4-GalaxyModulesAndDmtfPlaybooks

April 6, 2021

1 Ansible Galaxy Redfish collections

Version 0.130

1.1 Introduction

The Ansible Galaxy community.general collection proposes three Redfish modules:

- Redfish info - Retrieves information like systems and accounts inventories. The exhaustive list of possible commands is in the `CATEGORY_COMMANDS_ALL` array in the module sources.
- Redfish command - Performs `set` operations on log management, user management, and power operations (e.g. on, off, reboot, etc.). The exhaustive list of possible commands is in the `CATEGORY_COMMANDS_ALL` array in the module sources.
- Redfish config - Performs configuration operations on BIOS and other subsystems. The exhaustive list of possible commands is in the `CATEGORY_COMMANDS_ALL` array in the module sources.

The above Ansible Redfish modules are based upon the redfish_utils.py python utility module.

Redfish Ansible Playbook examples using the above modules are present in the DMTF Redfish-Ansible-Playbooks public GitHub repository.

> **Note**: The Ansible Galaxy Redfish modules can be extended using the DMTF Oem extensions instructions. This part is not covered in this workshop.

1.2 Environment preparation

The following cell sets environment variables and checks the connectivity toward the various BMCs used in this notebook.

[1]:
```
########## Environment preparation (Version: 0.133) #############

# Set Student ID number
export stdid=825
Id=$(id --user --name)
NbId=4
InvFile=${NbId}/hosts

# location and ports variables
IloSyBasePort=46000
let iLO5SimulatorBasePort=$IloSyBasePort
```

```
let iLO5SimulatorPort=${iLO5SimulatorBasePort}+${stdid}

iLO5SimulatorIP=ilo5simulators
iLOSimulator=${iLO5SimulatorIP}:${iLO5SimulatorPort}
iLO5SimulatorURI=https://${iLOSimulator}

# Fake Credentials as we are testing against a BMC simulator
OvSsoToken="FakeOvSsoToken"

# Miscellaneous
WorkshopDir=$PWD
HpePythonRedfishVenv="${NbId}/HpePythonRedfishVenv"
export PYTHONPATH="${WorkshopDir}/${NbId}/library/"
w=$(basename $PWD)
alias ResetSimulators="../create-globalbmc.shc.x &>/dev/null; sleep 1"

# Verify we can reach the remote Bmcs on the right HTTPS ports.
for bmc in iLO5Simulator ; do
    ip="${bmc}IP" ; port=$(echo ${bmc}Port)
    nc -vz  $(eval echo "\$${ip}") $(eval echo "\$${port}") &>/dev/null &&
        echo -e "\n\tGood News: $bmc is reachable" \
        || echo "WARNING: Problem reaching $bmc"
done


# Create the Ansible inventory file
cat > ${InvFile} << __EOF__
[OneViewManagedBmcs]
${iLO5SimulatorIP} ansible_port=${iLO5SimulatorPort}

[OneViewManagedBmcs:vars]
ansible_python_interpreter=${WorkshopDir}/${HpePythonRedfishVenv}/bin/python3
ansible_search_path=${HpePythonRedfishVenv}

# Below is a fake session token as we are testing against an iLO 5 simulator.
# In real life, you should populate this variable with the token obtained
# lab 1 of this workshop.
token="${OvSsoToken}"
__EOF__
```

```
        Good News: iLO5Simulator is reachable
```

1.3 Virtual Python environment creation

In order to completely isolate this notebook environment from other notebooks and student environments, it is safer to create a dedicated Python and Ansible virtual environment.

Note: The installation of Ansible in this `Venv` will also install the `community.general` Galaxy collection in your `~/.ansible` personal directory.

NOTE: This `Venv` creation can take up to **2 minutes**. Just wait until message `Finished creating Venv` is displayed.

[2]:
```
# Create Virtual Python environment (Venv)
[ -d ${HpePythonRedfishVenv} ] && rm -rf ${HpePythonRedfishVenv} &>/dev/null
python3 -m venv ${HpePythonRedfishVenv}                          &>/dev/null
source ${HpePythonRedfishVenv}/bin/activate                      &>/dev/null
PS1="[PEXP\[\]ECT_PROMPT>"                                       # Avoid Venv␣
 ↪long prompt messing up outputs

# Install latest Ansible in the Venv
pip install wheel                                                &>/dev/null
pip install jmespath                                             &>/dev/null
pip install ansible                                              &>/dev/null

echo -e "\n\n\tFinished creating Venv\n\n"
```

```
(HpePythonRedfishVenv)

        Finished creating Venv
```

1.3.1 Restart iLO 5 simulator

If you need or desire to restart your iLO 5 simulator to restart this workshop from scratch or for other reasons, run the following cell at any time.

[3]:
```
# iLO 5 Simulator restart
ResetSimulators

# Verify we can reach the remote Bmcs on the right HTTPS ports.
for bmc in iLO5Simulator ; do
    ip="${bmc}IP" ; port=$(echo ${bmc}Port)
    nc -vz  $(eval echo "\$${ip}") $(eval echo "\$${port}") &> /dev/null &&
        echo "$bmc is reachable" \
        || echo "WARNING: Problem reaching $bmc"
done
```

```
iLO5Simulator is reachable
```

1.4 Get and set Redfish properties using Ansible Galaxy collections

This third method of using Ansible for Redfish tasks uses the redfish_info and redfish_command Galaxy collections to get and set the Indicator LEDs of an HPE Synergy compute node and its enclosure. Then, it does the same exercise against a rack-mount server.

The Redfish Ansible collections presented here are very handy in RedHat Tower or AWX environments as they are built-in and can be extended using the DMTF instructions.

1.4.1 Indicator LED management

The next cell launches an Ansible Playbook using an authentication token for performing the same three sections as the previous playbooks:

- Get status of indicator LEDs
- Modify indicator LED(s) status
- Verify that the action was successful.

A convenient way to study the playbook of the next cell is to open it in a different view in this pane. Right click on this Notebook tab name and select `New View for Notebook` to open a new view.

Then, click on this file link.

If you need more space, type `Ctrl-B` (or `Command-B` on a Mac) to hide the left pane. You can have it reappear by hitting `Ctrl-B` again.

Read carefully all the embedded comments in the `.yml` file of the following cell to better understand how the Ansible Galaxy collection works.

[4]:
```
# Set Indicator LED using Galaxy community.general collections against a␣
↪Synergy compute node
ansible-playbook -i ${InvFile} ${NbId}/SetIndicatorLEDUsingGalaxy.yml
```

```
PLAY [OneViewManagedBmcs] ******************************************************

TASK [1.0- redfish_info: Retrieve Indicator LEDs status] ***********************
ok: [ilo5simulators]

TASK [debug] *******************************************************************
ok: [ilo5simulators] => {
    "msg": [
        {
            "ChassisType": "Blade",
            "Id": "1",
            "IndicatorLED": "Off"
        },
        {
            "ChassisType": "Enclosure",
            "Id": "enclosurechassis",
            "IndicatorLED": "Off"
        }
    ]
}

TASK [1.1 Save indicator LEDs values] ******************************************
```

```
ok: [ilo5simulators]

TASK [2.0- redfish_command: SET chassis new indicator LED values] **************
changed: [ilo5simulators] => (item={'Id': '1', 'IndicatorLED': 'Off'})
changed: [ilo5simulators] => (item={'Id': 'enclosurechassis', 'IndicatorLED':
'Off'})

TASK [3.0- Get chassis inventory] **********************************************
ok: [ilo5simulators]

TASK [debug] *******************************************************************
ok: [ilo5simulators] => {
    "msg": [
        {
            "ChassisType": "Blade",
            "Id": "1",
            "IndicatorLED": "Lit"
        },
        {
            "ChassisType": "Enclosure",
            "Id": "enclosurechassis",
            "IndicatorLED": "Lit"
        }
    ]
}

PLAY RECAP *********************************************************************
ilo5simulators             : ok=6    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

1.4.2 Test the same playbook against a rack-mount server

The following cell switches your environment toward an **HPE DL360 Gen10** simulator and then runs again the Ansible Playbook. You will notice that the same playbook works for both a Synergy compute node and a rack-mount server although it is not enclosed in any frame or enclosure chassis.

[5]:
```
# location and ports variables
IloDlBasePort=45000
let iLO5SimulatorBasePort=$IloDlBasePort
let iLO5SimulatorPort=${iLO5SimulatorBasePort}+${stdid}

iLO5SimulatorIP=ilo5simulators
iLOSimulator=${iLO5SimulatorIP}:${iLO5SimulatorPort}
iLO5SimulatorURI=https://${iLOSimulator}

# Adapt the Ansible inventory file
cat > ${InvFile} << __EOF__
```

```
[RackMountBmcs]
${iLO5SimulatorIP} ansible_port=${iLO5SimulatorPort}

[RackMountBmcs:vars]
ansible_python_interpreter=${WorkshopDir}/${HpePythonRedfishVenv}/bin/python3
ansible_search_path=${HpePythonRedfishVenv}
username=Foo
password=Bar
__EOF__


# Set Indicator LED using Galaxy community.general collections against a␣
 ↪rack-mount DL360 Gen10 server
ansible-playbook -i ${InvFile} ${NbId}/SetIndicatorLEDUsingGalaxySessions.yml
```

```
PLAY [RackMountBmcs] ***********************************************************

TASK [1.0- redfish_info: Retrieve Indicator LEDs status] ***********************
ok: [ilo5simulators]

TASK [debug] *******************************************************************
ok: [ilo5simulators] => {
    "msg": [
        {
            "ChassisType": "RackMount",
            "Id": "1",
            "IndicatorLED": "Off"
        }
    ]
}

TASK [1.1 Save indicator LEDs values] ******************************************
ok: [ilo5simulators]

TASK [2.0- redfish_command: SET chassis new indicator LED values] **************
changed: [ilo5simulators] => (item={'Id': '1', 'IndicatorLED': 'Off'})

TASK [3.0- Get chassis inventory] **********************************************
ok: [ilo5simulators]

TASK [debug] *******************************************************************
ok: [ilo5simulators] => {
    "msg": [
        {
            "ChassisType": "RackMount",
            "Id": "1",
```

```
            "IndicatorLED": "Lit"
        }
    ]
}

PLAY RECAP *********************************************************************
ilo5simulators             : ok=6    changed=1    unreachable=0    failed=0
skipped=0    rescued=0    ignored=0
```

1.5 Summary

In this workshop, you used two modules from the `community.general` Ansible Galaxy collection to modify the `IndicatorLED` of an HPE Synergy compute node and its enclosure. This playbooks is generic and also works for rack-mount servers and you could verify this assertion.

The crawling of the Redfish tree to locate the IndicatorLED resources has been mainly performed by the provided Ansible collections. However, you had to extract the properties using the Ansible `yml` syntax in a similar manner as what you did in the second notebook of this workshop.

The Ansible Redfish collections are constantly growing, in terms of new commands. However, if you don't find what you need in the Ansible Redfish collections, you can extend them with your own code as mentioned by the DMTF.

If you are finished, you can go to the next Notebook.