singularity 的基本用法

一、简介

Singularity 是一种容器技术，可以让用户在不同的计算机上运行相同的应用程序。而不需要担心环境差异。singularity与docker功能相似，但是相比与docker需要root权限，或者专门添加用户组，singularity在非root时也可使用，且轻量级，修改方便。singularity镜像有两种格式：sif格式可用于部署；sanbox格式是可写的文件系统，用于开发过程，方便根据需要修改其中的内容。两种格式之间相互转换，即开发完成后转换为sif。具体可参考官网。

二、官网

Singularity已经加入了Linux基金会，现在是Apptainer!

新的官网网址为：https://apptainer.org/docs/。也可以通过旧的官网网址（https://singularity.lbl.gov/）进行跳转。

Singularity 技术几个重要概念：

- 容器(container): 容器是一个包含用户软件和依赖的镜像系统，可独立运行某一条或者多条命令。Singularity没有镜像的概念，用户创建和运行的都是一个一个容器。
- SIF（Singularity Image File）：压缩后的只读（read-only）的Singularity镜像文件，是生产使用的主要形式。
- Sandbox： 可写(writable)的容器存在形式，是文件系统中的一个目录，常用于开发或者创建自己的容器，是开发使用的主要形式。

三、singularity 常用命令

主要有四种操作：run、exec、shell和test（test不常用）。可以在命令行中通过--help参数查看各个指令的文档，比如singularity run --help，可以查看run指令的操作文档。

1、构建容器

可以使用 build 命令来构建一个新的Singularity容器，例如：

```
singularity build lolcow.simg shub://GodloveD/lolcow
# 第一个参数 (lolcow.simg) 指定容器的路径和名称。
# 第二个参数 (shub://GodloveD/lolcow) 给出了从中下载的容器库 URI。
# 默认情况下，容器将转换为压缩的只读 SIF。如果希望容器采用可写格式，请使用 –sandbox 选项。

singularity build my-container.sif my-container.def
# 其中，my-container.def是一个定义了容器的文本文件，该文件指定了容器的基础镜像、安装的软件和配置选项等。 此外，build还可以通过以下目标创建容器：
# - 以library:// 开头的 URI 从容器库生成
# - 以docker:// 开头的 URI 从 Docker Hub 构建
# - 以shub:// 开头的 URI 从奇点中心构建
# - 本地计算机上现有容器的路径
# - 从沙盒构建的目录的路径
# 详细用法见[官方文档 Build a Container](https://apptainer.org/user-docs/master/build_a_container.html)
```

```
(base) [yanlp@ln02 docker_library]$ singularity build lolcow.simg shub://GodloveD/lolcow
INFO:    Starting build...
INFO:    Downloading shub image
87.6MiB / 87.6MiB [==========================================================================================] 100 % 5.1 MiB/s 0s
INFO:    Creating SIF file...
INFO:    Build complete: lolcow.simg
(base) [yanlp@ln02 docker_library]$ ll
total 182400
-rwxr-xr-x 1 yanlp scnu 94824197 Aug 10 22:50 lolcow_latest.sif
-rwxr-xr-x 1 yanlp scnu 91865088 Aug 10 23:00 lolcow.simg
(base) [yanlp@ln02 docker_library]$
```

SIF和sandbox两种格式的镜像是可以相互转换的；

```
# 1. 将SIF格式的容器转换成sandbox;
singularity build --sandbox centos76 centos76.sif

# 2. 将sandbox容器镜像转化成SIF格式;
singularity build centos76.sif centos76
```

```
(base) [yanlp@ln02 docker_library]$ singularity build --sandbox lolcow lolcow_latest.sif
INFO:    Starting build...
INFO:    Verifying bootstrap image lolcow_latest.sif
INFO:    Creating sandbox directory...
INFO:    Build complete: lolcow
(base) [yanlp@ln02 docker_library]$ ll
total 182401
drwxr-xr-x 18 yanlp scnu     4096 Aug 10 23:11 lolcow
-rwxr-xr-x  1 yanlp scnu 94824197 Aug 10 22:50 lolcow_latest.sif
-rwxr-xr-x  1 yanlp scnu 91865088 Aug 10 23:00 lolcow.simg
(base) [yanlp@ln02 docker_library]$
```

2、拉取镜像

可以通过 pull 命令从容器库中拉取镜像，例如：

```
$ singularity pull library://lolcow
```

```
(base) [yanlp@ln02 docker_library]$ singularity pull library://lolcow
INFO:    Using cached image
(base) [yanlp@ln02 docker_library]$ ll
total 92672
-rwxr-xr-x 1 yanlp scnu 94824197 Aug 10 22:50 lolcow_latest.sif
```

也可通过 `singularity pull` 或 `singularity build` 命令从外部容器仓库下载容器到本地，存储为SIF文件。用户通过指定不同的容器的URI，从不同的源下载容器。

例如：可以通过如下的命令从Docker Hub（存储标准的Docker镜像）下载容器，自动转换成Singularity容器，存储为SIF格式:

```
singularity build ./jason-tensorflow.sif docker://tensorflow/tensorflow:latest-gpu

# 或者

singularity pull ./jason-tensorflow.sif docker://tensorflow/tensorflow:latest-gpu
```

2、运行容器

可以使用 run 命令来运行一个Singularity容器，例如：

```
$ singularity run my-container.sif
```

```
(base) [yanlp@ln02 docker_library]$ singularity run lolcow_latest.sif
 _______________________________
< Thu Aug 10 22:51:11 CST 2023 >
 -------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
(base) [yanlp@ln02 docker_library]$
```

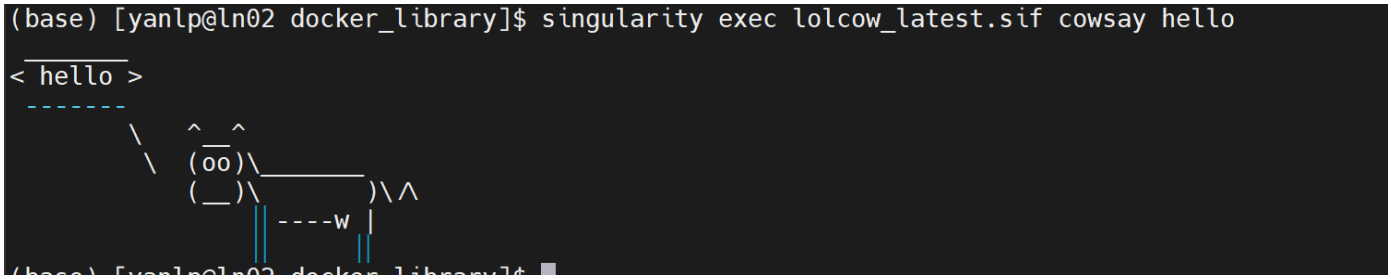
3、执行命令

通过 exec 命令可以在容器内执行程序。例如：

```
$ singularity exec my-container.sif my-command
```

其中，my-command是在容器中运行的命令。此命令将在容器中启动一个新的进程，并运行指定的命令。

示例：在lolcow_latest 容器中执行cowsay程序，并让牛说出hello。

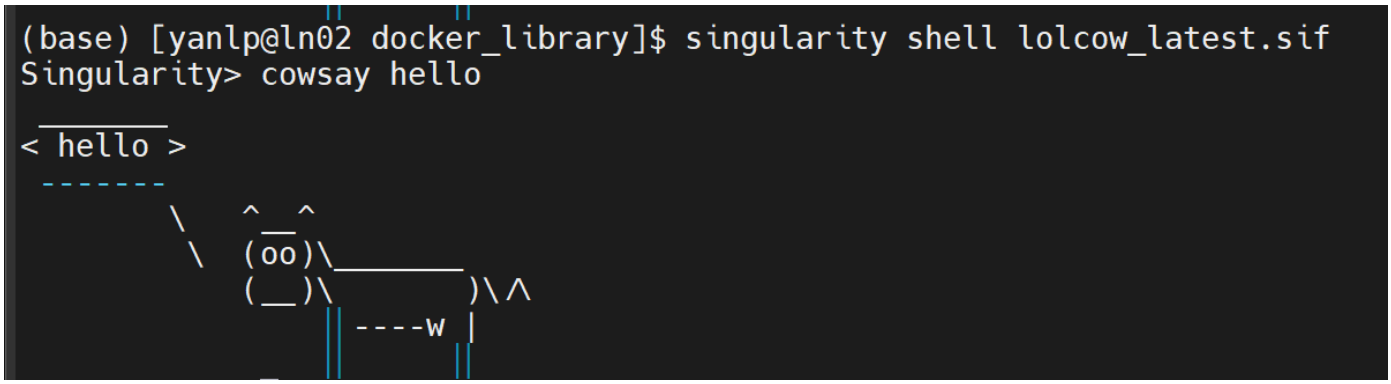


4、运行交互式shell

使用 shell 命令可以在容器中生成一个新的Shell，就像我们使用虚拟机一样。

```
$ singularity shell my-container.sif
```

示例：进入lolcow容器的交互式shell，您可以在其中执行任何可用的命令，例如：执行cowsay程序，并让牛说出hello。



5、查看容器信息

可以使用 inspect 命令来查看一个Singularity容器的信息，例如：

```
$ singularity inspect my-container.sif
```

此命令将输出容器的元数据、环境变量、挂载点等信息。

```
(base) [yanlp@ln02 docker_library]$ singularity inspect lolcow_latest.sif
org.label-schema.build-arch: amd64
org.label-schema.build-date: Friday_29_July_2022_9:23:15_CDT
org.label-schema.schema-version: 1.0
org.label-schema.usage.singularity.deffile.bootstrap: docker
org.label-schema.usage.singularity.deffile.from: ubuntu:22.04
org.label-schema.usage.singularity.version: 3.10.2-dirty
(base) [yanlp@ln02 docker_library]$
```

四、singularity 和 slurm 简单集成使用示例

1、在slurm作业中运行singularity命令

```
(base) [yanlp@ln02 docker_library]$ cat my_job.sh
#!/bin/bash
#SBATCH --job-name=test # 作业名
#SBATCH --partition=gpu # cpu 队列
#SBATCH -n 20 # 总核数 20
#SBATCH --ntasks-per-node=20 # 每节点核数
#SBATCH --output=%j.out
#SBATCH --error=%j.err

singularity exec lolcow_latest.sif cowsay hello
(base) [yanlp@ln02 docker_library]$ sbatch -p gpu-debug my_job.sh
Submitted batch job 190630
(base) [yanlp@ln02 docker_library]$ cat 1906
190629.err  190629.out  190630.err  190630.out
(base) [yanlp@ln02 docker_library]$ cat 190630.out
 _______
< hello >
 -------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
(base) [yanlp@ln02 docker_library]$
```

2、使用slurm作为调度器的集群上使用singularity

```
(base) [yanlp@ln02 docker_library]$ salloc -p gpu-debug -c 1 -t 00-00:10 --mem=4000
salloc: Pending job allocation 190631
salloc: job 190631 queued and waiting for resources
salloc: job 190631 has been allocated resources
salloc: Granted job allocation 190631
salloc: Waiting for resource configuration
salloc: Nodes gpu024 are ready for job
(base) [yanlp@ln02 docker_library]$ ^C
(base) [yanlp@ln02 docker_library]$ squeue -u yanlp
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
            190631 gpu-debug interact    yanlp  R       1:08      1 gpu024
(base) [yanlp@ln02 docker_library]$ ssh gpu024
Last login: Mon Jul 31 18:48:31 2023 from ln02
(base) [yanlp@gpu024 ~]$ which singularity
/usr/bin/singularity
(base) [yanlp@gpu024 ~]$ singularity shell docker_library/lolcow_latest.sif
Singularity>
```

3、其他案例

- https://docs.rc.fas.harvard.edu/kb/singularity-on-the-cluster/：该网页是哈佛大学的文档，介绍了如何在一个使用slurm作为调度器的HPC集群上使用singularity。网页中给出了一个交互式的使用singularity的示例，包括如何申请一个测试节点，如何加载singularity模块，如何拉取或构建一个singularity镜像，以及如何在镜像中执行命令或启动一个shell。

本手册内容参考的网址有：

- 官网：https://docs.sylabs.io/guides/3.7/user-guide/index.html
- Singularity日常使用指南：http://www.xtaohub.com/Container-Tech/Singularity-in-nutshell.html
- Singularity的基本用法：https://hpclib.com/soft/Singularity.html