

SKA Training on Containers

Introduction to Singularity

Containers

Manuel Parra-Royón – IAA CSIC (Spain)

Outline

- > Why Singularity (and ... Docker?)
- > Preparing the working environment
- > Basic use of Singularity containers
- > Your own Container Hub and Singularity Hub
- > Creating our first container
- > Share your work

Singularity

- > Singularity is a (another) container platform 😊
as well as: Docker, Podman, LXD, Moby, ... among other.
- > Singularity allows you to create and to manage containers.
Wait! → What is a container? → Mateusz

Packaging of software parts so that they are portable, reproducible and interoperable.

Build the container on your laptop, and then deploy in HPC, cluster, cloud or wherever you want.

A container is a single file or an image file, just copy it, use it or delete it.

Singularity

> **Advantages:**

- Easy to learn, without many options.
- Works well in HPC given its design.
- You can use Docker containers and run them directly:
 - If it is already in Docker, we can use it !
- SingularityHub: Singularity's DockerHub
 - However, we can have our own Containers Hub without depending on external services.

Singularity

> Disadvantages:

- Under very active development -> Many changes and versions.
- Less mature than docker
- Not such easy to install than Docker.

Singularity

> Aims

- Mobility of Compute:
 - transfer and store containers with standard data mobility tools (rsync, scp, gridftp, http, NFS, etc..)
- Reproducibility:
 - verifiable signature to ensure that the format is not changed

Singularity

> Aims

- User Freedom
 - install the applications, versions, and dependencies for their workflows without impacting the system in any way
- Support on Existing Traditional HPC:
 - can run on Linux distributions from RHEL6 and similar vintages, and the contained images have been tested as far back as Linux 2.2. SLURM, Torque, SGE, ...

High Level view of Singularity

> **Security and user privileges**

- User inside a container is the same user outside the container:
 - So this preserves your user environment (user, group, ...) inside the container.
- Singularity prevents user context escalation:
 - If you needs to be root to make changes -> You need to have an endpoint with root access (your laptop?)

High Level view of Singularity

> **Container image**

- A container image is simply a file, which physically includes the container.
- Since it is a file, you can use all the commands of your system and use linux ACL security for access, groups, permissions, etc.
 - Copy, move, branches,
- Example: Singularity Hub and **Container Hub** -> Shared directory with many indexed containers.

High Level view of Singularity

> Container image

- Support for container format:
 - squashfs, ext3, a directory, tar.gz, tar, ...
- Supported URI's:
 - shub:// for containers in SingularityHub
 - docker:// for containers stored in DockerHub
 - instance:// for containers as service

Workflow with singularity

Interactive Development

```
sudo singularity build --sandbox tmpdir/ Singularity
```

```
sudo singularity build --writable container.img Singularity
```

Build from Recipe

```
sudo singularity build container.img Singularity
```

Build from Singularity

```
sudo singularity build container.img shub://vsoch/hello-world
```

Build from Docker

```
sudo singularity build container.img docker://ubuntu
```

BUILD ENVIRONMENT

Container Execution

```
singularity run container.img  
singularity shell container.img  
singularity exec container.img ...
```

Reproducible Sharing

```
singularity pull shub://...  
singularity pull docker://... *
```

PRODUCTION ENVIRONMENT

* Docker construction from layers not guaranteed to replicate between pulls

Singularity commands

- **build** : Build a container on your user endpoint or build environment
- **exec** : Execute a command to your container
- **inspect** : See labels, run and test scripts, and environment variables
- **pull** : pull an image from Docker or Singularity Hub
- **run** : Run your image as an executable
- **shell** : Shell into your image

Let's go with singularity !

Go to SKA Training on Containers repository -> :

<https://gitlab.com/ska-telescope/src/ska-src-training-containers/>

Select [Introduction to Singularity Containers](#)

Open your terminal  !