Los Alamos NATIONAL LABORATORY

OpenCHAMI on OpenHPC Expanding Ease of Entry

HPC-DES

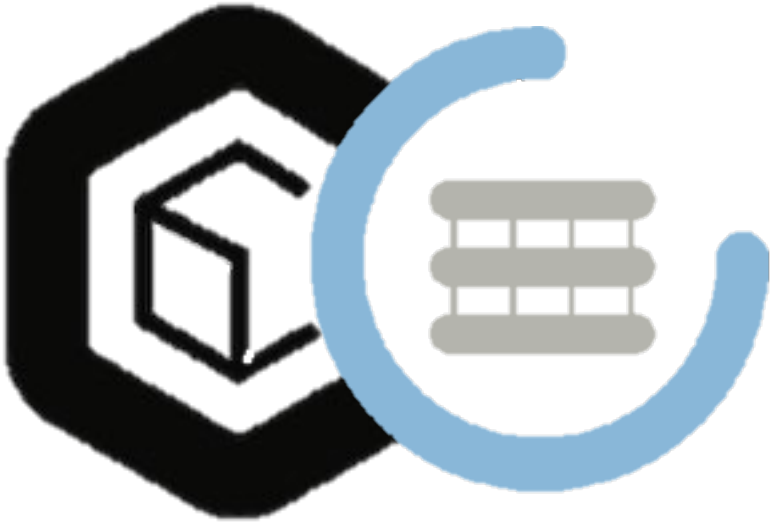
Travis Powell

ROSY ID: #bf84a981

LA-UR-25-28258

Presentation Topics

1. Problem
2. Introduction to OpenHPC
3. How It Works
4. Benefits



Problem

- Manual Setup is Complex
- Small HPC centers lack the automation tools
- Reproducible, and Scalable
- Sites Use OpenHPC packages, but not provisioners

Manual:

- Time Consuming
- Documentation
- Manual Steps
  - OS Install
  - Config
  - Networking
- Updates

Automated:

- Builds once in pipeline
- Easily repeatable
- Scales Effortlessly
- Deploys in Minutes
- Simplifies maintenance and updates

Introduction to OpenHPC

- Open Source
  - Backed by several organizations
- Pre-built
- Targets Smaller HPC centers

How It Works

OpenHPC uses LaTeX files with command blocks to generate a bash executable file

```
% begin_ohpc_run
% ohpc_validation_newline
% ohpc_comment_header Enable OpenCHAMI Repositories and Support \ref{sec:enable_openchami}
\begin{lstlisting}[language=bash,keywords={},basicstyle=\fontencoding{T1}\fontsize{8.0}{10}\ttfamily,
    literate={VER}{\OHPCVerTree{}}1 {OSREPO}{\OSTree{}}1 {TAG}{\OSTag{}}1 {ARCH}{\arch{}}1 {-}{-}1]i
[sms](*\#*) (*\install*) podman buildah ansible-core nfs-utils tcpdump tftp curl
[sms](*\#*) rpm --import https://github.com/OpenCHAMI/release/releases/download/vVERLONG/public_gpg_key.asc
[sms](*\#*) (*\install*) https://github.com/OpenCHAMI/release/releases/download/vVERLONG/openchami-VERLONG.rpm
\end{lstlisting}
% ohpc_validation_newline
% end_ohpc_run
```

```
# ----------------------------------------
# Enable OpenCHAMI Repositories and Support
# ----------------------------------------
dnf -y install podman buildah ansible-core nfs-utils tcpdump tftp curl
rpm --import https://github.com/OpenCHAMI/release/releases/download/v0.0.30/public_gpg_key.asc
dnf -y install https://github.com/OpenCHAMI/release/releases/download/v0.0.30/openchami-0.0.30.rpm
```

How It Works

```
# ------------------------
# compute node settings
# ------------------------
# total number of computes
num_computes=1
compute_regex="c*"
compute_prefix="c"
# compute hostnames
c_name[0]='c1'
# compute node IP addresses
c_ip[0]='172.16.1.1'
# compute node MAC addreses for provisioning interface
c_mac[0]='b4:2e:99:a6:07:21'
# compute node BMC addresses
c_bmc[0]='10.16.1.1'
```

How It Works

```
# --------------------------
# SMS (master) node settings
# --------------------------
# Hostname for master server (SMS)
sms_name="vt.openchami.cluster"
# Local (internal) IP address on SMS
sms_ip="10.15.3.44"
# Internal ethernet interface on SMS
sms_eth_internal="${sms_eth_internal:-eth1}"
# Subnet netmask for internal cluster network
internal_netmask="${internal_netmask:-255.255.0.0}”
```

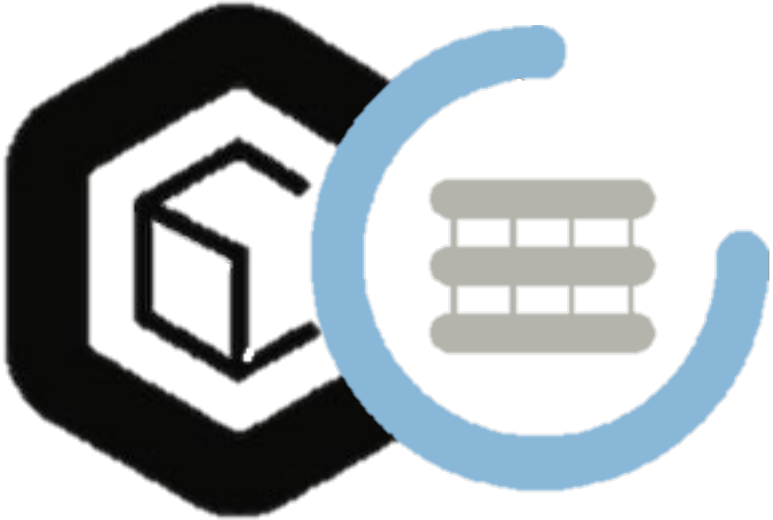
How It Works

Then you just

sh <ohpc_dir>/recipe.sh

Benefits

- Lowers Barrier for Adoption for Small HPC Centers
- Reusable
- Improves Reproducibility
- DevOps/CloudOps Practices



Thank You

Mentors

- Travis Cotton
- Alex Lovell-Troy

Special Thank You

- Julie Wiens

Others

- Adrian Reber (OHPC)
- Chris Simmons (OHPC)

Questions?