AWS Service Management Connector for ServiceNow v4.5.0

Syncing Updated Keys Programmatically in ServiceNow

To help customers provision and manage secure, compliant AWS resources into their ServiceNow portal, AWS created the AWS Service Management Connector for ServiceNow (formerly the AWS Service Catalog Connector).

The AWS Service Management Connector for ServiceNow allows ServiceNow end-users to provision, manage and operate AWS resources natively via ServiceNow. ServiceNow administrators can provide pre-approved, secured and governed AWS resources to end-users via AWS Service Catalog, create and manage Security Findings via AWS Security Hub, execute automation playbooks via AWS Systems Manager Automation and track resources in a configuration item view in ServiceNow CMDB powered by AWS Config seamlessly on the ServiceNow with the AWS Service Management Connector.

This document will focus on instructions for programmatically synchronizing updated keys used for the SyncUser and EndUser AWS IAM users. For full documentation details, please refer to this link: https://docs.aws.amazon.com/servicecatalog/latest/adminguide/integrationsservicenow.Html

Syncing Updated Keys Programmatically in ServiceNow

AWS Service Management Connector for ServiceNow allows key rotation by any automation/integration, via a new REST endpoint. Requests can be sent to rotate keys for one or more AWS accounts registered within the connector, for either the sync and/or end user.
Authentication

The endpoint is protected by standard ServiceNow authentication. The user performing the request *must have* the role `x_126749_aws_sc_account_admin` to be successful. Failure to meet these requirements will result in an error.

To Synchronize Updated AWS Keys via REST API

1. Log into ServiceNow as system administrator (Pen.Tester).
2. Go the REST endpoint. The endpoint is available by doing a `POST` request to `https://<servicenow-instance>.servicenow.com/api/x_126749_aws_sc/account/rotation`
3. Query parameters
    a. `validate`: (Optional) whether or not validate the credentials, prior to rotating them. Default is `true`. Allowed values: `true` or `false`

Body Example:

```
[
{
   "name": "TestAWS",                    ----------------------→ Replace with your Account Name
   "accountNumber": "123456789",         ----------------------→Replace with your AWS  Account
    "syncUser": {
       "accessKey": "AKIAIOSSYNCUSEREXAMPLE",
       "secretAccessKey": "wJalrXUtnFEMI/SYNCUSER/bPxRfiCYEXAMPLEKEY"
     },
    "endUser": {
       "accessKey": "AKIAIOSENDUSEREXAMPLE",
        "secretAccessKey": "wJalrXUtnFEMI/ENDUSER/bPxRfiCYEXAMPLEKEY"
     }
}
]
```

The body is expected to be an `Array<KeyRotationRequest>`. The body is validated on all requests and anything not matching the expected body will be rejected.

KeyRotationRequest Object

Example:

```
[
{
        "name": "AWS account display name in ServiceNow",
        "accountNumber": "123456789",
        "syncUser": {
                "accessKey": "AKIAIOSSYNCUSEREXAMPLE",
                "secretAccessKey": "wJalrXUtnFEMI/SYNCUSER/bPxRfiCYEXAMPLEKEY"
        },
        "endUser": {
                "accessKey": "AKIAIOSENDUSEREXAMPLE",
                "secretAccessKey": "wJalrXUtnFEMI/ENDUSER/bPxRfiCYEXAMPLEKEY"
        }
}
]
```

`name` and `accountNumber` are used to identify the account to rotate keys for and are required attributes. Each `KeyRotationRequest` can update the keys of either the sync and/or enduser.

- *name*: (Required) String - The ServiceNow display name of the AWS account to rotate Keys
- *accountNumber*: (Required) String - The AWS account number to rotate keys
- *syncUser*: (Optional) Keys object - The new keys for the sync-user, to set to the AWS account
- *endUser*: (Optional) Keys object - The new keys for the end-user, to set to the AWS account

To validate connectivity to AWS account with new keys

1. In the AWS Service Management scoped app, choose **Accounts**.
2. Select AWS account (i.e., Connector_Demo) and choose **Validate Account**.
3. A successful connection result in the message, “Successfully validating AWS account in each referenced Region.”

If the AWS IAM access key or secret access key are incorrect, you will receive an error message.

To validate an end user without permissions to the role `x_126749_aws_sc_account_admin’ cannot perform this action

1. Log into ServiceNow instance as the end user provided in the credential’s spreadsheet (i.e., Abel Tuter).
2. Go the REST endpoint. The endpoint is available by doing a `POST` request to `https://<servicenow-instance>.servicenow.com/api/x_126749_aws_sc/account/rotation`
4. Query parameters and repeat the steps above.

Python Code Snippet :

```
#Need to install requests package for python
#easy_install requests
import requests

# Set the request parameters
url = 'https://ven02856.service-
now.com/api/x_126749_aws_sc/account/rotation?validate=false'

# Eg. User name="admin", Password="admin" for this code sample.
user = 'admin'
pwd = 'admin'

# Set proper headers
headers = {"Content-Type":"application/json","Accept":"application/json"}

# Do the HTTP request
response = requests.post(url, auth=(user, pwd), headers=headers ,data="[
{
    \"name\": \"TestAWS\",
    \"accountNumber\": \"1234567890\",
     \"syncUser\": {
          \"accessKey\": \"AKIAIOSSYNCUSEREXAMPLE\",
          \"secretAccessKey\": \"wJalrXUtnFEMI/SYNCUSER/bPxRfiCYEXAMPLEKEY\"
       },
     \"endUser\": {
          \"accessKey\": \"AKIAIOSENDUSEREXAMPLE\",
           \"secretAccessKey\": \"wJalrXUtnFEMI/ENDUSER/bPxRfiCYEXAMPLEKEY\"
       }
}
```

```
]")

# Check for HTTP codes other than 200
if response.status_code != 200:
    print('Status:', response.status_code, 'Headers:', response.headers,
'Error Response:',response.json())
    exit()

# Decode the JSON response into a dictionary and use the data
data = response.json()
print(data)
```