Magic Castle — Enabling Scalable HPC Training through Scalable Supporting Infrastructures

Félix-Antoine Fortin
Université Laval
Québec, Canada
felix@calculquebec.ca

Alan Ó Cais
Jülich Supercomputing Centre
Jülich, Germany
a.ocais@fz-juelich.de

ABSTRACT

The potential HPC community grows ever wider as methodologies such as AI and big data analytics push the computational needs of more and more researchers into the HPC space. As a result, requirements for training are exploding as HPC adoption continues to gather pace. However, the number of topics that can be thoroughly addressed without providing access to actual HPC resources is very limited, even at the introductory level. In cases where access to production HPC resources is available, security concerns and the typical overhead of arranging for account provision and training reservations make the scalability of this approach challenging.

Magic Castle aims to recreate the supercomputer user experience in public or private clouds. To define the virtual machines, volumes, and networks that are required in a cloud-provider agnostic way, it uses the open-source software Terraform and HashiCorp Language (HCL). These resources are then configured using the configuration management and deployment tool Puppet to replicate a virtual HPC infrastructure with a full scientific software stack, and including a feature-rich JupyterHub environment. The final resource is accessible both through a web browser and via SSH, making it trivially OS-agnostic for the trainees.

Through the use of Magic Castle, we demonstrate that it is possible to dynamically provision virtual HPC system(s) in public or private cloud infrastructure easily, quickly, and cheaply. We also show that such infrastructures can support accelerators and fast interconnects, meaning that they can still be considered "true" HPC resources.

KEYWORDS

Education, Training, HPC, Cloud-computing

1 INTRODUCTION

Compute Canada [2] provides HPC infrastructure and support to every academic research institution in Canada. It uses CVMFS [5], a software distribution system developed at CERN, to make the Compute Canada research software stack available on its HPC clusters and anywhere else with internet access [1]. This enables replication of the Compute Canada experience outside of its physical infrastructure.

Building upon this capability, an open-source software project named Magic Castle [8] emerged that aims to recreate the Compute Canada user experience in public or private clouds. Once Magic Castle is configured and deployed, the user is provided with a complete HPC cluster software environment including a Slurm scheduler, a Globus Endpoint, JupyterHub, LDAP, DNS, and thousands of research software applications compiled by experts with EasyBuild [10].

Magic Castle is compatible with AWS, Microsoft Azure, Google Cloud, OpenStack, and OVH. It can also be easily extended to support other cloud providers. While there are quite a few other cloud HPC open source projects (such as [6, 11, 13, 16]), Magic Castle has extensive provider support and ships with a complete production-ready scientific software stack. This makes it an exciting pedagogical platform for *scalable* HPC training. It allows for the possibility of quickly and easily creating event-specific HPC training clusters at minimal cost and side stepping thorny issues such as local site security or resource configuration policies.

2 DESIGN

The Magic Castle project is defined by an infrastructure-as-code component that is responsible for generating a cluster architecture in a public or private cloud infrastructure. Magic Castle does this in a cloud-provider-agnostic way using Terraform and HashiCorp Language (HCL) [9], which defines the virtual machines, volumes, and networks that are required to replicate a virtual HPC infrastructure. The infrastructure definition is packaged as a Terraform module that users can customize as they require.

A Puppet [14] environment component configures the cluster instances based on their role. This includes the configuration of the scientific software stack. Magic Castle has recently been extended to include support for the *European Environment for Scientific Software Installations* (EESSI) software stack [7], which also uses the Compute Canada software distribution system as a reference design.

In Figure 1, the final architecture of the configured infrastructure is shown. Starting from the Terraform module, it takes about 20 minutes to fully provision the system (including configuration of support for GPUs and/or special interconnects).

3 CURRENT STATUS

Compute Canada delivers about 150 training workshops per year, and Magic Castle is used extensively by many of these workshops since 2018.

The LearnHPC project [12] has also adopted Magic Castle with the goal of creating an EESSI HPC user experience for training purposes, primarily on the Fenix Research Infrastructure [15]. LearnHPC

https://doi.org/10.22369/issn.2153-4136/13/1/3

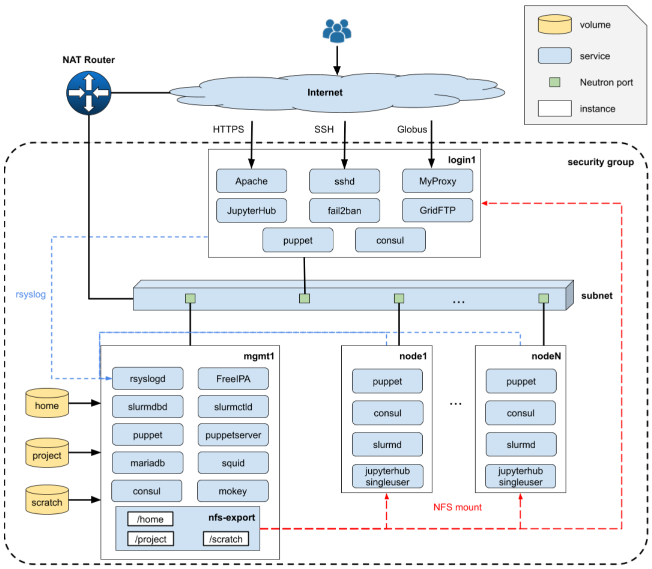

Figure 1: Overview of the architecture of Magic Castle.

is also collaborating with HPC Carpentry to ensure that the full set of lessons connected to HPC Carpentry [3] are supported on the training infrastructure.

OS support is available for RedHat and the RedHat-based variants CentOS, Rocky Linux, and AlmaLinux. As regards hardware capabilities, GPU support on Azure, AWS, Google Cloud, and OpenStack deployments has been tested with Magic Castle, as have support for the Infiniband interconnect provided by Azure and the EFA interconnect provided by AWS. Support for additional hardware capabilities with specific providers are driven by user requests and contributions.

Developer documentation for how to add support for an additional cloud provider is available [4] with a specific example for Alibaba Cloud given.

3.1 Cost Optimisation

"Spot" instances are allocated from the spare compute capacity of the cloud provider, usually at heavily discounted rates. However, these may be withdrawn/replaced by the provider at any time. On AWS, for example, typical eviction rates for high-end instances are below 5% and are therefore well-suited to compute nodes when coupled with the resilience features of Slurm. Spot instances can lead to savings of more than 70% of the cost of provision and are currently supported by Magic Castle on AWS, Microsoft Azure, and Google Cloud.

4 FUTURE WORK

Due to their typical high-availability nature, training clusters are likely to remain idle for a significant portion of their lifetime. Dynamic scalability of the provided resources has been a much-requested feature for Magic Castle and would likely greatly reduce the cost of provisioning. However, dynamic scalability would typically require Slurm to have access to the cloud provider API, meaning there would be a risk of exposing the cloud-provider credentials of the organisation if the cluster was compromised. This was seen to be an unacceptable prospect and would also require a custom implementation per provider. An implementation of dynamic provisioning is currently under development that is provider-agnostic and does not have this flaw.

ACKNOWLEDGMENTS

Alan Ó Cais acknowledges support from the European Union's Horizon 2020 research and innovation program, under grant agreement No. 676531 (project E-CAM) and grant agreement No. 823964 (project FocusCoE).

REFERENCES

[1] Maxime Boissonneault, Bart E. Oldeman, and Ryan P. Taylor. 2019. Providing a Unified Software Environment for Canada's National Advanced Computing Centers. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (PEARC '19)*. Association for Computing Machinery, New York, NY, USA, Article 51, 6 pages. https://doi.org/10.1145/3332186.3332210
[2] Compute Canada. 2021. Main website. (2021). https://www.computecanada.ca/
[3] HPC Carpentry. 2021. Main website. (2021). https://www.hpc-carpentry.org/ https://www.hpc-carpentry.org/.
[4] Magic Castle. 2021. Extending Magic Castle to support a new cloud provider. (2021). https://github.com/ComputeCanada/magic_castle/blob/main/docs/design.md#using-reference-design-to-extend-for-a-new-cloud-provider
[5] C. Condurache and I. Collier. 2014. CernVM-FS – beyond LHC computing. *Journal of Physics: Conference Series* 513, 3 (2014), 032020.
[6] elasticluster. 2021. elasticluster/elasticluster: Create clusters of VMs on the cloud and configure them with Ansible. (2021). https://github.com/elasticluster/elasticluster
[7] European Environment for Scientific Software Installations (EESSI). 2021. Documentation. (2021). https://eessi.github.io/docs/
[8] FÃľlix-Antoine Fortin et al. 2021. ComputeCanada/magic_castle: Magic Castle 11.3. (July 2021). https://doi.org/10.5281/zenodo.5121948
[9] HashiCorp. 2021. Terraform. (2021). https://www.terraform.io/ https://www.terraform.io/.
[10] Kenneth Hoste, Jens Timmerman, Andy Georges, and Stijn De Weirdt. 2012. EasyBuild: Building Software With Ease. In *High Performance Computing, Networking, Storage and Analysis, Proceedings*. IEEE, 572–582. http://dx.doi.org/10.1109/SC.Companion.2012.81
[11] Cluster in the Cloud. 2021. Cluster in the Cloud documentation. (2021). https://cluster-in-the-cloud.readthedocs.io/en/latest/
[12] LearnHPC. 2021. LearnHPC - Scalable HPC Training. (2021). https://www.learnhpc.eu/
[13] AWS ParallelCluster. 2021. aws/aws-parallelcluster: AWS ParallelCluster is an AWS supported Open Source cluster management tool to deploy and manage HPC clusters in the AWS cloud. (2021). https://github.com/aws/aws-parallelcluster
[14] Puppet. 2021. Powerful infrastructure automation and delivery. (2021). https://puppet.com/ https://puppet.com/.
[15] Fenix RI. 2021. FENIX Research Infrastructure. (2021). https://fenix-ri.eu/
[16] stackhpc.openhpc. 2021. stackhpc/ansible-role-openhpc: Ansible role for OpenHPC. (2021). https://github.com/stackhpc/ansible-role-openhpc

A ARTIFACT DESCRIPTION: MAGIC CASTLE — ENABLING SCALABLE HPC TRAINING THROUGH SCALABLE SUPPORTING INFRASTRUCTURES

A.1 Abstract

This paper does not contain computational results. The reference release of Magic Castle for this publication is version 11.3 [8].