Secrets Manager project set up in AWS

Introduction

Secrets Manager (SM) is a service from AWS which helps you protect secrets needed to access your applications, services, and IT resources. The service enables you to easily rotate, manage, and retrieve database credentials, API keys, and other secrets throughout their lifecycle.

Secrets Manager will be the new Vault solution in AWS to access the Secrets for Non-Interactive use cases for new and existing FNMA applications in the AWS Cloud and the applications that are getting migrated to the AWS Cloud. Currently the applications interact with AWS PAM and On-Prem CyberARK interfaces to access the Secrets.

Secrets Manager will be deployed in each AWS life cycle. There will be two components to Secrets manager set up:

- Wrapper Layer - to provide interface for Apps to access the Secrets
- Management layer - to perform Secret Management.

As part of this initiative, there will be changes in the Resource Provisioning process for RDS and Redshift databases. The new process will have IAG integration to validate if the Identity part of the resource provisioning request is present in IAG. EDBSS team will make changes in the process to integrate Secrets Manager.

Scope

The scope of this project includes:

- Develop software libraries (SDKs) for FNMA applications to access secrets from AWS Secrets Manager. This will replace the existing libraries being used with requiring any application code change.
- Integrate the Resource Provisioning process in FNMA AWS Cloud with the AWS Secrets Manager
- Develop secrets management functions to enable industry standard secrets management policies while accommodating all the use cases of FNMA applications deployed in AWS Cloud.
- Migrate existing secrets from AWS PAM and CyberArk to AWS Secrets Manager
- Develop SDLC automation pipeline for SDKs
- Develop SDLC automation pipeline for secrets management functions
- Establish process and control for secrets stored in AWS Secrets Manager

Assumptions

- Application team will be provided with new libraries (ex: new JAR in case of Java), however there will not be any code change required for applications
- IAG team will be updating the logic to integrate with the Secrets Manager
- EDBSS team will make changes in the process to integrate secrets manager
- Secrets Manager will be deployed in all the FNMA AWS lifecycles

Constraints

- App teams will have to deploy the new InfoSec developed Wrapper API (SDK) in their Application framework
- The SDK should preserve the function interfaces provided by the AWS PAM SDK

Use cases

Here are the main use cases for Secrets Management:

Management Use Cases:

- On-board and Offboard the System and Application Ids Secrets through automation, including access management for the secret as part of the provisioning process
- Authorized administrators to On-board and Offboard the Secrets those don't support automation. Ex. External Service Credentials such Bloomberg Credential, market data retrieval credential
- Rotate Secrets for all databases and platforms supported at Fannie Mae by manual initiation and policy-based automation
- Update/Store Secrets for platforms that doesn't support rotation programmatically
- Ability to create secrets rotation policies specific for the Application
- Ability for authorized users to Test/Verify a Secret within a Secret Manager without revealing the secret

Run-time Use Cases:

- Ability for authorized Applications to retrieve the Secrets programmatically
- Logging, monitoring and reporting of user and host activities of secret management and usage
- Ability to prevent one application from calling another applications credentials

Types of Secrets

Secrets Manger will be used to manage the application credentials for these resource types

1. AWS RDS
2. AWS Aurora Databases
3. AWS RedShift
4. Active Directory

Secrets Manger will also be used to store the application credential of External Systems. The following Non-Unique Identity types are in Scope:

1. System ID
2. Application ID

Fannie Mae Application Categories

There are the main Categories of FNMA Applications:

1. FNMA existing or new AWS Applications
2. Re-host: VMWare's VMC (VMWare Cloud) in AWS. This would be treated like it is on premise.
3. Re-Factor is using all native AWS services. The container environment will be located in this space.
4. Re-Purchase: this will be utilized for COTS products (i.e. DataSunrise, Guardium)
5. Re-platform is the hybrid. You can have VMs in the SDDC and then utilize the RDS, S3, SQS that is in AWS.

Key Stake Holders, Roles, Responsibilities

1. IAG - For My Services Integration
2. Cloud Engineering - For AWS resource provisioning
3. EDBMS - For Database services
4. InfoSec - For Secrets Set up

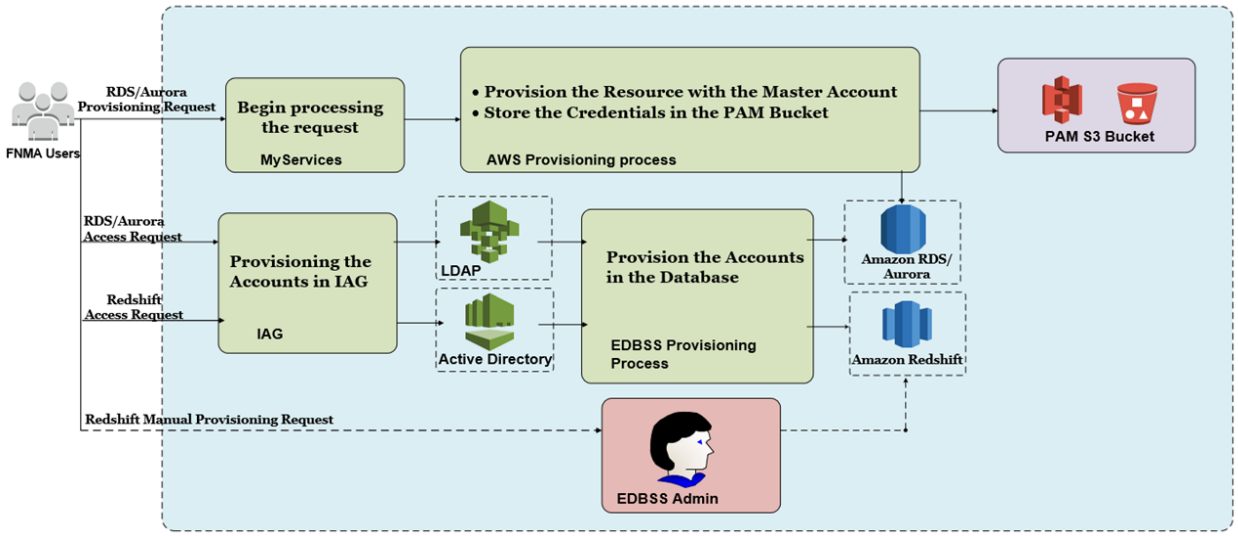
Secrets Manager: Integration Overview

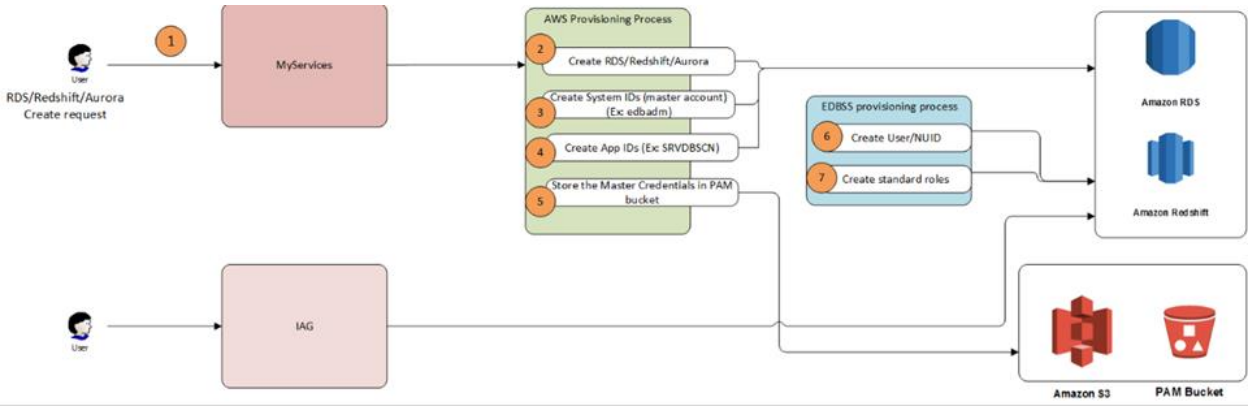
Current Process for NUID Credential Management

The current process for credential handling for RDS databases in AWS is follows:

- The system credentials (master accounts) created during provisioning are stored in S3 based PAM solution
- No other credentials are created during day 0 provisioning
- Any other NUIDs (system accounts, application accounts) requests initiated through ServiceNow are manually executed bu IAG and EDBSS using some helper scripts.
- The cerdetials for the above new NUIDs created are not stored in the PAM S3 bucket
- Currently there is now automation for periodically rotating these credentials for the master and NUID accounts for RDS and RedShift
- The RedShift provisioning requests are manually executed by IAG+EDBSS. The master account is updated in applications PAM S3 bucket

The following diagrams summarize the current process



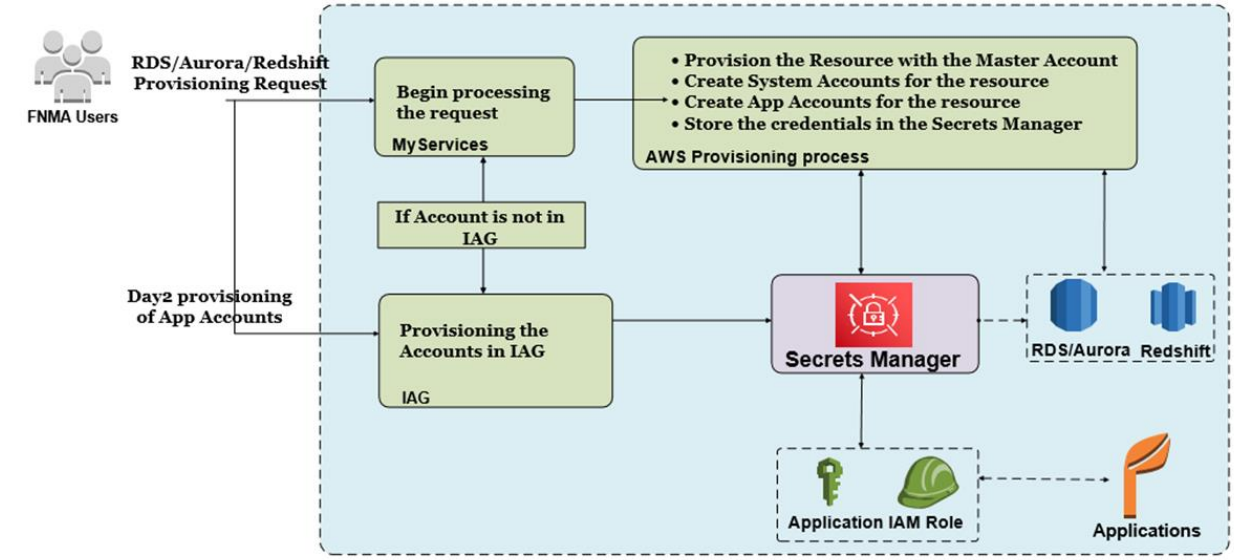


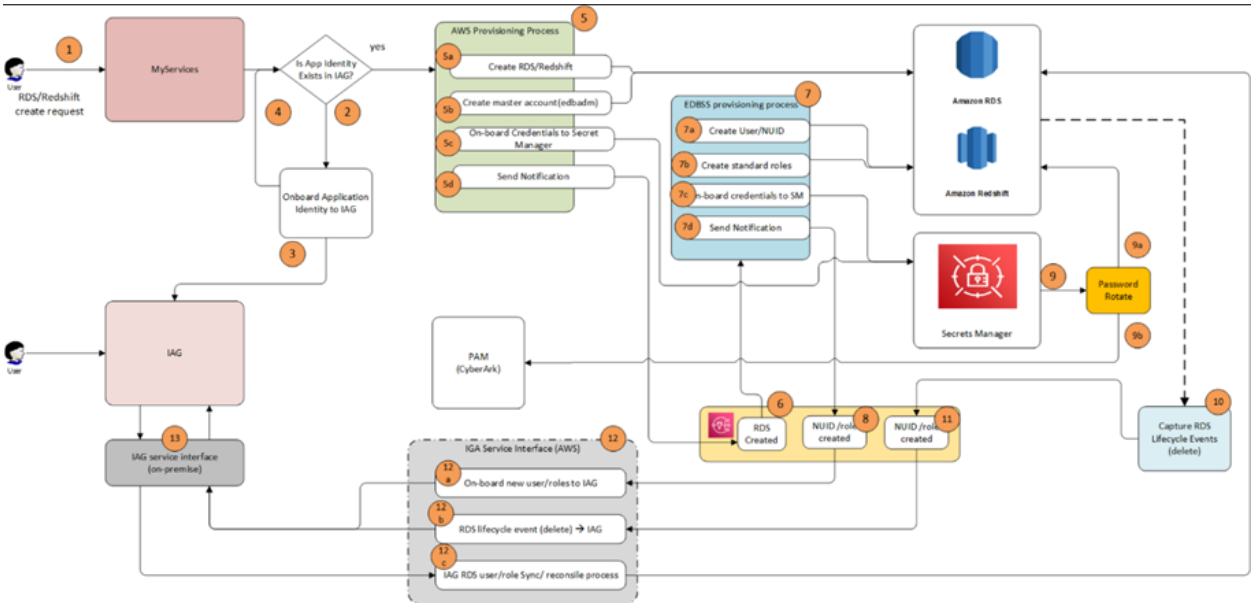
Proposed New Process for NUID Credential Management

The proposed new process addresses shortcomings of the current process and replaces PAM S3 with AWS Secrets Manager (ASM) for storing credentials for master accounts and other NUIDs created during and after the resource provisioning. The highlights of the new process are:

1. Automation will be developed for creating master account and any other system accounts for RDS and RedShift and adding them to ASM during resource provisioning (Day 0). This will require integration of provisioning process with IAG for requesting new credentials, on-boarding and checking for existing ones.
2. Requests for any new NUIDs, like application accounts (day 2) will be handled by IAG automation and added to ASM
3. Automation for rotating credentials for RDS, RedShift and other applications and products will be developed and integrate with ASM where ever possible.
4. Automation will be developed to delete credentials from ASM and off-board from IAG during resource decommissioning

The following diagrams summarize the new process.

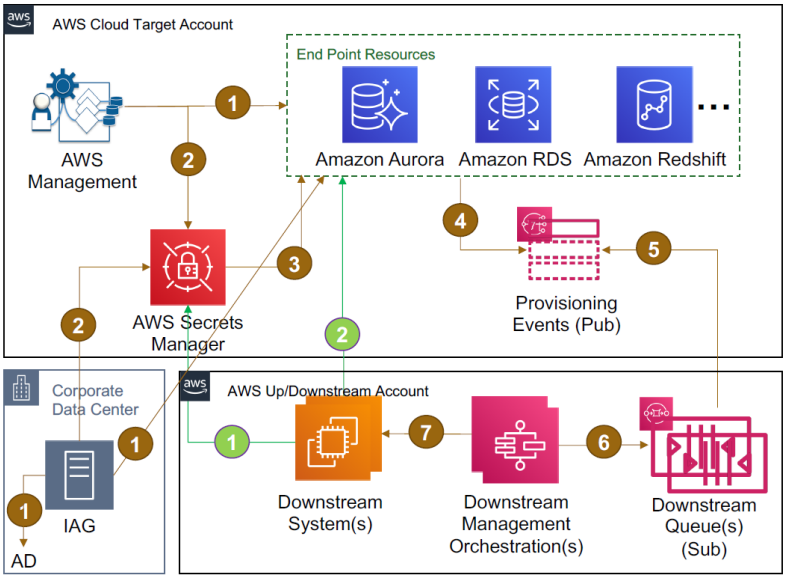

FNMA Users
RDS/Aurora/Redshift Provisioning Request
Begin processing the request
MyServices
• Provision the Resource with the Master Account
• Create System Accounts for the resource
• Create App Accounts for the resource
• Store the credentials in the Secrets Manager
AWS Provisioning process
If Account is not in IAG
Day2 provisioning of App Accounts
Provisioning the Accounts in IAG
IAG
Secrets Manager
RDS/Aurora
Redshift
Application IAM Role
Applications


RDS/Redshift create request
MyServices
Is App Identity Exists in IAG?
yes
Onboard Application Identity to IAG
AWS Provisioning Process
Create RDS/Redshift
Create master account(edbadm)
On-board Credentials to Secret Manager
Send Notification
EDBSS provisioning process
Create User/NUID
Create standard roles
On-board credentials to SM
Send Notification
Amazon RDS
Amazon Redshift
Secrets Manager
Password Rotate
IAG
PAM (CyberArk)
RDS Created
Capture RDS Lifecycle Events (delete)
IAG service interface (on-premise)
IGA Service Interface (AWS)
On-board new user/roles to IAG
RDS lifecycle event (delete) → IAG
IAG RDS user/role Sync/ reconsile process

Overview of the new Secrets Management Architecture using ASM and Other Automation

The key features of the new proposed architecture for secrets management are:

- Automation for creation for all NUIDs - master, system and application
- Automation for on-boarding of secrets into ASM and rotation of credentials upon creation.
- Pub-Sub model for resource provisioning and decommissioning events to allow down-stream applications to subscribe and these events and take appropriate actions related to credentials stored in ASM
- ASM is a pre account and regional service. ASM in all active regions in each Fannie Mae account will store credentials for resources in that region.
- Downstream applications like IAG, EDBSS, Guardium may be in a specific AWS account (e.g. Prod P2P or P2X). These services will used AWS Cross Account Access Roles (CARs) to fetch credentials from ASM in account and region corresponding to the resource being accessed



1 AWS Resource Management Process:
- Provisions/De-Provisions System ID Accounts (fmsm, iam, guardium, edbss, etc.)
- Provisions /De-Provisions Application ID Account/s

2
- Vault Secrets in Secret Manager

3 Rotate Secrets upon on-boarding into Secret Manager with Random Credential

4 Commissioning / De-Commissioning Events Published to the SNS Topic.

5 Downstream System (Ex. Guardium, IAG, EDBSS) specific SQS Queue.

6 Downstream System Management Orchestration to On-board/Off-board the

7 Resources (Aurora, RDS, Redshift etc.) by reading the Queue and calling Downstream REST API

Downstream System Examples: IAG, EDBSS Mgmt, Guardium, Data Sunrise, etc.

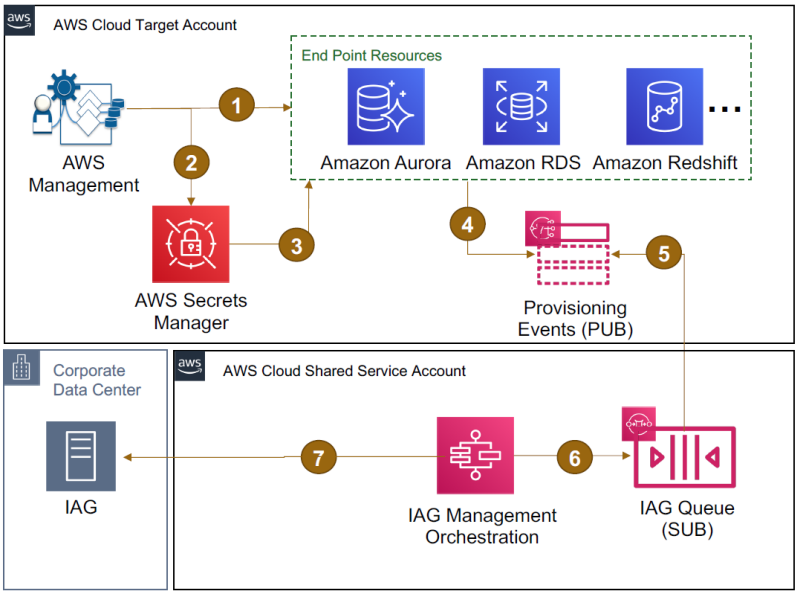
1 Downstream System runs with AWS resource role. Assumes the Cross Account Role from the AWS Target Account to Retrieve Secrets

2 Downstream System connects to the target Resource (Aurora/RDS/Redshift/etc.) using the Retrieved Secret to perform its functionality.

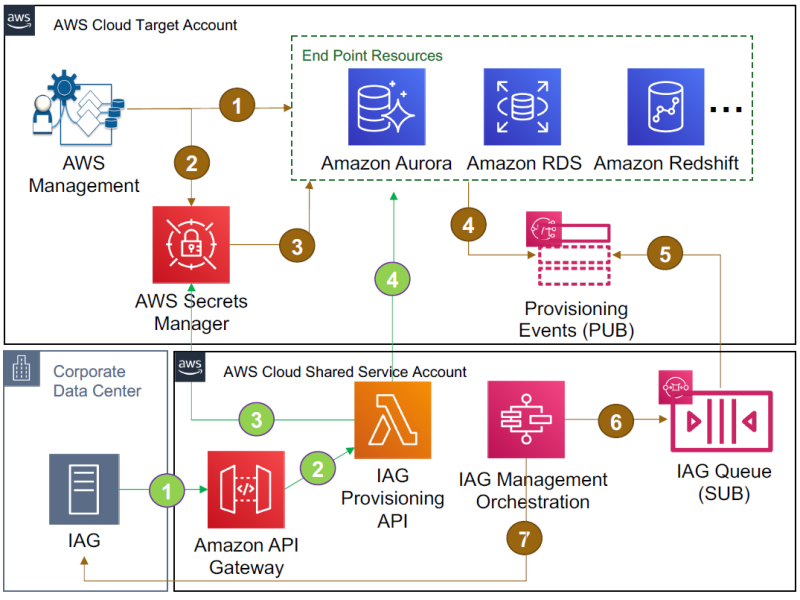
Specific Use Cases

The following sections describe the secrets management architecture in more details with specific use cases

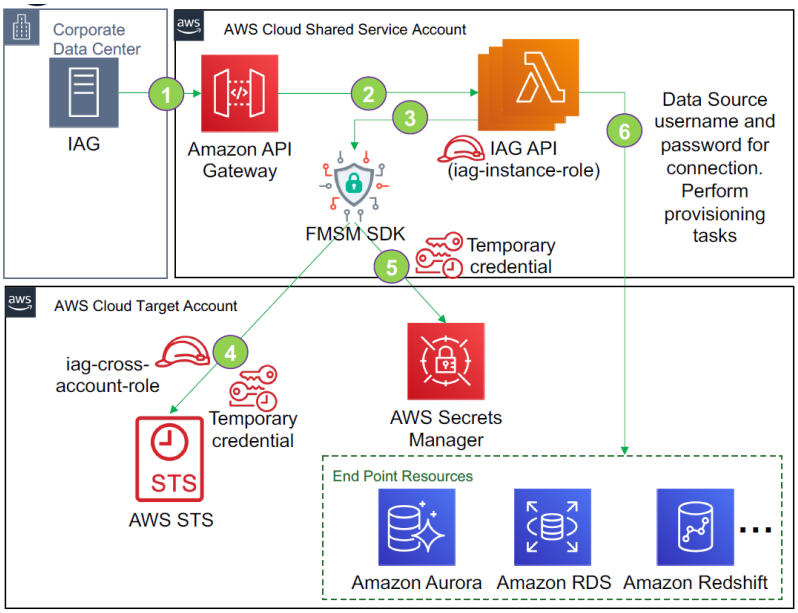
IAG Integration



1. AWS Resource Management Process:
   - Provisions/De-Provisions System ID Accounts (fmsm, iam, guardium, edbss, etc.)
   - Provisions /De-Provisions Application ID Account/s
   - Vault Secrets in Secret Manager
2. Rotate Secrets upon on-boarding into Secret Manager with Random Credential
3. Commissioning / De-Commissioning Events Published to the SNS Topic.
4. IAG specific SQS Queue subscribed to Provisioning Events.
5. IAG On/Off-Boarding Orchestrator for Resources (Aurora, RDS, Redshift, etc.) Reads Queue.
6. IAG On/Off-Boarding Orchestrator calls **IAG REST API** to On/Off-Boarding Resources (Aurora, RDS, Redshift, etc.) into IAG.



1. AWS Resource Management Process:
   - Provisions/De-Provisions System ID Accounts (fmsm, iam, guardium, edbss, etc.)
   - Provisions /De-Provisions Application ID Account/s
   - Vault Secrets in Secret Manager
2. Rotate Secrets upon on-boarding into Secret Manager with Random Credential
3. Commissioning / De-Commissioning Events Published to the SNS Topic.
4. IAG specific SQS Queue subscribed to Provisioning Events.
5. IAG On/Off-Boarding Orchestration for Resources (Aurora, RDS, Redshift, etc.) Reads Queue and calls **IAG REST API**
6. IAG calls IAG Provisioning API in AWS
7. IAG Provisioning API runs with AWS resource role, assumes the Cross Account Role from the AWS Target Account to Retrieve Secrets
8. Downstream System connects to the target Resource (Aurora/RDS/Redshift/etc.) using the Retrieved Secret to perform its functionality.



Data Sources are deployed in multiple AWS Accounts-Regions. Each AWS Account-Regions will have AWS Secret Manager with Secrets specific to the Data Sources in that region for the AWS Account.

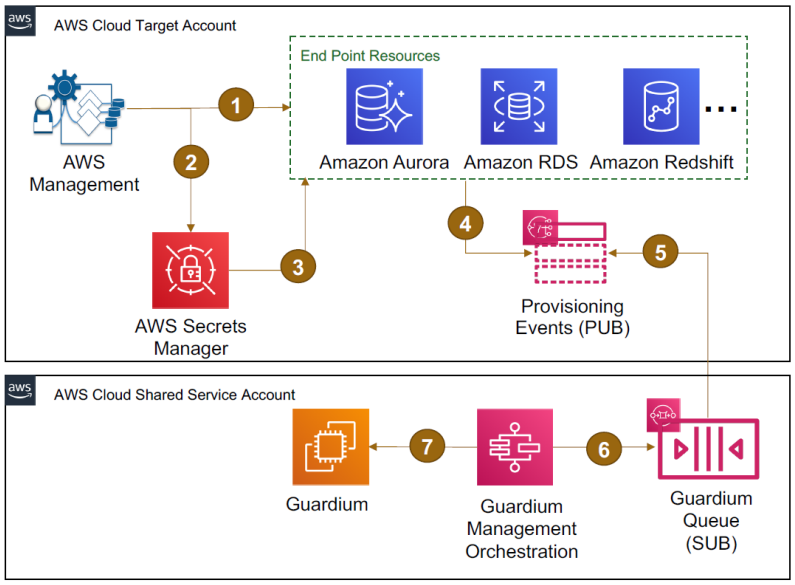
Each of the Target AWS Accounts will have a role, say, *iag-cross-account-role* that can be assumed by *iag-instance-role* from AWS Account where IAG API is deployed.

IAG instances are attached with *iag-instance-role*.

To get the Data Source Credential from AWS Secret Manager...

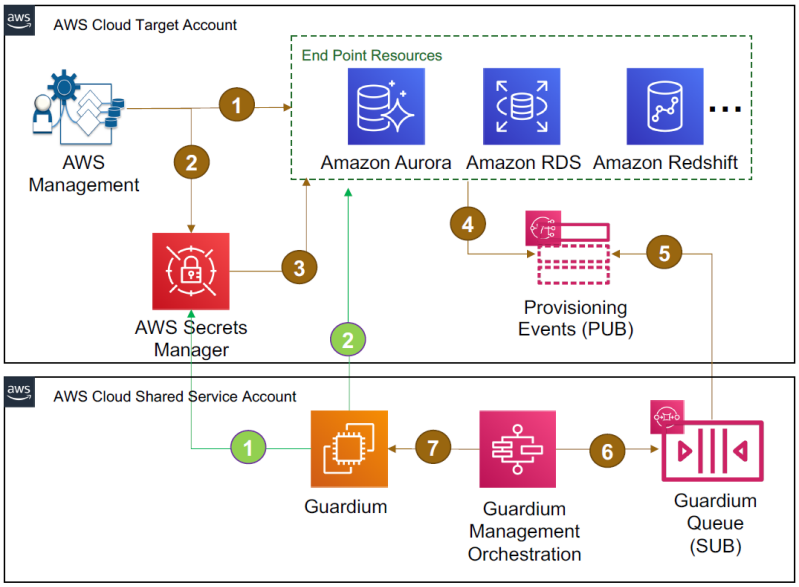
3. IAG API Lambda Assumes the *iag-cross-account-role* from Target AWS Account where the Data Source is deployed to get Temporary AWS security credential.
4. IAG API Lambda call AWS Secret Manager API, *get_secret_value*, to get the Data Source Secrets using the Temporary AWS security credential retrieved in the previous step.
5. Using the retrieved Data Source Secrets; IAG API Lambda connect to the Data Source to perform provisioning tasks

COTS (Guardium) Integration



AWS Resource Management Process:

1. Provisions/De-Provisions System ID Accounts (fmsm, iam, guardium, edbss, etc.)
2. Provisions /De-Provisions Application ID Account/s
   - Vault Secrets in Secret Manager
3. Rotate Secrets upon on-boarding into Secret Manager with Random Credential
4. Commissioning / De-Commissioning Events Published to the SNS Topic.
5. Guardium specific SQS Queue subscribed to Provisioning Events.
6. Guardium On/Off-Boarding Orchestrator for Resources (Aurora, RDS, Redshift, etc.) Reads Queue.
7. Guardium On/Off-Boarding Orchestrator calls **Guardium REST API** to On/Off-Boarding Resources (Aurora, RDS, Redshift, etc.) into Guardium



AWS Resource Management Process:

1. Provisions/De-Provisions System ID Accounts (fmsm, iam, guardium, edbss, etc.)
2. Provisions /De-Provisions Application ID Account/s
   - Vault Secrets in Secret Manager
3. Rotate Secrets upon on-boarding into Secret Manager with Random Credential
4. Commissioning / De-Commissioning Events Published to the SNS Topic.
5. Guardium specific SQS Queue subscribed to Provisioning Events.
6. Guardium On/Off-Boarding Orchestration for Resources (Aurora, RDS, Redshift, etc.)
7. Reads Queue and calls **Guardium REST API**

1. Guardium runs with AWS resource role. Guardium Assumes the Cross Account Role from the AWS Target Account to Retrieve Secrets
2. Downstream System connects to the target Resource (Aurora/RDS/Redshift/etc.) using the Retrieved Secret to perform its functionality.

Secrets Manager Disaster Recovery

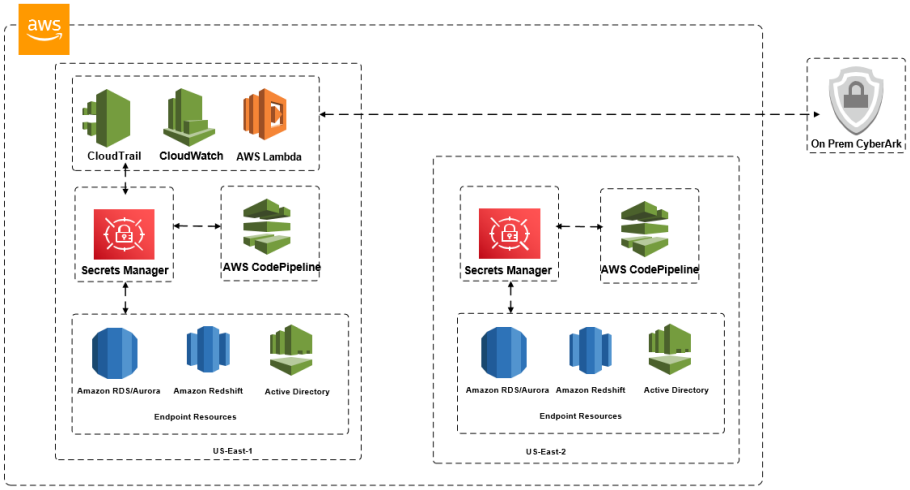
- Secrets Manager will be deployed in all FNMA AWS Accounts, which means it will be deployed in US-East-2 (Contingency) for Production accounts as well.
- Secrets Manager from US-East-2 regions will have the secret values for the Resources deployed in US-East-2
- US-East-1 Secrets will also be copied in CyberARK using a Lambda function which will also pick up rotation requests

DR Use Case1: Entire US-East-1 Region failure

- Whenever there is a failure of US-East-1, there will not be any disruption since the Secrets in US-East-2 will not be impacted.

DR Use Case2: Only Secrets Manager service from US-East-1 is down

- Option1: Work in progress with Amazon to enable CRR (Cross Region Replication) in which case Applications can fail over to Region2 Secrets Manager. No ETA at this point of time
- Option2: SDK will failover to Secrets Manager from Region2



Application Software Libraries (SDKs)

1. Software libraries will be developed in following languages
    - Java
    - Python

2. Functional requirements
    - This library will replace the exiting AWS-PAM-SDK which reads secrets stored in application S3 buckets. The existing function interfaces from AWS-PAM-SDK will be preserved in the new libraries which will read secrets from AWS Secrets Manager.
    - The SDK will use the input parameters and applications to construct the application secret's namespace. It will the use the applications IAM role to restrict access to application's namespace (vertical segmentation)
    - The application secrets will be fetched from the AWS Secrets Manager in the requesting application's AWS account and region.
    - The requested secrets will be cashed for 60 minutes (configurable).
    - How to manage stale cache? Refresh initiated by SDK or application? Subscribe for notifications?
    - The application IAM role should also have access to the application's KMS-CMK used for encrypting the secret

Integration with Resource Provision Automation

The following task need to be performed for on-boarding and off-boarding of secrets during provisioning and decommissioning of AWS resources that use secrets stored in AWS Secrets Manager:

Initial setup tasks

1. Deploy management lambda functions in each account
2. Create required roles and policies in each account
3. Create SNS topic in each account
4. Create/configure SQS in downstream accounts (IAG, Guardium)

Day 0 provisioning functions:

5. Check if the required accounts for new resources already exists in IAG
6. Create secret in AWS SM with access control
7. Rotate secret after resource provisioning (if auto-rotation enables for secret)
8. Send commission events to SNS topic

Day 2 provisioning functions:

9. Accept account creation requests from IAG
10. Create secret in AWS SM with access control
11. Create accounts in target resource
12. Rotate passwords if auto-rotation enabled for secret
13. Send secret creation requests to downstream SNS topic

AWS-SM: Password and Secrets Policies

Password Requirements

Fannie Mae passwords must adhere to the following:

- Passwords must be a minimum of twelve characters in length.
- Passwords must contain at least one character from three of the following four classes:

    1. Any upper case letter
    2. Any lower case letter
    3. Any numerical (0, 1, 2,)
    4. Any non-alphanumeric (special) character

Password Change Guidance for Non-Unique IDs

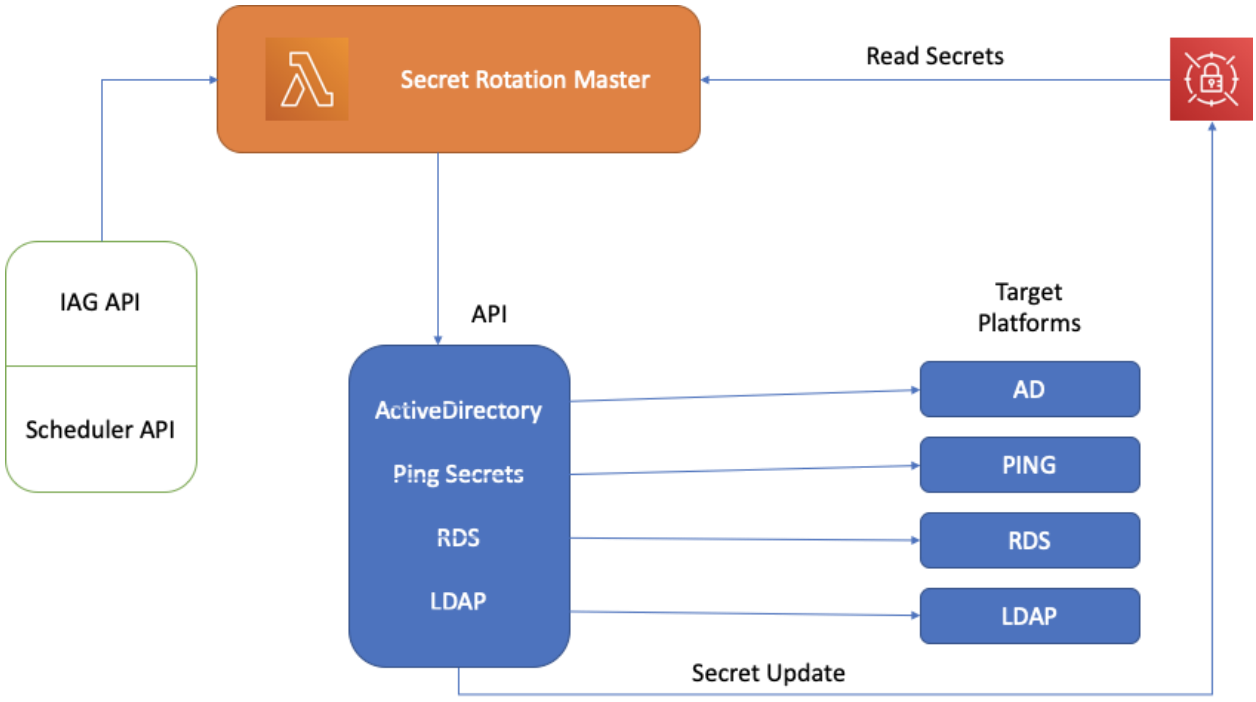
Non-Unique ID (NUID) Type	Production Environment: High, Medium & Low Risk Assets	Non-Production (Lower Environment) Assets
Non-Interactive IDs	No Password Reset Required	No Password Reset Required
Direct Login Disabled	No Password Reset Required	No Password Reset Required
Privileged Interactive IDs (Administrative Privilege or Privilege Access)	Password reset required once every year or more frequently **Or** Password reset upon the following triggering events: user termination, ownership change, or job responsibility change of ID password users	Password reset required once every year or more frequently.
Privileged Interactive IDs: Required Timeframes	Password change request initiated within 10 business days of triggering event Password changed within 30 business days after password change request	

	submission	
Non-privileged Interactive IDs (Not Administrative Privilege or Privilege Access)	Password reset required every two years or more frequently, **Or** Password reset upon the following triggering events: user termination, ownership change, or job responsibility change of ID password users	Password reset required once every two years or more frequently.
Non-privileged Interactive IDs: Timeframes	Password change request initiated within 10 business days of triggering event Password changed within 30 business days after password change request submission	

SecretsMgnt API Documentation

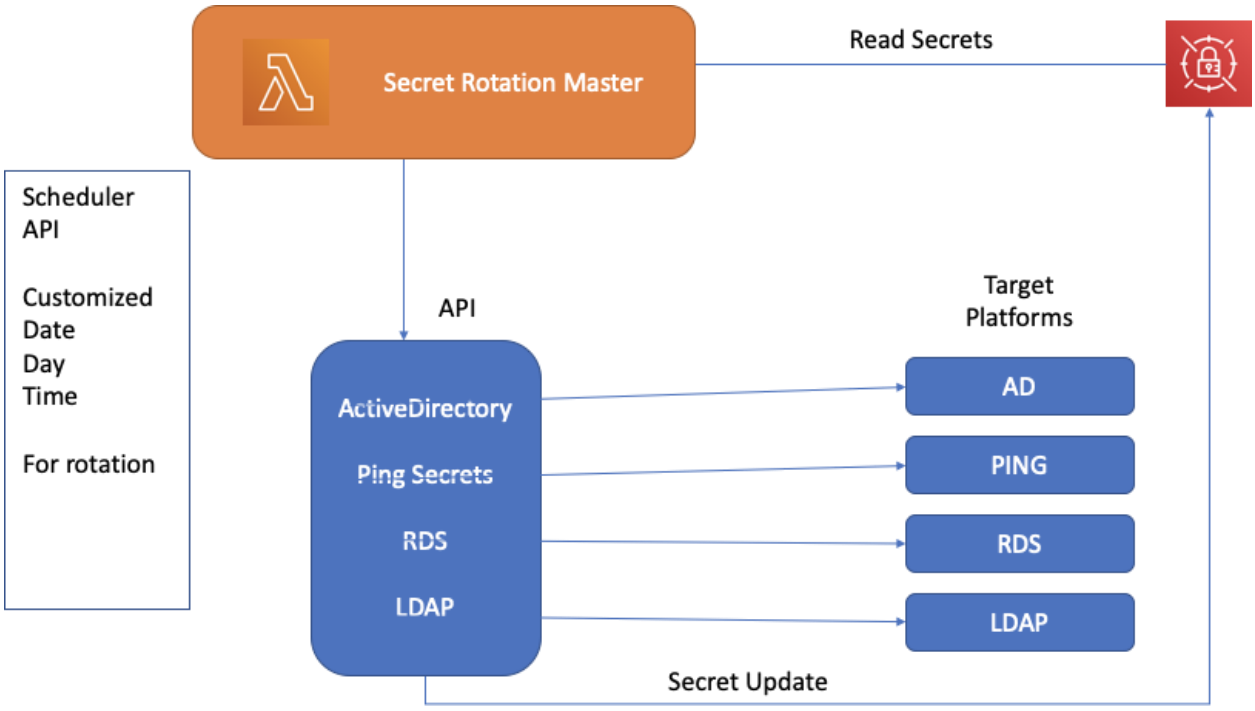
Use Cases for rotation API

- Password reset initiated immediately after provisioning the secret in secret manager. This is currently applicable for Active Directory secrets
- Password reset on a scheduled basis
- Password reset as adhoc request, self-service for Operations team (e.g PSS)
- Password reset initiated immediately after provisioning the secret in secret manager. This is currently applicable for Active Directory secrets
- Active Directory accounts, when provisioned in IAG, temporary password is available for NUID owners to use. When Active Directory account is on-boarded to Secret Manager, password reset will be initiated immediately.
- Provisioning process to initiate Password reset master lambda with Secret Name. Master lambda will retrieve the Secret id and host name and initiates platform specific lambda to do password reset in both Secret manager and platform



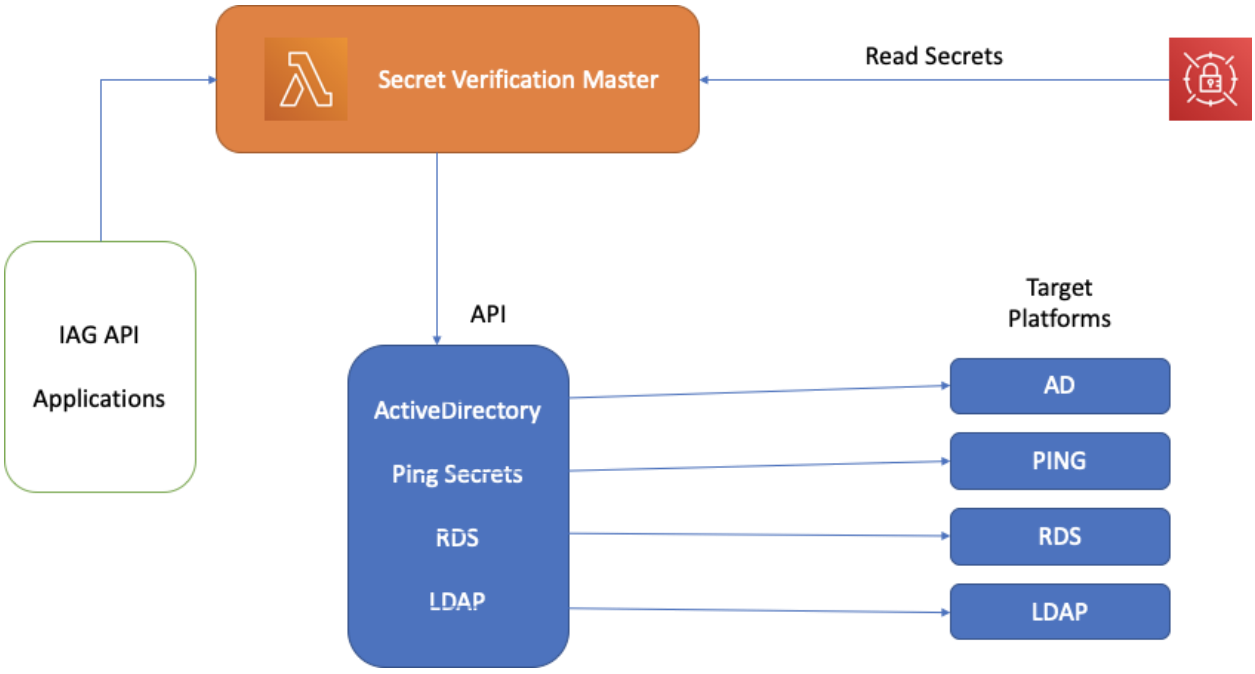
2. Password reset on scheduled basis

Scheduler lambda to reset password in Secret Manager and target platforms on a date and time set . Applications need to have the ability to attach the schedule lambda to their secrets based on their compliance requirements.



3. Password reset as adhoc request, self-service for Operations team (e.g PSS). Secret rotation for emergency changes.

4. Password Verification API



AWS-SM: Access Control for Secrets

IAG Cross-Account Role to Manage Secret

IAG Cross Account Roles shall be provisioned as part of the Account Provisioning Process

- Access to describe and get Master Accounts Secrets, including KMS Key Access
- Access to manage Service Account Secrets, including KMS Key Access
- Application Account Secrets, Application Specific KMS Key for Secret is provisioned as part of the Application Provisioning Process and that Process need to grant KMS Access to these roles in respective AWS Account Regions. Refer "Access Control for Secrets and its KMS Key by Persona" for specification

Syntax

Syntax for Cross Account Role: **CAR-<lifecycle>-identity-management**

Where **lifecycle** - AWS Lifecycle, The lifecycle information for the account by region shall be provided in AWS Account Lifecycle Events

Examples:

CAR in **devl-sfbu** AWS Account (only us-east-1 region):

```
CAR-devl-sfbu-identity-management
```

CAR in **prod-sfbu** AWS Account (in us-east-1 region):

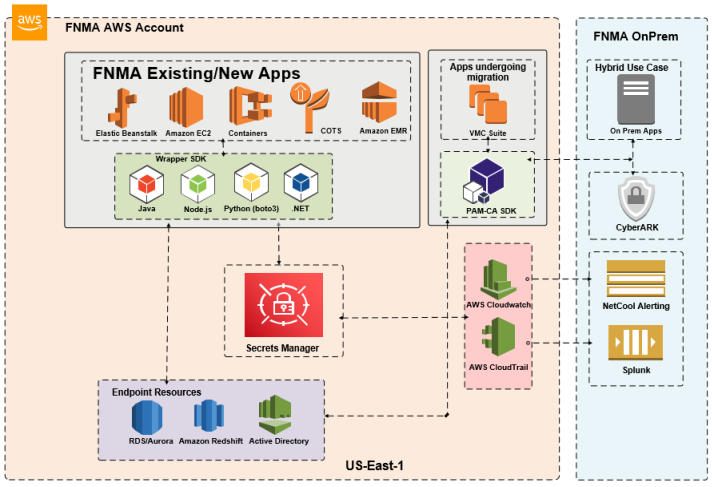
```
CAR-prod-sfbu-identity-management
```

CAR in **cont-sfbu** AWS Account (in us-east-2 region):

```
CAR-cont-sfbu-identity-management
```

Secrets Manager Usage Overview

Secrets Manager Usage: Option 1



VMC Suite and OnPrem Apps will leverage PAM-CA SDK to fetch secret from CyberARK to connect to AWS app databases.

Pros:

Simplified SDK

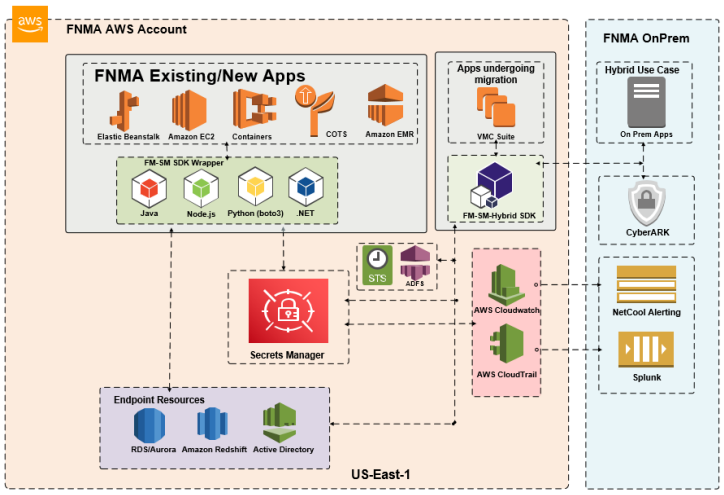
No changes to existing application

Cons:

Password Synchronization between vaults

Management will be complex

Secrets Manager Usage: Option 2



VMC Suite and OnPrem Apps will leverage FM-SM-Hybrid SDK to fetch secret from Secrets Manager using STS and ADFS to connect to AWS app databases
The SDK needs to fetch secret from CyberARK for connecting with OnPrem Databases

Pros:

- One SDK supports AWS and PAM-CA
- No Password Synchronization

Cons:

- Complex SDK
- App may need to provide additional configuration

Secrets Manager is a regional service, it will be set up in each AWS Accounts

FM-SM SDK Wrapper

- For Applications to retrieve credentials from Secrets Manager to access the Application Databases.
- Caching
- Supported technologies: Java, Python, NodeJS and .NET, JDBC Driver, Spring Beans

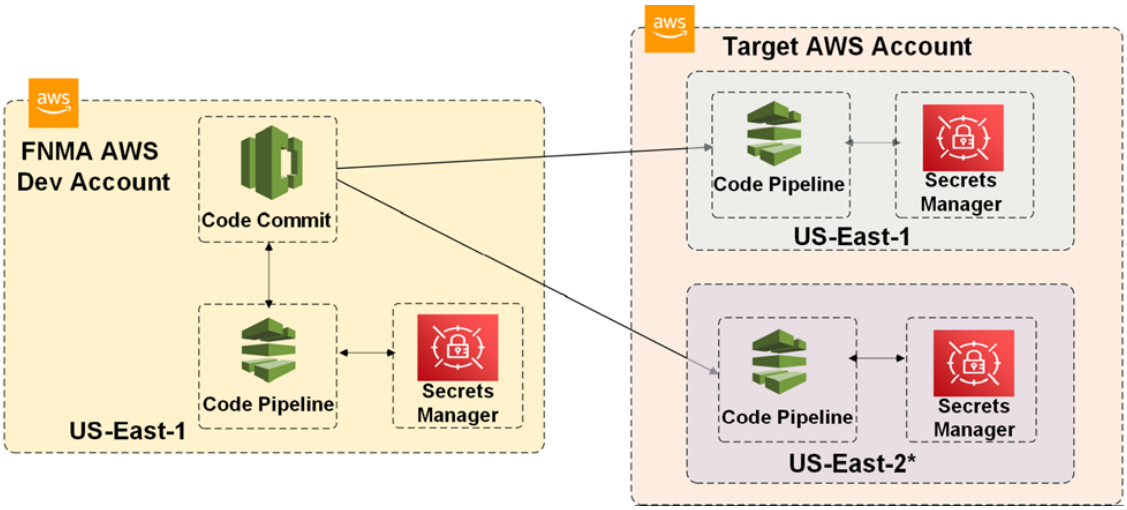
Password Management

- Secrets Rotation can be scheduled on the Secret depending on the Application Needs
- Rotation will also be enabled manually if required

Policy and code migration

- Lambda functions and CI/CD pipeline will be used for policy and code migration
- This will also interface with Cloud watch for NetCool Alerting purposes
- Cloud Trail will be used for Auditing and results can be monitored in Splunk

CI/CD Overview:



*US-East-2 is applicable only for Production accounts like Shared, SF, EDL