

# BLUE WATERS

SUSTAINED PETASCALE COMPUTING

## Interacting with Shifter on Blue Waters



GREAT LAKES CONSORTIUM  
FOR PETASCALE COMPUTATION

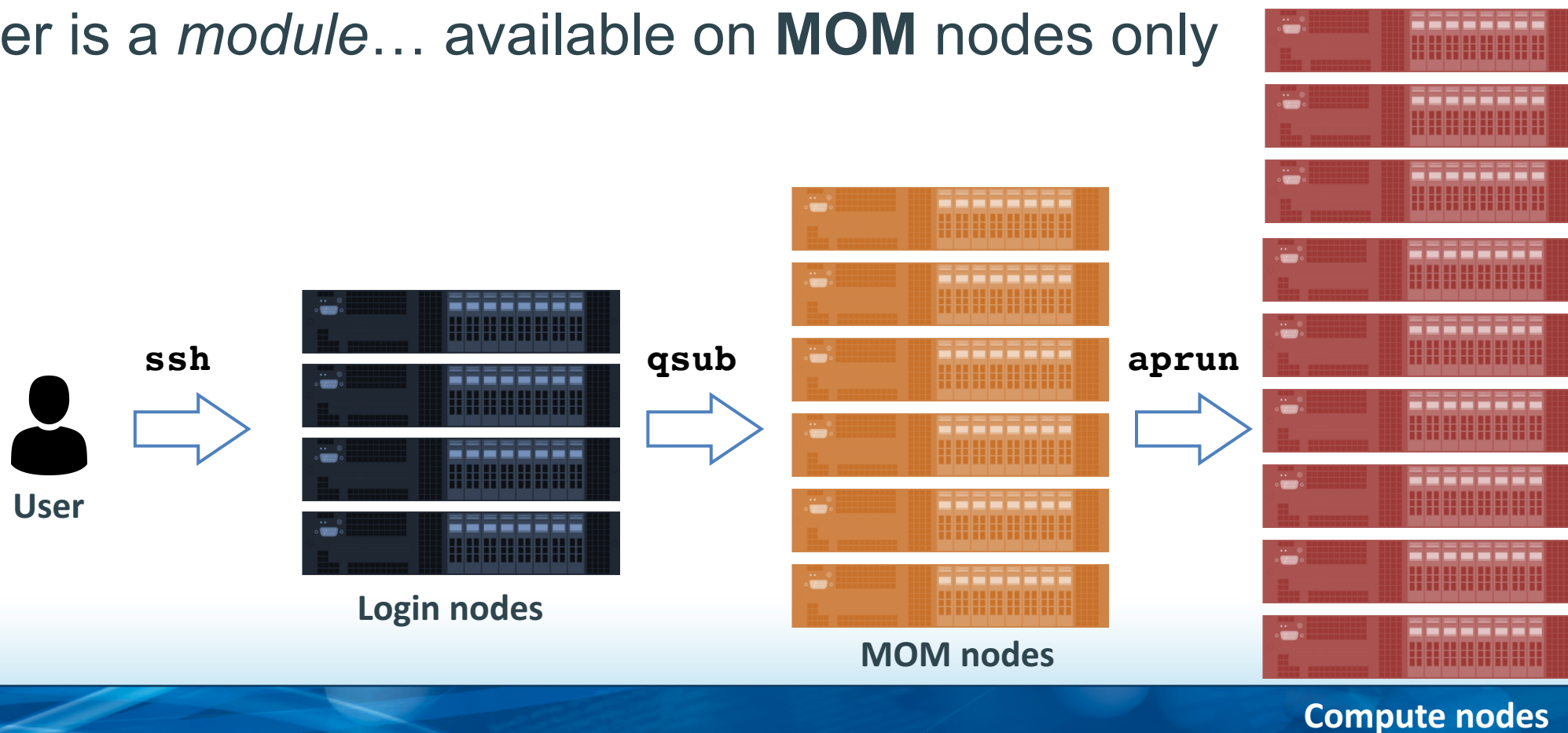
CRAY®



- Container solution for HPC systems
- Developed at NERSC: <https://github.com/NERSC/shifter>
- Works with Docker containers
- “Images” are optimized for faster redistribution across compute nodes
- Use custom software stacks
- “Integrated” into Blue Waters & implemented as a module
- Limited (compared to Docker) functionality

# User-Defined Images

- Shifter containers are called: **User-Defined Images**, or **UDI**
- Shifter is a *module*... available on **MOM** nodes only



# Getting a Docker image onto Blue Waters

```
$ qsub -I -l walltime=00:30:00 -l nodes=1:ppn=1 Login node
```

```
$ module load shifter MOM node
```

```
$ getDockerImage
```

```
usage: getDockerImage images
```

```
getDockerImage lookup <image>:<tag>
```

```
getDockerImage pull <image>:<tag>
```

Docker image name: <repo>/<imagename>:<tag>

## \$ `getDockerImage` `images`

Returns names, ID, sizes, as well as creation dates of *User-Defined Images* that are available on Blue Waters.

---

...

```
Image: centos:5 (...omitted...), Size: 271.48MB, Date: 2016/08/30 13:19:35
```

```
Image: centos:latest (...omitted...), Size: 182.95MB, Date: 2016/12/15 12:21:23
```

...

## \$ `getDockerImage lookup imagename:tag`

Returns ID of an image if it is available on the system or an error otherwise.

---

```
$ getDockerImage lookup centos:7
```

```
Retrieved docker image centos:7 resolving to ID:
```

```
9baab0af79c4fab5200255fe226cb147f95255028bd400761a8242da43688512
```

```
$ getDockerImage lookup ubuntu:zesty
```

```
ERR: Unable to find image 'ubuntu:zesty'
```

**\$ getDockerImage pull imagename:tag**

Download (update) Docker images *from Docker Hub (hub.docker.com) only.*

---

```
$ getDockerImage pull centos:latest
```

```
Retrieved docker image centos:latest resolving to
```

```
ID:d4350798c2ee9f080caff7559bf4d5a48a1862330e145fe7118ac721da74a445
```

```
$ getDockerImage pull ncsa/polyglot:latest
```

```
Downloading: c9d83ebb2deb [=====>]          32 B/32 B
```

```
...
```

```
Retrieved docker image ncsa/polyglot:latest resolving to
```

```
ID:eb21e6d7bc24d5e0f37afacdb7327378b9ac1435fd32fd29d8cd3a95078ebbf8
```

# Docker image not on *Docker Hub*?

Simply upload it to *Docker Hub*!

```
$ docker pull <repo>/<image>:<tag>
```

or

```
$ docker build -t <your-repo>/<image>:<tag> -f Dockerfile
```

```
$ docker login
```

```
$ docker push <your-repo>/<image>:<tag>
```



# Using a Shifter UDI on Blue Waters in a Batch Job

1. Special generic resource request:

```
$ qsub ... -l gres=shifter ...
```

2. Set *UDI* environment variable:

```
$ qsub ... -v UDI=<image>:<tag> ...
```

3. Set *CRAY\_ROOTFS* environment variable in a job:

```
export CRAY_ROOTFS=UDI
```

4. Use *aprun* with '-b' option:

```
aprun ... -b -- <app> <arguments>
```

# Using a Shifter UDI on Blue Waters in a Batch Job

```
$ qsub -I -l nodes=2:ppn=1 -l gres=shifter -v UDI=centos:latest  
...  
$ export CRAY_ROOTFS=UDI
```

# Using a Shifter UDI on Blue Waters in a Batch Job

```
$ /bin/bash --version
```

MOM node

```
GNU bash, version 3.2.51(1)-release (x86_64-suse-linux-gnu)
```

```
Copyright (C) 2007 Free Software Foundation, Inc.
```

```
$ aprun -n 1 -N 1 -b -- /bin/bash --version
```

```
GNU bash, version 4.2.46(1)-release (x86_64-redhat-linux-gnu)
```

Compute node

```
Copyright (C) 2011 Free Software Foundation, Inc.
```

```
License GPLv3+: GNU GPL version 3 or later http://gnu.org/licenses/gpl.html
```

```
This is free software; you are free to change and redistribute it.
```

```
There is NO WARRANTY, to the extent permitted by law.
```

```
Application 382134 resources: utime ~0s, stime ~1s, Rss ~4244, inblocks ~7, outblocks ~0
```

# Using a Shifter UDI on Blue Waters in a Batch Job

```
$ cat sample_job.pbs
#!/bin/bash
#PBS -l gres=shifter
#PBS -v UDI=centos:latest
#PBS -l nodes=1:ppn=32:xe
#PBS -l walltime=00:30:00
export CRAY_ROOTFS=UDI
cd $PBS_O_WORKDIR
aprun -n 16 -N 1 -b -- <app1> <args1> ... &
aprun -n 16 -N 1 -b -- <app2> <args2> ... &
wait
```



/projects and /scratch folders are mapped from *UDI* to *Blue Waters*

# SSH to compute nodes running Shifter

## 1. Generate SSH key pair

```
$ mkdir -p ~/.shifter
```

```
$ ssh-keygen -t rsa -f ~/.shifter/id_rsa -N ''
```

## 2. Figure out assigned compute nodes

```
$ aprun -n $PBS_NUM_NODES -N 1 -b -- hostname
```

## 3. SSH to a compute node!

```
$ ssh -p 204 -i ~/.shifter/id_rsa -o UserKnownHostsFile=/dev/null \  
-o StrictHostKeyChecking=no -o LogLevel=error nidXXXXX
```

# SSH to compute nodes running Shifter

```
$ cat << EOF > ~/.shifter/config
Host * Port 204
IdentityFile ~/.shifter/id_rsa
StrictHostKeyChecking no
UserKnownHostsFile /dev/null
LogLevel error
EOF
$ ssh -F ~/.shifter/config nidXXXXX
```