SCIM Delta Query

Anjali Sehgal (AWS) and Danny Zollner (Microsoft)

Goal

Goal

- Enable ***incremental retrieval of resources*** that have been updated or deleted in a SCIM service provider.
- This allows for more efficient interactions between SCIM clients and service providers and addresses problems that have inhibited ***large-scale*** implementation of use cases such as ***synchronization***, ***entropy detection***.

Why is it Important?

- Potential synchronization inaccuracies could lead to data divergence between the SCIM client and SCIM service provider. Undetected diverging data between a SCIM client and SCIM service provider can lead to undesirable authorization decisions.
- End to End Reconciliation processes, reduces the risk of incorrect authorization decisions based on divergent states between client and server

This data divergence detection may be used for r**eporting purposes** or may be extended to either **trigger provisioning** of those resources **into the target system** or **pulling changes from the target system** into the source.

Requirement

- Resources modified since a specific point can be returned by query
- Current state of resources returned
- Able to convey that a previously existing resource was deleted since specified point
- Able to convey changes to group memberships
- Performant at large scale with accurate results
- … others?

How does it work?

Step1: Obtaining the First Delta Token

The process starts with a full Scan Query as below

```
GET /Users?deltaQuery
```

In response to the full scan query the server

1. MUST return the resources that currently exist in the collection.
2. Resources returned will represent the latest state of the resource at the time processing of the request.
3. MUST return the **nextDeltaToken** on the last page of the full scan response.

This **nextDeltaToken** *will be used by SCIM client in subsequent delta query requests.*

Response

```
HTTP/1.1 200 OK
Content-Type: application/scim+json
{
     "totalResults":45,
     "itemsPerPage":50,
     "nextDeltaToken":"VTHKLOUTREO",
     "schemas":["urn:ietf:params:scim:
api:messages:2.0:ListResponse"],
     "Resources": [{
     …
     }]
}
```

How does it work?

Step2: Using Delta Token to perform a Delta Scan

```
GET /Users?deltaQuery&deltaToken=VTHKLOUTREO
```

In response to the delta scan query the server

1. MUST return the resources modified (created, updated or deleted) after the point represented by the delta token's value.
2. Resources returned will represent the latest state of the resource at the time processing of the request.
3. MUST return the **nextDeltaToken** on the last page of the delta scan response.

This **nextDeltaToken** *will be used by SCIM client in subsequent delta query requests*

Response

```
HTTP/1.1 200 OK
Content-Type: application/scim+json
{
  "totalResults":13,
  "itemsPerPage":50,
  "nextDeltaToken": "OPUTREWSFDE",
  "schemas":
["urn:ietf:params:scim:api:messages:2.
0:ListResponse"],
  "Resources": [{
  ...}]}
```

Resource Representation

Newly created and ***Updated resources*** are represented in the delta query response using their standard representation and their current state is returned.

Deleted instances MUST return common attribute **id** and complex attribute **meta** with sub attributes **resourceType** and **isDeleted** attribute with value **True**.

Minimal Representation for Deleted Instances

```
{
  "schemas":
      ["urn:ietf:params:scim:schemas:core:2.0:User"
],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "meta": {
    "resourceType": "User",
    "isDeleted": true
  }
}
```

Pagination - Getting the first page

```
GET /Users?deltaQuery&deltaToken=VTHKLOUTREO
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD


HTTP/1.1 200 OK
Content-Type: application/scim+json
{
  "totalResults":8649,
  "itemsPerPage":100,
  "nextCursor": "CVHNJKUYFRT",
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
  "Resources": [{
    ...
  }]
  }
```

Pagination - Getting Subsequent Pages

Client can retrieve subsequent pages by providing **cursor** value updated with the value received from the server in **nextCursor** parameter in the previous response.

```
GET /Users?deltaQuery&deltaToken=VTHKLOUTREO&cursor=CVHNJKUYFRT
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD

HTTP/1.1 200 OK
Content-Type: application/scim+json
{
  "totalResults":8649,
  "itemsPerPage":63,
  "prevCursor": "CVHNJKUYFRT",
  "nextCursor": PPEMQPXNAZX",
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
  "Resources": [{
    ...
  }]}
```

Pagination - Getting the DeltaToken

*When returning the last page of results, the service provider will omit the **nextCursor** value and will include the **nextDeltaToken** value.*

```
GET /Users?deltaQuery&deltaToken=VTHKLOUTREO&cursor=PPEMQPXNAZX
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD

HTTP/1.1 200 OK
Content-Type: application/scim+json
{
  "totalResults":8649,
  "itemsPerPage":63,
  "prevCursor": "PPEMQPXNAZX",
  "nextDeltaToken": ALEMQPXNAZX",
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
  "Resources": [{
    ...
  }]}
```

Pagination Consideration

- Service providers MUST NOT prevent resources from being updated (locking resources) while implementing delta query.

- New items can be added or existing items can be removed or updated while paginating through the response of the delta queries. The result set will contain eventually consistent data, however some implementations may choose to enforce strongly consistent data.

- The delta query MUST guarantee that the records modified (created, updated, or deleted) after any query that generates a delta token are returned when that same delta token is provided back by the client.

Service Provider Configuration

A SCIM Service provider implementing delta query SHOULD include the following additional attribute in JSON document returned by the /ServiceProviderConfig endpoint:

deltaQuery
A complex type that indicates delta query configuration options. OPTIONAL.

supported
A Boolean value specifying support of delta query. REQUIRED.

deltaTokenTimeOut
Non-negative integer specifying the maximum number hours that a deltaToken is valid between delta Scan requests. Clients waiting too long between subsequent delta scan requests may receive an invalid delta token error response. This is OPTIONAL as it may depend on server implementers if there is a timeout defined for the delta token issued by the server.

Group Memberships

SCIM currently does not support pagination of large complex multi-values attributes such as group.member.

Steps to Retrieve Group membership Deltas for a SCIM Client:

1. Get Groups that have members added to or deleted from.

   GET /Group?deltaQuery&deltaToken=VTHKLOUTREO

2. For each group returned in above query, retrieve the current active members for each affected group.

   GET /Users?groups.id=value

3. Compare the set of group members received from above query with the current set of members in Source System.

Above approach works but in case the groups contain large set of members this approach will become inefficient.

GITHub Link

https://github.com/ietf-scim-wg/draft-sehgal-scim-delta-query

Or Use QR Code to open the link