



AWS Connector Guide v2020.x

Copyright

saviynt.com

© 2023 Saviynt. All rights reserved. No part of this document may be reproduced or used in any manner without the prior written permission of the copyright owner.

CONTENTS

ABOUT THIS GUIDE	1
AWS INTEGRATION OVERVIEW	3
PREPARING FOR INTEGRATION	16
DEPLOYMENT OPTIONS	39
CONFIGURING THE INTEGRATION FOR IMPORTING ACCOUNTS	42
CONFIGURING THE INTEGRATION FOR IMPORTING ENTITLEMENTS	49
CONFIGURING THE INTEGRATION FOR PROVISIONING ACCOUNTS AND ENTITLEMENTS	57
TAKING ACTION ON SENSITIVE VIOLATIONS	59
MAPPING ACTIVE DIRECTORY GROUPS TO AWS ROLES	64
CREATING USERS WITH EMERGENCY ACCESS ROLES	69
SETTING UP REAL TIME MONITORING	70

INTEGRATING EIC WITH CLOUDKNOX FOR AWS RESOURCES	76
TROUBLESHOOTING	95
APPENDIX	106

About this Guide

This guide describes the integration between Saviynt Enterprise Identity Cloud (EIC) and the Amazon Web Services (AWS) account of the customer.

Audience

This guide is intended for administrators and target application integration teams responsible for implementing a secure integration service with the AWS account of the customer.

Text Conventions

The following text conventions have been used in this document:

Convention	Meaning
bold	Indicates graphical user interface elements that are associated with an action.
<i>italic</i>	Indicates guide titles and placeholder text for which you specify values.

Convention	Meaning
<code>inline code</code>	Indicates code elements, executable commands, cmd prompt input or output details, and URLs.
<code>courier new</code>	Indicates parameter values and directory or file paths.

Related Documents

In addition to the information provided in this guide, refer to the see [Saviynt Enterprise Identity Cloud Connectors](#) page for related information.

Access to Saviynt Support

Saviynt customers can contact Saviynt Support at <https://saviynt.freshdesk.com/support/home>.

AWS Integration Overview

AWS is the world's most comprehensive and broadly adopted cloud platform, offering over two hundred fully featured services from data centers globally. The AWS Connector allows you to create a connection between your AWS account and EIC using the AWS API. When you are connected, you can import the AWS data to Saviynt. The imported data includes IAM Users, resources hosted on the AWS account and all the metadata associated with AWS resources, such as Elastic Compute Cloud (EC2) instances, Amazon Relational Database Service (RDS) DB instances, Elastic load balancers, Elastic Block Store (EBS) volume, Elastic File System (EFS), and Simple Storage Service (S3) buckets.

The way you use a AWS integration depends on key choices you make about the function you need the integration to perform:

- Do you want to import IAM Users, and resources, or import only IAM Users or just resources?
- Do you want to perform both import and provisioning operations?
- Do you want to import all resources or only some resources?

Your answers to these questions will determine the parameters that you configure while creating AWS connection.

**Note**


The term target application refers to your (customer's) AWS account in this document.

Supported Features

The AWS integration supports the following features:

AWS Object	EIC Object	Import			Provisioning	Add or Remove Resources	Additional Configurations
		Full Import	Incremental Import	Custom Import	Lifecycle Management		
IAM Users	Accounts	Yes	No	No	Support for creating accounts and deleting accounts	Not applicable	



AWS Object	EIC Object	Import			Provisioning		Additional Configurations
		Full Import	Incremental Import	Custom Import	Lifecycle Management	Add or Remove Resources	
Resources	For more information on the entitlement types that the connector supports for the import operation, see List of	Yes	No	Yes For more information, see Customizing Entitlement Import .	Not applicable	Support for provisioning or deprovisioning of AWSRole, AWSPolicy, and AWSGroup from accounts through the Access Request System (ARS)	


AWS Object	EIC Object	Import			Provisioning		Additional Configurations
		Full Import	Incremental Import	Custom Import	Lifecycle Management	Add or Remove Resources	
	Entitlement Types.					 Info From the Release v2020.1, you can deprovision the UserInlinePolicy from the accounts through Certification.	



The AWS integration supports the following advanced features:



Feature Name	Description
Preventive controls	<p>Supports setting alerts and remediation when an action against organization policy has occurred</p> <p>For example, if the organization policy defines that access key creation is not allowed for IAM users, then on creation of an organization policy, an alert with the details is sent. As a remediation measure the newly generated access key is deleted.</p> <p>For more information, see Setting up Real Time Monitoring.</p>
Detective controls	<p>Supports configuring analytics controls to take detective actions. For example, if there are EC2 instances running with the termination protection disabled, you can take actions on such instances to stop the EC2 instances or to enable termination protection for the EC2 instances.</p> <p>For more information, see Taking Action on Sensitive Violations.</p>
Security analyzer	<ul style="list-style-type: none">• Supports continuous compliance monitoring and remediation using out-of-box security controls(250+ security controls are available) and dashboards• Supports out-of-box compliance checks against standards such as CIS, PCI, SOX, and NIST 800-53 <p>The following are some of the examples of compliance checks: Password expiration of the AWS account is disabled and IAM users have non-rotated credentials.</p>

Feature Name	Description
Map AD Groups to AWS Roles	<p>Provides AWS access visibility in EIC for federation scenarios</p> <p>For example, EIC shows all AWS roles mapped to the Active Directory (AD) group as child entitlements after access import.</p> <p>For more information, see Mapping Active Directory Groups to AWS Roles.</p>
Create Emergency roles	<p>Supports creating emergency roles</p> <p>For more information, see Creating Users with Emergency Access Roles.</p>
Certification	<p>Supports various access reviews such as User Manager review, Entitlement Owner review, and Risk-based review</p> <p>For more information, see the following topics in the <i>Enterprise Identity Cloud User Guide</i>:</p> <ul style="list-style-type: none">• Reviewing and Signing-off User Manager Campaign Certification• Reviewing and Signing-off Entitlement Owner Certification

Feature Name	Description
	<ul style="list-style-type: none"> • Reviewing and Signing-off Role Owner Campaign Certification
Rules	<ul style="list-style-type: none"> • Supports various scenarios such as Rules joiner, mover, leaver • Supports automated provisioning or deprovisioning of rules including birthright provisioning to AWS <p>For more information, see Policies in the <i>Enterprise Identity Cloud User Guide</i>.</p>
CloudKnox Integration	<div>  Info <p>This feature is available from Release v2021.0.</p> </div> <p>Supports integration with CloudKnox</p> <p>For more information, see CloudKnox Integration Overview.</p>
Discovery and onboarding	<div>  Info <p>This feature is available from CPAM 2020.0 onwards.</p> </div>

Feature Name	Description
	<ul style="list-style-type: none">• Supports on-demand onboarding (also known as bootstrapping)• Supports real-time discovery of workloads and privileged accounts• Supports real-time onboarding <p>For more information, see Manage Accounts.</p>
Console Access	<div> Info</div> <p>This feature is available from CPAM Release v2020.0.</p> <ul style="list-style-type: none">• Supports JIT role elevation with zero standing privileges• Supports zero standing accounts and privileges• Supports credential-less access (App Launcher) <p>For more information, see Privileged Access to Amazon Web Services in the <i>Cloud Privileged Access Management User Guide</i>.</p>

Feature Name	Description
CLI Access	<div> Info</div> <p>This feature is available from CPAM Release v2020.0.</p> <ul style="list-style-type: none">• Supports credentials checkout <p>For more information, see Application to Application Password Management.</p>
Workload Access (Windows, Linux and DB)	<div> Info</div> <p>This feature is available from CPAM Release v2020.0.</p> <ul style="list-style-type: none">• Supports credential-less access• Supports credentials checkout for Windows and databases• Supports JIT role elevation with zero standing privileges• Supports shared accounts <p>For more information, see Privileged Access to Amazon Web Services</p>

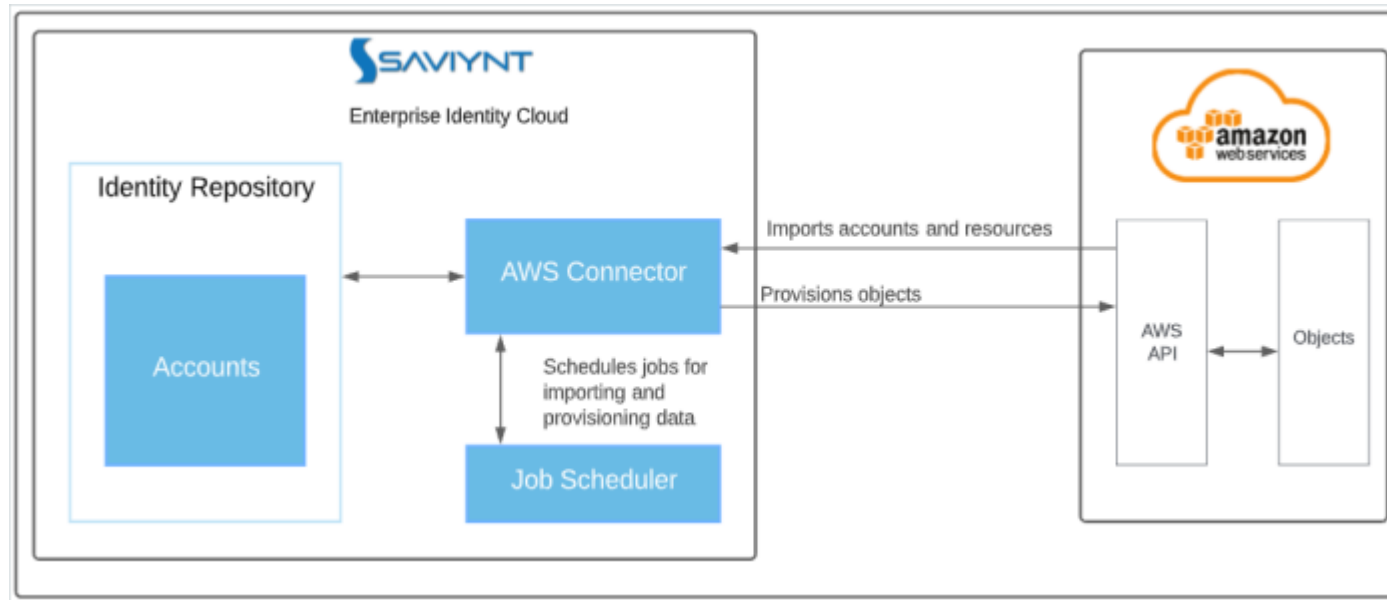
Feature Name	Description
Session Monitoring	<div> Note</div> <p>This feature is available from CPAM Release v2020.0.</p> <ul style="list-style-type: none">• Supports live session monitoring• Supports past session recording• Supports terminate session or revoke access operation• Supports SIEM integration <p>For more information, see Monitoring and Managing Privileged Sessions (Managers or Administrators).</p>
Vault Connection	<div> Info</div> <p>This feature is available from Release v2020.1.</p> <p>Supports Vault connection</p> <p>For more information, see Configuring PAM for Amazon Web Services (AWS).</p>

Supported Software Version

Software	Version
Saviynt	Release v4.6 and later

Connector Architecture

You must create an integration between EIC and the target application to perform import, provisioning, and deprovisioning operations. Saviynt can be integrated with multiple AWS accounts also. The following diagram illustrates the components involved in the integration when EIC trusts one AWS account.



- **AWS account** is the target application for which EIC manages the identity lifecycle.
- **Accounts** represent AWS IAM users imported as accounts in EIC. These accounts are provisioned to resources hosted on AWS account and the metadata associated with AWS resources such as IAM Policies, IAM Groups, IAM Roles, EC2 instances, Amazon RDS DB instances, Elastic load balancers, EBS volume, EFS, and S3 buckets.
- **Objects** are imported as entitlement types into EIC.
- **Connector** is a software component that enables communication between EIC and the target application. It provides a simplified integration mechanism and lets you create a connection with minimal connectivity information for your target application. For example, to create a connection for importing accounts from AWS, select the AWS connector and specify

the `AWS_ACCOUNT_ID` to connect it with the target application. For more information about creating a connection, see [Creating a Connection](#) in the *Enterprise Identity Cloud Administration Guide*.

- **Job Scheduler** is a software component that executes a job based on the configured schedule to perform import or provisioning operations from EIC.

When a provisioning job is triggered, it creates provisioning tasks in EIC. When these tasks are completed, the provisioning action is performed on the target application through the configured connector. If you want to instantly provision requests for completing the tasks without running the provisioning job, you must enable Instant Provisioning at the security system level and the Instant Provisioning Tasks global configuration. For more information about the jobs used by the connectors used in AWS integration, see Data Jobs and Provisioning Jobs under [Job Categories for Flat Job Control Panel](#) in the *Enterprise Identity Cloud Administration Guide*.

Preparing for the Integration

Before you configure the integration, make sure that the following prerequisites are met:

- You have clarity of integration use cases that you want to implement. For more information, see [AWS Integration Overview](#).
- You have set up a Cross Account Role. For more information, see [Setting up a Cross Account Role](#).
- You have established a trust between EIC and AWS Accounts. There are two options available for establishing the trust when you have multiple AWS accounts.
 - [Option 1: EIC trusts each AWS account individually](#)
 - [Option 2: EIC trusts the First Cross Account](#)



Note

You must make a note of the master account ID of the AWS instance where EIC is hosted. This value will be used while configuring all customer AWS accounts that Saviynt needs to trust. You can raise a Freshdesk ticket to request the master account ID.

Setting up a Cross Account Role

A Cross Account Role enables the Saviynt AWS instance on which EIC is installed to trust your AWS accounts. This trust defines the permissions to allow EIC to connect to your AWS accounts.

In order to setup the Cross Account Role, a Stack is created. The Stack is the collection of AWS resources that is managed by EIC. The Stack is created by applying preconfigured AWS CloudFormation templates. You can create a Stack through a wizard by performing the following steps.



Note

All the steps must be performed in the same order on all the scoped AWS accounts.

1. [Selecting Stack Templates](#) (Mandatory)
2. [Specifying Stack Details](#) (Mandatory)
3. [Configuring Stack options](#) (Optional)
4. [Reviewing your Stack](#) (Mandatory)

Selecting Stack Templates

The AWS CloudFormation template (CF template) is a formatted text file in the JSON format that describes your AWS infrastructure. In EIC, the following CF templates listed in the table below are widely used. The Security Analyzer template is provided by default.

However, you must choose a template based on your requirement. See **Step 1** of [Option 1: EIC trusts each AWS account individually](#) or [Option 2: EIC trusts the First Cross Account](#)


Table 1: CF templates

Template Name	Permissions	Access Type	Template URL
Security Analyzer	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none">• Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2.• Produce a detailed list of security findings prioritized by the level of severity.	Read-only	Link to template
Security Analyzer + IGA	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none">• Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2• Produce a detailed list of security findings prioritized by the level of severity	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
	<ul style="list-style-type: none"> • Create or delete accounts • Add or remove access from groups, IAM role, and policies 		
Security Analyzer + IGA + PAM	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none"> • Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2 • Produce a detailed list of security findings prioritized by the level of severity • Create or delete accounts • Add or remove access from groups, IAM role, and policies • Create and update security group 	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
	<ul style="list-style-type: none"> • Create and attach policies • Create and associate IAM instance profile 		
Security Analyzer + IGA + Detective action	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none"> • Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2 • Produce a detailed list of security findings prioritized by the level of severity • Perform detective actions 	Read and Write	Link to template
Security Analyzer + IGA + Detective	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none"> • Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2 	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
action + PAM	<ul style="list-style-type: none"> • Produce a detailed list of security findings prioritized by the level of severity • Perform detective actions • Create and update security group • Create and attach policies • Create and associate IAM instance profile 		
Security Analyzer + IGA + Real Time Monitoring with	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none"> • Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2 • Produce a detailed list of security findings prioritized by the level of severity • Create or delete accounts 	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
Detective action	<ul style="list-style-type: none"> • Add or remove access from groups, IAM role, and policies • Perform preventive actions and detective actions • Detect any suspicious activity, set alarms, take automated actions, troubleshoot issues, and discover insights <div>  Note <p>Download the SaviyntIncSetup.bat file from the S3 bucket location and configure the AWS CLI with correct permissions. For more information on the template, click here.</p> <p>For more information on setting CloudWatch event for real time monitoring (preventive actions), see Setting up Real Time Monitoring.</p> </div>		

Template Name	Permissions	Access Type	Template URL
Security Analyzer + IGA + Real Time Monitoring with Detective action + PAM	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none"> • Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2 • Produce a detailed list of security findings prioritized by the level of severity • Create or delete accounts • Add or remove access from groups, IAM role, and policies • Perform preventive actions and detective actions • Detect any suspicious behavior, set alarms, take automated actions, troubleshoot issues, and discover insights • Create and update security group 	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
	<ul style="list-style-type: none"> • Create and attach policies • Create and associate IAM instance profile 		

You can specify the Amazon S3 URL (CloudFormation template) based on the type of role you are performing.

Specifying Stack Details

You can specify the Stack name and parameter values used in the template. See **Step 2** of [Option 1: EIC trusts each AWS account individually](#) or [Option 2: EIC trusts the First Cross Account](#)

Configuring Stack options

You can add additional options for your Stack such as specifying tags. Tags are arbitrary key-value pairs used to identify your Stack. A key and a value can include alphanumeric characters or spaces. Tag keys can be 127 characters long and tag values can be 255 characters long. See **Step 3** of [Option 1: EIC trusts each AWS account individually](#) or [Option 2: EIC trusts the First Cross Account](#)

Reviewing your Stack

You can review the values entered while creating the Stack before your Stack is launched. See **Step 4** of [Option 1: EIC trusts each AWS account individually](#) or [Option 2: EIC trusts the First Cross Account](#)

Establishing Trust between EIC and AWS Accounts

To connect EIC with your AWS accounts, you must establish a trust between them. To establish a trust between EIC and multiple AWS accounts, set up a Cross Account Role for each AWS account that you want to integrate with EIC. Each AWS account is created as an AWS connection in EIC.

Option 1: EIC trusts each AWS account individually

When you have to establish a trust between EIC and multiple AWS accounts, use this option only if you want to establish trust with each AWS account separately. Otherwise, use [Option 2: EIC trusts the First Cross Account](#)

Here, you are establishing a trust between Saviynt's AWS account and your accounts: AWS Account1, AWS Account2, AWS Account3, and AWS Account4.

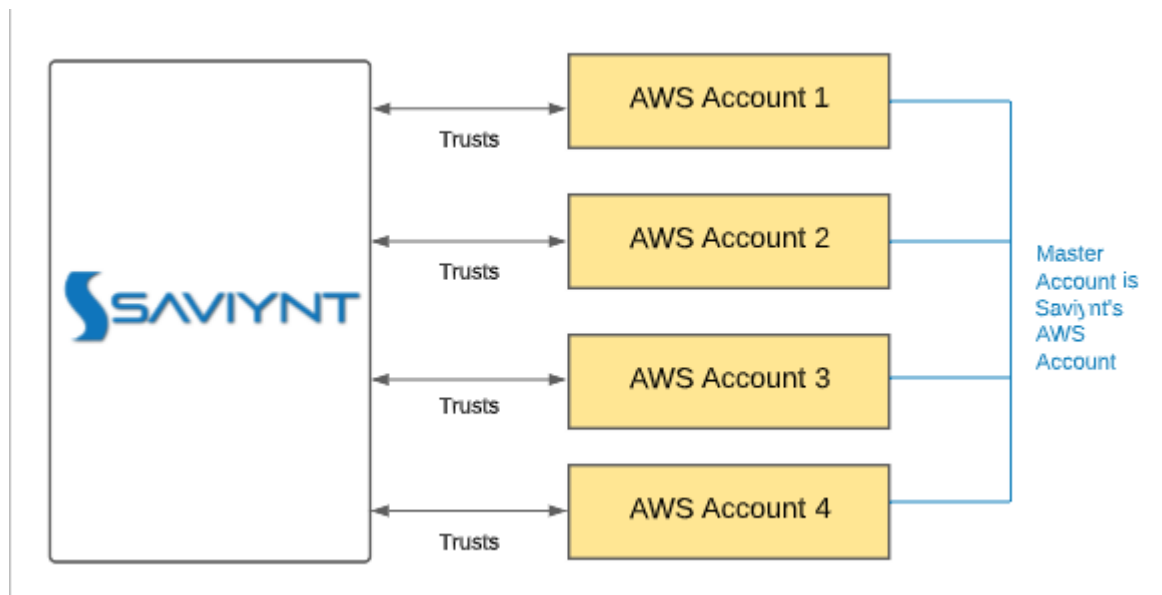


Figure: Establish trust between EIC and each AWS Account

Step 1: Create a Stack:

1. Open the [AWS console](#).
2. Log in to AWS console using AWS admin credentials.
3. In the **Home** page that displays, select **CloudFormation** under **Services**.
4. In the **Cloud Formation** page, click **Create Stack**.

5. In the **Create Stack** wizard that displays, select **Specify an Amazon S3 template URL** as a template source.
6. Specify the Amazon S3 URL based on the type of role you are performing. This URL will create and configure the role and policy required for cross account import. For more information on templates, see the [Table 1: CF templates](#) above.
7. Click **Next** to navigate to **Step 2: Specify stack details**.

Step 2: Specify stack details:

1. In the **Specify Stack Details** page, specify the following details.

Field	Description	Mandatory?	Default Value or Suggested Value
Stack Name	Specify the Stack name. The Stack name is an identifier that helps you find a particular Stack from a list of Stacks. A Stack name can contain only alphanumeric characters (case-sensitive) and hyphens. It must start with an alphabetic character and cannot be longer than 128 characters.	Yes	-

Field	Description	Mandatory?	Default Value or Suggested Value
MasterAccID	Specify the AWS account ID of the master account that your AWS account will trust. Master account is the AWS account where EIC is hosted. An AWS account ID is a 12-digit number used to construct Amazon Resource Names (ARNs).	Yes	533811351211
EXTERNAL ID	Specify the external ID to provide an additional security layer for the connection.	Mandatory for roles created with External ID. For example, cross account role.	345687

2. Click **Next** to navigate to **Step 3: Configure stack options**.
3. Note the EXTERNAL ID. You will need this value while creating AWS Connection.

Step 3: Configure Stack options:

1. In the **Configure Stack Options** page, specify appropriate tags.
2. Click **Next** to navigate to **Step 4: Review the stack**.

Step 4: Review the stack:

1. In the **Review** page, review the Stack details.
2. [Optional] Click **Edit** on the appropriate section to make changes prior to the Stack launch.
3. Select **I acknowledge that AWS Cloud Formation might create IAM resources** to acknowledge and launch the IAM resources.
4. Click **Create Stack**.

The Stack gets created.

Note that the Stack status is `CREATE_COMPLETE`.

5. [Optional] Click **Outputs** to view your Stack's output.
6. Note the following details. You will use them for establishing a connection with AWS:
 - `CROSS_ACCOUNT_ROLE_ARN`: This is listed under the key value **SaviyntAWSRole** in the Stack output.
 - `MasterAccID`: See field description table in Step 2 of the [Option 1: EIC trusts each AWS account individually](#) section for the default value.

- ExternalID: See field description table in Step 2 of the [Option 1: EIC trusts each AWS account individually](#) section for the default value.

Option 2: EIC trusts the First Cross Account

When you have to establish a trust between EIC and multiple AWS accounts, use this option only if you want to establish trust with the First Cross Account. For the remaining AWS accounts, you can establish a trust with the First Cross Account instead of EIC.

Here, you are establishing a trust between the AWS account of Saviynt and your First Cross Account. The First Cross Account establishes a trust with the remaining accounts: AWS Account1, AWS Account2, AWS Account3, and AWS Account4.

Raise a Freshdesk ticket to contact the Saviynt Support team for adding the `aws.saas.firstCrossAccountRoleArn` configuration in the `externalconfig.properties` file.

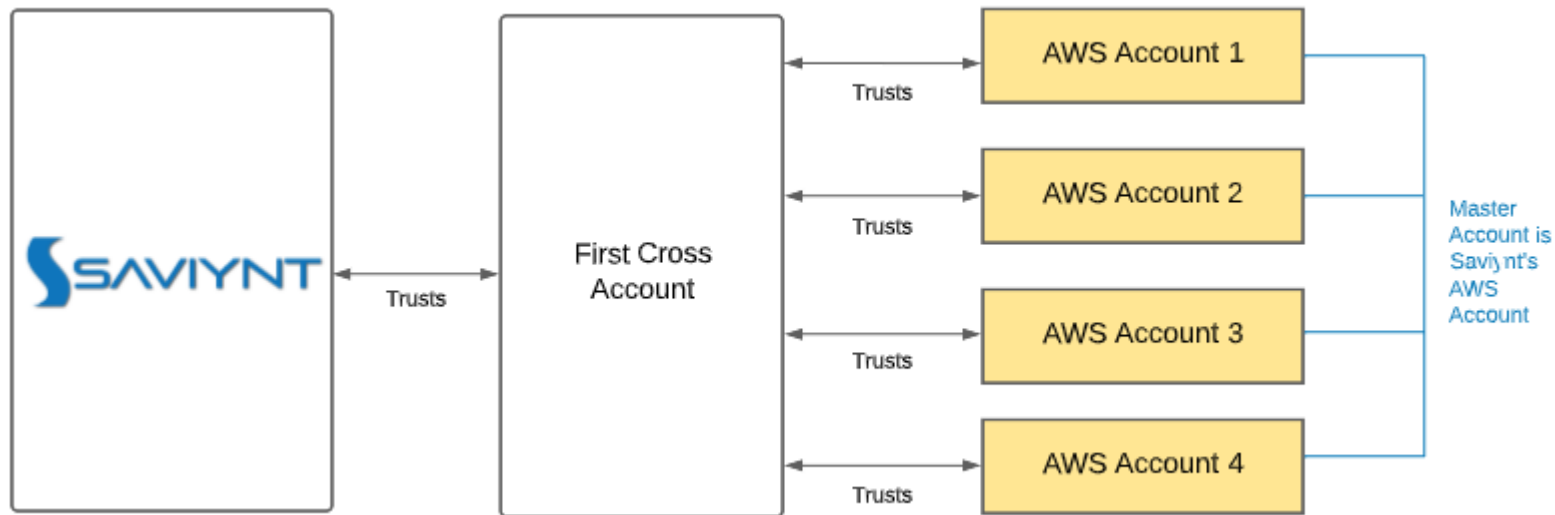


Figure: Establish trust between EIC and First Cross Account

Step 1: Create a Stack:

1. Open the [AWS console](#).
2. Log in to AWS console using AWS admin credentials.
3. In the **Home** page that displays, select **CloudFormation** under **Services**.
4. In the **Cloud Formation** page, click **Create Stack**.


5. In the **Create Stack** wizard that displays, select **Specify an Amazon S3 template URL** as a template source.
6. Specify the Amazon S3 URL based on the type of role you are performing. This URL creates and configures the role and policy required for cross account import. For more information about templates, see the [Table 1: CF templates](#) above. The following table provides the template URL required for the **first cross account**.



Note

If permission does not exist in the CloudFormation template, run the CloudFormation template in all AWS accounts.

Template Name	Permissions	Access Type	Template URL
Security Analyzer	<p>The template includes permissions for the following:</p> <ul style="list-style-type: none">• Assess applications for vulnerabilities and deviations from compliance standards such as PCI, CIS, and SOC2• Produce a detailed list of security findings prioritized by the level of severity <p>Write permission to establish a trust between first cross account and the remaining AWS accounts</p>	Read and Write	Link to template

Template Name	Permissions	Access Type	Template URL
	 Note Use this template for the first cross account only. For the remaining AWS accounts use this template .		

7. Click **Next** to navigate to **Step 2: Specify stack details**.

Step 2: Specify stack details:

1. In the **Specify Stack Details** page, specify the following details.

Field	Description	Mandatory?	Default Value or Suggested Value
Stack Name	Specify the Stack name. The Stack name is an identifier that helps you find a particular Stack from a list of Stacks. A Stack name can contain only	Yes	-

Field	Description	Mandatory?	Default Value or Suggested Value
	alphanumeric characters (case-sensitive) and hyphens. It must start with an alphabetic character and cannot be longer than 128 characters.		
MasterAccID	<p>Master account is the AWS account where EIC is hosted. An AWS account ID is a 12-digit number used to construct Amazon Resource Names (ARNs).</p> <p>Saviynt trusts the First Cross Account, so for remaining cross accounts that the First Cross Account trusts, specify the AWS account ID of the First Cross Account.</p>	Yes	533811351211
EXTERNAL ID	Specify the external ID to provide an additional security layer for the connection.	Mandatory for roles created with External ID. For	345687

Field	Description	Mandatory?	Default Value or Suggested Value
		example, cross account role.	

2. Click **Next** to navigate to **Step 3: Configure stack options**.
3. Note the EXTERNAL ID. You will need this value while creating AWS Connection.

Step 3: Configure Stack options:

1. In the **Configure Stack Options** page, specify appropriate tags.
2. Click **Next** to navigate to **Step 4: Reviewthe stack**.

Step 4: Review the stack:

1. In the **Review** page, review the Stack details.
2. [Optional] Click **Edit** on the appropriate section to make changes prior to the Stack launch.
3. Select **I acknowledge that AWS Cloud Formation might create IAM resources** to acknowledge and launch the IAM resources.

4. Click **Create Stack**.

The Stack gets created.

Note that the Stack status is CREATE_COMPLETE.

5. [Optional] Click **Outputs** to view your Stack's output.

6. Note the following details. You will use them for establishing a connection with AWS:

- CROSS_ACCOUNT_ROLE_ARN: This is listed under the key value **SaviyntAWSRole** in the Stack output.
- MasterAcclID: See field description table in Step 2 of the [Option 2: EIC trusts the First Cross Account](#) section for the default value.
- ExternalID: See field description table in Step 2 of the [Option 2: EIC trusts the First Cross Account](#) section for the default value.

Updating a Stack

You can update an existing Stack instead of deleting and creating a new Stack. For example, if you have a Stack for security analyzer, you can update the Stack for IGA capabilities.

When you update a Stack details, such as connection parameter values, and template. The AWS CF template compares these changes with the current state of your Stack and updates only the changed permissions.

To update existing stack details:

1. Open the [AWS console](#).
2. Log in to AWS console using AWS admin credentials.
3. In the **Home** page that displays, select **CloudFormation** under **Services**.
4. Select a Stack to be updated and click **Update**.
5. Select **Replace current template** under **Prepare template**.
6. Select **Amazon S3 URL** under **Template source**.
7. Specify the Amazon S3 URL based on the type of role you are performing.

Refer Step 3 in the *Selecting a Stack Template* section.

8. Click **Next** to navigate to **Step 2: Specify stack details**. Ignore this step if you do not need to update the Stack details.
9. Click **Next** to navigate to **Step 3: Configure stack options**. Ignore this step if you do not need to update the Stack options.
10. Select **I acknowledge that AWS Cloud Formation might create IAM resources** to acknowledge creating IAM resources.
11. Click **Create Stack**.

The Stack gets updated.

Note that the Stack status is UPDATE_COMPLETE.

Configuring the Integration for Deployments on Public AWS Account

This section provides high-level details about creating an integration for deployments on public AWS account.

1. Set up a Cross Account Role. For more information, see [Setting up a Cross Account Role](#).
2. Establish trust between EIC and AWS Accounts. For more information, see [Establishing Trust between EIC and AWS Accounts](#).
3. Create an integration by specifying the connection parameters in the user interface. Ensure that **Connection Type** is selected as **AWS** and you select **PULL_GOV_REGION_ONLY** as **No**. For more information, see [Creating a Connection using the User Interface](#).

Or

Create an integration using the testConnection API. For more information, see [Creating a Connection using the testConnection API](#).

Configuring the Integration for Deployments on GovCloud Account

This section provides high-level details about creating an integration for deployments on GovCloud Account.

1. Set up a Cross Account Role. For more information, see [Setting up a Cross Account Role](#).
 2. Establish trust between EIC and AWS Accounts. For more information, see [Establishing Trust between EIC and AWS Accounts](#).
 3. Create an integration by specifying the connection parameters in the user interface. Ensure that **Connection Type** is selected as **AWS** and you select **PULL_GOV_REGION_ONLY** as **Yes**. For more information, see [Creating a Connection using the User Interface](#).
- Or
- Create an integration using the testConnection API. For more information, see [Creating a Connection using the testConnection API](#).

Configuring the Integration for Deployments on Azure Subscription

This section provides high-level details about creating an integration for deployments on Azure subscription.

1. Establish trust between EIC and AWS Accounts. For more information, see [Establishing Trust between EIC and AWS Accounts](#).
 2. Create an integration by specifying the connection parameters in the user interface. Ensure that **Connection Type** is selected as **AWS** and you select **PULL_GOV_REGION_ONLY** as **No**. For more information, see [Creating a Connection using the User Interface](#).
- Or

Create an integration using the `testConnection` API. For more information, see [Creating a Connection using the testConnection API](#).



Note

You need not create a cross-account role in AWS when EIC is deployed on an Azure subscription.

To create this type of integration, you must specify the following additional parameters while creating an AWS connection:

- **AWS_ACCESS_KEY** - Use this parameter to specify the access key of the service account used to invoke the AWS Security Token Service (STS).
- **AWS_ACCESS_SECRET_PASSWORD** - Use this parameter to specify the secret credentials of the service account used to make API requests to AWS.

Configuring the Integration for Importing Accounts

This section provides high-level details about creating an integration for importing accounts.

1. Perform the prerequisite steps required for preparing the target application for integration. For more information, see [Preparing for Integration](#).
2. Create an integration by specifying values for the connection parameters. Ensure that the connection type is selected as **AWS**. For more information, see [Creating a Connection using the User Interface](#).
3. Create a security system. For more information, see [Creating a Security System](#) in the *Enterprise Identity Cloud Administration Guide*.
4. Create an endpoint for the security system. For more information, see [Creating Endpoints](#) in the *Enterprise Identity Cloud Administration Guide*.
5. Configure the **Application Data Import (Multi-Threaded)** job to import accounts. For more information, see Data in [Job Categories for Flat Job Control Panel](#) in the *Enterprise Identity Cloud Administration Guide*.

Creating a Connection using the User Interface

Connection refers to the configuration setup for connecting EIC to target applications. For more information about the procedure to create a connection, see [Creating a Connection](#) in the *Enterprise Identity Cloud Administration Guide*.

The AWS connector uses the following parameters for establishing a connection with AWS accounts.

Field	Description	Mandatory?	Default Value or Suggested Value
Connection Name	Specify the name to identify the connection.	Yes	-
Connection Description	Specify the description for the connection.	No	-
Connection Type	Select the connection type as AWS .	Yes	AWS
Email Template	Specify the email template for sending notifications. Email notifications are triggered to inform a user about an action that has been performed and if it demands an immediate action from the user.	No	-

Field	Description	Mandatory?	Default Value or Suggested Value
Default SAV Role	Specify the SAV role to assign for the connection. The SAV role is a Saviynt role that assigns specific access to users. For example, if a user is assigned the ROLE_ADMIN role, the user has access to all the sections of EIC. This parameter is valid only for importing users.	No	-
AWS_ACCOUNT_ID	Specify the 12-digit AWS Account ID of your account. See field description of MasterAccID in Step 2 of Option 1: EIC trusts each AWS account individually or Option 2: EIC trusts the First Cross Account.	Yes	533811351211
ADMIN_EMAIL	Specify the default email address of the admin user.	Yes	-

Field	Description	Mandatory?	Default Value or Suggested Value
CROSS_ACCOUNT_ROLE_ARN	<p>Specify the cross account role name shown in the key value SaviyntAWSRole.</p> <p>See CROSS_ACCOUNT_ROLE_ARN in Step 4 of Option 1: EIC trusts each AWS account individually or Option 2: EIC trusts the First Cross Account.</p>	Yes	arn:aws:iam::533811351211:role/AWS-SaviyntAWSRole-13G55PQK517VS
AWS_STACK_ROLE_NAME	Specify the AWS Stack role name.	No	-
EXTERNAL ID	<p>Specify the external ID that provides an additional security layer for the connection.</p> <p>See field description of EXTERNAL ID in Step 2 of</p>	Mandatory for roles created with External ID, for	345687

Field	Description	Mandatory?	Default Value or Suggested Value
	Option 1: EIC trusts each AWS account individually or Option 2: EIC trusts the First Cross Account.	example, cross account role.	
PULL_GOV_REGION_ONLY	Select this parameter value as No.	Yes	No

Creating a Connection using the testConnection API

You can create an AWS connection using the testConnection API. Use this API if you have multiple AWS accounts or if you want to automate the onboarding of AWS accounts.

Sample request body:

JSON

```
{
  "systemname": "Wint-API-may29",
  "connectiontype": "AWS",
  "connectionName": "Wint-API-may29",
  "saveconnection": "Y",
  "AWS_ACCOUNT_ID": "533811351211",
  "ADMIN_EMAIL": "test.email@testdomain.com",
  "CROSS_ACCOUNT_ROLE_ARN": "arn:aws:iam::533811351211:role/TestCorelogicAWS-SaviyntAWSRole-13G55PQK517VS",
  "AWS_STACK_ROLE_NAME": "awsv5-devclone-APP-SaviyntAWSRole-1LGTDNRN3740PL",
  "ConnectionDescription": "Api connection",
  "fullorincremental": "Full",
  "EXTERNAL_ID": "345678",
  "PREVENTATIVECONTROL_TURNED_ON": "SELECT",
  "accountsoraccess": "access"
}
```

where,

`systemname` is the security system name.

`saveconnection` indicates whether you want to save the connection. Specify the value as `Y` to save the connection.

`fullorincremental` indicates whether you want to perform full import or incremental import. Specify the value as **Full** to perform full import.

`accountsoraccess` indicates whether you want to import accounts or access. Specify the value as **access** to import access.

For more information about the request URL and response, see [API document](#).

Configuring Account Import

This section describes the configuration for the following import use cases:

- [Creating Users during Import](#)
- [Configuring the Integration for Importing Accounts](#)

Creating Users during Import

To create users during account import, select the **CREATEUSERS** parameter as **Yes**. If you select **No**, users are imported from the HR system.

Configuring the Integration for Importing Entitlements

This section provides high-level details about creating an integration for importing entitlements.

1. Perform the prerequisite steps required for preparing the target application for integration. For more information, see [Preparing for Integration](#).
2. Create an integration by specifying values for the connection parameters. Ensure that the connection type is selected as **AWS**. For more information, see [Customizing Entitlement Import](#).
3. Create a security system. For more information, see [Creating a Security System](#) in the *Enterprise Identity Cloud Administration Guide*.
4. Create an endpoint for the security system. For more information, see [Creating Endpoints](#) in the *Enterprise Identity Cloud Administration Guide*.
5. Configure the **Application Data Import (Multi-Threaded)** job to import accounts. For more information, see Data in [Job Categories for Flat Job Control Panel](#) in the *Enterprise Identity Cloud Administration Guide*.



Note

When you schedule an import job to import resources, the connector imports both IAM users and resources.

Customizing Entitlement Import

You can include or exclude entitlement types while importing entitlements using the **Import Config** parameter.

- Specify the list of entitlement types to be included for import under `importEntTypes`.
- Specify the list of entitlement types to be excluded from import under `excludeEntTypes`.

To customize the entitlement import to include or exclude entitlement types, select the **Import Type** as **Custom_access** while configuring the job trigger.

The following resources can be specified in **Import Config**:

- IAMPolicy
- AWSRole
- AWSGroup
- EC2Instance
- SecurityGroup
- AMI
- ElasticLoadBalancer
- DhcpOption
- VPC

- Subnet
- NACL
- S3Bucket
- EBSVolume
- EBSSnapshot
- DBSecurityGroup
- RsDbInstance
- RouteTable
- VpcPeering
- InternetGateway
- CloudTrail
- NetworkInterface
- RedShiftClusterSecurityGroup
- RedShiftCluster
- ElasticIP
- CloudFormation
- EncryptionKey

- NatGateway
- SnsTopic
- SQS
- AWSConfig
- DynamoDB
- VpcFlowLog
- Glacier
- RDSSnapshot
- EFS
- MountTarget
- ReputedIP
- ElasticSearch
- CloudFormationTemplatesFromS3
- EMR
- VpcEndpoint
- VirtualMFADevice
- CloudWatchLogGroup

- CloudWatchAlarm
- Workspace
- Directory
- WorkspaceBundle
- AppELB
- ACM
- AutoScaling
- LaunchConfig
- Route53
- CloudFront
- RDSEventSubscription
- AWSLambda
- GuardDuty
- WAFCondition
- WAFWebACL
- RedShiftParameterGroup
- WAFRule

- AWSAccountSettings

Sample value of **Import Config** to import IAM entitlements only:

JSON

```

{
  "importEntTypes": {
    "IAMPolicy": {},
    "AWSRole": {},
    "AWSGroup": {}
  },
  "excludeEntTypes": {
    "EC2Instance": {"storeIAMRoleForEC2Instance": "true"},
    "SecurityGroup": {},
    "AMI": {},
    "ElasticLoadBalancer": {},
    "DhcpOption": {},
    "VPC": {},
    "Subnet": {},
    "NACL": {},
    "S3Bucket": {},
    "EBSVolume": {},
    "EBSSnapshot": {},
    "DBSecurityGroup": {},
    "RdsDbInstance": {},
    "RouteTable": {},
    "VpcPeering": {},
    "InternetGateway": {},
    "CloudTrail": {},
    "NetworkInterface": {},
    "RedShiftClusterSecurityGroup": {},

```

where,

`storeIAMRoleForEC2Instance` imports the IAM role of the EC2 instance if set to true. If set to false, the IAM role of the EC2 instance is not stored in the customproperty of the EC2 instance.

Configuring the Integration for Provisioning Accounts and Entitlements

This section provides high-level details about creating an integration for provisioning accounts and entitlements.

1. Perform the prerequisite steps required for preparing the target application for integration. For more information, see [Preparing for Integration](#).
2. Create an integration by specifying values for the connection parameters. Ensure that the connection type is selected as **AWS**. For more information, see [Creating a Connection using the User Interface](#).
3. Create a security system. For more information, see [Creating a Security System](#) in the *Enterprise Identity Cloud Administration Guide*.
4. Create an endpoint for the security system. For more information, see [Creating Endpoints](#) in the *Enterprise Identity Cloud Administration Guide*.
5. Specify the Amazon S3 URL based on the type of role you are performing. For more information on templates, see [Table 1](#).
6. Create an ARS request. For detailed information about performing provisioning tasks, see [Requesting New Access](#) in the *Enterprise Identity Cloud User Guide*.

7. Run **Provisioning Job (WSRETRY)** to complete the provisioning operation. For more information about Provisioning Jobs, see [Job Categories for Flat Job Control Panel](#) in the *Enterprise Identity Cloud Administration Guide*.

Taking Action on Sensitive Violations

You can take the required action on sensitive violations performed by the AWS accounts. To do this, use the Allowed Actions feature in the Analytics module. You can perform allowed actions, such as executing Lambda functions, and stopping the EC2 instance. For more information on the list of allowed actions that EIC supports and the sample query to be used for different allowed actions, see [Configuring Allowed Actions](#) in the *Enterprise Identity Cloud Administration Guide*.

For more information on creating an AWS connection, see [Creating a Connection using the User Interface](#).

This section provides high-level details about creating and executing an allowed action:

1. Create an **Analytics Configurations Version 2** record **Using SQL Query**. For more information, see [Creating Elasticsearch-based Analytics Controls \(Version 2\) using SQL Query](#) in the *Enterprise Identity Cloud Administration Guide*.
2. Enter the SQL query.

The Analytics query must have the columns given below:

`entvaluekey` is the entitlement value key of the entitlement to which the account will be assigned.

- `externalConnectionKey` is the connection key of the AWS Connection.
- `lambdaName` is the name of the Lambda function to be executed.

- `region` is the region where the Lambda Function is present.
- `inputJson` is the JSON body to be sent as a test event to the Lambda Function.
- `lambdaAccountId` is the AWS Account ID of the AWS account where the Lambda function is present.

When Execute Lambda Function is configured as an allowed action, enter `Execute Lambda Function as Default_Action_For_Analytics` in the query.

The sample query given below executes a Lambda function to detect the access keys created by users:

JSON


```

Select excv.attributevalue as AccountID,ac.CUSTOMPROPERTY4 as 'Account ARN',ac.name as 'IAM
User',ac.CREATED_ON as 'AccountCreateDate',substring_index(acat.ATTRIBUTE_VALUE,',',1) as 'A
ccessKey',
substring_index(substring_index(acat.ATTRIBUTE_VALUE,',',2),',',-1) as
'AccessKey CreateDate', NULL as entvaluekey,'us-east-1' as region,'Execute
Lambda Function' as 'Default_Action_For_Analytics', 'deleteaccesskey' as lambdaName,
exc.EXTERNALCONNECTIONKEY as externalConnectionKey,CONCAT('{"awsiamuser": "',
ac.name , '"',"accesskeyid": "',substring_index(acat.ATTRIBUTE_VALUE,',',1),
'",',"crossaccountrole":','"arn:aws:iam::661222050851:role/saviynt-analyzer-poc-SaviyntAWSRole
-1M7ICRQQT66I0"}')
as inputJson, '533811351211' as lambdaAccountId from accounts ac
Inner join securitysystems sc on ac.SYSTEMID = sc.systemkey
Inner Join externalconnection exc on sc.externalconnection = exc.externalconnectionkey
Inner Join externalconnectiontype exct on exc.externalconnectiontype = exct.externalconnecti
ontypekey
and exct.connectiontype = 'AWS'
Inner join externalconnattvalue excv on excv.connectiontype = exc.externalconnectionkey
and excv.attributekey ='AWS_ACCOUNT_ID'
inner join account_attributes acat on ac.accountkey = acat.accountkey
and acat.attribute_name = 'accessKeyMetaData_Event'
where substring_index(substring_index(acat.ATTRIBUTE_VALUE,',',2),',',-1) >
(Select ah.UPDATEDate from analytics_analyticshistory as ah,analyticsconfig as aconf
where ah.ANALYTICSCONFIG=aconf.ANALYTICSKEY and aconf.ANALYTICSNAME = 'Access keys Creation'
order by ah.analyticshistory desc limit 1);

```

3. Select **Execute Lambda Function** as **Allowed Action** for AWS.

The screenshot shows a 'Create New Analytics' form with the following fields and values:

- Analytics Name ***: Execute Lambda Function
- Status**: Active (dropdown)
- Analytics Query * ⓘ**:

```
an.analyticsconfig as aconf where
ah.ANALYTICSCONFIG=aconf.ANALYTICSKEY and
aconf.ANALYTICSNAME = 'Access keys Creation' order by
ah.analyticshistory desc limit 1);
```
- Description**: (empty text area)
- Allowed Action**:
 - ☒ Accept
 - ☒ Revoke
 - ☒ Further Review
 - ☒ Execute Lambda Function
- Tags**: Select tags (text input)
- Category***: Test Category (dropdown)
- Sub Category**: (empty dropdown)
- Risk***: Medium (dropdown)
- Recommendations**: (empty text area)

Figure: Analytics query and allowed action

4. Run the analytics control.
5. View the values obtained from the database.

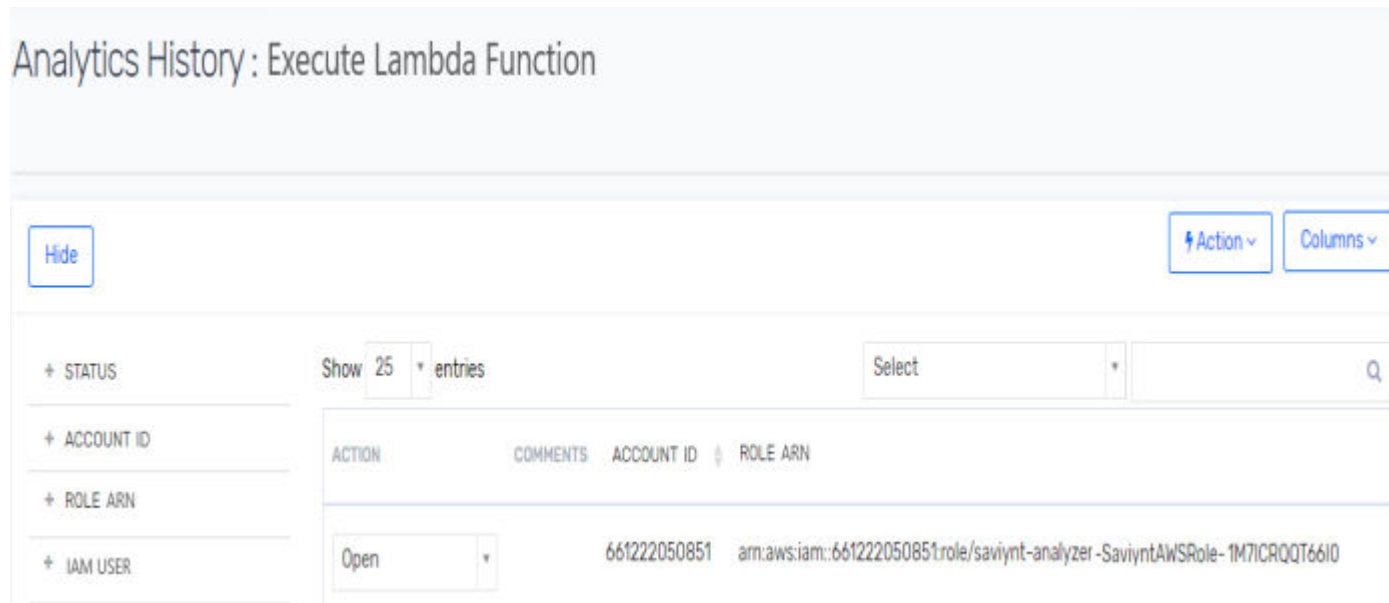


Figure: Analytics History page

6. Select an **Action**, enter comments, and click **Save**.

The selected **Allowed Action** is performed in EIC.

Mapping Active Directory Groups to AWS Roles

To view the Active Directory group mapped to AWS roles in Active Directory (AD), define the mapping rule in the **federatedADJSON** parameter. The rule explains the relationship between an AD group and AWS roles. When you run an AWS access import, the AD group mapped to AWS roles are shown as child entitlements. For more information on creating an AWS connection, see [Creating a Connection using the User Interface](#).

Sample value for **federatedADJSON**:

Example:

JSON

```
{ "endpoint": "ActiveDirectory", "secsys": "ActiveDirectory", "entitlementType": "memberOf", "rule": "substring(bstring(ev1.ENTITLEMENT_VALUE, 7, (position(',' IN ev1.ENTITLEMENT_VALUE) - 7))=lower(SUBSTRING_INDEX(ev2.ENTITLEMENT_VALUE, 'ADFS-533811351211-', -1)));"} }
```

where,

`endpoint` is the name of the AD endpoint.

`secsys` is the name of the AD security system.

`entitlementType` is the name of the entitlement type of the AD group, i.e. `memberOf`.

`rule` is the rule defined for mapping the AD group to the AWS role. Here, `ev1.ENTITLEMENT_VALUE` is the name of the AD group, and `ev2.ENTITLEMENT_VALUE` is the name of the AWS role.

For example, if `ev1.ENTITLEMENT_VALUE` is `CN=awsdeveloper,OU=AWS,DC=corpAD,DC=saviynt,DC=com` then the left part of the rule query `substring(ev1.ENTITLEMENT_VALUE,7,(position(',', ' IN ev1.ENTITLEMENT_VALUE) - 7))` returns the value as `developer`.

And if `ev2.ENTITLEMENT_VALUE` is `ADFS-533811351211-Developer`, then the right part of the rule query `lower(SUBSTRING_INDEX(ev2.ENTITLEMENT_VALUE, 'ADFS-533811351211-', -1))` returns the value as `developer`.

As both the left part and right part of the rule query matches, EIC shows the AD group `CN=awsdeveloper,OU=AWS,DC=corpAD,DC=saviynt,DC=com` as parent entitlement and the AWS role `ADFS-533811351211-Developer` as the child entitlement.

Perform the following steps to view the mapping details in EIC:

1. Log in to EIC.
2. Select **Admin > Identity Repository > Entitlements**.
3. In the **Entitlement List** page that displays, select the required entitlement. For example, `CN=awsdeveloper,OU=AWS,DC=corpAD,DC=saviynt,DC=com`.
4. In the **Entitlement Detail** page, view the details of the AD group.

The screenshot displays the 'Entitlement Details' page in the AWS IAM console. The breadcrumb trail at the top reads: 'Groups > Identity Repository > Entitlements Show > CN=jwsdevelopers,OU=AWS,DC=corpAD,DC=saviynt,DC=com'. The page has a navigation bar with tabs: 'Entitlement Detail' (selected), 'Other AD Groups', 'Name', 'Accounts', 'SOD', 'Child Entitlement', 'Associated Entitlement', 'Other Entitlement Details', 'Entitlement AD Groups', 'Group Rules', and 'History'.

The main content area is divided into two columns. The left column contains the following fields:

- Entitlement name:** CN=jwsdevelopers,OU=AWS,DC=corpAD,DC=saviynt,DC=com
- Directory System:** Active Directory
- Description:** Federated AD group that grants Developers EC2 full access on AWS Dev account
- Update Date:** Jul 27, 2020 16:17:01
- Is Official:** None
- Status:** Active
- Privileged:** None
- Module:** Default
- Priority ID:**

The right column contains the following fields:

- Entitlement Type:** Groups
- Group:** Active Directory
- Display Name:** jwsdevelopers
- Entitlement Domain:**
- Is Official:** None
- Role:** None
- Groupset:** None
- Access:** Default

Figure: Entitlement details of parent entitlement

5. Select **Child Entitlement** to view the list of the AWS roles mapped to the AD group after access import.

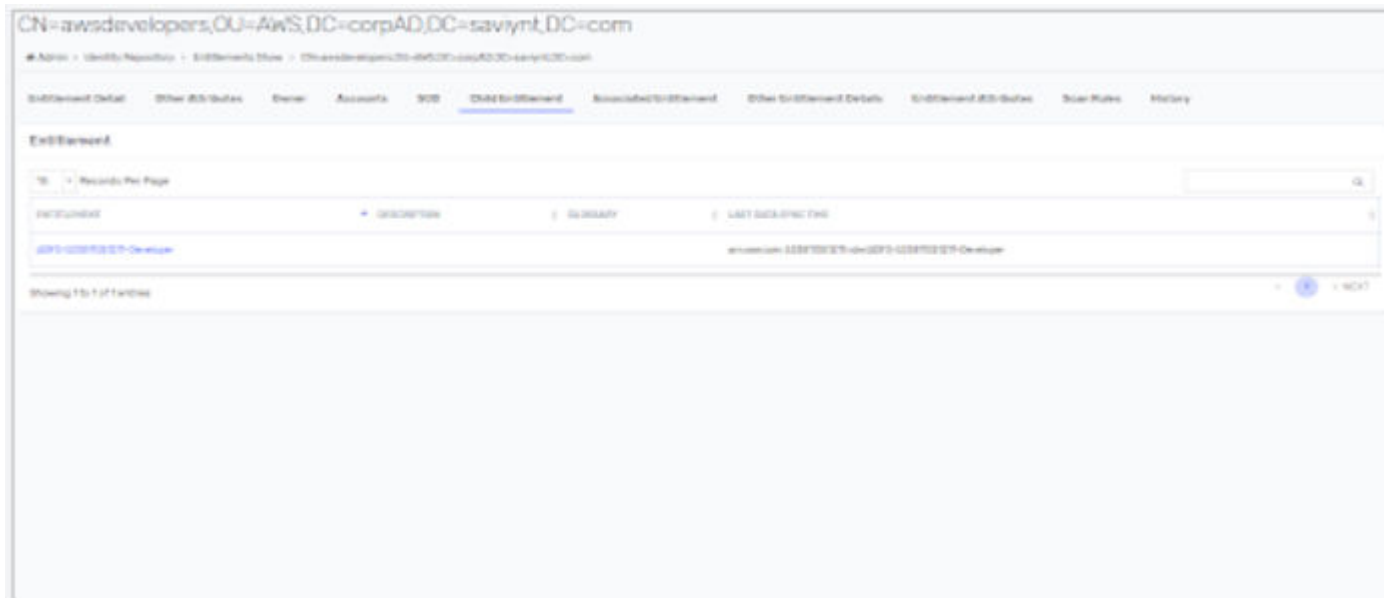


Figure: List of child entitlements

6. Select a child entitlement to view the details of the AWS role.

The screenshot displays the AWS IAM console interface for a specific entitlement. The breadcrumb navigation at the top reads: **ADFS-533811351211-Developer** > **Admin** > **Identity Repository** > **Entitlements Store** > **ADFS-533811351211-Developer**. Below this, a series of tabs are visible: **Entitlement Detail** (selected), **Other Attributes**, **Owner**, **Accounts**, **ROB**, **Child Entitlement**, **Associated Entitlement**, **Other Entitlement Details**, **Entitlement Attributes**, **Trust Rules**, and **History**.

The main content area is divided into two columns of form fields:

- Left Column:**
 - Entitlement name:** ADFS-533811351211-Developer
 - Security System:** AWS Managed
 - Description:** (Empty text field)
 - Update Date:** Sun, 20, 2019 04:12:11
 - Is Critical:** ☐ No
 - Status:** ☐ Active
 - Privileged:** ☐ No
 - Website:** ☐ Default
 - Display ID:** (Empty text field)
- Right Column:**
 - Entitlement Type:** AssumeRole
 - Endpoint:** AWS Managed
 - Display Name:** (Empty text field)
 - Entitlement Summary:** (Empty text field)
 - Is Critical:** ☐ No
 - Role:** ☐ No
 - Website:** ☐ No
 - Privileged:** ☐ No
 - Access:** ☐ Default

At the bottom right of the form, there are two buttons: **Back** and **Update**.

Figure: Entitlement details of child entitlement

Creating Users with Emergency Access Roles

You can use the Access Request System to request privileged access to the relevant AWS entitlements for business emergencies. The Emergency Access role are temporarily assigned to users to replace other users on leave and manage their responsibilities, assign a user for a day to deploy the product. You can raise this request for a stipulated time frame and this access is revoked as soon as the end date is reached. For more information, see [Managing Emergency Access Roles](#) in the *Enterprise Identity Cloud User Guide*.

Setting up Real Time Monitoring

CloudWatch is a service provided by AWS. This service is used to configure events recorded for different AWS services. These details are pushed to a target application such as Amazon Simple Queue Service (Amazon SQS).

This section provides high-level details about setting up real time monitoring:

1. Set up the CloudWatch event. For more information, see .
2. Specify the stack details. For more information, see [Specifying Stack Details](#).
3. Specify the parameters required to configure an AWS connection. For more information, see [Creating a Connection using the User Interface](#).
4. Run the **Application Data Import (Multi Threaded)** job and select the import type as incremental import. For more information, see [Performing Incremental Import](#).
5. Specify the parameters required to create an email template. For more information, see [Creating Email Templates](#).
6. Specify the query required to create and update analytics control. For more information, see [Creating and Updating Analytics Control](#).

Setting up the CloudWatch Event

You set up the CloudWatch Event for monitoring real time signatures.

1. Download the `SaviyntIncSetup.bat` batch file to the required location.

The batch file is available in the [S3 bucket](#).

The cloud formation template creates CloudWatch rule in all AWS regions with the list of events that have to be captured as part of real-time monitoring.

2. Configure AWS CLI with correct permissions to run the commands.



Note

If the CLI is not configured, use the following AWS URL to download and configure AWS CLI:

- **URL to download CLI:** <http://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html>
- **URL to configure CLI:** <http://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html>

3. Open the command prompt and enter the complete path of the downloaded batch file.

4. Press [Enter].

The batch file executes successfully. Setup is now complete and EIC has the permissions required to perform preventive actions. Inform the Saviynt team about the setup completion.

5. Note the following:

- Stack name - You will need this while specifying Stack details.

Stack name is created in ALL AWS regions. Stack ID is printed as the output of the batch file.

For example, `saviyntrealtimemonitoring`

- Credential files - You will need this to verify the access key ID and secret key password.

Credential files for **Windows OS** is available at `C:\Users\User\.aws`.

If you enter incorrect access key ID and secret key password, the following error is displayed:

“An error occurred (InvalidClientTokenId) when calling the CreateStack operation: The security token included in the request is invalid”.

- SQS URL - You will need this in the `CW_QUEUE_UL` parameter while creating an AWS connection and during incremental import.

For example,

`https://sqs.us-east-1.amazonaws.com/661222050851/SaviyntRealTimeMonitoringCloudwatchSQS`

where,

`sqs.us-east-1` is the name of the region.

`661222050851` is the AWS account ID.

For more information on adding stack name while specifying stack details, see [Specifying Stack Details](#).

Performing Incremental Import

After the first full import, you can use incremental import for subsequent imports. During the incremental import, the connector brings in only the changes that are made in Amazon SQS after the last import. While configuring the job, ensure that you select the **Job Type** as **Incremental Import** and **Import Type** as **Access** to import changes made after the last import. For more information about Data Jobs, see Data Jobs under [Job Categories for Flat Job Control Panel](#) in the *Enterprise Identity Cloud Administration Guide*.

Creating Email Templates

You create an email template to send alerts if any violation is reported. For more information, see [Creating and Managing Email Templates](#) in the *Enterprise Identity Cloud Administration Guide*.

Sample Email template details:

From - eic-alerts@saviynt.com

Name - Real time alerting email template to notify access key creation

To - \${ownerEmail}

Subject - Alert for creation of Access keys

Email body -

JSON

```

<div>Hi ${analyticsConfig.owner.firstname} ${analyticsConfig.owner.lastname},</div>
<div><br></div><div>One or more access keys have been created in your AWS accounts. </div>
<div>Please find attachment with details.</div><div><br></div>
<div> Name of control: ${analyticsConfig.analyticsName}</div>
<div>Risk : ${analyticsConfig.risk}</div><div>Category : ${analyticsConfig.category}</div>
<div>Event : CreateAccessKey</div><div><br></div>
<div> Note: This is a system generated email as this email box is not monitored.</div>
<div><br></div><div>Best Regards,</div><div>Saviynt Operations Team</div>

```

Creating and Updating Analytics Control

You run the SQL script to create and run Version 1 analytics controls. Then, edit the analytics control to add the email template. For more information, see [Creating Analytics Control \(V1\) using SQL Query](#) in the *Enterprise Identity Cloud Administration Guide*.

1. Run specific SQL queries in the EIC backend database



Info

This feature is applicable for EIC versions prior to Release v5.5.

The Saviynt Support Team can help you run them in your environment. For more information, contact the Saviynt Support Team at support@saviynt.com.

For queries required for predefined analytics report see [Appendix](#).

2. Create the **Version 1** analytic control in EIC.
3. Run the analytics control.
4. Edit the Analytics record to add the email template that you created. For more information, see [Creating and Managing Email Templates](#) in the *Enterprise Identity Cloud Administration Guide*.

CloudKnox Integration Overview

The integration provides visibility on usage of access granted to AWS resources. It allows the AWS application owners or system administrators to review the access usage on AWS resources, certify access assignments and perform the required access remediation using the Campaign and Control Center modules in EIC. You can also use this integration to perform privilege clipping to remove access of the user from the Role or Policy.

Supported Features

Module	Feature
Campaign	<ul style="list-style-type: none">• Allows an Entitlement Owner to view the date when the access was last used by the user.• Allows an Entitlement Owner to remove the access if the campaign is locked or has expired.• Allows a User Manager to view the date when access was last used by the user.• Allows a User Manager to approve or revoke access of the user.
Control Center	<ul style="list-style-type: none">• Allows accounts with access to resources or entitlements to view entitlements not used in last 60 days.• Allows accounts with access to resources or entitlements to view entitlements that were never used.• Allows the Control Center owner to view the date when the access was last used by the user.

Module	Feature
	<ul style="list-style-type: none">• Allows the Control Center owner to revoke access of the user.

Integration Architecture

CloudKnox fetches AWS identity and resources from AWS and EIC uses Saviynt Extensions to fetch AWS resource usage details from CloudKnox.

- The AWS connector is used to manage AWS identity and resources to AWS.
- Saviynt Extensions are used to fetch AWS resource usage details.

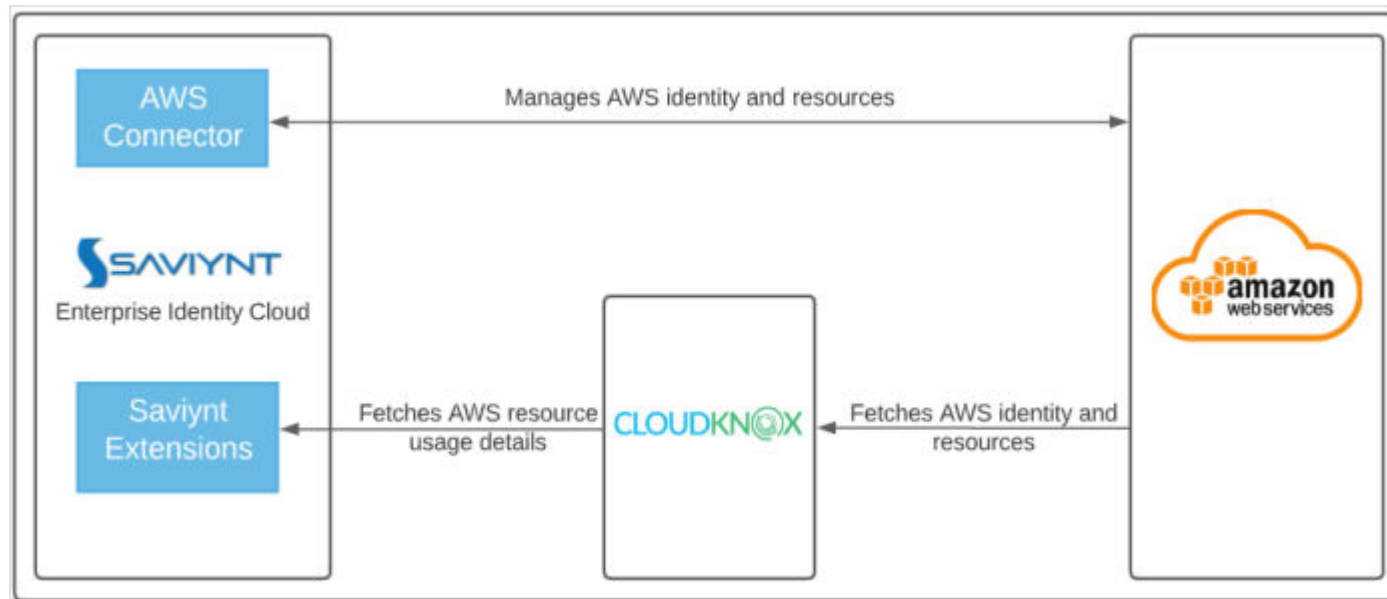


Figure: CloudKnox Integration Architecture

Entitlement Mapping

The following table details the mapping between database tables and entitlements imported for the integration:

Entitlement	Details Stored	Table	Column
AWSRole	Date when the role was last used by the user.	account_entitlements1	LASTUSEDENDDATE
AWSGroup	Date when permissions granted by the group was last used by the user.	account_entitlements1	LASTUSEDENDDATE
AWSPolicy	Date when permissions of the policy was last used by the user.	account_entitlements1	LASTUSEDENDDATE
	Date when permissions from the policy was last used by the user of the AWS group.	ENTITLEMENTS2	LASTUSEDENDDATE
	Date when permissions from the policy was last used by the AWS role.	ENTITLEMENTS2	LASTUSEDENDDATE

Entitlement	Details Stored	Table	Column
	Date when permissions from the policy was last used by resources of the AWS account.	ENTITLEMENT_VALUES	customproperty25, customproperty26

Creating an Integration

Prerequisites

Ensure that the following prerequisites are met before you start the integration procedure:

1. Download the following files:

You can download these files from [EIC Artifacts](#).

- Saviynt Extensions for Reconciliation
- Reconciliation properties file
- Dependency files:
 - secretsmanager file
 - encryption file
 - bcprov file

- sdk-core file
2. You must have administrator privilege for AWS console to create a secret manager.
 3. you must create a service account to get the access key and the secret key from CloudKnox.

Perform the following steps in sequence to integrate CloudKnox with EIC:

1. [Uploading Saviynt Extensions](#)
2. [Creating a Secret Manager](#)
3. [Encoding the Reconciliation Properties File](#)
4. [Executing the ExternalJar Job](#)

Guidelines to be followed during integration:

- Execute the ExternalJar job after running the AWS Account Import job to fetch the latest usage details. For more information on executing the job, see [Executing the ExternalJar Job](#). Create a Trigger Chain job and include these two jobs to run them sequentially.
- Run the Control Centre Analytics History job after running the Control Centre Analytics job to import data into Control Center. Create another Trigger Chain job and include these two jobs to run them sequentially.

- Create the Lambdas in the AWS account where the EIC is hosted and assign the required permissions via the Lambda role. This Lambda is used for remediation via the Control Center module.

Uploading Saviynt Extensions

Perform the following steps to upload Saviynt Extensions:

1. Log in to EIC.
2. Click **Apps > Admin**.
3. Click **Menu > Settings > File Directory**.
4. In the **File Directory** page that is displayed, select **externalJar**.
5. Click the **Upload** icon or you can also select **Upload New File**.
6. In the **File To Upload** window that is displayed, do the following:
 - a. Click **Select** next to the **Data File** box.
 - b. Select the Saviynt Extensions for Reconciliation to upload, and then click **Open**.
The Saviynt Extensions for Reconciliation file is displayed in the **Data File** box.
 - c. Click **Upload**.

7. Repeat steps 4 to 6 to upload the secretsmanager file.
8. Repeat steps 4 to 6 to upload the encryption file.
9. Repeat steps 4 to 6 to upload the bcprov file.
10. Repeat steps 4 to 6 to upload the AWS sdk-core file.

**Note**

The version of the AWS sdk-core file provided in the [EIC Artifacts](#) guide is 1.11.930, but the file is renamed to the existing version in the server, i.e., 1.11.519).

11. Restart EIC after uploading Saviynt Extensions. For more information, see [Restarting the Services](#) in the *Enterprise Identity Cloud Administration Guide*.

Creating a Secret Manager

1. Sign in to the AWS Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. In the AWS account where EIC is hosted, create a secret manager and store the following details:
 1. Select secret type as **Other type of secret**.
 2. Select **Plaintext** to enter the secret value in any format.

3. In the **Plaintext** page that displays, paste the following JSON with appropriate attribute values:

JSON

```
{
  "EIC_PASSWORD": "<specify password>",
  "EIC_USERNAME": "<specify username>",
  "CKX_SERVICE_ACCOUNT_ID": "<specify account ID>",
  "CKX_ACCESS_KEY": "<specify access key>",
  "CKX_SECRET_KEY": "<specify secret key>"
}
```

where,

EIC_PASSWORD: Specifies the password of an administrator in EIC.

EIC_USERNAME: Specifies the user name of an administrator in EIC.

CKX_SERVICE_ACCOUNT_ID: Specifies the service account ID of CloudKnox.

CKX_ACCESS_KEY: Specifies the access key of the service account in CloudKnox.

CKX_SECRET_KEY: Specifies the secret key of the service account in CloudKnox.

Encoding the Reconciliation Properties File

1. Modify the values in the following statement:

JSON


```
REGION=<specifies the Region details of AWS secret that you created>  
KEY_ARN=arn:<specifies the Key ARN details>  
SECRET_ARN=<specifies the secret ARN details>
```

2. Encode this with base64 using any online encoder.
3. Paste the encoded value in the CloudKnoxConfigs.properties file.

**Note**

The CloudKnoxConfigs.properties file is used to store the CloudKnox key. This file is available in following path:

/opt/saviynt/Conf/CloudKnoxConfigs.properties

or

/opt/sharedappdrive/saviynt/Conf/CloudKnoxConfigs.properties

Executing the ExternalJar Job

1. Log in to EIC.
2. Click **Apps > Admin**.
3. Click **Menu > Job Control Panel**.

4. In the **Job Control Panel** page that displays, select **Utility > Invoke Extension JAR Job (ExternalJarJob) > Add New Job**.
5. In the **Create New Trigger** window that displays, fill in the mandatory values.

Sample values:

JSON

```
FULL_CLASS_NAME_WITH_PACKAGE - com.cloudknox.Cloudknox
METHOD_NAME - policyProcess
ARGUMENTS -
{
  "data":{
    "awstype":"<specify aws type as comma separated values>",
    "endpoint":"<specify endpoint>",
    "EICUrl":"<specify URL>",
    "cloudknoxUrl":"<specify URL>"
  }
}
```

where,

awstype: Specifies the entity to be updated with usage details.

For example, **E2-AWSGroup,E2-AWSRole,AE-AWSGroup,AE-AWSRole,AE-AWSPolicy,ACC-Policy**. These values can also be

passed individually to sync only required combinations.

- E2-AWSGroup is used to synchronize the group to managed and inline policy usage details.
- E2-AWSRole is used to synchronize roles to managed and inline policy usage details.
- AE-AWSGroup is used to synchronize users to group usage details.
- AE-AWSRole is used to synchronize users to role usage details.
- AE-AWSPolicy is used to synchronize users to managed and inline policy usage details.
- ACC-Policy is used to synchronize policy usage for all policies for the AWS account.

endpoint: Specifies the endpoint key for which the sync must be run. Specify the value as 384.

EICUrl: Specifies the domain URL of EIC.

cloudknoxUrl: Specifies the domain URL of CloudKnox to sync data from CloudKnox.

6. Click **Save**.

7. Select the external job that was created, then click the **Start** icon.

Enhancements in Control Center

To support this integration, EIC provides additional controls to provide visibility to usage data fetched from CloudKnox. List of controls integrated with CloudKnox are added in the **Unused Permissions** book in the **Control Center** module.

To remove unused entitlements from target, perform the following steps:

1. Log in to EIC.

2. Click **Apps > Control Center**.
3. Click **Menu > Manage Applications**, and then click **Security Risks**.
4. Select the **Unused Permissions** book.
5. View the list of controls integrated with the CloudKnox.
6. Select **Execute Lambda Function** under **Select Bulk Action** to remove the entitlement from the target.

For more information about Control Center module, see [Understanding Control Center](#) in the *Enterprise Identity Cloud Administration Guide*.

The following table describes different controls integrated with CloudKnox:

Control Name	Description
Users with unused Groups	Detects AWS users assigned to Groups whose permissions is not used.
Users with unused Roles	Detects AWS users assigned to Roles whose permissions is not used.

Control Name	Description
Users with unused managed policies	Detects managed policies assigned to user but not used.
Users with unused inline policies	Detects inline policies assigned to user but not used.
Groups with unused managed policies	Detects managed policies assigned to Groups but not used by its members.
Groups with unused inline policies	Detects inline policies assigned to Groups but not used by its members.
Roles with unused managed policies	Detects managed policies assigned to Roles but not used.
Roles with unused inline policies	Detects inline policies assigned to Roles but not used.
Unused Custom policies in the AWS Account	Detects custom policies in the AWS Account not used by any entity to which they are associated.

Control Name	Description
AD groups without members having access to AWS Roles	Detects all AD groups having access to AWS Roles via Federation but are not assigned to any users.
AD groups having access to AWS roles which are not used for greater than 60 days	Detects all AD groups having access to AWS Roles via Federation but are unused by the group members.
AD groups having access to AWS roles where members have not signed in	Detects all AD groups having access to AWS Roles via Federation but the group members have not signed in.
Custom policies in the AWS Account which are not attached to any entity and have not been used for the last 60 days or more	Detects custom policies in the AWS Account not attached to any entity and not used for the last 60 days or more.
Groups with inline policies not used for the last 60 days or more	Detects inline policies assigned to Groups but not used by its members in last 60 days.

Control Name	Description
Groups with managed policies not used for the last 60 days or more	Detects groups with managed policies not used for the last 60 days or more.
Roles with inline policies not used for the last 60 days or more	Detects inline policies assigned to Roles but not used for last 60 days or more.
Roles with managed policies not used for the last 60 days or more	Detects managed policies assigned to Roles but not used for last 60 days or more.
Users with Groups not used for the last 60 days or more	Detects AWS users assigned to Groups but permissions not used for last 60 days or more.
Users with inline policies not used for the last 60 days or more	Detects inline policies assigned to user but not used for the last 60 days or more.

Control Name	Description
Users with managed policies not used for the last 60 days or more	Detects managed policies assigned to user but not used for the last 60 days or more.
Users with Roles not used for the last 60 days or more	Detects AWS users assigned with Roles but permissions not used for the last 60 days or more.

Enhancements in Campaigns

To support this integration, EIC provides a new field named **Entitlement Last Used End Date** in the **Campaign Configuration > List of Entitlements > Child Entitlements** page. This field displays the date when the child entitlement was last used by the entitlement. When you run an analytics query to display details in the **Control Center** module, all records unused for a specified duration are fetched based on this field value.

Perform the following steps to allow an Entitlement Owner to view Entitlement Last Used End Date field in the Entitlement Owner campaign:

1. Log in to EIC as an Admin user.
2. Click **Apps > Admin**.

3. Click **Menu > Configuration**, and then select **Global Configurations**.
4. Select **Campaign Config Entitlement Owner > Access Approval**.
5. Select **Entitlement Last Used End Date** under **Show Child Entitlement Attributes**.
6. The **Entitlement Last Used End Date** field is shown in campaigns for child entitlements.

Perform the following steps to allow an Entitlement Owner to create a task when the child entitlements are revoked:

When the Entitlement Owner campaign is locked after revoking a child entitlement, the assignment of the child entitlements are removed from the target application.

1. Log in to EIC as an Admin user.
2. Click **Apps > Admin**.
3. Click **Menu > Configuration**, and then select **Global Configurations**.
4. Select **Campaign Config Entitlement Owner > Revoke Tasks**.
5. Select **Create Revoke TaskforRevoked Child Entitlements on Locking**.

When the Entitlement Owner campaign expires after revoking a child entitlement, the assignment of the child entitlements are removed from the target application.

1. Log in to EIC as an Admin user.
2. Click **Apps > Admin**.
3. Click **Menu > Configuration**, and then select **Global Configurations**.
4. Select **Campaign Config Entitlement Owner > Revoke Tasks**.
5. Select **Create Revoke Task for Revoked Child Entitlements on Expiry**.

Perform the following steps to allow a User Manager to review the CloudKnox data:

1. Log in to EIC as an Admin user.
2. Click **Apps > Admin**.
3. Click **Menu > Configuration**, and then select **Global Configurations**.
4. Select **Campaign Config User Manager > Step 2 - Access Approval**.
5. Select **Last Used** under **Show Access Attributes**.

This helps an Entitlement Owner or an User Manager (reviewer) to approve or revoke access of the user. see [Managing Campaigns](#) in the *Enterprise Identity Cloud Administration Guide*.

Troubleshooting

To troubleshoot common problems or obtain answers for frequently asked questions for connectors, see the [Common Troubleshooting Guide for Connectors](#).

1. AWS Connection Fails

To provide external connection, the externalconfig.properties file must have the required SaaS properties. Raise a Freshdesk ticket to contact the Saviynt Support team for troubleshooting this issue.

2. Unable to Create New Accounts

To create new accounts ensure that the following details are correctly updated:

- The **CUSTOM_CONFIG_JSON** connection parameter must be updated with the following details:

JSON

```
{"createAccount":{"defaultInlinePolicyVersion":"2012-10-17"}}
```

where,

2012-10-17 is the default value for the configuration.

To update this field, contact the Saviynt Support team by raising a Freshdesk ticket.

- A password policy must be set in the AWS security system.

Perform the following steps to add a policy rule,

1. Log in to EIC.
2. Select **Admin > Identity Repository > Security Systems**.
3. Locate the security system.
4. Select a policy rule under **Policy Rule**.

The policy rule must match the password policy of the AWS account mentioned in the target.

Also ensure that the correct regex value is specified under **Admin > Identity Repository > Security Systems > Password Policies > Regex**.

This value must be appropriate for the AWS account.

For example, the Regex `[A-Za-z0-9~!@#$%^&*_-+=?/]{8,12}` means that the password can contain any of the following characters and the minimum allowed length is 8 and the maximum allowed length is 12:

- Uppercase characters (A-Z)
- Lowercase characters (a-z)
- Digits (0-9)
- Special characters (~!@#\$%^&*_-+=?/)

3. Unable to Provision Access to Policies, Groups and Roles

To provision access, ensure that the entitlement types AWSGroup, AWSRole and AWSPolicy are requestable and **Table** is selected as the **Request-Option**. Also ensure that you have configured the required workflows.

Perform the following steps to make the entitlement requestable:

1. Log in to EIC.
2. Select **Admin > Identity Repository > Security Systems**.
3. Locate the endpoint.
4. In the Endpoint page that displays, select **Entitlement Type**.
5. Select **Table** under **Request-Option**.

Perform the following steps to configure workflows,

1. Log in to EIC.
2. Select **Admin > Global Configurations > Roles**.
3. Select a workflow under **Roles Add Workflow**.
4. Select a workflow under **Roles Remove Workflow**.

5. Select a workflow under **Role Modification Workflow**.
6. [Optional] To make the modification with an auto-approval process, assign an auto-approval workflow in **Role Modification Workflow** or select **Role modification auto approve**.

4. An Exception Occurs and Stops the Configuration Recorder in ca-central-1 Region

Ensure that you have set the `PREVENTATIVECONTROL_TURNED_ON` parameter as `SELECT`.

Troubleshooting using AWS Policy Simulator when Assumed Role is not Authorized to Perform Action

You can use the AWS Policy Simulator to verify if the AWS policy is available for the given role. The following example considers that the role named,

SaviyntAWSAnalyzer-SaviyntAWSRole, is not authorized to perform the AWS policy named, `acm:ListCertificates`.

Perform the following steps to troubleshoot:

1. Open the [AWS console](#).
2. Log in to AWS console with your credentials.

**Note**

You must have required permissions to perform simulation.

3. Open the [IAM Policy Simulator Console](#).
4. In the **Users, Groups, and Roles** section, you can filter simulate policies by users, groups and roles. Select the filter as **Roles**.

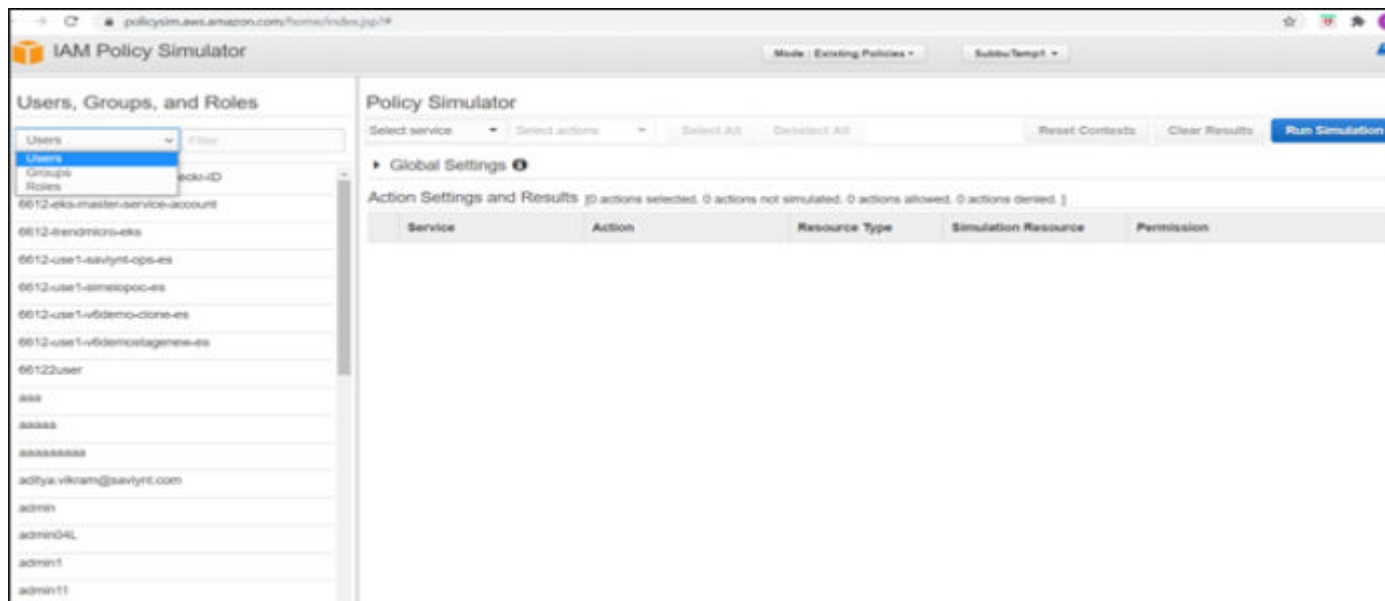


Figure: Options in the **Users, Groups, and Roles** section

- From the list of roles displayed, select the role created using Saviynt template. For example: SaviyntAnalyzerTest-SaviyntAWSRole-J0FYU8E2ABC

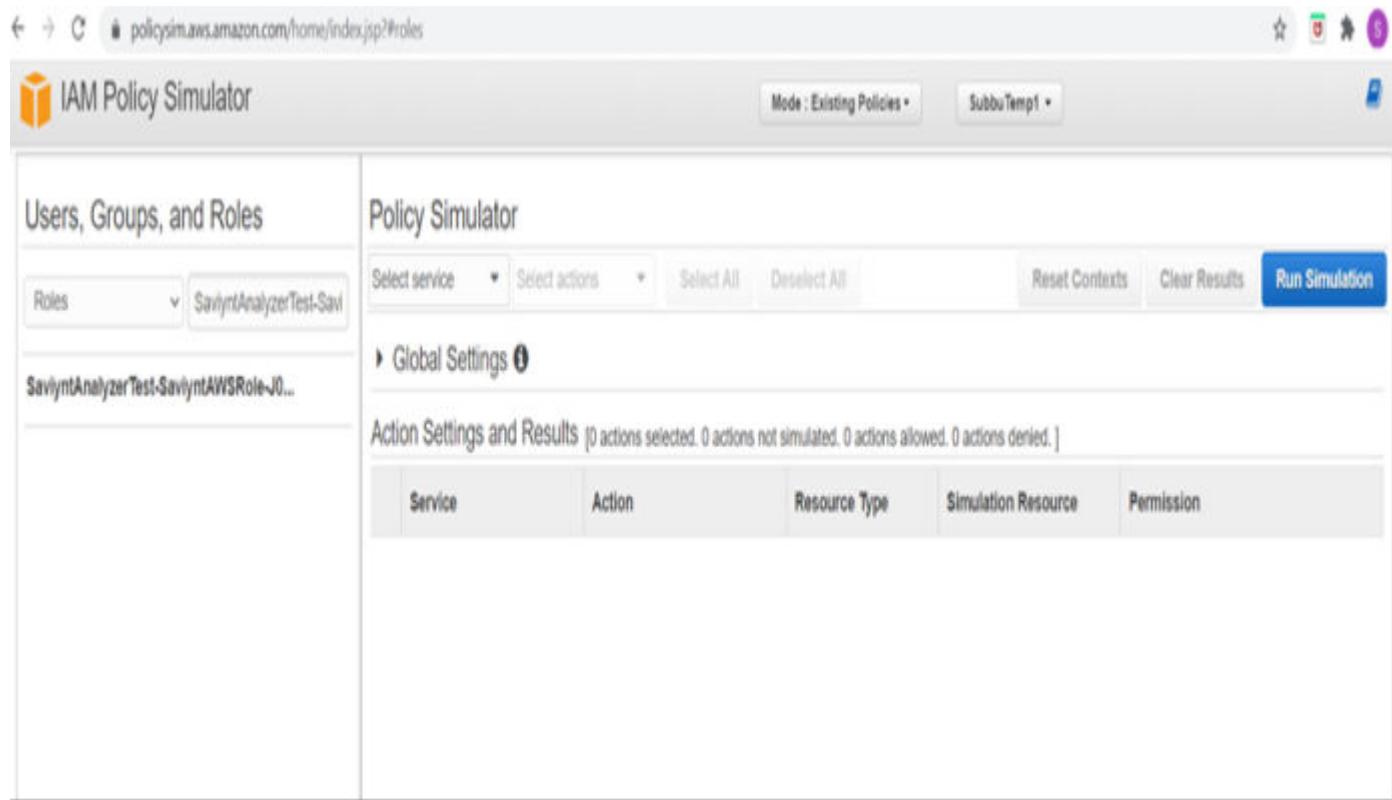


Figure: Role with Saviynt template

- In the **Policy Simulator** section, you can filter simulate policies by **service** and **actions**. For example, select the service as **Certificate Manager** and select the action as **ListCertificates**.

**Note**

Select the entitlement type (service) and the action for which the exception is shown.

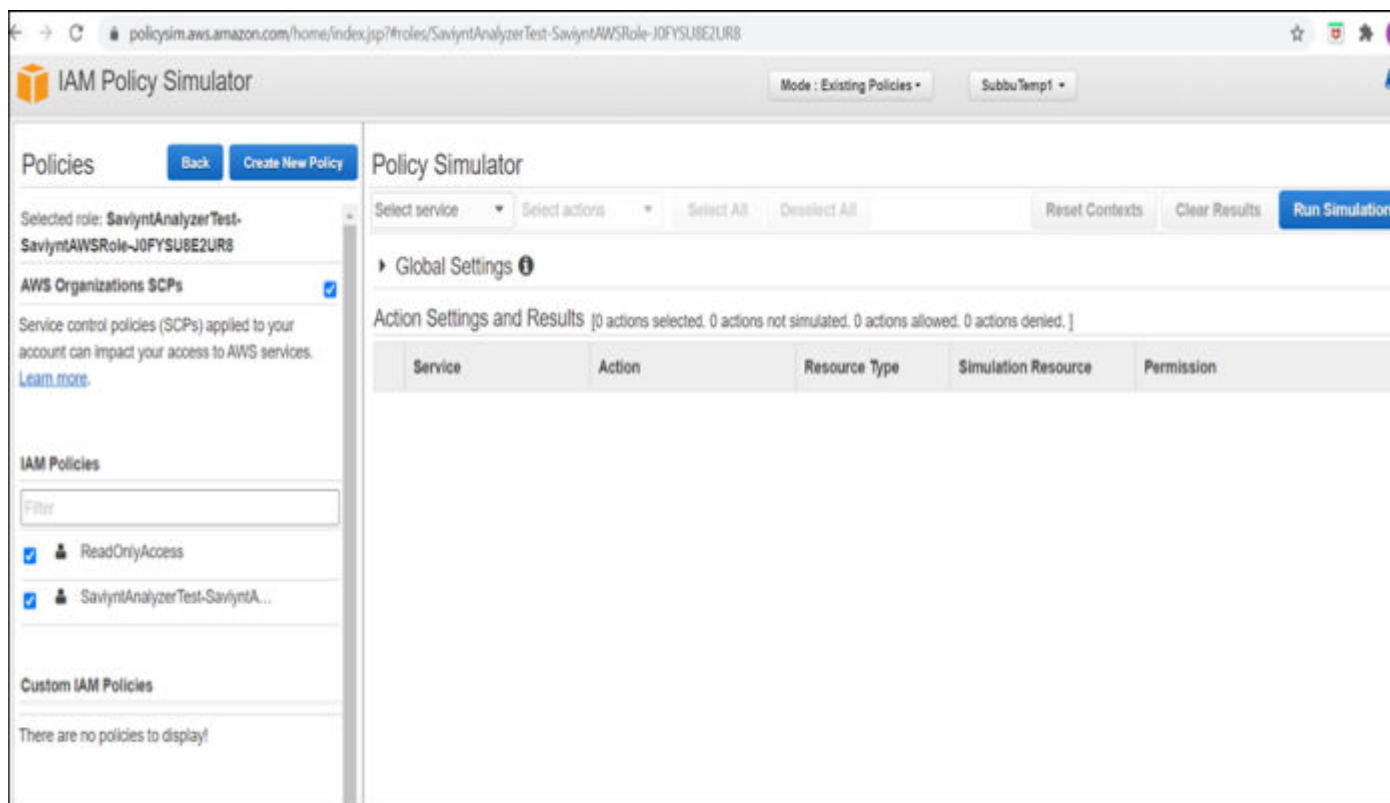


Figure: Options in the Policy Simulator section

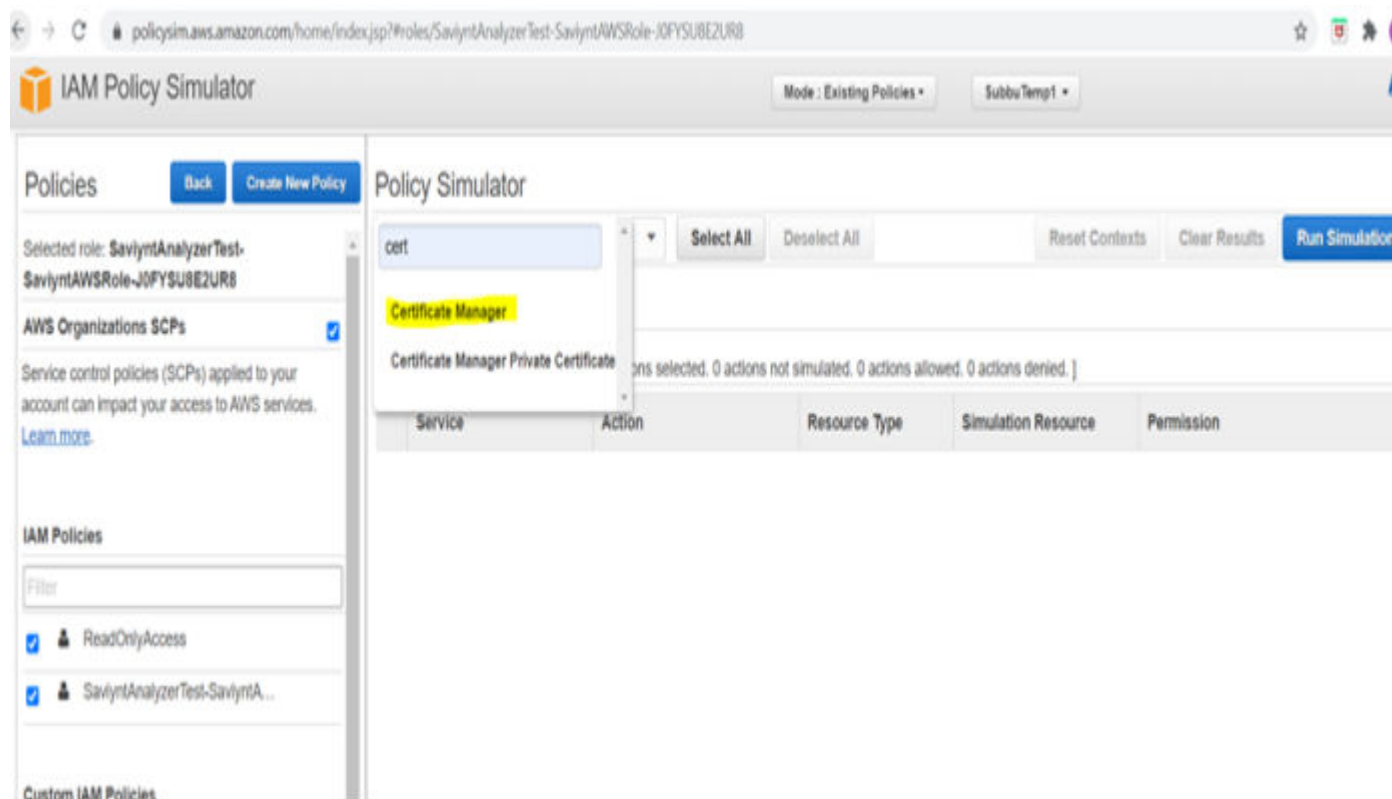


Figure: Service as Certificate Manager

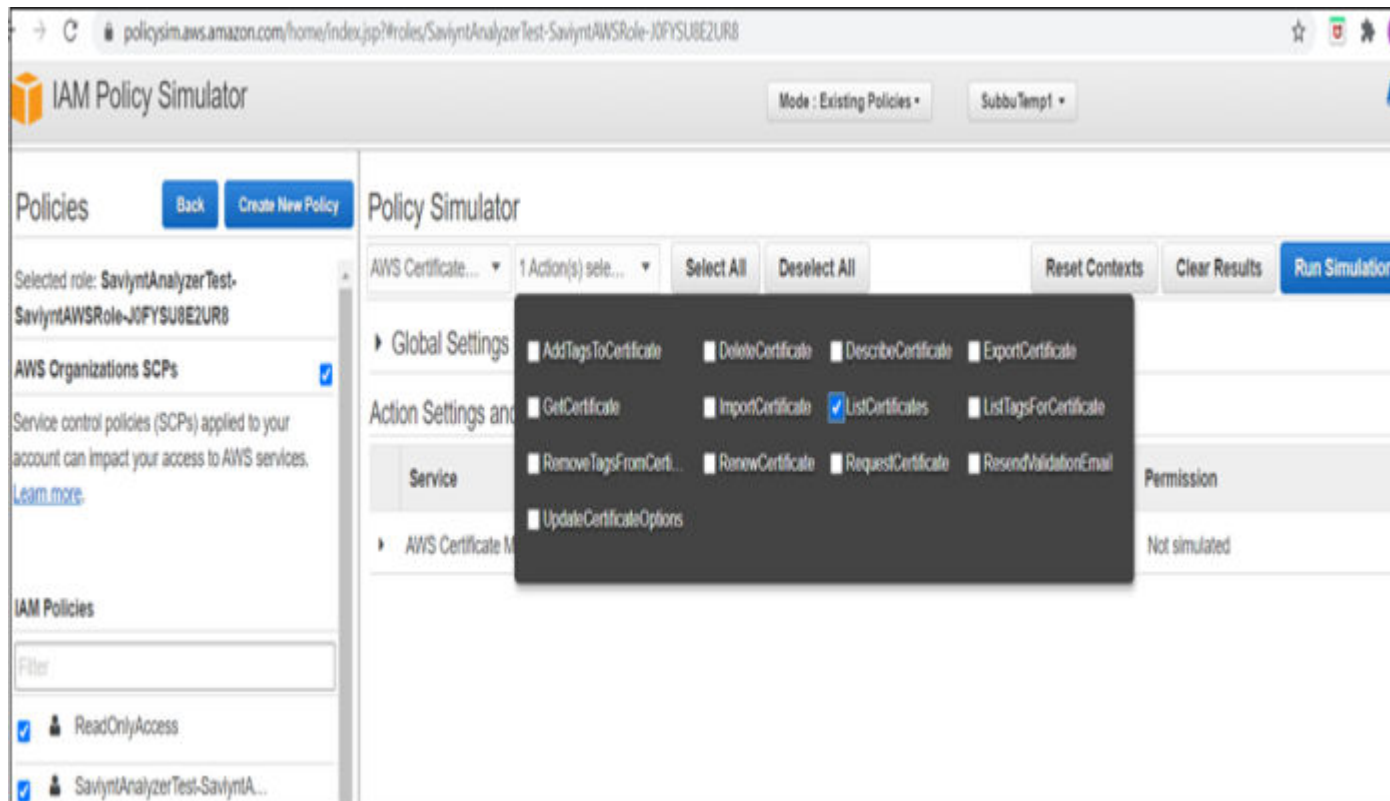


Figure: Option as ListCertificates

7. Click **Run Simulation**.

The Permission Simulator result is displayed under the **Permission** field.

The screenshot shows the IAM Policy Simulator web interface. The left sidebar contains a 'Policies' section with a 'Back' button and a 'Create New Policy' button. Below this, it shows the 'Selected role: SaviyntAnalyzerTest-SaviyntAWSRole-J0FYU8E2UR8'. Under 'AWS Organizations SCPs', there is a checkbox that is checked. Below that, it says 'Service control policies (SCPs) applied to your account can impact your access to AWS services. [Learn more.](#)'. Under 'IAM Policies', there is a 'Filter' input field and two checkboxes: 'ReadOnlyAccess' (checked) and 'SaviyntAnalyzerTest-SaviyntA...' (checked).

The main area is titled 'Policy Simulator'. It has a dropdown menu for 'AWS Certificate...' and a dropdown for '1 Action(s) sele...'. There are buttons for 'Select All', 'Deselect All', 'Reset Contexts', 'Clear Results', and 'Run Simulation'. Below these buttons, there is a section for 'Global Settings' and a section for 'Action Settings and Results'. The 'Action Settings and Results' section shows a table with the following data:

Service	Action	Resource Type	Simulation Resource	Permission
AWS Certificate Manager	ListCertificates	not required	*	allowed 1 matching statements.

Figure: Permission Simulate result

**Note**

Ensure that the value under the **Permission** field is **allowed**. To troubleshoot common problems or obtain answers for frequently asked questions for connectors, see the [Common Troubleshooting Guide for Connectors](#).

Appendix

Query for out-of-the-box Analytics Reports

JSON

```

/-- Query: select * from analyticsconfig where category='alerts'-- Date: 2020-06-17 21:00/INSERT
  INTO analyticsconfig
  (ANALYTICSKEY,ALLOWED_ACTION,ANALYTICSNAME,ANALYTICSQRY,ANALYTICSTYPE,APPLICATION,BASECOUNT,CATE
  GORY,COLUMNNAMEASCSV,
  CREATEDATE,customproperty1,customproperty10,customproperty2,customproperty3,customproperty4,cust
  omproperty5,customproperty6,
  customproperty7,customproperty8,customproperty9,description,EMAILTEMPLATE,EXTERNALCONNECTIONKEY,
  NOOFHISTORYTOKEEP,OWNER,
  OWNERTYPE,PERENDPOINT,PRECONFIGURED,QUERYBUILDERJSON,RECOMMENDATIONS,RISK,STATUS,SUBCATEGORY,tags,
  UNIQUEIDENTITY,UPDATEDATE,
  UPDATEUSER,USERGROUPS,ENABLEARCHIVAL,ERRORJSONMSG) VALUES (10029,'0:Open,1:Accept,2:Revoke,3:Fur
  ther Review','Accept VPCPeering Connection',
  'select distinct excv.attributevalue as AccountID, ev.customproperty9 as 'Region',ev.entitlement
  _value as 'VPCPeering ID',ev.customproperty6 as
  'Acceptor VPC Owner',ev.customproperty8 as 'Acceptor VPC ID',ev.customproperty10 as 'Requestor V
  PC Owner',ev.customproperty12 as
  'Requestor VPC ID',ev.customproperty2 as 'VPC Peering Status',ev.customproperty16 as 'Accept Dat
  eTime' from entitlement_values
  ev Inner join entitlement_types et on et.Entitlementname = 'VPCPeering' and ev.entitlementtypekey
  y=et.entitlementtypekey and
  COALESCE(ev.status,0) < 2 Inner join securitysystems sc on et.systemkey = sc.systemkey Inner Joi
  n externalconnection exc on
  sc.externalconnection = exc.externalconnectionkey Inner Join externalconnectiontype exct on exc.
  externalconnectiontype =
  excv.externalconnectiontypekey and exct.connectiontype = 'AWS' Inner join externalconnattvalue e
  xcv on excv.connectiontype =

```

List of Entitlement Types

The AWS Connector supports importing of the following entitlement types:

- Local IAM Users with additional attributes such as Access Keys SSH Public Keys, Virtual MFA Device, and login profile
- AWSPolicy
- AWSRole
- AWSGroup
- EC2Instance
- SecurityGroup
- AMI
- ElasticLoadBalancer
- TargetGroup
- AutoScaling
- LaunchConfig
- EBSVolume
- EBSSnapshot
- EFS
- NetworkInterface

- VPC
- DhcpOption
- Subnet
- NACL
- VpcPeering
- RouteTable
- NatGateway
- InternetGateway
- ElasticIP
- VpcFlowLog
- VpcEndpoint
- S3Bucket
- RdsDbInstance
- RDSSnapshot
- RDSEventSubscription
- Glacier
- RedShiftCluster with Param Groups

- CloudFormation
- EncryptionKey
- CloudTrail
- SnsTopic
- SQS
- CloudWatchLogGroup
- CloudWatchAlarm
- AWSConfig recorder status
- DynamoDB
- ElasticSearch
- EMR
- ACM
- Route53
- CloudFront
- AWSLambda
- GuardDuty enabled flag
- WAF

- PasswordPolicy
- AWSLambda

List of Entitlements that Store Tags

The following entitlements pull AWS tags during the custom_access import. For more information about custom_access, see [Customizing Entitlement Import](#):

- EC2Instance
- VPC
- AWSSecurityGroup
- AWSNacl
- S3Bucket
- EBSVolume
- ELB
- Subnets
- RdsDbInstance
- AWSCloudTrail
- RouteTable

- AMI
- VpcPeering
- InternetGateway
- NetworkInterface
- ClusterSecurityGroups
- RSCluster
- RSPparameterGroups
- RSEC2SecurityGroup
- CloudFormation
- S3CFTemplate
- DhcpOptions
- EBSSnapshot
- Glacier
- EFS
- EMR
- ElasticSearch
- Workspace

- AppELB
- AutoScaling
- AWSRole

**Info**

This entitlement is available from Release 2020.1 onwards.