**Automating Change Request (CR) Creation & Approval via ServiceNow API**

**Summary of Automation**

Create CR with ITSM permissions via ServiceNow API  
Auto-approve CR based on risk & impact conditions  
Python script to create & check CR approval  
 Jenkins pipeline to automate CR creation & deployment  
GitHub integration to create CRs from pull requests  
ServiceNow notifications in Jira for CR approvals  
Rollback mechanism if CR is rejected

**1. Create a Change Request (CR) via ServiceNow API**

**API Endpoint:**

POST https://your-instance.service-now.com/api/now/table/change\_request

**Sample cURL Request:**

curl -X POST "https://your-instance.service-now.com/api/now/table/change\_request" \

--header "Content-Type: application/json" \

--header "Authorization: Basic YOUR\_BASE64\_ENCODED\_CREDENTIALS" \

--data '{

"short\_description": "Automated Change Request with ITSM Permissions",

"description": "CR created via API with necessary ITSM access",

"category": "Software",

"risk": "low",

"impact": "low",

"assignment\_group": "ITSM Team",

"requested\_by": "umrao@example.com",

"state": "new"

}'

✅ This will create a CR in ServiceNow with ITSM permissions. ➡️ Ensure the "assignment\_group": "ITSM Team" is correct.

**2. Automate CR Approval Based on ITSM Rules**

You can auto-approve CRs if they meet criteria like: ✅ Risk = Low  
✅ Impact = Low

**Auto-Approval Business Rule in ServiceNow**

1. Go to **System Definition → Business Rules**
2. Click **New**
3. Configure the rule:
   * **Name:** Auto Approve Low-Risk CR
   * **Table:** Change Request
   * **When:** Insert & Update
4. **Condition:**

current.risk == "low" && current.impact == "low"

1. **Action:**

current.state = "approved";

current.update();

✅ Now, low-risk CRs will be automatically approved!

**3. Automate CR Creation & Approval via Python Script**

**Python Script:**

import requests

import time

import json

from requests.auth import HTTPBasicAuth

# ServiceNow Credentials & Instance

SNOW\_INSTANCE = "your-instance.service-now.com"

SNOW\_USER = "your-username"

SNOW\_PASS = "your-password"

# Create Change Request

def create\_cr():

url = f"https://{SNOW\_INSTANCE}/api/now/table/change\_request"

headers = {"Content-Type": "application/json"}

data = {

"short\_description": "Automated CR via Python",

"description": "This CR is created via API with ITSM permissions",

"category": "Software",

"risk": "low",

"impact": "low",

"assignment\_group": "ITSM Team",

"requested\_by": "umrao@example.com",

"state": "new"

}

response = requests.post(url, headers=headers, auth=HTTPBasicAuth(SNOW\_USER, SNOW\_PASS), json=data)

if response.status\_code == 201:

cr\_id = response.json().get("result", {}).get("sys\_id", "")

print(f"CR Created Successfully! CR ID: {cr\_id}")

return cr\_id

else:

print(f"Failed to create CR: {response.text}")

return None

# Check CR Approval

def check\_cr\_approval(cr\_id):

url = f"https://{SNOW\_INSTANCE}/api/now/table/change\_request/{cr\_id}"

headers = {"Accept": "application/json"}

for \_ in range(10): # Polling for 5 minutes

response = requests.get(url, headers=headers, auth=HTTPBasicAuth(SNOW\_USER, SNOW\_PASS))

if response.status\_code == 200:

state = response.json().get("result", {}).get("state", "")

if state == "approved":

print("CR Approved! Proceeding with deployment...")

return True

print("Waiting for CR approval...")

time.sleep(30)

print("CR was not approved in time!")

return False

# Main Execution

cr\_id = create\_cr()

if cr\_id:

check\_cr\_approval(cr\_id)

**4. GitHub Integration to Create CRs from Pull Requests**

**Automate CR Creation from GitHub PRs:**

1. Use a GitHub webhook to trigger CR creation when a pull request is opened.
2. Configure the webhook to send a POST request to a service that calls the ServiceNow API.

Example GitHub Action:

name: Create CR on PR

on:

pull\_request:

types: [opened, reopened]

jobs:

create-cr:

runs-on: ubuntu-latest

steps:

- name: Call ServiceNow API

run: |

curl -X POST "$SNOW\_URL" \

--user "$SNOW\_USER:$SNOW\_PASS" \

--header "Content-Type: application/json" \

--data '{"short\_description": "CR for PR #${{ github.event.pull\_request.number }}"}'

**5. ServiceNow Notifications in Jira for CR Approvals**

* Configure ServiceNow to send notifications to Jira using webhooks.
* When a CR is approved, trigger a Jira automation rule to update the related ticket.

Example Jira Webhook:

{

"trigger": "Change Request Approved",

"action": "Update Jira Ticket",

"jira\_ticket": "${ticket\_id}",

"status": "Change Approved"

}

**6. Rollback Mechanism if CR is Rejected**

* If a CR is rejected, trigger a rollback action.
* Use Jenkins or Ansible to revert the deployment.

Example Jenkins Rollback Step:

stage('Rollback') {

when {

expression { return env.CR\_STATE == 'rejected' }

}

steps {

echo "Rolling back deployment..."

sh 'ansible-playbook rollback.yml'

}

}