



# Integrating AWS S3 with File Access Manager

Version: 8.3 Revised: March 30, 2022

This document and the information contained herein is [SailPoint Confidential Information](#)

---

## Copyright and Trademark Notices.

Copyright © 2022 SailPoint Technologies, Inc. All Rights Reserved.

All logos, text, content, including underlying HTML code, designs, and graphics used and/or depicted on these written materials or in this Internet website are protected under United States and international copyright and trademark laws and treaties, and may not be used or reproduced without the prior express written permission of SailPoint Technologies, Inc.

“SailPoint,” “SailPoint & Design,” “SailPoint Technologies & Design,” “Identity Cube,” “Identity IQ,” “IdentityAI,” “IdentityNow,” “SailPoint Predictive Identity” and “SecurityIQ” are registered trademarks of SailPoint Technologies, Inc. None of the foregoing marks may be used without the prior express written permission of SailPoint Technologies, Inc. All other trademarks shown herein are owned by the respective companies or persons indicated.

SailPoint Technologies, Inc. makes no warranty of any kind with regard to this manual or the information included therein, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SailPoint Technologies shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Patents Notice. <https://www.sailpoint.com/patents>

Restricted Rights Legend. All rights are reserved. No part of this document may be published, distributed, reproduced, publicly displayed, used to create derivative works, or translated to another language, without the prior written consent of SailPoint Technologies. The information contained in this document is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

Regulatory/Export Compliance. The export and re-export of this software is controlled for export purposes by the U.S. Government. By accepting this software and/or documentation, licensee agrees to comply with all U.S. and foreign export laws and regulations as they relate to software and related documentation. Licensee will not export or re-export outside the United States software or documentation, whether directly or indirectly, to any Prohibited Party and will not cause, approve or otherwise intentionally facilitate others in so doing. A Prohibited Party includes: a party in a U.S. embargoed country or country the United States has named as a supporter of international terrorism; a party involved in proliferation; a party identified by the U.S. Government as a Denied Party; a party named on the U.S. Department of Commerce’s Entity List in Supplement No. 4 to 15 C.F.R. § 744; a party prohibited from participation in export or re-export transactions by a U.S. Government General Order; a party listed by the U.S. Government’s Office of Foreign Assets Control as ineligible to participate in transactions subject to U.S. jurisdiction; or any party that licensee knows or has reason to know has violated or plans to violate U.S. or foreign export laws or regulations. Licensee shall ensure that each of its software users complies with U.S. and foreign export laws and regulations as they relate to software and related documentation.

# Contents

---

<b>Contents</b> .....	<b>iii</b>
<b>Capabilities</b> .....	<b>5</b>
<b>Connector Overview</b> .....	<b>6</b>
Crawler .....	6
Permission Collector .....	6
Identity Collection .....	6
Cross Account Access .....	8
Cross Account by Assume Roles .....	8
Block Public Access .....	9
<b>Prerequisites for AWS</b> .....	<b>11</b>
Software Requirements .....	11
Configuring an EC2 for File Access Manager Connector .....	11
Creating a Dedicated IAM User .....	18
<b>AWS S3 Installation Flow Overview</b> .....	<b>25</b>
<b>Collecting Data Stored in an External Application</b> .....	<b>26</b>
<b>Adding an AWS S3 Application</b> .....	<b>27</b>
Select Wizard Type .....	27
General Details .....	27
Connection Details .....	27
Configuring and Scheduling the Permissions Collection .....	28
<b>Active Directory Integration with AWS</b> .....	<b>35</b>
<b>Mapping Extractions from IDPs</b> .....	<b>36</b>
<b>Installing Services: Activity Monitor and Collectors</b> .....	<b>46</b>
<b>Verifying the AWS S3 Connector Installation</b> .....	<b>48</b>
Installed Services .....	48
Log Files .....	48
Permissions Collection .....	48
<b>Appendix A: Json Scripts</b> .....	<b>49</b>

## Contents

---

IdentityIQ_FileAccessManagerRole.json [EC2] .....	49
IdentityIQ_FileAccessManagerRole.json [Dedicated User] .....	49
IdentityIQ_FileAccessManager_AssumeRolePolicy.json .....	50
IdentityIQ_FileAccessManager_S3IAMReadOnlyAccessPolicy.json .....	50

## Capabilities

This connector enables you to use File Access Manager to access and analyze data stored in AWS S3 and do the following:

- Analyze the structure of your stored data.
- Verify user permissions on the resources, and compare them against requirements.
- Identity collector – collect IAM users, groups and roles and the connections between them

See the File Access Manager documentation for a full description.

## Connector Overview

Accounts must be configured as described in [Prerequisites for AWS](#) , for them to be analyzed.

### Crawler

The crawler analyzes the structure of the organization and builds the hierarchy tree

- Organization Root container
- Organization Units (OUs)
- AWS Accounts
- S3 Buckets
- S3 Folders

#### **Analyze all Objects in S3 Buckets**

If Analyze all Objects is checked, the crawler will get also the S3 Objects (files) under the buckets, their size and total size of the containing folder.

### Permission Collector

The Permission collection will retrieve and analyze the following permissions:

- ACLs of buckets. If **Analyze ACLs** is checked, ACLs will be collected for the objects retrieved in the crawl.
- Bucket policies for the buckets and their objects.
- IAM policies which are relevant for the S3 buckets and Objects.
- Account and bucket level PublicAccessBlock configurations.
- Cross account permissions

Permission collection limitations and unsupported features:

- Permissions are analyzed for buckets and objects, not for folders since they are not an actual object in S3
- Permissions Boundary
- Policies Conditions
- Policies Variables
- Policies elements - NotPrincipal, NotAction, NotResource
- Only S3 related permissions are analyzed
- Access points and Jobs permissions are not analyzed

### Identity Collection

The AWS identities will be collected by the permission collector at the beginning of the task.

- The following identities are collected:
  - AWS Accounts (root users)
  - IAM Users
  - IAM Groups
  - IAM Roles
- The AWS predefined groups are represented as the following groups:
  - http://acs.amazonaws.com/groups/global/AllUsers***  
“Anonymous” with type “Everyone or Authenticated Users, or contains it”
  - http://acs.amazonaws.com/groups/global/AuthenticatedUsers***  
“AwsAuthenticatedUsers” with type “Everyone or Authenticated Users, or contains it”
  - http://acs.amazonaws.com/groups/s3/LogDelivery***  
“S3LogDelivery” with type “Local Group”.
- From each IAM Role, File Access Manager collects its trusted entities as members of the role.
- The AWS entities will be mapped to the following types:
  - IAM Users – will be saved as FAM “Local User” type.
  - IAM Groups – will be saved as FAM “Local Group” type.
  - IAM Roles – will be saved as FAM “Local Role” type.
  - AWS Account – will be saved as FAM “AWS Account” type.
  - AWS Service – will be saved as FAM “AWS Service” type.
  - All other types, including “Federated”, etc. , – will be saved as FAM “AWS External Account” type.
- IAM Role trusted Identity of type "\*" is represented as **“Anonymous”** with type “Everyone, Authenticated Users, or contains it”.
- "Principal": "\*" in bucket policy is represented as “Anonymous” with type “Everyone, Authenticated Users, or contains it”.
- For each Collected identity, the primary ID will be their Arn and Alternative Ids will be collected as well:
  - For AWS Accounts – Id, root user Arn ("arn:aws:iam::{iamRootUser.Id}:root") and canonical Id.
  - For other identities – Id.
- Additional information that is collected:
  - Name
  - Display Name
  - Description

- Domain – will be the AccountName(#AccountId)
- Email (Only for Aws Account)
- LastLogin (Only for IAM Users)

## Cross Account Access

To achieve cross account access, and allow an AWS IAM Identity from Account A to access AWS resources in account B (S3 resource in our case) two conditions must be met:

1. The IAM Identity owner account A should give permission X on the S3 resource in account B.  
In File Access Manager this permission will appear as **X-ByTrustedCrossAccount**
2. The S3 resource owner account B should give permission X on the resource to the IAM Identity from account A.  
In File Access Manager this permission will appear as **X-ByTrustingCrossAccount**.

Permission X will be effective only if both permissions are granted to the user / group on the resource. Otherwise, the user / group will not be allowed to perform X on this resource.

Permissions Forensics Saved Queries Global Options Apply

Filters (2) Save Clear All

User Name	Equals	testAccessToAnotherResource	✎	🗑️
Permission Type	Contains	GetBucketLocation	✎	🗑️

View by: Groups & Users Direct Permissions Mark permissions unused for longer than 6 months

<input type="checkbox"/>	Business Resource Full Path	Application	User Name	User Domain	User Entity Type	Permission Type	ACL Type Allow?
<input type="checkbox"/>	Root/FAM-QA2(#988979385348)/s3.us-east-1.bucket1-fam-qa2-user1adminpriv	s3-1	FAMAdminUser1	FAM-QA1(#961222436048)	User	GetBucketLocation-ByTrustingCrossAccount	Allow
<input type="checkbox"/>	Root/FAM-QA2(#988979385348)/s3.us-east-1.bucket1-fam-qa2-user1adminpriv	s3-1	FAMAdminUser1	FAM-QA1(#961222436048)	User	GetBucketLocation-ByTrustedCrossAccount	Allow

In the example above, the user “FAMAdminUser1” from account “FA-QA1” has both “GetBucketLocation-ByTrustingCrossAccount” and “GetBucketLocation-ByTrustedCrossAccount” permission on bucket “bucket1-fam-qa2-user-1adminpriv” from account “FAM-QA2”.

## Cross Account by Assume Roles

This scenario requires 4 conditions for user USER\_A from account A to have permission X on resource RESOURCE\_B from account B through role ASSUME\_ROLE\_B:

1. ASSUME\_ROLE\_B is defined in account B.
2. ASSUME\_ROLE\_B is attached to policy that gives permission X on RESOURCE\_B.
3. USER\_A should be a member of ASSUME\_ROLE - a trusted entity of the role.
4. USER\_A should have in account A, permission to assume ASSUME\_ROLE\_B in account B.

File Access Manager does not display this information in v8.2.

## Connector Overview

Permissions Forensics

Filters (4) Save Clear All

Global Options Apply

User Name	Equals	AmirTestUser1	
Business Resource Full Path	Equals	Root/TEST/Test12/FAM-Dev-Public#225054067900/s3.us-east-1.fam-dev-public-bucket1	
Permission Type	Equals	GetBucketPolicy	
Group Name	Equals	FAMConnectorRole	

View by: Users Direct & Group Membership Permissions Mark permissions unused for longer than 6 months

Business Resource Full Path	Application	User Name	Group Name	User Domain	Group Domain	Group Entity Type	Permission Type	ACL Type Allow?
Root/TEST/Test12/FAM-Dev-Public#225054067900/s3.us-east-1.fam-dev-public-bucket1	s3-1	AmirTestUser1	FAMConnectorRole	FAM-Org#420259386505	FAM-Dev-Public#225054067900	Role	GetBucketPolicy	Allow

In the example above, the role “FAMConnectorRole” allows “GetBucketPolicy” on bucket “fam-dev-public-bucket1”. The role and the bucket, both belong to account “FAM-Dev-Public”. The role has a member user (trusted entity) “AmirTestUser1” from account “Fam-Org”.

If in account FAM-Org, “AmirTestUser1” has a policy which allows it to assume the role “FAMConnectorRole” in account “FAM-Dev-Public” (Not supported in File Access Manager view in v8.2) – The permission will be active.

## Block Public Access

The Amazon S3 Block Public Access feature provides settings for buckets and accounts to help manage public access to Amazon S3 resources. By default, new buckets and objects don't allow public access. However, users can modify bucket policies or object permissions to allow public access. S3 Block Public Access settings override these policies and permissions and enable to limit public access to these resources.

There are 4 settings both on the bucket level, and the account level, If the PublicAccessBlock settings are different between the bucket and the account, Amazon S3 uses the most restrictive combination of the bucket-level and account-level settings.

In File Access Manager these permissions appear with the suffix “Account-Disabled” for the account level settings and “Bucket-Disabled” for the bucket level settings. If one of these settings is turned off, the Permission Forensics view shows these permissions as “Allow”.

Permissions Forensics

Filters (2) Save Clear All

Global Options Apply

Permission Type	Contains	-disabled	
Business Resource Full Path	Equals	Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	

View by: Groups & Users Direct Permissions Mark permissions unused for longer than 6 months

Business Resource Full Path	Application	User Name	User Domain	User Entity Type	Permission Type	ACL Type Allow?
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				BlockPublicAcls-Bucket-Disabled	Deny
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				BlockPublicPolicy-Bucket-Disabled	Deny
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				IgnorePublicAcls-Bucket-Disabled	Deny
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				RestrictPublicBuckets-Bucket-Disabled	Deny
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				BlockPublicAcls-Account-Disabled	Allow
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				BlockPublicPolicy-Account-Disabled	Allow
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				IgnorePublicAcls-Account-Disabled	Allow
Root/FAM-Dev1(#225054067900)/s3.ap-southeast-1.amir-bucket1-dev1	s3-1				RestrictPublicBuckets-Account-Disabled	Allow

In the admin client, in *Resources->Permissions->Simple View* they will appear with warnings.

The screenshot displays the SailPoint interface with the following components:

- Navigation Bar:** Includes 'Dashboard', 'Resources', 'My Tasks', 'Reports', 'Compliance', 'Forensics', 'Goals', 'Settings', and 'Admin'.
- Sub-Menu:** Includes 'Activities', 'Permissions', 'Data', 'Alerts', and 'Owners'.
- Resources Tree (Left):** A hierarchical tree structure under 'All Applications' showing folders like 's3-1', 'Root', 'AmirBU\_A', and various 'FAM' folders. The 'amir-bucket3-dev1' folder is highlighted.
- Permission Detail View (Right):**
  - Header: 'Permission > Simple View' with a refresh icon.
  - Resource: 'amir-bucket3-dev1'.
  - Application: 's3-1 | Path: Root/FAM-Dev1(#[redacted])s3.us-east-1.amir-bucket3-dev1'.
  - Section: 'Permissions Types' with a sub-header 'Permission Type: BlockPublicActs-Account-Disabled (0)'.
  - Alert: A yellow banner with a warning icon and text: 'Alert! Everyone has BlockPublicActs-Account-Disabled access to this resource! View Details'.
  - Table of Permissions Types:

Permissions Types	Count
OWNER	(1)
AllS3PolicyActions	(4)
BlockPublicActs-Bucket-Disabled	(0)
BlockPublicPolicy-Bucket-Disabled	(0)
IgnorePublicActs-Bucket-Disabled	(0)
RestrictPublicBuckets-Bucket-Disabled	(0)
BlockPublicActs-Account-Disabled	(0)
BlockPublicPolicy-Account-Disabled	(0)
IgnorePublicActs-Account-Disabled	(0)

## Prerequisites for AWS

This section describes the minimal set of permissions required to configure a File Access Manager AWS connector. It is a step-by-step guide, including AWS Console Screens.

Make sure your system fits the descriptions below before starting the installation

There are two methods to configure the AWS File Access Manager connector, and the require configuration is different for each.

- EC2 instance to run File Access Manager (This is the recommended method)
- Dedicated IAM user

### Software Requirements

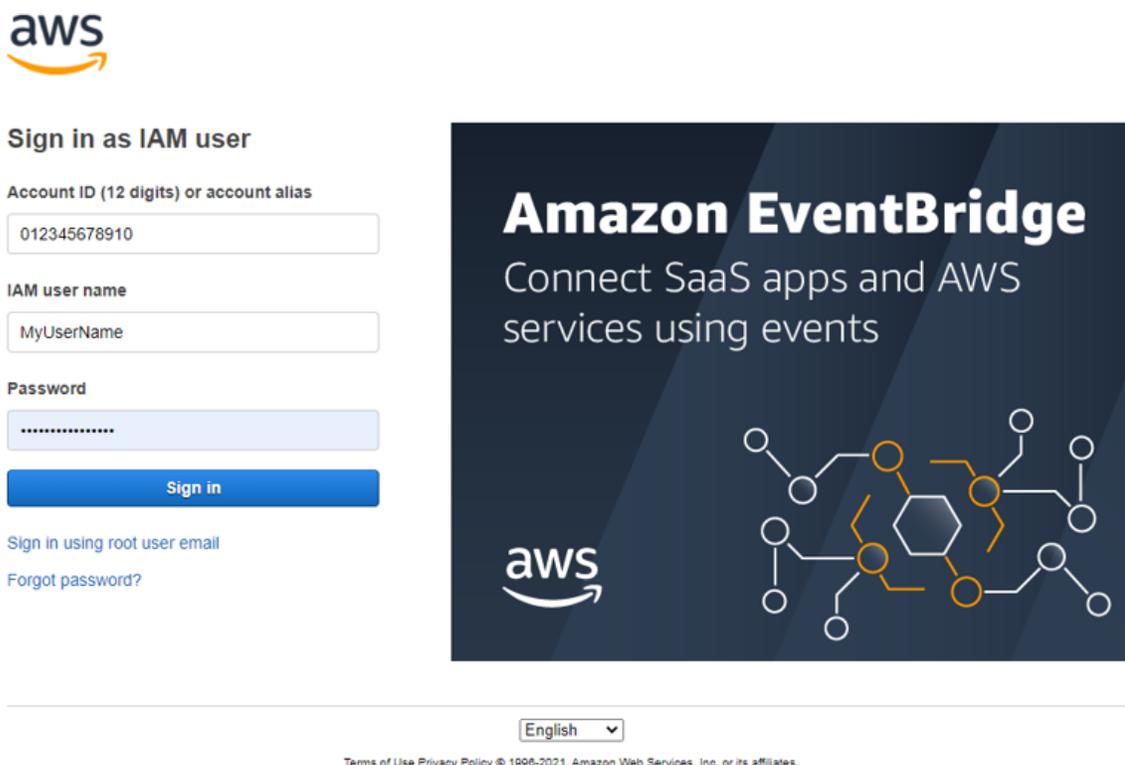
File Access Manager requires the latest ASP.NET Core 3.1.x Hosting Bundle. This bundle consists of .NET Runtime and ASP .NET Core Runtime. You can download the latest 3.1.x Hosting Bundle version from [here](#).

### Configuring an EC2 for File Access Manager Connector

This is the recommended connection method for the File Access Manager connector.

Create a role and policies to enable running the File Access Manager activities on all accounts in the organization.

1. Sign into your AWS account.



2. Create a new policy "FileAccessManager\_AssumeRolePolicy".

This policy will allow the File Access Manager application, created in the next step, to perform an **Assume Role** on the roles that will be created in each account.

See [IdentityIQ\\_FileAccessManager\\_AssumeRolePolicy.json](#) in Appendix A.

Create policy 1 2 3

---

Review policy

Name\*   
Use alphanumeric and "+=, @\_-." characters. Maximum 128 characters.

Description   
Maximum 1000 characters. Use alphanumeric and "+=, @\_-." characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 276 services) <a href="#">Show remaining 275</a>			
STS	Limited: Write	RoleName   string like   IdentityIQ_FileAccessManagerRole	None

Tags

Key	Value
No tags associated with the resource.	

\* Required Cancel Previous **Create policy**

3. Create a new role

- Select **AWS Service** as the trusted entity type.
- Select EC2 as the service.

Create role 1 2 3 4

Select type of trusted entity



**AWS service**  
EC2, Lambda and others



**Another AWS account**  
Belonging to you or 3rd party



**Web identity**  
Cognito or any OpenID provider



**SAML 2.0 federation**  
Your corporate directory

Allows AWS services to perform actions on your behalf. [Learn more](#)

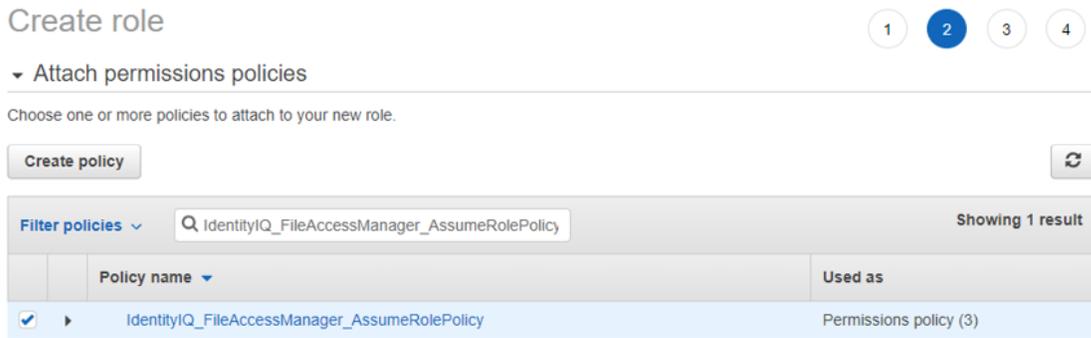
Choose a use case

**Common use cases**

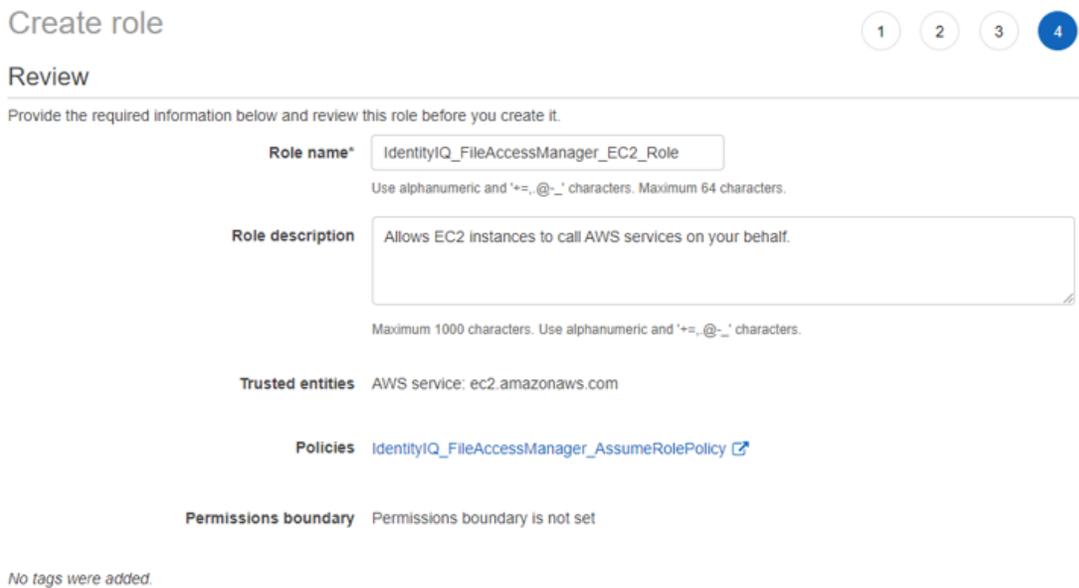
**EC2**  
Allows EC2 instances to call AWS services on your behalf.

**Lambda**  
Allows Lambda functions to call AWS services on your behalf.

- 4. Attach the role to the FileAccessManager\_AssumeRolePolicy policy created above.



- 5. Give the role a name (e.g. FileAccessManager\_EC2\_Role) and create it.



- 6. If you are creating a **new** EC2 instance select the above role as the IAM role for the instance.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the low prices.

**Number of instances** ⓘ  [Launch into Auto Scaling Group](#) ⓘ

**Purchasing option** ⓘ  Request Spot instances

**Network** ⓘ  [Create new VPC](#)

**Subnet** ⓘ  [Create new subnet](#)

**Auto-assign Public IP** ⓘ

**Placement group** ⓘ  Add instance to placement group

**Capacity Reservation** ⓘ

**Domain join directory** ⓘ  [Create new directory](#)

**IAM role** ⓘ  [Create new IAM role](#)

**CPU options** ⓘ  Specify CPU options

**Shutdown behavior** ⓘ

**Stop - Hibernate behavior** ⓘ  Enable hibernation as an additional stop behavior

**Enable termination protection** ⓘ  Protect against accidental termination

**Monitoring** ⓘ  Enable CloudWatch detailed monitoring  
[Additional charges apply.](#)

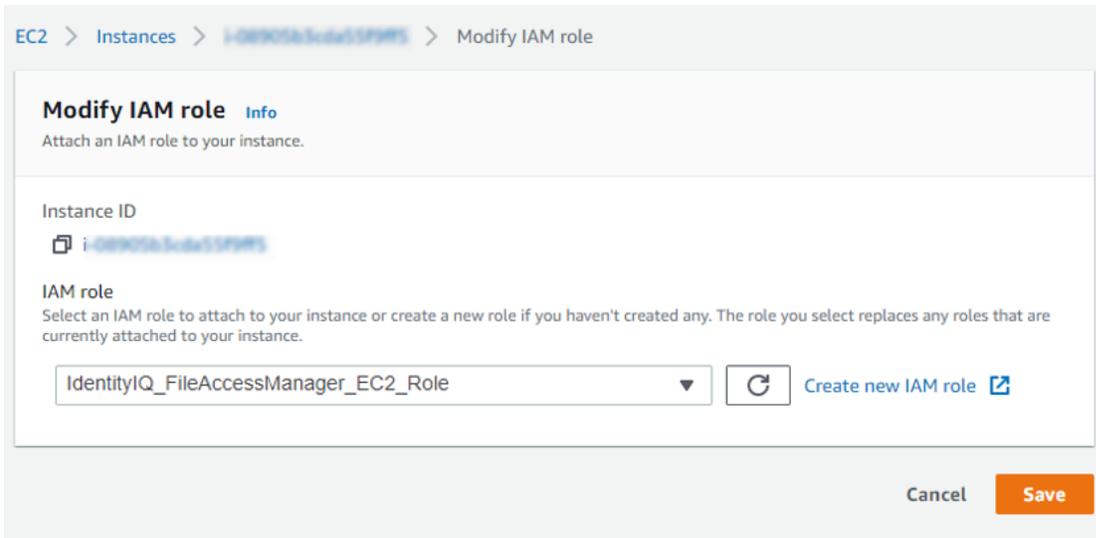
**EBS-optimized instance** ⓘ  Launch as EBS-optimized instance

**Tenancy** ⓘ  [Additional charges will apply for dedicated tenancy.](#)

**Elastic Graphics** ⓘ  Add Graphics Acceleration  
[Additional charges apply.](#)

**Credit specification** ⓘ  Unlimited  
[Additional charges may apply](#)

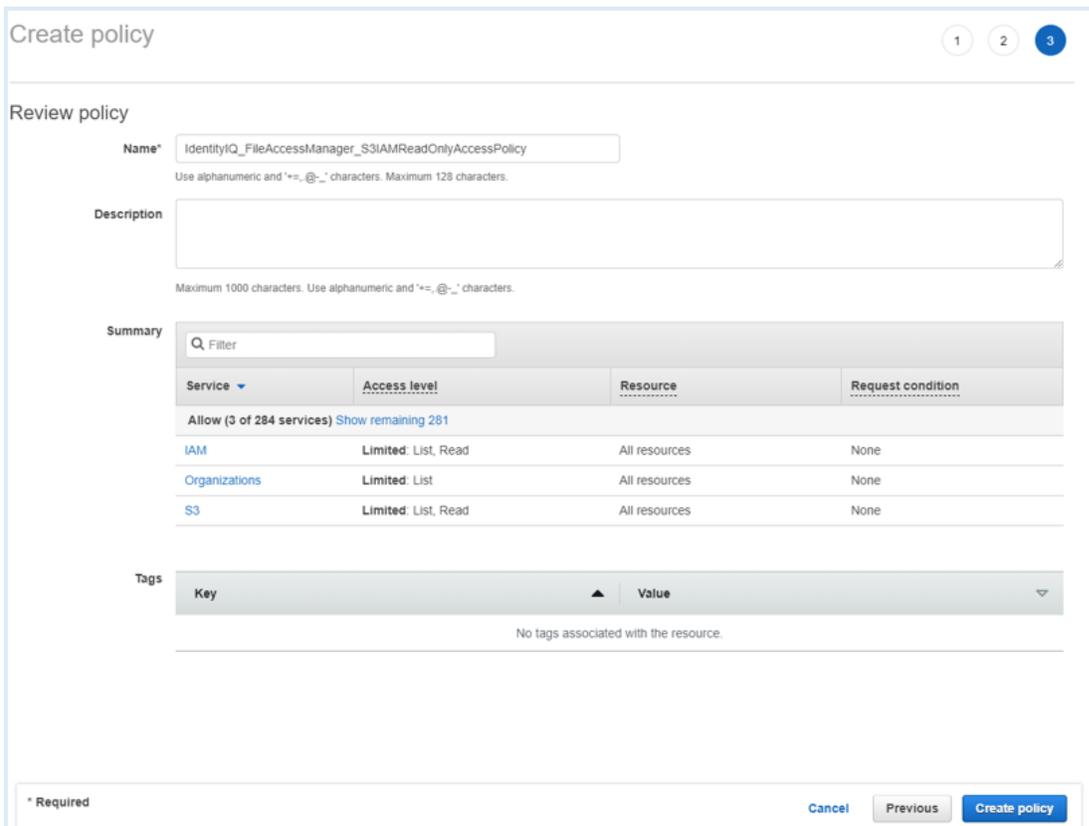
7. If you are using an **existing** EC2 instance, Modify the IAM role to the role above  
In the option  
*EC2 > Instances > Actions > Security > Modify IAM role*



8. Create a new policy for each organization account the connector is supposed to analyze

Create a new policy called “FileAccessManager\_S3IAMReadOnlyAccessPolicy” with all the required permissions for the connector.

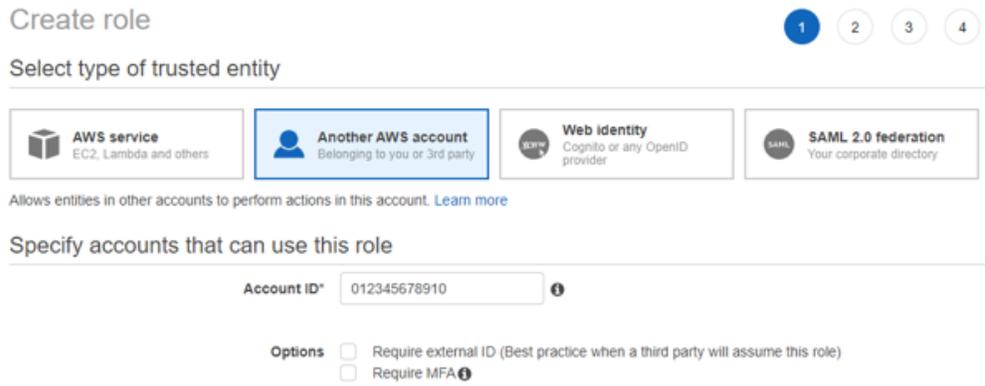
See [IdentityIQ\\_FileAccessManager\\_S3IAMReadOnlyAccessPolicy.json](#) in Appendix A.



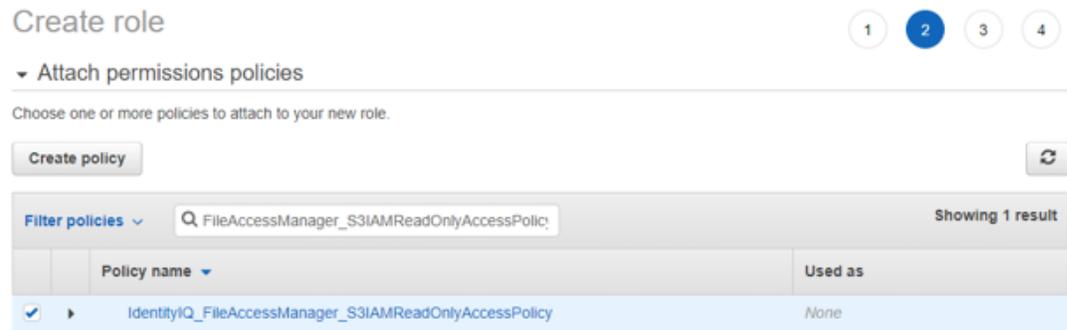
9. Create a new role for the File Access Manager user to assume.

On each organization account the connector should analyze, create a new role called “FileAccessManagerRole” which the FAM user will assume. Select “Another AWS Account” and enter the account Id of the organization’s management account.

The role name should be kept as **FileAccessManagerRole**.



10. Attach the FileAccessManager\_S3IAMReadOnlyAccessPolicy policy created above.



11. Enter the role name - FileAccessManagerRole.

## Create role



### Review

Provide the required information below and review this role before you create it.

**Role name\***   
Use alphanumeric and '+=, @-\_' characters. Maximum 64 characters.

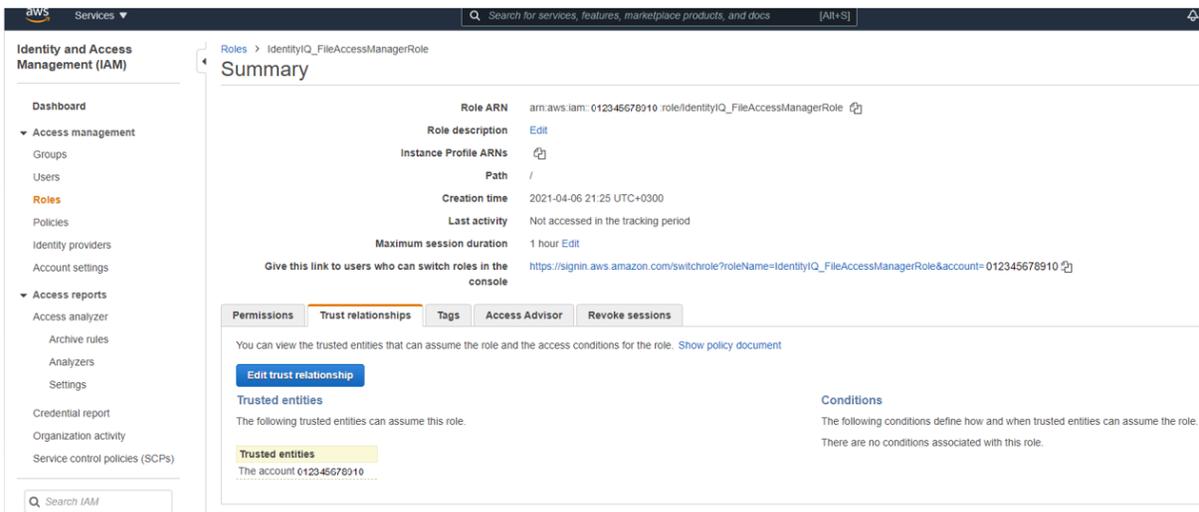
**Role description**   
Maximum 1000 characters. Use alphanumeric and '+=, @-\_' characters.

**Trusted entities** The account 012345678910

**Policies** [IdentityIQ\\_FileAccessManager\\_S3IAMReadOnlyAccessPolicy](#)

**Permissions boundary** Permissions boundary is not set

### 12. Edit the trust relationship of the new role.



### 13. Edit the json file

Replace “root” in the Principal section with

```
“assumed-role/{EC2 role name}/{EC2 instance ID}”
```

where “EC2 role name” is the name of the role created above (“FileAccessManager\_EC2\_Role“ in this manual) and “EC2 instance ID” is the ID of the instance on which the FAM application is installed.

See [IdentityIQ\\_FileAccessManagerRole.json \[EC2\]](#) in Appendix A.

### Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

Policy Document

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Principal": {
7-         "AWS": "arn:aws:sts::012345678910:assumed-role/IdentityIQ_FileAccessManager_EC2_Role/i-08f47a9c91225932e"
8-       },
9-       "Action": "sts:AssumeRole",
10-      "Condition": {}
11-     }
12-   ]
13- }
```

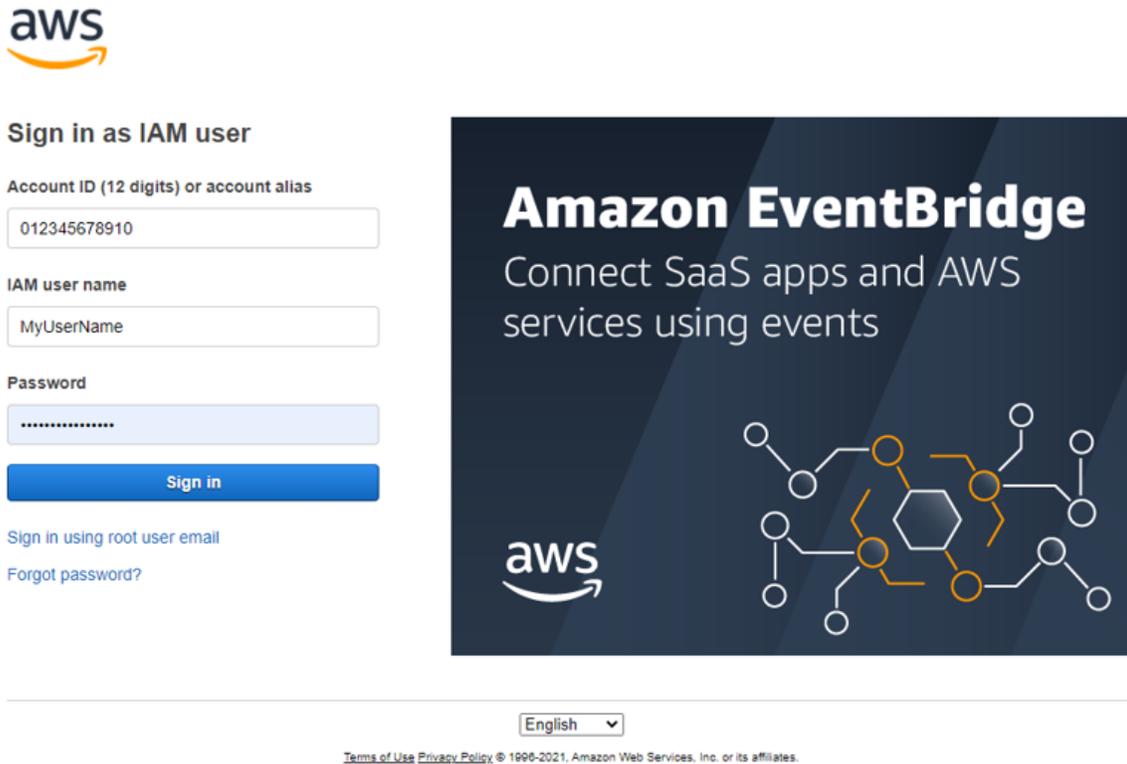
Cancel Update Trust Policy

## Creating a Dedicated IAM User

The recommended method to install the File Access Manager connector is using the EC2 Login method. See [Configuring an EC2 for File Access Manager Connector](#). If you wish to use a dedicated IAM user login instead, follow this section:

To configure the connector, create dedicated users with the appropriate users and policies

1. Sign into your organization's management account.



2. Create a new policy "IdentityIQ\_FileAccessManager\_AssumeRolePolicy". This policy will allow the File Access Manager user created in the next step to perform an **Assume Role** on the roles that will be created in each account.

See [IdentityIQ\\_FileAccessManager\\_AssumeRolePolicy.json](#) in Appendix A.

### Create policy

1 2 3

---

#### Review policy

**Name\***   
Use alphanumeric and '\*+,@-\_' characters. Maximum 128 characters.

**Description**   
Maximum 1000 characters. Use alphanumeric and '\*+,@-\_' characters.

**Summary**

Service	Access level	Resource	Request condition
Allow (1 of 276 services) <a href="#">Show remaining 275</a>			
STS	Limited: Write	RoleName   string like   IdentityIQ_FileAccessManagerRole	None

**Tags**

Key	Value
No tags associated with the resource.	

\* Required Cancel Previous Create policy

- 3. Create an IAM User for File Access Manager and select Programmatic access. This access requires an access key and secret Key.

### Add user

1 2 3 4 5

---

#### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

**User name\***   
[Add another user](#)

---

#### Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

**Access type\***  **Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

**AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

- 4. Attach the policy IdentityIQ\_FileAccessManager\_AssumeRolePolicy policy created above to the new user

[Alt+S]

## Add user 1 2 3 4 5

### Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

#### User details

<b>User name</b>	IdentityIQ_FileAccessManager_User
<b>AWS access type</b>	Programmatic access - with an access key
<b>Permissions boundary</b>	Permissions boundary is not set

#### Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	<a href="#">IdentityIQ_FileAccessManager_AssumeRolePolicy</a>

#### Tags

No tags were added.

- Save the generated Access Key and Secret Key in a secure place.

## Add user 1 2 3 4 5

✔ **Success**

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://012345678910.signin.aws.amazon.com/console>

Download .csv

	User	Access key ID	Secret access key
▶	<span style="color: green;">✔</span> IdentityIQ_FileAccessManager_User	XXXXXXXXXXXXXXXXXXXX	***** Show

- On each organization account the connector should analyze - Create new policy “*IdentityIQ\_FileAccessManager\_S3IAMReadOnlyAccessPolicy*” with all the required permissions for the connector. See the code [IdentityIQ\\_FileAccessManager\\_S3IAMReadOnlyAccessPolicy.json](#) in Appendix A.

Create policy

1 2 3

Review policy

Name\* IdentityIQ\_FileAccessManager\_S3IAMReadOnlyAccessPolicy

Use alphanumeric and '+-.\_@-' characters. Maximum 128 characters.

Description

Maximum 1000 characters. Use alphanumeric and '+-.\_@-' characters.

Summary

Filter

Service	Access level	Resource	Request condition
Allow (3 of 284 services) Show remaining 281			
IAM	Limited: List, Read	All resources	None
Organizations	Limited: List	All resources	None
S3	Limited: List, Read	All resources	None

Tags

Key Value

No tags associated with the resource.

7. Create a new role “*IdentityIQ\_FileAccessManagerRole*” which the File Access Manager user will assume on each organization account the connector should analyze. Select “**Another AWS Account**” and enter the user account ID.

Create role

1 2 3 4

Select type of trusted entity



**AWS service**  
EC2, Lambda and others



**Another AWS account**  
Belonging to you or 3rd party



**Web identity**  
Cognito or any OpenID provider



**SAML 2.0 federation**  
Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID\* 012345678910 ⓘ

- Options
- Require external ID (Best practice when a third party will assume this role)
  - Require MFA ⓘ

8. Attach the policy *IdentityIQ\_FileAccessManager\_S3IAMReadOnlyAccessPolicy* created above.

## Create role

- 1
- 2
- 3
- 4

### Attach permissions policies

Choose one or more policies to attach to your new role.

Filter policies  Showing 1 result

	Policy name	Used as
<input checked="" type="checkbox"/>	<a href="#">IdentityIQ_FileAccessManager_S3IAMReadOnlyAccessPolicy</a>	None

9. Enter the role name - *IdentityIQ\_FileAccessManagerRole*.

This name cannot be changed.

## Create role

- 1
- 2
- 3
- 4

### Review

Provide the required information below and review this role before you create it.

**Role name\***

Use alphanumeric and '+,=, @, \_' characters. Maximum 64 characters.

**Role description**

Maximum 1000 characters. Use alphanumeric and '+,=, @, \_' characters.

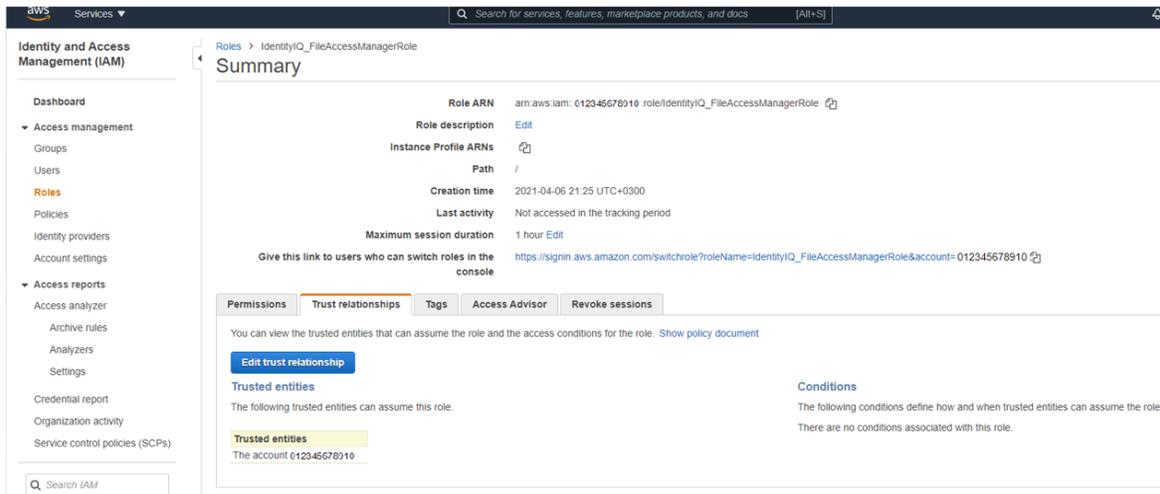
**Trusted entities** The account 012345678910

**Policies** [IdentityIQ\\_FileAccessManager\\_S3IAMReadOnlyAccessPolicy](#)

**Permissions boundary** Permissions boundary is not set

No tags were added.

10. Edit the trust relationship of the new role.



11. Edit the json file

Replace “root” in the Principal section with “user/{FAM IAM User username}” where “FAM IAM User username” is the user created above.

See [IdentityIQ\\_FileAccessManagerRole.json \[Dedicated User\]](#) in Appendix A.

## Edit Trust Relationship

You can customize trust relationships by editing the following access control policy document.

### Policy Document

```
1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": {
7         "AWS": "arn:aws:iam::012345678910:user/IdentityIQ_FileAccessManager_User"
8       },
9       "Action": "sts:AssumeRole",
10      "Condition": {}
11    }
12  ]
13 }
```

## AWS S3 Installation Flow Overview

To install the AWS S3 connector:

1. Configure all the prerequisites.
2. Add a new AWS S3 application in the Business Website.
3. Install the relevant services:
  - Permissions Collector

If you are using EC2 login, the collector should be installed on the EC2 instance.

## Collecting Data Stored in an External Application

### Terminology:

#### **Connector**

The collection of features, components and capabilities that comprise File Access Manager support for an endpoint.

#### **Collector**

The “Agent” component or service in a Permission Collection architecture.

#### **Engine**

The core service counterpart of this architecture.

#### **Identity Collector**

A logical component used to fetch identities from an identity store and holds the configuration, settings for that identity store, and the relations between these identities.

The identity collector has no “physical” manifest.

- The actual work is done by the Collector Synchronizer.

The list below describes the high level installation process required to collect and analyze data from an external application. Most of these should already be set up in your File Access Manager installation. See the server Installation guide for further details.

#### **Install a Permission Collection central engine**

One or more central engines, installed using the server installer

#### **Create an Application in File Access Manager**

From the Business Website. The application is linked to central engines listed above.

#### **Install Permission Collectors (optional)**

Optionally, you can install collectors that will run on a separate server and take some of the work from the central PC and DC engines (Where supported). When installing a collector, you attach it to an engine. If no collectors are installed, the central services act as both the engine and the collector.

To install a collector, you must have the **RabbitMQ** service installed for communication between the central engines and the collectors. RabbitMQ is installed

For further details, see section **Application > Central Service > Collector Relations** in the File Access Manager Administrator Guide

## Adding an AWS S3 Application

In order to integrate with AWS S3, we must first create an application entry in File Access Manager. This entry includes the identification, connection details, and other parameters necessary to create the link.

To add an application, use the **New Application Wizard**.

1. Navigate to *Admin > Applications*
2. Click **Add New** to open the wizard.

### Select Wizard Type

1. Click **Standard Application**
2. Click **Next** to open the **General Details** page.

### General Details

#### **Application Type**

AWS S3

#### **Application Name**

Logical name of the application

#### **Description**

Description of the application

#### **Tags**

Select tags for the application from the dropdown menu, and / or type a new name, and press **Enter** to create a new tag. The dropdown list of tags filters out matching tags as you type and displays up to 50 tags.

The **tags** replace the **Logical container** field that was used when creating applications in releases before 8.2

#### **Event Manager Server**

This option is available if there are more than one event manager servers configured in the system.

Select an event manager from the drop down menu.

Click **Next**. to open the Connection Details page.

### Connection Details

Complete the Connection Details fields:

#### **Server Name**

The name of the CTERA Master Gateway

#### **Domain Name**

The user defined in the prerequisites

### **User / Password**

Credentials of the user defined in the prerequisites (This must be an admin user on the CTERA master gateway)

Click **Next**.

## **Configuring and Scheduling the Permissions Collection**

Permissions can be analyzed to determine the application permissions of an out-of-the-box application, provided you have defined an identity store for File Access Manager to use in its analysis, and you have run a crawl for the application.

The permission collector is a software component responsible for analyzing the permissions in an application.

The Central Permission Collector Service is responsible for running the Permission Collector and Crawler tasks.

If the "FAM Central Permission Collector" wasn't installed during the installation of the server, this configuration setting will be disabled.

### **To configure the Permission Collection**

- Open the edit screen of the required application.
  - a. Navigate to **Admin > Applications**.
  - b. Scroll through the list, or use the filter to find the application.
  - c. Click the edit icon  on the line of the application.
- Press **Next** till you reach the **Crawler & Permissions Collection** settings page.

The actual entry fields vary according to the application type.

When entering this page in edit mode, you can navigate between the various configuration windows using the **Next** and **Back** buttons.

### **Central Permissions Collection Service**

Select a central permission collection service from the dropdown list. You can create permissions collection services as part of the service installation process. See section "Services Configuration" in the File Access Manager Administrator Guide for further details.

### **Skip Identities Sync during Permission Collection**

Skip identity synchronization before running permission collection tasks when the identity collector is common to different connector.

This option is checked by default.

### **Active Directory Group Regex**

If matching an Active Directory group to an AWS IAM role is done by the Active Directory group naming convention, enter a regex. This will enable extracting the AWS account ID and name role from the group.

The regex must include these exact named groups in this exact format:

- <rolename>
- <accountid>

## Scheduling a Task

### Create a Schedule

Click on this option to view the schedule setting parameters.

### Schedule Task Name

A name for this scheduling task

When creating a new schedule, the system generates a default name in the following format:

{appName} - {type} Scheduler

You can override or keep this name suggestion.

### Schedule

Select a scheduling frequency from the dropdown menu.

- **Schedule Types and Intervals**

#### Once

Single execution task runs.

#### Run After

Create dependency of tasks. The task starts running only upon successful completion of the first task.

#### Hourly

Set the start time.

#### Daily

Set the start date and time.

#### Weekly

Set the day(s) of the week on which to run.

#### Monthly

The start date defines the day of the month on which to run a task.

#### Quarterly

A monthly schedule with an interval of 3 months.

#### Half Yearly

A monthly schedule with an interval of 6 months.

### **Yearly**

A monthly schedule with an interval of 12 months.

### **Date and time fields**

Fill in the scheduling times. These fields differ, depending upon the scheduling frequency selected.

### **Active check box**

Check this to activate the schedule.

Click **Next**.

## **Configuring and Scheduling the Crawler**

### **To set or edit the Crawler configuration and scheduling**

- Open the edit screen of the required application.
  - a. Navigate to **Admin > Applications**.
  - b. Scroll through the list, or use the filter to find the application.
  - c. Click the edit icon  on the line of the application.
- Press **Next** till you reach the **Crawler & Permissions Collection** settings page.

The actual entry fields vary according to the application type.

### **Create a Schedule**

Click to open the schedule panel.

### **Setting the Crawl Scope**

There are several options to set the crawl scope:

- Setting explicit list of resources to include and / or exclude from the scan.
- Creating a regex to define resources to exclude.

### **Including and Excluding Paths by List**

#### **To set the paths to include or exclude in the crawl process for an application**

- Open the edit screen of the required application.
  - a. Navigate to **Admin > Applications**.
  - b. Scroll through the list, or use the filter to find the application.
  - c. Click the edit icon  on the line of the application.
- Press **Next** till you reach the **Crawler & Permissions Collection** settings page.

The actual entry fields vary according to the application type.

1. Scroll down to the Crawl configuration settings.
2. Click **Advanced Crawl Scope Configuration** to open the scope configuration panel.
3. Click Include / Exclude Resources to open the input fields.
4. To add a resource to a list, type in the full path to include / exclude in the top field and click **+** to add it to the list.
5. To remove a resource from a list, find the resource from the list, and click the x icon on the resource row.

When creating exclusion lists, excludes take precedence over includes.

### Excluding Paths by Regex

**To set filters of paths to exclude in the crawl process for an application using regex.**

- Open the edit screen of the required application.
  - a. Navigate to **Admin > Applications**.
  - b. Scroll through the list, or use the filter to find the application.
  - c. Click the edit icon  on the line of the application.
- Press **Next** till you reach the **Crawler & Permissions Collection** settings page.

The actual entry fields vary according to the application type.

1. Click **Exclude Paths by Regex** to open the configuration panel.
2. Type in the paths to exclude by Regex, See regex examples in the section below. Since the system does not collect BRs that match this Regex, it also does not analyze them for permissions.

### Crawler Regex Exclusion Example

The following are examples of crawler Regex exclusions:

**Exclude all bucket folders which start with one or more folder names:**

Example: All Starting with folderName under path

Regex: `^Root\[account_name\](#[AccountID])\s3.[region].[bucket_name]\folder_name`

Real Example:

Path: `Root/my-account(#1234567890)/s3.ap-south-1.bucket1/myFolder`

Regex: `^Root/my-account\(#1234567890)\s3.ap-south-1.bucket1\myFolder`

---

Example: All starting with folderName of otherFolderName under path

Regex: `^Root\[account_name\](#[AccountID])\s3.[region].[bucket_name]\(folder-Name|otherFolderName)`

---

**Include ONLY bucket folders that start with one or more folder names**

Example: Starting with folderName under path

Regex: `^(?!Root($|V[account_name]\#[AccountID])($|Vs3.[region].[bucket_name]($|V-folder_name($|/.*)$)))$`

Real Example:

Path: `Root/FAM_Test(#1234567890)/s3.us-west-1.service/logs/logs_01`

Path: `Root/FAM_Test(#1234567890)/s3.us-west-1.service/logs/logs_02`

Path: `Root/FAM_Test(#1234567890)/s3.us-west-1.service/logs/logs_03`

Regex: `^(?!Root\FAM_Test($|\(#1234567890\))($|Vs3.us-west-1.service($|Vlogs($|/.*)$)))$`

---

Example: Starting with folderName of otherFolderName under path

Regex: `^(?!Root($|V[account_name]\#[AccountID])($|Vs3.[region].[bucket_name]($|V-folder_Name|other_Folder_Name($|/.*)$)))$`

---

## Excluding Top Level Resources

Use the top level exclusion screen to select top level roots to exclude from the crawl. This setting is done per application.

### To exclude top level resources from the crawl process

1. Open the application screen

*Admin > Applications*

2. Find the application to configure and click the drop down menu on the application line. Select **Exclude Top Level Resources** to open the configuration panel.

3. **Run Task**

The Run Task button triggers a task that runs a short detection scan to detect the current top level resources.

Before running the task for the first time, the message above this button is:

#### **"Note: Run task to detect the top-level resources"**

If the top level resource list has changed in the application while you are on this screen, press this button to retrieve the updated structure.

Once triggered, you can see the task status in

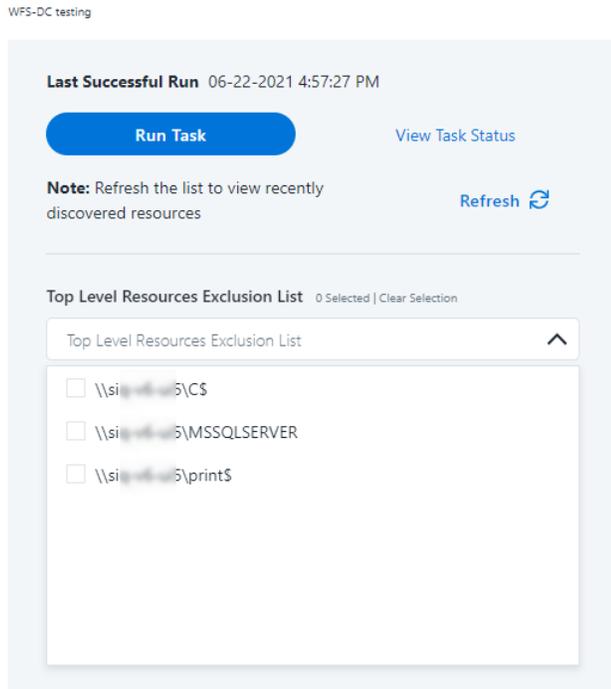
*Settings > Task Management > Tasks*

This will only work if the user has access to the task page

When the task has completed, press **Refresh** to update the page with the list of top level resources.

4. Click the top level resource list, and select top level resources to exclude.
5. Click **Save** to save the change.
6. To refresh the list of top level resources, run the task again. Running the task will not clear the list of top level resources to exclude.

### Top Level Resources Exclusion ×



### Special Consideration for Long File Paths in Crawl

If you need to support long file paths above 4,000 characters for the crawl, set the flag

**`excludeVeryLongResourcePaths`**

in the Permission Collection Engine App.config file to true.

By default this value will be commented out and set to false.

This key ensures, when enabled, that paths longer than 4000 characters are excluded from the applications' resource discovery (Crawl), to avoid issues while storing them in the SQLServer database.

When enabled, business resources with full paths longer than 4000 characters, and everything included in the hierarchical structure below them, will be excluded from the crawl, and will not be collected by File Access Manager. This scenario is extremely rare.

You should not enable exclusion of long paths, unless you experience an issue.

### Background

File Access Manager uses a hashing mechanism to create a unique identifier for each business resource stored in the File Access Manager database. The hashing mechanism in SQL Server versions 2014 and earlier, is unable to process (hash) values with 4,000 or more characters.

Though resources with paths of 4000 characters or longer are extremely rare, File Access Manager is designed to handle that limitation.

### ***Identifying the Problem***

When using an SQL Server database version 2014 and earlier

The following error message in the Permission Collection Engine log file:

```
System.Data.SqlClient.SqlException (0x80131904): String or binary data would be truncated.
```

In all other cases, this feature should not be enabled.

### ***Setting the Long Resource Path Key***

The Permission Collection Engine App.config file is `RoleAnalyticsServiceHost.exe.config`, and can be found in the folder

`%SailPoint_Home%\FileAccessManager\[Permission Collection instance]\`

Search for the key **`excludeVeryLongResourcePaths`** and correct it as described above.

## Active Directory Integration with AWS

Active Directory has the ability to be integrated with AWS environments and allow users to use their already established login credentials, manage their user identities outside of AWS, and give these external user identities permissions to use AWS resources in their account.

When integrating Active Directory to AWS, the AWS S3 permissions needs to be mapped to the Active Directory users and groups by using an Identity Provider (IdP).

We support AWS 'SAML' and 'OpenID Connect' IdPs in case this is done in one of the following two ways:

- An internal configuration inside the IDP. This would be supported by using a File Access Manager data source and would include a mapping file (Excel, CSV, etc.) that the client needs to provide and maintain.

	A	B	C	D	E	F	G	H
1	ADSid	ADDomain	AWSRoleArn					
2	5-1-5-21-554565666-4561215555-12345678-513	WHITEBOX	arn:aws:iam::832879123456:role/Okta_IDP_Role_2					
3	5-1-5-21-8787575454-232323232-5454545-572	OFFICE	arn:aws:iam::832879123456:role/Okta_IDP_Role_2					
4								
5								

- Using Active Directory group naming configuration. This method is ideal in case the client's IDP supports it and if the client created these groups.

**Example:** Active Directory group name – ad-aws-int-test1#Okta\_IDP\_Role\_2#832879285990

This is the Active Directory group name template: [some name]#[role name]#[account id].

The user configures (in File Access Manager) the regular expression (regex)

**Example:** S+\#(?<rolename>[\w\-\+]#\#(?<accountid>\d+)\\$. We then know to use this expression to extract the IAM Role name and the AWS account ID from the Active Directory group name and do the mapping.

## Mapping Extractions from IDPs

This section provide the steps to extract mappings from the following IDPs:

- Okta
- ADFS
- Azure
- Ping

### Okta

In Okta, use the Okta API [Reference Overview: Okta Developer](#), to get the Active Directory identities - AWS identities mappings.

1. Get the AWS application and extract the Account ID from the "identityProviderArn" property.

### Okta Documentation

[Apps: Okta Developer](#)

### Request Example

```
https://{yourOktaDomain}/api/v1/apps/{applicationId}
```

### Response Example

```
{
  "id": "0oapruvo3xnNEuI12345",
  "name": "amazon_aws",
  "label": "AWS Account Federation",
  "status": "ACTIVE",
  "lastUpdated": "2021-08-02T14:51:07.000Z",
  "created": "2021-07-22T11:00:28.000Z",
  "accessibility": {
    "selfService": false,
    "errorRedirectUrl": null,
    "loginRedirectUrl": null
  },
  "visibility": {
    "autoLaunch": false,
    "autoSubmitToolbar": true,
    "hide": {
      "iOS": false,
      "web": false
    },
    "appLinks": {
      "login": true
    }
  },
  "features": [
    "PUSH_NEW_USERS",
  ]
}
```

```

    "PUSH_PROFILE_UPDATES"
  ],
  "signOnMode": "SAML_2_0",
  "credentials": {
    "userNameTemplate": {
      "template": "${source.login}",
      "type": "BUILT_IN"
    },
    "signing": {
      "kid": "BNfWuNclhWcvmRpgv2C8MoP1A34vLbDMNQ2odOK97VY"
    }
  },
  "settings": {
    "app": {
      "appFilter": "okta",
      "groupFilter": "aws_(?{accountid}\\d+)_(?{role}[a-zA-Z0-9+.,@\\-_]+)",
      "secretKey": null,
      "useGroupMapping": true,
      "joinAllRoles": true,
      "identityProviderArn": "arn:aws:iam::832879212345:saml-provider/okta2",
      "overrideAcsURL": null,
      "sessionDuration": 3600,
      "roleValuePattern": "arn:aws:iam::${accountid}:saml-provider/okta2,
arn:aws:iam:${accountid}:role/${role}",
      "awsEnvironmentType": "aws.amazon",
      "accessKey": null,
      "loginURL": "https://console.aws.amazon.com/ec2/home",
      "secretKeyEnc": null
    },
    "notifications": {
      "vpn": {
        "network": {
          "connection": "DISABLED"
        },
        "message": null,
        "helpUrl": null
      }
    },
    "notes": {
      "admin": null,
      "enduser": null
    },
    "signOn": {
      "defaultRelayState": null,
      "ssoAcsUrlOverride": null,
      "audienceOverride": null,
      "recipientOverride": null,
      "destinationOverride": null,
      "attributeStatements": []
    }
  },
  "_links": {
    "help": {
      "href": "https://sailpointamirmono-admin.okta.com/app/amazon_
aws/0oapruvo3xnNEuI12345/setup/help/SAML_2_0/external-doc",

```

```

        "type": "text/html"
    },
    "metadata": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/sso/saml/metadata",
        "type": "application/xml"
    },
    "uploadLogo": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/logo",
        "hints": {
            "allow": [
                "POST"
            ]
        }
    },
    "appLinks": [
        {
            "name": "login",
            "href": "https://sailpointamirmono.okta.com/home/amazon_
aws/0oapruvo3xnNEuI12345/272",
            "type": "text/html"
        }
    ],
    "groups": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/groups"
    },
    "logo": [
        {
            "name": "medium",
            "href": "https://ok14static.oktacdn.com/fs/bcg/4/gfs1f2p5y2qNcK02w1d8",
            "type": "image/png"
        }
    ],
    "users": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/users"
    },
    "deactivate": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/lifecycle/deactivate"
    }
}
}

```

2. Get the applications users and groups and extract the role names from "profile" > "role".
3. Build the role ARN from the Account ID and Role Name and get the user and group Okta ID.

[Groups: Okta Developer](#)

[Users: Okta Developer](#)

### **Request Example**

https://{yourOktaDomain}/api/v1/apps/{applicationId}/users

https://{yourOktaDomain}/api/v1/apps/{applicationId}/groups

### Response Example

```
[
  {
    "id": "00gpsbh7o30J0foeV695",
    "lastUpdated": "2021-08-22T14:32:44.000Z",
    "priority": 0,
    "profile": {
      "role": "AWSServiceRoleForCloudTrail",
      "samlRoles": [
        "Okta_IDP_Role_2"
      ]
    },
    "_links": {
      "app": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345"
      },
      "self": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/groups/00gpsbh7o30J0fo12345"
      },
      "group": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/groups/00gpsbh7o30J0fo12345"
      }
    }
  },
  {
    "id": "00gyrmmrG0kWUyKGf695",
    "lastUpdated": "2021-08-22T14:35:17.000Z",
    "priority": 1,
    "profile": {
      "role": "AWSServiceRoleForCloudTrail",
      "samlRoles": [
        "Okta_IDP_Role"
      ]
    },
    "_links": {
      "app": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345"
      },
      "self": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/apps/0oapruvo3xnNEuI12345/groups/00gyrmmrG0kWUyK12345"
      },
      "group": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/groups/00gyrmmrG0kWUyK12345"
      }
    }
  }
]
```

```
    }  
  }  
]
```

4. List all the groups and users and get the groups and users names by the ID.

#### Okta Documentation

[Groups: Okta Developer](#)

[Users: Okta Developer](#)

#### Request Examples

`https://{yourOktaDomain}/api/v1/groups`

`https://{yourOktaDomain}/api/v1/users`

#### Response Example

```
[  
  {  
    "id": "00gymrmrG0kWUyK12345",  
    "created": "2021-07-29T10:40:08.000Z",  
    "lastUpdated": "2021-07-29T10:40:08.000Z",  
    "lastMembershipUpdated": "2021-07-29T10:41:25.000Z",  
    "objectClass": [  
      "okta:user_group"  
    ],  
    "type": "OKTA_GROUP",  
    "profile": {  
      "name": "aws_832879285990_Okta_IDP_Role_2",  
      "description": null  
    },  
    "_links": {  
      "logo": [  
        {  
          "name": "medium",  
          "href": "https://ok14-  
static.oktacdn.com/assets/img/logos/groups/odyssey/okta-medium.1a5ebe44c4244fb796c235d86b47e3bb.png",  
          "type": "image/png"  
        },  
        {  
          "name": "large",  
          "href": "https://ok14-  
static.oktacdn.com/assets/img/logos/groups/odyssey/okta-large.d9cf-  
bd8a00a4feac1aa5612ba02e99c0.png",  
          "type": "image/png"  
        }  
      ],  
      "users": {  
        "href": "https://-  
sailpointamirmono.okta.com/api/v1/groups/00gymrmrG0kWUyK12345/users"  
      }  
    }  
  }  
]
```

```

    },
    "apps": {
      "href": "https://-
sailpointamirmono.okta.com/api/v1/groups/00gymrnrG0kWUyK12345/apps"
    }
  },
  {
    "id": "00gpsbh7o30J0fo12345",
    "created": "2021-07-22T09:26:50.000Z",
    "lastUpdated": "2021-07-22T09:26:50.000Z",
    "lastMembershipUpdated": "2021-07-29T10:41:25.000Z",
    "objectClass": [
      "okta:user_group"
    ],
    "type": "BUILT_IN",
    "profile": {
      "name": "Everyone",
      "description": "All users in your organization"
    },
    "_links": {
      "logo": [
        {
          "name": "medium",
          "href": "https://ok14-
static.oktacdn.com/assets/img/logos/groups/odyssey/okta-medium.1a5ebe44c4244fb796c235d86b47e3bb.png",
          "type": "image/png"
        },
        {
          "name": "large",
          "href": "https://ok14-
static.oktacdn.com/assets/img/logos/groups/odyssey/okta-large.d9cf-
bd8a00a4feac1aa5612ba02e99c0.png",
          "type": "image/png"
        }
      ],
      "users": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/groups/00gpsbh7o30J0fo12345/users"
      },
      "apps": {
        "href": "https://-
sailpointamirmono.okta.com/api/v1/groups/00gpsbh7o30J0fo12345/apps"
      }
    }
  }
]

```

## ADFS

In ADFS, the Active Directory identities-AWS identities mapping is done on one of the Active Directory identity attributes.

For more information, see [Establish Federated Access to AWS Resources by Using AD User Attributes](#). See - A. Configure an AD user's account.

Filter all the users and groups with the specific attribute and export it to a csv or Excel file.

### PS Example

```
Get-ADUser -Filter 'url -like "*AWS*"' -properties "url" | Export-Csv c:\file.csv
```

### Response Example

```
#TYPE Microsoft.ActiveDirectory.Management.ADUser,,,,,,,,,
DistinctName,Enabled,GivenName,Name,ObjectClass,ObjectGUID,SamAccountName,SID,Surname,url,UserPrincipalName
"CN=Adiel,CN=Users,DC=office,DC=whitebox,DC=forest",TRUE,Adiel,Adiel,user,e3fe35c1-0daf-4379-a379-73364ec12345,Adiel,S-1-5-21-3335839157-1594281566-240188981-12345,Moshed,Microsoft.ActiveDirectory.Management.ADPropertyValueCollection,Adiel@office.whitebox.forest
```

Remember that the response will be exported to a csv or Excel file.

## Azure AD

In Azure AD, it is possible to get the AD identities-AWS identities mapping by using Microsoft Graph.

1. Get all the AWS account's roles by the "AWS Single-Account Access" Object ID (one account per request).
2. Acquire the roles ARNs.

### Request Example

```
https://graph.microsoft.com/beta/servicePrincipals/{AWS Single-Account Access object id}
```

### Response Example

```
{
  "@odata.context": "https://graph.microsoft.com/beta/$metadata#servicePrincipals/$entity",
  "@odata.id": "https://graph.microsoft.com/v2/154dccc9-b44e-4883-860c-12345/directoryObjects/726e2abf-b192-462d-a977-12345/Microsoft.DirectoryServices.ServicePrincipal",
  "id": "726e2abf-b192-462d-a977-12345",
  "deletedDateTime": null,
  "accountEnabled": true,
  "alternativeNames": [],
  "createdDateTime": "2021-09-05T11:27:45Z",
  "deviceManagementAppType": null,
  "appDescription": null,
  "appDisplayName": "AWS Single-Account Access",
  "appId": "944b9a2c-51dd-41eb-a018-12345",
}
```

```

"applicationTemplateId": "8b1025e4-1dd2-430b-a150-12345",
"appOwnerOrganizationId": "154dccc9-b44e-4883-860c-12345",
"appRoleAssignmentRequired": true,
"description": null,
"disabledByMicrosoftStatus": null,
"displayName": "AWS Single-Account Access",
"errorUrl": null,
"homepage": "https://signin.aws.amazon.com/saml?metadata=aws|ISV9.1|primary|z",
"isAuthorizationServiceEnabled": false,
"isManagementRestricted": null,
"loginUrl": null,
"logoutUrl": null,
"notes": null,
"notificationEmailAddresses": [
  "admin@501.sailpointtechnologies.com"
],
"preferredSingleSignOnMode": "saml",
"preferredTokenSigningKeyEndDateTime": null,
"preferredTokenSigningKeyThumbprint": null,
"publisherName": "SailPoint Technologies, Inc.",
"replyUrls": [
  "https://signin.aws.amazon.com/saml"
],
"samlMetadataUrl": null,
"servicePrincipalNames": [
  "944b9a2c-51dd-41eb-a018-12345"
],
"servicePrincipalType": "Application",
"signInAudience": "AzureADMyOrg",
"tags": [
  "WindowsAzureActiveDirectoryIntegratedApp"
],
"tokenEncryptionKeyId": null,
"samlSingleSignOnSettings": null,
"verifiedPublisher": {
  "displayName": null,
  "verifiedPublisherId": null,
  "addedDateTime": null
},
"addIns": [],
"api": {
  "resourceSpecificApplicationPermissions": []
},
"appRoles": [
  {
    "allowedMemberTypes": [
      "User"
    ],
    "description": "msiam_access",
    "displayName": "msiam_access",
    "id": "7dfd756e-8c27-4472-b2b7-12345",
    "isEnabled": true,
    "origin": "Application",
    "value": null
  }
]

```

```

    },
    {
      "allowedMemberTypes": [
        "User"
      ],
      "description": "ChessPlayersRole",
      "displayName": "ChessPlayersRole,Okta1",
      "id": "2d9e11e2-14c9-4f34-bf19-12345",
      "isEnabled": true,
      "origin": "ServicePrincipal",
      "value": "arn:aws:iam::832879212345:role/ChessPlay-
ersRole,arn:aws:iam::832879212345:saml-provider/Okta1"
    },
    {
      "allowedMemberTypes": [
        "User"
      ],
      "description": "DOMAIN_ALIAS_RID_ADMIN-AWS",
      "displayName": "DOMAIN_ALIAS_RID_ADMIN-AWS,Azure_test1",
      "id": "ad3d751a-b615-4bf7-930b-c06a62712345",
      "isEnabled": true,
      "origin": "ServicePrincipal",
      "value": "arn:aws:iam::832879212345:role/DOMAIN_ALIAS_RID_ADMIN-
AWS,arn:aws:iam::832879212345:saml-provider/Azure_test1"
    }
  ],
  "info": {
    "termsOfServiceUrl": null,
    "supportUrl": null,
    "privacyStatementUrl": null,
    "marketingUrl": null,
    "logoUrl": null
  },
  "keyCredentials": [],
  "publishedPermissionScopes": [
    {
      "adminConsentDescription": "Allow the application to access AWS Single-
Account Access on behalf of the signed-in user.",
      "adminConsentDisplayName": "Access AWS Single-Account Access",
      "id": "419e3996-3684-4265-890a-12345",
      "isEnabled": true,
      "type": "User",
      "userConsentDescription": "Allow the application to access AWS Single-Account
Access on your behalf.",
      "userConsentDisplayName": "Access AWS Single-Account Access",
      "value": "user_impersonation"
    }
  ],
  "passwordCredentials": [],
  "resourceSpecificApplicationPermissions": []
}

```

3. Get the users and groups which are assigned to the AWS roles.
4. Acquire the users and groups details.

**Request Example**

```
https://graph.microsoft.com/beta/servicePrincipals/{AWS Single-Account Access object id}/appRoleAssignedTo
```

**Response Example**

```
{
  "@odata.context": "https://graph.microsoft.com/beta/$metadata#appRoleAssignments",
  "value": [
    {
      "@odata.id": "https://graph.microsoft.com/v2/154dccc9-b44e-4883-860c-12345/directoryObjects/$/Microsoft.DirectoryServices.ServicePrincipal('726e2abf-b192-462d-a977-12345')/appRoleAssignedTo/v9raS1IPQkuV98HJH2Uqhsg4ilzG80Z0i00My-8m5iw",
      "id": "v9raS1IPQkuV98HJH2Uqhsg4ilzG80Z0i00My-8m5iw",
      "creationTimestamp": "2021-09-09T11:45:26.3084935Z",
      "appRoleId": "d3a9b01b-1736-4f1b-ac5f-12345",
      "principalDisplayName": "anatoly_azure_gr1",
      "principalId": "4bdadabf-0f52-4b42-95f7-12345",
      "principalType": "Group",
      "resourceDisplayName": "AWS Single-Account Access",
      "resourceId": "726e2abf-b192-462d-a977-12345"
    },
    {
      "@odata.id": "https://graph.microsoft.com/v2/154dccc9-b44e-4883-860c-12345/directoryObjects/$/Microsoft.DirectoryServices.ServicePrincipal('726e2abf-b192-462d-a977-12345')/appRoleAssignedTo/CF0PHVm9hka00WBTgEPxaoZKebW4inxCsBpqIGxRwFI",
      "id": "CF0PHVm9hka00WBTgEPxaoZKebW4inxCsBpqIGxRwFI",
      "creationTimestamp": "2021-09-09T11:45:26.3302622Z",
      "appRoleId": "d3a9b01b-1736-4f1b-ac5f-12345",
      "principalDisplayName": "anatoly_azure_group3",
      "principalId": "1d0f5d08-bd59-4686-b4d1-12345",
      "principalType": "Group",
      "resourceDisplayName": "AWS Single-Account Access",
      "resourceId": "726e2abf-b192-462d-a977-12345"
    },
    {
      "@odata.id": "https://graph.microsoft.com/v2/154dccc9-b44e-4883-860c-12345/directoryObjects/$/Microsoft.DirectoryServices.ServicePrincipal('726e2abf-b192-462d-a977-12345')/appRoleAssignedTo/INRoSKbmpUaZrnYaVRU3XMRgM8C1kZ9GjHjSB9vW1e4",
      "id": "INRoSKbmpUaZrnYaVRU3XMRgM8C1kZ9GjHjSB9vW1e4",
      "creationTimestamp": "2021-09-09T11:32:47.4228653Z",
      "appRoleId": "277f83e1-4903-4b06-baf7-12345",
      "principalDisplayName": "Adiel",
      "principalId": "4868d420-e6a6-46a5-99ae-12345",
      "principalType": "User",
      "resourceDisplayName": "AWS Single-Account Access",
      "resourceId": "726e2abf-b192-462d-a977-12345"
    }
  ]
}
```

## Installing Services: Activity Monitor and Collectors

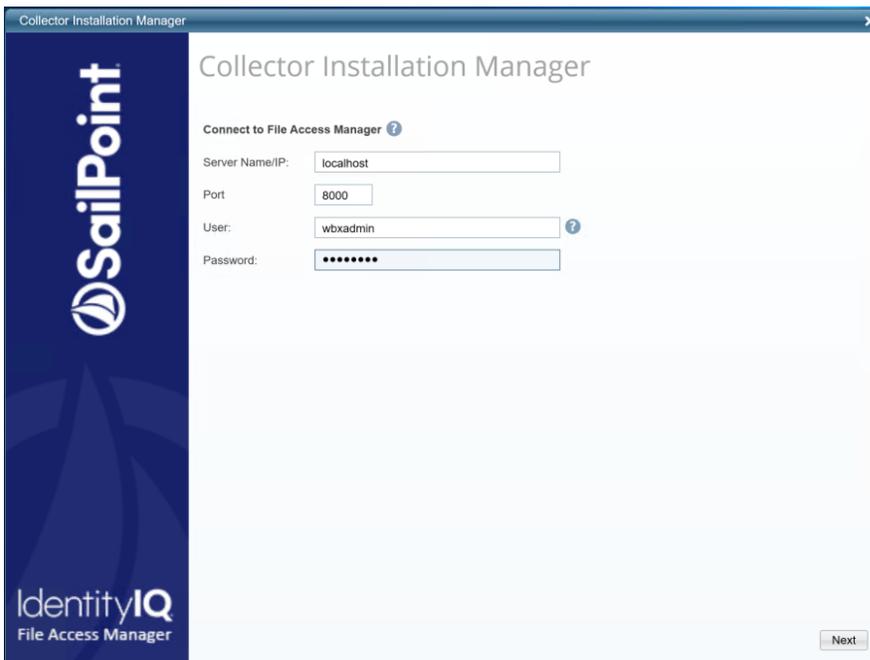
The Collector Installation Manager is part of the File Access Manager installation package. This tool is used to install the activity monitor, permission collector, and data classification collector.

### Activity Monitor

The activity monitor is installed per application, and collects SharePoint Audit entries and IIS activity logs.

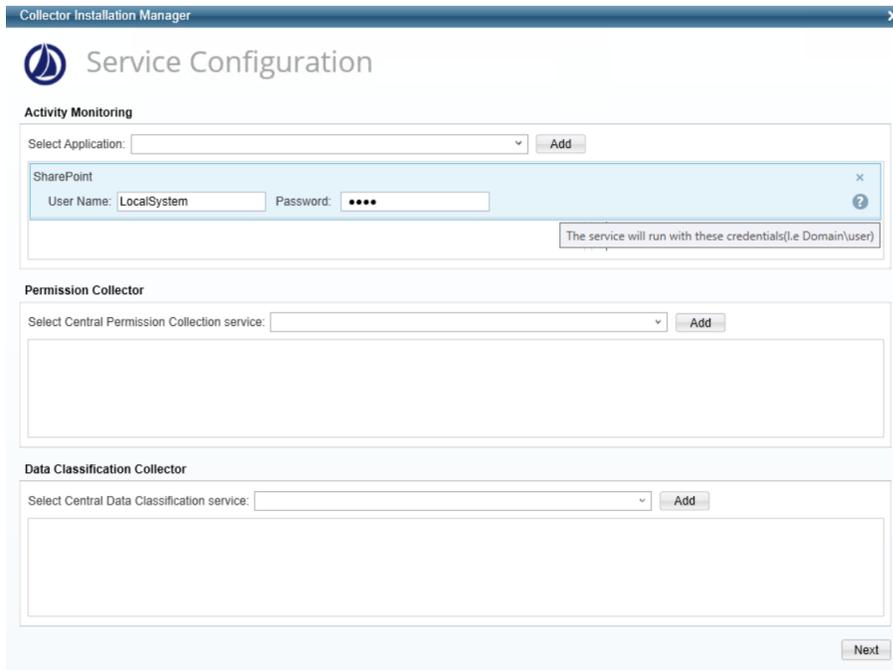
1. Run the **Collector Installation Manager** as an Administrator.  
The installation files are in the installation package under the folder **Collectors**.

The Collector Installation Manager window displays.



2. Enter the credentials to connect to File Access Manager.
  - a. ServerName/IP should be pointed to the Agent Configuration Manager service server.
  - b. An File Access Manager user with Collector Manager permission (permission to install collectors). For Active Directory authentication, use the format domain/username.
3. Click **Next**.

The Service Configuration window displays.



4. When installing a SharePoint Activity Monitor, you will be prompted for service account credentials. This service account will be used by the Activity Monitor service to run the service and authenticate against the SharePoint IIS servers to fetch the logs (“Log on as”). Make sure the service account provided has local administrator privileges on the local server (hosting the Activity Monitor service) and can access the activity logs on the IIS servers.
5. If you are installing the Permission Collector, select the Central Permission Collector to which to connect this service, and click **Add**
6. Click **Next**.

The Installation Folder window displays.

If this is the first time you are installing collectors on this machine, you will be prompted to select an installation folder, in which all future collectors will also be installed.

7. Browse and select the location of the target folder for installation.
8. Browse and select the location of the folder for system logs.
9. Click **Next**.
10. The system begins installing the selected components.
11. Click **Finish**

The Finish button is displayed after all the selected components have been installed.

The *File Access Manager Administrator Guide* provides more information on the collector services.

# Verifying the AWS S3 Connector Installation

## Installed Services

Verify that the services installed for the connector are available and active. Using windows Service manager, or other tool, look for the File Access Manager services, and see that they are running.

for example:

- File Access Manager Central Permissions Collection - <Application\_Name>

## Log Files

Check the log files listed below for errors

- "%SAILPOINT\_HOME\_LOGS%\PermissionCollection\_<Service\_Name>.log"

## Permissions Collection

1. Run the Crawler and Permissions Collector tasks (*Settings > Task Management > Scheduled Tasks*)
2. Verify that:
  - The tasks completed successfully
  - Business resources were created in the resource explorer (*Admin > Applications > [application column] > Manage Resources*)
  - Permissions display in the Permission Forensics page (*Forensics > Permissions*)

## Appendix A: Json Scripts

This appendix includes the scripts required for creating the roles and policies mentioned in this guide.

Please make sure not to change the file names.

### IdentityIQ\_FileAccessManagerRole.json [EC2]

This is the version of the role to create for installation using an EC2 login

IdentityIQ\_FileAccessManagerRole.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::{\The EC2 instance account Id}:assumed-role/{EC2 instance
role name}/{EC2 instance Id}"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

### IdentityIQ\_FileAccessManagerRole.json [Dedicated User]

This is the version of the role to create for installation using a dedicated IAM user login

IdentityIQ\_FileAccessManagerRole.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::{\The user account ID}:user/{FAM IAM User username}"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## IdentityIQ\_FileAccessManager\_AssumeRolePolicy.json

IdentityIQ\_FileAccessManager\_AssumeRolePolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::*:role/IdentityIQ_FileAccessManagerRole"
    }
  ]
}
```

## IdentityIQ\_FileAccessManager\_S3IAMReadOnlyAccessPolicy.json

IdentityIQ\_FileAccessManager\_S3IAMReadOnlyAccessPolicy.json

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "s3:GetBucketPolicy",
        "s3:GetBucketPolicyStatus",
        "s3:GetBucketPublicAccessBlock",
        "s3:GetAccountPublicAccessBlock",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "iam:ListAttachedGroupPolicies",
        "iam:ListAttachedRolePolicies",
        "iam:ListAttachedUserPolicies",
        "iam:ListGroupPolicies",
        "iam:ListGroups",
        "iam:ListPolicies",
        "iam:ListPolicyVersions",
        "iam:ListRolePolicies",
        "iam:ListRoles",
        "iam:ListUserPolicies",
        "iam:ListUsers",
        "iam:GetGroup",
        "iam:GetGroupPolicy",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRolePolicy",

```

```
        "iam:GetUserPolicy",
        "organizations:ListAccountsForParent",
        "organizations:ListRoots",
        "organizations:ListAccounts",
        "organizations:ListOrganizationalUnitsForParent"
    ],
    "Resource": "*"
}
]
```