FOSS Gbg

Singularity container platform
optimized for high performance
computing, enterprise and security

HPC (High Performance Computing)

- Multiple users per server
- Shared network home folder
- No privileged user access
- Strict security requirements
- Typically older system versions
- Very high needs for performance (CPU, I/O, etc)
- Access to specialized hardware (GPU, MPI, etc)

Docker

- **“This is a Linux system, I know this!”**
- **Let’s run Docker (find docker client, `docker run`)**

```
Cannot connect to the Docker daemon at
unix:///var/run/docker.sock. Is the docker daemon running?
```

- **Let’s start it then (`sudo systemctl start docker`)**

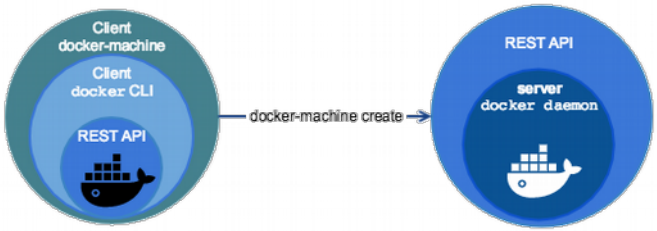
```
user1234 is not in the sudoers file.  This incident will be
reported.
```

- *“Dear Security Staff, I would like to be able to run a server that lets me run random programs from the Internet - as root, with network access. Thanks in advance /L. User“*

Virtual Machines

- **Docker Machine to the rescue**

```
$ docker-machine create default
$ eval $(docker-machine env)
```



- **But...**
  - Why is it so small ? And slow ?
  - And where are all my files now
  - How can I access my tools on NFS
  - What about those special devices
  - And I still can’t access Docker Hub

Rootless

- It is possible to run Linux containers without root
- If you have a new enough host kernel, that is (4.19)
- Requires some system support to map the uid/gid
- You might have some lower performance, though
  - Filesystem
  - Network
- And it would still run as root within the container
- My files, tools and devices are still “gone”and so on

Singularity

- **Singularity 3.5**
  - Linux container platform optimized for High Performance Computing (HPC) and Enterprise Performance Computing (EPC)

- **Version 1.0 in 2016, version 3.0 in 2018**
- **Written in the Go programming language**

- **Developed at the University of California, Lawrence Berkeley National Laboratory (LBNL)**
- **Licensed under the 3-Clause BSD License**

Singularity Image Format

- Single compressed .sif file
- Executable (`singularity run`)
- Immutable (r/o) by default
- Cryptographically signed
- Built automatically from docker image or hub
- Or build .sif file on demand (`singularity build`)
- Custom .def definition file, including bootstrap

Demo - Singularity

Possible to do some of this with Docker

- **If you really really like to use your old Big Fat Daemon**
- **Custom image entrypoint**
  - Mirror uid / gid
  - Create user
  - Volume mount
- **Best practices for image**
  - No /root or /home
  - No /run or /tmp
  - Read-only /

```
docker run --read-only --tmpfs /run --tmpfs /tmp
```

Runtime Options

- **Writable**

```
-w, --writable          by default all Singularity containers are
                        available as read only. This option makes
                        the file system accessible as read/write.
```

- **Contain**

```
-c, --contain           use minimal /dev and empty other
                        directories (e.g. /tmp and $HOME) instead
                        of sharing filesystems from your host
-C, --containall        contain not only file systems, but also
                        PID, IPC, and environment
```

- **Fakeroot**

```
-f, --fakeroot          run container in new user namespace as uid 0
```

Additional Projects

- **Singularity CRI (beta)**
  - Kubernetes Container Runtime Interface (CRI) support
- **Singularity Desktop (beta)**
  - Native support for macOS and Windows (coming soon)
- **SingularityPRO**
- **Sylabs.io Cloud**
  - Container Library
  - Remote Builder

Questions?

Anders F Björklund
github.com/afbjorklund

For more info on singularity:
https://sylabs.io/singularity/