Running Web Applications on HPC using containers

NZ RSE Conference
Virtual 10/09/2020

Marko Laban (NeSI)
marko.laban@nesi.org.nz

Chris Scott (NeSI)
chris.scott@nesi.org.nz

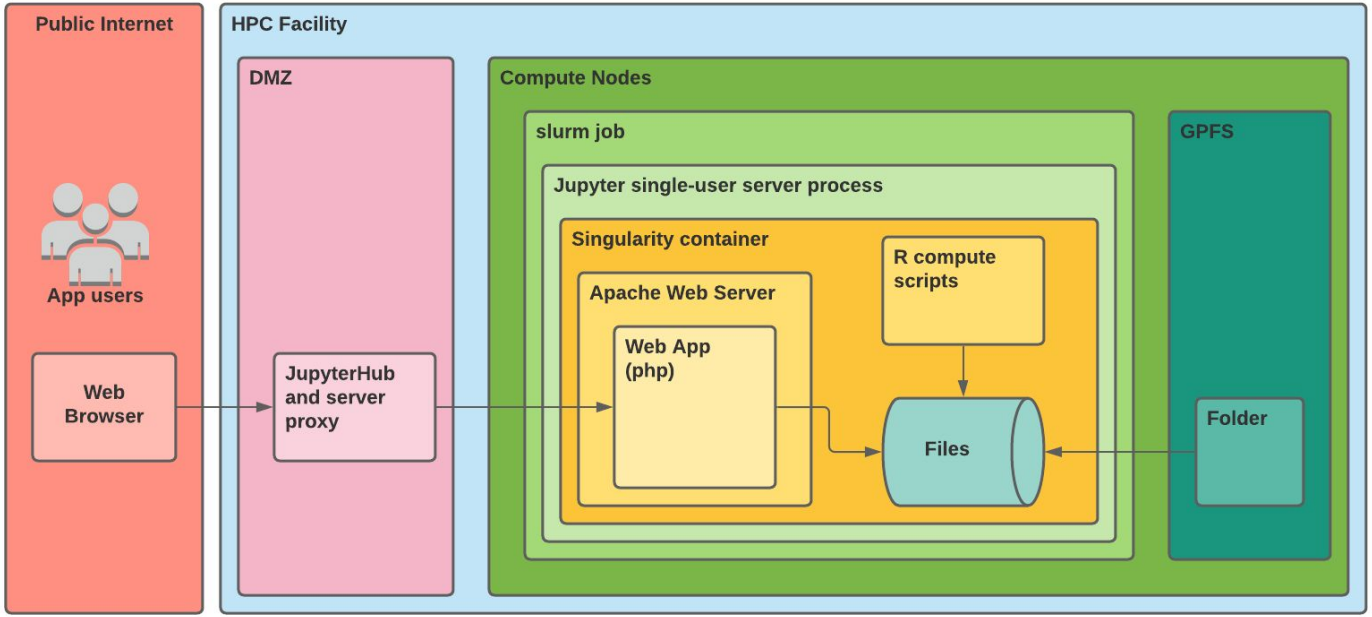
What problem are we trying to solve?

- Hosting public facing web apps in a modern cloud environment is ubiquitous, not so common in an HPC environment
- Easy to do in the cloud because:
  - Infrastructure designed to support this concept at scale from day 1
  - Huge community of practice, ecosystem of tools, DevOps
  - Standard service fully supported by commercial operators
- Still hard to do for a Researcher on HPC because:
  - No standard HPC specific tooling/workflows/support to do this job so usually need to roll-your-own
  - HPC system architecture is rarely designed to specifically cater for this scenario
  - Security, availability, maintainability, complexity challenges

Who needs this?

- Q: Why bother at all to expose web apps on HPC when it is much easier and almost free to do so in the cloud?
- A: Web apps provide an easy way to submit jobs and access job results - this extends the audience for an experiment/analysis tool that requires a lot of CPU/Memory/GPU or has specific data sovereignty needs. Doing this in the commercial cloud would be too costly or geographically infeasible.

What is our solution?



Note: Our aim was to use the set of available tools to deliver the solution in a timely manner, re-designing the infrastructure for this purpose was out of scope

Concrete example and story - running a web app on the HPC using containers

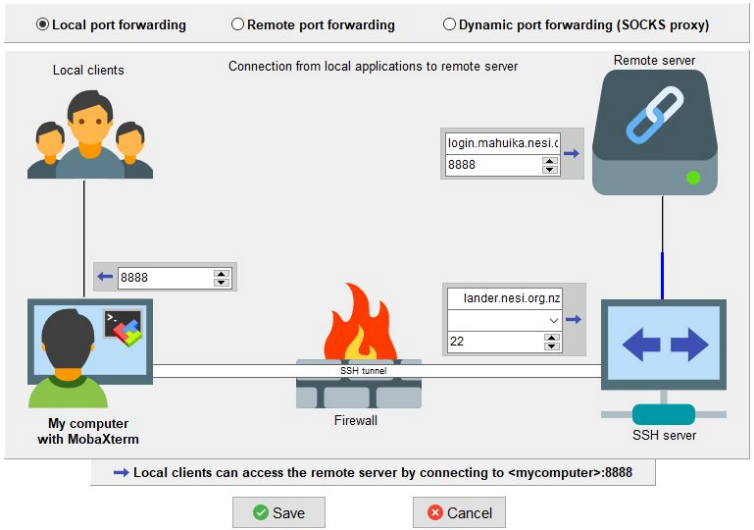
- Working with Basharat Bhat (University of Otago and Genomics Aotearoa)
  - He is developing a web app for running an Epigenome Wide Association Study (EWAS) pipeline
  - Requires lots of memory to run (>100GB)
- Docker and singularity images for portable web app: https://github.com/GenomicsAotearoa/EWAS_DAP
- Looked into the options for running the web app on NeSI

This work was done through NeSI's Consultancy Service: https://www.nesi.org.nz/services/consultancy

Contact us if interested to find out how we can support your project.

Concrete example and story

- Initial solution (SSH based)
  - Researcher SSH's into the cluster, edits Slurm script, chooses a port and submits job
  - Then sets up SSH tunnel on chosen port from their machine to cluster



Concrete example and story

- Better solution (Jupyter Server Proxy based)
  - Connect to Jupyter - https://jupyter.nesi.org.nz
  - Request required resources (memory, wall time, …)
  - From the JupyterLab launcher, click the “EWAS” button to launch the web app (runs within the resources you requested above)
  - Completely browser based, no SSH or terminal required

Demo

- Use Jupyter Server Proxy to run the web app via NeSI's JupyterLab service
  - Simple to configure
  - Requires a script to run web app
  - Accept options to specify port and base_url

jupyter_notebook_config.py

```
1 c.ServerProxy.servers = {
2   'EWASP': {
3     'command': ['/path/to/run_ewas.sh', '{port}', '{base_url}'],
4     'timeout': 10,
5   },
6 }
```

Demo - https://jupyter.nesi.org.nz

Solution specifics

- Containers
  - Docker image bundling the php web app, web server, compute R scripts and dependent libraries
  - Singularity image built from the Docker image
  - Data mapped from HPC file system via container volumes
  - Various hacks in the application to enable serving through tunnel/proxy (i.e. relative paths)
- Jupyter Proxy - https://github.com/jupyterhub/jupyter-server-proxy
  - Allows the user to proxy to a server running somewhere on the cluster (can be within the same node/allocation or elsewhere)
  - Can also start/manage the process if required (shows up as a button in the launcher next to the kernels)
  - Workaround for not being able to expose services running on the HPC to the outside world

What were the challenges?

- Can’t use docker on the HPC
  - Still wanted a docker version for portability
  - Convert docker to singularity (workarounds for non-root)
- Can’t easily expose to the outside world
  - SSH is the usual way to access the HPC
  - Requires multi-hop SSH tunnelling to expose web app
- HPC has a scheduler / queueing system
  - Not ideal for interactive use, may have to wait for resources
- Complex to deploy/use
  - Can’t build on the HPC - need to build locally or in the cloud
  - Many R packages need to be built from source - time consuming build process
  - Web app relative paths don’t work out of the box when running through a proxy/tunnel

Q&A

Questions?

support@nesi.org.nz

https://www.nesi.org.nz/apply

www.nesi.org.nz

@NeSI_NZ