

Introduction to Containerization (Docker, Docker Compose, and Singularity)

Ritu Arora

Nov 14, 2022

With Contributions from Charlie Dey, Carlos Redondo, and other colleagues from the
Texas Advanced Computing Center

WHAT IS A CONTAINER?

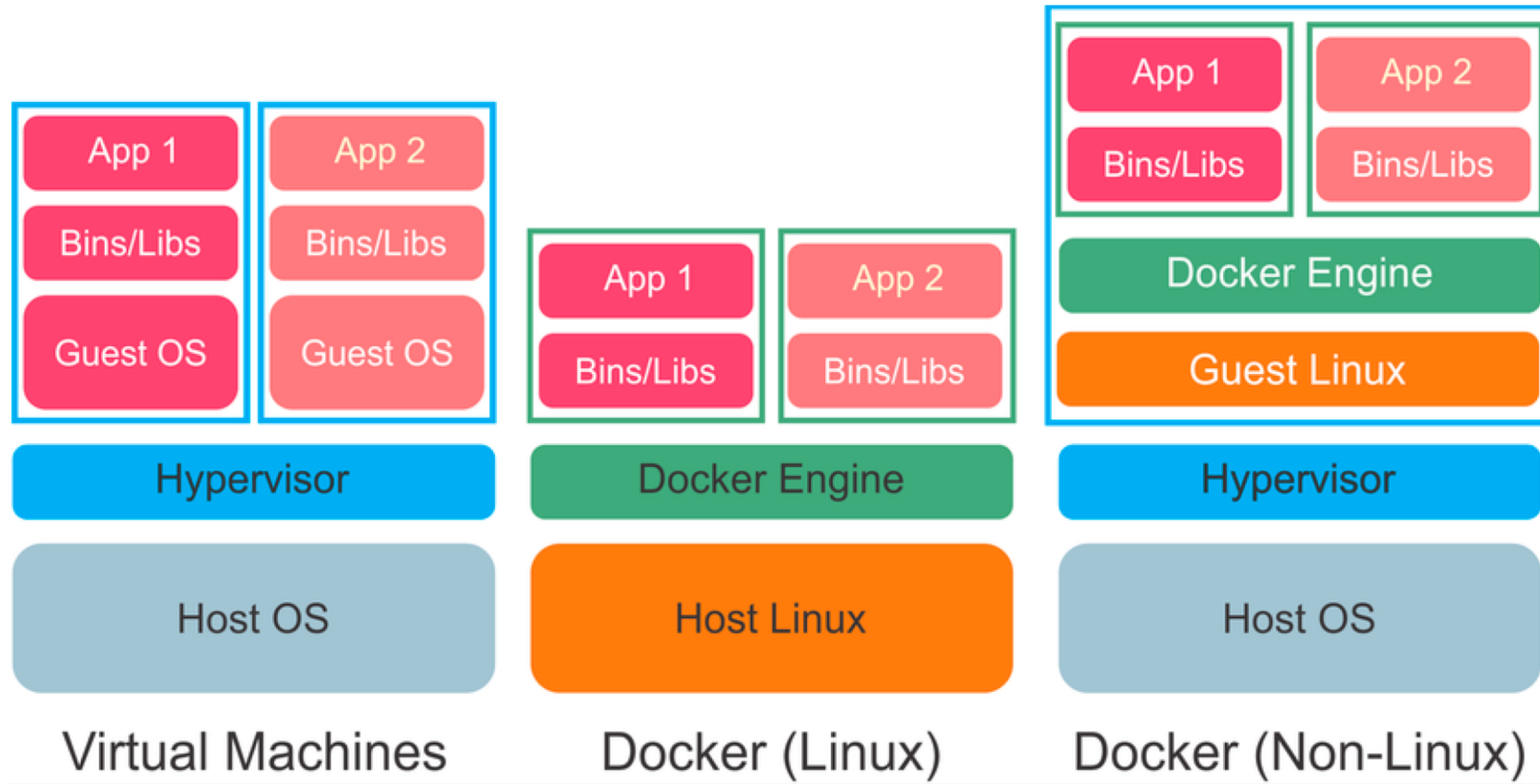
Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.

Examples: Docker, Singularity

ADVANTAGES OF CONTAINERIZATION

- Help in **deploying future-proof applications** by creating packages that are almost self-contained
- Make it **convenient to distribute production and trial versions of the code** that can run on the customers' devices without requiring an application-specific installation and configuration, hence, helps in developing **a scalable software distribution model**
- **Save time in complicated installs** - build a Docker application - push it to Docker Hub - and **reuse** it on any number of systems as you desire
- **Mitigate the portability issues related to the applications** - Dockerized applications can be ported conveniently across different cloud computing service providers - of course, you may need to install Docker and additional tools to run the Docker container depending upon the system that you are on
- **Help in doing reproducible science**

CONTAINER VERSUS VIRTUAL MACHINES



Source: Guldock: Using Docker Containers with a Common Graphics User Interface to Address the Reproducibility of Research - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/A-comparison-of-the-architecture-of-virtual-machines-and-Docker-software_fig4_299771559 [accessed 24 Sep, 2018]

WHAT IS DOCKER?

- Docker is an open-source project designed for creating, deploying, and running applications by using containers. These containers are portable, self-sufficient and can be run on any infrastructure in the cloud or on-premises on which Docker is installed/supported.
- Three important concepts related to Docker that we will cover today:
 - Docker container
 - As mentioned on slide # 1: containers are used to package an application along with all its dependencies (such as runtime system, libraries, and data) into a single unit
 - Docker image
 - Dockerfile

HOW POPULAR IS DOCKER?

Accelerate how you build, share, and run modern applications.

13 million +

developers

7 million +

applications

13 billion +

monthly image downloads

Source: docker.com [accessed November 11, 2022]

WHAT IS THE PROCESS FOR BUILDING DOCKER IMAGES?

1. Install Docker
2. Write a Dockerfile
 - + **Dockerfile** is a text file that contains instructions, using which, Docker can build images automatically
 - + Name of the file: Dockerfile
3. Build the Docker image and tag it
 - + A file containing the snapshot of a container is known as a **Docker image**
 - + It is created using the build command, and produces a container when it starts running
 - + You can add tags to the images during the build step or while saving it to a repository - the default tag is "latest"
4. Run the image that you built
5. Register on DockerHub - use the credentials to push the image
 - + Images are stored in a Docker registry such as registry.hub.docker.com
 - + Pull Images later on any system that you want to run the application on

HOW CAN CONTAINERS BE CREATED FROM DOCKER IMAGES?

- You can use Docker commands such as “docker pull” or “docker run”
 - + “Pull” will always fetch the latest version of the image from Docker Hub before running
 - + “Run” will search for an image locally and run it, and if there is nothing available locally, it will go to Docker Hub.
- We will learn more about this topic during the hands-on session

SOME PROJECTS USING DOCKER (1)

[Home](#) [Catalog](#) [CompChecker](#) [Select-A-License](#) [Message Board](#) [Blogs](#) [About Us](#) [Contact Us](#)

[Sign Up](#) [Login](#)



Opuntia

Opuntia (pronounced as "up-un-chia") is a software infrastructure for facilitating the assessment, discovery, dissemination, and reuse of publicly accessible software and data products. It consists of a catalog of software and data products, and tools named as iTracker, CompChecker, and Select-A-License.

The Opuntia project is under active research and development, and new features are added to the public release iteratively.

Catalog

The catalog of products facilitates their discovery, dissemination, and reuse. A product can be added to the Opuntia catalog after logging in to this website and filling the form on the "Add Product" page. The login can be done either by using the credentials associated with an account created on the Opuntia website, or by clicking the link for "Login using CILogon" and choosing one of the existing accounts such as Google, GitHub, ORCID, or the home organization (if the organization is a part of the InCommon Federation). CILogon is integrated with the Opuntia project

iTracker

iTracker helps in tracking the user-defined metrics for evaluating software and data products deployed and used on different platforms and computing environments. While some types of metrics are dynamically tracked and updated in iTracker's database, others are currently updated statically. For the statically updated metrics, at the time of editing their products' information in the catalog, the product owners can specify the name of the metric of interest, and the measured value. They can also provide the URL to a publicly accessible file

CompChecker

CompChecker is a tool that can assist with checking the legal and technical interoperability of the products. Currently, it can assist with comparing any two cataloged products or licenses directly and provide suggestions on whether the products or licenses being compared can be mixed with each other or not. Specific clauses of licenses, such as those around sublicensing, commercialization, warranties, modifications, and reuse are compared with each other to provide guidance. In the next phase of research and development, CompChecker will

Select-A-License

Select-A-License is a decision-support tool that can help in selecting appropriate licenses for software and data products. It is designed to help in selecting a license from amongst the 65 open-source licenses in the Opuntia database. The output of this tool is driven by the responses it receives to a set of questions it presents. For convenience, this tool also provides the text that can be used to prepare the license agreement for each license that it suggests.

SOME PROJECTS USING DOCKER(2)

INTERACTIVE PARALLELIZATION TOOL

[Terminal](#) [Compile](#) [Run](#) [Job History](#) [Help](#) [Admin](#) [Messageboard](#)

Terminal

```
23bc6a740cf1 login: term
Password:
Last login: Tue Sep 25 23:11:57 UTC 2018 on pts/4
Linux 23bc6a740cf1 4.4.0-133-generic #159-Ubuntu SMP Fri Aug 10 07:31:43 UTC 2018 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

# term @ 23bc6a740cf1 in ~ [23:12:40]
$ █
```

File Upload

File upload Folder upload

No file chosen

Download File/Folder

--Select-- ▾

Getting Docker

A **Docker Image** is a container template from which one or more containers can be run. It is a rooted filesystem that, by definition, contains all of the file dependencies needed for whatever application(s) will be run within the containers launched from it. The image also contains metadata describing options available to the operator running containers from the image.

One of the great things about Docker is that a lot of software has already been packaged into Docker images. One source of 100s of thousands of public images is the official **Docker Hub**:

<https://hub.docker.com>

The Docker Hub contains images contributed by individual users and organizations as well as "official images".

Thanks!

Any questions, comments, or concerns?

<https://github.com/ritua2/Basil/tree/main/training>