Artifact: IAM Approach

Overview

Current IAM Approach for Current On-Premise Model

Processes and Policies

Note: Documents and pdfs of wiki pages have been attached because they were unaccessible to some users due to their location in the wiki.

Overview

ESD Integration Review Team Presentation March 2017 - LINK (Attachment)

IDM Integration Review Team Presentation March 2017 - LINK (Attachment)

Account Creation

Emory Account Lifecycle for Staff, Faculty, Students, and Sponsored Accounts - LINK (Attachment)

Creating/Deleting a Sponsored Account - LINK

Listing and Renewing your Sponsored Accounts - LINK

Account Disable and De-provisioning

IDM Process - Mass EUV Delete - MyNetID - LINK (Attachment)

IDM Process - EUV Account Disable - MyNetID - LINK (Attachment)

IDM Process - EUV Account Delete - MyNetID - LINK (Attachment)

Authorization Approach

Authorization is handled by the application. Shibboleth releases the attributes to a Service Provider required to make an authorization decision. Some applications interface with LDS or AD directly and make authorization decisions based upon the values of specific attributes. ESD views can be used for authorization decisions. Membership in MyNetID roles are used by the VPCP application.

Correlating Multiple User Accounts

University and Healthcare accounts are linked by public person identifier.

Alternate Tokens

MFA/2FA

Duo

Federation Approach with External Systems and SSO

The Emory Shibboleth IDP is used to federate with external Service Providers. Microsoft LDS is the identity store used by Shibboleth for authentication and attribute resolution. ADFS and AD are used for Microsoft specific applications.

User, Group, Role, and Policy Management

Users

Emory User Accounts

Emory user accounts are managed through normal Emory account lifecycle. Refer to the Account Creation section of this document.

AWS user accounts

It is preferred to not use local AWS IAM accounts. Users should use their NetID for federated access to the AWS Console and CLI. A user will assume a role within AWS during authentication. With that said, there may be an occasional need for an AWS user accounts. These requests will be evaluated on a case by case basis and will be approved by Security.

There is a need to create new processes for logging these accounts, credential rotation, key rotation, and to determine if these accounts expire.

Groups

LDS Groups

Group membership is managed by MyNetID using the role and resource model. The MyNetID web application and the ESB IDMService are used to add and remove users from roles. Removal from a role removes the user from the associated resource and entitlement(LDS group). Users are not added/removed from an LDS group directly. They are added/removed from the role that provisions the LDS group. More details can be found in the "Emory Enterprise IdM" section of this document.

AD Groups

Group membership is managed directly with Active Directory Users and Computers (ADUC) and with MyNetID using the role and resource model. The MyNetID web application and the ESB IDMService are used to add and remove users from roles. Removal from a role removes the user from the associated resource and entitlement(AD group). Within MyNetID, users are not added/removed from an AD group directly. They are added/removed from the role that provisions the AD group. More details can be found in the "Emory Enterprise IdM" section of this document.

AWS Groups

AWS Group membership can be maintained using the AWS console, CLI, and SDKs. AWS groups will generally not be used because AWS groups only work with AWS IAM accounts. They do not work with federated account.

Role

MyNetID Roles

Role Creation

MyNetID roles are managed using MyNetID. New roles are requested by issuing a ServiceNow requests. The requests will be processed by the IDM Team. More details can be found in the "Emory Enterprise IdM" section of this document.

Role Membership

Users are add/removed from a role using the MyNetID web application or the ESB IDMService. Membership can be maintained automatically based on a predefined set of attributes values or a role manager can manually maintain membership. More details can be found in the "Emory Enterprise IdM" section of this document.

AWS Roles

AWS Roles are created by Cloud Formation Templates. These are the roles that a federated user may assume. The AWS roles need exactly match the AWS Role ARN released by the Shibboleth IDP.

Policy Management

Policies are listed in the "Current IAM Approach for Current On-Premise Model" section of this document.

Account Capabilities

Account Type	MFA	AWS Console	AWS CLI /API	Custom Application	RDP	SSH

AWS IAM Accounts (password)	Yes	Yes	No	No	No	No
AWS IAM Accounts (keys)	Yes	No	Yes	No	No	No
EC2 Key Pairs	No	No	No	No	Yes	Yes
Emory University NetID - Federated Accounts (Shibboleth IDP)	Yes	Yes	Yes(Using TKI Service or another custom solution)	Yes	No	No
Emory University NetID - University AD /LDS Accounts	Yes (requires RDP and SSH configuration on the instances)	No	No	Yes	Yes	Yes
Federated External Accounts	N/A	N/A	N/A	N/A	N/A	N/A
Application-specific Accounts	Yes (if application supports it)	No	No	Yes	No	No

Authentication

AWS IAM User Accounts

AWS IAM User Account Settings

- SMX Recommendation
    - AWS Console Access
        - Password rotated at an interval.
    - Programmatic Access not available
    - MFA if possible and makes sense.
    - Alert on any sign-ins to this IAM User account.
- Emory Approved (TODO)
    - This is still to be determined and will be revisited.

Account Management Policy

IAM approach from AWS@Emory Project (CIMP-702)

AWS SCP Policies

emory-aws-org-hosting-standard-scp.json - Cloudformation template used to create Horizontal IAM resources within AWS accounts - LINK

emory-aws-org-hosting-enhancedsecurity-scp.
json - Cloudformation template used to create Horizontal IAM resources within AWS accounts - LINK

emory-aws-org-hosting-hipaa-scp.json - Cloudformation template used to create Horizontal IAM resources within AWS accounts - LINK

AWS IAM Policies

emory-aws-hosting-account-cfn.json - Cloudformation template used to create Horizontal IAM resources within AWS accounts - LINK

emory-aws-hosting-account-vertical-roles.
json - Cloudformation template used to create Vertical IAM resources within AWS accounts - LINK

AWS IAM User Management

AWS Root User Account Escrow

The initial root password is unknowable, undisclosed, and is guaranteed by Amazon to be at least 64 characters (or greater . .. smartly, they do not disclose the length). Paul Petersen, John Connerat, and James Reed were able to create and test global transport rules in Exchange to redirect password reset links to a non-publicized Exchange mailbox that is only accessible to a few people.

1. Resetting all the aforementioned root password to a highly complex, extremely long password that exceeds Amazon's complexity requirements. The exact length will vary, but given that Amazon allows up to 128 complex characters, the lengths will extremely long.
2. Adding MFA to these accounts (TOTP standard).
3. Storing these credentials and tokens within LastPas s Enterprise accounts for this specific purpose.
4. Printing out the credentials and MFA tokens and QR codes (on a **non**-networked, **non**-BizHub-with-disk-drive) printer for Derek to store on paper in the LITS Security safe in case Paul and John somehow loose access or are unavailable.

IAM User Account Suspension

Events

- Unauthorized/abnormal sign-in.
- Using a resource for a purpose other than it's original function.

Actions

In the event that an IAM User account should be suspended, the actions that should be taken against the resource should be the following:

- Remove all IAM permissions

MFA Support

Root AWS IAM Users

Printing out the credentials and MFA tokens and QR codes (on a **non**-networked, **non**-BizHub-with-disk-drive) printer for Derek to store on paper in the LITS Security safe in case Paul and John somehow loose access or are unavailable.

AWS IAM User Accounts

MFA must be enabled on all IAM User accounts that requires AWS Console access.

Any of the following options, with the exception of SMS, should be used:

https://aws.amazon.com/iam/details/mfa/

Emory University AD Accounts

Account Settings

Local Accounts

- Minimal password requirements are enforced (8 characters long, 24 passwords remembered).
- Lockout policy: 50 invalid logins over a 30 minute period causes a 10 minute lockout (50/30/10).

LITS Centrally Managed Accounts

- Passwords are stored within AD but password complexity rules and expiration are enforced by University IDM (MyNetID). IDM enforces the Enterprise Password Policy.
- Lockout policy: 50 invalid logins over a 30 minute period causes a 10 minute lockout (50/30/10).

User Management

Account provisioning/de-provisioning

Local Accounts

Local accounts are created in OUs managed by local IT professionals.

LITS Centrally Managed Accounts

Accounts are provisioned by the University IDM system (MyNetID) according to the University account lifecycle policy. This policy can be viewed in the "Current IAM Approach for Current On-Premise Model" section of this artifact

Account Suspension

Local Accounts

Controlled by local IT professionals.

LITS Centrally Managed Accounts

Handled according to University account lifecycle policy. This policy is listed in the "Current IAM Approach for Current On-Premise Model" section of this artifact.

MFA Support

There is no MFA support with direct AD authentication. AD authentication is only the authentication portion of a two factor authentication.

Account Access Auditing

University AD logs are available in Splunk, but are not proactively monitored. Logs are analyzed when an issue arises.

Emory University LDS Accounts

Account Settings

Local Accounts

There are no local accounts in LDS.

LITS Centrally Managed Accounts

- Passwords are not stored in LDS. Passwords are proxied to University AD. Passwords policies are enforced by the underlying University AD.
- Lockout policy is dependent upon the underlying University AD.

User Management

Local accounts do not exist in LDS.

Account provisioning/de-provisioning

Local Accounts

N/A

LITS Centrally Managed Accounts

Accounts are provisioned by the University IDM system (MyNetID) according to the University account lifecycle policy. This policy can be viewed in the "Current IAM Approach for Current On-Premise Model" section of this artifact

Account Suspension

Local Accounts

N/A

LITS Centrally Managed Accounts

Handled according to University account lifecycle policy. This policy is listed in the "Current IAM Approach for Current On-Premise Model" section of this artifact.

MFA Support

There is no MFA support with direct LDS authentication. LDS proxies the authentication request to University AD.

Account Access Auditing

Logs are available in Emory central logging. Authentication attempts can be found in Splunk because they are proxied to AD. Logs are not proactively monitored, but are analyzed when an issue arises.

Federated Accounts

Shibboleth Internal Federated Accounts

LDS is used as the datastore for Shibboleth. Refer to the "Emory University LDS Accounts" section for more information on LDS accounts.

Account Settings

- AWS Account Shibboleth Configuration - LINK (Attachment)
- Passwords are not stored in LDS (Shibboleth's datastore). Passwords are proxied to University AD.
- Lockout policy is dependent upon the underlying University AD.

User Management

Account provisioning/de-provisioning

Handled according to University account lifecycle policy. This policy is listed in the "Current IAM Approach for Current On-Premise Model" section of this artifact.

Account Suspension

Handled according to University account lifecycle policy. This policy is listed in the "Current IAM Approach for Current On-Premise Model" section of this artifact.

MFA Support

DUO 2FA is enabled on a per Service Provider basis through the Shibboleth IDP. DUO is enabled for federated access to the AWS console. If using the TKI Service from AWS@Emory or similar software, AWS CLI access is also protected by DUO

Account Access Auditing

Shibboleth logs are available in Emory central logging and on each IDP Server. Logs are not proactively monitored, but are analyzed when an issue arises.

Federated External Accounts

Federating with other Universities or Federations is not currently implemented.

Account Settings

N/A

User Management

Account provisioning/de-provisioning

N/A

Account Suspension

N/A

MFA Support

N/A

Account Access Auditing

N/A

Application-Specific Accounts

Identified Applications

- Bitbucket
- Clinical Trials - Local only. All the documentation in the CMDB and elsewhere on the wiki indicates that there is no authentication beyond local accounts. Check with Rohith to confirm.

Noted Applications

- Confluence (Wiki) - Shibboleth
- Web Hosting - Shibboleth
- Shibboleth - N/A since it is an authentication method itself
- OnBase - Shibboleth
- Peoplesoft ELM - LDAP/LDS

Guidelines/Process

Application local accounts must meet Emory's password policy requirements and the app owner is 100% responsible for ensuring that accounts are removed on a timely basis when people leave.

X.509 Certificate Validation

Currently we are not using any Smartcard implementation. In the future, if we implement this, we will need to revisit this section.

Specific teams are currently wrapping SSH keys with X.509 certificates (which is tied to local accounts). Since we are looking to authenticate using LDS/AD, this method of authentication should not be utilized in the cloud.

Authorization

AWS Console

Federated Account

Authorization is controlled by LDS group membership. The Shibboleth IDP releases a list of AWS role ARNS that a user can assume based on the user's membership in the appropriate LDS groups.

AWS IAM User

Authorization is controlled by the AWS IAM group in which the user is a member and the AWS IAM policy assigned to that group.

Federation-Aware Applications

An authorization decision is made by evaluating the attribute values released by the Shibboleth IDP. Each federated application (SP) can request a set of attributes from the available attributes provided by the Shibboleth IDP.

ESB IDMService and MyNetID Web Service

Authorization decisions can be made based on role membership within the University IDM system(MyNetID). Role membership can be evaluated using the ESB IDM Service or the MyNetID web service calls.

Non-Federation-Aware Applications

Authorization decisions can be made based on attribute values from an accessible datastore. The datastore may be LDS, AD, Emory Shared Data(ESD) Views, MyNetID roles, custom database, etc.

Virtual Machine Console Access

Windows

Authorization is controlled using Active Directory groups and users. The UTS_Windows security group, which contains all members of the Windows Systems Team, is automatically added as an administrator to each server through GPOs assigned to the UTS Servers OU. Users and/or groups are then manually added as administrators to the server by the

Windows System Team. Administrative access is processed by ServiceNow requests made by the service owner.

Linux

Authorization will be controlled by membership in LDS groups. LDS group membership will be maintained using the MyNetID web application.

IAM Components

Documented below

Components Involved

Identity Providers / Asserting Parties

Applications/Service Providers /Relying Parties

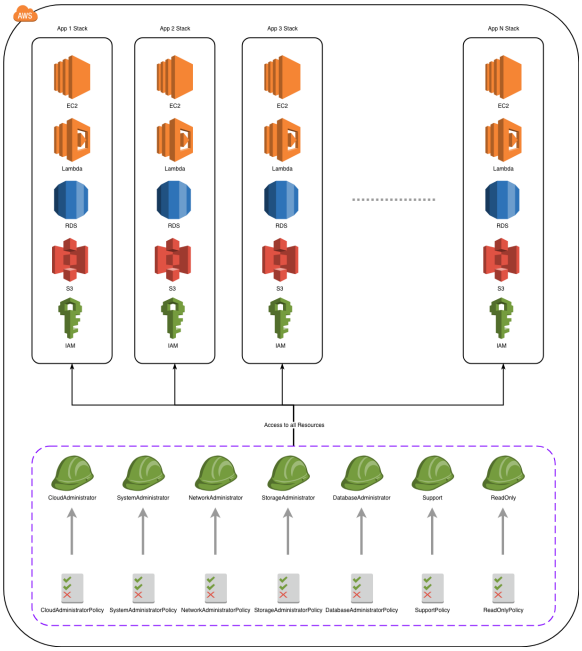
Support

IAM Structures

Horizontal Structure

Everyone, Application and LITS individuals, uses a set of defined IAM Roles to access all AWS resources within an AWS account. Similar to what is done on-premise.

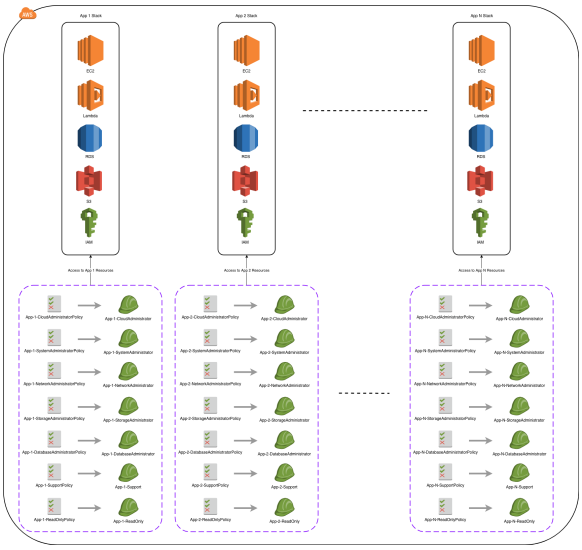
- Number of IAM Roles are still TBD
- Names of IAM Roles are still TBD
- The IAM Policy for each IAM Role is still TBD
- Assumptions in the graph below:
    - The number of applications per AWS account is still TBD.
    - Each application stack can vary, meaning that an application stack can contain none, one, or more of the AWS services listed.
    - The purple boxes encapsulates any IAM Roles that is created at a specific time
        - Horizontal, during AWS account creation
        - Vertical, during application stack creation
    - Each IAM Policy is only attached to its corresponding IAM Role. (One-to-one relationship)
    - Individuals in different teams/groups can be assigned to multiple IAM Roles. (Many-to-many)



Vertical Structure

Everyone, Application and LITS individuals, uses application specific IAM Roles to access all AWS resources within an AWS account. Each IAM Role allows access to its assigned application stack of AWS resources.

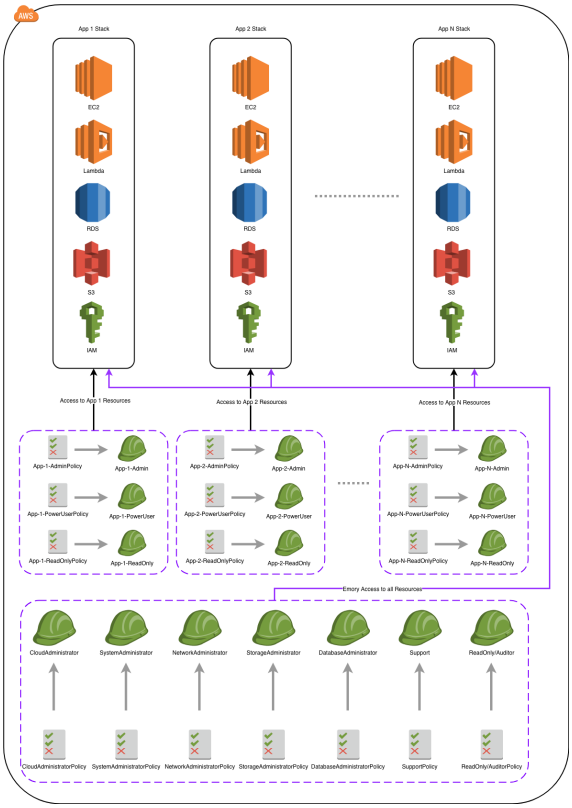
- Number of IAM Roles are still TBD
- Names of IAM Roles are still TBD
- The IAM Policy for each IAM Role is still TBD
- Assumptions in the graph below:
    - The number of applications per AWS account is still TBD.
    - Each application stack can vary, meaning that an application stack can contain none, one, or more of the AWS services listed.
    - The purple boxes encapsulates any IAM Roles that is created at a specific time
        - Horizontal, during AWS account creation
        - Vertical, during application stack creation
    - Each IAM Policy is only attached to its corresponding IAM Role. (One-to-one relationship)
    - Individuals in different teams/groups can be assigned to multiple IAM Roles. (Many-to-many)



Mixed Structure

Application individuals will use application specific IAM Roles (Vertical) to access their specific AWS resources. LITS individuals will use AWS Account defined IAM Roles (Horizontal) to access all AWS resources within an AWS account.

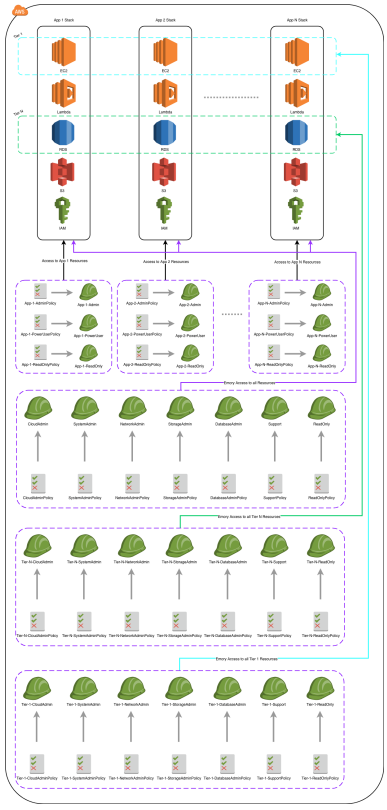
- Number of IAM Roles are still TBD
- Names of IAM Roles are still TBD
- The location of each IAM Role, Horizontal or Vertical, is still TBD
- The IAM Policy for each IAM Role is still TBD
- Assumptions in the graph below:
    - The number of applications per AWS account is still TBD.
    - Each application stack can vary, meaning that an application stack can contain none, one, or more of the AWS services listed.
    - The purple boxes encapsulates any IAM Roles that is created at a specific time
        - Horizontal, during AWS account creation
        - Vertical, during application stack creation
    - Each IAM Policy is only attached to its corresponding IAM Role. (One-to-one relationship)
    - Individuals in different teams/groups can be assigned to multiple IAM Roles. (Many-to-many)



Mixed-Complex Structure

Similar to Mix Structure, however with added horizontal roles to cover different tier levels within application stacks.

- Number of IAM Roles are still TBD
- Names of IAM Roles are still TBD
- The IAM Policy for each IAM Role is still TBD
- Tier levels, tier names, tier definitions are still TBD
- A single tier covers a set of TBD resources within all application stacks within the AWS account
- Assumptions in the graph below:
    - The number of applications per AWS account is still TBD.
    - Each application stack can vary, meaning that an application stack can contain none, one, or more of the AWS services listed.
    - The purple boxes encapsulates any IAM Roles that is created at a specific time
        - Horizontal, during AWS account creation
        - Vertical, during application stack creation
    - Each IAM Policy is only attached to its corresponding IAM Role. (One-to-one relationship)
    - Individuals in different teams/groups can be assigned to multiple IAM Roles. (Many-to-many)
    - Sets of horizontal IAM Roles that are assigned a tier are assigned to a single tier in the diagram. This does NOT mean that a set of horizontal IAM Roles cannot be assigned to multiple tiers, this is still TBD.



Comparison Table

ID	Characteristic	Horizontal	Vertical	Mixed	Mixed-Complex
ID	Characteristic	Horizontal	Vertical	Mixed	Mixed-Complex
1	Number of provisioned IAM Roles per AWS Account	Minimal	Estimated 90% of Mixed-Complex value	Estimated 75% of Verticals value	Maximum
2	Amount of IAM Role deployments since AWS account creation	One	One less than Mixed	Estimated 90% of Mixed-Complex value	Maximum
3	Potential to follow security best practice of "Least Permissive"	Minimal	Maximum	High, horizontal roles are created that allows access to all resources	Higher, more defined horizontal roles created than "Mixed"

4	Possible levels of granularity in IAM Policies	Minimal, since users from different applications will be sharing IAM Roles and IAM cannot dynamically determine which app user has access to	Maximum, users will be assigned to application specific IAM Roles	High	Higher
5	AWS Network (VPN) Segmentation (Specifying access to specific Instances, Subnets, VPC, AWS Environment)	Andy-TBD	Andy-TBD	Andy-TBD	Andy-TBD
6	AWS Network (VPN) Accessibility for IAM Roles	Andy-TBD	Andy-TBD	Andy-TBD	Andy-TBD
7	Number of LDS Groups per AWS Account	Minimal	Estimated 90% of Mixed-Complex value	Estimated 75% of Verticals value	Maximum

User Access to IAM Resources

NOTE: This effort may change in the future due to specific needs!

NOTE: This is currently the initial pass and there is current efforts to research each IAM Role to determine if changes are needed. The page(s) that reflects the latest changes are here:

CIMP-205 - Define Emory Roles and Conceptual IAM Policies

Vertical IAM Roles

- **ApplicationAdmin**
  - Full access to application resources
  - LITS managed resources are blocked
  - Network Connectivity
  - Tasks

        - Backups
        - Monitoring
        - Patching
        - Building/deployment
        - Troubleshooting/Support
        - Maintain Golden AMIs
        - S3 ? - for an AWS equivalent for today's on-prem NAS
        - import InCommon generated cert for POC
        - manage certificates on Load Balancer
        - Manage application SG for EC2 internal to VPC
    - Permissions
        - Compute (EC2, Elastic Beanstalk)
        - Load Balancer
        - ACM
        - ~~Ansible (whatever permissions required for this)~~
        - S3
        - EFS
        - Glacier
        - CloudSearch
        - CloudTrail
        - CloudWatch
        - CloudFormation
        - ElastiCache
        - Launch Templates
        - Auto Scaling Groups (EC2)
        - Launch Configurations
        - Target Groups (EC2)
        - Lambda
        - Security Groups
        - Snapshots
        - Volumes
        - AMIs (**create, delete**)
        - Certificate Manager
        - IAM
    - Questions/Conflicts
        - Compliance ?
        - Will Route 53 continue to be handled just by networking team? At some point as we migrate existing websites to AWS, there could be quite a few requests that could be automated if this role had access - or maybe DNS is staying on prem?
- **ApplicationOperator**
    - Network Connectivity
    - Permissions
        - Read only access to application resources
            - Database - RDS: view only for:
                - Connectivity
                - Monitoring
                - Configuration
            - Maintenance & backups — ? could we possibly do this (in the future) as a self-service function (snapshots)
        - Partial EC2 (create, delete, stop, start, restart.)
    - LITS managed resources are blocked
    - Application Operator:
        - System developer
        - Network Operations Center (NOC /TOC)
    - Questions/Conflicts
- **ApplicationReadOnly**
    - Read only access to application resources

Horizontal IAM Roles

- **Link: https://bitbucket.org/itarch/emory-aws-hosting-account-cfn/src/master/emory-aws-hosting-account-cfn.json**

- **EmoryCloudAdministratorRole** (Point of Contact: Paul Petersen)
    - Permissions
        - Full access
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
    - Tasks
        - VPC/Subnet Provisioning
        - VPN Provisioning
        - Routing Table Provisioning
    - Question/Conflict

- **EmorySystemAdministratorRole** (Point of Contact: Derek Cox/Steve S.)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - Lambda
        - CloudWatch
        - SSM
        - CloudTrail
        - Inspector
        - Config
        - S3
    - Tasks
        - Backups
        - Monitoring
        - Patching
        - Building/deployment
        - Troubleshooting/Support
        - Compliance
        - Maintain Golden AMIs
        - EBS administration
    - Question/Conflict
        - SecurityGroup management? Systems or App team (whoever manages the box)
- **EmoryNetworkAdministratorRole** (Point of Contact: Alex Berry)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - DirectConnect
        - EC2
        - Route53

        - RDS
    - Tasks
        - Troubleshoot EC2/RDS connectivity
        - AWS DirectConnect troubleshooting
    - Question/Conflict
        - NACLs and SGs management by Network or Security? SG (look at systems)/ Depends on where the specific blocks are going to be placed (firewall or NACLs). But give permissions initially until decided. Paul mentioned that it would most likely be on the firewall, however we are still waiting for SMX Brad to present his latest regarding the NACLs.

- **EmoryDataProtectionAdministratorRole** (Point of Contact: Sergey)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - EC2
        - RDS
        - SSM
        - KMS - formerly part of EncryptionAdmin role
        - EFS
    - Tasks
        - Backups of storage
        - Backups of servers
    - Question/Conflict

- **EmoryFirewallAdministratorRole** (Point of Contact: Andy Efting)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - EC2
        - ACM - formerly part of EncryptionAdmin role
        - CloudTrail
        - CloudWatch
    - Tasks
        - Firewall management (Palo Alto)
        - Create Palo Alto firewalls in EC2
        - Create VPN tunnels
        - Application/Network Load Balancer management
        - View logs
        - NACLs
    - Question/Conflict
        - Certificate management? Application owner
        - Transit Gateway? Networking

        - Route Table management? Networking
        - Security Groups? Systems/App teams
- **EmoryIRAdministratorRole** (Point of Contact: Derek Spransy/Zachery Cox)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - EC2
        - CloudTrail
        - CloudWatch
        - Config
        - GuardDuty
    - Tasks
        - Updating Nessus scanners
        - View logs
    - Question/Conflict
        - Compromised instances? SecOps automation to place in Quarantine account? In a different SecOps role for automation.
- **EmoryDatabaseAdministratorRole** (Point of Contact: Ramya)
    - Network Access
    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
        - EC2
            - Conditional: Instance must be tagged a specific way to allow permissions
        - RDS
            - Conditional: Instance must be tagged a specific way to allow permissions
        - KMS
    - Tasks
        - Performance Tuning
        - Patching
        - Upgrading
        - Cost Optimization
        - Migrations
        - Monitoring
        - Backups
        - Cloning
        - Data Refresh in different environments
        - Scaling
        - Manage DB AMIs
        - Designing solutions
        - Advisory
        - Launching DBs
        - Support Auditing
        - Redshift Admin
        - Dynamo DB admin
        - EBS Snapshots

        - Question/Conflict

- **EmorySupportRole** (Point of Contact: TBD) *See Next Section*

    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
    - Tasks
    - Question/Conflict
        - Partial EC2 permissions Currently no need/requirement for this IAM Role and this permission

- **EmoryReadOnlyRole** (Point of Contact: TBD)

    - Permissions
        - ReadOnly
        - Deny:
            - Logging modifications (S3, Glacier, and CloudWatch)
            - Modification of Horizontal Roles (IAM)
            - Non-US regions
    - Tasks
        - Auditing
    - Question/Conflict
- SRD/SRR
    - An additional layer of security for AWS accounts to detect and remediate security /configuration issues.
    - IAM Role
- Nessus Scanner
    - Elliot Kendall has a custom script to pull all of the EC2 instances from the member AWS accounts from the AWS Organization to add to Nessus.
        - If this would need to be a part of the SRD/SRR application, we are open for another team developing /maintaining it.
    - IAM Role
    - https://svn.service.emory.edu:8443/repos/emoryoit/project/emory-aws-arp/
- AwsAccountService(?)
    - AWS metadata store from AWS at Emory project
    - IAM Role
    - Still undecided

Support Role

Research Need for a "Support" Role (CIMP-884)

Description:

Need to research if the Support IAM Role is needed within the AWS environment. If so, we need to determine the tasks this IAM Role will perform. Also need to identify/determine which group/team will be using this IAM Role (thinking of a discussion with John Ellis).

Acceptance:

Create wiki page that contains information regarding what information was found, what team will be consuming this role, if this role is needed, and tasks that this IAM Role will require.

Research:

After a quick email exchange with John Ellis it seems there will likely be a need for the "Support" IAM role that will be used by the Service Desk or, more likely, the NOC. Here is his response to my question:

> - Probably should have a sprint on just this question! My understanding from Wayne is that support for AWS will be handed off to the NOC – depending on what we provide, AWS access could be required, or they could do what is needed through an app like the VPCP app, or ServiceNow (via integration). We should certainly start including Service Desk and NOC resources in the discussions. -John Ellis 12/18/2018

Recommendation:

Given the strong likelihood of the Service Desk and/or NOC providing support either for direct customer contact or monitoring alerts, the recommendation is that we create a Support role called "EmorySupportRole". Since it seems like it will be awhile before the exact details of who will be in this role and what this role be doing, the recommendation is to initially give this role ReadOnly access. When a specific use case is known we will look at whether this policy needs to be adjusted.

IAM Users

- Ansible Tower
  - EC2 instance configuration and deployment tool
  - IAM User
- Shibboleth
  - There is a risk identified that if Shib is down, users will not be able to sign in to the AWS environment. In this case, we would want an IAM User to be created to be able to access the AWS account it resides within to debug.
  - IAM User
  - Still undecided

Emory Enterprise IdM

Determine requirements for new system connection to IDM automation process (CIMP-202)

Overview

Emory Login (ShibbolethIDP) and SAML 2.0 will be used to facilitate federated login to the Emory University AWS Accounts. Users will access the AWS Management Console and AWS CLI using their University credentials. Access is granted by the user authenticating to the University IDP and then "assuming a role" within AWS. The current roles a user might assume are on this page and are listed below:

- Cloud Admin

- Systems Admin
- Network Admin
- Storage Admin
- Security Admin
- DataBase Admin
- Support Staff
- Read Only/Manager

The infrastructure to support assuming the above roles will be MyNetid (NetIQ IDM), Microsoft LDS (LDAP), and Emory Login (Shibboleth) . The MyNetID role and resource model will be used to provision LDS groups. The membership of these LDS will be used by Shibboleth to generate AWS ARNs (Amazon Resource Names) and to facilitate federated login to the AWS console. MyNetID provides a web application and web services to control membership in these role. The MyNetID roles can also be used to provision users to a variety of resources that provide various functionality. Some of the possible functions are:

- Elevated VPN access for members
- Provisioned to a distribution list use to communicate with all users who have AWS account access
- Control access to AWS section of Confluence

The AWS@Emory project is currently using this design.

Definitions

Single Sign On (SSO) - Using the same credentials to access multiple applications. In some cases, the credentials need to only be provided once per session.

Federation - Using enterprise credential to access resources outside the enterprise. Example, using an Emory University NetID and password to access Lynda.com

Reference

CIMP-113 & CIMP-16 Draft definition of IAM Users, Groups, Roles and Policies (including cross-account roles) - Draft

CIMP-357 and CIMP-15 AWS Organizationals Primer and Service Overview (outline and rough draft)

AWS IAM Best Practices

Identity Federation in the AWS Cloud

About SAML 2.0-based Federation

EPIC: Future Requirements

Configuration

Access to each AWS account will be federated so that users can use their current University credentials. There are three components that need to be configured are:

1. Microsoft LDS Groups
2. MyNetID Roles and Resources
3. Emory Login (Shibboleth)

Microsoft LDS Groups

Description

NOTE: The current plan is to create a new OU that will contain all of the CIMP related AWS Account IDs. This is not reflected in the screenshots below.

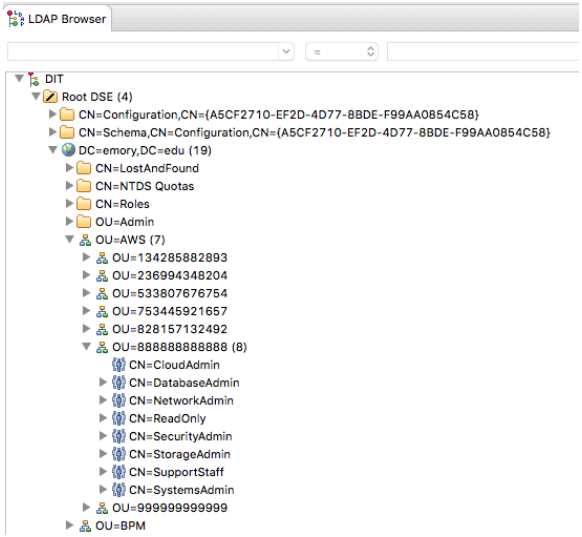
An OU gets created for each AWS Account. Under each OU, a group will be created for every AWS role that will be assumed. LDS Groups can be created by the Messaging Team, ESB, or the IDM team. The IDM team has various web applications and automation tools that can are used to create the AWS groups and OUs.

Steps

1. Create OU (ex. ou=888888888888)
2. Create Groups (ex. cn=CloudAdmin, cn=SystemsAdmin, etc.)

Screenshots

ApacheDirectortyStudio (LDAP Browser) showing all of the LDS groups created for the AWS Account : 888888888888



MyNetID Roles and Resources

Description

MyNetID is used to create the the roles and resources that allow the management of user membership in the various AWS roles. The MyNetID web application allows administrators to add/remove users from these various roles.

The MyNetID roles and resources can be created using using the MyNetID web application, the IDM teams's automation tools, or ESB messages.

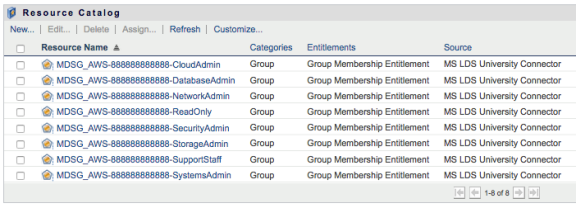
Steps

The MyNetID steps in the creating roles and resources are the following:

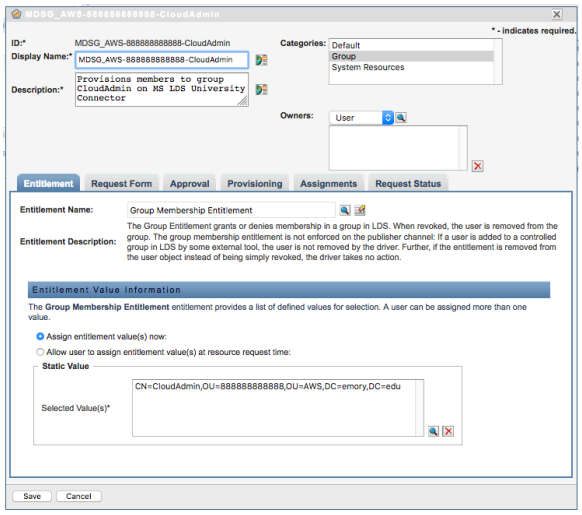
1. Create Resource
2. Add Entitlement to Resource
3. Create Role
4. Create Role Association

Screenshots

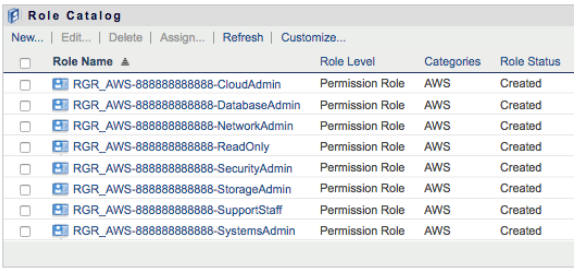
Screenshot from the MyNetID web application showing step #1. This shows all of the resources created to support AWS Account : 888888888888.



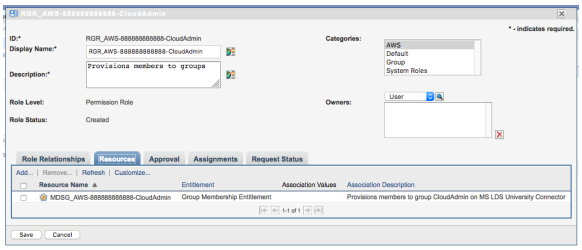
Screenshot from the MyNetID web application showing step #2. The LDS group (cn=CloudAdmin) is being added to the resource called MDSG_AWS-888888888888-CloudAdmin. There is a one to one relationship between entitlements (in this case, an LDAP group) and resources.



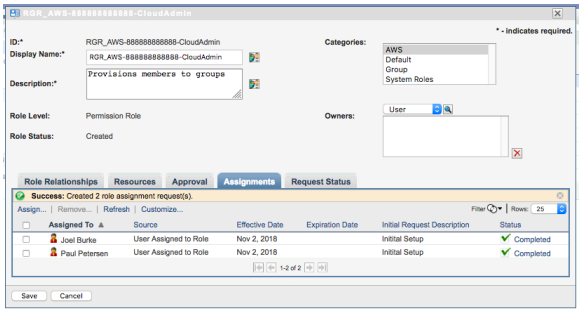
Screenshot from the MyNetID web application showing step #3. This shows all of the roles needed to support AWS Account : 888888888888.



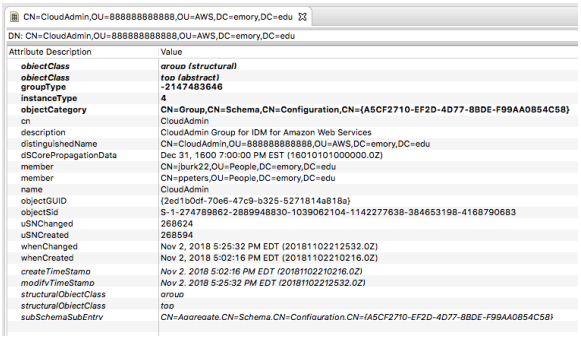
Screenshot from the MyNetID web application showing step #4. This shows how a resource is associated with a role. Any user who is added to this role will be provisioned the associated resource.



After the above steps have been completed, users can be added and removed from the MyNetIDroles. Below, shows how this is done using the MyNetID web application. This can also be done using web services.

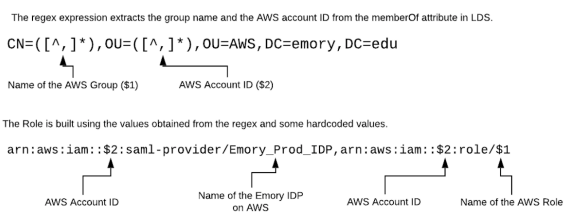


The end result of adding a user to the role is that the user gets added to the LDS group.



Emory Login (Shibboleth)

Shibboleth parses the memberOf attribute in LDS and generates a list of AWS Role ARNS for each user. These ARNS are then released to the AWS Account console service provider. The generated ARNS are the AWS roles a user can assume. The graphic below describes what makes up an ARN:



The following images illustrate the authentication process to an AWS console.

1. User attempt to go directly to the AWS Console. Since the user has not authenticated to an SSO application, the user is redirected to the Emory IDP.

The user authenticates and a SAML assertion is generated and sent to the AWS console. Note the role attribute in the assertion below. These are the roles the user is allowed to assume.

SAML

```
<? xml version = "1.0" encoding = "UTF-8" ?>
< saml2p:Response Destination = "https://signin.aws.amazon.com/saml"
     ID = "_1635fac898920bb326da9d9dafc0e5a8" IssueInstant = "2018-11-02T21:34:31.758Z" Version = "2.0"
     xmlns:saml2p = "urn:oasis:names:tc:SAML:2.0:protocol" >
     < saml2:Issuer xmlns:saml2 = "urn:oasis:names:tc:SAML:2.0:assertion" >https://login.emory.edu:4444/idp/shibboleth</ saml2:Issuer >
     < saml2p:Status >< saml2p:StatusCode Value = "urn:oasis:names:tc:SAML:2.0:status:Success" /></ saml2p:Status >
     < saml2:Assertion ID = "_e409c4920e67c7cae7a4609e1b7d80cc" IssueInstant = "2018-11-02T21:34:31.758Z"
         Version = "2.0" xmlns:saml2 = "urn:oasis:names:tc:SAML:2.0:assertion"
         xmlns:xsd = "http://www.w3.org/2001/XMLSchema" >
         < saml2:Issuer >https://login.emory.edu:4444/idp/shibboleth</ saml2:Issuer >
         < ds:Signature xmlns:ds = "http://www.w3.org/2000/09/xmldsig#" >
             < ds:SignedInfo >
                 < ds:CanonicalizationMethod Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n#" />
                 < ds:SignatureMethod Algorithm = "http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
                 < ds:Reference URI = "#_e409c4920e67c7cae7a4609e1b7d80cc" >
                     < ds:Transforms >
                         < ds:Transform Algorithm = "http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                         < ds:Transform Algorithm = "http://www.w3.org/2001/10/xml-exc-c14n#" >< ec:InclusiveNamespaces Prefix
```

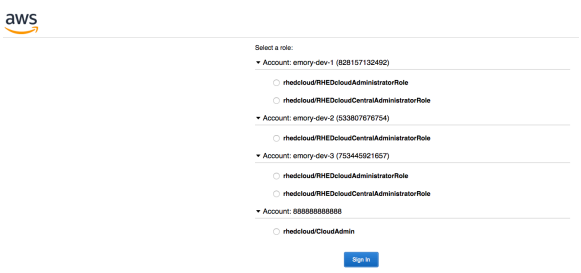
```
List = "xsd" xmlns:ec = "http://www.w3.org
/2001/10/xml-exc-c14n#" /></ ds:Transform >

                    </ ds:Transforms >
                    < ds:DigestMethod Algo
rithm = "http://www.w3.org/2001/04
/xmlenc#sha256" />
                    < ds:DigestValue >UfYd
7tE4r5/GCt3dfYiD/DY9LQXLFzUrXtzwxRIl40Y=</
ds:DigestValue >
                </ ds:Reference >
            </ ds:SignedInfo >
            < ds:SignatureValue >
                ifrjHExwza9WMf0gWRjXI6U+F
CQU9aZqh1rL+/PD11GBED84/qAVdNd7l2vRZppbNE
/GkmzAxUwA
                3MycYEAZmlGxskaOpoRRgtzZm
9Rpx1ewCKCA2Y61VPt7/eARacOEwxH+XndYLUY
/rw0BX35OqtOW
                ucbop493DKdmMRD3o3wj6fKLb
lFU2WI1vAFKyVnu1KhjtiXZ9ZZaiym0vXNXIEhASzX
Rc8/kTzLn
                tnjyJzTDlSQR0p57t8Rwm97Q2
OygrlvaY74J/Nbvc/gBaMw7svOcUZatMmKODDSQd
/d+J9mjn6U+
                Ext5BLgKentliVAchKwKs27lE
ZDSVubaPTu7yA==
            </ ds:SignatureValue >
            < ds:KeyInfo >
                < ds:X509Data >
                    < ds:X509Certificate >
MIIDJzCCAg+gAwIBAgIUfGsUnIe4ehQbVOK7801PiW
LKSX8wDQYJKoZIhvcNAQEFBQAwGjEYMBYG
                    A1UEAxMPbG9naW4uZ
W1vcnkuZWR1MB4XDTEwMDMyOTE2MjAwNVoXDTMwMDM
yOTE2MjAwNVowGjEY
                    MBYGA1UEAxMPbG9na
W4uZW1vcnkuZWR1MIIBIjANBgkqhkiG9w0BAQEFAAO
CAQ8AMIIBCgKCAQEA
                    0RQxoj9S3eD6vcGOB
39jZiPBRx1fTJf6MV96bHNwD
/M+fBCR7t+07VScdH0pffvN0x7g5co0jWJ4
                    KT/16tPpA
/14qzpKd3oHIFz5rgnG1qBSla+kqe9lzls72HMDA5v
daRpPib1ZsaB7JUyTTHMdvVcy
                    aovLILB
/xt2cdy8CCJjIdNt+Uxl5gAwhKfU6DVYspFAWp4uVM
JqO774Xn5Sa68ANjrRLTbMO7Bnq
                    gxCyAAN8ztGlsn+91
Z5JnlMPI/q9uLh6xiUffiwo17S9KOQ+hqQx6XDQGI
/U7TOanuohkC29qt09
                    tLYFXjAB3sfy66qsQ
fS
/rXulScfJsSYNaziMaQIDAQABo2UwYzBCBgNVHREEO
zA5gg9sb2dpbi5l
                    bW9yeS5lZHWGJmh0d
HBzOi8vbG9naW4uZW1vcnkuZWR1L2lkcC9zaGliYm9
sZXRoMB0GA1UdDgQW
                    BBSC4HCY5liG+oEdD
cyxnHhEZt8cKjANBgkqhkiG9w0BAQUFAAOCAQEAFdO
SL9HrkUxokD0zm07e
                    gEpnltaHsusrsZvVh
ayfNqiRlgvhHBpJTa9xUaAyZ45VUsieP4olSdfyiMV
MIRkVgo/gF4W//ZlG
                    xV28CnlalLR10NMzG
6Kz03eRf6v5MQtVL+0EBTO7wiZAFIKYOv8nxpZoL7L
V9l9x5SIDjsgmwV7w
                    Publ0X3aoGq6+IHIk
1TZAA/xWJm1Z1W9fFRWXSLVS1
/eCjPY4rjIktauEsi8Tj6QlhYFuo5WiEaV
                    tNluIP3MzpBSxCUyE
ov49a4rxpHNKPxpSZyFGTDS8Zsm6tV8cZfn0eCypcm
```

```
3lqm/X3HSODuN6n5+ cVrRig01srF/2D9B1A==</ d
s:X509Certificate >
                </ ds:X509Data >
            </ ds:KeyInfo >
        </ ds:Signature >
        < saml2:Subject >
            < saml2:NameID Format = "urn:
oasis:names:tc:SAML:2.0:nameid-format:
transient"
                NameQualifier = "https://l
ogin.emory.edu:4444/idp/shibboleth"  SPNam
eQualifier = "urn:amazon:webservices"
                xmlns:saml2 = "urn:oasis:
names:tc:SAML:2.0:assertion" >_4ced473bb7b
f13389a57d0e79720fa51</ saml2:NameID >
            < saml2:SubjectConfirmation Me
thod = "urn:oasis:names:tc:SAML:2.0:cm:
bearer" >< saml2:SubjectConfirmationData Ad
dress = "10.110.29.67" NotOnOrAfter = "2018-
11-02T21:39:31.763Z"
                Recipient = "https://signi
n.aws.amazon.com/saml"  /></ saml2:
SubjectConfirmation >
        </ saml2:Subject >
        < saml2:Conditions NotBefore = "201
8-11-02T21:34:31.758Z" NotOnOrAfter = "2018
-11-02T21:39:31.758Z" >
            < saml2:AudienceRestriction >
                < saml2:Audience >urn:
amazon:webservices</ saml2:Audience >
            </ saml2:AudienceRestriction >
        </ saml2:Conditions >
        < saml2:AuthnStatement AuthnInstant
= "2018-11-02T21:31:33.264Z"
            SessionIndex = "_1e89c76ef6e3b
e82b3e129e2e3f1c70b" SessionNotOnOrAfter =
"2018-11-03T09:34:31.761Z" >< saml2:
SubjectLocality Address = "10.110.29.67" />
            < saml2:AuthnContext >
                < saml2:
AuthnContextClassRef >https://login.emory.
edu/duo</ saml2:AuthnContextClassRef >
            </ saml2:AuthnContext >
        </ saml2:AuthnStatement >
        < saml2:AttributeStatement >
            < saml2:Attribute FriendlyName =
"https://aws.amazon.com/SAML/Attributes
/SessionDuration"
                Name = "https://aws.
amazon.com/SAML/Attributes
/SessionDuration"
                NameFormat = "urn:oasis:
names:tc:SAML:2.0:attrname-format:uri" >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >43200</
saml2:AttributeValue >
            </ saml2:Attribute >
            < saml2:Attribute FriendlyName =
"Role" Name = "https://aws.amazon.com/SAML
/Attributes/Role"
                NameFormat = "urn:oasis:
names:tc:SAML:2.0:attrname-format:uri" >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::753445921657:saml-provider
/Emory_Dev_IDP,arn:aws:iam::753445921657:
role/rhedcloud
/RHEDcloudAdministratorRole</ saml2:
AttributeValue >
                < saml2:AttributeValue xml
```

```
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::533807676754:saml-provider
/Emory_Dev_IDP,arn:aws:iam::533807676754:
role/rhedcloud
/RHEDcloudCentralAdministratorRole</ saml2
:AttributeValue >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::753445921657:saml-provider
/Emory_Dev_IDP,arn:aws:iam::753445921657:
role/rhedcloud
/RHEDcloudCentralAdministratorRole</ saml2
:AttributeValue >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::828157132492:saml-provider
/Emory_Dev_IDP,arn:aws:iam::828157132492:
role/rhedcloud
/RHEDcloudAdministratorRole</ saml2:
AttributeValue >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::888888888888:saml-provider
/Emory_Dev_IDP,arn:aws:iam::888888888888:
role/rhedcloud/CloudAdmin</ saml2:
AttributeValue >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >arn:aws:
iam::828157132492:saml-provider
/Emory_Dev_IDP,arn:aws:iam::828157132492:
role/rhedcloud
/RHEDcloudCentralAdministratorRole</ saml2
:AttributeValue >
            </ saml2:Attribute >
            < saml2:Attribute FriendlyName =
"RoleSessionName"
                Name = "https://aws.
amazon.com/SAML/Attributes
/RoleSessionName"
                NameFormat = "urn:oasis:
names:tc:SAML:2.0:attrname-format:uri" >
                < saml2:AttributeValue xml
ns:xsi = "http://www.w3.org/2001/XMLSchema-
instance"  xsi:type = "xsd:string" >P3184803
</ saml2:AttributeValue >
            </ saml2:Attribute >
        </ saml2:AttributeStatement >
    </ saml2:Assertion >
</ saml2p:Response >
```

The AWS Console page then generates a page of role the user may select to assume. There are the roles that were passed in the SAML assertion.



CIMP-206 - Create mechanisms to deprovision access to roles / permissions in an automated fashion if not authorized periodically.

Users are automatically de-provisioned from all MyNetID roles when they leave the University according to the Account LifeCycle policy.

Currently, MyNetID does not have an automated process that requires a user's roles to be authorized periodically. This functionality could be implemented by developing workflows within MyNetID, but there are a number of questions that have to be answered.

CIMP-207 - Create mechanisms to authorize users to particular roles & privs. Audit all requests & authorizations.

IDM will create a management role for each AWS permission roles that is created in MyNetID. Members of the management roles can add and remove members from the appropriate AWS permissions roles. Members of the AWS permissions roles are able to "assume the role" of the analogous roles created within AWS. IDM will assign members to these management roles by request made through ServiceNow.

Some auditing can be manually done using logs and data in MyNetID's backend datastore, but there is currently no formal process in place.

MyNetID

IAM approach from AWS@Emory Project (CIMP-702)

MyNetID Roles and Resources - LINK (Attachment)

MyPassword

Administrative and self service password resets on enterprise accounts.

University LDS

IAM approach from AWS@Emory Project (CIMP-702)

Microsoft LDS (LDAP) Configuration - LINK (Attachment)

University AD

All users are stored in OU=People.

Shibboleth IDP(Federated SSO)

IAM approach from AWS@Emory Project (CIMP-702)

AWS Account Shibboleth Configuration - LINK (Attachment)

VPCP

IAM approach from AWS@Emory Project (CIMP-702)

Emory Amazon Web Services Service Description - LINK

SRD/SRR

IAM approach from AWS@Emory Project (CIMP-702)

Security Risk Detector Technical Design Document - LINK

AWS SCP Resources

IAM approach from AWS@Emory Project (CIMP-702)

rhedcloud-aws-org-standard-scp.json - Latest standard SCP policies - LINK

rhedcloud-aws-org-hipaa-scp.json - Latest HIPAA SCP policies - LINK

AWS IAM Resources

IAM approach from AWS@Emory Project (CIMP-702)

rhedcloud-aws-rs-account-cfn.json - Cloudformation template used to create resources within AWS accounts - LINK

AWS IAM Policies

IAM approach from AWS@Emory Project (CIMP-702)

rhedcloud-aws-rs-account-cfn.json - Cloudformation template used to create resources within AWS accounts - LINK

rhedcloud-aws-vpc-type1-cfn.json - Cloudformation template used to create resources within AWS accounts - LINK

rhedcloud-aws-vpc-type2-cfn.json - Cloudformation template used to create resources within AWS accounts - LINK

Logging and Auditing Requirements

Content transferred to Artifact: Performance and Operational Health Systems - Section 4.1: Identity management logs