Leveraging Software Defined Infrastructure to Move Research Platforms Between and Within Clouds

Matt Vander Werf, Steve Bogol, Paul Brenner, Scott Hampton
Center for Research Computing
University of Notre Dame

***Abstract*—In this paper we document three initiatives we have recently undertaken in which software defined infrastructure helped facilitate the movement of projects within and between research clouds. In our first example, we demonstrate the benefits of creating a platform for cloud templates with core shared services that allows for quick reuse in a variety of compliant environments. Next, we discuss how a cloud implementation for one research partner was taken from proof-of-concept and then formed the basis of a project for a separate partner. The third example involved replicating a time-sensitive project between institutions and how available cloud tools allowed us to expedite the process. In addition, we discuss the importance of the human factor in this endeavor, how we approached each solution, and how collaboration and sound communication are keys to success.**

***Index Terms*—Cloud computing, Portability, IaaS, AWS**

I. Introduction

Like many businesses and other universities, Notre Dame has moved the majority of its enterprise IT services to the cloud. There are a number of HPC centers at these same institutions following suit, at least in situations where it makes sense from a financial or implementation point of view. In particular, areas in secure data management and access have found a specific niche in cloud operations. For example, federal agencies often contract with private research organizations, resulting in some of their sensitive data being housed externally. On November 4, 2010, President Barack Obama issued Executive Order 13556 with the goal of establishing an "open and uniform program for managing" controlled unclassified information (CUI) [1]. This led in part to the creation of NIST Special Publication (SP) 800-171 Rev. 1 in 2016 [2], which set the regulation standards for data resident in non-federal IT infrastructure. A growing number of agencies have been requiring companies and institutions housing their information to meet some or all of the NIST SP 800-171 requirements. Multiple federal grants at the University of Notre Dame (ND) now have these requirements. Recent peer work supports our observations at ND. For example, in the HPC Security & Compliance Workshop (PEARC18), Pennsylvania State University's Joseph Gridley gave a presentation on the "Validation of CUI Environments Using NIST 800-171A" [3] and Preston Smith outlined the Purdue University's compliant environment in his presentation, "REED: from Service to Ecosystem" [4]. Other work such as Kelly W. Bennett's paper on a cloud-based security architecture [5] and the University of Arizona's CUI compliant environment [6] further highlighted research institutions' work to ensure compliance. In recognition of the expanding market for cloud services that meet various federal regulations, the major US public cloud vendors have taken strides to expand this aspect of their business [7] [8] [9]. These cloud security services help customers meet a variety of compliance requirements introduced by HIPAA, ITAR, EAR, Federal Risk and Authorization Management Program (FedRAMP), FISMA High Baselines, and the DOD Security Requirements Guide.

Given the complexity of implementing, verifying, and validating cloud services in support of these standards, it follows that we should make the most of the initial investment by building upon existing resources where possible. For many years software developers have made use of tools and frameworks that assist in reusing existing knowledge and common core elements when developing new projects. Reuse saves both time and effort and allows for quicker deployment and lower costs. Likewise, there are many tools that serve similar purposes and function within the realm of cloud computing. In this paper we demonstrate three common paradigms where we have recently leveraged existing infrastructure to implement new services or move existing ones. These examples represent interorganizational, intraorganizational, and interprovider.

II. Case Studies in Cloud Mobility

A. Interorganizational Secure Cloud Platform Templates

The University of Notre Dame's Compliant Cloud Computing (C3) environment embodies our institution's approach to complying with the growing number of data and infrastructure security controls specified in the regulations. C3 utilizes the Amazon Web Services (AWS) GovCloud region to implement a compliant environment. We provided an overview of the C3 architecture and its hybrid campus+cloud use in a prior work [10]. Here we specifically focus on the architecture's design intent to enable mobility of new projects into the C3 environment and current practical limitations to more generalized mobility.

The C3 environment (shown at a high level in Fig. 1) provides a core shared services virtual private cloud (VPC) managed by the ND Office of Information Technology (OIT) while individual project VPCs can be instantiated and maintained by the appropriate campus IT unit supporting a research project with particular compliance requirements. This capitalizes on OIT's expertise in core services such as identity management, software license servers, system monitoring, and logging. Research-focused IT teams can then devote more user-facing effort to build customized infrastructure for the various research projects and labs.

This hub and spoke distribution of expertise and infrastructure efficiently allows for the rapid movement of lab infrastructure (whether physical or virtual) from outside of the environment into a compliant new VPC. CloudFormation scripts for the new project VPCs already have all of the content needed to tie into the shared services components and research facing IT staff can simply add in the custom AWS components. We now find it takes much longer to determine specific data compliance and infrastructure requirements than it does to deploy them. Moving research is no longer a physical infrastructure hurdle.

There remain, however, some significant limitations to true infrastructure mobility. C3 still mandates integration with the core shared services VPC which is not fully generalized within AWS or other cloud providers. It is not possible to take a CloudFormation script for an externally managed VPC and simply run it within the C3 environment without modification. Not only are there custom network configuration parameters but also a host of specific remote desktop and identity management software dependencies. We are currently working to remove custom dependencies from the C3 environment; replacing them with AWS alternatives (for full AWS mobility) but we are far from suitably generalized for direct (automated) translation of AWS CloudFormation scripts into their analogues for Azure and Google Cloud.

B. Interprovider Transfer

As is often the case, the successful implementation of a project resource may be applicable in other contexts with only a modicum of changes. In this instance, we wanted to use an existing cloud infrastructure from the Notre Dame AWS environment as a basis for a new AWS project in the environment of a research partner. The original infrastructure from the ND environment was developed as part of the CRAFT Secure Vault for the CRAFT RACE program, funded through DARPA [11]. The new project is funded through a different government funding agency and is currently in its early stages.

The purpose of the infrastructure in the ND AWS environment was to support and allow for the prototyping of a new service that had been developed at Notre Dame based on the Blockchain technology. This service prototype was very successful, so much so that the new research partner and new funding agency wanted to replicate and expand on this new service in the new environment of the research partner. This led to the need to port the infrastructure that existed to support the service in the previous environment to the new environment hosted by the research partner. Since this had been set up previously in ND's AWS environment, we wanted to be able to re-use the infrastructure as much as we could, through various software tools that are detailed further below.

While the previous project environment was located within Notre Dame's compliant C3 environment within the AWS GovCloud domain, the new project hosted by the research partner is located within the normal, commercial AWS domain. This does require some adjustments to the infrastructure, but overall, this change is not expected to cause additional issues or delays for the porting process.

In order for this porting process to be successful, a lot of collaboration and communication is needed with the research partner to understand what is going to be provided in their AWS environment for us to use and what we would need to create. While porting from one AWS environment to another AWS environment makes the process much easier, there are still always going to be differences between the two environments. For example, in the previous environment, we did not need to worry about the VPC and networking configuration, as that was provided to us by Notre Dame's central IT department. However, for the new project, we will be responsible for setting up those aspects of the infrastructure in the research partner's environment. It's important to understand what the research partner is expecting from us and what we can expect from them during this process. Another important aspect is to find out when the research partner's environment will be ready for us to set up the infrastructure. If it will not be ready for some time, then working outside the partner's environment may be necessary until it is ready for us to build everything out in the new environment.

There are several different software tools that are being used to define the infrastructure and the individual types of instances in the environment in a consistent manner. The software tools allow someone to capture what was defined in one environment and deploy the same configuration and setup in another environment with little to no reconfiguration needed.

AWS CloudFormation [12] is being used to configure the infrastructure in the environment. This infrastructure includes the VPC, networking, security groups, storage, and the individual instances themselves. This could also include identity management and the use of any other AWS services in the environment. While this is a very beneficial tool for this kind of use case, CloudFormation might not be a good tool to use if you are porting from/to a non-AWS environment, since CloudFormation is specific to AWS.

For the configuration of the instances themselves, a tool called Packer [13] is being utilized. Packer is a software tool that allows you to configure a system through various existing provisioning tools, including Ansible, Puppet, Chef, and basic shell scripting. For this project, we are using Packer to create an Amazon Machine Image (AMI) in the environment that will then be used as the base images for the instances created through the CloudFormation scripts. The AMIs will have the OS configured and all the software installed that is needed. Different AMIs are being created for different types

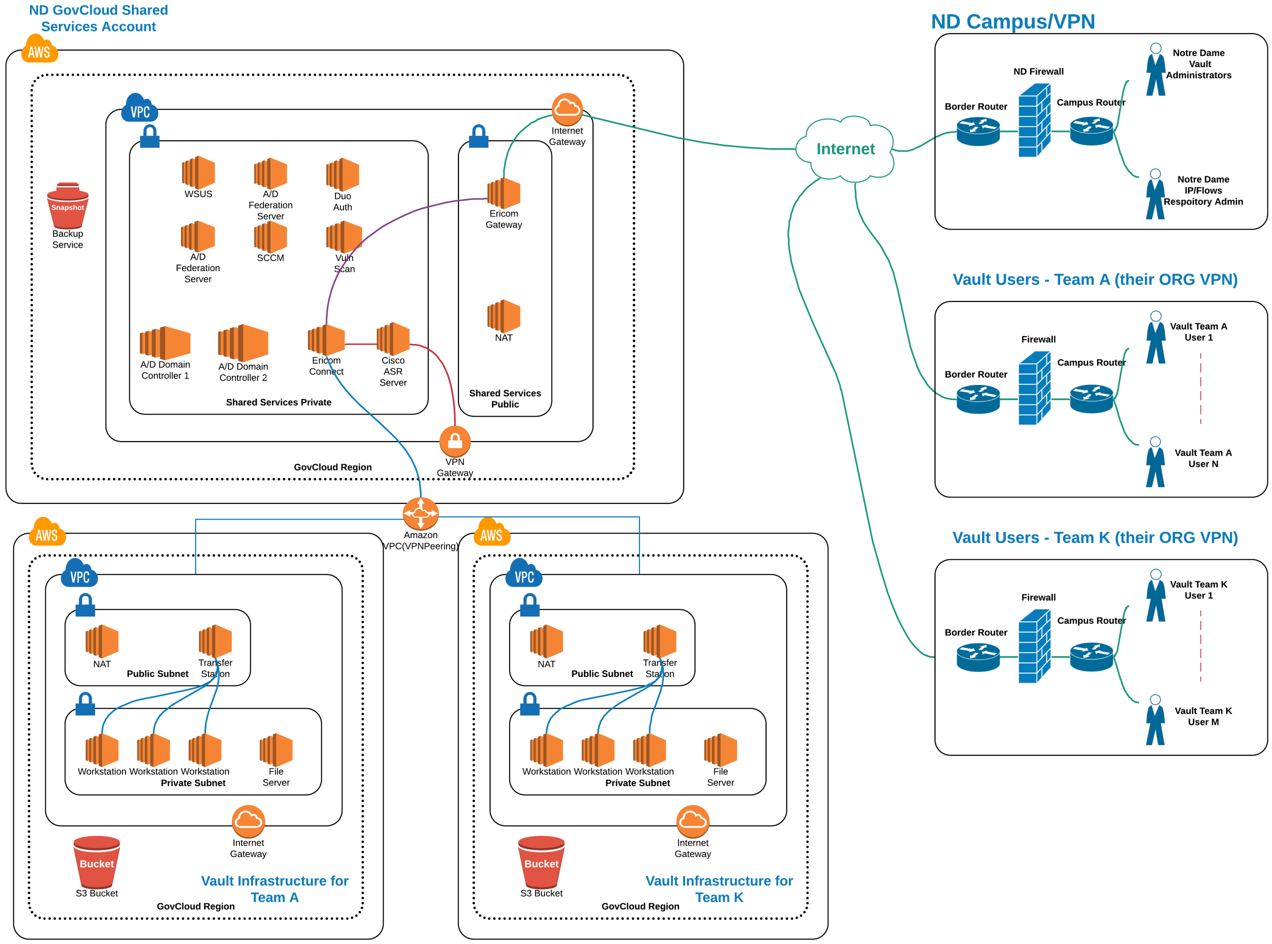

Fig. 1. ND C3 Architectural Summary

of instances with different purposes, each with different tool sets installed. Ansible is being used to deploy and configure the software and OS in a consistent manner within the AMI image built using Packer. Given a build configuration, Packer will create a temporary instance in AWS using the AWS CLI, configure that instance with all the necessary software as specified, and then create an AWS AMI based on that instance in the AWS environment.

One benefit to Packer is that it is vendor agnostic, so it can be used with various cloud or non-cloud platforms. Another tool that might be used for porting research platforms is Terraform [14]. Like Packer, Terraform can integrate with different cloud and non-cloud platforms and can help manage and provision your infrastructure. However, after an investigation, it was decided that Terraform would not be used for this porting process.

C. Intraorganizational Service Setup

In this use case cloud mobility was achieved after a researcher from a partner institution wanted to bring a complicated software stack to Notre Dame for use in a new project with government mandated HIPAA compliance. This remote researcher had created a database containing user information of persons affected by a natural disaster. The researcher wanted to deploy the same software stack with minor tweaks to be able to be used on tracking user information relating to the COVID-19 pandemic. Since both institutions use Amazon's AWS services, the project was ported within an hour instead of taking days or weeks to recreate the software defined infrastructure and software from scratch. This was especially important given the topical nature of the content.

This project also benefited from portable cloud development tools as the original system architect did not have to create this system from the ground up. Instead, they were able to use a predefined HIPAA template provided by Amazon [15]. The provided template builds multiple VPCs, multiple networking subnets, routing elements for communication between the VPCs, security elements to control the flow of data, profiles to control the creation of resources (IAM), event alerting (CloudWatch), event logging (CloudTrails), a database instance (EDS), and several virtual machines (EC2 instances). This is done by using Amazon's provisioning system, CloudFormation [12]. In CloudFormation, the user executes a template file to build the resources which result in a infrastructure and software stack. The HIPAA template is not just one template file but consists of a master template that uses multiple other templates to create all the individual virtual resources.

For purposes of making the transition, the original architect stored the template files in a GitHub repository and granted access to a Notre Dame Cloud Engineer. That engineer created an S3 storage bucket in AWS, manually edited the template files to reference the S3 bucket, and then uploaded the templates to the S3 bucket. Since this template needs to call multiple other templates, the S3 storage is needed, but in a single template design the S3 bucket would not be needed. Once the templates are stored in S3, the CloudFormation tool can be used to create a new stack by selecting the "Template is Ready" option and pointing to the S3 bucket. At this time you can choose to enter the Designer to see a graphical layout of the template and to make drag-and-drop changes to it or directly edit the template in JSON or YAML. The next step is to fill out parameters in the template that are unique to this build, such as a name, availability zones to use, security credentials, and a database password. Once the parameters are chosen the stack can be launched which automatically builds the infrastructure. Once the stack is complete, the instances can be accessed for further configuration by the use of an SSH client. Once the life cycle of the stack comes to end, all the elements that were created with the stack are removed with the deletion of the stack. The only remaining elements are the S3 bucket containing the original template files and a system generated S3 bucket which holds the logs of the creation of the stack. The template files can then be shared with other institutions or AWS users and the stack can be recreated in a matter of minutes instead of days. This particular stack was able to be launched in just under 25 minutes once all the proper changes were made. Using the stacks in CloudFormation does require a mild level of AWS proficiency and patience, as debugging does require waiting for the build to crash and generate useful event information and logs.

III. Discussion

We have shown three distinct cases of leveraging software defined infrastructure for the purpose of moving research platforms between and within clouds. The authors found the two most common hurdles were an over abundance of configuration options as well as a significant effort to manually tune those options specifically for our site. The number of options attests to the power of the platforms, and with time one could become proficient in all aspects. We propose creating a hierarchical documentation that favors the most common tasks first and allows for deeper inspection if and when the need arises. While there are a number of tools that support movement of cloud infrastructure within and between platforms, there is still room for improvement in defining "local" configurations that can be quickly inserted for even faster implementation.

A. Future Work

There is tremendous need for research support personnel to have the knowledge and tools to use the cloud efficiently and effectively. We are now in a situation with multiple cloud providers offering similar, yet distinct, options. As more people implement cloud solutions, and in turn disperse among the different providers, we feel that the ability to quickly and efficiently move between them will become even more important. To this end, we plan to investigate and document the implementation of our previously mentioned HIPAA application on both Azure and Google Cloud. The focus is on training the trainers so that we will be better prepared to support our researchers going forward.

As one author is a supported participant of HARC [16], our goal is to make contributions to the community GitHub, and offer training and outreach opportunities from what we have learned during this process both locally and within the community. Although researchers and support staff may be familiar with the technologies, learning the best methods for incorporating existing technologies is something we plan to offer assistance with.

Acknowledgment

The authors would like to recognize continued support from the University of Notre Dame Center for Research computing and the diverse faculty research teams we serve. Additional support to more effectively leverage cloud resources for science has been provided by the Humans Advancing Research in the Cloud (HARC) program operated by the Pervasive Technology Institute at Indiana University. Mr. James Smith in the Notre Dame Office of Information Technologies provided expert consult on various technical AWS facets.

References

[1] (2010) 3 cfr 13556 - executive order 13556 of november 4, 2010. controlled unclassified information. U.S. Government Publishing Office. [Online]. Available: https://www.gpo.gov/fdsys/granule/CFR-2011-title3-vol1/CFR-2011-title3-vol1-eo13556

[2] (2016) Protecting controlled unclassified information in nonfederal systems and organizations. National Institute of Standards and Technology. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-171/rev-1/final

[3] J. Gridley, "Validation of cui environments using nist 800-171a," presented at the HPC Security & Compliance Workshop (PEARC18), July 2018.

[4] P. Smith, "Reed: from service to ecosystem," presented at the HPC Security & Compliance Workshop (PEARC18), July 2018.

[5] K. W. Bennett, D. W. Ward, and J. Robertson, "Cloud-based security architecture supporting army research laboratory's collaborative research environments," in *Proc. SPIE Defense & Security Volume 10635*, Orlando, Florida, May 2018.

[6] (2018) Controlled unclassified information (cui) environment. University of Arizona. [Online]. Available: https://it.arizona.edu/cui

[7] (2020) What is aws govcloud (us)? Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/govcloud-us/latest/UserGuide/whatis.html

[8] (2018) Azure government. Microsoft Azure. [Online]. Available: https://azure.microsoft.com/en-us/global-infrastructure/government/

[9] F. Konkel. (2018, March) Google cloud targets federal government. [Online]. Available: https://www.nextgov.com/it-modernization/2018/03/google-cloud-targets-federal-government/146917/

[10] B. Judson, M. V. Werf, and P. Brenner, "Compliant cloud+campus hybrid hpc infrastructure," in *High Performance Computing Systems Professionals Workshop, SC18*, Dallas, TX, USA, November 2018.

[11] S. Bogol, P. Brenner, A. Brinckman, E. Deelman, R. F. da Silva, S. Gupta, J. Nabrzyski, S. Park, D. Perez, S. Rucker, M. Rynge, I. J. Taylor, K. Vahi, M. V. Werf, and S. Wyngaard, "A secure gateway for enabling application specific integrated circuit design collaborations," in *Proceedings of the 11th International Workshop on Science Gateways (IWSG 2019)*, Ljubljana, Slovenia, 2019.

[12] (2020) Aws cloudformation. Amazon Web Services. [Online]. Available: https://aws.amazon.com/cloudformation/

[13] (2020) Packer by hashicorp. Packer. [Online]. Available: https://www.packer.io/

[14] (2020) Terraform by hashicorp. Terraform. [Online]. Available: https://www.terraform.io/

[15] (2017) Reference architecture for hipaa on the aws cloud: Quick start reference deployment. Amazon Web Services. [Online]. Available: https://docs.aws.amazon.com/quickstart/latest/compliance-hipaa/welcome.html

[16] (2020) Humans advancing research in the cloud (harc). Indiana University. [Online]. Available: https://harc.iu.edu/