Securely Using PBS with Docker

March 2018

Introduction

We describe here an early version of a feature that will be available in PBS 18.2.

Job submitters can run their jobs in Docker containers by setting the CONTAINER_IMAGE environment variable to the name of the container in which the job should run. PBS uses a hook to monitor job submissions for this environment variable, launches the appropriate container, and starts the job in the container. When the job finishes, PBS removes the Docker container. PBS starts a separate container for each job. Jobs are matched to hosts running Docker daemons via a custom resource named "allows_container", which when True indicates that a host is running a Docker daemon.

We see a security shortcoming in Docker in the case where multiple users are on the same host, where users can see into each others' containers. We have implemented a security enhancement for this. We allow the job to run inside the container, but we don't add job submitters to the Docker group, and we don't allow job submitters to connect to the Docker container.

Users can run multi-vnode, multi-host, and interactive jobs in Docker containers, and PBS tracks resource usage for these jobs.

The PBS Docker Hook

PBS has a hook named "pbs_hpc_container" which has the following events and actions:

queuejob (job submission)

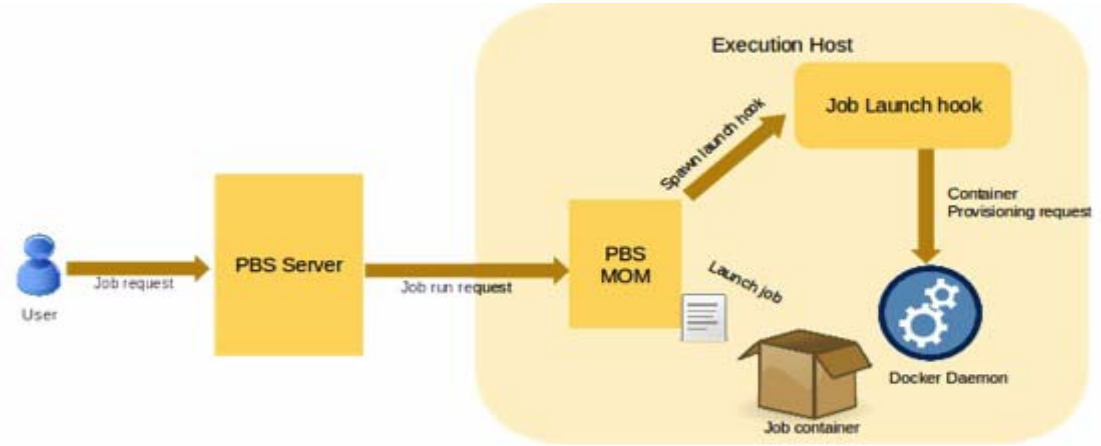
The `pbs_hpc_container` hook adds the allows_container resource to the job's select statement. This allows the scheduler to match the job to a host running a Docker daemon.

execjob_launch (job start)

The hook starts a container instance from the requested image, and sets up the job's environment. The image is specified in the job's CONTAINER_IMAGE environment variable.

- The name of the container is the job ID.
- If the job is interactive, the hook runs the job in the container in interactive mode.
- If the job has multiple chunks, the hook runs all of the job's child processes in the container.
- If the job runs on multiple hosts, the hook ensures that containers created on sister MoMs are network linked to the container running on the Mother Superior.

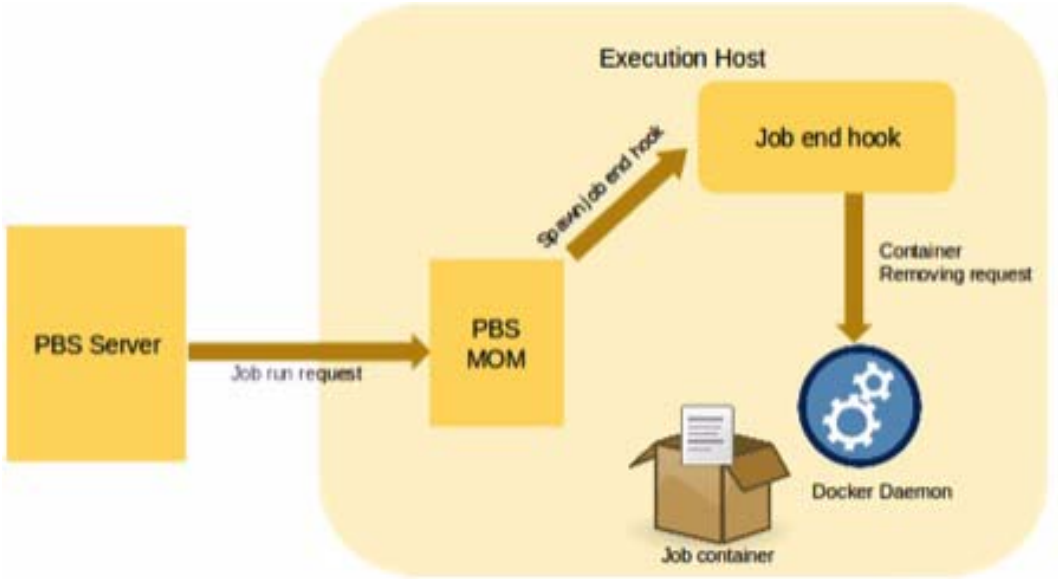
Figure 1-1:Launching a container



execjob_end (job termination)

The hook updates the resources used by the job, and removes the job's container.

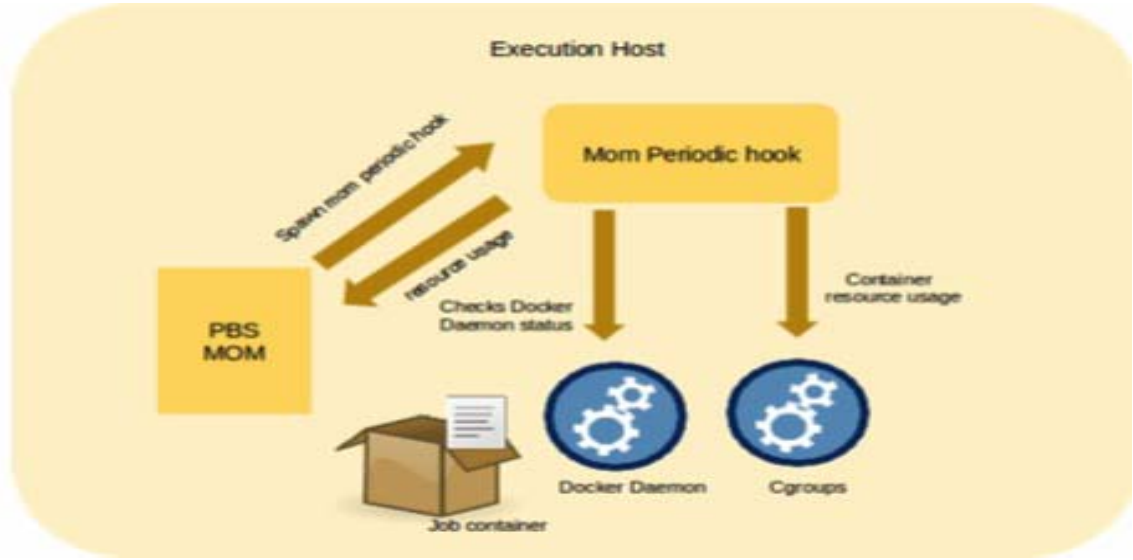
Figure 1-2: Removing a container



exechost_periodic (periodically on execution host)

- While the job runs, the hook periodically polls the job's cgroup and updates the job's resource usage.
- The hook periodically checks the status of the Docker daemon and updates the value of the allows_container resource on the host.
- The hook cleans up any orphaned containers left behind by previous jobs on the host.

Figure 1-3: Cleaning up container



Prerequisites

- The Docker daemon must be running on all hosts where users will run jobs in Docker containers.
- The container in which a job will run must exist before the user submits the job to PBS.

Configuring PBS for Docker

Use Boolean For Resource Matching

1. Create a custom Boolean resource named “allows_container”:
   Qmgr: create resource allows_container type=boolean
2. Notify scheduler of hosts where a Docker daemon is running.

   Create list of hostnames where Docker daemon is running, and make them active:

   Qmgr: active node <hostname>,<hostname>, ...

   Set the Boolean named “allows_container” to *True*:

   Qmgr: set node resources_available.allows_container=true

   Or, set the value of the resource individually for each hostname:

   Qmgr: set node <hostname> resources_available.allows_container=true
   Qmgr: set node <hostname> resources_available.allows_container=true
   ...

Configure PBS Docker Hook

Set global parameters in the PBS Docker hook configuration file to match your site. The configuration file must conform to JSON syntax. The file is named “container_config.json”. These are the parameters:

Table 1-1: PBS Docker Hook Configuration File Parameters

Parameter Name	Default Value	Description
docker_cmd	/usr/bin/docker	Path to docker
nvidia_docker_cmd	/bin/nvidia-docker	Path to nvidia-docker
remove_env_keys	[]	List of environment variables not to export to job container
mount_paths	[]	Additional paths to mount into container at creation

We show a sample configuration file here:

```
{
    "docker_cmd": "/usr/bin/docker",
    "nvidia_docker_cmd": "/bin/nvidia-docker",
    "remove_env_keys": ["LS_COLORS", "MY_JUNK_VAR"],
    "mount_paths": ["/home/Mpi"]
}
```

Create and Configure PBS Docker Hook

The following is an example of creating and configuring a PBS Docker hook:

```
Qmgr: create hook pbs_hpc_container
Qmgr: set hook pbs_hpc_container type = site
Qmgr: set hook pbs_hpc_container enabled = true
Qmgr: set hook pbs_hpc_container event = execjob_end
Qmgr: set hook pbs_hpc_container event += execjob_launch
Qmgr: set hook pbs_hpc_container event += exechost_periodic
Qmgr: set hook pbs_hpc_container event += queuejob
Qmgr: set hook pbs_hpc_container alarm = 50
Qmgr: set hook pbs_hpc_container order = 1
Qmgr: set hook pbs_hpc_container debug = true
Qmgr: set hook pbs_hpc_container freq = 30
Qmgr: import hook pbs_hpc_container application/x-config default container_config.json
Qmgr: import hook pbs_hpc_container application/x-python default pbs_hpc_container.py
```

Create Docker Groups

Create a Docker group on each host where a Docker daemon will run. No user should be part of these groups.

Configure Security Enhancement

Our security enhancement for Docker integration allows jobs to run inside the container, but prevents job submitters from connecting to the Docker container. We use `pbs_container` to accomplish this. Here are the steps:

For this early version, download `libpbs.h` from `https://github.com/PBSPro/pbspro/blob/master/src/include/libpbs.h`.

Compile the `pbs_container` C code:

```
gcc pbs_container.c -o pbs_container -I /pbspro/src/include/ -L/opt/pbs/lib -lpbs -lcrypto -
    lpthread
```

Put the compiled code in `PBS_EXEC/sbin`:

```
mv pbs_container PBS_EXEC/sbin/pbs_container
```

Make `PBS_EXEC/sbin/pbs_container` a part of the Docker group. Set its SGID permissions:

```
chgrp docker PBS_EXEC/sbin/pbs_container
chmod 2755 PBS_EXEC/sbin/pbs_container
```

Submitting Jobs Using Containers

Job submitters specify a container in which to run a job by setting the CONTAINER_IMAGE environment variable to the name of the image and passing the environment variable with the job:

```
qsub ... -v CONTAINER_IMAGE=<name of container image> ...
```

For a multi-host job, you can use any version of OpenMPI with this feature.