z/OS V2.5 IBM Education Assistant

Solution Name: z/OSMF Ansible Collection Drives Workflow

Solution Element(s): z/OSMF Ansible collection (*ibm_zosmf)*

July 2021

Agenda

- Trademarks
- Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Upgrade & Coexistence Considerations
- Installation & Configuration
- Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.
- Additional Trademarks:
  - None

Objectives

- As a system admin, I wish to use both modern automation technology, such like Ansible, and existing z/OS workflow resources seamlessly at the same time, so that I can benefit from the modern technology and at the same time the previous investment of z/OS workflow won’t be wasted.
- As an application developer, I wish to run an Ansible playbook to deploy software and services on z/OS systems, so that I can maintain both z/OS environment and open-platform environment in a consistent way.

Overview

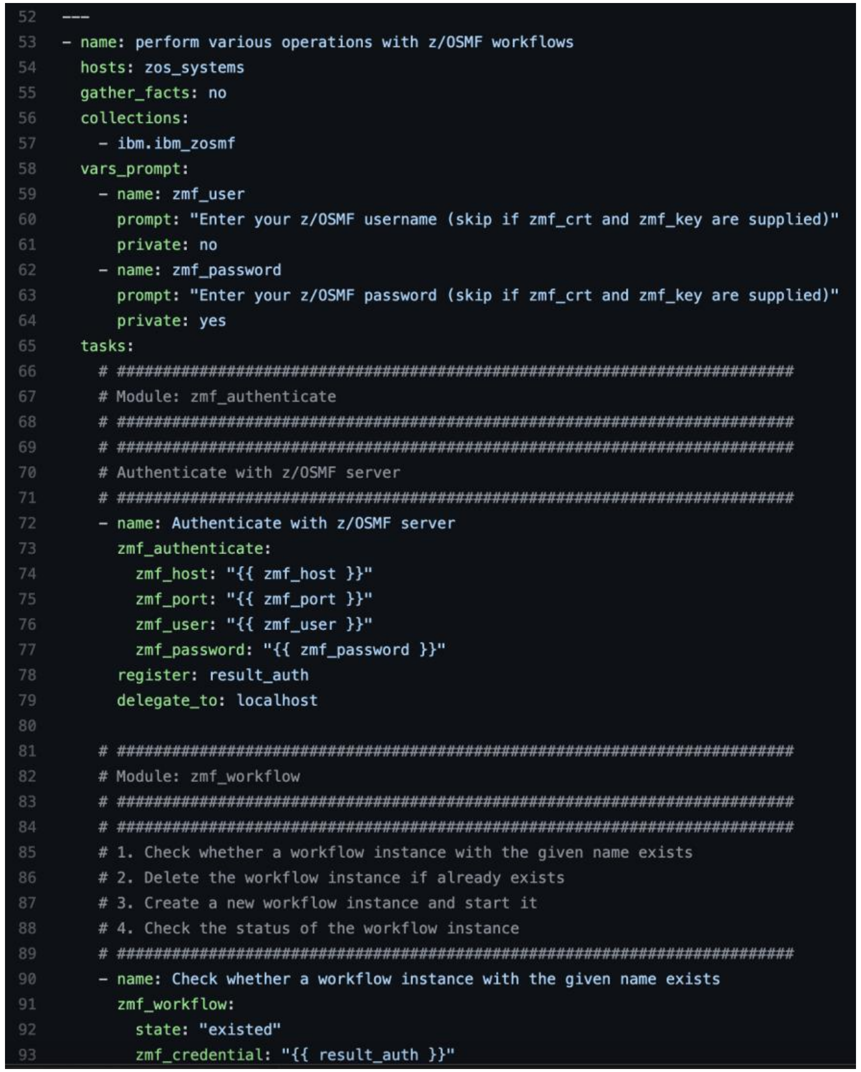
- Who (Audience)
  - System admin, DevOps operator and z/OS application programmers
- What (Solution)
  - z/OSMF Ansible collection supports to operate z/OSMF workflow, including create a workflow instance, perform a workflow instance and delete a workflow instance.
  - z/OSMF Ansible collection supports to provision and manage z/OS software instances (e.g., CICS, DB2, MQ, IMS).
- Wow (Benefit / Value, Need Addressed)
  - Ansible is a very important automation framework in multi-cloud environment, and z/OS as part of the multi-cloud structure, should also support Ansible. Make z/OSMF workflow, which is a modern way to config/deploy software on z/OS, be driven by Ansible, supplies a solution that customers can benefit the ansible framework and reuse the existing workflow resources at the same time.
  - By supporting ansible, we supply flexibility for users to orchestrate their automation work. And by running workflow resource under z/OSMF structure, we supply another solution to limit the Ansible user’s authentication which makes z/OS more secure.

Usage & Invocation – module: zmf_workflow

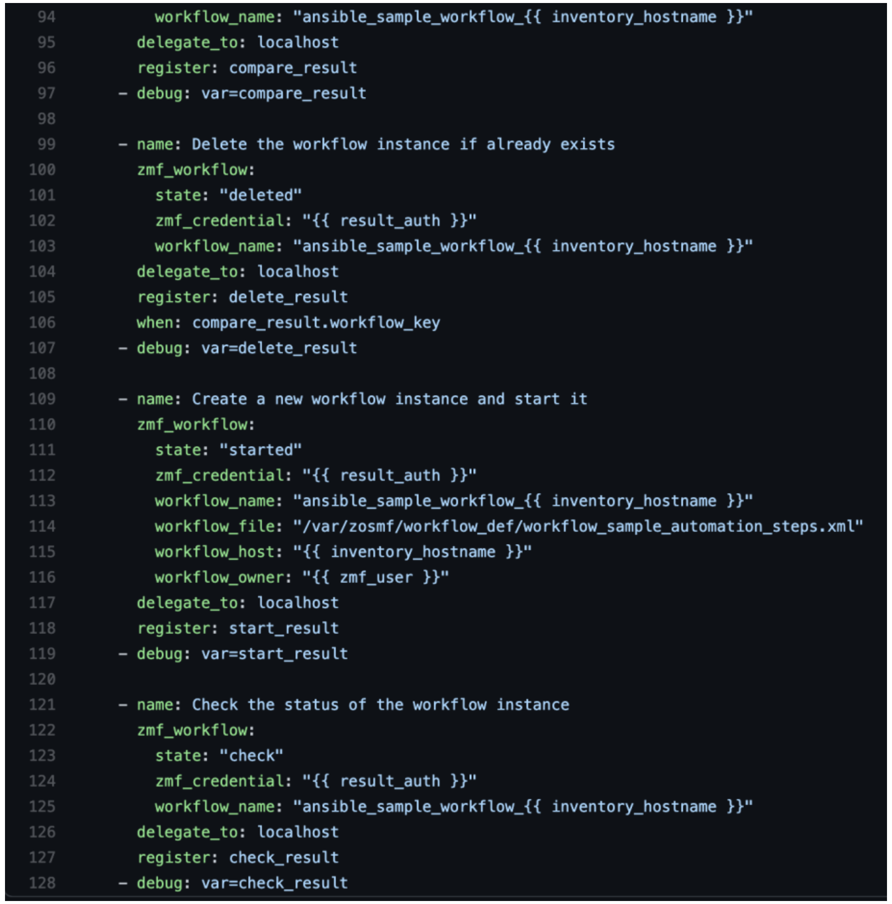
- z/OSMF Ansible collection provides a module ***zmf_workflow*** to operate z/OSMF workflow via z/OSMF workflow RESTful services:
  - This module supports the following final states for working with workflow instances:
    - **existed**: Indicate whether a workflow instance with the given name already exists in the z/OSMF server and has the same definition file, variables and properties.
    - **started**: Create the workflow instance if it does not exist in the z/OSMF server and start it on each of the target z/OS systems.
    - **deleted**: Delete the workflow instance from the z/OSMF server.
    - **check**: Check the status of the workflow instance in the z/OSMF server.

Usage & Invocation – module: zmf_workflow

- Sample playbook: https://github.com/IBM/z_ansible_collections_samples/blob/master/zos_management/zosmf_workflows/workflow_basic.yml



```yaml
---
- name: perform various operations with z/OSMF workflows
  hosts: zos_systems
  gather_facts: no
  collections:
    - ibm.ibm_zosmf
  vars_prompt:
    - name: zmf_user
      prompt: "Enter your z/OSMF username (skip if zmf_crt and zmf_key are supplied)"
      private: no
    - name: zmf_password
      prompt: "Enter your z/OSMF password (skip if zmf_crt and zmf_key are supplied)"
      private: yes
  tasks:
    # ###########################################################################
    # Module: zmf_authenticate
    # ###########################################################################
    # ###########################################################################
    # Authenticate with z/OSMF server
    # ###########################################################################
    - name: Authenticate with z/OSMF server
      zmf_authenticate:
        zmf_host: "{{ zmf_host }}"
        zmf_port: "{{ zmf_port }}"
        zmf_user: "{{ zmf_user }}"
        zmf_password: "{{ zmf_password }}"
      register: result_auth
      delegate_to: localhost

    # ###########################################################################
    # Module: zmf_workflow
    # ###########################################################################
    # ###########################################################################
    # 1. Check whether a workflow instance with the given name exists
    # 2. Delete the workflow instance if already exists
    # 3. Create a new workflow instance and start it
    # 4. Check the status of the workflow instance
    # ###########################################################################
    - name: Check whether a workflow instance with the given name exists
      zmf_workflow:
        state: "existed"
        zmf_credential: "{{ result_auth }}"
```



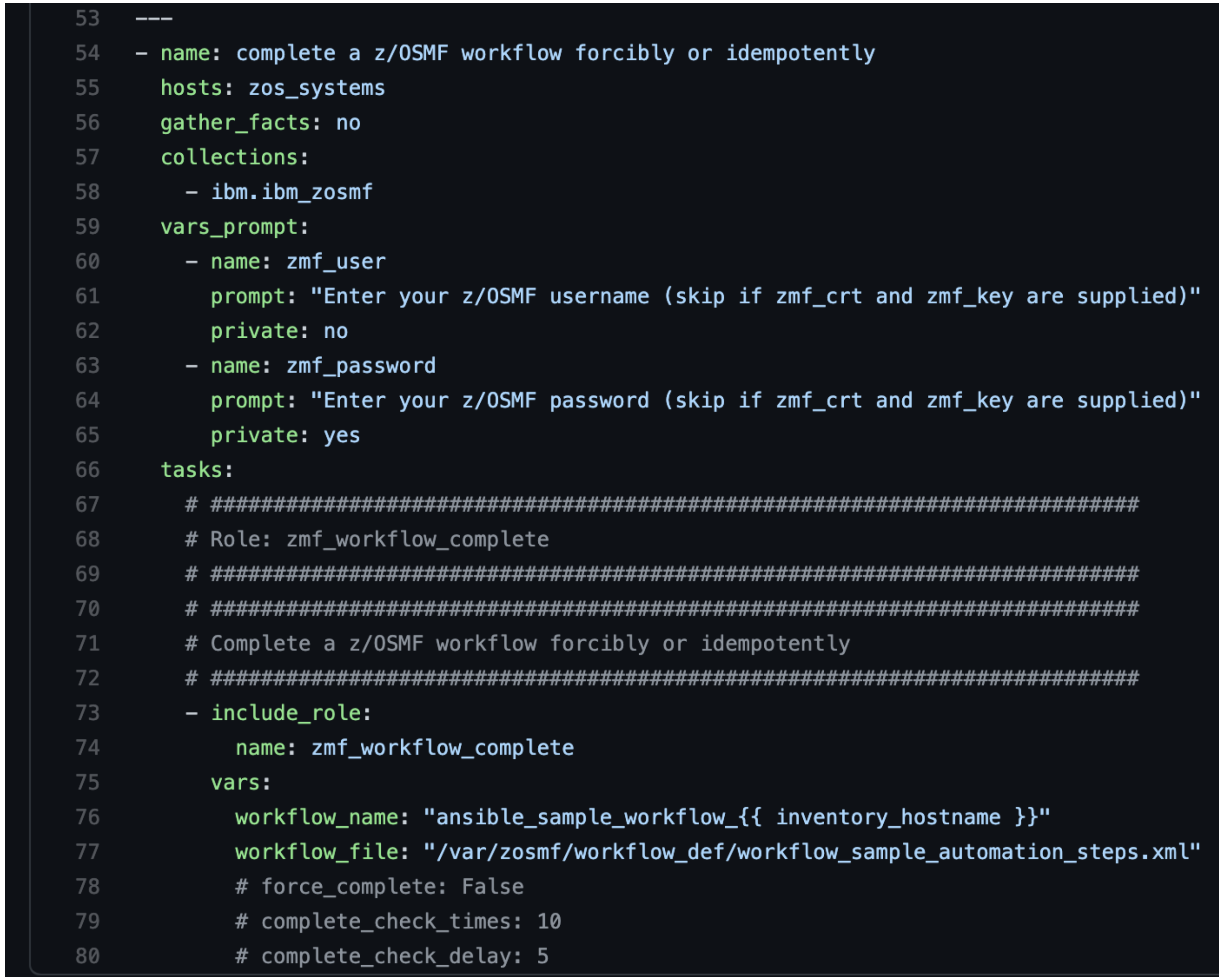
```yaml
        workflow_name: "ansible_sample_workflow_{{ inventory_hostname }}"
      delegate_to: localhost
      register: compare_result
    - debug: var=compare_result

    - name: Delete the workflow instance if already exists
      zmf_workflow:
        state: "deleted"
        zmf_credential: "{{ result_auth }}"
        workflow_name: "ansible_sample_workflow_{{ inventory_hostname }}"
      delegate_to: localhost
      register: delete_result
      when: compare_result.workflow_key
    - debug: var=delete_result

    - name: Create a new workflow instance and start it
      zmf_workflow:
        state: "started"
        zmf_credential: "{{ result_auth }}"
        workflow_name: "ansible_sample_workflow_{{ inventory_hostname }}"
        workflow_file: "/var/zosmf/workflow_def/workflow_sample_automation_steps.xml"
        workflow_host: "{{ inventory_hostname }}"
        workflow_owner: "{{ zmf_user }}"
      delegate_to: localhost
      register: start_result
    - debug: var=start_result

    - name: Check the status of the workflow instance
      zmf_workflow:
        state: "check"
        zmf_credential: "{{ result_auth }}"
        workflow_name: "ansible_sample_workflow_{{ inventory_hostname }}"
      delegate_to: localhost
      register: check_result
    - debug: var=check_result
```

Usage & Invocation – role: zmf_workflow_complete

- z/OSMF Ansible collection provides a role ***zmf_workflow_complete*** to operate z/OSMF workflow by using *zmf_workflow* module:
  - This role is used for completing a workflow instance, either forcibly or idempotently:
    - **forcibly (force_complete: True)**: Delete the workflow instance if it exists in the z/OSMF server. Create a new workflow instance and start it on each of the target z/OS systems. Periodically check the workflow status and return the result when the workflow stops running.
    - **idempotently (force_complete: False)**: Create the workflow instance if it does not exist in the z/OSMF server. Start the workflow on each of the target z/OS systems. Periodically check the workflow status and return the result when the workflow stops running.
  - This role will execute z/OSMF workflows that are located on the target z/OS systems. The workflows must be z/OSMF XML workflows and located on UNIX System Services (USS).

Usage & Invocation – role: zmf_workflow_complete

- Sample playbook: https://github.com/IBM/z_ansible_collections_samples/blob/master/zos_management/zosmf_workflows/workflow_complete.yml



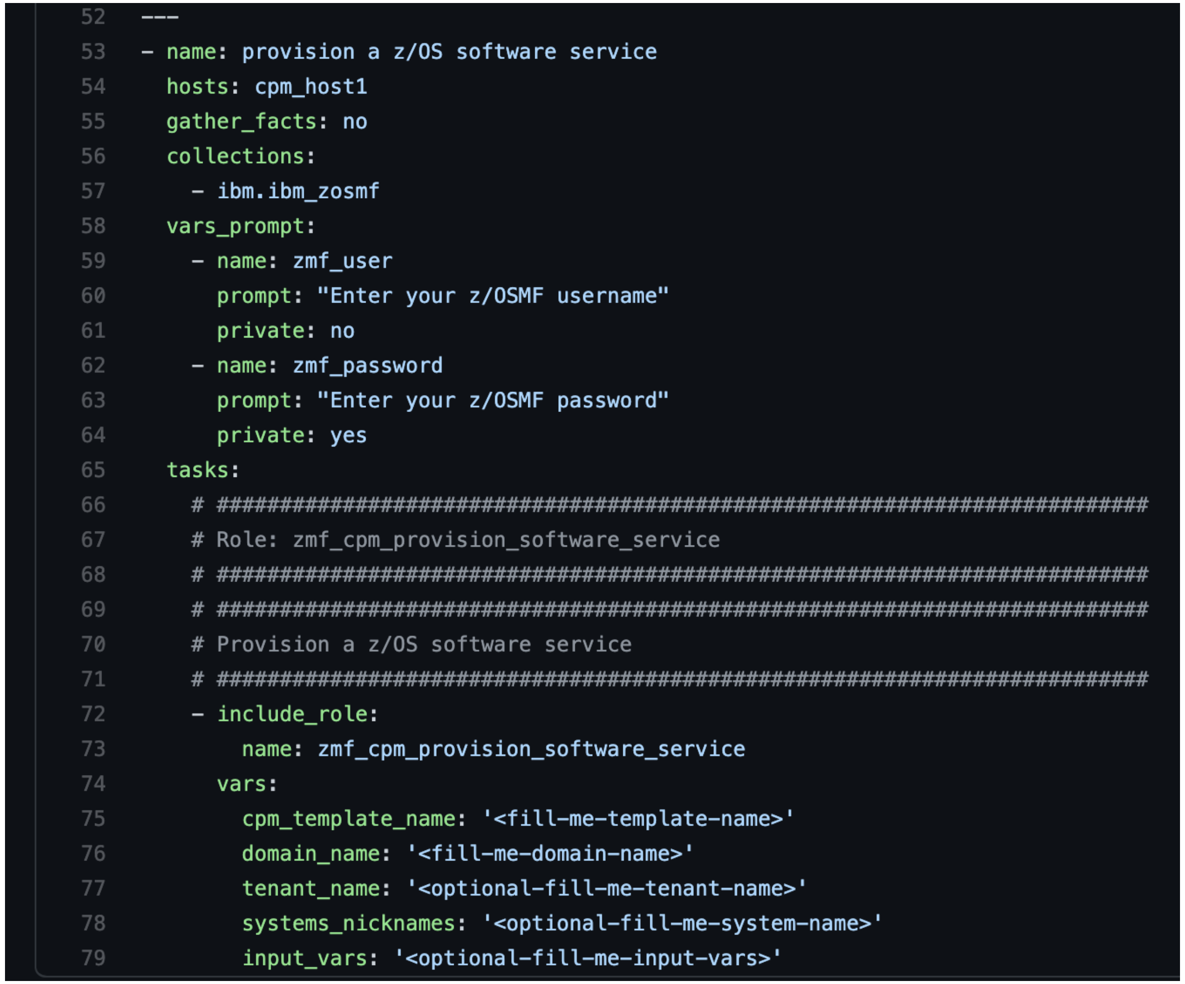
```
---
- name: complete a z/OSMF workflow forcibly or idempotently
  hosts: zos_systems
  gather_facts: no
  collections:
    - ibm.ibm_zosmf
  vars_prompt:
    - name: zmf_user
      prompt: "Enter your z/OSMF username (skip if zmf_crt and zmf_key are supplied)"
      private: no
    - name: zmf_password
      prompt: "Enter your z/OSMF password (skip if zmf_crt and zmf_key are supplied)"
      private: yes
  tasks:
    # ##########################################################################
    # Role: zmf_workflow_complete
    # ##########################################################################
    # ##########################################################################
    # Complete a z/OSMF workflow forcibly or idempotently
    # ##########################################################################
    - include_role:
        name: zmf_workflow_complete
      vars:
        workflow_name: "ansible_sample_workflow_{{ inventory_hostname }}"
        workflow_file: "/var/zosmf/workflow_def/workflow_sample_automation_steps.xml"
        # force_complete: False
        # complete_check_times: 10
        # complete_check_delay: 5
```

Usage & Invocation – role: zmf_cpm_provision_software_service

- z/OSMF Ansible collection provides a role ***zmf_cpm_provision_software_service*** to provision software services via z/OSMF RESTful services provided by Cloud Provisioning and Management (CP&M):
  - This role can be used to provision a z/OS software service using CP&M template.
  - The provision role will create a local record file of instance information that is responded from registry API in json format, and this file will be served to other CP&M roles, such like zmf_cpm*_manage_software_instance* and *zmf_cpm_remove_software_instance*. The local record file is generated in the directory specified via *instance_record_dir* variable defined in a host specific variable files under “host_vars”.

Usage & Invocation – role: zmf_cpm_provision_software_service

- Sample playbook: https://github.com/IBM/z_ansible_collections_samples/blob/master/zos_management/zosmf_cloud_provisioning_and_management/cpm_provision_software_service.yml



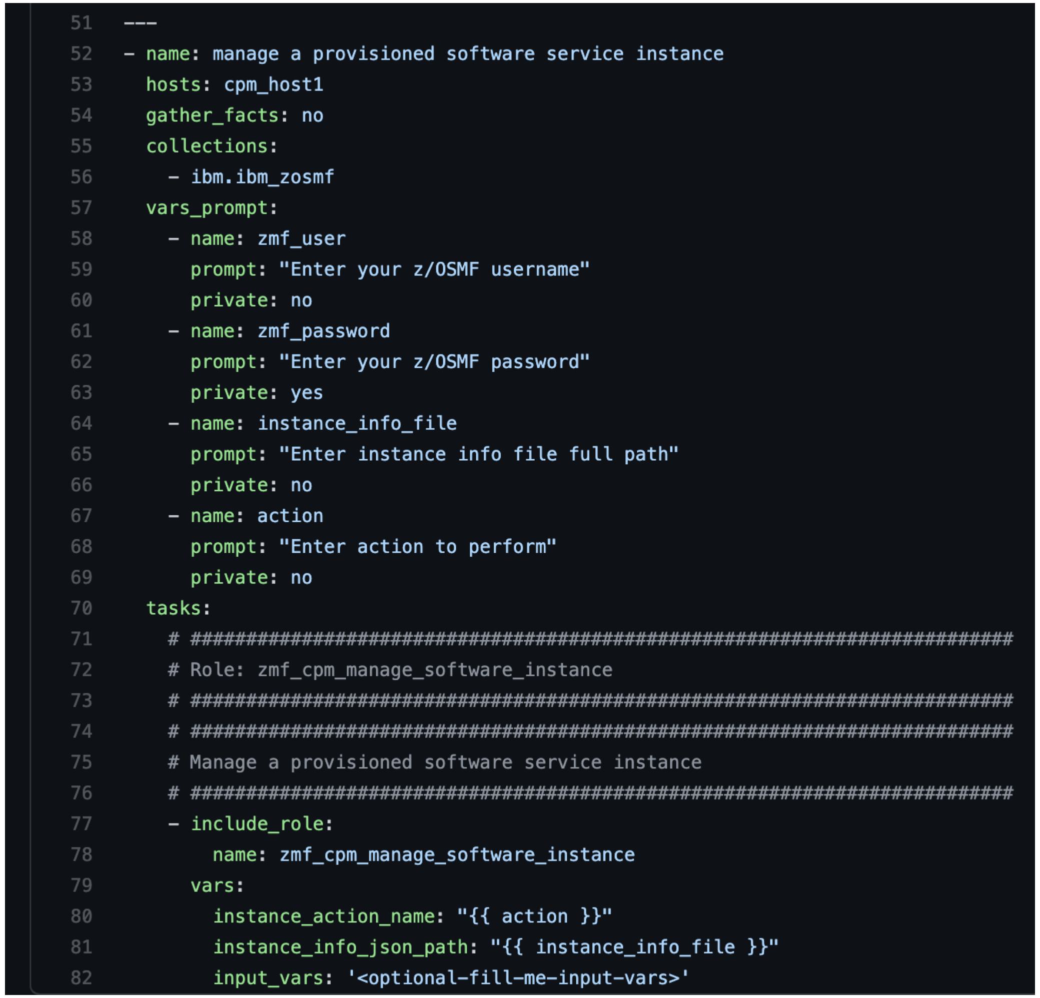
```yaml
---
- name: provision a z/OS software service
  hosts: cpm_host1
  gather_facts: no
  collections:
    - ibm.ibm_zosmf
  vars_prompt:
    - name: zmf_user
      prompt: "Enter your z/OSMF username"
      private: no
    - name: zmf_password
      prompt: "Enter your z/OSMF password"
      private: yes
  tasks:
    # ##########################################################################
    # Role: zmf_cpm_provision_software_service
    # ##########################################################################
    # ##########################################################################
    # Provision a z/OS software service
    # ##########################################################################
    - include_role:
        name: zmf_cpm_provision_software_service
      vars:
        cpm_template_name: '<fill-me-template-name>'
        domain_name: '<fill-me-domain-name>'
        tenant_name: '<optional-fill-me-tenant-name>'
        systems_nicknames: '<optional-fill-me-system-name>'
        input_vars: '<optional-fill-me-input-vars>'
```

Usage & Invocation – role: zmf_cpm_manage_software_service

- z/OSMF Ansible collection provides a role ***zmf_cpm_manage_software_service*** to manage software services via z/OSMF RESTful services provided by Cloud Provisioning and Management (CP&M):
  - This role can be used to manage a provisioned software service instance.
  - Various actions can be performed on a provisioned instance using this role, such as starting, stopping or deprovisioning the instance. Actions that can be performed on a provisioned instance are described in local record file associated with the provisioned instance. The name variable in actions array under registry-info identifies various actions that can be performed on the instance.

Usage & Invocation – role: zmf_cpm_manage_software_service

- Sample playbook: https://github.com/IBM/z_ansible_collections_samples/blob/master/zos_management/zosmf_cloud_provisioning_and_management/cpm_manage_software_instance.yml



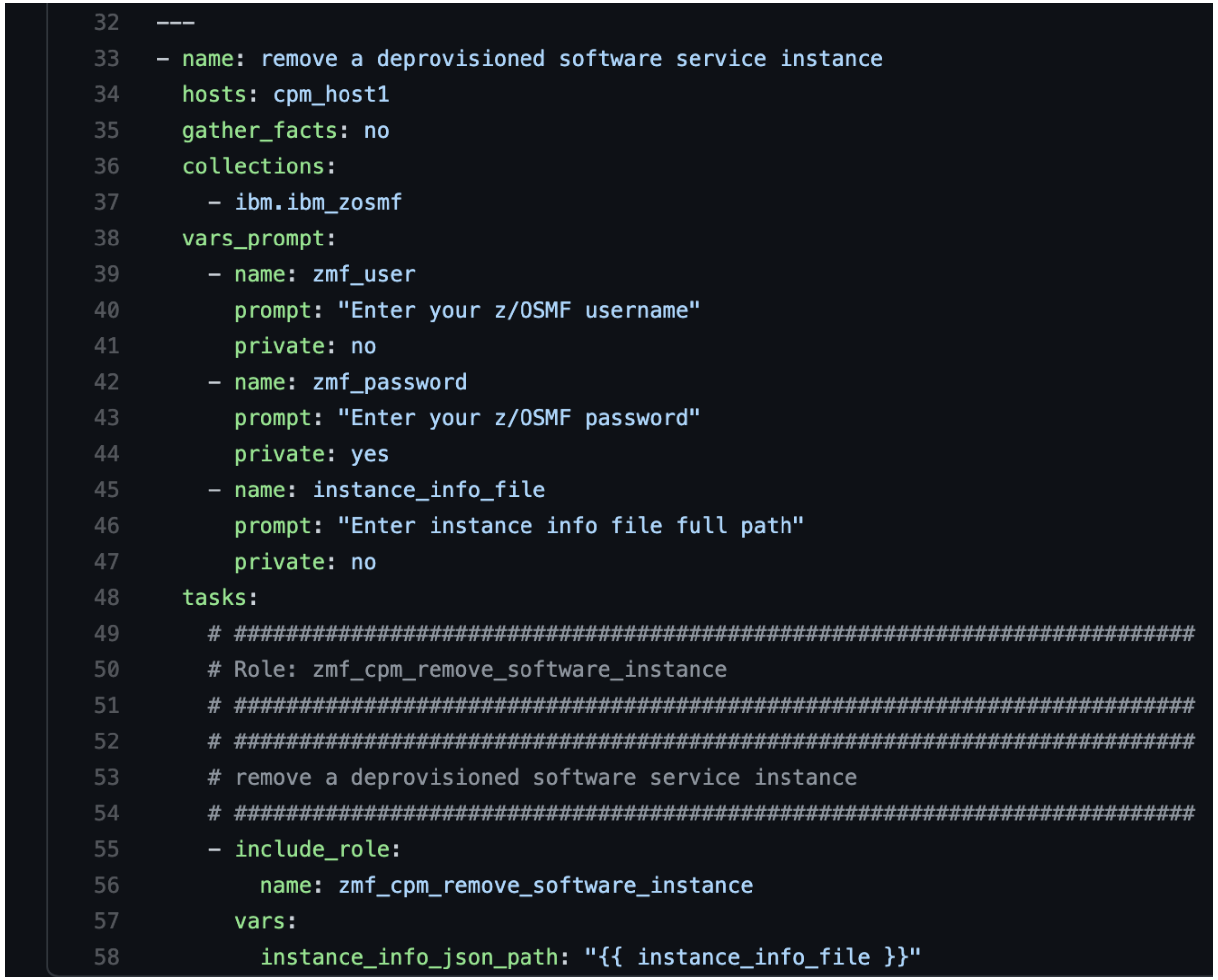
```yaml
---
- name: manage a provisioned software service instance
  hosts: cpm_host1
  gather_facts: no
  collections:
    - ibm.ibm_zosmf
  vars_prompt:
    - name: zmf_user
      prompt: "Enter your z/OSMF username"
      private: no
    - name: zmf_password
      prompt: "Enter your z/OSMF password"
      private: yes
    - name: instance_info_file
      prompt: "Enter instance info file full path"
      private: no
    - name: action
      prompt: "Enter action to perform"
      private: no
  tasks:
    # ##########################################################################
    # Role: zmf_cpm_manage_software_instance
    # ##########################################################################
    # ##########################################################################
    # Manage a provisioned software service instance
    # ##########################################################################
    - include_role:
        name: zmf_cpm_manage_software_instance
      vars:
        instance_action_name: "{{ action }}"
        instance_info_json_path: "{{ instance_info_file }}"
        input_vars: '<optional-fill-me-input-vars>'
```

Usage & Invocation – role: zmf_cpm_remove_software_service

- z/OSMF Ansible collection provides a role ***zmf_cpm_remove_software_service*** to remove software services via z/OSMF RESTful services provided by Cloud Provisioning and Management (CP&M):
  - This role can be used to remove a de-provisioned software service instance.

Usage & Invocation – role: zmf_cpm_remove_software_service

- Sample playbook: https://github.com/IBM/z_ansible_collections_samples/blob/master/zos_management/zosmf_cloud_provisioning_and_management/cpm_remove_software_instance.yml



```yaml
---
- name: remove a deprovisioned software service instance
  hosts: cpm_host1
  gather_facts: no
  collections:
    - ibm.ibm_zosmf
  vars_prompt:
    - name: zmf_user
      prompt: "Enter your z/OSMF username"
      private: no
    - name: zmf_password
      prompt: "Enter your z/OSMF password"
      private: yes
    - name: instance_info_file
      prompt: "Enter instance info file full path"
      private: no
  tasks:
    # ##########################################################################
    # Role: zmf_cpm_remove_software_instance
    # ##########################################################################
    # ##########################################################################
    # remove a deprovisioned software service instance
    # ##########################################################################
    - include_role:
        name: zmf_cpm_remove_software_instance
      vars:
        instance_info_json_path: "{{ instance_info_file }}"
```

Interactions & Dependencies

- Software Dependencies
  - z/OSMF Workflow Services.
  - Cloud Provisioning and Management (CP&M) Services.
- Hardware Dependencies
  - N/A
- Exploiters
  - N/A

Upgrade & Coexistence Considerations

- To exploit this solution, all systems in the Plex must be at the new z/OS level:
  - No
- List any toleration/coexistence APARs/PTFs.
  - N/A
- List anything that doesn't work the same anymore.
  - N/A
- Upgrade involves only those actions required to make the new system behave as the old one did.
  - N/A
- Coexistence applies to lower level systems which coexist (share resources) with latest z/OS systems.
  - N/A

Installation & Configuration

- z/OSMF Ansible collection can be installed from Ansible Galaxy: https://galaxy.ansible.com/ibm/ibm_zosmf

Summary

- The following z/OS V2R5 item has been explained:

  Ansible drives workflow

Appendix

- Website: https://ibm.github.io/z_ansible_collections_doc/index.html