fn-aws-iam Functions for IBM Resilient

- Release Notes
- Overview
- Requirements
- Installation
- Uninstall
- Troubleshooting
- Support

Release Notes

v1.0.1

- Support added for App Host.

v1.0.0

- Initial Release

Overview

Amazon AWS IAM Integration for Resilient

Amazon Web Services Identity and Access Management (AWS IAM) allows management of access to AWS services and resources securely. You can use IAM to create and manage AWS users and groups, and use permissions to allow or deny access to AWS resources. The AWS IAM integration with the Resilient platform allows you to query and update users or access keys for an AWS account.

You can execute the following types of queries:

- Get a list of users and associated items (login profile, access keys, groups, policies).
- Get a list of access keys.
- List objects associated with a user:
    - Access keys.
    - Groups.
    - Policies.
    - SSH public keys.
    - Service-specific credentials.
    - Signing certificates.
    - Active MFA devices (Virtual devices flagged).

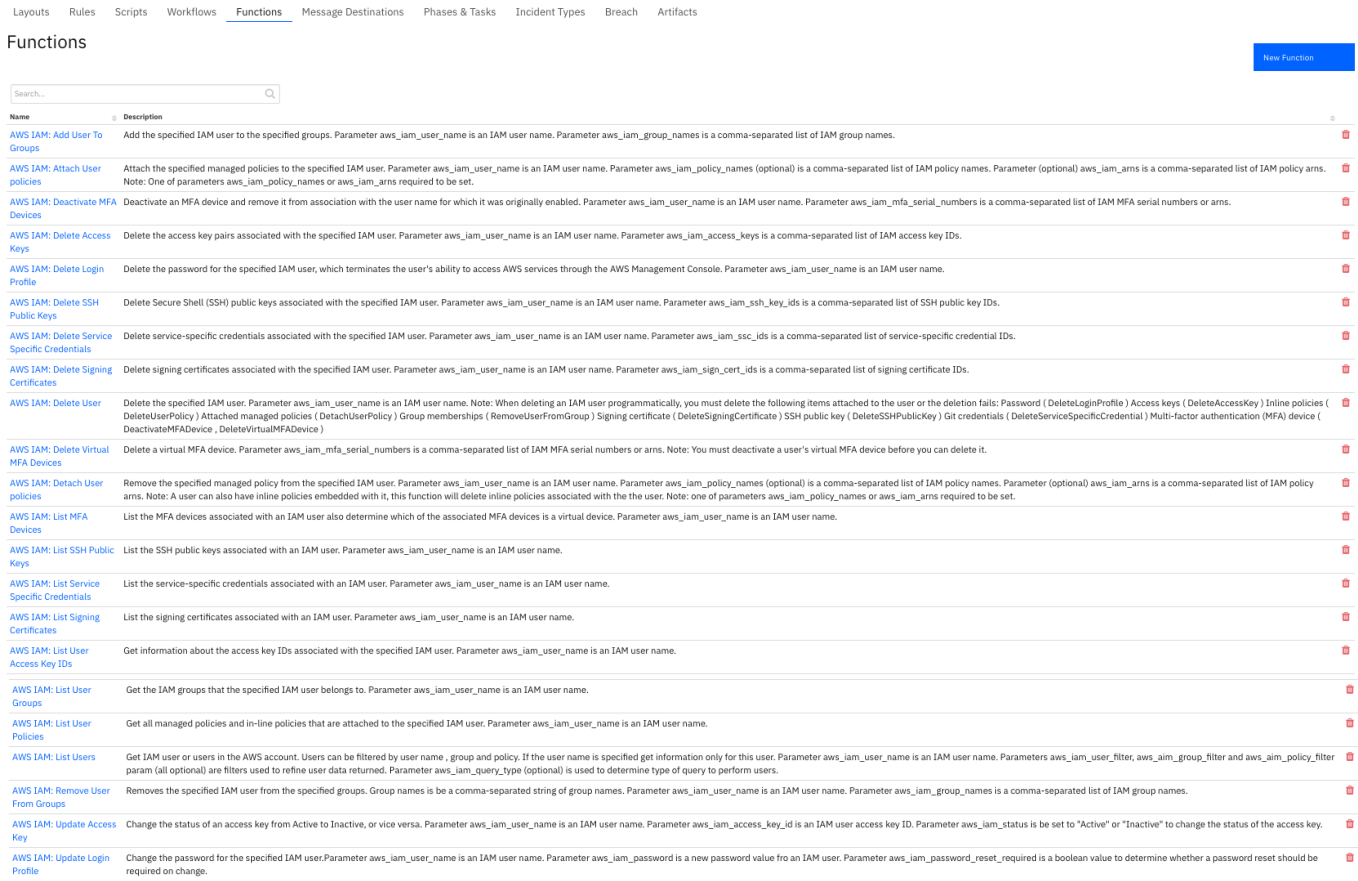
You can also use the integration to make the following changes to an AWS IAM environment:

- Delete a user and delete or remove items associated with the user.
- Attach a user policy.
- Detach all policies for a user.
- Add a user to a group.
- Remove a user from all groups.
- Change a user profile password.
- Delete an access key.
- Delete all access keys for a user.
- Delete the login profile for a user.

- Delete all SSH Public Keys for a user.
- Delete all service-specific credentials for a user.
- Delete all signing certificates for a user.
- De-activate all active MFA devices for a user.
- Delete all active MFA virtual devices for a user.

The integration contains the following functions:

Customization Settings



Layouts Rules Scripts Workflows Functions Message Destinations Phases & Tasks Incident Types Breach Artifacts

Functions

New Function

Search...

Name	Description
AWS IAM: Add User To Groups	Add the specified IAM user to the specified groups. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_group_names is a comma-separated list of IAM group names.
AWS IAM: Attach User policies	Attach the specified managed policies to the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_policy_names (optional) is a comma-separated list of IAM policy names. Parameter (optional) aws_iam_arns is a comma-separated list of IAM policy arns. Note: One of parameters aws_iam_policy_names or aws_iam_arns required to be set.
AWS IAM: Deactivate MFA Devices	Deactivate an MFA device and remove it from association with the user name for which it was originally enabled. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_mfa_serial_numbers is a comma-separated list of IAM MFA serial numbers or arns.
AWS IAM: Delete Access Keys	Delete the access key pairs associated with the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_access_keys is a comma-separated list of IAM access key IDs.
AWS IAM: Delete Login Profile	Delete the password for the specified IAM user, which terminates the user's ability to access AWS services through the AWS Management Console. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: Delete SSH Public Keys	Delete Secure Shell (SSH) public keys associated with the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_ssh_key_ids is a comma-separated list of SSH public key IDs.
AWS IAM: Delete Service Specific Credentials	Delete service-specific credentials associated with the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_ssc_ids is a comma-separated list of service-specific credential IDs.
AWS IAM: Delete Signing Certificates	Delete signing certificates associated with the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_sign_cert_ids is a comma-separated list of signing certificate IDs.
AWS IAM: Delete User	Delete the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Note: When deleting an IAM user programmatically, you must delete the following items attached to the user or the deletion fails: Password (DeleteLoginProfile) Access keys (DeleteAccessKey) Inline policies (DeleteUserPolicy) Attached managed policies (DetachUserPolicy) Group memberships (RemoveUserFromGroup) Signing certificate (DeleteSigningCertificate) SSH public key (DeleteSSHPublicKey) Git credentials (DeleteServiceSpecificCredential) Multi-factor authentication (MFA) device (DeactivateMFADevice , DeleteVirtualMFADevice)
AWS IAM: Delete Virtual MFA Devices	Delete a virtual MFA device. Parameter aws_iam_mfa_serial_numbers is a comma-separated list of IAM MFA serial numbers or arns. Note: You must deactivate a user's virtual MFA device before you can delete it.
AWS IAM: Detach User policies	Remove the specified managed policy from the specified IAM user. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_policy_names (optional) is a comma-separated list of IAM policy names. Parameter (optional) aws_iam_arns is a comma-separated list of IAM policy arns. Note: A user can also have inline policies embedded with it, this function will delete inline policies associated with the the user. Note: one of parameters aws_iam_policy_names or aws_iam_arns required to be set.
AWS IAM: List MFA Devices	List the MFA devices associated with an IAM user also determine which of the associated MFA devices is a virtual device. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List SSH Public Keys	List the SSH public keys associated with an IAM user. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List Service Specific Credentials	List the service-specific credentials associated with an IAM user. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List Signing Certificates	List the signing certificates associated with an IAM user. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List User Access Key IDs	Get information about the access key IDs associated with the specified IAM user. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List User Groups	Get the IAM groups that the specified IAM user belongs to. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List User Policies	Get all managed policies and in-line policies that are attached to the specified IAM user. Parameter aws_iam_user_name is an IAM user name.
AWS IAM: List Users	Get IAM user or users in the AWS account. Users can be filtered by user name , group and policy. If the user name is specified get information only for this user. Parameter aws_iam_user_name is an IAM user name. Parameters aws_iam_user_filter, aws_aim_group_filter and aws_aim_policy_filter param (all optional) are filters used to refine user data returned. Parameter aws_iam_query_type (optional) is used to determine type of query to perform users.
AWS IAM: Remove User From Groups	Removes the specified IAM user from the specified groups. Group names is be a comma-separated string of group names. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_group_names is a comma-separated list of IAM group names.
AWS IAM: Update Access Key	Change the status of an access key from Active to Inactive, or vice versa. Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_access_key_id is an IAM user access key ID. Parameter aws_iam_status is be set to "Active" or "Inactive" to change the status of the access key.
AWS IAM: Update Login Profile	Change the password for the specified IAM user.Parameter aws_iam_user_name is an IAM user name. Parameter aws_iam_password is a new password value fro an IAM user. Parameter aws_iam_password_reset_required is a boolean value to determine whether a password reset should be required on change.

Requirements

- Resilient platform >= `v35.0.5468`
- An Integration Server running:
  - `resilient_circuits>=35.0.0`
  - `resilient_lib>=35.0.0`
  - The minimum set of Resilient API permissions for this integration if using an API key account:
    - Edit Org Data
    - Incidents.Edit.Fields
    - Functions.Read
    - Functions.Edit
    - Layouts.Read
    - Other.ReadIncidentsActionInvocations
    - Scripts.Create
    - Scripts.Edit
    - Workflows.Create
    - Workflow.Edit

    - To set up an Integration Server see: ibm.biz/res-int-server-guide

- An AWS IAM user dedicated for this integration with the following configuration:
    - User account is not the AWS IAM root account user.
    - User is added to an "Administrators" group to which is attached the `AdministratorAccess` policy.
    - An access key created for the user. The user access key ID and secret access key are used by the integration.

Installation - App Host

All the components for running this integration in a container already exist when using the App Host app.

- Download the app `app-fn_aws_iam-x.x.x.zip`.
- Navigate to Administrative Settings and then the Apps tab.
- Click the Install button and select the downloaded file: app-fn_aws_iam-x.x.x.zip.
- Go to the Configuration tab and edit the app.config file, editing the access key values for Amazon AWS IAM and add a proxy setting if required.

```
[fn_aws_iam]
aws_iam_access_key_id=<AWS_IAM_ACCESS_KEY_ID>
aws_iam_secret_access_key=<AWS_IAM_SECRET_ACCESS_KEY>
# Optional settings for access to AWS IAM via a proxy.
#http_proxy=http://proxy:80
#https_proxy=http://proxy:80
```

- [Optional]: Test the configuration.
- Deploy the app.

Installation - Integration server

- Download the app `app-fn_aws_iam-x.x.x.zip`.
- Copy the `.zip` to your Integration Server and SSH into it.
- **Unzip** the app:

  ```
  $ unzip app-fn_aws_iam-x.x.x.zip
  ```

- **Install** the package:

  ```
  $ pip install fn_aws_iam-x.x.x.tar.gz
  ```

- Import the **configurations** into your app.config file:

  ```
  $ resilient-circuits config -u
  ```

- Import the fn_aws_iam **customizations** into the Resilient platform:

  ```
  $ resilient-circuits customize -y -l fn-aws-iam
  ```

- Open the config file, scroll to the bottom and edit your fn_aws_iam configurations:

```
$ nano ~/.resilient/app.config
```

Config	Required	Example	Description
aws_iam_access_key_id	Yes	ABCD1EFGHI2JK3L4MNOP	*AWS access key id of user with programmatic (API) access to AWS IAM services for an AWS account. Note: User must have sufficent permissions to be able to manage IAM resources for the AWS account.*
aws_iam_secret_access_key	Yes	aBcdeFGH/iJkl1MNo2P3Q4rs5tuV6wXYZAbc+Def	*AWS secret access key used for programmatic (API) access to AWS services.*
http_proxy	No	http://proxy:80	*Optional setting for an http proxy if required.*
https_proxy	No	http://proxy:80	*Optional setting for an http proxy if required.*

- **Save** and **Close** the app.config file.
- [Optional]: Run selftest to test the Integration you configured:

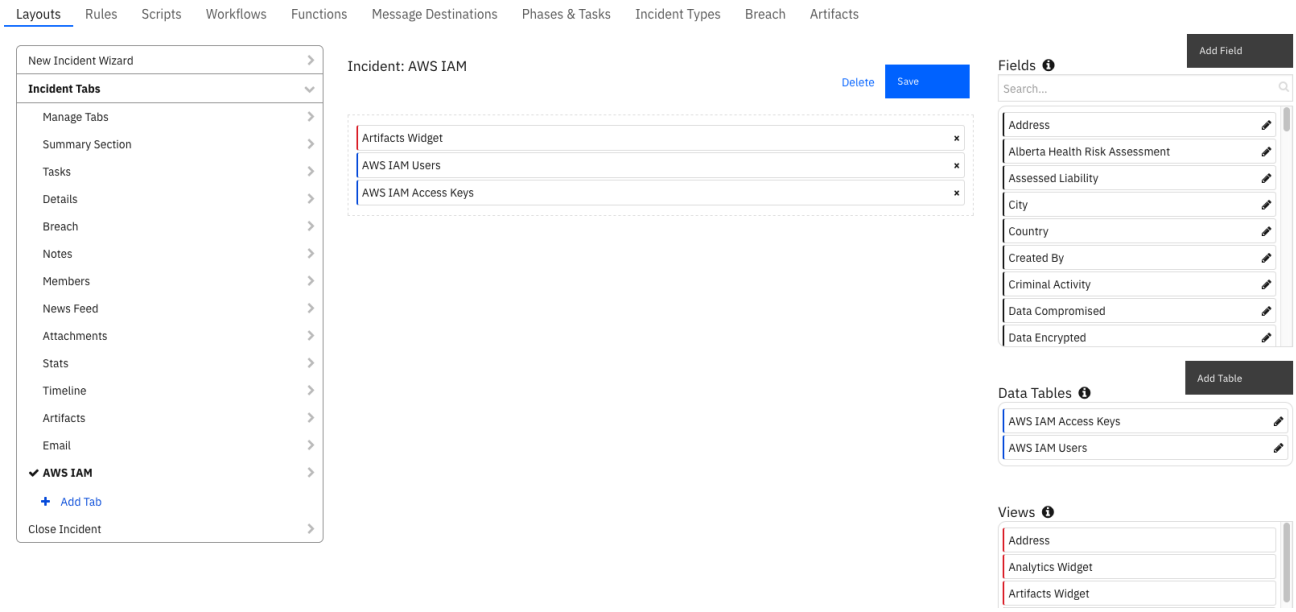
```
$ resilient-circuits selftest -l fn-aws-iam
```

- **Run** resilient-circuits or restart the Service on Windows/Linux:

```
$ resilient-circuits run
```

Custom Layouts

- To use the functions, the Resilient playbook designer should create a new Incident tab containing the data tables. Drag the AWS IAM data tables on to the layout and click Save as shown in the screenshot below:



Uninstall

- SSH into your Integration Server.
- **Uninstall** the package:

```
$ pip uninstall fn-aws-iam
```

- Open the config file, scroll to the [fn_aws_iam] section and remove the section or prefix # to comment out the section.
- **Save** and **Close** the app.config file.

Troubleshooting

There are several ways to verify the successful operation of a function.

Resilient Action Status

- When viewing an incident, use the Actions menu to view **Action Status**.
- By default, pending and errors are displayed.
- Modify the filter for actions to also show Completed actions.
- Clicking on an action displays additional information on the progress made or what error occurred.

Resilient Scripting Log

- A separate log file is available to review scripting errors.
- This is useful when issues occur in the pre-processing or post-processing scripts.

- The default location for this log file is: `/var/log/resilient-scripting/resilient-scripting.log`.

Resilient Logs

- By default, Resilient logs are retained at `/usr/share/co3/logs`.
- The `client.log` may contain additional information regarding the execution of functions.

Resilient-Circuits

- The log is controlled in the `.resilient/app.config` file under the section [resilient] and the property `logdir`.
- The default file name is `app.log`.
- Each function will create progress information.
- Failures will show up as errors and may contain python trace statements.

Support

Name	Version	Author	Support URL
fn_aws_iam	1.0.0	IBM Resilient Support	https://ibm.com/mysupport