

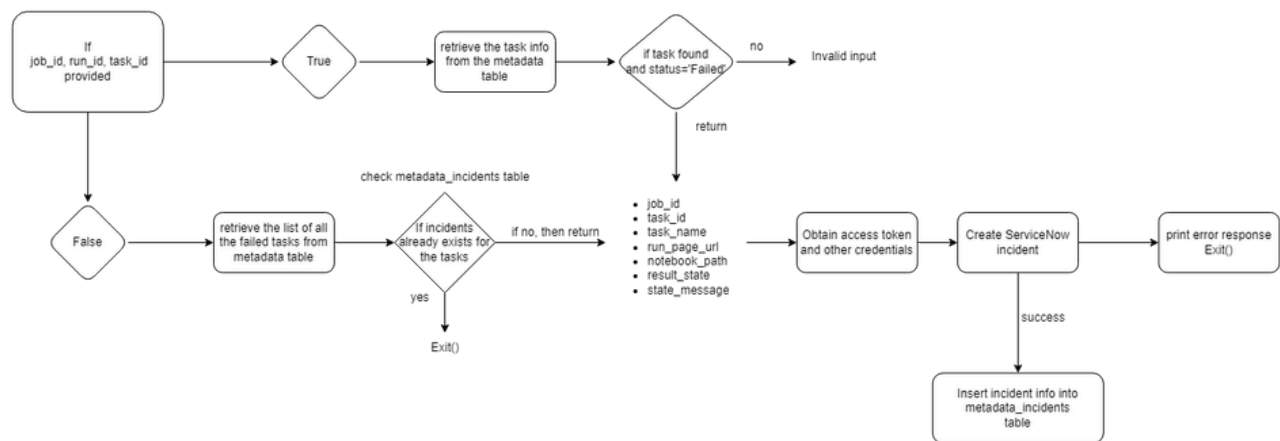
Workflow ServiceNow Integration for continuous job

- [1. Overview](#)
- [2. Retrieve failed tasks from metadata table](#)
- [3. Obtaining Access Token to use the API Services](#)
- [4. Creating ServiceNow incident](#)
- [5. Insert into 'metadata_incidents' table](#)

1. Overview

This is useful for continuous jobs. The idea is to retrieve all the failed tasks information from the audit metadata table and create ServiceNow incidents accordingly if not already created. Once the incidents are created, the information (job_id, run_id, task_id, incident_number) is logged in the table.

Another option is to retrieve job_id, run_id, task_id from workflow and retrieve the task information from the metadata table. If any of the tasks are Failed, an incident is created if not already exists in the table.



2. Retrieve failed tasks from metadata table

- retrieve the info of all the failed tasks from audit metadata table and create a ServiceNow incident for each task if the incident not exists in table

```
select JOB_ID,TASK_NAME,RUN_ID,JOB_STATUS,EXCEPTION from table_name
```

The information required to population description fields can be directly retrieved from this table.

3. Obtaining Access Token to use the ServiceNow API Services

```
def get_access_token(access_token_url,client_id,client_secret,resource):

    headers={ 'Content-Type':'application/x-www-form-urlencoded'}
    payload= {
        'f':'pjson',
        'grant_type':'client_credentials',
        'client_id':client_id,
        'client_secret':client_secret,
        'resource':resource
    }

    try:

        response=requests.post(access_token_url,headers=headers,data=payload)
        if response.status_code==200:
            token=response.json().get('access_token')
            return (token)
        else:
            print(f'Failed to obtain access token. Status Code:{response.status_code}')
            print(f'Response content: {response.content.decode()}')

    except Exception as e:
        print(f'Error:{str(e)}')
```

4. Creating ServiceNow incident

creating incident with necessary information(urgency, impact, cmdb_ci, caller_id, location, assignment_group, short_description and description)

HTTP Verb: POST

URL: <https://xxxxxxxxxxxxx/apg-001-servicenow/v1/now/table/incident/>

Header Key/Value Pairs:

- Authorization: "Bearer access-token"
- Accept: application/json

Body (raw json)

```
{
  "incident_object":{
    "urgency":"3",
    "short_description":"test",
    "impact":"2",
    "cmdb_ci":
    "caller_id":
    "assignment_group":
    "description":"test purpose"
  }
}
```

If the incident is successfully created, you will see the http code 200

```
create_servicenow_incident(incident_url,access_token_url,client_id,client_secret,resource,urgency,short_description,impact,cmdb_ci,caller_id,location,assignment_group,description)
```

<Response [200]>

```
{'result': {'status_code': '200',  
'status_message': 'Success',  
'return_response': {'number': 'INC000027970958',  
'u_state': 'Assigned',  
'state': 'Assigned'}}}
```

Command took 3.21 seconds

5. Insert into 'metadata_incidents' table

If the incident is successfully created, the info (job_id, run_id, task_id, incident_number, insert_ts) is logged into the metadata table.

Results ▾ +

#	job_id	run_id	task_id	incident_number	insert_ts	
1	735613540208879	366540696547207	841713507811913	INC000027975510	2024-02-14 19:19:53.378	
2	1017276741009767	1108300049185632	309429262755067	INC000027975506	2024-02-14 19:19:26.709	
3	1017276741009767	813533526897177	977239418249308	INC000027975507	2024-02-14 19:19:33.012	
4	735613540208879	366540696547207	379816679395618	INC000027975509	2024-02-14 19:19:45.050	
5	946750497411856	1008705271044927	177617531209515	INC000027975565	2024-02-14 21:19:44.267	
6	746665389274270	90564139465672	218912937437551	INC000027975511	2024-02-14 19:19:59.722	
7	746665389274270	148176397221081	55910317137531	INC000027975505	2024-02-14 19:19:03.299	
8	735613540208879	366540696547207	429307283463946	INC000027975508	2024-02-14 19:19:39.186	