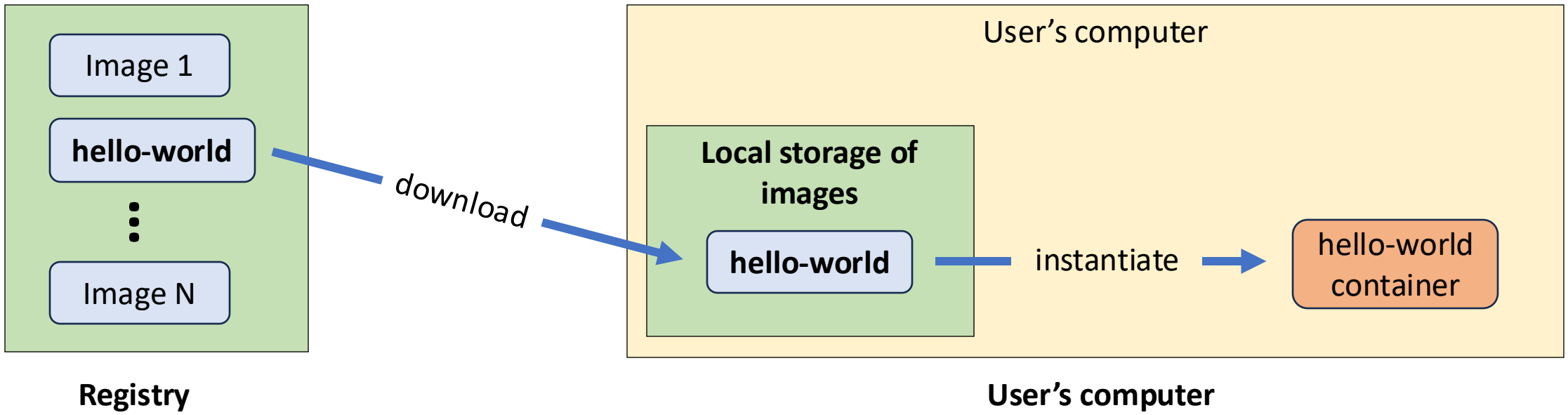
- Fact: Docker did not invent containerization

  1979: chroot in Unix V7
  2000: FreeBSD Jails
  2001: Linux Vserver
  2004: Solaris containers
  2008: LXC (Linux containers)
  2009: Apache Mesos (originally from UC Berkeley) → container orchestration
  2013: Docker
  2014: Kubernetes (originally from Google) → container orchestration

Deploying a Docker container

- Image = Template for creating a container
- Container = Runnable instance of an image (lightweight runtime system)
- Registry = Storage system for docker images

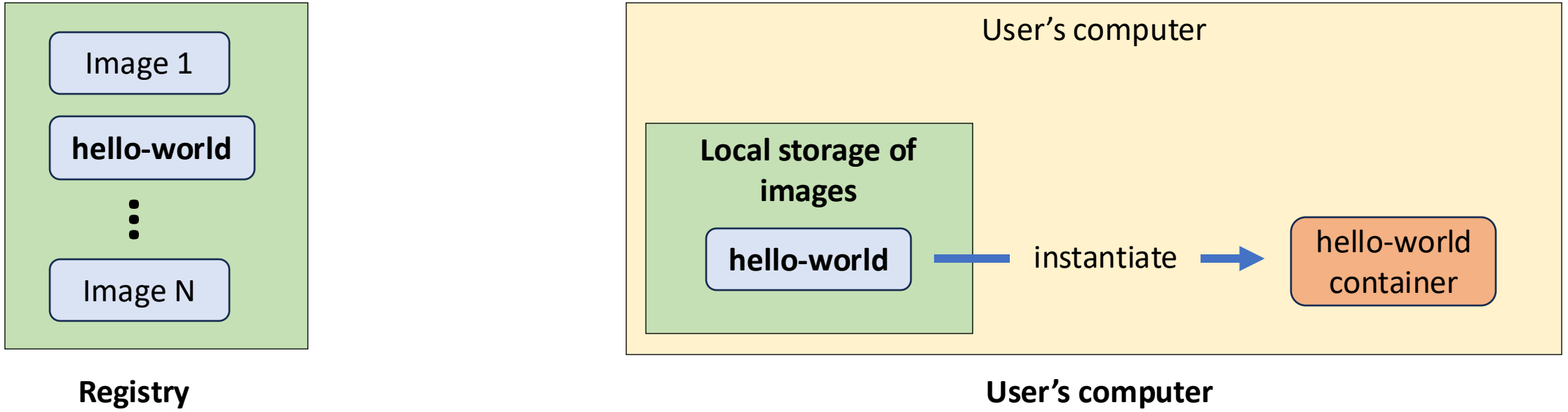
```
$ docker run hello-world
```



Deploying a Docker container

- Image = Template for creating a container
- Container = Runnable instance of an image (lightweight runtime system)
- Registry = Storage system for docker images

```
$ docker run hello-world
```



How to create a custom Docker image?

1. **Create a Dockerfile:** The Dockerfile contains commands to build a Docker image. It contains:
   a) Base image which is the starting point for a Docker image, e.g., Linux or an operating system with pre-installed software (Python).
   b) RUN commands To install additional software
   c) COPY commands to copy your own application code to the image
   d) WORKDIR command specifies the directory where your application will run
   e) ENTRYPOINT and CMD commands specify the commands that will be executed when the container starts, e.g., the command to run your application.
2. **Build the image:** The `docker build` command executes the commands in the Dockerfile and creates a new image.
3. **Test the image:** Run a container from the new image
4. **Push the image to a registry:** Push image it to a registry like Docker Hub, so it can be downloaded by others

Continue with workshop handout