OAK RIDGE National Laboratory

Status of Containers at OLCF

Ryan Prout

HPC Engineer – User Assistance

ORNL is managed by UT-Battelle LLC for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

Quick Agenda

- Container Technologies
- Where we are (high-level) with …..
  - building containers
  - running containers
- Example workflow overview

Container Technologies Being Tested in OLCF

- Default/stock container engine shipped with RHEL8 (Docker replacement)
- Supports OCI containers
  - Standards around image formats and runtimes
- Not originally designed for HPC
- Frontend for crun/runc container runtimes

- Frist developed at LBNL
- OCI *compliant* (full compliance in the past couple years)
- Designed for HPC
- "built-in" runtime

Container Builds and Runs

- **Base goals**
  - Enable users to build locally, on Summit (rootless)
  - Enable native MPI in container (Spectrum MPI)
    - Run at scale
  - Enable easy access to GPU from container
  - Keep performance
  - Advance software flexibility, sharing, and reproducibility at OLCF

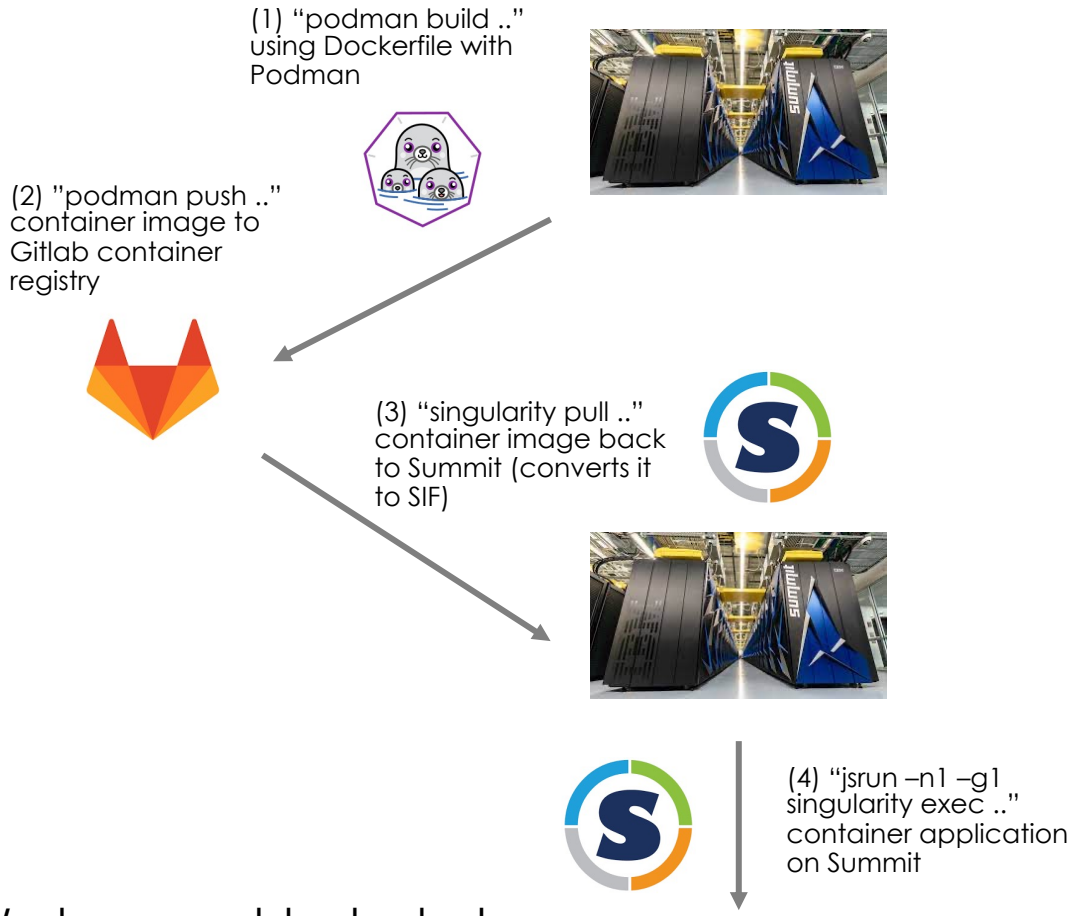
- **What we have tested**
  - MPI application builds with Podman and Singularity
    - Podman builds work on Summit **[1]**
    - Singularity builds do not work on Summit
      - Difference is due to how rootless is handled
  - GPU application builds with Podman and Singularity
    - Podman builds *appear* to work for Nvidia GPUs
      - Issues with running through **[2]**
      - Still no on-system builds with Singularity
    - Singularity has no issues running with GPUs (bring pre-built Power container)
  - MPI Runtime performance was initially degraded with Podman. We have success improving this with some tweaking
  - Singularity runtime performance is not degraded

Issue References

- **[1]**Podman: https://github.com/containers/podman/issues/8580
- **[2]**Nvidia: https://github.com/NVIDIA/nvidia-container-runtime/issues/85 - issuecomment-811080338

Example Workflow (build with Podman and Convert to Singularity for runtime)

1) Build container image on login node **with Podman**
2) Push container image to container registry **with Podman** (can create registry from Gitlab repository)
3) Pull image from registry **using Singularity** (*necessary to convert container image to Singularity image format (SIF) – to be used by Singularity runtime*)
4) Run container job using **Singularity runtime**



NOTE: We have not tested at scale yet.

Recap of what looks promising and updates

- Builds with Podman - then use the Singularity runtime
  - Podman's underlying runtimes could progress though (perhaps allowing us to fully switch to Podman). Current issues seem to mainly revolve around GPU interoperability.

- Summit will have Podman and Singularity after next upgrade (*no timeframe for upgrade yet*)
  - We want to work with users on use cases and making progress. This will open-up with the addition of Podman to Summit.
  - The example of building with Podman and running with Singularity will be available.

Shout Out ☺

- **Subil Abraham** – HPC Engineer - User Assistance Group
- **Matt Davis** – HPC Engineer - System Acceptance and User Environment Group
- **Don Maxwell** – HPC Systems Engineer – Scalable Systems
- **Joseph Voss** – HPC Systems Engineer – Scalable Systems

Jupyter Table-Top Breakout Session Today!

12-1PM

Bring your ideas and questions ☺

Matt Davis and Subil Abraham will also be present at Table-Top session