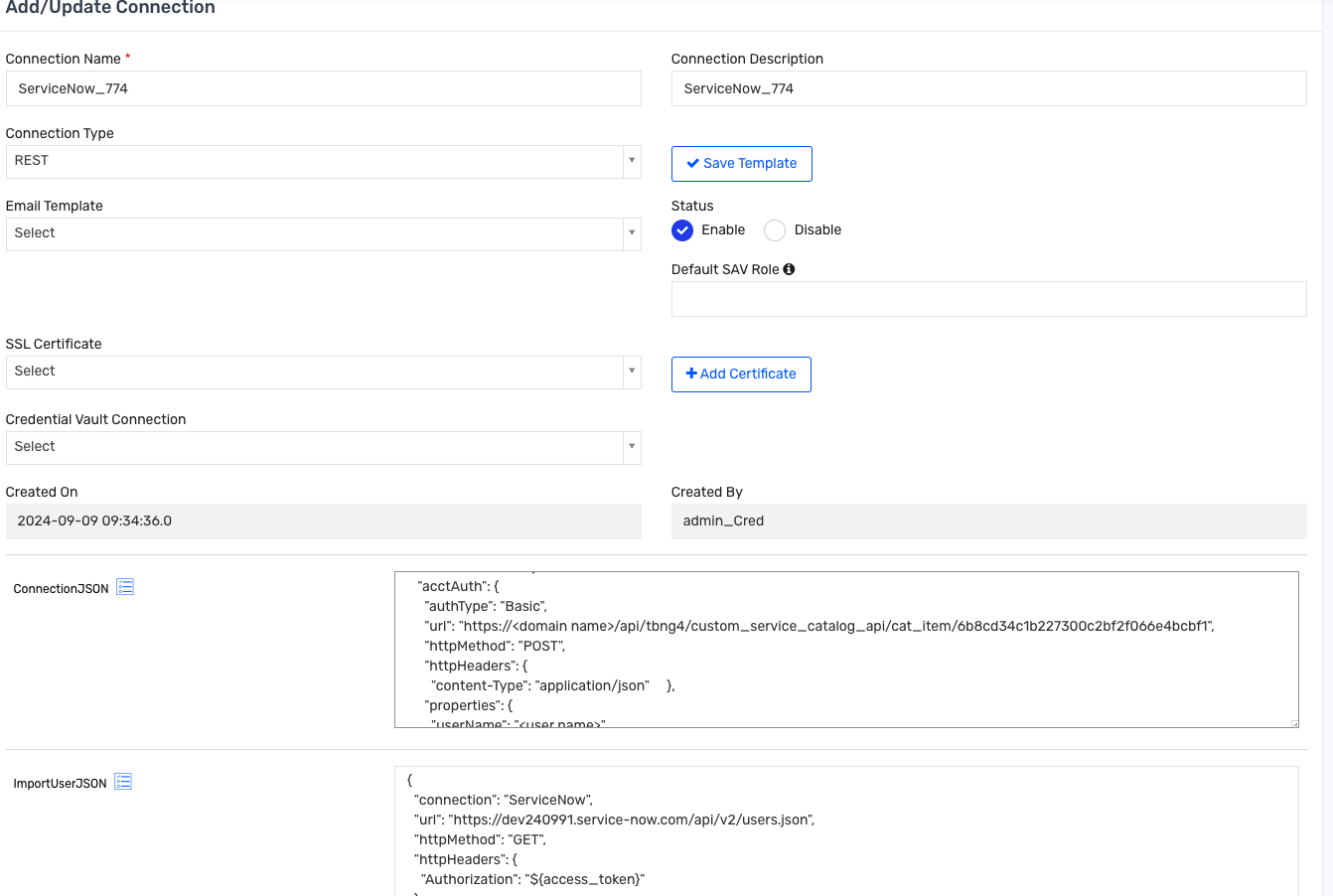
Service now as a Managed Application

Introduction:  
Saviynt EIC offers capability to set up ServiceNow as a managed application within Saviynt EIC for both reconciliation and provisioning (including de-provisioning), follow this approach:  
  
  
How it works:  
  
Saviynt offers a connector module which can be configured to make a connection with the application.  
  
  
Prerequisites:   
  
Network connectivity should be established before we can begin operations like recon and provisioning. The network needs to be whitelisted, and firewall should allow the network to flow from Saviynt to EIC and vice versa. The Credentials for the Rest API are valid.  
  
Integration Steps:  
  
Create a Rest connection. Let’s look at the integration steps.



1. Create a connection of Rest and it will populate the template accordingly.

2) Provide a Connection Json:  
  
Sample Json:  
{

"authentications": {

"acctAuth": {

"authType": "Basic",

"url": "https://<domain name>/api/tbng4/custom\_service\_catalog\_api/cat\_item/6b8cd34c1b227300c2bf2f066e4bcbf1",

"httpMethod": "POST",

"httpHeaders": {

"content-Type": "application/json" },

"properties": {

"userName": "<user name>",

"password": "<password>" },

"httpContentType": "application/json",

"expiryError": "ExpiredAuthenticationToken",

"retryFailureStatusCode": [401],

"authError": [

"Internal server error" ],

"timeOutError": "Read timed out",

"errorPath": "message",

"maxRefreshTryCount": 5,

"tokenResponsePath": "access\_token",

"tokenType": "Basic",

"accessToken": "Basic abcd" }

}

}  
  
  
Let’s look at the Json in detail

**Parameter Details**

* **authType**: Specifies Basic Authentication, which uses a userName and password for authentication.
* **url**: The endpoint for ServiceNow’s API to which the REST request is sent. Replace <domain name> with your ServiceNow instance's domain name.
* **httpMethod**: The HTTP method to be used, here set to POST.
* **httpHeaders**: Defines the HTTP headers; in this case, it specifies "Content-Type": "application/json", ensuring the request body is JSON-formatted.
* **properties**:
  + **userName** and **password**: Credentials required for Basic Authentication with ServiceNow.
* **httpContentType**: Specifies that the content type for requests is application/json.
* **Error Handling**:
  + **expiryError**: Defines the specific error for an expired authentication token.
  + **retryFailureStatusCode**: Contains status codes that trigger retry attempts (e.g., 401 for unauthorized).
  + **authError**: Lists potential authentication errors (like “Internal server error”).
  + **timeOutError**: Specifies the error message if the connection times out.
* **errorPath**: Points to the JSON path for the error message within the response (e.g., "message").
* **maxRefreshTryCount**: Limits the number of retry attempts (set to 5).
* **tokenResponsePath**: Specifies the path to the token in the response JSON, though this is typically more relevant for OAuth2 configurations. For Basic Authentication, it can remain unused.
* **tokenType** and **accessToken**: Token type is set to Basic, with an example access token provided.

A close up of text

Description automatically generated

3)  
Provide Import AccountEnTJson:

{

"accountParams": {

"connection": "userAuth",

"processingType": "SequentialAndIterative",

"call": {

"call1": {

"callOrder": 0,

"stageNumber": 0,

"http": {

"url": "https://domain.service-now.com/api/now/table/sys\_user?sysparm\_fields=user\_name,sys\_id,name,email,active,sys\_created\_on,sys\_updated\_on,last\_login\_time&sysparm\_offset=0&sysparm\_limit=20000",

"httpHeaders": {

"Authorization": "${access\_token}"

},

"httpContentType": "application/x-www-form-urlencoded",

"httpMethod": "GET"

},

"statusConfig": {

"active": "true",

"inactive": "false"

},

"listField": "result",

"keyField": "accountID",

"colsToPropsMap": {

"accountID": "sys\_id~#~char",

"customproperty3": "name~#~char",

"CUSTOMPROPERTY7": "email~#~char",

"DISPLAYNAME": "name~#~char",

"CUSTOMPROPERTY1": "user\_name~#~char",

"name": "user\_name~#~char",

"CREATED\_ON": "sys\_created\_on~#~char",

"UPDATEDATE": "sys\_updated\_on~#~char",

"LASTLOGONDATE": "last\_login\_time~#~char",

"status": "active~#~char"

},

"pagination": {

"nextUrl": {

"nextUrlPath": "${headers?.Link?.split(';')?.size()==5?headers?.Link?.split(';')[2]?.replace('rel=\"prev\",<', '')?.replace('>','')?.trim():headers?.Link?.split(';')?.size()==4&&headers?.Link?.split(';')[2]?.contains('next')?headers?.Link?.split(';')[1]?.replace('rel=\"first\",<', '')?.replace('>','')?.trim():null}"

}

}

}

}

},

"entitlementParams": {

"processingType": "SequentialAndIterative",

"entTypes": {

"Group": {

"entTypeOrder": 0,

"entTypeLabels": {},

"call": {

"call1": {

"connection": "userAuth",

"callOrder": 0,

"stageNumber": 0,

"http": {

"httpHeaders": {

"Authorization": "${access\_token}"

},

"url": "https://domain.service-now.com/api/now/table/sys\_user\_group?sysparm\_fields=name%2Csys\_id%2Cmanager",

"httpContentType": "application/json",

"httpMethod": "GET"

},

"listField": "result",

"keyField": "entitlementID",

"colsToPropsMap": {

"entitlementID": "sys\_id~#~char",

"customproperty4": "manager.value~#~char",

"entitlement\_value": "name~#~char"

}

},

"call2": {

"connection": "userAuth",

"callOrder": 1,

"stageNumber": 0,

"http": {

"httpHeaders": {

"Authorization": "${access\_token}"

},

"url": "https://domain.service-now.com/api/now/table/sys\_user\_group?sysparm\_display\_value=true",

"httpContentType": "application/json",

"httpMethod": "GET"

},

"statusConfig": {

"active": "true",

"inactive": "false"

},

"listField": "result",

"keyField": "entitlementID",

"colsToPropsMap": {

"entitlementID": "sys\_id~#~char",

"description": "description~#~char",

"entitlement\_value": "name~#~char",

"status": "active~#~char",

"customproperty1": "type~#~char",

"customproperty2": "u\_company\_parent.display\_value~#~char",

"customproperty30": "#CONST#OTHER - Not an IT user, this selection will show all groups with business access~#~char",

"customproperty3": "parent.display\_value~#~char"

}

},

"call3": {

"connection": "userAuth",

"callOrder": 2,

"stageNumber": 0,

"http": {

"httpHeaders": {

"Authorization": "${access\_token}"

},

"url": "https://domain.service-now.com/api/now/table/sys\_group\_has\_role?sysparm\_query=role.name%3DItil%5Egroup.active%3Dtrue",

"httpContentType": "application/json",

"httpMethod": "GET"

},

"listField": "result",

"keyField": "entitlementID",

"colsToPropsMap": {

"entitlementID": "group.value~#~char",

"customproperty30": "#CONST#ANALYST - Work and resolve incidents and requests. Create changes.~#~char",

"confidentiality": "#CONST#5~#~char"

}

},

"call4": {

"connection": "userAuth",

"callOrder": 3,

"stageNumber": 0,

"http": {

"httpHeaders": {

"Authorization": "${access\_token}"

},

"url": "https://domain.service-now.com/api/now/table/sys\_group\_has\_role?sysparm\_query=role.name%3Dbusiness\_stakeholder%5Egroup.active%3Dtrue",

"httpContentType": "application/json",

"httpMethod": "GET"

},

"listField": "result",

"keyField": "entitlementID",

"colsToPropsMap": {

"entitlementID": "group.value~#~char",

"customproperty30": "#CONST#MANAGER - Create and use dashboards and reporting. View ticket details and add comments.~#~char"

}

}

},

"entMappings": {

"Group": {

"listPath": "parent",

"idPath": "value",

"idColumn": "entitlementID",

"mappingTypes": [

"ENTMAP"

]

}

}

},

"Roles": {

"entTypeOrder": 1,

"entTypeLabels": {},

"call": {

"call1": {

"connection": "userAuth",

"callOrder": 0,

"stageNumber": 0,

"http": {

"httpHeaders": {

"Authorization": "${access\_token}"

},

"url": "https://domain.service-now.com/api/now/table/sys\_user\_role?sysparm\_display\_value=false&sysparm\_query=nameNOT%20LIKEPNOW",

"httpContentType": "application/json",

"httpMethod": "GET"

},

"listField": "result",

"keyField": "entitlementID",

"colsToPropsMap": {

"description": "description~#~char",

"customproperty3": "elevated\_privilege~#~char",

"entitlementID": "sys\_id~#~char",

"entitlement\_Value": "name~#~char",

"displayname": "sys\_name~#~char"

}

}

}

}

}

},

"acctEntParams": {

"entTypes": {

"Group": {

"call": {

"call1": {

"processingType": "http",

"connection": "userAuth",

"http": {

"url": "https://domain.service-now.com/api/now/v1/table/sys\_user\_grmember?sysparm\_display\_value=false&sysparm\_limit=10000",

"httpContentType": "application/json",

"httpMethod": "GET",

"httpHeaders": {

"Authorization": "${access\_token}"

}

},

"listField": "result",

"acctKeyField": "accountID",

"entKeyField": "entitlementID",

"acctIdPath": "user.value",

"entIdPath": "group.value",

"pagination": {

"nextUrl": {

"nextUrlPath": "${headers?.Link?.split(';')?.size()==5?headers?.Link?.split(';')[2]?.replace('rel=\"prev\",<', '')?.replace('>','')?.trim():headers?.Link?.split(';')?.size()==4 && headers?.Link?.split(';')[2].contains('rel=\"next\",<')?headers?.Link?.split(';')[1]?.replace('rel=\"first\",<', '')?.replace('>','')?.trim():null}"

}

}

}

}

},

"Roles": {

"call": {

"call1": {

"processingType": "http",

"connection": "userAuth",

"http": {

"url": "https://domain.service-now.com/api/now/table/sys\_user\_has\_role?sysparm\_display\_value=false",

"httpContentType": "application/json",

"httpMethod": "GET",

"httpHeaders": {

"Authorization": "${access\_token}"

}

},

"listField": "result",

"acctKeyField": "accountID",

"entKeyField": "entitlementID",

"acctIdPath": "user.value",

"entIdPath": "role.value"

}

}

}

}

}

}  
Leys look at the json configuration in details  
  
 **Account Parameters (accountParams)**

* **connection**: Specifies "userAuth" as the connection, which presumably includes authentication details.
* **processingType**: Set to "SequentialAndIterative", meaning requests will process in sequence.
* **call1**:
  + Retrieves **user accounts** from ServiceNow (sys\_user table).
  + **URL** includes parameters to fetch fields like user\_name, sys\_id, email, status, etc.
  + **colsToPropsMap**: Maps ServiceNow response fields to EIC attributes.
  + **pagination**: Uses nextUrlPath to handle paginated responses from ServiceNow.

**Entitlement Parameters (entitlementParams)**

* Contains **entitlements** for **Groups** and **Roles** from ServiceNow.

**a. Groups**

* **call1** to **call4**:
  + Fetches group data (including name, sys\_id, manager).
  + Some calls (e.g., call3 and call4) use filters like role.name=Itil and role.name=business\_stakeholder.
  + Each call has specific mappings in **colsToPropsMap** to align data with EIC’s attribute structure.
* **entMappings**: Defines entitlement mappings, specifying which fields in ServiceNow (parent, value) map to entitlement fields in EIC.

**b. Roles**

* **call1**:
  + Retrieves **roles** using the sys\_user\_role endpoint, filtering out specific names (e.g., nameNOT LIKEPNOW).
  + Maps various fields, such as sys\_id to entitlementID and name to entitlement\_Value.

**Account-Entitlement Parameters (acctEntParams)**

* Defines **account-entitlement** relationships for both Groups and Roles.

**a. Groups**

* **call1**:
  + Uses sys\_user\_grmember endpoint to map **user memberships in groups**.
  + Retrieves mappings for accountID (user) and entitlementID (group).
  + **pagination**: Uses the same logic as in accountParams for handling paginated results.

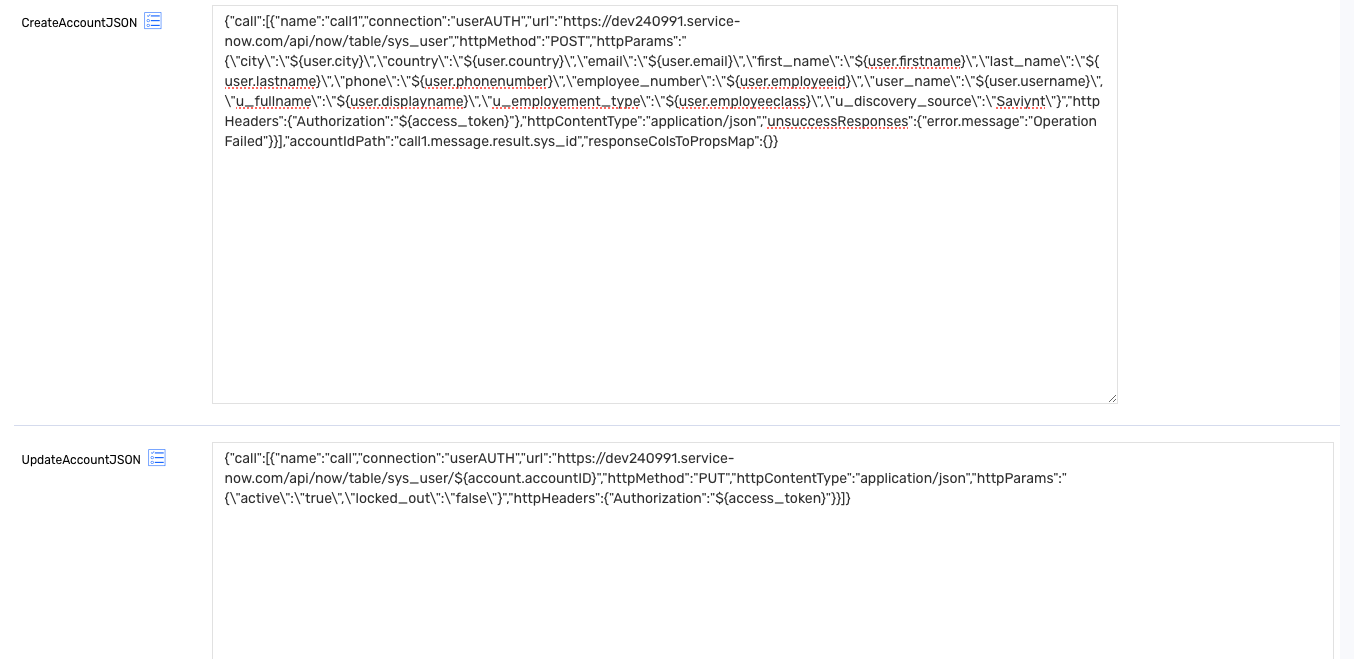
**b. Roles**

* **call1**:
  + Uses sys\_user\_has\_role to map **user role assignments**.
  + Aligns accountID and entitlementID with ServiceNow’s user and role values.

**Key Points in JSON Configuration**

* **Error Handling**: In each section, error handling is set up to manage different response issues and retry if necessary.
* **Headers**: The "Authorization": "${access\_token}" header is included to pass an access token in requests.
* **Pagination**: Implements a custom nextUrl logic to handle paginated data from ServiceNow API responses.

Provisioning and Deprovisioning details



4) Provide the Create account Json  
  
Sample Create Account Json:

{"call":[{"name":"call1","connection":"userAUTH","url":"https://dev240991.service-now.com/api/now/table/sys\_user","httpMethod":"POST","httpParams":"{\"city\":\"${user.city}\",\"country\":\"${user.country}\",\"email\":\"${user.email}\",\"first\_name\":\"${user.firstname}\",\"last\_name\":\"${user.lastname}\",\"phone\":\"${user.phonenumber}\",\"employee\_number\":\"${user.employeeid}\",\"user\_name\":\"${user.username}\",\"u\_fullname\":\"${user.displayname}\",\"u\_employement\_type\":\"${user.employeeclass}\",\"u\_discovery\_source\":\"Saviynt\"}","httpHeaders":{"Authorization":"${access\_token}"},"httpContentType":"application/json","unsuccessResponses":{"error.message":"Operation Failed"}}],"accountIdPath":"call1.message.result.sys\_id","responseColsToPropsMap":{}}  
  
Lets look the json in detail

**Main Structure (call)**

* **call1**:
  + **name**: Identifies this call as "call1".
  + **connection**: Uses "userAUTH" to establish the required authentication.
  + **url**: Points to the sys\_user table endpoint on ServiceNow, which is used to create user accounts.
  + **httpMethod**: Set to "POST" to create new records in ServiceNow.
  + **httpParams**: A JSON object containing user attributes as parameters.
    - Each field (e.g., "city", "country", "email") is mapped dynamically using ${user.<attribute>} placeholders, pulling data from the user object.
    - **Static Field**: "u\_discovery\_source" is set to "Saviynt" to mark the data origin.
  + **httpHeaders**: Sets the Authorization header with "${access\_token}", which provides an access token for authenticated requests.
  + **httpContentType**: Defined as "application/json", indicating JSON formatting for the request body.
  + **unsuccessResponses**: If the API returns an error, such as in the error.message field, this field logs it as "Operation Failed".

**Account ID Path (accountIdPath)**

* Specifies where in the response the unique sys\_id of the newly created user can be found, which is "call1.message.result.sys\_id".

**Response Columns to Properties Map (responseColsToPropsMap)**

* This field is left empty ({}), implying no specific mapping is required from response properties to EIC attributes.

A white background with black lines

Description automatically generated

5) Provide the Add and Remove Access Json  
  
  
Sample Add Access Json:  
  
{

"call": [

{

"name": "Group",

"connection": "userAuth",

"url":"https://domain.service-now.com/api/now/v1/table/sys\_user\_grmember",

"httpMethod": "POST",

"httpParams": "{\"group\":\"${entitlementValue.entitlementID}\",\"user\":\"${account.accountID}\"}",

"httpHeaders":

{

"Authorization": "${access\_token}"

},

"httpContentType": "application/json"

}

]

}  
  
Lets look at the json in detail  
  
**Main Structure (call)**

* **Group Call**:
  + **name**: Identifies this call as "Group" which is the entitle,emt type.
  + **connection**: Uses "userAuth" to authenticate the request.
  + **url**: The endpoint URL points to the sys\_user\_grmember table in ServiceNow, which manages user-group memberships.
  + **httpMethod**: Set to "POST" to create a new membership record (i.e., add a user to a group).
  + **httpParams**: Defines the body of the request in JSON format:
    - "group": Populates with the ${entitlementValue.entitlementID}, which corresponds to the ID of the group.
    - "user": Populates with the ${account.accountID}, which represents the user's unique ID in ServiceNow.
  + **httpHeaders**: Sets the Authorization header using "${access\_token}" for authentication.
  + **httpContentType**: Specifies "application/json", indicating the request body is formatted in JSON.

Sample Remove Access Json:  
  
{

"call": [{

"name": "Group",

"connection": "userAuth",

"url": "https://khcprod.service-now.com/api/now/v1/table/sys\_user\_grmember?user=${account.accountID}",

"httpMethod": "GET",

"httpHeaders": {

"Authorization": "${access\_token}"

},

"httpContentType": "application/json",

"successResponses": {

"statusCode": [200]

},

"unsuccessResponses": {

"statusCode": [403]

}

}, {

"name": "Group",

"connection": "userAuth",

"url": "https://domain.service-now.com/api/now/v1/table/sys\_user\_grmember/${for (Map map : response.Group1.message.result){if (map.group.value.toString().equals(entitlementValue.entitlementID)){return map.sys\_id;}}}",

"httpMethod": "DELETE",

"httpHeaders": {

"Authorization": "${access\_token}"

},

"httpContentType": "application/json",

"successResponses": {

"statusCode": [204]

},

"unsuccessResponses": {

"statusCode": [403]

}

}]

}

Lets look at the Json in detail  
  
**Retrieving the User-Group Membership Record**:

* The **first call** performs a **GET request** to check if a user is currently a member of the specified group by querying the sys\_user\_grmember table with the user=${account.accountID} filter.
* If successful (statusCode 200), it retrieves all group memberships associated with the specified user.

**Deleting the User-Group Membership Record**:

* The **second call** performs a **DELETE request** to remove the user from the target group.
* This call dynamically resolves the URL endpoint for the specific membership record (sys\_user\_grmember) to delete, using a Groovy script embedded in the URL:
  + The script iterates through the response of the first call (response.Group1.message.result), locating the sys\_id of the membership record where the group matches entitlementID.
  + Once located, it returns the sys\_id to complete the URL for deletion.
* If successful, the response status code is expected to be 204 (indicating successful deletion without returning content).

**Troubleshooting:  
  
In Most of the service now connection, the token expires and it does not refresh .  
  
  
Resolution:  
  
Make sure that the error code that is coming in token expiry is defined properly in connection json so that the token can refresh.**