# Implementing Centrally Managed Cloud Security and Compliance Controls for Amazon Web Services within a Large Research Enterprise and Healthcare System

Stephen A. Wheat

Enabling innovation with appropriate information security is a common goal many organizations share as they transition to the Cloud. Innovations without necessary security measures will never realize their full potential of delivering impactful, public applications operating on sensitive data. Security measures that do not adapt to the elastic, on-demand nature of the Cloud will stifle or negate the ability of Cloud services to drive innovation. To create a secure platform for research innovation in the Cloud, Emory University and Emory Healthcare implemented a three-pronged strategy including:

1. Researcher-managed Amazon Web Services (AWS) accounts
2. Centrally-managed Virtual Private Cloud (VPC) structures, security policies, and service control policies with provisioning and administration automation
3. Cloud competency center staffed by Emory IT, AWS enterprise support, and preferred cloud consultants

The results have been…[story yet to be written; hopefully the results will be great]

## Inception of the Emory Cloud Research Service with AWS

In 2016 in response to requests from Emory researchers interested in using AWS, Emory Libraries and Information Technology Services (LITS) set goals to define a potential service offering with input from interested research groups, Emory Information Security, and other research universities and to pilot a service offering.

The team conducted an initial survey of other leading institutions and found they were typically:

1. Brokering AWS accounts to research units to achieve some billing and accounting benefits, but implementing few if any security controls and providing researchers with policies or terms of use that they should follow in their AWS accounts
2. Implementing a data center migration to AWS and brokering specific services like EC2 compute instances much like they had offered with their on-premises services
3. Just starting out and searching for a strategy

Neither of the two pervasive approaches seemed suitable for Emory because the Emory researchers consulted wanted to use and administer a wide variety of services in their AWS accounts. Emory Information Security had considerable data indicating that public cloud resources presented unacceptable risk of compromise and disclosure without specific countermeasures and diligent, central control of those countermeasures.

Emory researchers wanted to implement compute use cases for genomics and data analytics, storage, web application platform as-a-service, serverless mobile backends, and consume and share application stacks on the Amazon Marketplace. These use cases suggested researchers needed direct access to their own AWS accounts with a broad range of services and features.

Emory Information Security cited industry and institutional data that assets in the public Cloud (even within a VPC) without substantial controls, countermeasures, and oversight were prone to compromise and data disclosure. Information Security also pointed out that without such controls, Emory could not meet the requirements of its HIPAA compliance policies, which would limit the usefulness of the service to non-HIPAA applications and research.

Given these conflicting requirements, Emory took a new path to give researchers their own AWS accounts with as many services and features as possible but also with many centrally defined controls, centrally administered firewalls, and centrally administered security and service control policies.

## Analysis and Proving the Concept

Emory recruited a group of five focus groups of Emory researchers to help document research computing use cases for the Cloud and eventually pilot the service. The group met regularly for several months and documented over 20 individual use cases in the areas of operational systems, application development, teaching and training, and faculty research. Applications identified included DNA and RNA sequencing, analysis of clinical warehouse data, mobile apps for 100,000 study subjects, training in public health and biomedical informatics, remote desktop applications, and more. [1]

A team comprised of units across LITS, including Information Security, Network Architecture, IT Architecture, Infrastructure, and University IT worked with senior AWS Architects from AWS Professional Services to develop a high-level design to address major risks. The team focused first on network-level controls. Information Security found the network-level controls within AWS to be particularly deficient when compared to Emory's on-premises infrastructure. Specifically, the lack of next-generation firewalls (third generation), intrusion detection, and security information and event management (SIEM) were major deficiencies that needed to be addressed.

AWS certainly uses all of these technologies in delivery of AWS services, but at the time of the analysis these technologies were not an integrated part of the service offering that is exposed to customers. AWS account owners do not have access or visibility into the configuration and logs of such infrastructure but instead use AWS constructs such as AWS Security Groups, Network Access Control Lists, and CloudTrail logs to implement many features of firewall, intrusion detection, and SIEM functions. While many developers and deployers value such abstraction of the security infrastructure into easy-to-use constructs, Emory Information Security found this abstraction of security infrastructure to be a significant deficiency in security features. For example, not being able to see dropped traffic, take real-time steps to stop an emerging threat, and view and analyze netflow and other activity logs quickly and efficiently were steps backward when compared with Emory's current information security infrastructure.

In order to facilitate a design with technical features and specific security controls the group of researchers and central IT units reduced all of the use cases to three major categories: [2]

1. Using AWS services without exposing applications or services to the Internet (roughly 60% of the use cases)
2. Using AWS services and exposing applications or services through Emory's existing on-premises network (30% of the use cases)
3. Using AWS services and exposing applications or services to the Internet through AWS (10% of the use cases)

This approach allowed Emory to leverage its existing information security infrastructure for use case categories one and two, extending Emory's private network into the Cloud with AWS VPCs and controlling all ingress and egress through Emory's existing network and security controls. Many use cases such as DNA sequencing do not require exposing new applications or services to the internet at all and only require outbound access for instance updates, launch, and other known dependencies that can either be allowed by rules or provided internally on Emory's own network. Many analytics applications fall into Category 1 as well, but some have a caveat, which defines use case Category 2. These use cases have a web application dashboard or some web services that may need to be exposed to collaborators or other users on the Internet beyond Emory's network.

In order to implement use cases in Categories 1 and 2 and leverage Emory's existing security infrastructure, a key design element for Emory's virtual private clouds (VPCs) emerged—withhold the ability of AWS account holders to create and manage internet gateways and routes. Without the ability to access the Internet directly and with all inbound and outbound traffic routed through a site-to-site VPN tunnel or AWS DirectConnect connection to Emory's network, Emory retained the ability to implement most security controls already in place on its own network.

Use cases in Category 3 suggested that Emory should leverage the AWS connections to the Internet and many of their managed services for large-scale apps with high-availability requirements. Such apps target some segment of the general public. These are use cases such as mobile app backends for 100,000 study participants performing millions of transactions per day, transmitting gigabytes of data per day to Emory research databases, and streaming multimedia resources to deliver health education for specific conditions. In order to implement these use cases, Emory would need to implement adequate third-generation firewalls between Emory VPCs and AWS internet gateways and additional intrusion detection and response measures.

Emory's technical design emerged with four major elements:

1. Emory researchers could be issued their own accounts and retain the ability to administer these accounts with some access restricted or limited such as the ability to create internet gateways and change network topology
2. Type 1 VPC structures and controls, extending Emory's private network into the cloud for use case categories one and two
3. Type 2 VPC structures and controls, exposing high-volume or high-availability services directly to the Internet through AWS
4. Risk assessments, remediation, and countermeasures for every AWS service in the form of service control policies, Identity and Access Management (IAM) policies, and security risk detector (SRD) commands to detect and automatically implement important security measures that could not be implemented declaratively by AWS policies.

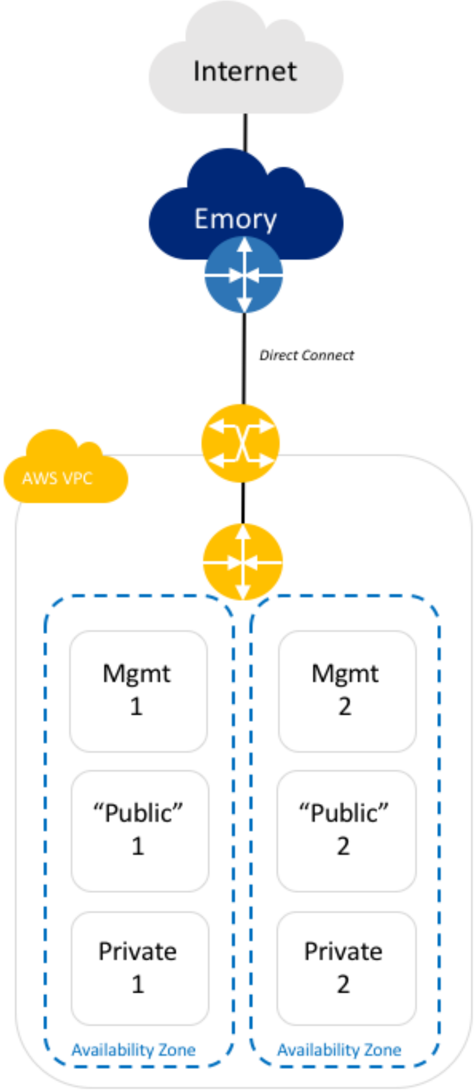

***Figure 1: Emory Type 1 VPC***

The Type 1 VPC is configured with two availability zones. Each availability zone is configured with three subnets: a management subnet, a public subnet, and a private subnet. The management subnet is for use by LITS to deploy administrative assets. The public and private subnets allow customers to build multi-tier applications where forward-facing servers, like web servers, will be placed in the "public" subnets and backend servers, like application and database servers, will be placed in the private subnets. All traffic to and from the VPC traverses an AWS DirectConnect connection to Emory. Customers who want to expose infrastructure in a Type 1 VPC to the Internet request an Emory Elastic IP address and configure it to map to their desired private IP. [3]

Figure 2 illustrates Emory's Firewall VPC and a Type 2 VPC for exposing applications in the Cloud directly to the internet using a VPC firewall.

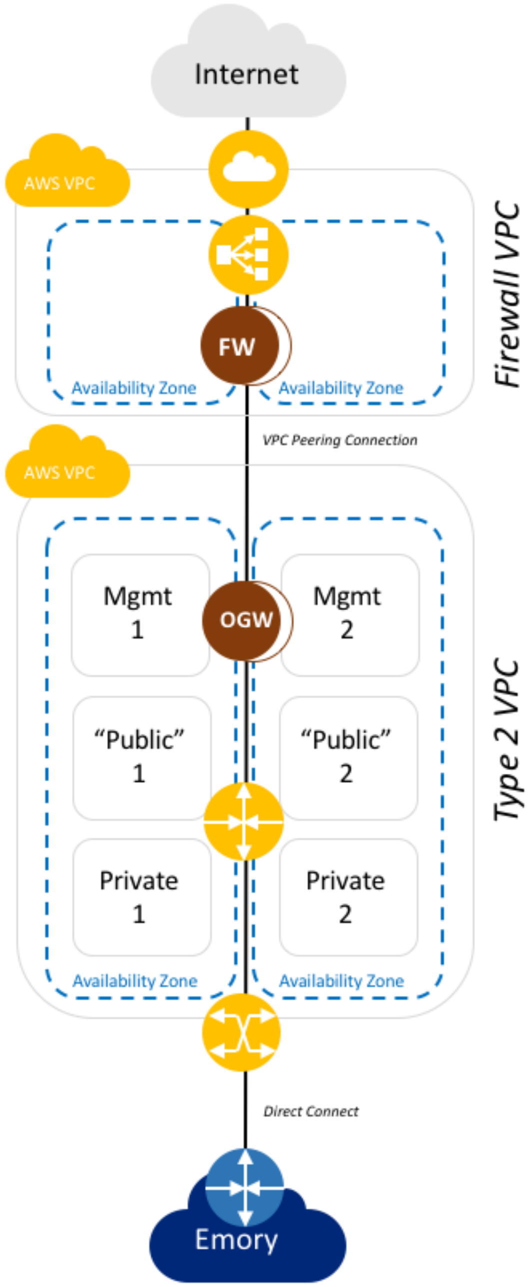

*Figure 2: Emory Type 2 VPC*

The Firewall VPC contains the virtual firewall infrastructure that monitors Internet traffic to and from the Type 2 VPCs. The firewall VPC is managed by LITS and is used by multiple customers. The Type 2 VPC is very similar in structure to the Type 1 VPC. Like a Type 1 VPC, there are two availability zones and six subnets. The main difference between the Type 2 and Type 1 VPC is network traffic flow. Traffic from the Internet bound for Emory applications in the VPC is routed through the Firewall VPC, a Palo Alto firewall, and into Type 2 VPCs. Traffic originating in Type 2 VPCS bound for the Internet will be routed through the Outbound Gateway (OGW) in the Type 2 VPC to the Firewall VPC and then onto the Internet. Traffic destined to Emory will be routed over the direct connection. [4]

With this basic framework in place, LITS created AWS accounts for the research groups participating in the focus groups, implemented a VPC for each with the basic restrictions prescribed by their use case and type, and asked them to perform their work with data that was not sensitive. Although Emory already had significant controls in place for these focus groups, the teams determined not to operate with sensitive data until the design was finalized and the research service was in production.

The focus groups worked for six months to implement their workloads in Emory AWS focus group accounts while LITS and AWS supported them with account administration assistance and, when necessary, solution design. Some of the research groups were already expert in AWS architectures, but others were brand new to AWS. Some broke the site-to-site VPN connections in the first couple of weeks and complained about the inadequate solution for AWS command-line access with federated authentication out of the gate. Others needed help migrating their application on a server in their office to AWS.

While the focus groups worked and reported problems and suggestions for improvement, the LITS and AWS team continued to refine the details of the Type 1 and Type 2 VPCs and perform detailed risk assessments for 37 of the 79 AWS services available in late 2016 through early 2017. They articulated specific risks and controls for each of the initial 37 services that could be implemented via policies or security risk detector commands, triggered upon events occurring in AWS or run on a schedule. After six months of focus group activity and design refinement, the research teams and LITS were comfortable with the high-level design and ready to prepare for an implementation project.

## Automating Account Provisioning and Related Processes

After basic functional and security considerations, the next major design element considered by the team was automation, which directly impacts user adoption. Emory researchers had been creating AWS accounts in three minutes online and immediately making productive use of AWS services. If LITS implemented a service that took days or weeks to deliver a working account and if all of the security controls and restrictions were implemented unreliably or inconsistently, Emory researchers would not adopt the service. Users would prefer to create their own unrestricted accounts, which were more vulnerable. LITS did not believe it was feasible or advisable to prohibit Emory researchers from creating their own accounts, but rather the better approach was to create a service that provided security and compliance value that Emory researchers would acknowledge and willingly select for non-sensitive data as well as sensitive data workloads.

With these considerations in mind, the Emory team set the goal of preserving as much of the AWS user experience as possible. For account provisioning this meant delivering an AWS account and VPC within minutes of the request. This type of automation was a substantial change for Emory. In the past, requests for infrastructure services like virtual machines had been manually provisioned when requested from Emory's customer service portal. Such requests for a virtual machine would take days or weeks. By acknowledging that this approach was not adequate for new cloud-based services, Emory embarked on a path of web service development and service orchestration that was traditionally used in application development, but new to the delivery of infrastructure services. Emory designed a provisioning orchestration that interacts with 8 systems:

1. ServiceNow, Emory's service request system to place VPC requests and where the provisioning process creates incidents in the event of provisioning problems
2. AWS Account Service, a web service at Emory that implements 9 microservices, the overall provisioning orchestration, and the AWS to PeopleSoft financial system billing integration
3. PeopleSoft, Emory's enterprise resource planning system and for the purposes of this orchestration, financial system to validate financial account numbers to which AWS accounts will be charged
4. E-mail validation service, which validates e-mail addresses used in account provisioning
5. CIDR service, which issues network address ranges on Emory's network to the account provisioning process
6. Identity management service, a web service that exposes Emory's NetIQ identity management system to the provisioning process to create roles for each new AWS account and manage user assignments to these roles
7. LDS service, which is a web service that exposes Emory's Lightweight Directory Service to the provisioning process for creating LDS groups that implement the NetIQ roles
8. AT&T NetBond service, a web service that exposes the ability to initiate AWS DirectConnect connections between Emory and new AWS VPCs with AT&T NetBond.

The Emory AWS provisioning process consists of 36 individual steps and multiple interactions with each of these services. A listing of each specific step of the process and an interaction diagram is available on the Emory wiki. [5] The entire provisioning process runs in approximately 15 minutes. Users requesting an AWS account can place a request and be notified when the account is ready 15 minutes later or watch the request processing interface update them on the status of each step of the provisioning process as the work is completed for them.

In order to implement a number of the controls prescribed by Emory policies, the team had to limit or completely remove the ability of AWS users to perform certain functions. For example, one implication of not having AWS internet gateways in Emory Type 1 VPCs and not exposing internet gateways to users to manage in Emory Type 2 VPCs is that Emory AWS users cannot make use of AWS elastic IP addresses. Instead, they must request an Emory network configuration change, specifically called static network address translation or static NAT, to map an available Emory public IP address to the address of the EC2 instance they wish to expose to the internet. Keeping with Emory's design goal of keeping the user experience as similar to AWS as possible, for features like this that are removed from the user's control in AWS, Emory decided to develop a web application to re-implement all of these features in a single place for customer convenience, as a single pane of glass (SPOG) into all of the AWS console features Emory had to withhold from users for security and compliance reasons. This application is called the VPC Provisioning Application or VPCP app. This application contains all of the Emory metadata about Emory AWS accounts, VPCs, roles, users, CIDRs, Emory elastic IP (Static NAT), Emory firewall rule requests, and security risks detected and remediated. The VPCP app interacts with the same 8 systems as the provisioning orchestration, but also network and firewall systems to implement static NAT and request and display current Emory firewall rules in place for each VPC. [6]

## **Training and Mentoring the Emory Implementation Team**

Prior to the inception of this project in 2016, LITS staff had limited knowledge, skill, and even exposure to AWS, primarily because IT operations were limited to on-premises assets and cloud-based software-as-a-service (SaaS) providers. Infrastructure- and platform-as-a-service (IaaS and PaaS) providers had not yet been approved for production use and were actively discouraged by Emory Information Security. They cited numerous breaches of organizations using the Cloud, many of these organizations assets were used to mount attacks against Emory networks. As a result, LITS staff had not learned and experimented with many AWS services and most LITS staff had no direct exposure to AWS at all. IT Architecture worked with AWS extensively for five years, mainly on collaborations with other organizations, open source projects, and DevOps activities at Emory and assumed a leadership role in the design, training, and mentoring of LITS staff.

At the inception of the project, LITS invested $100,000 in on-site AWS training for its staff and also directed the team to online resources and books for training and certification. AWS matched Emory's training investment for a total of $200,000. This initial training was critical in order to build common vocabulary and knowledge with which to have cogent design discussions. Many of the team members could not actively participate in design and planning discussions without a basic knowledge of AWS, its core services, and how it differs from LITS current infrastructure and practices. After this initial training staff were encouraged to pursue further training and conferences and given access to AWS sandbox environments with some specific tasks to perform to build their knowledge and experience.

At the beginning of the implementation project IT Architecture launched a rotation program to bring lead technical staff from systems, middleware, and application development into the IT Architecture group for a three-month, full-time commitment. These rotations consisted of five weeks of intensive AWS training and exercises followed by infrastructure, application development, and testing work. The rotation also included high-level training in other architectures used in the implementation such as Emory's service-oriented and event-driven architectures, service orchestration, and testing. In the first rotation, participants focused on implementing AWS CloudFormation templates and service control policies. They learned Python and the Pytest framework to develop tests to verify the VPC structures, Identity and Access Management (IAM) roles, service controls implemented with service control policies, and security risk detector commands.

In the second IT Architecture rotation, the participants covered all of the same training and background concepts as the first rotation, but focused most of the joint work sessions on developing and testing the automated provisioning, account, and VPC management web services and web applications. Prior to these IT Architecture rotations there were only two or three people in LITS with detailed, in-depth knowledge of the complete research service design and the capability to implement key pieces of that design. The IT Architecture rotation trained six more LITS staff in nearly all details of the design and ensured LITS had triple the bench strength to implement and support the design.

In spite of these staff development efforts, external expertise and staff augmentation were still required to implement all of the work at the brisk pace of the implementation project. Emory invested in AWS enterprise support, which provides dedicated account managers with direct lines to technical resources at AWS. Emory has also cultivated several preferred vendors for mobile, web, and cloud development.

Emory engaged these key partners to accelerate test development and web service development in support of the automation and integration tasks of the project.

## Implementing the Emory AWS Research Service

The project to implement the Emory AWS Research Service began in October 2017 and ran through June 2018. It consisted of 20 subprojects in the areas of network, security, application development, integration, training, and service launch and service management. The total effort of the project was 7,000 hours. The major areas of activity were:

1. Defining, implementing, and testing specific controls for AWS Accounts and VPCs
2. Developing and testing applications, services, and orchestrations
3. Launching the Emory AWS Research Service
4. Supporting Emory researchers with AWS expertise, documentation, and other resources

Nearly every participating group in the team contributed in some way to each of these cross-cutting work streams.

## Defining, Implementing, and Testing Specific Controls for AWS Accounts and VPCs

The analysis, design, and focus group phases of the project defined many of the high-level network access, authentication, and authorization controls Emory needed to meet its security and compliance requirements. However, few of the details were developed and tested in those early stages. During the implementation phase, the Infrastructure, IT Architecture, and Information Security teams took the network and security structures piloted and tested by the focus groups and implemented them with the rigor of software engineering practices. The team refactored the CloudFormation templates and service control policies into the following separate projects that reflect the relationships and dependencies between these structures and policies:

1. Service control policies for Emory AWS Organizations, which restrict the services and features available in accounts within an AWS organization. These controls are over and above any controls implemented within each AWS account. They are an absolute method to deny services or features that should not be available in an account for all roles. At Emory these service control policies differ by compliance class, HIPAA or non-HIPAA.
2. Account-related objects, which are not specific to any VPC in the account and which must persist in the account in the event any VPC is deleted. These include roles for LITS administrators, customer administrators, the policies attached to these roles, CloudTrail audit logging, etc.
3. Emory Type 1 VPC objects, which include the VPC network topology and controls such as subnets, subnet policies, route tables, network access control lists (NACL), elements, etc.
4. Emory Type 2 VPC objects, which are similar to Type 1 above, but include the structures on the customer VPC side to interact with Emory's VPC firewall in the Cloud.
5. Network Structures, Controls, and Firewall Instances for the Emory Firewall VPC, which provides firewall services for Emory Type 2 VPCs

Once these projects were identified, the combined team refactored the proof-of-concept templates and policies into these constituent projects, reconsidered naming of all objects based on this project structure, and implemented structural and functional tests for each project.

Test-driven development was critical to the process of developing and maintaining the CloudFormation templates for VPC structures, policies, and service control policies. First, there are structures and controls implemented at multiple levels. Unit tests for each project and integration or service-level tests were critical to determine if all of the structures and controls actually worked together to achieve the intended results. Second, Emory anticipated the need to update these CloudFormation stacks and policies quickly and frequently once the service was launched. As Information Security detects new vulnerabilities and threats and as AWS launches new services, controls and policies will need to be updated, retested, and rolled into all of the customer accounts in short order. There is no way to accomplish these goals without a robust set of unit and integration tests to verify our changes do not break things and that the code and policies perform as expected.

The practices and tools of test-driven development, continuous integration and continuous delivery were new to many of the Infrastructure team members on the project, and IT Architecture developers focused on these topics early in the IT Architecture rotation. The team created code repositories; build, package, and deploy pipelines; and selected the language in which to develop tests and a test framework. The team select Python for these tests as it seemed more accessible to Emory's systems and security engineers than Java and Ruby. The team selected Pytest as the test framework, and developed an extensive common library of test routines for test setup, teardown, and other common steps for AWS security and authorization tests. The team implemented detailed tracking of the development effort, reporting on how many development and test artifacts were estimated and implement by the number of artifacts and lines of code developed in total and by each team member.

In addition to testing structures and policies for the Emory AWS accounts, Type 1, and Type 2 VPCs, Emory also needed to define and implement tests for each AWS service it enabled in its accounts. The AWS service security risk assessments performed in the analysis, design, and focus group stage identified security risks for each service that Emory Information Security believed required controls or countermeasures of some kind. In the implementation phase, the team had to define and implement specific tests to demonstrate that permitted services operated normally, those that were disallowed did not operate, and those that were allowed with controls or countermeasures performed appropriately in a restricted manner. [7]

This was a significant effort, because at the time AWS had 79 named services the team had to evaluate and implement tests for. They team broke this into 37 services that were required for launch and focused on those first, assigning them out to a broad team for test descriptions. The Infrastructure, IT Architecture, and Information Security team then reviewed these tests descriptions and implemented those that were adequately designed. They returned the descriptions that needed more work back to the test writers. This process continued iteratively for four months until all tests were adequately described and developed. Emory now has a standing function to assess all new AWS services and features for major security vulnerabilities and implement appropriate controls, countermeasures, and tests. Presently Emory manages assessments, test descriptions, and tests for 65 services and adds more each month. Given the AWS rate of innovation, accession of new services, and feature enhancements to existing services, this function will be on-going at Emory and likely never complete. This aspect of Emory's service design has been understandably criticized—perhaps central IT is trying to control too much, and this is an impossible task. This is a valid concern. On the other hand, security risk assessment, controls, and countermeasures is an on-going activity in any computing environment. Thinking this activity is completed or done "good enough" is wishful thinking. By pursuing this strategy Emory committed to continually review and constantly strive to understand AWS services better to help protect Emory and its data. This level of commitment, if properly resourced, will have many benefits for Emory.

The Emory team also needed to provide a mechanism to maintain these CloudFormation stacks and policies with all of the changes and updates that would be required over time to keep all of the AWS accounts and VPCs in sync with the desired structures and policies. The team tested and documented a set of procedures using AWS stack sets to manage stack and policy updates.

**Developing and Testing Applications, Services, and Orchestrations**

The application and integration development effort consisted of 13 subprojects for the following components:

1. AWS Account Service
2. CIDR Service
3. Emory Elastic IP Service
4. AT&T NetBond Service
5. VPCP Web Application
6. Security Risk Detection and Remediation Commands
7. E-mail Address Validation Service
8. Identity Management ESB Service
9. Identity Management NetIQ Development
10. Lightweight Directory Service ESB Service
11. ServiceNow ESB Service
12. ServiceNow Development
13. PeopleSoft ESB Service Enhancements

The AWS Account Service Exposes AWS APIs to the Emory ESB for AccountAlias, AccountOrganizationMembership, CloudFormation Stack, SamlProvider, Peering, and Route objects. These are all operations, which at the time of design and development could not be performed from within CloudFormation templates. Emory's design calls for doing as much account provisioning in AWS CloudFormation templates as possible to support consistent updates and account and VPC maintenance. This service also exposes Emory's metadata store for Account and VirtualPrivateCloud data objects and implements the VirtualPrivateCloud and Account provisioning orchestration.

The CIDR Service Exposes a registry of Classless Inter-Domain Routing (CIDR) ranges for use in various provisioning operations such as VPC and DirectConnect provisioning. At the time of design and development, Emory did not have a single authoritative source for its network ranges that could be exposed with a web service to the orchestration, so this service was created with a database registry of all CIDRs allocated for use with AWS VPCs and VLANs.

The Emory Elastic IP Service exposes a registry of Emory public IPs available for static NAT to internal Emory addresses within AWS VPCs. It can also invoke an operation to perform static NAT on demand.

The AT&T NetBond Service Exposes the AT&T NetBond API for provisioning a VLAN for DirectConnect to Emory's ESB, so that it can be invoked from the AWS Account Service provisioning orchestration and other appropriate contexts. This allows the account provisioning orchestration to add a direct connection between AWS and Emory using AT&T NetBond.

The VPCP web application provides views into account and VPC metadata for customers and LITS administrators and provides a user interface to manage account administrator role membership, Emory elastic IPs, and firewall rules. The VPCP web app also provides a user interface to invoke the VPC and account provisioning process for LITS administrators to develop and test the provisioning process. End users will invoke the provisioning process through a ServiceNow request form that also displays for them the status of this process in a detailed and clean interface.

The Security Risk Detection (SRD) commands implement security vulnerability detection and remediation when triggered by events or on a schedule. These commands run as AWS Lambda functions or in scheduled application contexts and are used to detect and correct security vulnerabilities that cannot be managed by policy alone. For example, at the time of the analysis it was not possible to prohibit the creation of public AWS Simple Storage Service (S3) buckets, so a security risk detection command was implemented to detect public buckets and make them private. This command can be executed as an AWS Lambda function trigged by a configuration event when a bucket is created or run on a schedule to evaluate the status of all buckets. At the time of the analysis the following critical controls were identified that could not be implemented with polices and required SRDs:

1. Unencrypted, secondary EBS volumes. Emory policy requires that these volumes be encrypted
2. Credentials that are not used for a period of time must be removed
3. For HIPAA accounts, forbidden database types (not HIPAA eligible) in the AWS Relational Database Service (RDS) must be removed
4. Unencrypted RDS databases must be encrypted
5. Public and unencrypted S3 buckets must be made private and encrypted

The E-mail Address Validation Service validates whether or not e-mail addresses exist and if e-mail can be delivered to them. This service presently uses the Neverbounce commercial e-mail validation API. The AWS VPC and account provisioning process invokes this service to determine if the next set of distribution lists pre-provisioned by the messaging team actually exist and are working properly. It also calls this API to count the distribution list inventory and alert the messaging team if the inventory is running low by creating a ServiceNow request or incident. This is all necessary, because the messaging team determined that the creation of Office365 distribution lists cannot presently be automated reliably.

The IDM Service exposes the Role and RoleAssignment objects to allow the provisioning orchestration to create IDM roles for new AWS accounts from the provisioning process and add users into those roles. The VPCP web all also interacts with this service to create and update RoleAssignments.

The IDM NetIQ development subproject is additional work within the NetIQ system to expose the Role and RoleAssignment microservices to the provisioning orchestration.

The LDS Service exposes operations to query, create, update, and delete LDS groups to support the provisioning process. LDS groups are required to implement some aspects of role and distribution list membership and must be created in coordination with the NetIQ role creation.

The ServiceNow Service Exposes the ServiceNow Incident object and a couple specific ServiceNow request objects for Emory FirewallRule and ElasticIp. Incident operations can be invoked from the AWS Account Service's provisioning process when it encounters errors and request operations are invoked from the VPCP web app to implement ElasticIp and FirewallRule features.

The ServiceNow internal development effort implements a ServiceNow request form, which will invoke the web service that orchestrates the AWS account and VPC provisioning process. There was also some additional development work required in ServiceNow to expose custom requests to the ESB service.

The PeopleSoft Service needed to add support for an additional object for this project to validate the financial system account number (known as a SpeedType). As a validation operation, this is query only. This is used from applications like the AWS Account Service, VPCP web app, and ServiceNow form to validate financial account number input and also to validate stored financial account numbers on a regular interval as these account numbers expire. This ensures that there is valid Emory billing information for all AWS accounts.

Most of these development efforts could proceed in parallel with different development resources, given the decoupled nature of Emory's service- and event-oriented architecture. Each application or web service could be developed and unit tested using stubs or straw men for services they needed to access. This development effort lasted from December 2017 through March 2018 with considerable integration testing and refinement from April through July 2018.

## Launching the Emory AWS Research Service

The IT Service Management Office coordinates the effort of launching new services for LITS. The service launch for the Emory AWS Research Service involved preparing a detailed service description and identify the appropriate support roles and staff for each aspect of the service. The key roles were frontline support for customers, expert AWS operational support, AWS design and solution consulting, and billing and cost management support.

## Supporting Emory Researchers with AWS Expertise

Frontline and operational support are shared by AWS enterprise support and the LITS helpdesk. Given Emory's investment in AWS "white glove" enterprise support, customers of Emory's AWS research service have direct access to AWS account managers and expertise via phone, chat, or web console. When AWS cannot help as in the case of problems with Emory security controls or countermeasures, users file a ticket with the LITS helpdesk or call the helpdesk support line. These requests for assistance are routed to the LITS Infrastructure team for resolution. LITS infrastructure can draw on further assistance from AWS or Emory IT Architecture.

AWS solution architecture and consulting is provided primarily by the LITS University Information Technology (UIT) unit with resident expertise familiar with AWS and researchers use cases. UIT can draw on resource from AWS and Emory IT Architecture as well.

Billing and cost management is implemented and supported by LITS Finance and Administration. All groups involved in providing AWS support are trained to assess and advise the factors that influence the cost of AWS solutions.

The middleware team in the LITS UIT division are the frontline support for most of the integration and automation work as they operate and administer the Emory ESB and service- and event-oriented integrations. The middleware team can draw upon Emory IT Architecture for further support.

In conclusion, Emory's three-pronged strategy to enable rapid innovation in the Cloud with AWS while maintaining appropriate security and compliance controls has resulted in…[story yet to be written]

S. A. Wheat is with the Libraries and Information Technology Department of Emory University, Atlanta, GA 30322 USA. He is also with IT Architecture for Emory Healthcare, Atlanta, GA 30322 USA and the Atlanta Clinical & Translational Science Institute (e-mail: swheat@emory.edu).

***Keywords*—**Cloud Infrastructure, Amazon Web Services, AWS, Cloud Research Applications, Virtual Private Cloud, VPC, Research Cloud, HIPAA Compliance, Stringent AWS Security Controls

[1] Emory University, Cloud Business Use Cases, https://wiki.service.emory.edu/x/RMO0BQ (Last Updated May 9, 2016)

[2] Emory University, Emory Amazon Web Services Focus Group Documentation, https://wiki.service.emory.edu/x/XZjGBQ (Last Updated September 24, 2017)

[3] Emory University, Emory Amazon Web Services Focus Group Documentation, https://wiki.service.emory.edu/x/XZjGBQ (Last Updated September 24, 2017)

[4] Emory University, Emory Amazon Web Services Focus Group Documentation, https://wiki.service.emory.edu/x/XZjGBQ (Last Updated September 24, 2017)

[5] A complete listing of the steps of the Emory AWS account provisioning process is available in Emory University, AWS Account Service Technical Design Documentation, https://serviceforge.atlassian.net/wiki/x/nwM6 (Last Updated November 14, 2017). Swim lane diagrams for the provisioning process and management applications are in Emory University, AWS Research Service Integration & Application Development Artifact Inventory and Status, https://serviceforge.atlassian.net/wiki/x/AQAfAQ (Last Updated December 5, 2017)

[6] A swim lane diagram for the VPCP application appears in Emory University, AWS Research Service Integration & Application Development Artifact Inventory and Status, https://serviceforge.atlassian.net/wiki/x/AQAfAQ (Last Updated December 5, 2017)

[7] Emory University, AWS Service Inventory and Security Risk Assessment, https://wiki.service.emory.edu/x/CAb5BQ (Last Updated December 6, 2017)

[8] Emory University, AWS Research Service Integration & Application Development Artifact Inventory and Status, https://serviceforge.atlassian.net/wiki/x/AQAfAQ (Last Updated December 5, 2017)