DOI: 10.36315/2020end087
Education and New Developments 2020

A CLOUD IN A CUPBOARD: HOW TO TEACH MODERN TECHNOLOGIES IN A RESOURCE-CONSTRAINED ENVIRONMENT

David Cutting[1], Andrew McDowell[1], Neil Anderson[1], Paul Sage[1], Matthew Collins[1], & Angela Allen
[1]*Dr.*
School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast (United Kingdom)

Abstract

It is highly desirable to provide our computer science and software engineering students with hands-on exposure to currently industry techniques and technologies. One area of extreme growth and skills shortage is cloud computing which led to a curriculum redesign and a new module to teach these skills to final year students. Cloud computing products are available from a number of vendors, but their academic programmes often leave much to be desired, with limited credits and features. In the resource-constrained university sector simply buying vendor credits for students would be expensive but also wasteful as there is no capital investment. In an attempt to provide hands-on experience with the latest technologies an in-house "cloud in a cupboard" was designed and created at Queen's University Belfast. This took advantage of the wide range of free open-source software systems which underpin modern cloud vendors and, for very little cost thanks to re-purposed equipment, was able to support a large final-year module and several undergraduate dissertation projects successfully. Student feedback was very positive, and plans continue to grow this system further to support the current functionality and add more in the coming years.

Keywords: *Cloud, innovation, resources, teaching.*

1. Introduction

The area of cloud computing has seen impressive growth during the past decade. Cloud computing is largely seen as "using someone else's computer" i.e. rather than running your own computer programs on your own (expensive) equipment, you write them then upload them "to the cloud". The actual running of the programs is done by someone else (such as Amazon Web Services or Google Cloud Platform) and you pay a utility cost i.e. pay-as-you-go for as much computing power as you need. Consequently, this is often referred to as a "utility computing" model.

This paradigm is widely used in modern business practice and consequently skills in using cloud are much in demand by employers, something computer science schools have to take on board as industry alignment is critical to including the skills successful graduates will need (Plice and Reining, 2007; Benamati et al, 2010). Following industry consultation and as part of our attempt to teach modern techniques including cloud a new module was defined for 2019/20 which would cover a wide range of cloud techniques and technologies.

Learners learn in different ways (Schmeck, 2013) including learning by doing i.e. kinesthetic learning (Ayala et al, 2013). In an ideal world we could teach technologies using the actual tools supplied by the vendors i.e. students could have hands-on experience with the large cloud service providers gaining first hand experiences of exactly the same environments they would operate in once graduated.

While most providers do offer some form of educational grant or access our experience was far from straightforward, with students being rejected from academic programmes despite providing ID, or just running out of provided credit before completing their work. The option of just buying service time from these providers was simply not available in the resource constrained world of UK higher education, something some providers did not seem to understand being US-centric in their education where financial resources are more readily available. Another restriction on buying service was the transient nature of the spend, at the end of the year the students would move on but no long-term investment would have been made in the university facilities.

However, many of the technologies used are actually freely available, mostly built on free open-source software. The vendors charge for use of their computing resources and storage but the actual technologies and stacks can be installed by anyone. A plan was therefore formed to build the "cloud in a cupboard", a state-of-the-art cloud computing lab available on campus facilitating hands-on learning but using resources which represented an investment rather than an ongoing cost.

2. Design of the cloud in a cupboard

From an initial overview of the curriculum and learning outcomes of the new cloud computing module a set of functional requirements were defined:

- **Latest industry practice** – the latest industry practice in cloud deployment should be implemented as far as possible. Given the focus of the module and the widespread use of containers and container orchestrations a working implementation of Kubernetes was considered the minimum viable product. This also provides full freedom in terms of language and deployment choices for the learners.
- **Open access architecture** – learners should be free to interact with the resources directly in user interfaces or through APIs to allow open expansion and access from third-party tools such as Terraform.
- **Shell access** – learners must be provided remote Unix shell access to allow for command-line use and execution of tools within the environment.
- **Pipelines** – learners must have access to pipeline systems to allow for continuous integration (CI) and continuous deployment (CD) within a source-code management (SCM) context. The pipelines must be integrable with the execution environment to allow seamless testing and deployment should the learner with to implement this.

Having defined the functional requirements of the overall system (what we must deliver for the learners) a set of further non-functional requirements were created representing the environment and constraints:

- **Secure** – the system should provide a minimal security risk to the corporate network.
- **Reuse of hardware** – where possible the system should be able to reuse existing or "gifted" hardware to fulfil its function allowing minimal capital cost.
- **Immediate creation pending further investment** – given the funding and budgetary cycles of the university the system must be able to deliver a minimum viable product using only reused/repurposed hardware in the immediate term.
- **Robust** – as a critical component to a final year module the system must be robust enough to handle any minor issues without compromising performance and the ability of learners to do and submit their work.
- **Scalable** – the system must have the capacity to scale in future should this be required.
- **Evolvable** – the system should have the general design characteristics to allow the evolution of specific technology platforms deployed upon it.

3. System implementation

Before moving to implementation, a period of experimentation and up-skilling was required. During this time academic staff experimented with a variety of specific technologies, learning the systems from both a consumer (aka learner) and administrative perspective to support the live implementation. The decision was made the best fundamental architecture to meet the requirements, especially evolutionary needs, would be a semi-virtualised infrastructure allowing maximum flexibility.

During summer 2019 the system was built around a hardware cluster named the Hal cluster. Hal would provide a number of hardware nodes which initially were re-purposed from other projects with some providing generic virtualised platforms (able to run any specific technology) and others dedicated hardware "compute nodes" to allow learners to deploy their software within the system. The support and assistance of the school-based systems administration and support staff was pivotal during this time. Ultimately the system went live in late summer 2019 with three hardware nodes (hal01-03) gaining six more nodes while it was live (see Table 1).

Deployment was in time to deliver a minimal viable product ready for cloud computing teaching when the cohort started in September 2019.

Table 1. Hal Cluster Hardware Nodes.

Node	Threads	RAM (GB)	Storage (TB)	Role	Added
hal01	16	64	16	Virtualisation Host	Summer 2019
hal02	16	64	12	Virtualisation Host	Summer 2019
hal03	16	64	12	Kubernetes Compute Node	Summer 2019
hal04	8	16	2	Kubernetes Compute Node / CI Runner	October 2019
hal05	8	16	2	Kubernetes Compute Node / CI Runner	October 2019
hal06	8	16	2	Kubernetes Compute Node / CI Runner	October 2019
hal07	8	16	2	Kubernetes Compute Node / CI Runner	October 2019
hal08	32	64	12	Spare	January 2020
hal09	32	64	12	Spare	January 2020

Table 2. Hal Cluster Primary Virtual Guests.

Node ID	Purpose/Role
Gitlab	Gitlab SCM Server
Login	Shell Server
Kluster-worker01	Kubernetes Compute Node
Kluster-worker02	Kubernetes Compute Node
Kluster-worker03	Kubernetes Compute Node
Kluster-worker04	Kubernetes Compute Node
Kluster-master	Kubernetes Master Controller
Rancher	Rancher Management Node
Gitlab-runner-01	Gitlab runner with shell+docker access
Gitlab-runner-02	Gitlab runner with shell+docker access

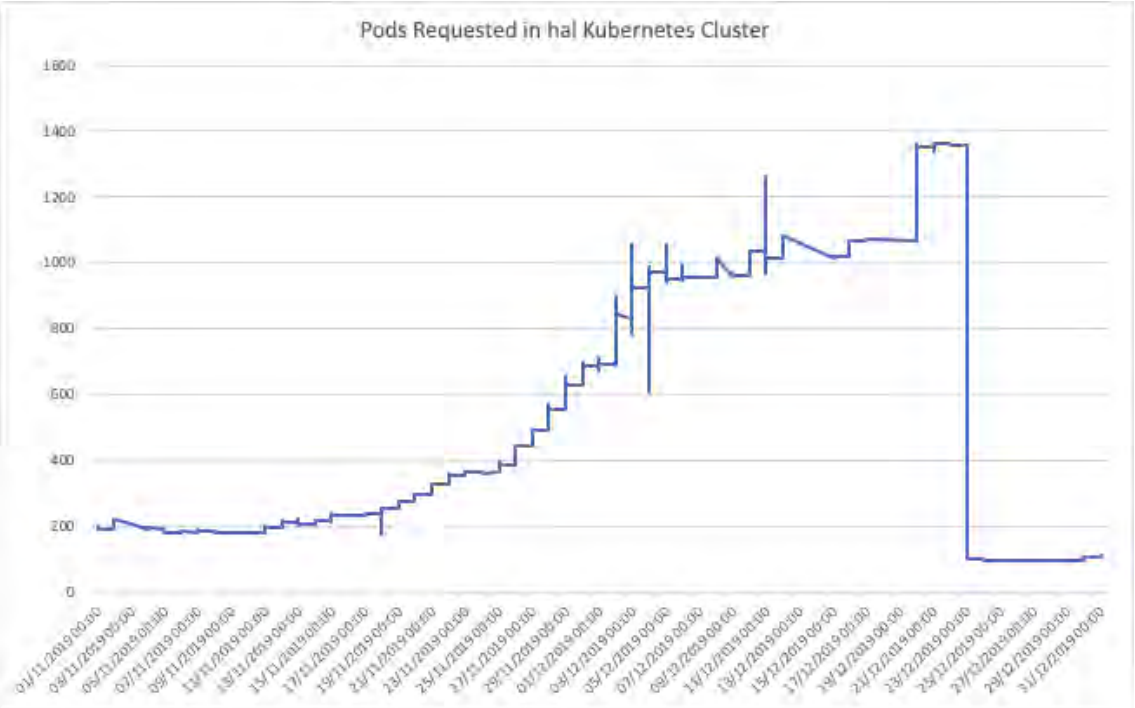
4. System operation for teaching

The Hal cluster and it's supported cloud infrastructure operated to support teaching in semester 1 (September – December), over the Christmas holidays and into semester 2 (January – April), for the taught cloud computing module. This module was a final year 20 CAT (1/6 of final year credits) module with a report and two project pieces of assessed project work. As would be expected, demand on the system fluctuated and reached a peak during the testing and delivery of a piece of assessment in which the learners were required to use the cluster.

The demand on the system in terms of pods (container units) requested and fulfilled for use by learners can be seen in Figure 1. At peak the system operated beyond the original planned capacity leading to some node failures which were, thanks to the nature of Kubernetes and the distributed model of the Hal cluster, largely mitigated against until nodes could be brought back online.

p-ISSN: 2184-044X e-ISSN: 2184-1489 ISBN: 978-989-54815-2-1 © 2020

Figure 1. Kubernetes Cluster Pod Requests from November to December 2019 Inclusive.



After the conclusion of the taught module for which the Hal cluster was specifically provisioned, a number of enquiries were made about using the cluster in final-year dissertation projects. As this was an added use of the cluster every effort was made to support these projects including some custom configuration and provisioning of specific environments for project work. In many cases the learners were able to use academic accounts on cloud vendors but were deeply concerned as to the potential to run out of credit at a critical phase, a concern which wouldn't be present in using the Hal cluster which was provided free to them. These projects were all completed successfully by April 2020.

5. Conclusions, lessons learnt, and future work

In this paper we have briefly introduced our "cloud in a cupboard", an attempt to facilitate hands-on learning of latest industry technologies in a highly resource-constrained environment where we see the systems as an investment not just a written-off cost. The approach was highly successful in both the hands-on teaching of cloud computing in a taught module and also to support final-year projects allowing a flexible industry-standard system which learners could use to develop their systems without external financial constraints.

Providing hands-on experience is essential in training the developers of tomorrow as these skills are specifically identified by industry as essential in the workplace. Graduates of this course will be able to make immediate use of standard technologies such as containers and Kubernetes and hit the ground running once employed.

Feedback from taught students was generally extremely positive with many expressing specifically their appreciation of the technical infrastructure provided. Feedback from project students who had used the cluster also validated the approach.

In the process of building and operating the systems not everything went smoothly which allowed for reflective feedback and lessons to learnt by staff. Primarily robust testing of systems is required, the loads that learners put onto a system is likely to be beyond what was initially planned for and their demand patterns outside of the tested scope. Ideally a group of learners should be found to "battle test" the system by trying to deploy and use as many features as possible in a short period of time. Staffing and support is another lesson learnt, in this case a single staff member was responsible for the setting up and administration of the cluster. Although fine on this occasion having one member of staff as a single point of failure on such an important resource was far from ideal.

Following completion of most of the teaching a further two funded-for-project nodes arrived brand new (hal08, hal09). These are currently “spare” nodes ready to be swapped into any critical function. The addition in summer 2020 of more repurposed nodes and storage will allow a more flexible automatically-balancing cluster to be created, dividing the same load over roughly 2.5x the compute power available in academic year 2019/20. The intention is to run for one more academic year with the same configuration but with much more power available avoiding any bottlenecks and slowdowns, then consider a bi-yearly refresh to latest industry technology.

Acknowledgements

The authors wish to acknowledge the thank Neil Lowry, Keith Stewart, Martin Kinkead, and Michael Garland for their invaluable support in getting the cloud cluster installed and working.

References

Ayala, N. A. R., et al. (2013). "Kinesthetic learning applied to mathematics using kinect." Procedia Computer Science 25: 131-135.

Benamati, J. H., et al. (2010). "Aligning undergraduate IS curricula with industry needs." Communications of the ACM 53(3): 152-156.

Plice, R. K. and B. A. Reinig (2007). "Aligning the information systems curriculum with the needs of industry and graduates." Journal of Computer Information Systems 48(1): 22-30.

Schmeck, R. R. (2013). Learning strategies and learning styles, Springer Science & Business Media.