Containers and Slurm

Nathan Rini
SchedMD

NHR Container Workshop 2021

Asking questions

- Feel free to ask questions throughout through presentation via chat.
- I will try to answer during the presentation but may need to defer some questions to the end.

Please make sure to read our Container documentation:
https://slurm.schedmd.com/containers.html

Containers in Slurm (before 21.08 release)

- User calling container runtimes directly in jobs
  - Users will directly manage their container images and will explicitly call the container runtime inside of the job.
  - Sites have complete freedom to decide what containers are used, when, where and how.
  - Users (and site admins) have to manage their workflows and containers and all the details.
- Containers via SPANK plugins
  - Site installs SPANK plugin that will augment Slurm's commands to accept new arguments to make it easier for users to request.
  - Limitations in Slurm will result in limited effectiveness of SPANK plugins.
- slurmd in a container
  - Running slurmd inside of a privileged container to control mount namespaces (and possibly others namespaces).
  - Very limited in scope.

OCI Container Support (21.08 release)

Technical Preview

This functionality has been added as a technical preview as we continue to work out all the features and site requirements. **RFE tickets are always welcome.**

Functionality and interfaces may change dramatically between releases.

- Container is a little ambiguous of a term. This is specifically Open Container Initiative (OCI) containers which follow the set of standards here:
  - https://github.com/opencontainers
- Slurm container documentation:
  - https://slurm.schedmd.com/containers.html
- Note: job_container/tmpfs is independent from OCI Container functionality

OCI Container Support (21.08)

- Slurm now supports (limited) executing of OCI Containers via OCI runtimes
  - Relevant standards: OCI Runtime & OCI Image
  - OCI containers were originally developed by Docker but are now used in a few places including Kubernetes.
    - Docker appears to update the OCI standard on major releases and they are not always compatible changes
  - All OCI containers are started/controlled via an OCI runtime executable
    - There are several OCI runtimes, each of varying level of compliance with the standard.
  - Existing containers:
    - Limited OCI container support already exists for Singularity and charlie-cloud.
    - Sarus already has full OCI container support due to their use of runc.

OCI Container Support (21.08)

- Added ‘--container’ support to the following:
  - srun
  - salloc
  - sbatch
- Added viewing job container to the following:
  - scontrol show jobs
  - scontrol show steps
  - sacct
    - If passed as part of the ‘--format” argument using “Container”
  - slurmdbd & slurmctld logs (too many places to list)

OCI Container Support (21.08)

- Slurm cgroups features apply to the OCI containers
  - All processes should be cleaned up even if the container anchor process dies or processes attempt to become daemons and detach from the session.
  - Resource usage can be hard limited and monitored
- Slurm is only going to support unprivileged containers in 21.08
  - Use existing kernel support for containers
  - Users can already call all of these commands directly
  - Containers must be able to function in an existing host network
- New ‘oci.conf’ in /etc/slurm/
  - If ‘oci.conf’ is not populated, the ‘--container’ request will only be recorded.
    - Environment variable SLURM_CONTAINER will always be set with value (if present).

OCI Container Support (21.08)

srun & salloc examples

```
$ srun --container=/tmp/centos grep ^NAME /etc/os-release
NAME="CentOS Linux"
$ salloc --container=/tmp/centos
salloc: Granted job allocation 24418
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
```

Note: containers have limited permissions and can result in some warnings

OCI Container Support (21.08)

sbatch example

```
$ sbatch --container=/tmp/centos --wrap 'grep ^NAME /etc/os-release'
Submitted batch job 24419
$ cat slurm-24419.out
NAME="CentOS Linux"
```

OCI Containers Images (21.08)

- User **must** prepare their OCI images per specifications:
  - https://github.com/opencontainers/image-spec
- Container images **must** already be visible on executing compute node.
  - Slurm does not copy or mount any images directly from job submission node.
- OCI image structure (folder structure)
  - Slurm currently only cares about the ‘config.json’ file in the root directory.
  - config.json provides all the required information for Slurm to request a container instance.
  - Slurm will produce a per step copy (in spool dir) of the config.json and provide it to the OCI runtime to allow the editing of the requests. This is required to be able to run different arguments inside of the container.
  - The OCI runtime handles all the details of mount types including overlays

Future plans?

- We are requesting input from sites about Container support.

- Slurm is a scheduler, not a Container runtime:
  - Goal is to ensure as many container solutions work with Slurm as possible.
  - OCI standard was chosen due to its popularity and general ease of converting existing images to the format.
- Lua burst buffer plugin can be used to stage container images.
  - Currently, no integration with Container requests. Looking for input on how sites want to handle container image movement. On one or two nodes, it may not be an issue but when clusters have 5000 nodes then image locality starts to matter a lot.

Questions?

Please feel free to ask any questions. I will cover the prepared questions first.

Question:

- Will there be a support of non OCI, rootless runtimes such as Singularity in native containers support of Slurm (not plugin) or should we wait for rootless implementation of Singularity/Apptainer?
  - The goal is to move to OCI containers. Singularity/Apptainer has OCI Container support (at time writing this response). We provide an example oci.conf settings to use Singularity in the Containers guide [https://slurm.schedmd.com/containers.html].
    - OCIRunTimeQuery="sudo singularity oci state %n.%u.%j.%s.%t"
    - OCIRunTimeCreate="sudo singularity oci create --bundle %b %n.%u.%j.%s.%t"
    - OCIRunTimeStart="sudo singularity oci start %n.%u.%j.%s.%t"
    - OCIRunTimeKill="sudo singularity oci kill %n.%u.%j.%s.%t"
    - OCIRunTimeDelete="sudo singularity oci delete %n.%u.%j.%s.%t
  - Please note that the container’s implementation is very generic and a site could instead configure the commands to work against an non-OCI container but it is not advised due to the many gotchas related to this strategy.

Question:

- What are the benefits of running OCI Containers with built in native support in comparison to running a job and invoking the Container runtime within the jobscript?
  - The provided ‘--container=’ argument is Slurm’s first step into properly supporting containers. The provided container value will be present in all of the Slurm commands such as sacct and squeue unlike embedding the call to the containers in the job.
  - The goal is the existing container solutions will work with (or can be modified) to work with the OCI container design. If the oci.conf is not populated, then specifying ‘--container’ can be used as an input to an existing container solution via the environment variable SLURM_CONTAINER.
  - Most container systems already have support for OCI containers. Most non-OCI containers images can be converted to OCI container images.

Question:

- Which container runtimes are recommended and are there any known limitations/issues with them?
  - We provided documented examples for the following:
    - runc
    - crun
    - nvidia-container-runtime
    - hpcng singularity v3.8.0
  - The container support is designed to be generic and many other container systems should be possible to use.
- The current container support in Slurm is explicitly unprivileged and run as the Job's user. While it is possible to use setuid or sudo, we consider this ill-advised for shared HPC systems.

Organizers Questions:

- Would it be possible to spawn a container on the cloud machines with Slurm to provide additional resources for the HPC cluster on-demand?
  - Yes, container support works the same on Cloud nodes as it does on-premises nodes.
    - To make it work, just ensure the following:
      - oci.conf must configured on the cloud nodes.
        - The chosen OCI runtime works outside of Slurm on the Cloud nodes.
      - OCI image is on storage that is visible to the OCI runtime.
      - Cloud nodes will have slurmd running on them as root.
        - Cloud nodes don't have to have Slurm manage them via power controls but it is suggested.
    - A job with a container is scheduled the same as any other job. A site will need to ensure Slurm is configured for cloud bursting which is independent of container support.

End Of Presentation

- Thanks for watching!
- Please post additional questions or enhancement requests on Slurm's bugzilla:
  - https://bugs.schedmd.com/