The Use of Container in ASKAPsoft Deployment

Simplifying deployment for end-users and software build process for developers

Paulus Lahur, Mike Birchall, Eric Bastholm (Space & Astronomy, CSIRO)

ASKAPsoft is a specialized software suite developed by CSIRO to process a massive amount of radio astronomy data collected by the **Australian Square Kilometre Array Pathfinder (ASKAP)** telescope. ASKAP supports many science projects, due to its massive field of view and survey speed. ASKAPsoft is developed in-house, written in C++ and utilising Message Passing Interface (MPI). It's currently deployed in Pawsey's HPC facility called Setonix. Container technology is used for the software deployment as well as the development process, which significantly improves efficiency, portability and consistency.

From Dependencies Comes Complexity

ASKAPsoft has a **complex dependencies**, some of which have strong tie to hardware. In the old days, anybody who wanted to use ASKAPsoft on their own machine needed to build it from the ground up with help of documentation and developer support. It was a nightmare.

Container for End Users

Things became significantly simpler when we switched to using container, which is **a lightweight, standalone, and executable package of software that includes everything needed to run an application**. It uses the Linux kernel of the host machine. Unlike VM (Virtual Machine) it practically has **no performance penalty**.

The container that we use is based on Docker. We first make **Dockerfile**, which is the step-by-step recipe on how to build the software. The recipe is then used to build **Docker image**, which is a blueprint template of the software. It gets distributed to end-users, who are going to run it as **Docker container**, which is a running instance of the image.

For end-users, this is a clear win. They can simply execute the app inside the container, without having to compromise on performance. This is also great for **reproducibility** of the software. It also removes friction in **collaboration** effort.

For users in Setonix (Pawsey's current HPC), this container is deployed under the wrap of **module**, so users are not even aware that they are running software within a container. Note that for this deployment, we need to convert Docker image into **Singularity** image, because running Docker requires administration privilege.

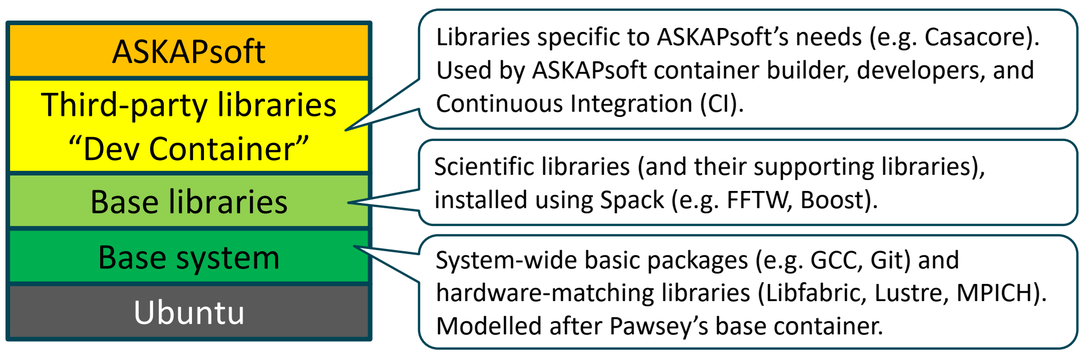


Figure 1: ASKAPsoft container stages.

Container for Developers

To save time during ASKAPsoft build process, the container is divided into **stages** (see Figure 1), depending on **frequency of change**. Packages that do not change frequently are build at the lower stage. In a routine ASKAPsoft build, only the top stage gets rebuilt, which saves a lot of time.

This division also maps neatly into **domain of concern**. The lower stage is more general in nature, whereas the upper stage is more specific to the needs of ASKAPsoft. It allows us to focus on certain domain at a time. For example, the base system is concerned only about system-related packages. Pawsey provides such base images to facilitate software development. We follow Pawsey's Docker recipe closely to build our base image.

The stage just below ASKAPsoft stage is used as "**Dev Container**" for our developers. This container provides us with:

- **Efficiency**: no need to figure out dependencies.
- **Portability**: development can be done on various machine (laptop, HPC).
- **Consistency**: minimum variation among developers.

Workflow: Another Source of Complexity

Our workflow starts with source code. It has multiple end points:

- ASKAPsoft module inside Setonix (wrapper around container).
- ASKAPsoft and Dev Containers, stored in **GHCR (GitHub Container Registry).**

When a GitHub Actions workflow is triggered (either by code changes being pushed to the repository or by manual trigger), a container is automatically built and stored in GHCR (see Fig. 2).

We are in the process of simplifying the workflow by moving everything into GitHub ecosystem. The remaining component is CI (Continuous Integration), especially the tests, which is currently still relying on GitLab's infrastructure.

At the same time, we have an organisation-wide tightening in security policies. This affects the way our workflow is designed and implemented.

Notes on Complexity

- Introducing a tool to handle complexity will have its own complexity. Container is good case in point.
- Transition has its own complexity. Our move into GitHub is an example.
- Complexity is dynamic. It's not "figure it out once and forget".
- Complexity has human aspect. A library has maintainers.
- Reducing complexity requires a deliberate and dedicated effort.

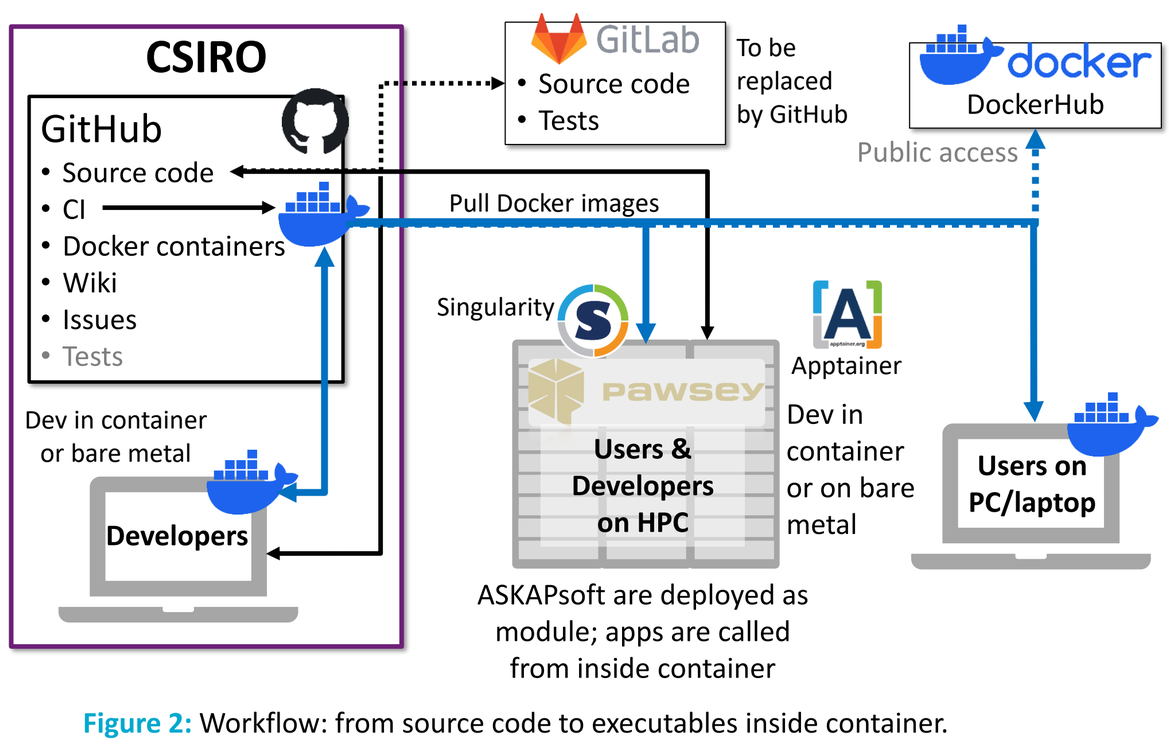


Figure 2: Workflow: from source code to executables inside container.

Our Team and Helpers

ASKAPsoft extended Dev Team, including previous members: Daniel Mitchell, Eric Bastholm, Fatemeh Moghaddam, Ian Czekala, Juan Guzman, Mark Wieringa, Matt Austin, Matthew Whiting, Max Voronkov, Mike Birchall, Minh Vuong, Ozgur Cekmer, Pero Manojlovic, Randika Hewage, Rodrigo Tobar, Stephen Ord, Tim Galvin, Vitaliy Ogarko, Wasim Raja.

Big thanks to our helpers: Pawsey team (especially Pascal Elahi & Maciej Cytowski).

CSIRO is Australia's national science agency, delivering solutions for a productive, sustainable and secure future.

CSIRO. Improving the life of every Australian.

FOR FURTHER INFORMATION

Paulus Lahur
Space & Astronomy Research Unit
paulus.lahur@csiro.au
https://www.atnf.csiro.au/facilities/askap-radio-telescope/

ACKNOWLEDGEMENTS

This scientific work uses data obtained from Ilyarrimanha Ilgari Bundara, the CSIRO Murchison Radio-astronomy Observatory. We acknowledge the Wajarri Yamaji People as the Traditional Owners and native title holders of the Observatory site. The Australian SKA Pathfinder is part of the Australia Telescope National Facility (https://ror.org/05qajvd42) which is managed by CSIRO. Operation of ASKAP is funded by the Australian Government with support from the National Collaborative Research Infrastructure Strategy. ASKAP uses the resources of the Pawsey Supercomputing Centre. Establishment of ASKAP, the Murchison Radio-astronomy Observatory and the Pawsey Supercomputing Centre are initiatives of the Australian Government, with support from the Government of Western Australia and the Science and Industry Endowment Fund.