



imgw

Institut für Meteorologie
und Geophysik



**universität
wien**

Faculty of Earth Sciences,
Geography and Astronomy

Singularity



User Workshop on Singularity on HPC
24.11.2021, MB

Agenda



- Introduction
 - What is a container
 - Difference to docker
 - Security on HPC + MPI + Infiniband
 - Building a container
- Hands on
 - Build a container
 - Modify a definition files / customized image
- Examples & Problems
 - MPI
 - Performance
 - Cloud.sylabs.io
 - Build service, overlay, signing, best practices

Before containerization

- Goods must be loaded and unloaded **individually**, many times by hand
- **Inefficient** – more time spend loading and unloading goods than transporting them
- **Insecure** – goods must be stored and handled by intermediaries during transport; potential loss and theft of goods
- **Local** – only luxury and specialty goods shipped long-distance



After containerization

- **Standardized** – containers of known dimensions and permissible weight tolerance
- **Efficient** – portable containers allow fast loading and unloading from multiple modes of transportation
- **Secure** – goods remained stored within the same container
- **Global** – cost effective shipping



What is the problem?

```
mkandes — mkandes@comet-ln3:~/gpse — ssh comet — 80x43
[mkandes@comet-ln3 ~]$ git clone https://github.com/mkandes/gpse.git
Cloning into 'gpse'...
remote: Enumerating objects: 528, done.
remote: Total 528 (delta 0), reused 0 (delta 0), pack-reused 528
Receiving objects: 100% (528/528), 311.06 KiB | 1.31 MiB/s, done.
Resolving deltas: 100% (347/347), done.
Checking out files: 100% (15/15), done.
[mkandes@comet-ln3 ~]$ cd gpse/
[mkandes@comet-ln3 gpse]$ ls
CHANGELOG  gpse.input  LICENSE  Makefile  README  source
[mkandes@comet-ln3 gpse]$ cat /etc/os-release | grep PRETTY_NAME
PRETTY_NAME="CentOS Linux 7 (Core)"
[mkandes@comet-ln3 gpse]$ uname -a
Linux comet-ln3.sdsc.edu 3.10.0-957.12.2.el7.x86_64 #1 SMP Tue May 14 21:24:32 U
TC 2019 x86_64 x86_64 x86_64 GNU/Linux
[mkandes@comet-ln3 gpse]$ module list
Currently Loaded Modulefiles:
  1) intel/2018.1.163      2) mvapich2_ib/2.3.2
[mkandes@comet-ln3 gpse]$ module purge
[mkandes@comet-ln3 gpse]$ module load gnu/7.2.0
[mkandes@comet-ln3 gpse]$ module load mvapich2_ib/2.3.2
[mkandes@comet-ln3 gpse]$ module list
Currently Loaded Modulefiles:
  1) gnu/7.2.0             2) mvapich2_ib/2.3.2
[mkandes@comet-ln3 gpse]$ make
mpif90 -Jbuild -fimplicit-none -fmodule-private -ffree-form -ffree-line-length-n
one -std=gnu -fdefault-real-8 -O2 -mtune=native -c source/math.f90 -o build/mat
h.o
source/math.f90:45:23:

      USE, INTRINSIC :: ISO_FORTRAN_ENV
      1
Warning: Use of the NUMERIC_STORAGE_SIZE named constant from intrinsic module IS
O_FORTRAN_ENV at (1) is incompatible with option -fdefault-real-8
mpif90 -Jbuild -fimplicit-none -fmodule-private -ffree-form -ffree-line-length-n
one -std=gnu -fdefault-real-8 -O2 -mtune=native -c source/io.f90 -o build/io.o
source/io.f90:45:23:

      USE, INTRINSIC :: ISO_FORTRAN_ENV
      1
Warning: Use of the NUMERIC_STORAGE_SIZE named constant from intrinsic module IS
O_FORTRAN_ENV at (1) is incompatible with option -fdefault-real-8
mpif90 -Jbuild -fimplicit-none -fmodule-private -ffree-form -ffree-line-length-n
```

```
mkandes — mckandes@login2.stampede2:~/gpse — ssh stampede2 — 80x43
login2.stampede2(1002)$ git clone https://github.com/mkandes/gpse.git
Cloning into 'gpse'...
remote: Enumerating objects: 528, done.
remote: Total 528 (delta 0), reused 0 (delta 0), pack-reused 528
Receiving objects: 100% (528/528), 311.06 KiB | 2.06 MiB/s, done.
Resolving deltas: 100% (347/347), done.
login2.stampede2(1003)$ cd gpse/
login2.stampede2(1004)$ ls
CHANGELOG  gpse.input  LICENSE  Makefile  README  source
login2.stampede2(1005)$ cat /etc/os-release | grep PRETTY_NAME
PRETTY_NAME="CentOS Linux 7 (Core)"
login2.stampede2(1006)$ uname -a
Linux login2.stampede2.tacc.utexas.edu 3.10.0-957.5.1.el7.x86_64 #1 SMP Fri Feb
1 14:54:57 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
login2.stampede2(1007)$ module list

Currently Loaded Modules:
  1) intel/18.0.2          4) git/2.24.1           7) cmake/3.16.1
  2) libfabric/1.7.0      5) autotools/1.1       8) xalt/2.8
  3) impi/18.0.2         6) python2/2.7.15      9) TACC

login2.stampede2(1008)$ sed -i 's/ := gfortran/ := ifort/g' Makefile
login2.stampede2(1009)$ make
mpif90 -Jbuild -implicitnone -free -stand none -module build -real-size 64 -ipo
-O3 -no-prec-div -fp-model fast=2 -xHost -c source/math.f90 -o build/math.o
ifort: command line warning #10006: ignoring unknown option '-Jbuild'
mpif90 -Jbuild -implicitnone -free -stand none -module build -real-size 64 -ipo
-O3 -no-prec-div -fp-model fast=2 -xHost -c source/grid.f90 -o build/grid.o
ifort: command line warning #10006: ignoring unknown option '-Jbuild'
source/grid.f90(255): warning #6178: The return value of this FUNCTION has not b
een defined. [GRID_BINARY_SEARCH]
      INTEGER RECURSIVE FUNCTION grid_binary_search()
      -----^
mpif90 -Jbuild -implicitnone -free -stand none -module build -real-size 64 -ipo
-O3 -no-prec-div -fp-model fast=2 -xHost -c source/pmca.f90 -o build/pmca.o
ifort: command line warning #10006: ignoring unknown option '-Jbuild'
mpif90 -Jbuild -implicitnone -free -stand none -module build -real-size 64 -ipo
-O3 -no-prec-div -fp-model fast=2 -xHost -c source/vex.f90 -o build/vex.o
ifort: command line warning #10006: ignoring unknown option '-Jbuild'
mpif90 -Jbuild -implicitnone -free -stand none -module build -real-size 64 -ipo
-O3 -no-prec-div -fp-model fast=2 -xHost -c source/rot.f90 -o build/rot.o
```

What is a container?

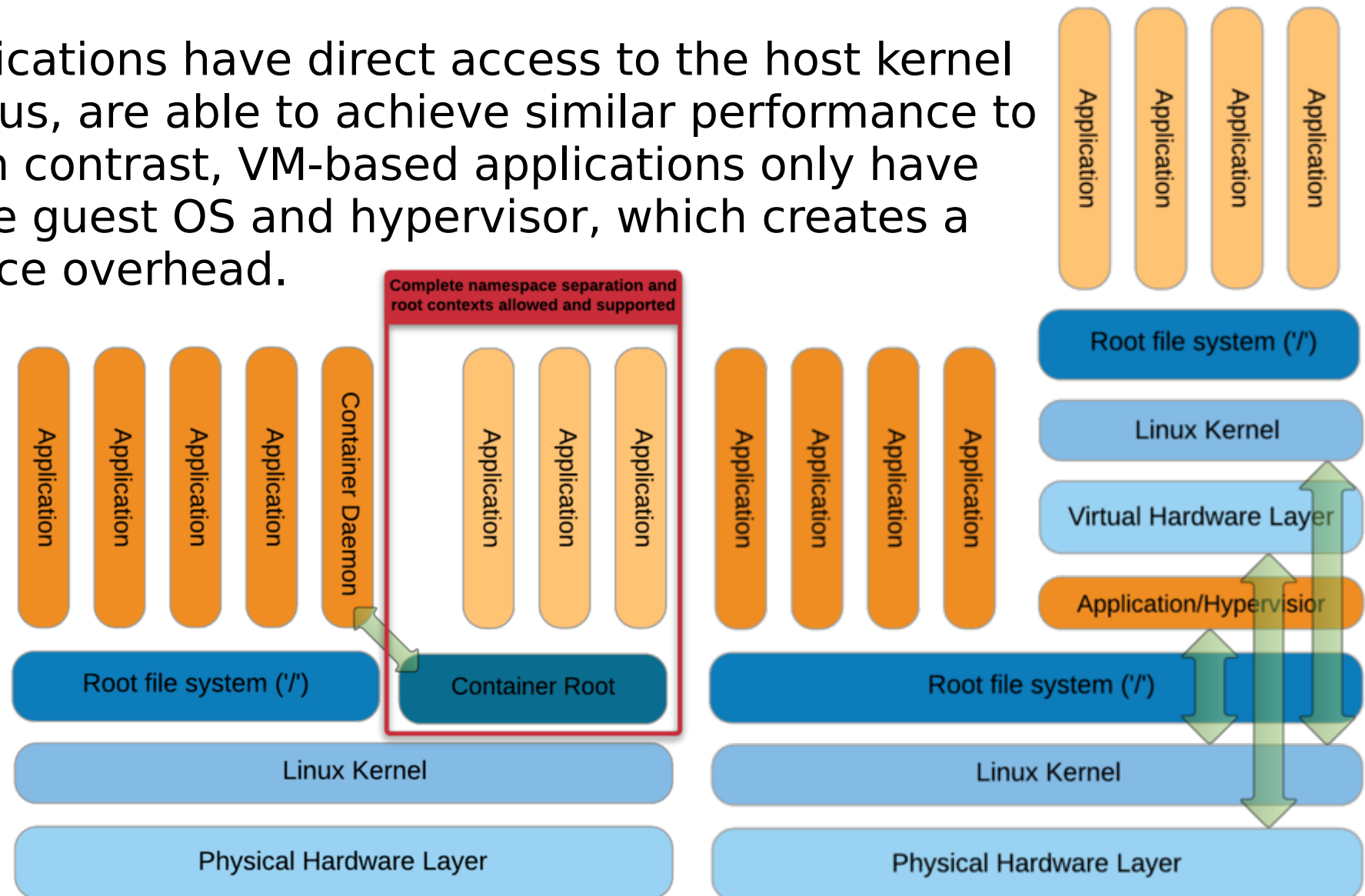


- *A (software) container is an abstraction for a set of technologies that aim to solve the problem of how to get software to run reliably when moved from one computing environment to another.*
- A container image is simply a file (or collection of files) saved on disk that stores everything you need to run a target application or applications: code, runtime, system tools, libraries, etc.
- A container process is simply a standard (Linux) process running on top of the underlying host's operating system and kernel

Containers vs. Virtual Machines



Container-based applications have direct access to the host kernel and hardware and, thus, are able to achieve similar performance to native applications. In contrast, VM-based applications only have indirect access via the guest OS and hypervisor, which creates a significant performance overhead.



Advantages of Containers

- **Performance** – Near-native application performance
- **Freedom** – Bring your own software environment
- **Reproducibility** – Package complex software applications into easy to manage, verifiable software units
- **Compatibility** – Built on open standards available in all major Linux distributions
- **Portability** – Build once, run (almost) anywhere

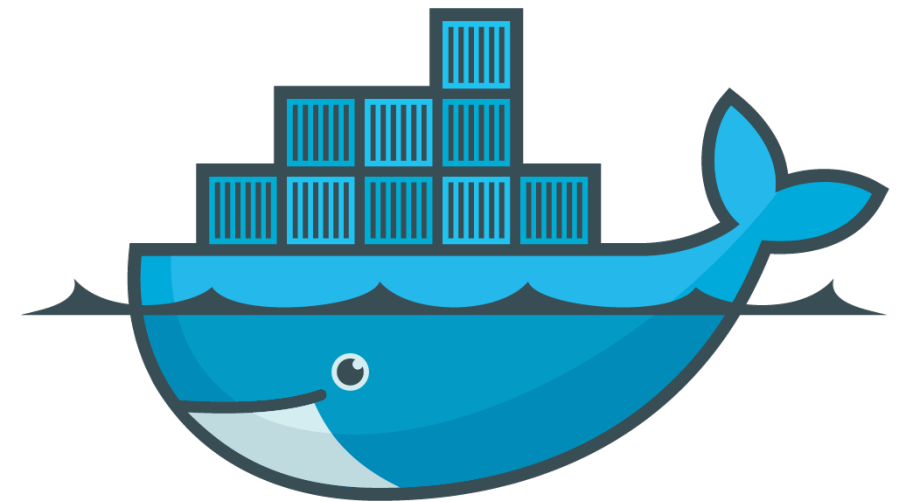
Limitations of Containers

#9

- **Architecture dependent** – Always limited by CPU architecture (x86_64, ARM) and binary format (ELF)
- **Portability** - Requires glibc and kernel compatibility between host and container; also requires any other kernel-user space API compatibility (e.g., OFED/IB, NVIDIA/GPUs)
- **Filesystem isolation** - filesystem paths are (mostly) different when viewed inside and outside container

Why singularity? DOCKER?

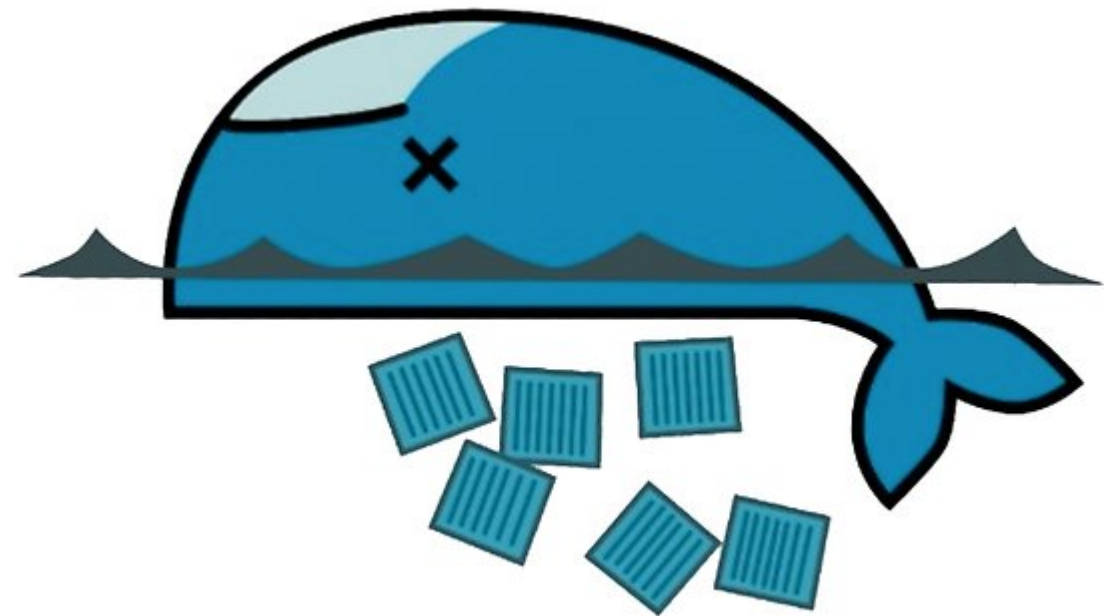
- Docker is the most common container format.
- Provides container environment
- For network services
- Easy to use
- DockerHub
- Industry Standard.



docker

Docker on HPC?

- HPC systems are **shared** resources
- Docker's **security** model is designed to support trusted users running trusted containers; e.g., users can escalate to root
- Docker not designed to support **batch-based workflows**
- Docker not designed to support tightly-coupled, highly distributed parallel applications (**MPI**).
- Docker is changing



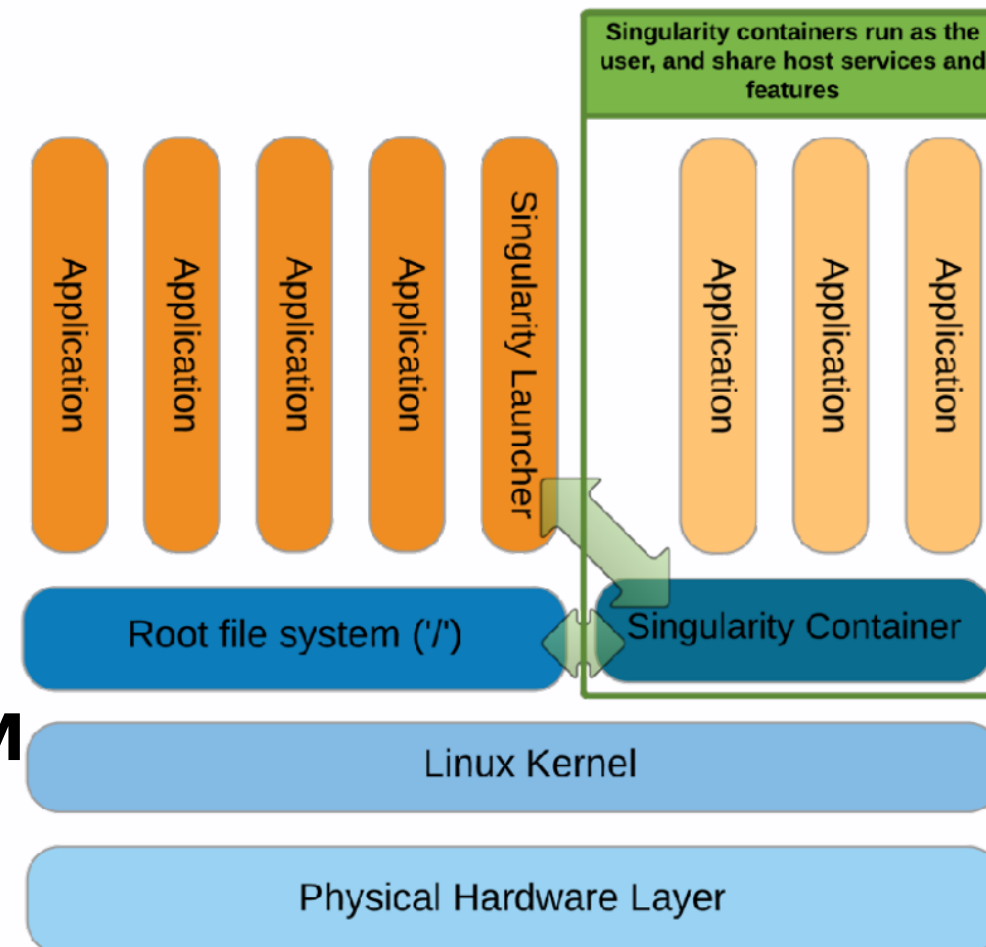
Singularity ?

- A Container Platform for HPC
- **Reproducible, portable, sharable**, and distributable containers
- No trust security model: **untrusted users running untrusted containers**
- Support HPC hardware and scientific applications



Features of Singularity

- Each container is a **single image** file
- **No root owned daemon process**
- **No** user contextual changes or **root escalation** allowed; user inside container is always the same user who started the container
- Supports **shared**/multitenant **resource** environments
- Supports **HPC hardware**: Infiniband, GPUs
- Supports HPC applications: **MPI, SLURM**



Essential Singularity



singularity [options] <subcommand> [subcommand options] ...

- **build**: Build your **own** container from scratch using a Singularity definition file; **download** and assemble any existing Singularity container; or **convert** your containers from one format to another (e.g., from Docker to Singularity)
- **shell**: Spawn an interactive shell session in your container.
- **exec**: Execute an arbitrary command within your container.

Build a singularity container

#15

```
→ singularity git:(master) x sudo singularity build lolcow.sif library://sylabs-jms/testing/lolcow  
  
WARNING: 'nodev' mount option set on /tmp, it could be a source of failure during build process  
INFO: Starting build...  
INFO: Using cached image  
INFO: Verifying bootstrap image /root/.singularity/cache/library/sha256.5022b5e7c7249c40119a875c1ace0700ce  
d4099e077acc75d0132190254563a4  
WARNING: integrity: signature not found for object group 1  
WARNING: Bootstrap image could not be verified, but build will continue.  
INFO: Creating SIF file...  
INFO: Build complete: lolcow.sif
```

Pull a singularity container

```
→ ~ singularity pull ubuntu.sif docker://ubuntu:20.04
INFO:      Converting OCI blobs to SIF format
WARNING: 'nodev' mount option set on /tmp, it could be a source of failure du
ring build process
INFO:      Starting build...
Getting image source signatures
Copying blob 7b1a6ab2e44d done
Copying config e7132beceb done
Writing manifest to image destination
Storing signatures
2021/11/23 16:19:23  info unpack layer: sha256:7b1a6ab2e44dbac178598dabe7cff5
9bd67233dba0b27e4fbd1f9d4b3c877a54
2021/11/23 16:19:23  warn xattr{etc/gshadow} ignoring ENOTSUP on setxattr "us
er.rootlesscontainers"
2021/11/23 16:19:23  warn xattr{/tmp/build-temp-2959038344/rootfs/etc/gshadow
} destination filesystem does not support xattrs, further warnings will be su
ppressed
INFO:      Creating SIF file...
```

Singularity Registries: Docker, shub, sylab cloud, datalad, ...

Interacting with a Singularity container



- SHELL
- Overlapping e.g. /etc
- Apps inside: apt

```
→ ~ cat /etc/lsb-release
DISTRIB_ID=ManjaroLinux
DISTRIB_RELEASE=21.2.0
DISTRIB_CODENAME=Qonos
DISTRIB_DESCRIPTION="Manjaro Linux"
→ ~ singularity shell ubuntu.sif
Singularity> cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.3 LTS"
Singularity> which apt
/usr/bin/apt
Singularity> apt --version
apt 2.0.6 (amd64)
Singularity> exit
exit
→ ~ █
```


Running a Singularity Container

- EXEC
- Simply run an application inside like a normal bash command
- Arguments allowed, Pipes welcome.

```
→ ~ singularity exec ubuntu.sif python --version
FATAL:  "python": executable file not found in $PATH
→ ~ singularity exec ubuntu.sif apt --version
apt 2.0.6 (amd64)
→ ~ python --version
Python 3.9.7
→ ~ █
```

Singularity Definition Files

- Definition files == **Recipe**
- It is a **manifest** of all software to be installed within the container, environment variables to be set, files to be added, directories to be mounted, container metadata, etc.
- You can even write a **help** section, or define modular components in the container (apps)
- **Miniconda3 Example** → Hands-On
- **IMGW Repository** - **Gitlab**

```
1 # Bootstrap: library
2 # From: mblaschek/imgw/ubuntu:18.04
3 Bootstrap: localimage
4 From: ubuntu.sif
5
6 %labels
7
8 ... APPLICATION_NAME miniconda3
9 ... APPLICATION_VERSION py39-4.9.2-Linux-x86_64
10 ... APPLICATION_URL https://docs.conda.io
11
12 ... AUTHOR_NAME Michael Blaschek
13 ... AUTHOR_EMAIL michael.blaschek@univie.ac.at
14
15 ... LAST_UPDATED 20211118
16
17 %setup
18
19 %environment
20
21 ... # Set the conda distribution type, its version number, the python
22 ... # version it utilizes, the root and installation directories where
23 ... # the distribution will be installed within the container, and the
24 ... # root URL to the installer
25 ... export CONDA_DISTRIBUTION='miniconda'
26 ... export CONDA_VERSION='3'
27 ... export CONDA_PYTHON_VERSION='py39'
28 ... export CONDA_INSTALLER_VERSION='4.9.2'
29 ... export CONDA_ARCH='Linux-x86_64'
30 ... export CONDA_INSTALL_DIR="/opt/${CONDA_DISTRIBUTION}${CONDA_VERSION}"
31
32 ... # Set PATH to conda distribution
33 ... export PATH="${CONDA_INSTALL_DIR}/bin:${PATH}"
34
35 %post -c /bin/bash
36
37 ... # Set operating system mirror URL
38 ... export MIRRORURL='http://at.archive.ubuntu.com/ubuntu'
39
40 ... # Set operating system version
41 ... export OSVERSION='bionic'
42
43 ... # Set system locale
44 ... export LC_ALL='C'
45
46 ... # Set debian frontend interface
47 ... export DEBIAN_FRONTEND='noninteractive'
48
49 ... # Upgrade all software packages to their latest versions
50 ... apt-get -y update && apt-get -y upgrade
51
52 ... cd /tmp
```

Singularity Recipes @IMGW

- A **repository** of definition files for building Singularity containers around the software applications, frameworks, and libraries you need to run on high-performance computing systems.
 - JET – Multi-Node, MPI
 - SRVX1
 - VSC 4, 5, 6 – Multi-Node, MPI, GPU
- Not complete yet, but it will grow.

Workflow

- **Build** your Singularity containers on a local system where you have root or sudo access
 - **Sign** your container
- **Transfer** your Singularity containers to the HPC system where you want to run them
 - **Share** your container
- **Run** your Singularity containers on that HPC system



imgw

Institut für Meteorologie
und Geophysik



**universität
wien**

Faculty of Earth Sciences,
Geography and Astronomy

Questions?

Hands-On - 30min

- Please go to <https://gitlab.phaidra.org/imgw/singularity/workshop>
- Schedule:
 - Introduction to commandline of singularity
 - Definition files
 - Connecting to VM on JET01 (Credential via Zoom)
 - Building containers

