

GxITs with a cluster

Galaxy Interactive Tools Hackathon - 7/10 décembre 2020

Why a cluster

Default configuration runs Docker containers on the Galaxy server

Limited resources (cpu, ram)

Can slow down the whole server => slow/unresponsive Galaxy

Solution:

Galaxy launches the container ***inside*** a normal cluster job on a compute node

Feedback on configuring it with a Slurm cluster (GenOquest)

Job destination (job_conf.xml)

```
<plugins>
  <plugin id="slurm" type="runner" load="galaxy.jobs.runners.slurm:SlurmJobRunner"/>
</plugins>

<destinations>
  <destination id="docker_slurm" runner="slurm">
    <param id="docker_enabled">true</param>
    <param id="docker_volumes">
      $tool_directory:ro,$job_directory:rw,$working_directory:rw,$default_file_path:ro</param>
    <param id="docker_sudo">false</param>
    <param id="docker_net">bridge</param>
    <param id="docker_auto_rm">true</param>
    <param id="require_container">true</param>
    <param id="docker_set_user"></param>
    <param id="nativeSpecification">-p galaxy</param>
    <param id="docker_run_extra_arguments">--memory 6g</param>
  </destination>
</destinations>

<tools>
  <tool id="interactive_tool_ethercalc" destination="docker_slurm"/>
</tools>
```

Slurm configuration

Install Docker on compute nodes

Allow (only the) “galaxy” user to use docker (add to “docker” group”)

Specific partition with time limit

Docker containers will “escape” Slurm cgroups:

- need to ensure container is ***always*** killed at the end of the job
- need to restrict cpu/mem usage at the Docker level

Ensure container is always killed: epilog script

```
$ cat /etc/slurm/epilog.d/galaxy_docker_cleanup.sh
#!/bin/bash
```

```
if [ "$USER" != "galaxy" ]; then
    exit 0
fi
```

```
[ -n "$SLURM_JOB_ID" ] || { echo "No job id!"; exit 1; }
```

```
workdir=$(scontrol show job "$SLURM_JOB_ID" | sed 's/.*\(\ \|^\)\StdOut=\([^ ]*\)\|/stdout\(\ \|$\).*/\2/p;d')
container_config="$workdir}/configs/container_config.json"
```

```
if [ -d "$workdir" -a -f "$container_config" ]; then
    container_id=$(jq -r '.container_name' "$container_config")
    docker rm -f "$container_id"
fi
```

```
exit 0
```

```
$ grep Epilog /etc/slurm/slurm.conf
TaskEpilog=/etc/slurm/epilog.d/galaxy_docker_cleanup.sh
```

Restrict cpu/mem usage at the docker level

Might not be needed in future versions

```
<destinations>
  <destination id="docker_slurm" runner="slurm">
    (...)
    <param id="docker_run_extra_arguments">--memory 6g</param>
    (...)
  </destination>
</destinations>
```

Disk space

Docker images can be huge: climate jupyter image is 21Gb!

Give enough disk space to `/var/lib/docker/`

Privacy problem

```
<param id="docker_volumes">  
    $tool_directory:ro,$job_directory:rw,$working_directory:rw,$default_file_path:ro  
</param>
```

All Galaxy data mounted (read only) in the Docker container

Needed to access selected datasets in the web form

Ok for some GxIT (RShiny), not for others (Jupyter)

Two solutions:

- Embedded Pulsar mode
- Trusted/Untrusted destinations

Embedded Pulsar

As described in tutorial: good solution to data privacy problem

Only selected datasets are accessible

```
$ cat config/pulsar_app.yml
# The path where per-job directories will be created
staging_directory: "/galaxy/database/jobs_directory/_interactive"

# Where Pulsar state information will be stored (e.g. currently active jobs)
persistence_directory: "/galaxy/database/pulsar"

# Where to find Galaxy tool dependencies
tool_dependency_dir: "/galaxy/dependencies"

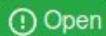
# How to run jobs (see https://pulsar.readthedocs.io/en/latest/job_managers.html)
managers:
  _default_:
    type: queued_drmaa
    native_specification: "-p galaxy -c 2 --mem=6000"
```

Embedded Pulsar

But... problem while using embedded pulsar mode:

<https://github.com/galaxyproject/galaxy/issues/10093>

PulsarEmbeddedJobRunner not passing tool's environment_variable to interactive tools #10093



Open

abretau opened this issue on 12 Aug · 1 comment

Trusted/Untrusted destinations

Workaround: only mount whole data dir to **trusted** GxIT

Downside: can't access selected datasets

```
<destinations>
  <destination id="docker_slurm_trusted" runner="slurm">
    (...)
    <param id="docker_volumes">
      $tool_directory:ro,$job_directory:rw,$working_directory:rw,$default_file_path:ro</param>
    (...)
  </destination>

  <destination id="docker_slurm_untrusted" runner="slurm">
    (...)
    <param id="docker_volumes">
      $tool_directory:ro,$job_directory:rw,$working_directory:rw</param>
    (...)
  </destination>
</destinations>
```

Other (untested) things

Singularity: might be possible to run some GxIT with it instead of Docker

Pulsar network to distribute GxIT jobs in other countries

