ansible-satellite

This repository includes Ansible playbooks and roles for configuring a Red Hat satellite system. Currently it's mainly used to configure satellite for supporting SAP on RHEL workflows

Included roles

- *sap_satellite_configure*: Configure a newly install satellite system for SAP usage
- *sap_satellite_update*: Update and/or promote a given composite content view and all component content views
- *tower_integration*: Integrate *sap_satellite_configure* into a given Ansible Tower instance

Role: sap_satellite_configure

This role creates the following Satellite objects to further automate the deployment and patching of hosts running SAP:

- Configure a global http proxy if *sap_satellite_configure_http_proxy* is defined
- Creates a Satellite organization
- Creates required repositories to configure hosts running SAP. For example for RHEL 7 these currently are
  - rhel-7-server-e4s-rpms
  - rhel-ha-for-rhel-7-server-e4s-rpms
  - rhel-sap-for-rhel-7-server-e4s-rpms
  - rhel-sap-hana-for-rhel-7-server-e4s-rpms
- Creates a weekly sync plan for the repositories mentioned above
- Creates and publishes an initial version of Satellite content views for managing the repositories above, these are
  - CV: RHEL for SAP Server
  - CV: RHEL for SAP High Availability
  - CV: RHEL for SAP Applications
  - CV: RHEL for SAP HANA
- A composite content view

Variables

- *sap_satellite_configure_rhel_release*: The RHEL Release to use for SAP (default 7.6)

- *sap_satellite_configure_architeture*: RHEL architecture (default x86_64)
- *sap_satellite_sat_subscription_name*: The name of the subscription to use (default SKU)

Role sap_satellite_update

This role updates and promotes the composite content view defined via the variable *satellite_ccv_name* (see Variables shared between all roles).

The role implements the following workflow

1. We would like to create a new version of the composite content view defined in _satellite_ccv_name.
2. We use the role with *sap_satellite_update_promote* set to false to create a new version in the lifecycle environment **Library**
3. Now we can promote the new version from the lifecycle environment **Library** to for example **DEV** by setting
    - *sap_satellite_update_promote* to *true*
    - *sap_satellite_update_src_environment* to *Library*
    - *sap_satellite_update_dest_environment* to *DEV*

Variables

- *sap_satellite_update_promote*: When true update the CCV defined via *satellite_ccv_name* and all content views within the CCV. If false promote the CCV in lifecycle environment *sap_satellite_update_dest_environment* to the version of *sap_satellite_update_src_environment*
- *sap_satellite_update_src_environment*: Take the version of this environment and promote *sap_satellite_update_dest_environment* to this version.
- *sap_satellite_update_dest_environment*: The environment we would like to update to the version defined in _sap_satellite_update_src_environment.

Variables shared between all roles

- *lifecycle_environments*: contains a list of Satellite lifecycle environments to create

  e.g.

  ```
          lifecycle_environments:
            - name: DEV
              prior: Library
            - name: UAT
              prior: DEV
            - name: PROD
              prior: UAT
  ```

- *satellite_username*: Username for connecting to Satellite (default: admin)
- *satellite_password*: Password for connecting to Satellite (default: admin)
- *satellite_url*: Satellite URL (default: http://localhost)
- *satellite_organization*: Organization for creating content views (default: Default)
- *satellite_ccv_name*: Name of the Composite Content View to create (default: CCV: RHEL for SAP)