**------ Date increment ------------------------------------------------------------------------------------------------**

---

- hosts: localhost

tasks:

- set\_fact:

planned\_start\_date: '2018-12-19'

planned\_end\_date: '2018-12-25'

change\_iteration: 2

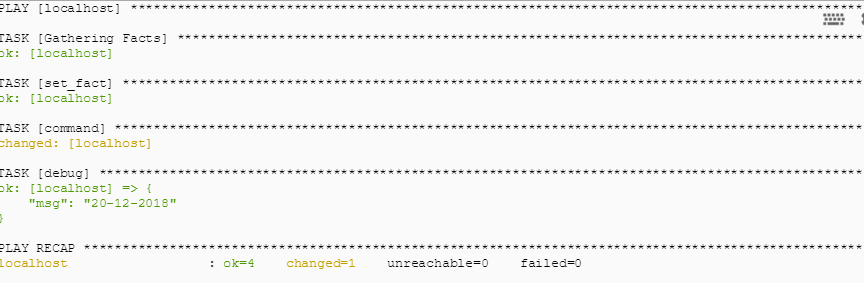
- command: date +'%d-%m-%Y' -d "{{planned\_start\_date}} + {{change\_iteration}} days"

register: end\_date

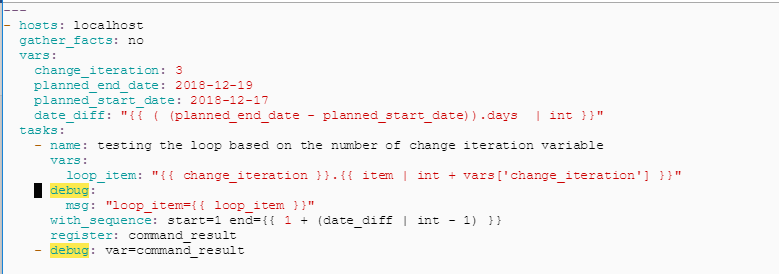
- debug: msg="{{end\_date. stdout}}"

**---- end ----------------------------------------------------------------------------------------------------------------**

**Output**:



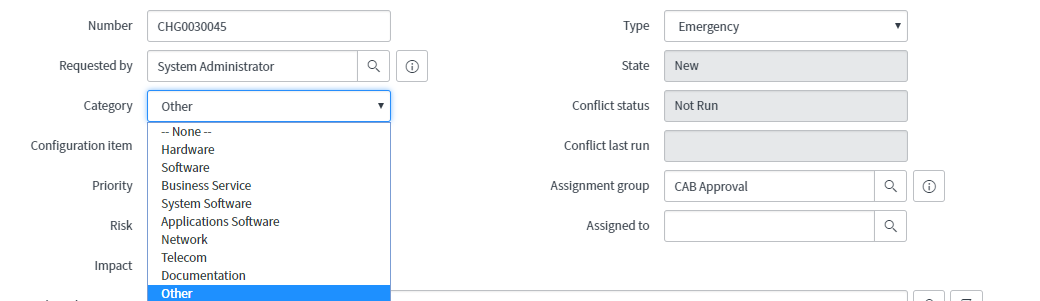
**Snapshot**:



Explanation for the above program:

* Every YAML program starts with three hyphens (---)
* Every new block or tasks starts with hyphen (-)
* We can create any number of tasks or blocks within our program.
* Currently we are working in only one server that is localhost so we mentioned as host as localhost in our program
* As part of our change creation process we have to construct a program and it should loop the number of times and create a number of change requests in SNOW so we calculate it here based on the planned end date minus planned start date and get the number of days.
* Once we get the number of days and will start our loop to perform the operation.
* There is no easier way to achieve the loop here like other programs such as simple for loop concept.
* Here we are using **with\_sequenence** command to loop our program. We can also use **with\_items** command to perform the operation.
* Whenever we use **with\_sequence** or **with\_items** command ansible create a **item** variable automatically
* As part of our requirement we need change start date and end date so that will create a number of changes requests. I have declared two variables in the YAML script and calculate the difference in the dates and store it in **date\_diff** variable.
* Next we are performing the loop using the **with\_sequence** command and perform the operation till loop completes. (based on our number of days’ calculation. **date\_diff**)
* Register command stores our output in the variable.
* Debug command prints our variable output.

**For Testing purpose:** Below is the list of categories.



**List of assignment groups below for testing purpose**:

CAB Approval

Change Management

Database

Network

Software

List of Parameters to be checked at run time:

**Single parameter to playbook**:

ansible-playbook snow\_playbook.yml -e change\_type="Normal"

**multiple parameters example**:

ansible-playbook snow\_playbook.yml -e "change\_type=Normal category=Network"

**multiple parameters example**:

ansible-playbook snow\_playbook.yml -e "change\_type=Normal category=Network group\_name='CAB Approval' planned\_start\_date='12/20/2018 23:00:00' planned\_end\_date='12/23/2018 05:00:00' "

---

- hosts: localhost

tasks:

- name: Executing script to get succeeding dates

shell: |

current\_date=2018-12-31

for increment in {0..3}; do

subsequent\_date=$(date +%d-%m-%Y -d "$current\_date + $increment day")

echo "$subsequent\_date"

done

args:

executable: /bin/bash

register: output

- debug: var=output.stdout\_lines

- local\_action: copy content={{ output.stdout\_lines }} dest=/var/log/log.txt

**Xlrd – module required to read the excel file.**

pip install xlrd

-------------------------------- Filter records from excel file ---------------------------------------------

#!/usr/bin/env python

import pandas as pd

from datetime import datetime as DateTime, timedelta as TimeDelta

from datetime import datetime

from datetime import timedelta

df= pd.read\_excel('Server Inventory.xlsx')

master\_file=pd.read\_excel('2019\_Calendar template.xlsx')

df.set\_index('Patching\_day',inplace=True)

#print(df.loc['Dev\_03'])

print(df.loc['Day\_05'])

date\_1 = DateTime.today()

next\_day = date\_1 + TimeDelta(days=1)

date\_now\_more\_5\_days = (datetime.now() + timedelta(days=1) ).strftime('%Y-%m-%d')

i=1

while i < 5:

date\_now\_more\_5\_days = (datetime.now() + timedelta(days=i) ).strftime('%Y-%m-%d')

print(date\_now\_more\_5\_days)

i += 1

------------------------------- end ----------------------------------------------------------------------------

#!/usr/bin/env python

import pandas as pd

from datetime import date

from datetime import time

from datetime import datetime as DateTime, timedelta as TimeDelta

from datetime import datetime

from datetime import timedelta

try:

now=datetime.now()+ timedelta(days=1)

current\_month=now.strftime("%b")

current\_month="'"+current\_month+"'"

# print(current\_month)

# File declarations - Master and inventory files

df= pd.read\_excel('Server Inventory.xlsx')

master\_file=pd.read\_excel('2019\_Calendar template.xlsx')

# print(master\_file.iloc[1,25])

dt=master\_file.lookup(master\_file[master\_file['Date']==22].index,['Jan'])

# print(dt)

i=1

while i<5:

date\_now\_more\_5\_days = (datetime.now() + timedelta(days=i) ).strftime('%Y-%m-%d')

print(date\_now\_more\_5\_days)

dt=master\_file.lookup(master\_file[master\_file['Date']==24+i].index,['Jan'])

print(dt)

i += 1

# df.set\_index('Patching\_Group',inplace=True)

# second = df.loc["DEV\_21"]

except Exception as e:

print(e)

**Main module** : Creation of change request :

#!/usr/bin/env python

import openpyxl

import yaml

import os

import time

import json

import calendar

from datetime import datetime

try:

#dt = datetime.now()

wb =openpyxl.load\_workbook('ansible\_project/Server\_Inventory\_Linux.xlsx')

source = wb['Sheet1']

user\_data = wb.get\_sheet\_by\_name('Sheet1')

**def create\_change**(date,desc):

stream=open('change\_creation.yml','r')

data=yaml.load(stream)

# print(data)

data[0]['tasks'][0]['snow\_record']['data']['short\_description']=desc

data[0]['tasks'][0]['snow\_record']['data']['start\_date']=date

print(data[0]['tasks'][0]['snow\_record']['data']['short\_description'])

print(data[0]['tasks'][0]['snow\_record']['data']['start\_date'])

stream = open('change\_creation.yml', 'w')

yaml.dump(data,stream, default\_flow\_style=False)

# print(yaml.dump(data))

**for x in range(1,user\_data.max\_row+1):**

#print(str(user\_data[x][3].value)) # ------ Release date---- #

date=str(user\_data[x][3].value) # ------ Date column ------ #

desc="Linux change for Dec 1/31-Date format updated on 12/28 "

print(date)

create\_change(date,desc)

os.system ('ansible-playbook change\_creation.yml')

**except Exception as e**:

print(e)

------------- #####################################################################