Managing dbGaP Data with Stratus, a Research Cloud for Protected Data

Extended Abstract

Evan F. Bollig
MN Supercomputing Institute
University of Minnesota
117 Pleasant St SE
Minneapolis, Minnesota 55455
bollig@umn.edu

Graham T. Allan
MN Supercomputing Institute
gta@umn.edu

Benjamin J. Lynch
MN Supercomputing Institute
blynch@umn.edu

Yectli Huerta
MN Supercomputing Institute
yhuerta@umn.edu

Mathew Mix
MN Supercomputing Institute
mattmix@umn.edu

Brent Swartz
MN Supercomputing Institute
swartzbr@umn.edu

Edward A. Munsell
MN Supercomputing Institute
emunsell@umn.edu

Joshua Leibfried
MN Supercomputing Institute
leibfrie@umn.edu

Naomi Hospodarsky
MN Supercomputing Institute
nhospoda@umn.edu

1 INTRODUCTION

While traditional high performance computing (HPC) continues to satisfy the majority of needs across research disciplines, there is a growing trend toward data-intensive work with ever larger storage and compute requirements, as well as more stringent data-use agreements. Data-use agreements in particular are cumbersome to satisfy on HPC clusters where one-off modifications would be necessary to satisfy agency requirements for logging, backups (or lack thereof), and isolation. The overhead in additional maintenance, monitoring, and user education may be acceptable for a small number of edge-cases, but the burden snowballs quickly to become unsustainable.

Facing the need to accommodate and sustain this trend, the Minnesota Supercomputing Institute (MSI) designed and built Stratus, an on-premise cloud for protected data. Stratus is a subscription-based Infrastructure-as-a-Service that enables users to operate within their own virtual machines (VMs). Stratus is powered by the Newton version of the OpenStack [5] cloud platform, and is backed by the Kraken release of Ceph [7] storage. Stratus was made available to MSI users in July, 2017. Beyond managing protected data, Stratus offers three features not present in MSI's existing cyberinfrastructure: a) on-demand availability of compute resources with self-service user control; b) long-running jobs (e.g., $>$ 30 days) that are live-migrated between compute nodes, and persist through maintenance periods; and c) container-based computing with Docker, which comes pre-loaded on MSI-blessed VMs.

OpenStack was chosen for multiple reasons: a) it has built-in support for multi-tenancy, software defined networking, logging, SSL/TLS encrypted traffic and other features to make compliance easier; b) the modular OpenStack services enable mix-and-match configuration and are decoupled for fault tolerance; and c) a massive open source community distinguishes the project as the leading software for cloud deployments across industry and academia. Ceph is free, open source, and backed by much of the same community as OpenStack. Ceph excels as an efficient, low-cost storage platform with the ability to scale to many PBs. The stability, scalability, and value of OpenStack and Ceph have been vetted by large deployments like CERN [1] and the NSF funded JetStream [6].

In its initial release, Stratus is designed expressly to satisfy the NIH Genomic Data Sharing (GDS) Policy for the Database of Genotypes and Phenotypes (dbGaP) [3]. Although other more stringent regulations and standards exist (e.g., HIPAA, FISMA, and ITAR to name a few), tackling dbGaP data is a good launching point for Stratus in that it addresses basic needs for access logging, data encryption, two-factor authentication, etc. Furthermore, it satisfies an immediate need at the University of Minnesota where researchers have no alternative available to analyze dbGaP data. Compliance with other regulations can be added in the future.

Stratus is most similar in spirit to the Bionimbus Protected Data Cloud (PDC) [2]; a fully GDS- and HIPAA-compliant platform at the University of Chicago. Both Stratus and Bionimbus PDC are built on OpenStack and have Ceph storage. Stratus storage is entirely Ceph, in contrast to the current Bionimbus PDC, which has two object stores for protected data: 400 TB Ceph S3 and 1.7 PB IBM CleverSafe S3. Furthermore, Bionimbus PDC is a NIH Trusted Partner and allowed to maintain a complete persistent clone of the dbGaP data, while Stratus presents users a multi-tiered storage environment with a "dbGaP Cache" to store only active subsets of the dbGaP data in a scratch-like space.

2 STRATUS CLOUD PLATFORM AND STORAGE

The Stratus compute hardware is currently twenty HPE ProLiant XL230a nodes, each with two Intel E5-2680v4 (14-core) CPUs, 256 GB of RAM, and 10 GbE networking to the outside world. Hyperthreading is enabled for a total of 1120 logical processors. An array of eight HPE Apollo 4200 storage servers are each connected to compute nodes via two redundant 40 GbE switches. Each storage node has 256 GB memory, two 800 GB Intel NVMe P3700 accelerator boards, and 198 TB raw capacity; yielding 1.5 PB raw capacity for the

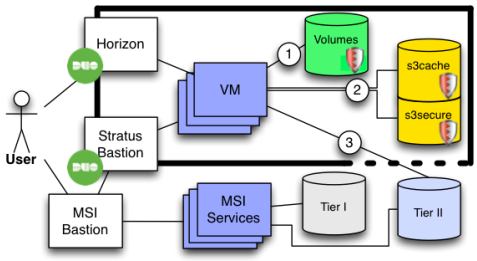

Figure 1: Stratus is sandboxed from other MSI services, but VMs have access to three tiers of storage. Shields denote suitable for dbGaP data.

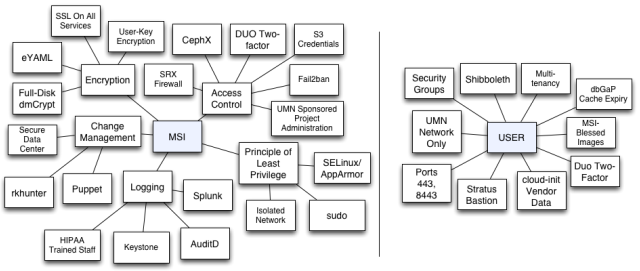

Figure 2: Compliance is met with an MSI-first policy that focuses on securing MSI, OpenStack, and Ceph. Users are protected with sane defaults and network security.

entire cluster. Ten additional 8-core Advanced Clustering control servers, each with 32 GB RAM, run the OpenStack APIs, monitor storage operations and orchestrate VM lifecycles.

Stratus storage is partitioned into three: a) a 3x replicated block device storage for Cinder volumes with 200 TB usable capacity; b) an S3 object store for caching dbGaP data with 500 TB capacity; and c) a persistent S3 object store for secure archive. Both b) and c) are 4:2 erasure-coded. S3 object lifecycle (expiration) was added as a feature in Ceph Kraken, and is essential to the design of the dbGaP Cache. Although this is still experimental, the plan is to enforce a default expiration of 60-days. If greater than 80% of the dbGaP Cache is filled, objects are deleted following a First-In-First-Out policy until utilization falls below the 80% threshold.

3 TIERED STORAGE

From the user perspective, Stratus has three storage tiers that are illustrated in Figure 1. Volumes are the first tier, and the layer where active data processing occurs. The second tier, secure archive, is the combination of Stratus' two secure object stores where users can target either dbGaP Cache ("s3cache") or persistent secure storage ("s3secure"). The third tier, referred to as Tier II because it is the second-level archive for the rest of MSI's cyberinfrastructure, is also a 4:2 erasure-coded Ceph S3 object storage. It is through Tier II that data can be brought in from other MSI resources. Stratus users can move data into Tier II when it is sanitized and would not violate the data-use agreement to be stored in an unprotected environment.

Data movement is orchestrated with S3 clients (e.g., s3cmd, minio, and awscli) on VMs. A typical workflow begins with booting a VM on a small boot volume (e.g., 15 GB) from the Horizon web interface. Next, the user connects to the VM via the Stratus Bastion server and directly streams dbGaP data from the NIH endpoints into the dbGaP Cache. With the data spooled, Horizon is used to attach a large (e.g. 1 TB) workspace volume, into which data is streamed from the dbGaP Cache for processing. Intermediate data produced during analysis can be stashed in the dbGaP Cache, while final results can be stored in persistent secure storage. At any point during analysis, the user can grow the size of the workspace, snapshot it, or detach it from one VM and attach it elsewhere (e.g., if workflow stages are setup as separate VMs). When analysis is complete, the user can post sanitized data to MSI Tier II storage where they can access it from other MSI resources or share with colleagues.

4 GDS COMPLIANCE

Most cloud providers absolve themselves from responsibility of protecting users, focusing instead on the minimum effort necessary for regulatory compliance within the underlying hardware, and leaving users to individually manage and vet whatever is run on top of the cloud. This makes sense in a self-service Infrastructure-as-a-Service model, but creates a potentially dangerous situation for novice users with protected data. Stratus was built with an MSI-first mentality, prioritizing security of the hardware, OpenStack, and Ceph. However, as an academic unit supporting computationally intensive research at the University of Minnesota since 1984, MSI has a culture of going the extra mile to train, protect, and assist. In this case, a fair amount of effort also went into sandboxing user VMs and providing sane defaults for security that users can opt-out of at their own risk. This included following the best practices guide for GDS data [4] as a checklist. All bullets therein were treated as required controls. Note that the NIH best practices go beyond what most large research computing facilities can or need to do.

Figure 2 illustrates a sampling of protections for MSI and users as two trees. While outside the scope of this document to exhaustively detail each measure, there is clearly a bias toward protecting MSI. The MSI tree on the left has two levels of nodes. The first level are five significant controls mentioned by the best practices guide (i.e., Encryption, Logging, Principle of Least Privilege, Access Control, and Change Management), and the second level summarizes some of the ways we have addressed them. The majority of the GDS controls are intangible to users, but represent substantial administrative and operational staff effort.

The USER tree offers measures that users are aware of. A few relate to GDS controls (e.g., two-factor authentication), but others go above and beyond. The most impactful measures deal with network traffic. In most clouds, users get unrestricted traffic in and out of VMs. On Stratus, egress traffic is unrestricted, but ingress is only open between VMs within the same project, or from the Stratus Bastion (so long as Security Group exceptions exist). Traffic is blocked on all ports from the world, but when it originates from the campus network, ports 443 and 8443 are accessible with caveats: a) SSL is required on both, and b) users must create a Security Group exception to opt-out of the default network protection. These rules sandbox Stratus VMs away from other MSI services, the University, and the world.

REFERENCES

[1] T Bell, B Bompastor, S Bukowiec, J Castro Leon, M K Denis, J van Eldik, M Fermin Lobo, L Fernandez Alvarez, D Fernandez Rodriguez, A Marino, B Moreira, B Noel, T Oulevey, W Takase, A Wiebalck, and S Zilli. 2015. Scaling the CERN OpenStack cloud. *Journal of Physics: Conference Series* 664, 2 (2015), 022003.

[2] Allison P Heath, Matthew Greenway, Raymond Powell, Jonathan Spring, Rafael Suarez, David Hanley, Chai Bandlamudi, Megan E McNerney, Kevin P White, and Robert L Grossman. 2014. Bionimbus: a cloud for managing, analyzing and sharing large genomics datasets. *Journal of the American Medical Informatics Association : JAMIA* 21, 6 (11 2014), 969–975.

[3] Matthew D. Mailman, Michael Feolo, Yumi Jin, Masato Kimura, Kimberly Tryka, Rinat Bagoutdinov, Luning Hao, Anne Kiang, Justin Paschall, Lon Phan, Natalia Popova, Stephanie Pretel, Lora Ziyabari, Moira Lee, Yu Shao, Zhen Y. Wang, Karl Sirotkin, Minghong Ward, Michael Kholodov, Kerry Zbicz, Jeffrey Beck, Michael Kimelman, Sergey Shevelev, Don Preuss, Eugene Yaschenko, Alan Graeff, James Ostell, and Stephen T. Sherry. 2007. The NCBI dbGaP database of genotypes and phenotypes. *Nature genetics* 39, 10 (01 Oct. 2007), 1181–1186. https://doi.org/10.1038/ng1007-1181

[4] NIH. 2015. NIH Security Best Practices for Controlled-Access Data Subject to the NIH Genomic Data Sharing (GDS) Policy. (March 2015). Retrieved June 1, 2017 from https://www.ncbi.nlm.nih.gov/projects/gap/pdf/dbgap_2b_security_procedures.pdf

[5] OpenStack. 2017. Open Source Cloud Computing Software. (June 2017). Retrieved June 1, 2017 from https://www.openstack.org/

[6] Craig A. Stewart, Timothy M. Cockerill, Ian Foster, David Hancock, Nirav Merchant, Edwin Skidmore, Daniel Stanzione, James Taylor, Steven Tuecke, George Turner, Matthew Vaughn, and Niall I. Gaffney. 2015. Jetstream: A Self-provisioned, Scalable Science and Engineering Cloud Environment. In *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure (XSEDE '15)*. ACM, New York, NY, USA, Article 29, 8 pages. https://doi.org/10.1145/2792745.2792774

[7] Sage A. Weil, Scott A. Brandt, Ethan L. Miller, Darrell D. E. Long, and Carlos Maltzahn. 2006. Ceph: A Scalable, High-performance Distributed File System. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI '06)*. USENIX Association, Berkeley, CA, USA, 307–320. http://dl.acm.org/citation.cfm?id=1298455.1298485