National Aeronautics and Space Administration

Scientific Software Common Interest Group (SciCIG)

Docker:
Using Software Containers in Development

November 1, 2016

Presented by Ed Sabol
GSFC Code 660, HEASARC Web/DB Development

Agenda

- What is Docker?
- How is Docker different from virtual machines?
- Why would you want to use Docker?
- How is the HEASARC using Docker?
- Docker CLI and API
- Resources
- Conclusion/Q & A

What is Docker?

- Docker is open source container technology for deploying and running applications in isolated, sandboxed environments.

- A container is a package that wraps application code with everything it needs to work (OS libraries/tools, any/all dependencies, and related files). The resulting code always acts the same, regardless of where you run it.

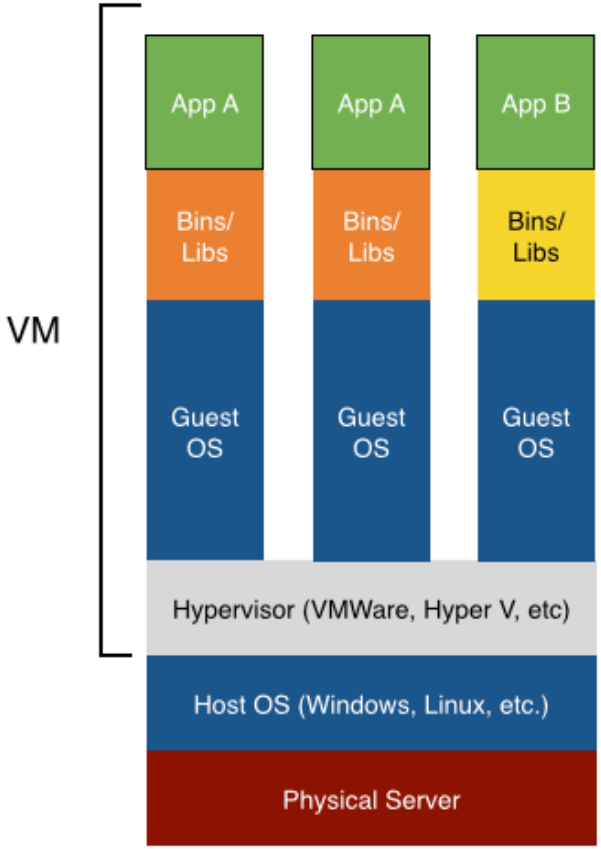
- As a developer, you can focus on writing the code rather than worrying about potentially inconsistent runtime environments.

- Docker runs on Linux, macOS, and Windows, but works best on Linux. Supported by Amazon Web Services/EC2, Google App Engine, and many other cloud platforms.

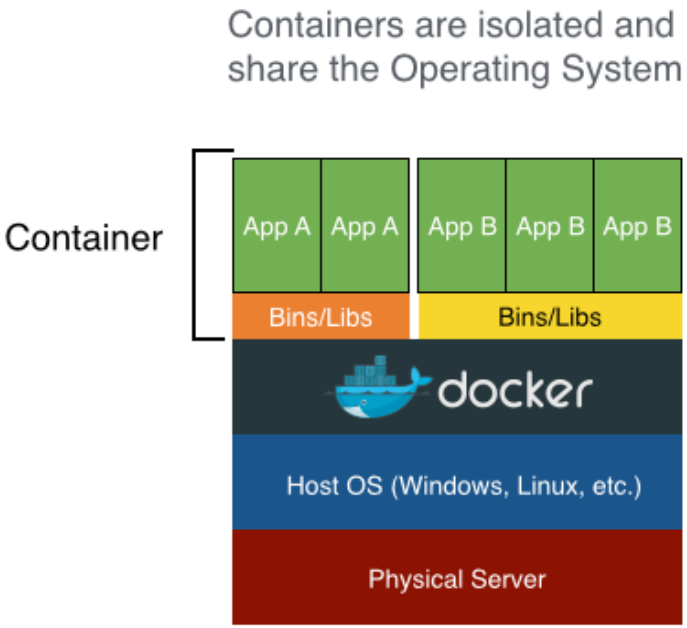
How is Docker different from a VM?

- Docker containers use the same kernel as the host OS, but it uses its own userland libraries and binaries.
- Example: Ubuntu 14.04 container running on a Scientific Linux (RHEL/CentOS) 7.2 host. Both the container and the host use the same Linux kernel (3.10.x).
- No Hypervisor, no architecture or hardware emulation:
  Much more efficient!
- Still provides the process isolation (sandbox) and OS abstraction.

How is Docker different from a VM?



Virtual Machine



Container

Why would you want to use Docker?

- **Security**: Chroot on steroids and easier to implement. Processes are isolated, sandboxed, with fine-grain control over network and access to specific files or file systems. With caveats.

- **Lightweight**: Startup time and network latency (if you use port mapping) only marginally slower than running native on the host. Execution speed, I/O rates are identical.

- **Better Workflow**: Develop/test locally, easy to deploy to production environments.

- **More Convenient for End Users**: No more download, gunzip, untar, ./configure, make && make install (and pray you have the proper dependencies). Just download the container image and run.

- **Scalability**: Docker Swarm provides clustering capabilities tested on up to a thousand nodes with high availability and failover.

How is the HEASARC using Docker?

- HEAsoft is a mammoth hodgepodge of multi-mission data analysis software developed over 2+ decades in Fortran, C, C++, Tcl, Perl. Latest version has nearly 800 installed binaries! Lengthy install. Difficult to learn how to use.

- WebHera was designed to be an easy-to-use, tool-generic interface to using HEAsoft on the web to decrease the barriers of entry to data analysis. Need to support users securely uploading their own files for data analysis, downloading any generated files, and transparently accessing the HEASARC data archives.

- Initial versions had security problems. Sandboxing with Docker is the solution (we hope).

- Packaging HEAsoft as a Docker container image for end users?

Docker Basics

- Docker containers are instances of Docker images.
  OOP analogy:
  container : image :: object : class

- Docker Hub is a repository of official, user-contributed, and private Docker images. Over 100,000 images. Most every open source package has official images there.

- A “Dockerfile” is a script that you use to build a custom Docker image. Like Makefile but for Docker. Syntax is easy, but getting it exactly right takes time. Lots of examples online.

Docker Command-Line Interface

- docker pull ubuntu
- docker run ubuntu /bin/echo "this runs inside the container"
- docker run -it ubuntu bash
- docker ps -a
- docker stop *container_ID*
- docker rm *container_ID*
- docker images
- docker build -t *image_name:tag* .
- docker rmi *image_name*

Docker API

- API is mostly RESTful with a JSON representation (basically HTTP). Dizzying number of options.
- API client libraries/frameworks available for most languages
- Perl example:

```
my $api = Net::Docker->new;
my $id = $api->create(Cmd => [ 'somecommand', 'args' ],
                      Image => 'hera',
                      User => 'herauser',
                      WorkingDir => '/home/herauser/data',
                      NetworkDisabled => 1,
                      HostConfig => { Binds => [ '/FTP:/FTP:ro' ] });
$api->start($id);
# handling I/O left as an exercise
$api->stop($id);
$api->remove_container($id);
```

Resources

- Installing Docker:
  https://docs.docker.com/engine/installation/

- Docker Hub:
  https://hub.docker.com/

- Docker Remote API Reference:
  https://docs.docker.com/engine/reference/api/docker_remote_api_v1.24/

- Docker Remote API Client Libraries:
  https://docs.docker.com/engine/reference/api/remote_api_client_libraries/

Conclusion

- Broad support in the industry. Google, Amazon, Microsoft, etc. All the major players.
- Stepping stone to the cloud. Start using it now for local environments for better development/operations workflows, and you will have an easier time deploying to major cloud platforms later.
- Wave of the future? No, it’s here now.
- Any questions?