

ogs-container-maker: create and run OGS within portable Linux container

OGS Community Meeting 2019

Lars Bilke

Wednesday 27th March, 2019

Helmholtz Centre for Environmental Research GmbH – UFZ

Table of contents

1. Motivation and approach

2. ogs-container-maker

3. Usage

4. Outlook & conclusions

Motivation and approach

Challenges in configuration and setup of complex scientific simulation software

- Complex library and tool dependencies
- Different execution environments (workstation vs. HPC system)
- Dependencies for older software versions may not be available anymore

⇒ Full reproducibility of scientific result data of simulations is a challenging task and may even be impossible.

Approach

Encapsulate code, dependencies, execution environment and data into a portable unit using container technologies

- ▶ Allow encapsulation of complex software setups
- ▶ Recent developments in scalable (HPC-enabled) container technologies, e.g. Singularity [1]
- ▶ Light-weight form of computer virtualization \implies High performance [2]

Objectives

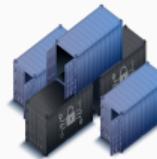
Outcome

- Provide portable containerized simulation software
 - Hiding complex software setups from the user
 - Executable on workstations (Linux, macOS) and arbitrary HPC systems
 - Integrate into typical scientific workflows
- Ensure reproducible simulation results
 - Signed and check-summed container image enhanced with meta-data about the software configuration
 - Archivable container images containing everything to reproduce results
 - Investigate long-term archival to meet regulations in safety-critical research areas

ogs-container-maker

Singularity¹

- Linux container technology, compatible to Docker images
- HPC-enabled by using regular schedulers (single executable)
- Easy to use, flexible workflows, image mobility
- Cryptographically signed immutable images (optional)



¹<http://singularity.lbl.gov>

Container generation: definition

Listing 1: example.def – Example singularity definition file (recipe)²

```
1 Bootstrap: docker
2 From: ubuntu
3
4 %help
5   Help me. I'm in the container.
6
7 %post
8   apt-get update
9   apt-get install -y tool_x
10
11 %runscript
12   exec tool_x "$@"
```

²Singularity defintion file syntax:

https://www.sylabs.io/guides/2.6/user-guide/container_recipes.html

Container generation: definition (real-world example)

```

1  FROM ubuntu:17.10
2
3  # Generated with https://github.com/ufz/ogs-container-maker/commit/cb0c25d
4
5  RUN apt-get update -y && \
6    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
7      wget \
8      tar \
9      curl &&
10   rm -rf /var/lib/apt/lists/*
11
12  # OGS compiler
13  RUN apt-get update -y && \
14    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends software-properties-common &&
15    apt-add-repository ppa:ubuntu-toolchain-r/test -y &&
16    apt-get update -y &&
17    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
18      gcc-6 \
19      g++-6 \
20      gfortran-6 &&
21   rm -rf /var/lib/apt/lists/*
22  RUN update-alternatives --install /usr/bin/gcc gcc $(which gcc-6) 30 &&
23  update-alternatives --install /usr/bin/g++ g++ $(which g++-6) 30 &&
24  update-alternatives --install /usr/bin/gfortran gfortran $(which gfortran-6) 30 &&
25  update-alternatives --install /usr/bin/gcov gcov $(which gcov-6) 30
26
27  # OGS base building block
28  # Python
29  RUN apt-get update -y && \
30    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
31      python3 \
32      libpython-dev &&
33   rm -rf /var/lib/apt/lists/*
34
35  RUN apt-get update -y && \
36    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
37      python3-gip \
38      python3-setuptools \
39      python3-wheel &&
40   rm -rf /var/lib/apt/lists/*
41  RUN pip3 install --upgrade pip
42
43  RUN apt-get update -y &&
44    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
45      python3-pip \
46      python3-setuptools \
47      python3-wheel &&
48   rm -rf /var/lib/apt/lists/*
49  RUN pip3 install virtualenv
50
51  # CMake version 3.13.4
52  RUN apt-get update -y &&
53    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
54      wget &&
55   rm -rf /var/lib/apt/lists/*
56  RUN mkdir -p /var/tmp && wget -q -nc --no-check-certificate -P /var/tmp https://cmake.org/files/v3.13/cmake-3.13.4-linux-x86_64.sh &&
57   /bin/sh /var/tmp/cmake-3.13.4-linux-x86_64.sh --prefix=/usr/local --skip-license &&
58   rm -rf /var/tmp/cmake-3.13.4-Linux-x86_64.sh
59
60  RUN apt-get update &&
61   curl -s "https://packages.cloud.google.com/install/repositories/github/git-lts/script.deb.sh" | bash
62
63  RUN apt-get update -y &&
64    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends software-properties-common &&
65    apt-add-repository ppa:git-core/ppa &&
66
67
68  apt-get update -y &&
69    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
70      git \
71      git-lfs \
72      make \
73      ninja-build &&
74   rm -rf /var/lib/apt/lists/*
75  RUN git lfs install &&
76   mkdir -p /apps /scratch /lustre /work /projects
77
78  # Package manager Conan building block
79  # pip
80  RUN apt-get update -y &&
81    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
82      python3-pip \
83      python3-setuptools \
84      python3-wheel &&
85   rm -rf /var/lib/apt/lists/*
86  RUN pip3 install cmake==3.13.2
87  RUN pip3 install numpy &&
88   chmod 777 /opt/conan
89
90  ENV CONAN_USER_HOME=/opt/conan
91  LABEL org.opensuseos.py.conan \
92    org.opensuseos.py.conan.user.home=/opt/conan \
93    org.opensuseos.py.conan.version=3.13.2
94
95  # cppcheck version 1.87
96  RUN mkdir -p /var/tmp && wget -q -nc --no-check-certificate -P /var/tmp https://github.com/danmarr/cppcheck/archive/1.87.tar.gz &&
97   mkdir -p /var/tmp & tar -x -f /var/tmp/1.87.tar.gz -C /var/tmp & rm -rf /var/tmp/1.87 \
98  RUN mkdir -p /var/tmp/build && cd /var/tmp/build && cmake -DCMAKE_INSTALL_PREFIX=/usr/local/cpptools -DCMAKE_BUILD_TYPE=RELEASE /var/tmp/cppcheck \
99   -1.87 &&
100  cmake --build /var/tmp/build --target install -- -j$(nproc) &&
101   rm -rf /var/tmp/1.87.tar.gz /var/tmp/build /var/tmp/cppcheck-1.87
102
103  ENV PATH=/usr/local/cpptools/bin:$PATH
104
105  RUN apt-get update -y &&
106    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
107      doxygen \
108      graphviz \
109      libeigen3-base &&
110   rm -rf /var/lib/apt/lists/*
111
112  # pip
113  RUN apt-get update -y &&
114    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
115      python3-pip \
116      python3-setuptools \
117      python3-wheel &&
118   rm -rf /var/lib/apt/lists/*
119  RUN pip3 install gevent
120
121  # Package manager Conan building block
122  RUN apt-get update -y &&
123    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
124      cache &
125   rm -rf /var/lib/apt/lists/
126  RUN mkdir -p /opt/cache & chmod 777 /opt/cache
127
128  ENV CCACHE_DIR=/opt/cache \
129    CCACHE_MAXSIZE=15G \
130    CCACHE_SLOPPINESS=pchDefines,time_macros
131
132  LABEL ccache.dir=/opt/cache \
133    ccache.size=15G

```

ogs-container-maker: create and run OGS within portable Linux container, Lars Bilke

Container generation: building image

Turn the definition into an image:

```
1 sudo singularity build example.simg example.def
```

This creates the container image **example.simg**

hpc-container-maker [3] by NVidia³

- Generates container definitions from a high level Python recipe
- Contains a library of HPC building blocks
- Recipes become programmable, e.g. generate multiple variations
- Creates Singularity and Docker definitions

³<https://github.com/NVIDIA/hpc-container-maker>

Listing 2: basic.py – Install GNU compiler on Ubuntu

```
1 Stage0.baseimage('ubuntu:16.04')
2 Stage0 += packages(ospackages=['gcc', 'g++', 'gfortran'])
```

Listing 3: Generate Singularity definition and build image

```
1 hpccm --recipe basic.py --format singularity > basic.def
2 sudo singularity build basic.simg basic.def
```

ogs-container-maker⁴ builds upon hpc-container-maker and provides

- Building blocks for OGS requirements (e.g. VTK, PETSc)
- A parameterized combinatorial recipe for building OGS and all its requirements
- Scripts for building common OGS configurations

⁴<https://github.com/ufz/ogs-container-maker>

ogs-container-maker: options i

Listing 4: Combinatorial options

```
1 All combinations of the given options will be generated
2
3 --format [{docker,singularity} [{docker,singularity} ...]]
4 --pm [{system,conan} [{system,conan} ...]]
5             Package manager to install third-party dependencies
6             (default: ['conan'])
7 --ompi [OMPI [OMPI ...]]
8             OpenMPI version, e.g. 2.1.1, 2.1.5, 3.0.1, 3.1.2
9             (default: ['off'])
10 --ogs [OGS [OGS ...]]
11             OGS GitHub repo in the form 'user/repo@branch' or
12             'off' to disable OGS building (default:
13             ['ufz/ogs@master'])
14 --cmake_args [CMAKE_ARGS [CMAKE_ARGS ...]]
15             CMake argument sets have to be quoted and **must**
16             start with a space. e.g. --cmake_args ' -DFIRST=TRUE
17             -DFOO=BAR' ' -DSECOND=TRUE' (default: [''])
```

ogs-container-maker: options ii

Listing 5: Build options

```
1 --build, -B          Build the images from the definition files (default:  
2                           False)  
3 --upload, -U          Upload Docker image to registry (default: False)  
4 --registry REGISTRY   The docker registry the image is tagged and uploaded  
5                           to. (default: registry.opengeosys.org/ogs/ogs)  
6 --convert, -C          Convert Docker image to Singularity image (default:  
7                           False)  
8 --runtime-only, -R     Generate multi-stage Dockerfiles for small runtime  
9                           images (default: False)
```

ogs-container-maker: options iii

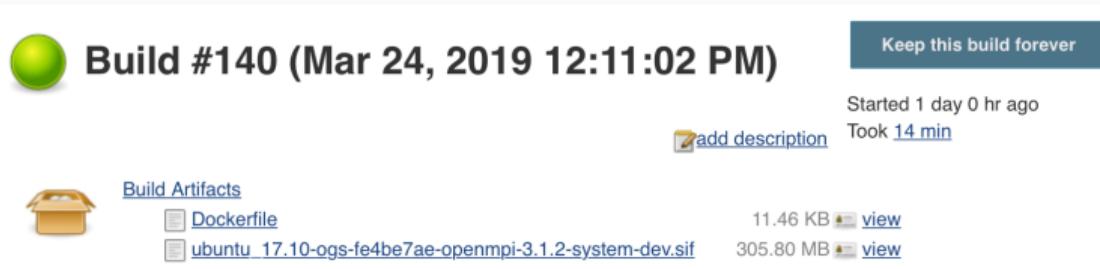
Listing 6 : Additional options

```
1 --base_image BASE_IMAGE
2                                     The base image. 'centos:7' is supported too. (default:
3                                         ubuntu:17.10)
4 --clang                         Use clang instead of gcc (default: False)
5 --gui                            Builds the GUI (Data Explorer) (default: False)
6 --docs                           Setup documentation requirements (Doxygen) (default:
7                                         False)
```

Usage

Available at jenkins.opengeosys.org/job/ufz/job/ogs-container-maker/job/master/build

- User defined container image generation
 - Specified OGS configurations are bundled into images
 - Image is generated by Jenkins
 - Generated images are temporarily archived on Jenkins for easy access



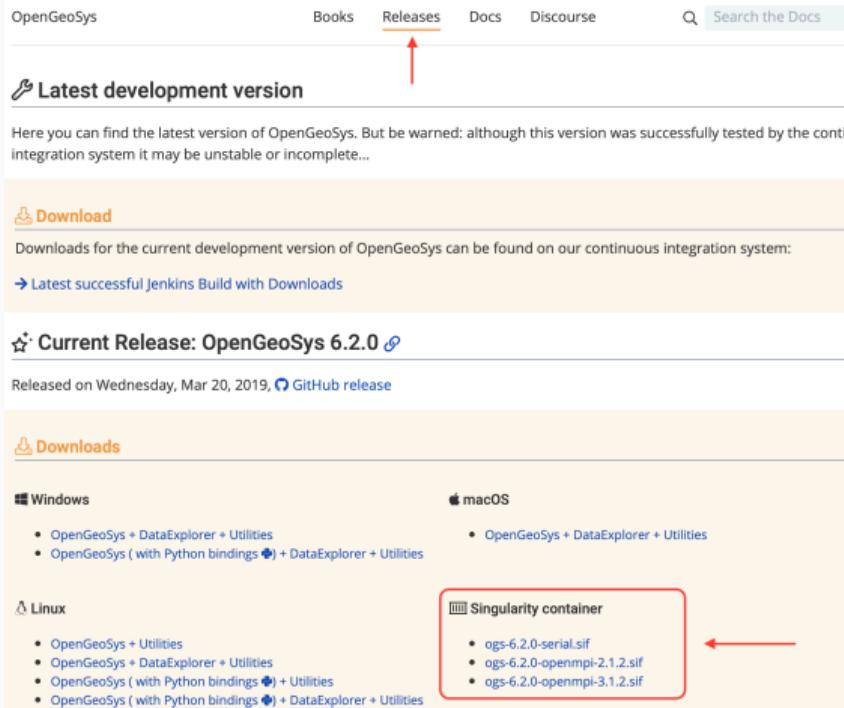
The screenshot shows a Jenkins build summary for Build #140, which was run on March 24, 2019, at 12:11:02 PM. The build status is green. A button labeled "Keep this build forever" is visible. The build took 14 minutes and started 1 day ago. There is an "add description" link. Below the summary, there is a "Build Artifacts" section with a folder icon. It lists two artifacts: "Dockerfile" (11.46 KB) and "ubuntu_17.10-ogs-fe4be7ae-openmpi-3.1.2-system-dev.sif" (305.80 MB). Each artifact has a "view" link next to it.

Artifact	Size	Action
Dockerfile	11.46 KB	view
ubuntu_17.10-ogs-fe4be7ae-openmpi-3.1.2-system-dev.sif	305.80 MB	view

Container maker service on Jenkins

ogs	<input type="text" value="ufz/ogs@master"/>
	Build OGS in container (Github user/repo@branch)
format	<input type="text" value="docker"/>
	Container format, e.g.: docker singularity
openmpi_versions	<input type="text" value="3.1.2"/>
	OpenMPI versions, e.g.: off 2.1.2 2.1.5 3.0.1 3.1.2, ...
pm	<input type="text" value="system"/>
	Package manager to install third-party libs, e.g.: system conan
cmake	<input type="text"/>
	CMake args, have to be quoted and must start a space e.g. "' -DFOO=BAR"
convert	<input checked="" type="checkbox"/>
	Convert docker image to Singularity?
runtime	<input type="checkbox"/>
	Create a runtime only image (contains just the built binaries and runtime dependencies)
deploy	<input type="checkbox"/>
	Deploy Singularity images
	<input type="button" value="Build"/>

Download prebuilt container



The screenshot shows the 'Releases' section of the OpenGeoSys documentation. An orange arrow points from the 'Releases' tab in the top navigation bar down to the 'Latest development version' heading. A red box highlights the 'Singularity container' section under the 'Linux' heading, which contains three download links: 'ogs-6.2.0-serial.sif', 'ogs-6.2.0-openmpi-2.1.2.sif', and 'ogs-6.2.0-openmpi-3.1.sif'. A red arrow points from the bottom right towards this red box.

OpenGeoSys Books Releases Docs Discourse  Search the Docs

Latest development version

Here you can find the latest version of OpenGeoSys. But be warned: although this version was successfully tested by the contin integration system it may be unstable or incomplete...

Download

Downloads for the current development version of OpenGeoSys can be found on our continuous integration system:

→ [Latest successful Jenkins Build with Downloads](#)

Current Release: OpenGeoSys 6.2.0

Released on Wednesday, Mar 20, 2019, [GitHub release](#)

Downloads

Windows

- [OpenGeoSys + DataExplorer + Utilities](#)
- [OpenGeoSys \(with Python bindings\) + DataExplorer + Utilities](#)

macOS

- [OpenGeoSys + DataExplorer + Utilities](#)

Linux

- [OpenGeoSys + Utilities](#)
- [OpenGeoSys + DataExplorer + Utilities](#)
- [OpenGeoSys \(with Python bindings\) + Utilities](#)
- [OpenGeoSys \(with Python bindings\) + DataExplorer + Utilities](#)

Singularity container

- [ogs-6.2.0-serial.sif](#)
- [ogs-6.2.0-openmpi-2.1.2.sif](#)
- [ogs-6.2.0-openmpi-3.1.sif](#)

Container usage

```
1 singularity exec ogs-6.2.0-serial.sif ogs some/path/project.prj
```

- Starts a container,
- mounts home directory inside the container,
- passes current working directory,
- runs OGS exe with passed project file.

Container usage

Listing 7: Container usage examples

```
1 # Run serial benchmark with output validation (via vtkdiff)
2 singularity exec ogs-6.2.0-serial.sif ogs -o _out -r [ogs-sources]/Tests/Data/Mechanics/Linear [ogs-sources]/
   Tests/Data/Mechanics/Linear/disc_with_hole.prj
3 # Run parallel benchmark with MPI
4 mpirun -np 4 singularity exec ogs-6.2.0-openmpi-2.1.2.sif ogs -o _out_mpi [ogs-sources]/Tests/Data/Mechanics/
   Linear/disc_with_hole.prj
5 # Run other contained executables, e.g. vtkdiff
6 singularity exec ogs-6.2.0-serial.sif vtkdiff --help
7
8 # Shell into container
9 singularity shell ogs-6.2.0-serial.sif
10 # List files in the container
11 Singularity ogs-6.2.0-serial.sif:...> ls /usr/local/ogs/bin
12 ... ogs tetgen vtkdiff
13 # Exit the container and get back to your hosts shell
14 Singularity ogs-6.2.0-serial.sif:...> exit
```

Outlook & conclusions

- Check MPI compatibility (versions, host vs. container) on HPC environments
- Utilize multi-application / -configuration container with SCIF [4]
- Conduct HPC simulations on multiple environments
- Implement signing of containers
- Implement user feature requests ...

Conclusions

- Encapsulate OGS with all development and/or runtime dependencies in a single file (+ data)
- Use ogs-container-maker to generate arbitrary container definitions
- Use the Jenkins interface to generate container images
- Build the container once, run everywhere⁵

- Download prebuilt container: opengeosys.org/releases
- ogs-container-maker: github.com/ufz/ogs-container-maker

⁵...has to be proven...

Thank you!

lars.bilke@ufz.de

References i

- [1] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer.
Singularity: Scientific containers for mobility of compute.
PLOS ONE, 12(5):e0177459, May 2017.
- [2] Carlos Arango, Rémy Dernat, and John Sanabria.
Performance Evaluation of Container-based Virtualization for High Performance Computing Environments.
arXiv preprint arXiv:1709.10140, 2017.
- [3] Scott McMillan.
Making containers easier with hpc container maker.
In *In HPCSYPROS18: HPC System Professionals Workshop*, Dallas, TX, November 2018.
- [4] Vanessa Sochat.
The Scientific Filesystem.
GigaScience, 7(5), 03 2018.

References ii