

Using Docker for Data Processing & Inspection

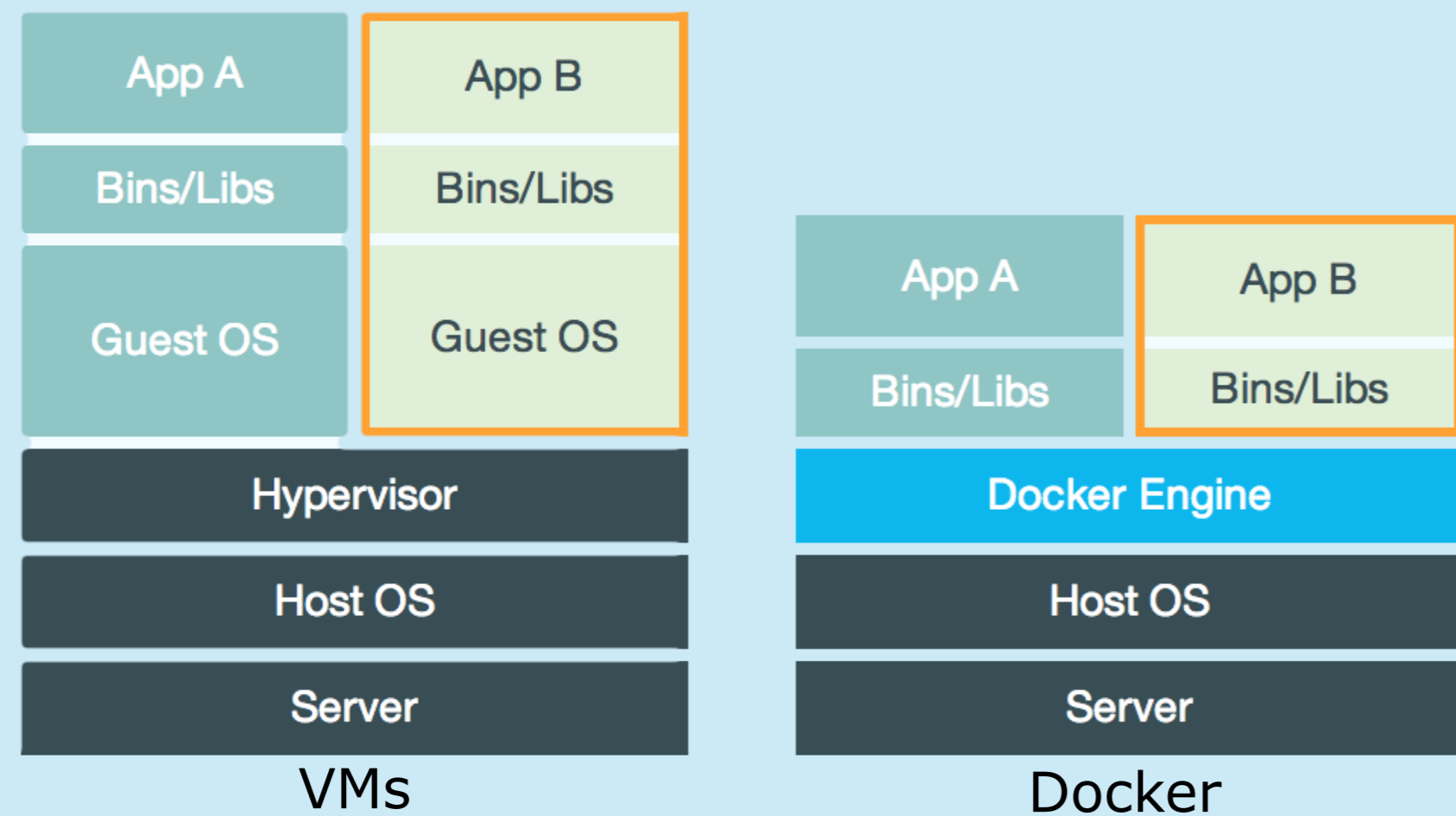
Author: Dr. ir. Jan David Mol (mol@astron.nl)



What is Docker (to us)?

Docker is an (LXC) *virtualization technology*, providing an easy way to create, distribute, and run application bundles:

- A docker *image* contains a full software stack (libraries, application) independent of host OS.
- Starting copies an image, leaving original intact.
- Similar to VM, but light-weight:



How to use:

- Base images like centos:6, ubuntu:16.04, scratch available (and >100,000 more!).
- Images are extended using recipes in text files (Dockerfile) → automatic builds!
- Images are easy to build, tag, distribute, run.

Why Docker in operations?

- DevOps: shift dependency management from sysadmin → developer & subversion/git.
- Multiple versions of software stacks can coexist.
- *Complete freedom to test new OS, libraries, code.*
- *Full production stack remains untouched.*

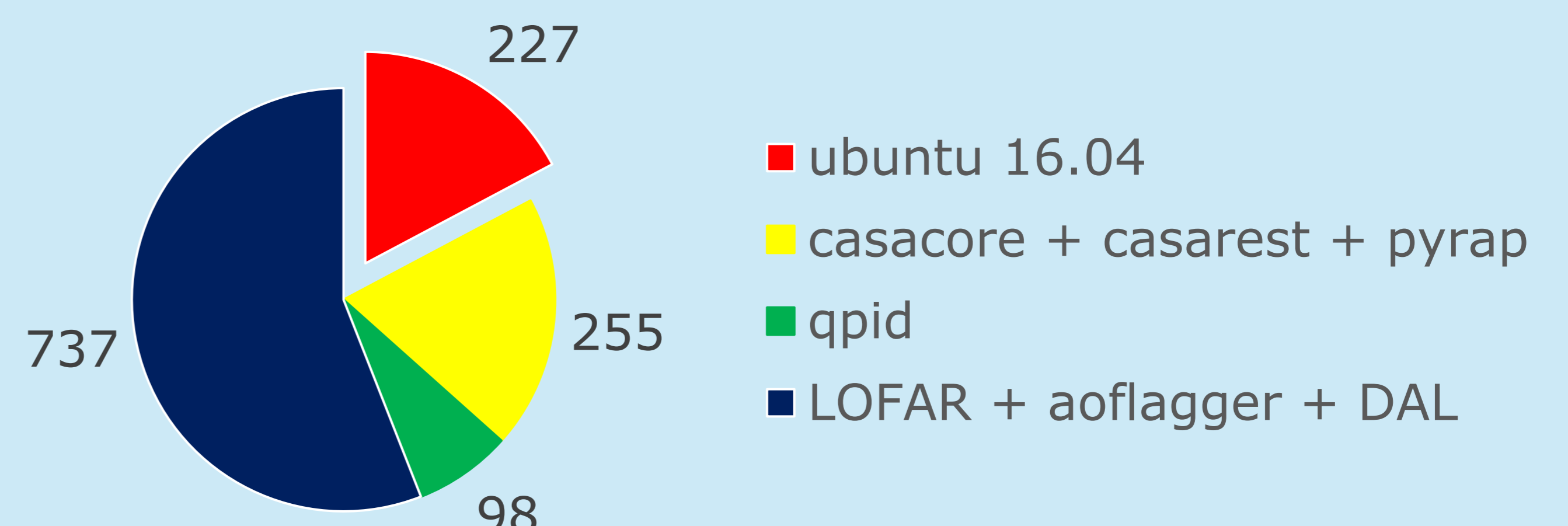
Docker for data processing

We have pipeline images installed for:

- All production software versions ≥2.17
- Any testing & commissioning
- Software version is pipeline specification parameter

REPOSITORY	TAG
lofar-pipeline	LOFAR-Release-2_17
lofar-pipeline	LOFAR-Release-2_18
lofar-pipeline	ubuntu14
lofar-pipeline	ubuntu16

lofar-pipeline image breakdown (MByte)



We publish our images, so *anyone* can run our code:

```

|-----image name & version-----| |--program & args
$ docker run lofar/lofar-pipeline:LOFAR-Release-2_17 DPPP
Usage: DPPP [-v] [parsetfile] [parsetkeys...]
...
    
```

Docker for data inspection

- (Astronomy) packages tend to favour a specific OS
- Used to take weeks to get them all on one system.
- Hard to reproduce.
- Some require incompatible libraries.
- Multiple versions of same package hard to support.
- Upgrading can be difficult.
- Docker isolates each package + dependencies.
- Example: DS9. Dockerfile:

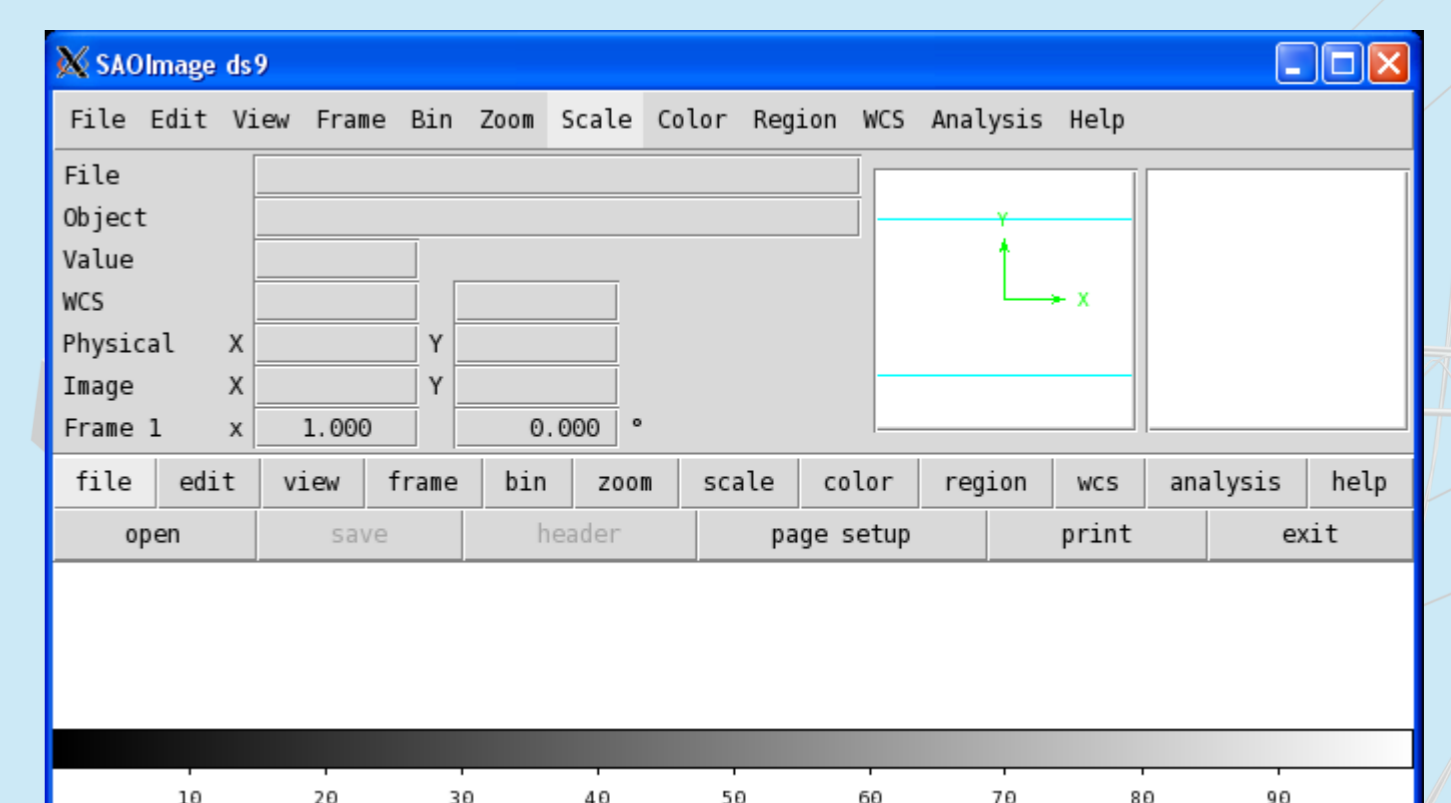
```

FROM centos:6 <- base OS, or other image to configure further
RUN yum install -y libXScrnSaver libX11 libXft dejavu-sans-mono-fonts
ADD http://ds9.si.edu/download/linux64/ds9.linux64.7.4.tar.gz /root/
RUN cd /usr/local/bin && tar xzf /root/ds9.linux64.7.4.tar.gz
ENTRYPOINT ["ds9"] <- program to start by default
    
```

- To build and run:

```

$ docker build -t ds9 /parent/directory/of/Dockerfile
|--user identity & homedir--| |--X11 forwarding--|
$ docker run -u $UID -e HOME -v $HOME:$HOME -e DISPLAY --net=host ds9
    
```



- We wrap 'docker run' for users:

```

$ cat docker-run.sh
#!/bin/bash
CONTAINER_NAME='id -nu'-'date +%F_%H-%M-%S' <- tag each container with username & timestamp
exec docker run -v /data:/data:ro -v /localdata/scratch:/localdata/scratch \ <- forward data dirs
-u 'id -u':'id -g' -e USER -v $HOME:$HOME -e HOME -w $HOME \ <- forward identity & homedir
-e DISPLAY --net=host \ <- forward X11
--rm -it --name $CONTAINER_NAME \ <- typical for interactive use
"$@" <- forward supplied "docker run" parameters
    
```

LOFAR's Docker deployment

