Demystifying Graph API and Service Now

Using a Service account, CrowdStrike and Ansible to glue it all together

Hailaeos.Troy@DenverGov.org

Red Hat Summit

AnsibleFest

Hailaeos.Troy@denvergov.org

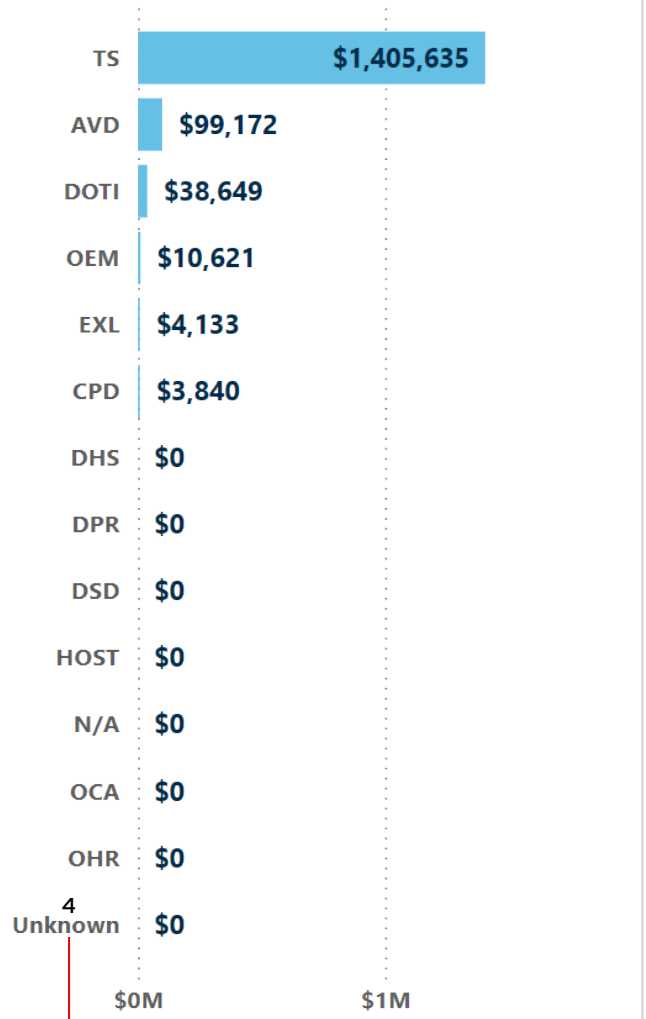
Sanitized Code
For All
Dudes and Dudetes

Red Hat

TS Automation

Last Refresh: 5/8/2024 3:30:10 PM

Total Dollar Savings by Agency



Total Dollar Savings*

*Data from 2023 to present

$1,562,051

$1,025,923 TS Soft Savings	$310,800 One-Time Savings
$171,433 Non-TS Soft Savings	$53,895 Hard Savings

Total Refocused Hours

24,804

20,518 TS Staff Hours Saved	
4,286 Non-TS Staff Hours Saved	

Averages

Rolling Average - Past 30 Days

$4,513 Dollars Saved per Day

81.8 Hours Saved per Day

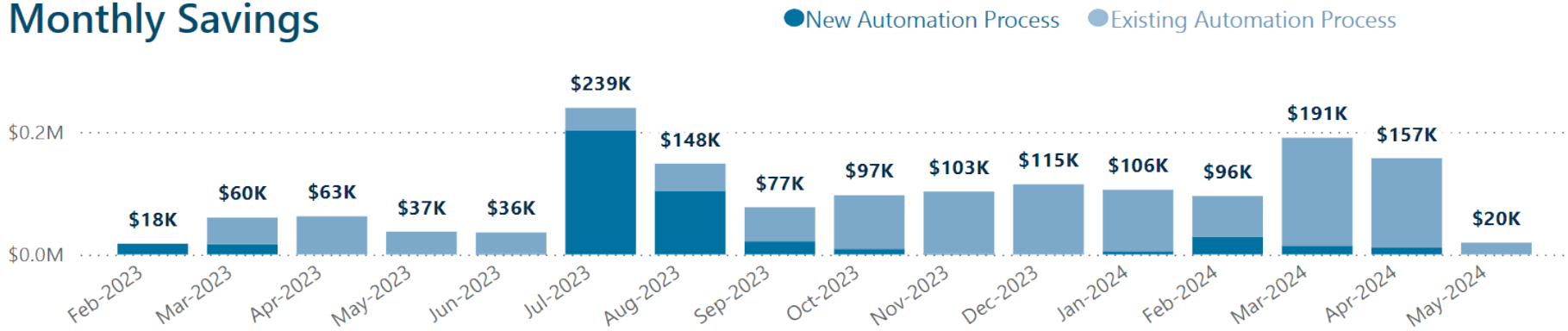
Automated Processes

133

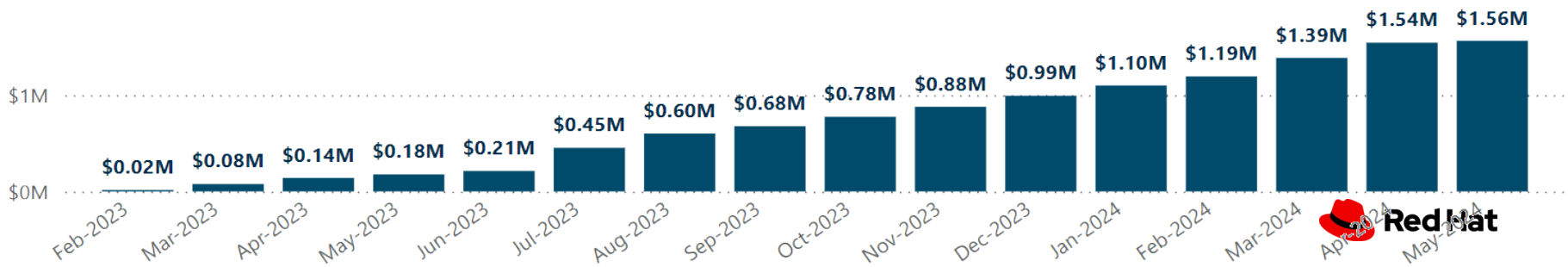
Year: All

Tool: Ansible

Monthly Savings



Accrued Savings



Red Hat

Step 1

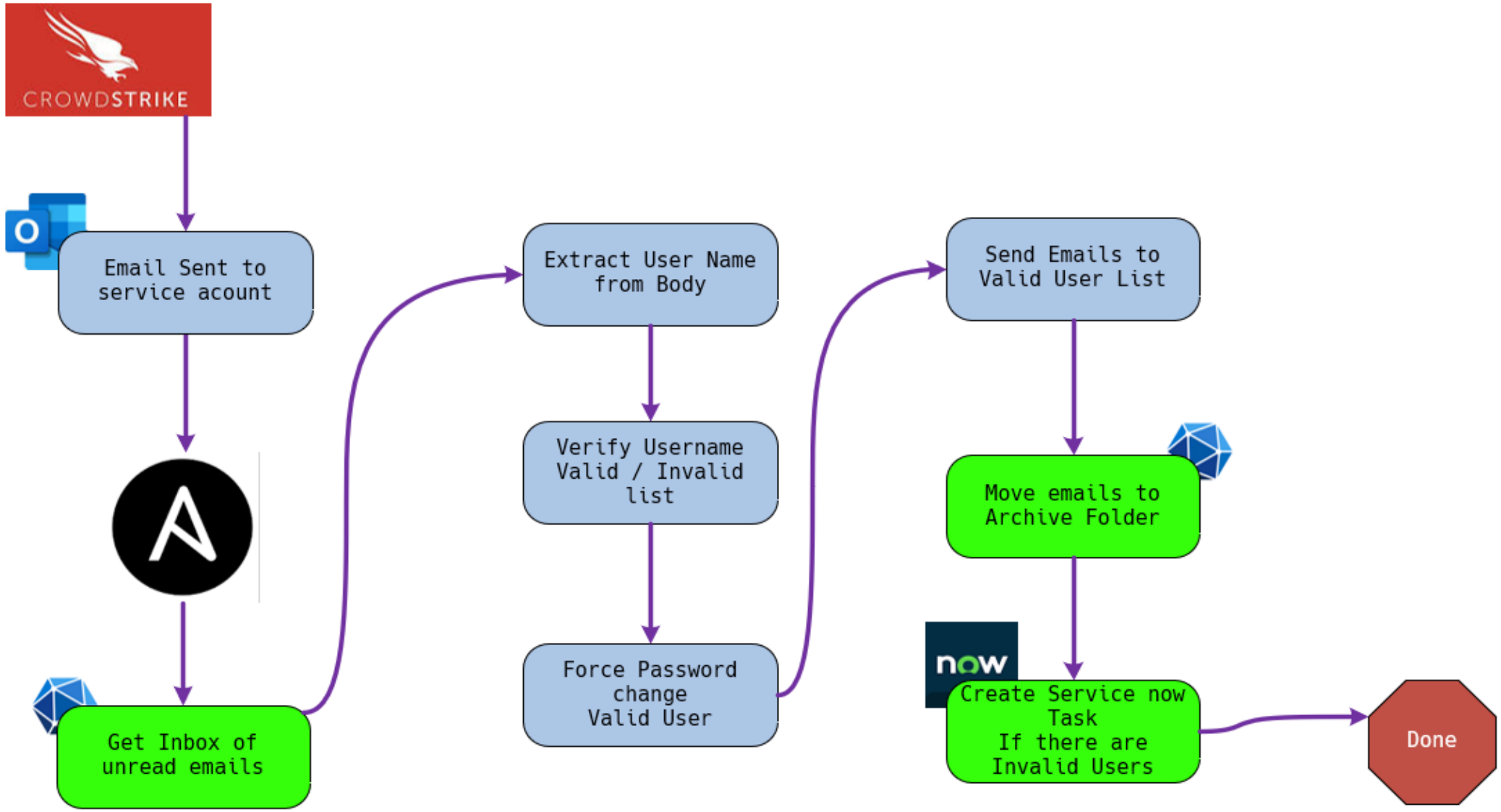
50,000 foot view

The Problem

- **CrowdStike sends an email to a service account, each email is a user that is using a password that is on a known compromise list. Like “password123”**
- **Ansible once an hour will do the following:**
- **Grab any new mail and process them (Graph API)**
- **Ansible then will match the account and send a notification and force a password change (A D)**
- **Move all the messages that were processed to the Archive folder (Graph API)**
- **Unmatched accounts will get a ServiceNow task assigned to Security (Service Now)**

Step 1

The Problem



Red Hat

25,000 foot view

Pieces to get the job done

1. **Get with the Azure Admin and create an Enterprise application and get the Delegated and Application permissions you need.** (in my case I have the needed permissions)

2. **Put your app secrets in your Ansible vault or your favorite secure location like Secret Server**

3. **Get with the ServiceNow folks get access to your Dev/Test/Sandbox environment and permission to access the Rest API Explorer its your friend, like me.**

4. **Get a ServiceNow service account setup with the correct permissions to create TASK**

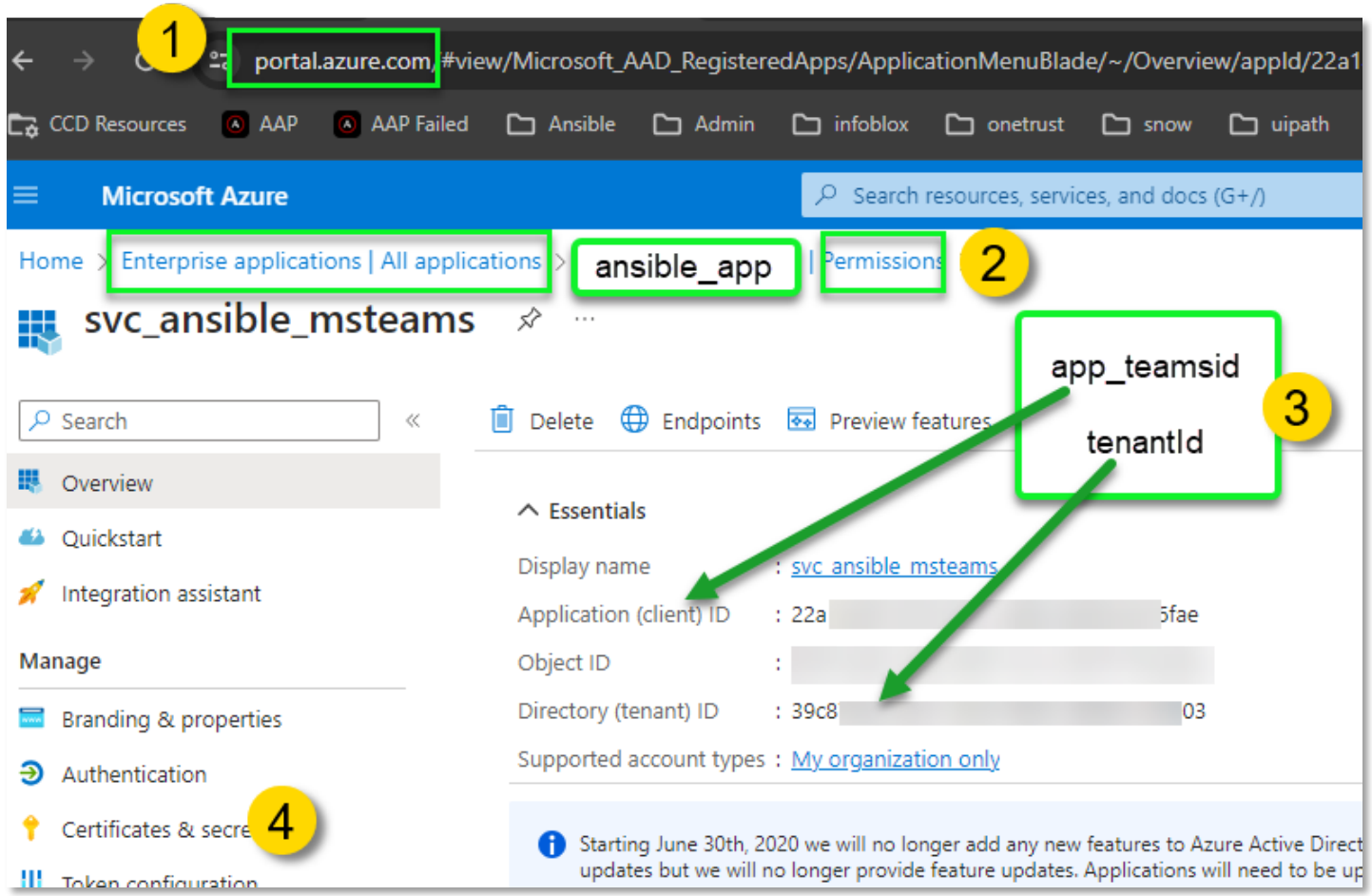
   Bonus points for getting impersonate rights in Dev/Test/Sandbox or full Admin.

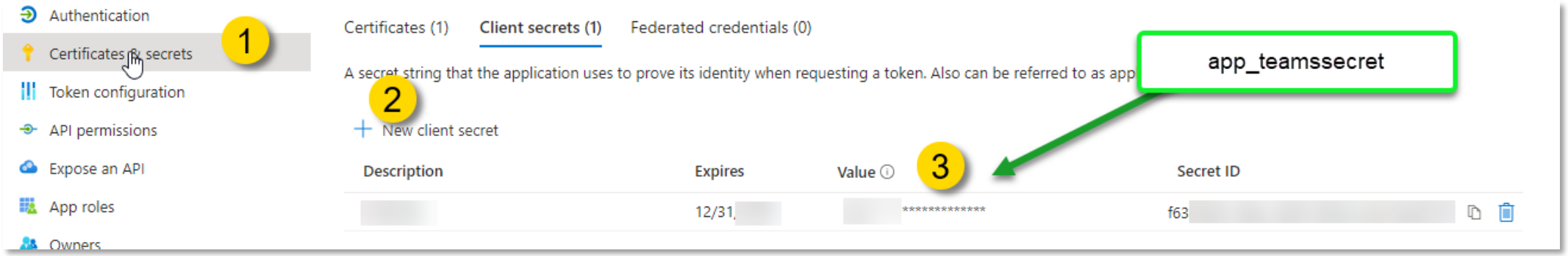
   Trust me it helps

Down in the weeds

Graph API task pieces

This is where you need your Azure Admin



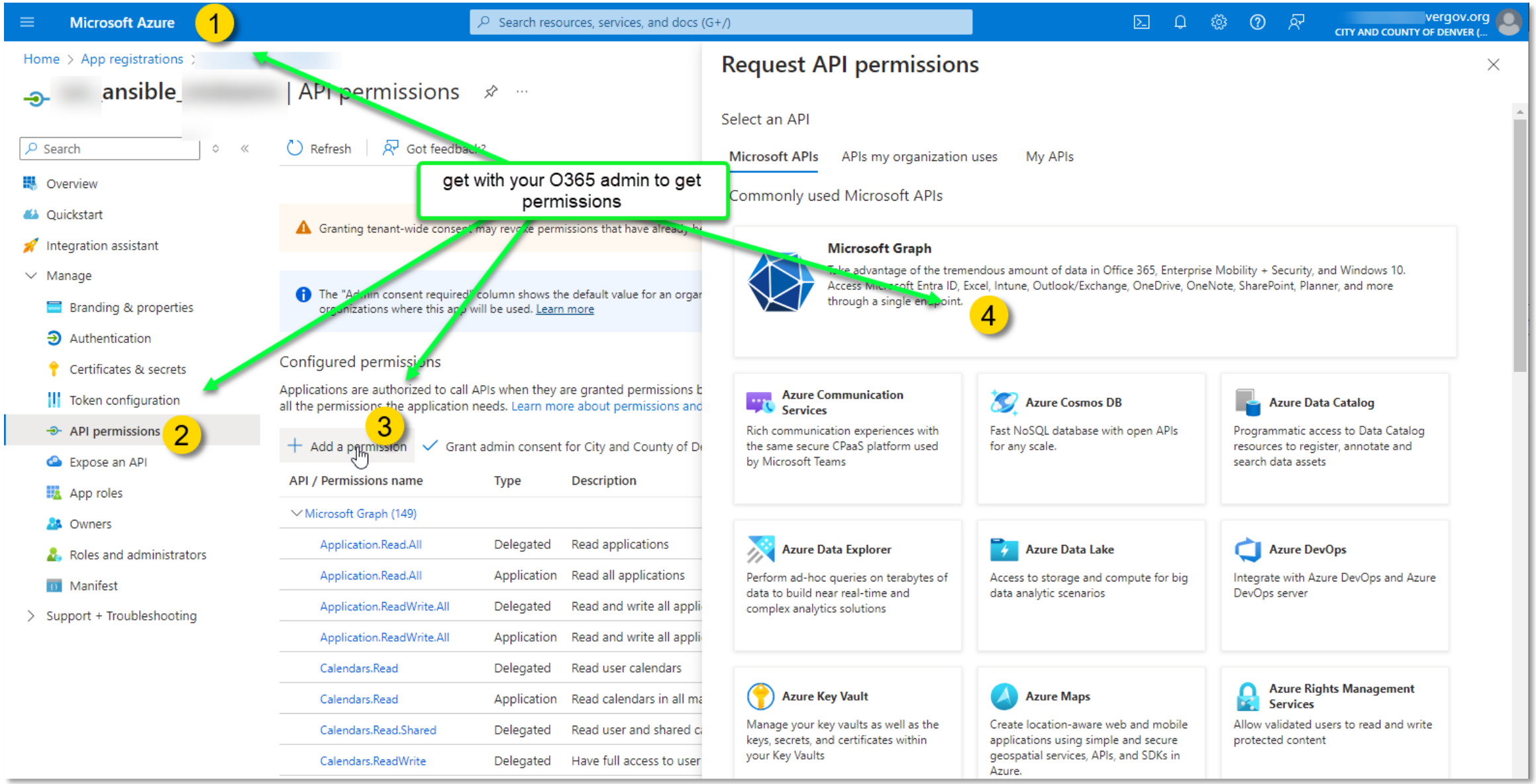


Red Hat

Down in the weeds

Graph API task pieces

This is where you need your Azure Admin



Red Hat

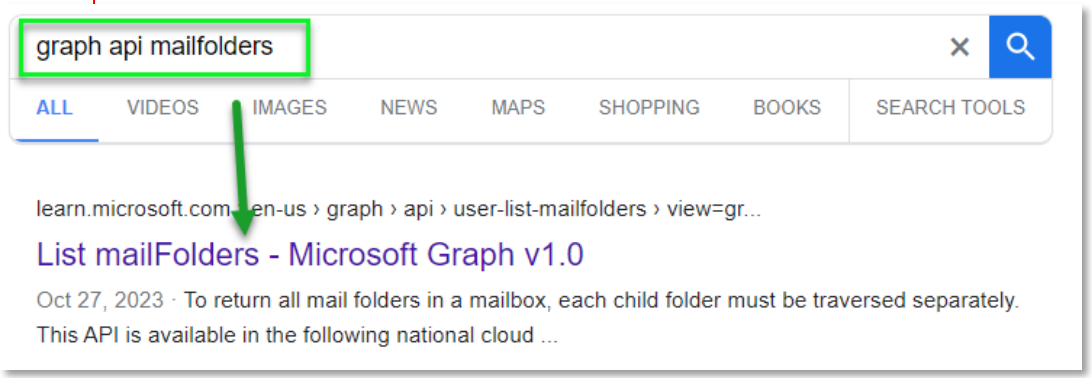
Get a Token

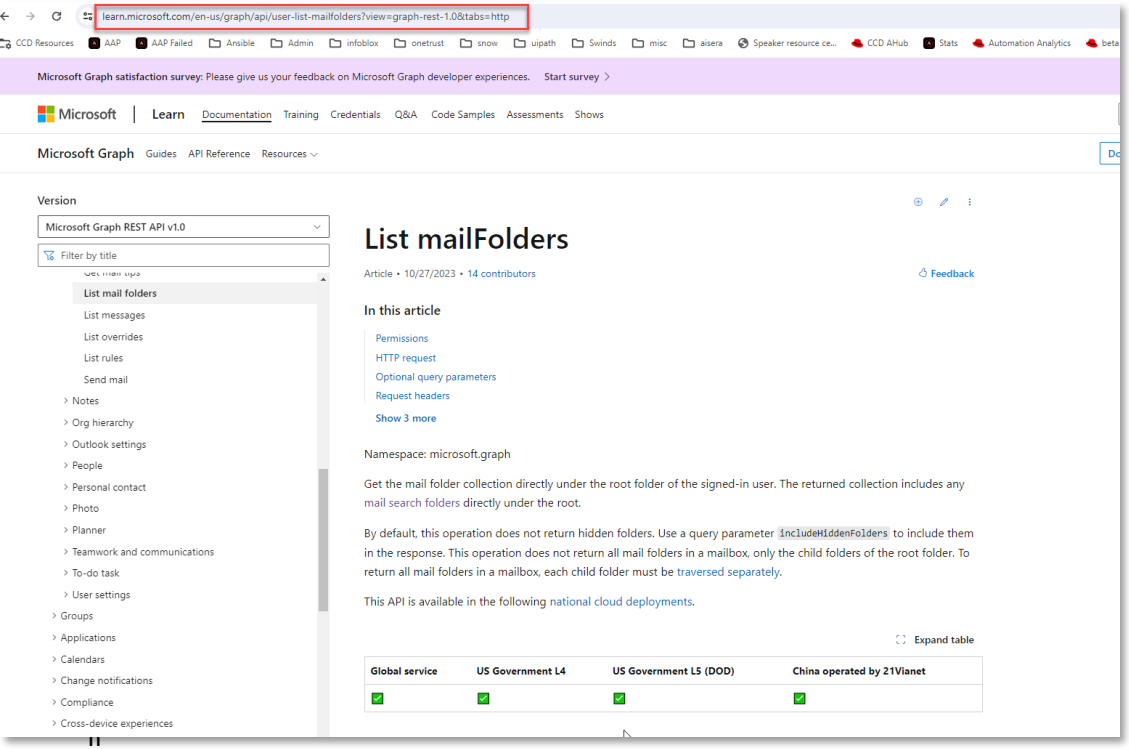
To curl:// or to ansible.builtin.uri

```yaml
- name: Get Application Token for microsoft graph api
  ansible.builtin.shell: |
    curl -s -k -X POST -d "grant_type=client_credentials\
    &client_id={{ app_teamsid }}&client_secret={{ app_teamssecret }}&scope=https://graph.microsoft.com/.default" \
    https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token \
    |jq -r ".access_token"
  register: shell_token
  changed_when: false
```

```yaml
- name: Get Graph API Application Token
  ansible.builtin.uri:
    url: "https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token"
    method: POST
    headers:
      Content-Type: "application/x-www-form-urlencoded"
    body_format: form-urlencoded
    body:
      grant_type: "client_credentials"
      client_id: "{{ app_teamsid }}"
      client_secret: "{{ app_teamssecret }}"
      scope: "https://graph.microsoft.com/.default"
  register: tokena
```

Microsoft Graph API





Permissions

Choose the permission or permissions marked as least privileged for this API. Use a higher privileged permission or permissions only if your app requires it. For details about delegated and application permissions, see Permission types. To learn more about these permissions, see the permissions reference.

Expand table

Permission type	Least privileged permissions	Higher privileged permissions
Delegated (work or school account)	Mail.ReadBasic	Mail.ReadWrite, Mail.Read
Delegated (personal Microsoft account)	Mail.ReadBasic	Mail.ReadWrite, Mail.Read
Application	Mail.ReadBasic.All	Mail.ReadWrite, Mail.Read

HTTP request

To get all the mail folders in the root folder in the specified user's mailbox, excluding those that are hidden:

HTTP Copy

```
GET /me/mailFolders
GET /users/{id | userPrincipalName}/mailFolders
```

To include *hidden* mail folders in the response:

HTTP Copy

```
GET /me/mailFolders/?includeHiddenFolders=true
GET /users/{id | userPrincipalName}/mailFolders/?includeHiddenFolders=true
```

Optional query parameters

To return a list of all mailFolders including those that are hidden (their **isHidden** property is true), in the request URL, specify the `includeHiddenFolders` query parameter as `true`, as shown in the HTTP request section.

This method supports OData query parameters to help customize the response.

Red Hat

```yaml
- name: "Get root folder of email "
  ansible.builtin.uri:
    url: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/mailFolders"
    method: GET
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    status_code:
      - 200
      - 201
      - 404
  register: root_folders
```

```yaml
- name: show results of root folders
  ansible.builtin.debug:
    msg:
      - "{{ root_folders }}"
    verbosity: "{{ verbosity_level }}"

- name: show results
  ansible.builtin.debug:
    msg:
      - "{{ root_folders | json_query(query) }}"
    verbosity: "{{ verbosity_level }}"
  vars:
    query: "json.value[?displayName==`Inbox`].id"
```

```yaml
- name: set fact for root folder
  ansible.builtin.set_fact:
    inbox: "{{ (root_folders | json_query(query))[0] }}"
  vars:
    query: "json.value[?displayName==`Inbox`].id"
```

```
# subject_search: "CrowdStrike Identity Protection Policy Notification - Rule 'Compromised Password Detected
'"
- name: "Get Inbox email"
  ansible.builtin.uri:
    url: "{{ url_search }}"
    method: GET
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    status_code:
      - 200
      - 201
      - 404
      - 400
  vars:
    url_search: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/mailFolders/{{ inbox }}/messages?
    \\$filter=startswith(Subject,'{{ subject_search|urlencode }}')&$top=2000"
  register: messages
```

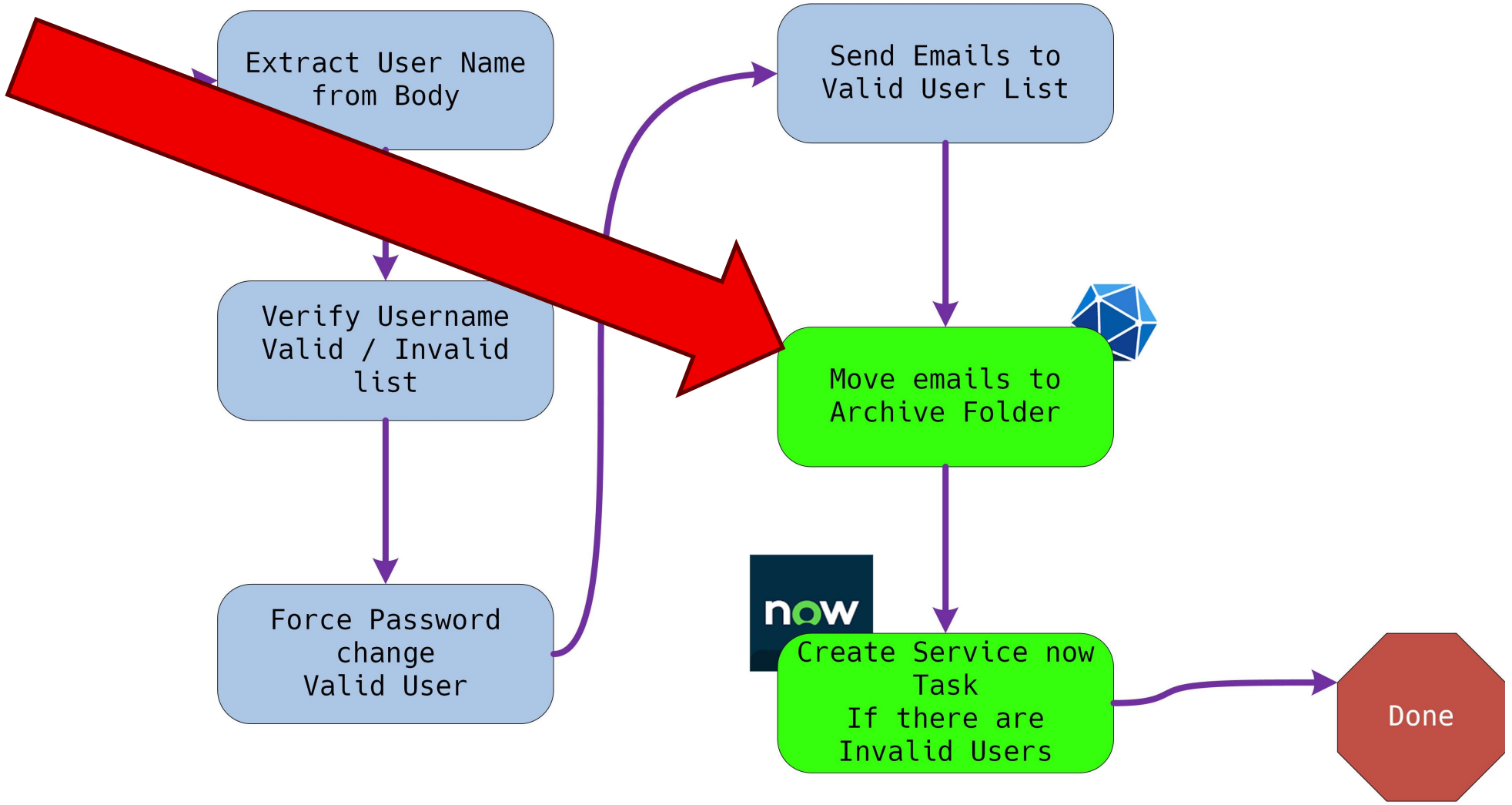
1

2

The reason behind using $top=2000 is because I am lazy, I didn't want to loop using the @odata.nextLink but if you do wind up having to do pagination Microsoft style this is now you do it

```
- set_fact:
    nextpage: "graphapidataresult.json['@odata.nextLink']"
```

Mark the messages read and move them to the archive folder



Bonus points if you guess the year I started using Visio, clue take a look at the beard

Mark messages read and move to Archive folder

Set a fact to the email id's but do it as a batch of 100 messages to avoid a graph timeout

1
```yaml
- name: set fact to emails id
  ansible.builtin.set_fact:
    email_id: "{{ messages | json_query(queryid) }}"
  vars:
    queryid: "json.value[*].id"
```

2
```yaml
- name: set fact to email list batch
  set_fact:
    email_id_batch: "{{ email_id |batch(100) }}"
```

3
```yaml
- name: include graph batch
  include_tasks:
    tasks/graph_batch.yml
  loop: "{{ email_id_batch }}"
```

4
```yaml
- name: Get Graph API Application Token
  ansible.builtin.uri:
    url: "https://login.microsoftonline.com/{{ tenantId }}/oauth2/v2.0/token"
    method: POST
    headers:
      Content-Type: "application/x-www-form-urlencoded"
    body_format: form-urlencoded
    body:
      grant_type: "client_credentials"
      client_id: "{{ app_teamsid }}"
      client_secret: "{{ app_teamssecret }}"
      scope: "https://graph.microsoft.com/.default"
  register: tokena
```

Mark messages read and move to Archive folder

Patch the mail record and mark it read

```yaml
- name: "Mark messages read"
  ansible.builtin.uri:
    url: "{{ url_search }}"
    method: PATCH
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    body_format: json
    body:
      isRead: true
    status_code:
      - 200
      - 201
      - 404
      - 400
  vars:
    url_search: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/messages/{{ item_z }}"
  ignore_errors: true
  until: message_patch is not failed
  retries: 20
  delay: 60
  loop: "{{ item }}"
  loop_control:
    extended: true
    loop_var: item_z
  register: message_patch
```

1) Patch the mail id
2) loop our 100 messages
3) set a custom loop
4) item_z = our message id
5) mark the message read
6) if the message is failed
   retry 20 times every minute

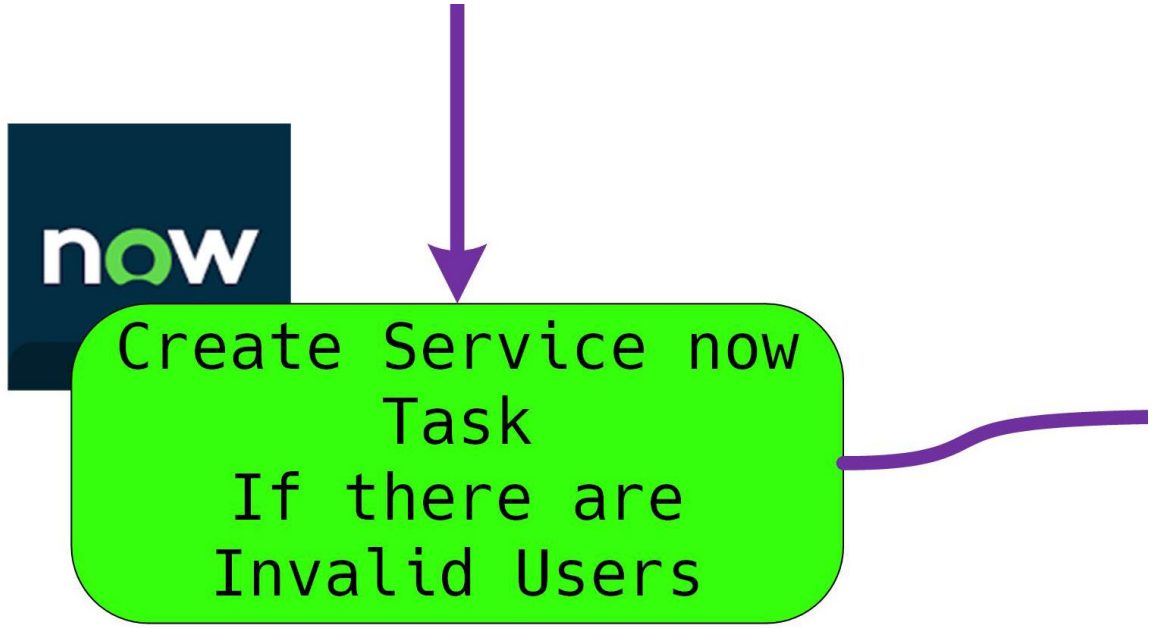
Mark messages read and move to Archive folder

Move the message now

```yaml
- name: "Move message to Archive "
  ansible.builtin.uri:
    url: "{{ url_move }}"
    method: POST
    headers:
      Authorization: 'Bearer {{ token }}'
      Content-Type: application/json
    body_format: json
    body:
      destinationId: "{{ (root_folders | json_query(destfolder))[0] }}"
    status_code:
      - 200
      - 201
      - 404
      - 400
  vars:
    url_move: "https://graph.microsoft.com/v1.0/users/{{ mailaccount }}/messages/{{ item_z }}/move"
    destfolder: "json.value[?displayName==`Archive`].id"
  register: message_move
  ignore_errors: true                    ##################################
  until: message_move is not failed      # This is for unstable API's     #
  retries: 20                            #                                #
  delay: 60                              ##################################
  loop: "{{ item }}"
  loop_control:
    loop_var: item_z
```

1,000 foot view

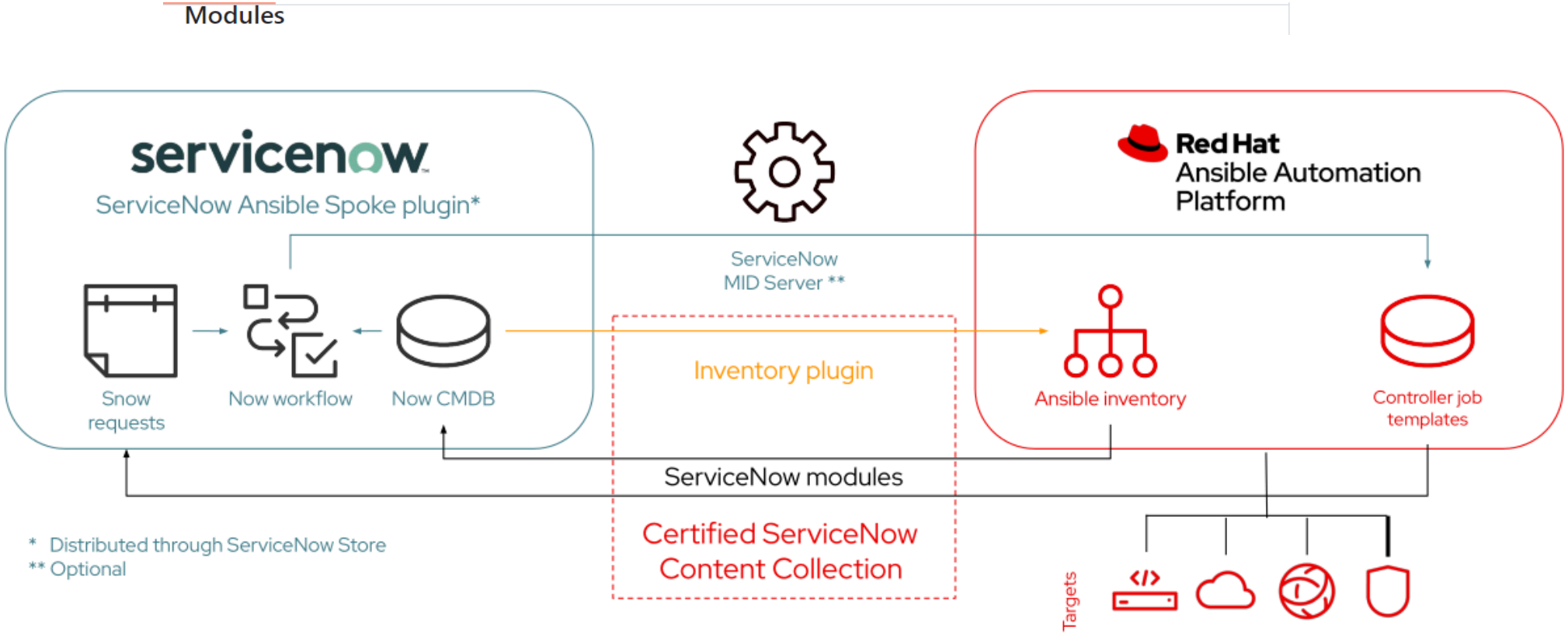
Lay it out, use your favorite diagram program I use Visio



Bonus points if you guess the year I started using Visio, clue take a look at the beard

ServiceNow, SNOW

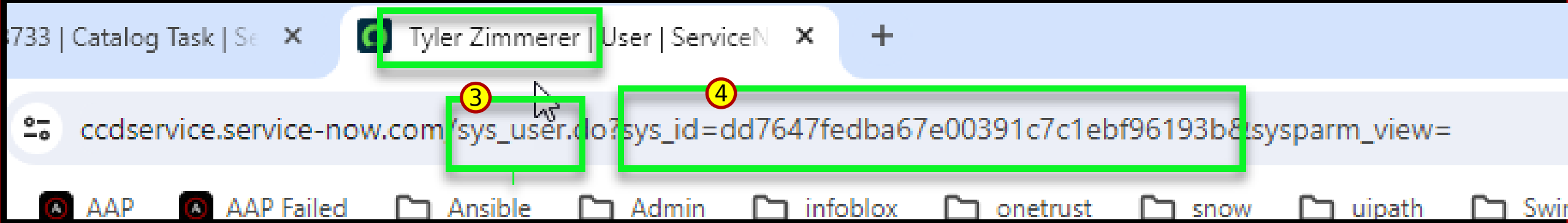
servicenow.itsm vs. ansible.builtin.uri



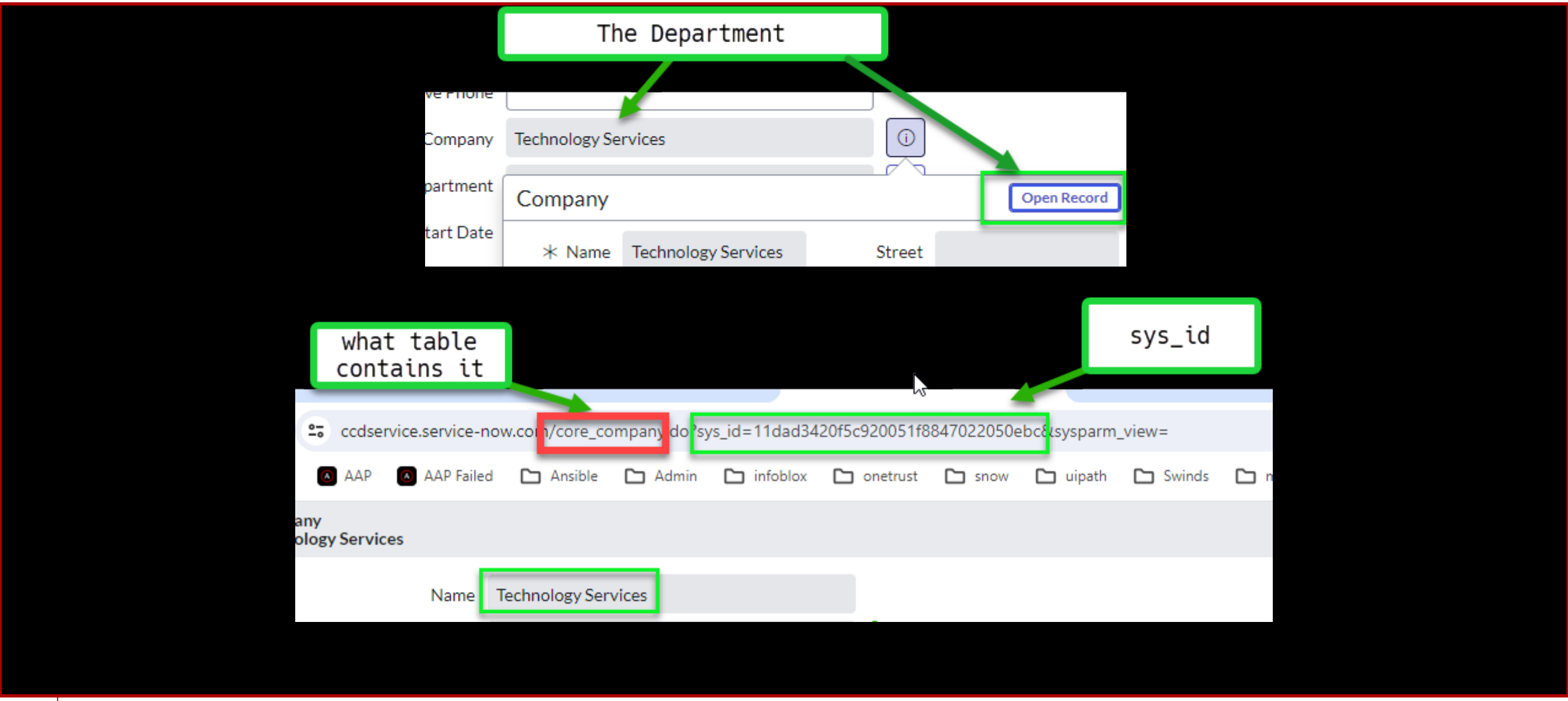
servicenow.itsm.problem	Manage ServiceNow problems
servicenow.itsm.problem_info	List ServiceNow problems
servicenow.itsm.problem_task	Manage ServiceNow problem tasks
servicenow.itsm.problem_task_info	List ServiceNow problem tasks

Red Hat

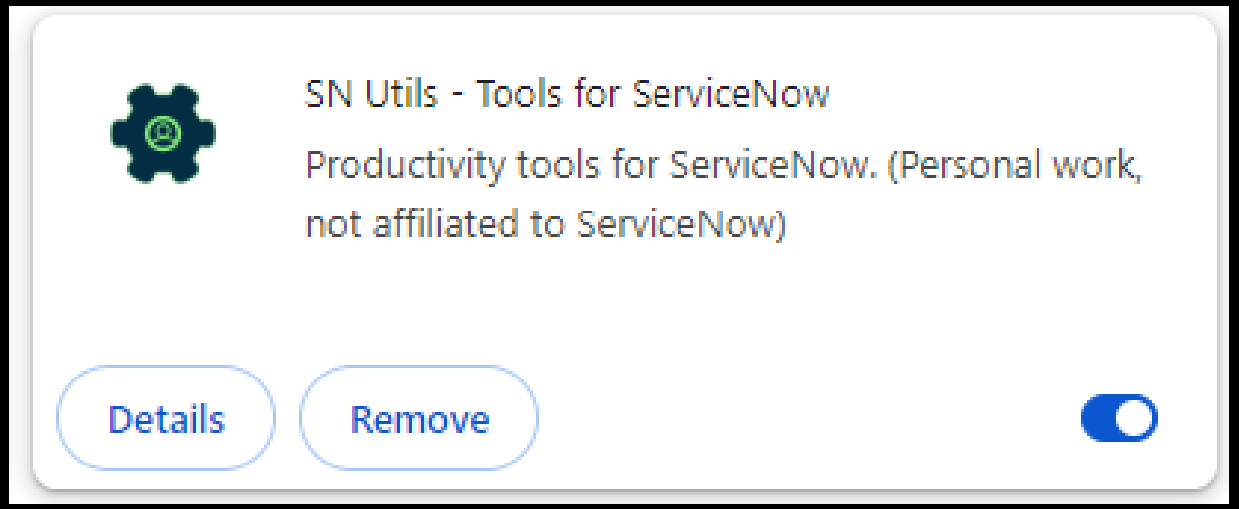
ServiceNow – Its in front of you just Look



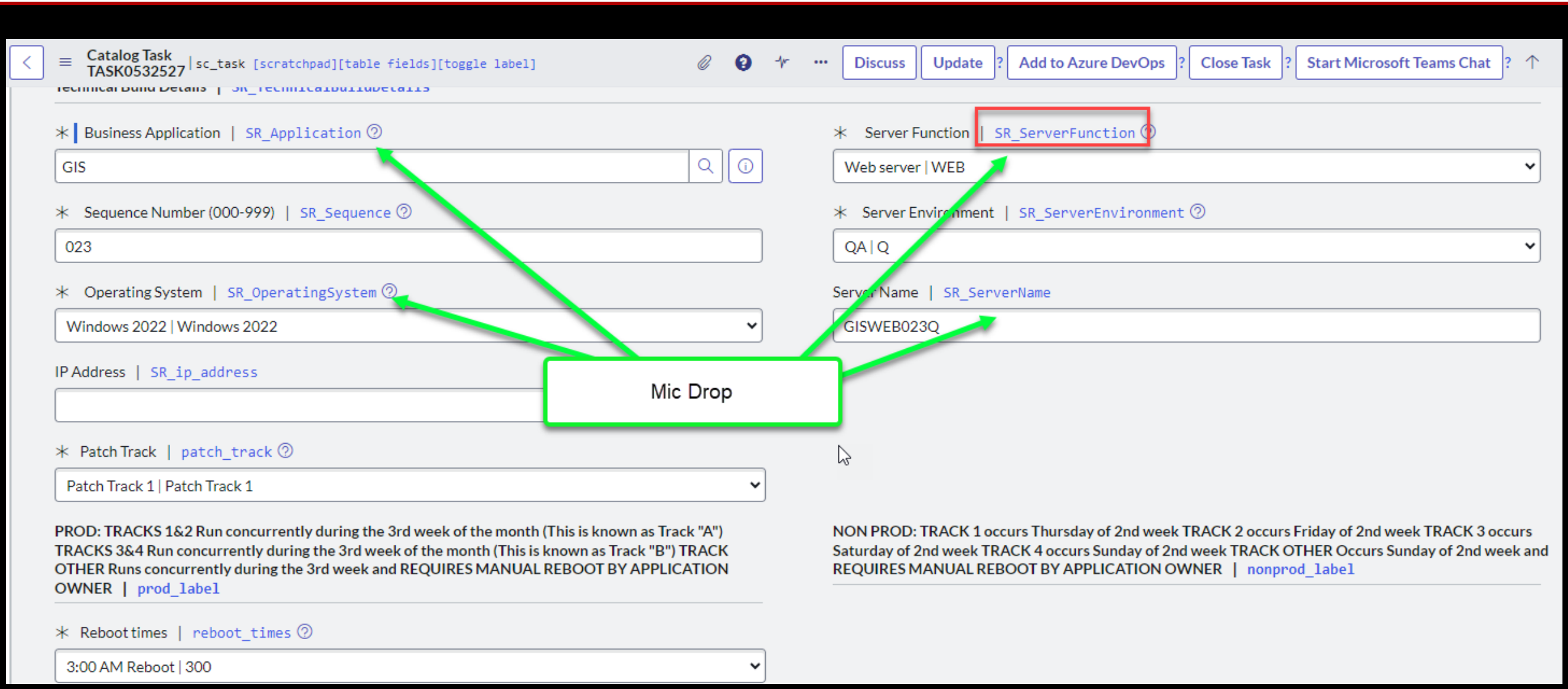
ServiceNow – Its in front of you just Look



ServiceNow – There is a tool



ServiceNow – There is a tool



ServiceNow – Rest API Explorer – get a user

Table API

Allows you to perform create, read, update and delete (CRUD) operations on existing tables

Retrieve records from a table

```
GET https://ccdsandbox.service-now.com/api/now/table/{tableName}
```

Prepare request

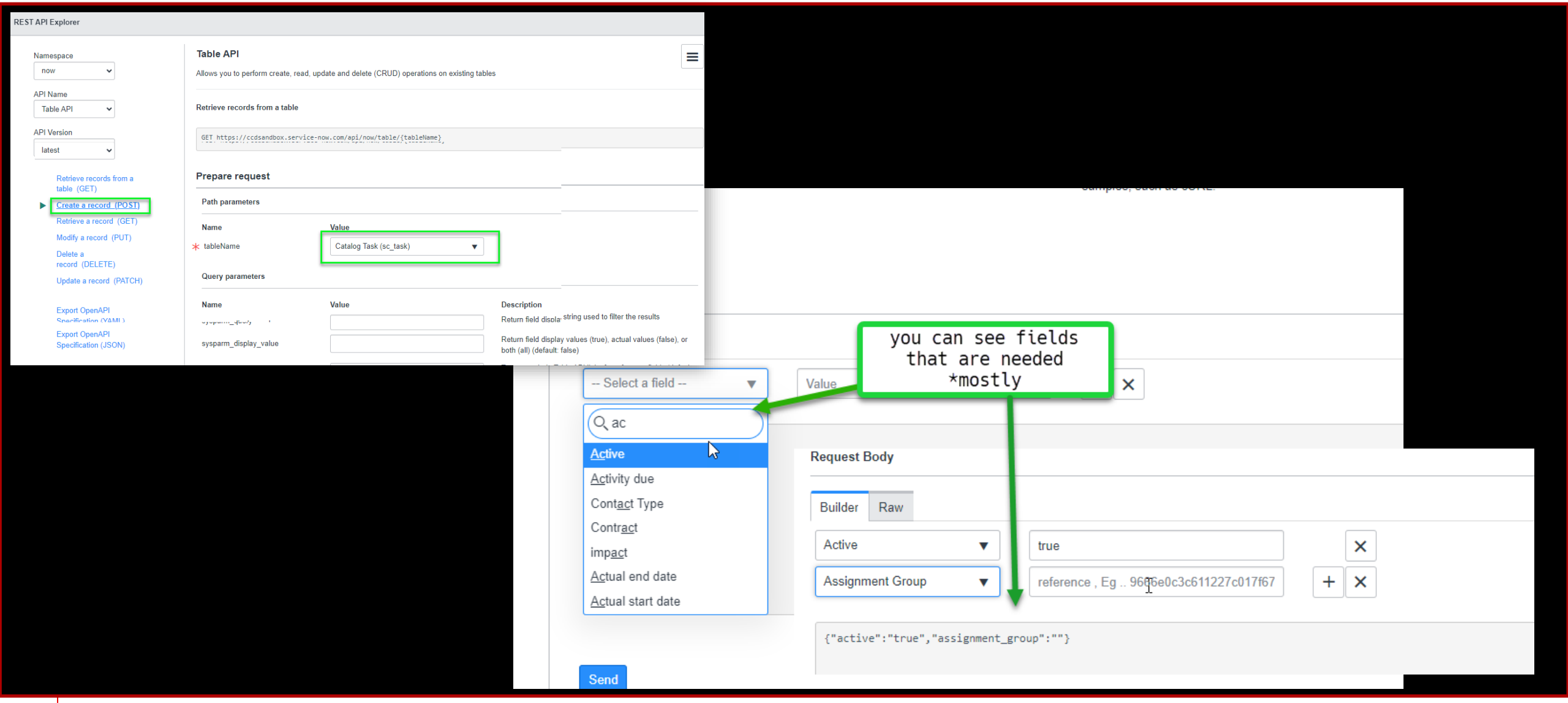
Path parameters

Name	Value
* tableName	-- Select a table --

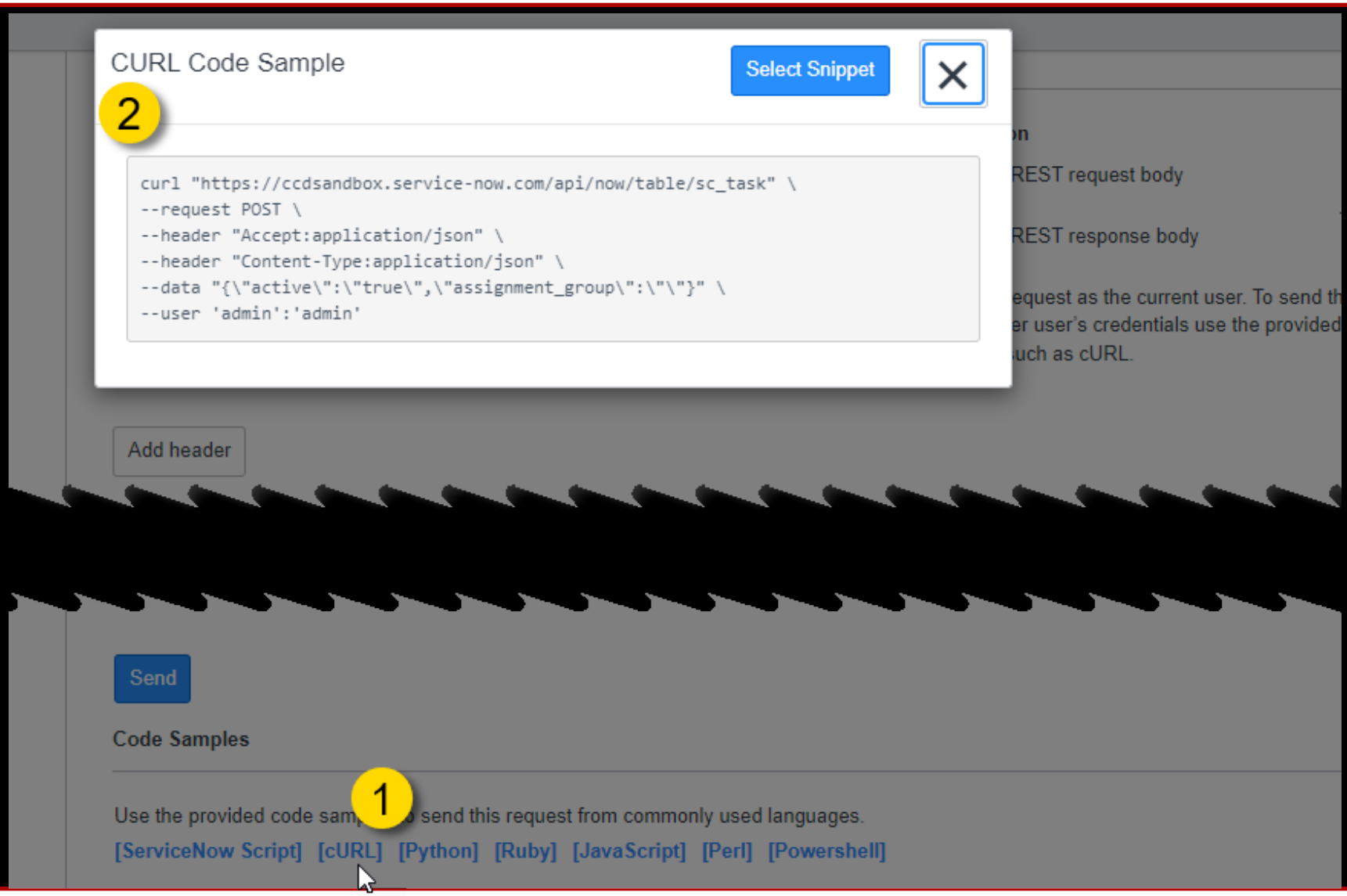
Query parameters

Name	Value	Description
sysparm_query		An encoded query string used to filter the results
sysparm_display_value		Return field display values (true), actual values (false), or both (all) (default: false)
sysparm_exclude_reference_link		True to exclude Table API links for reference fields (default: false)
sysparm_suppress_pagination_header		True to supress pagination header (default: false)

ServiceNow – Rest API Explorer



ServiceNow – Rest API Explorer



ServiceNow – uri get user

```yaml
- name: Search for user
  ansible.builtin.uri:
    url: "https://{{ snow_instance }}.service-now.com/api/now/table/sys_user?name={{ task_assignment_user|urlencode }}"
    method: GET
    user: "{{ Uname }}"
    password: "{{ Upass }}"
    headers:
      Accept: application/json
      Content-Type: application/json
    body_format: json
    status_code:
      - 200
      - 201
  register: snow_user_results
```

1 2 3

```yaml
- name: set key facts based searches
  ansible.builtin.set_fact:
    assignedto: "{{ snow_user_results.json.result[0].sys_id }}"
    cmdbci: "{{ snow_cmdb_ci_results.json.result[0].sys_id }}"
    assignedgroup: "{{ snow_group_results.json.result[0].sys_id }}"
```

ServiceNow – uri create task

```yaml
- name: Create Task
  ansible.builtin.uri:
    url: "https://{{ snow_instance }}.service-now.com/api/now/table/sc_task"
    method: POST
    body_format: json
    headers:
      Accept: "application/json"
      Content-Type: "application/json"
    body:
      active: "true"
      assigned_to: "{{ assignedto }}"
      u_ci_class: "{{ task_ci_class }}"
      assignment_group: "{{ assignedgroup }}"
      cmdb_ci: "{{ cmdbci }}"
      description: "{{ task_description }}"
      short_description: "{{ task_short_name }}"
      priority: "{{ task_priority }}"
      work_notes: "{{ task_work_notes }}"
      Approval: "approved"
    url_password: "{{ Upass }}"
    url_username: "{{ Uname }}"
    status_code: 201
    validate_certs: false
  register: create_task
```

1

ServiceNow – Make is a role in your collection

```
- name: Create Task for Security for accounts not found in AD
  when: '(email_addresses_notfound |length | int ) > 0'
  include_role:
    name: ccd.master_roles.snow_create_task                    # if using automation hub
    public: true
  vars:
    snow_instance: "{{ servicenow }}"                          # snow instance to use
    task_short_name: "CS compromised accounts not found"       # short description
    task_description: "Crowdstrike compromised accounts users not found: [code]{{ email_addresses_notfound  |regex_replace(sqq,'<br>')|
    regex_replace(sq,'')|regex_replace('\\[','<br>') |regex_replace('\\]','<br>')  }}[/code]" # long description
    task_assignment_group: "TS Information Security"           # assignment group if empty then gotten from user below
    task_cmdb_ci: "Crowdstrike"                                # ci aka SERVER
    task_ci_class: "Business Application"                      # can be Server just take a look at snow drop down
    task_work_notes: "[code]{{ email_addresses_notfound  |regex_replace(sqq,'<br>')|regex_replace(sq,'')|regex_replace('\\[','<br>') |
    regex_replace('\\]','<br>')  }}[/code]"
    task_priority: "4" #1 = high 3 moderate 4 low
```

1 2 3 4 5 6 7 8

Hailaeos.Troy@DenverGov.org

Red Hat Summit

AnsibleFest

Hailaeos.Troy@denvergov.org

Sanitized Code
For All
Dudes and Dudetes

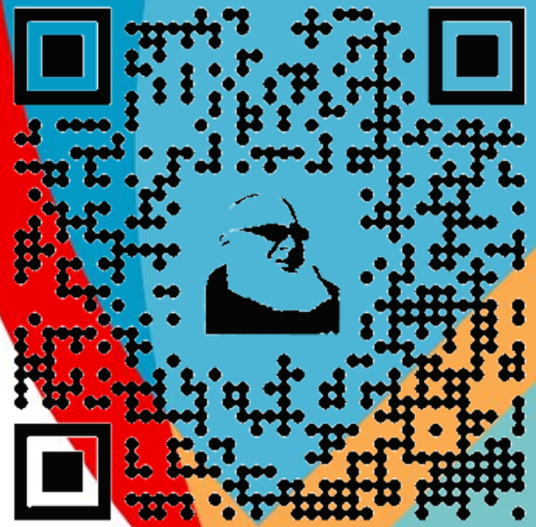
Red Hat

Red Hat Summit

AnsibleFest

Thank you

Hailaeos.Troy@denvergov.org



Sanitized Code
For All
Dudes and Dudetes

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat