

# VCC: A framework for building containerized reproducible cluster software environments

Joshua Higgins<sup>1</sup>, Violeta Holmes<sup>1</sup>, and Colin Venters<sup>1</sup>

DOI: [10.21105/joss.00208](https://doi.org/10.21105/joss.00208)

<sup>1</sup> High Performance Computing Research Group, University of Huddersfield

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

## Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

## Summary

The problem of portability and reproducibility of the software used to conduct computational experiments has recently come to the fore. Container virtualisation has proved to be a powerful tool to achieve portability of a code and its execution environment, through runtimes such as Docker, LXC, Singularity and others - without the performance cost of traditional Virtual Machines (Chamberlain, Invenshure, and Schommer 2014; Felter et al. 2014).

However, scientific software often depends on a system foundation that provides middleware, libraries, and other supporting software in order for the code to execute as intended. Typically, container virtualisation addresses only the portability of the code itself, which does not make it inherently reproducible. For example, a containerized MPI application may offer binary compatibility between different systems, but for execution *as intended*, it must be run on an existing cluster that provides the correct interfaces for parallel MPI execution.

As a greater demand to accommodate a diverse range of disciplines is placed on high performance and cluster resources, the ability to quickly create and teardown reproducible, transitory virtual environments that are tailored for an individual task or experiment will be essential.

The Virtual Container Cluster (VCC) is a framework for building containers that achieve this goal, by encapsulating a parallel application along with an execution model, through a set of dependency linked services and built-in process orchestration. This promotes a high degree of portability, and offers easier reproducibility by shipping the application along with the foundation required to execute it - whether that be an MPI cluster, big data processing framework, bioinformatics pipeline, or any other execution model (Higgins, Holmes, and Venters 2017).

## References

- Chamberlain, Ryan, L Invenshure, and Jennifer Schommer. 2014. "Using Docker to Support Reproducible Research." <http://dx.doi.org/10.6084/m9.figshare.1101910>.
- Emenaker, W., and D. Stanzione. 2007. "Dynamic Virtual Clustering." In *Cluster Computing, 2007 IEEE International Conference on*, 84–90. doi:10.1109/CLUSTER.2007.4629220.
- Felter, Wes, Alexandre Ferreira, Ram Rajamony, and Juan Rubio. 2014. "An Updated Performance Comparison of Virtual Machines and Linux Containers." *Technology* 28: 32.
- Higgins, Joshua, Violeta Holmes, and Colin Venters. 2017. "Autonomous Discovery and Management in Virtual Container Clusters." *The Computer Journal* 60 (2): 240. doi:10.1093/comjnl/bxw102.