F5 Professional Services

Automation Practices

Lab Guide

Version 1.0

THE WORLD RUNS BETTER WITH F5

Table of Contents

Executive Summary

Labs focusing on typical F5 product OAM Automation practices in the field, including:

- Sample F5 BIG-IP devices OAM tasks (initial config, software upgrade, etc.)
- Sample F5 BIG-IQ OAM tasks (devices discover/import, license management, etc.)

Notes:

- Lab environment setup and installation details are not described. Assumed that the user has certain degree of familiarity working in F5 UDF environment.
- No use case in the labs related to public/private cloud scenarios.
- Assumed the BIG-IP device pair is in normal A/S topology, not A/A (vCMP, etc.).
- As "practices in the field", the labs might not follow best practices/standardized methodologies recommended by F5 PD/Support.

Required hardware and software

- F5 hardware/VE
  - UDF lab course: Automation_Practices
- F5 software
  - BIG-IP 12.1+
  - BIG-IQ 6.1+
- 3rd party software
  - Ansible Tower 3.x/Ansible Engine 2.7x +

Components in UDF

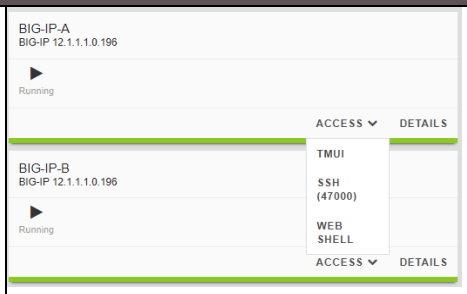
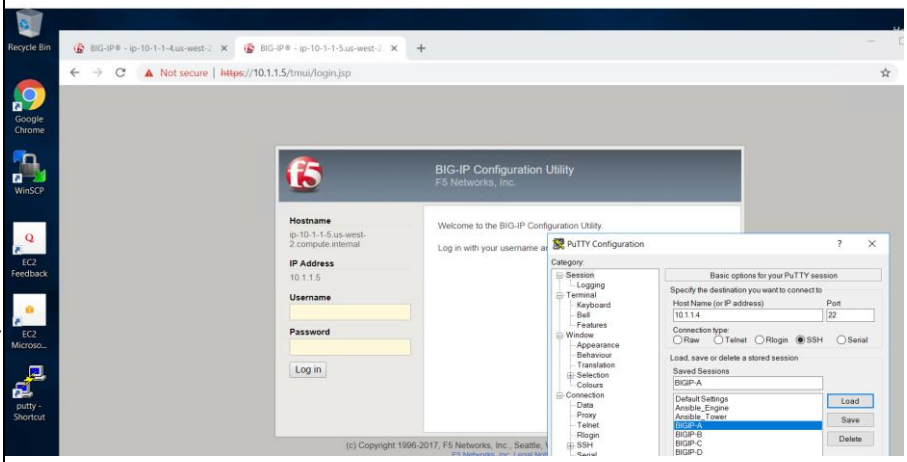
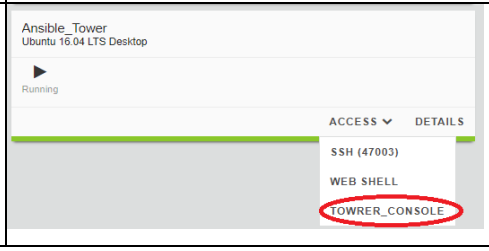
- 1 Windows Client
  - RDP – Administrator/rdyGXoHr
- 1 Ansible Tower
  - GUI – admin/admin
  - CLI – root/default
- 1 Ansible Engine
  - User – root/default
- 4 BIG-IPs
  - GUI – admin/admin
  - CLI – root/default
- 2 BIG-IQs
  - GUI – admin/admin
  - CLI – root/default

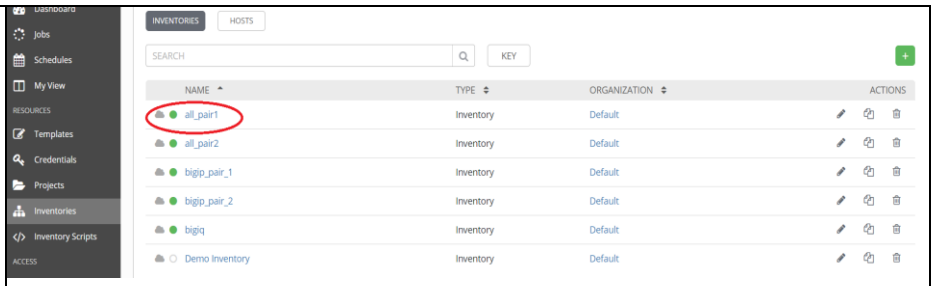
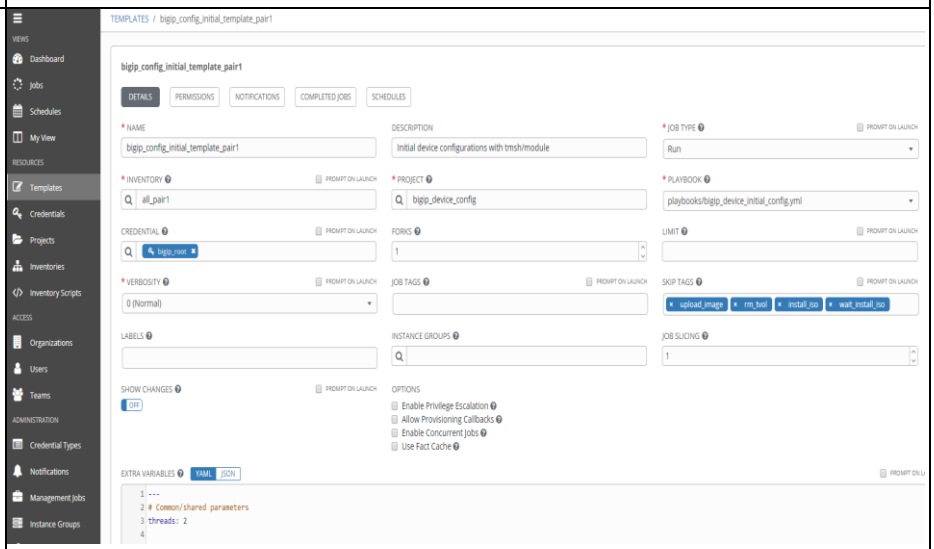
Lab 1 BIGIP New Device Configuration Exercises

Sample new F5 Physical/VE BIG-IP devices initial configurations automations.

Note: F5 Ansible Modules are not used in the scenarios, as which have been widely employed and familiar to the users already.

Scenario 1 - New BIGIP Device Configuration (TMSH)

Perform new device pair configurations (Ansible Tower)	
Open BIG-IP-A, BIG-IP-B and BIG-IQ-2 TMUI Consoles in UDF lab Login the BIG-IP devices GUI with credential: admin/admin, and confirm no object configured. Login the BIG-IQ device GUI with credential: admin/admin, and confirm no device added and licensed. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab, and access the BIG-IP devices Web GUIs and CLIs* • *In case the BIG-IP devices already configured, or user needs to re-try the lab, following the "Reset Lab BIG-IP Device Configurations" procedure in Appendix.*	 
Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	

Click the "Inventories" menu under "Resources" in the left panel. Locate and select the "all_pair1" from the list, which includes the target new BIG-IP device pair (BIG-IP A & B) and BIG-IQ-2 used by this scenario. *Notes:* • *Pay attention to the parameters defined at inventory level and group level.*	
Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list: bigip_config_initial_template_pair1 Click to open the job template. Review the template configurations and parameters. *Notes:* • *User can clone a new job template by copying this existing sample template, then modify/personalize the new template as needed.* • *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template.* device_init_config_var s.xlsx *The variables in the spreadsheet only covers sample components for new devices configurations.*	
Trigger the job by clicking on the Start icon in the right hand-side of the selected job template. The job status window will open.	

Monitor the job execution status in the right hand-side status window.

Review each task executed in the playbook.

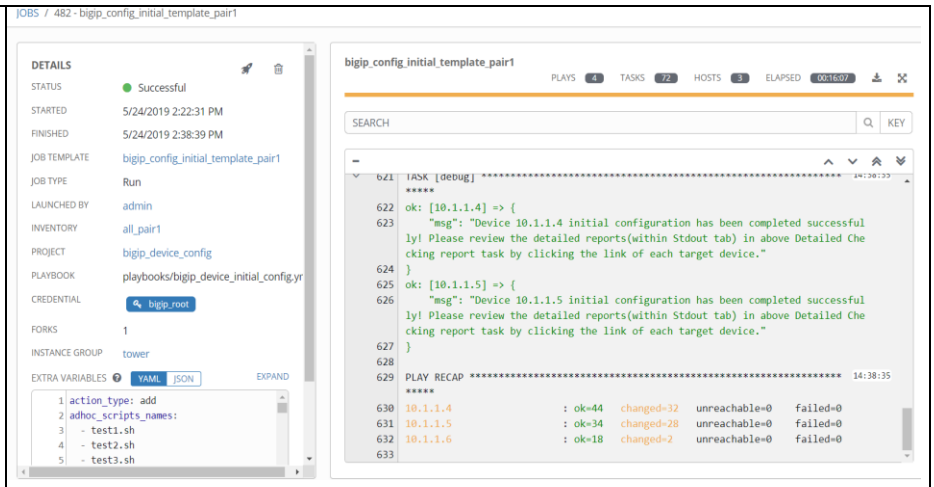
After the job completed successfully:

(1) Login the target BIG-IP device pair GUIs, confirm the following device basic configuration and application objects on the target devices:

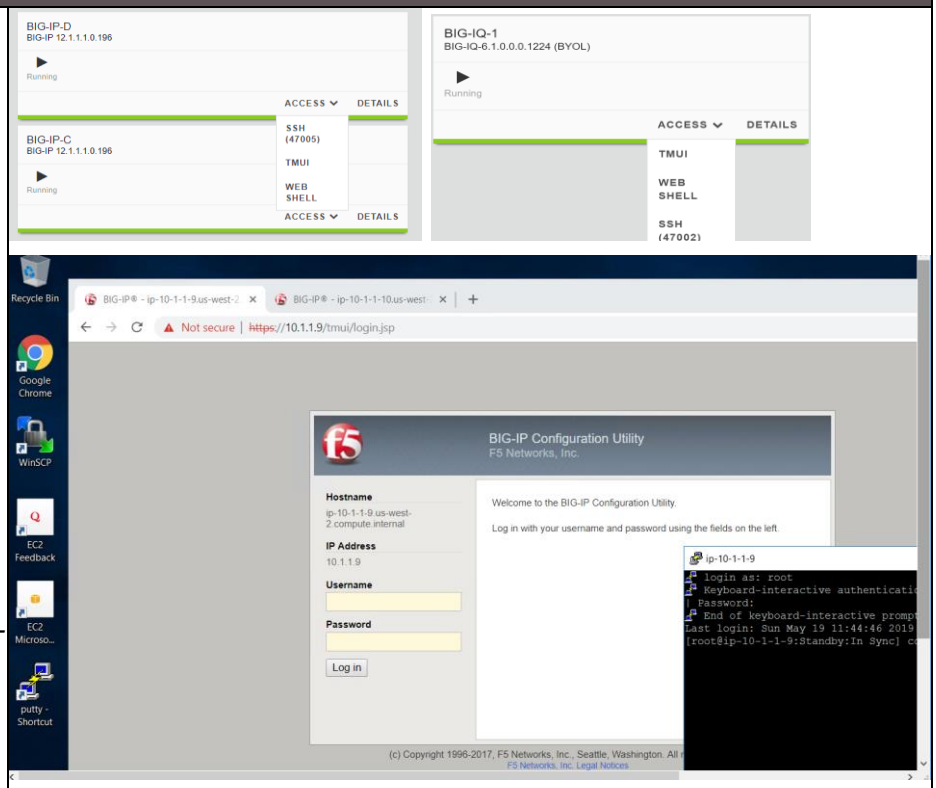
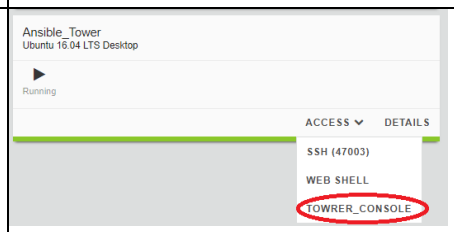
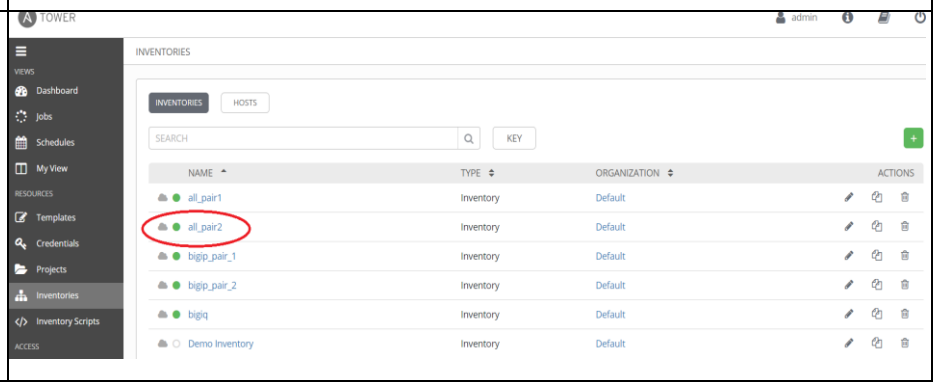
- System (Configuration, Provisioning, Platform, Software, High Availability, SNMP)
- Network (Route Domains, Routes, VLANs, Self IPs)
- Device Management (sync status, etc.)
- LTM components under "Lab" partition (virtual servers, pools, profiles, monitors)

(2) Login the BIG-IQ-2 GUI, confirm:

- BIG_IP devices are added under "Devices".
- BIG-IP devices are licensed with the reg keys in the license pool.
- BIG-IP devices services (LTM, DNS) are discovered and imported.



Scenario 2 - New BIGIP Device Configuration (DO)

Perform new device pair configurations with DO (Ansible Tower)	
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab Login the BIG-IP devices GUI with credential: admin/admin, and confirm no object configured. Login the BIG-IQ device GUI with credential: admin/admin, and confirm no device added and licensed. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab, and access the BIG-IP devices Web GUIs and CLIs* • *In case the BIG-IP devices already configured, or user needs to re-try the lab, following the "Reset Lab BIG-IP Device Configurations" procedure in Appendix.*	
Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	
Click the "Inventories" menu under "Resources" in the left panel. Locate and select the "all_pair2" from the list, which includes the target new BIG-IP device pair (BIG-IP C & D) and BIG-IQ used by this scenario. *Notes:* • *Pay attention to the parameters defined at inventory level and group level.*	

Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list:

bigip_config_initial_do_template_pair2

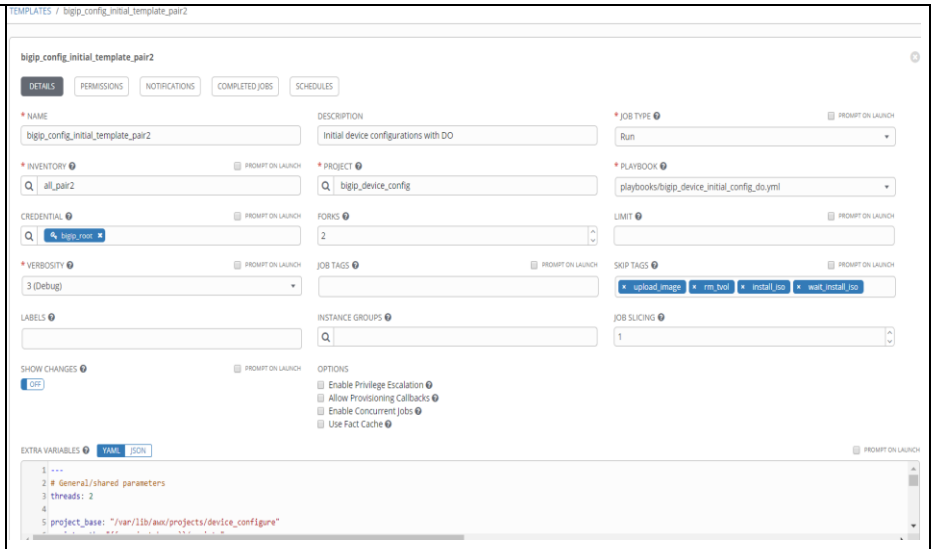
Click on & open the job template. Review the template configurations and parameters in "Extra Variables" window.

Notes:

- *User could clone a new job template by copying the existing sample template, then modify/personalize the new template as needed.*
- *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template.*

device_do_onboard_vars.xlsx

The variables in the spreadsheet only cover sample components for new devices configurations.



Trigger the job by clicking on the Start icon in the right hand-side of the selected job template. The job status window will open.

Monitor the job execution status in the right hand-side status window.

Review message for each task executed by the playbook.

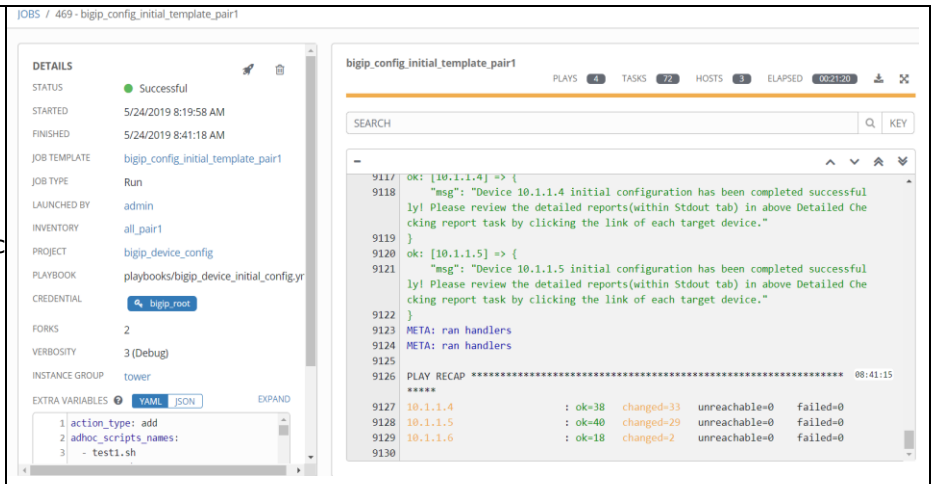
After the job completed successfully:

(1) Login the target BIG-IP device pair GUIs, confirm the following basic configuration objects on the target devices:

- System (Configuration, Provisioning, Platform, Software)
- Network (VLANs, Self IPs)
- Device Management (sync status, etc.)
- LTM components under tenant partition (virtual servers, pools)
- iApps -> Package Management LX: f5-declarative-onboarding, f5-appsvcs

(2) Login the BIG-IQ-1 GUI, confirm:

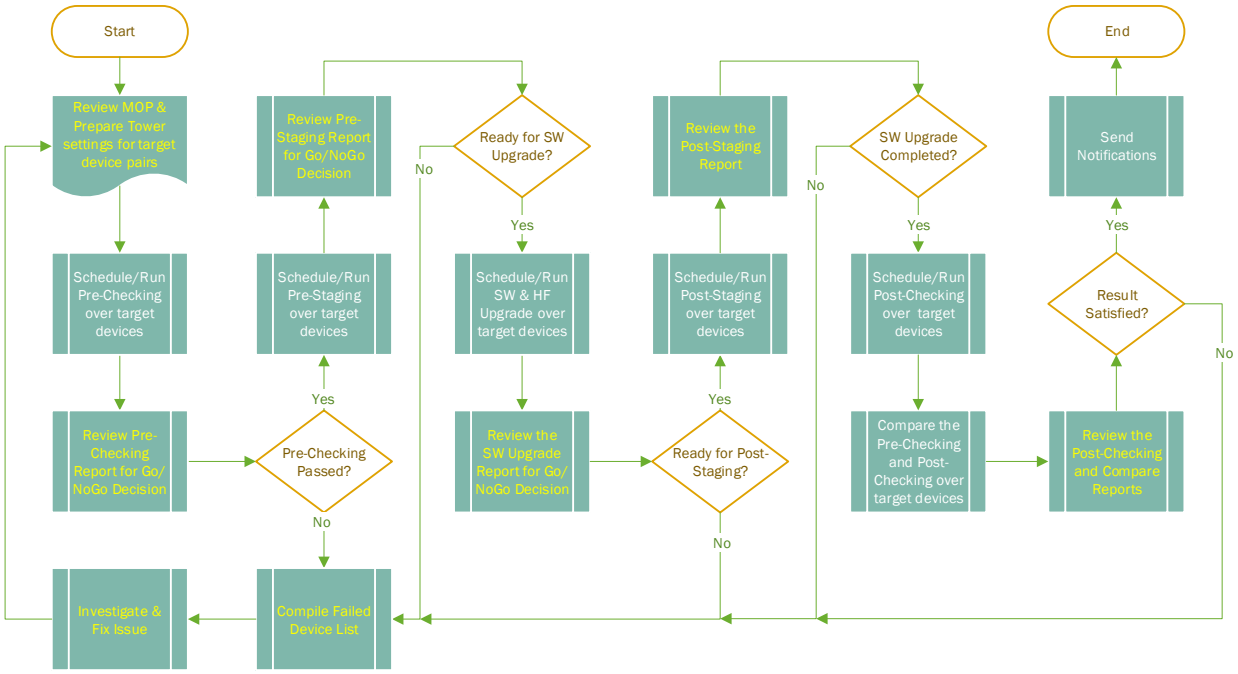
- BIG_IP devices are added under "Devices".
- BIG-IP devices are licensed with the reg keys in the license pool.
- BIG-IP devices services are discovered and imported.



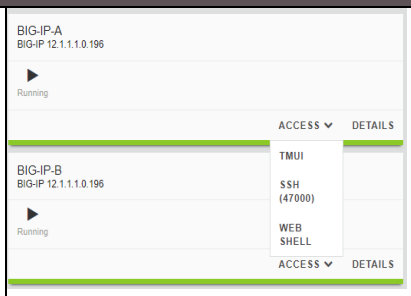
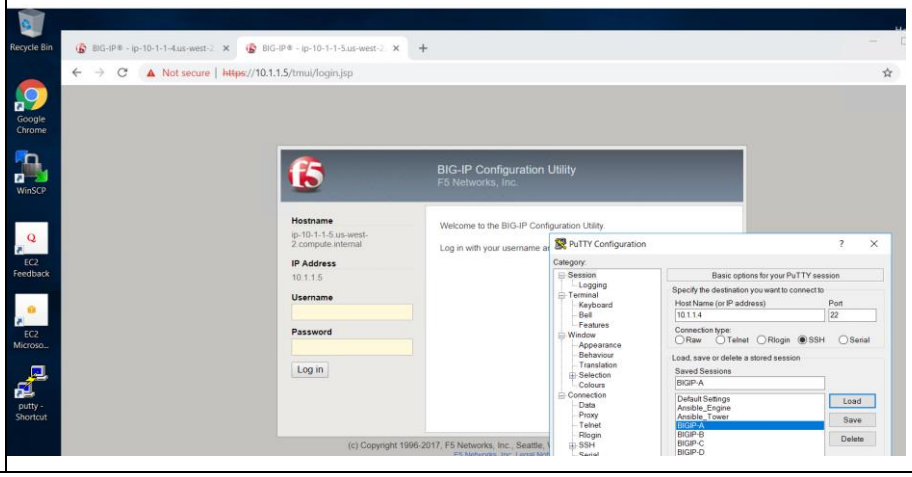
Note: Estimated time for the lab is ~ 20 minutes without software upgrade, and ~30 minutes with software upgrade.

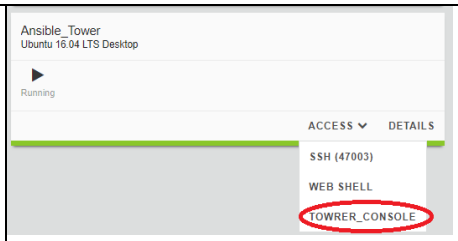
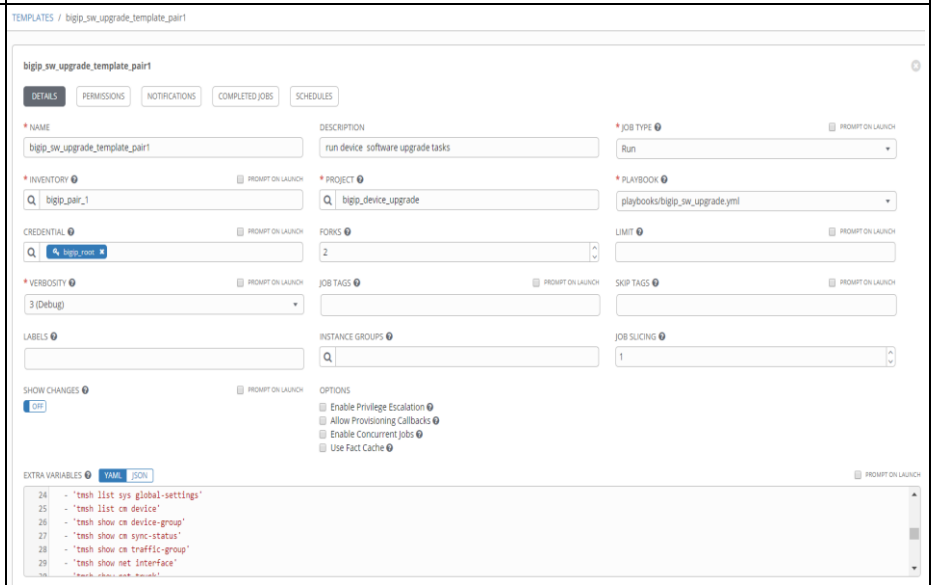
Lab 2 BIGIP Device SW Upgrade Exercises

Sample F5 Physical/VE BIG-IP devices software upgrade procedure automations.



Scenario 1 – BIGIP Device Software Upgrade (TMSH)

Perform device pair software upgrade (Ansible Tower)	
Open BIG-IP-A and BIG-IP-B TMUI Consoles in UDF lab Login the BIG-IP devices GUI with credential: admin/admin, and confirm the device pair has already been configured by Lab1 scenario 1. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab, and access the BIG-IP devices Web GUIs and CLIs*	 

Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	
Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list: bigip_sw_upgrade_template_pair1 Click on & open the job template. Review the template configurations and parameters. *Notes:* • *User can clone a new job template by copying the existing sample template, then modify/personalize the new template as needed.* • *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template.*  device_sw_upgrade_vars.xlsx *The variables in the spreadsheet only cover sample components for new devices configurations*	
Trigger the job by clicking on the Start icon in the right hand-side of the selected job template. The job status window will open.	

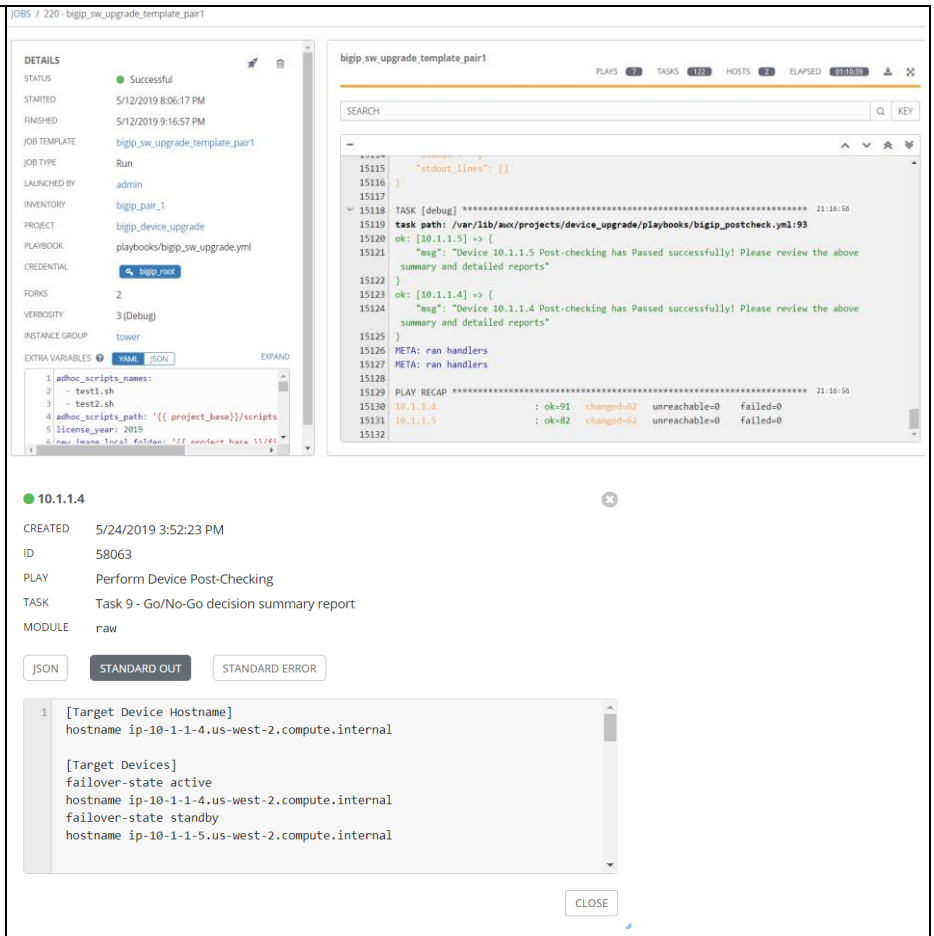
Monitor the job execution status in the right hand-side status window.

Review each task executed in the playbook.

After the job completed successfully, login the target BIG-IP device pair GUIs, confirm the software has been upgraded on the devices.

Note:

- *The procedure includes several sub plays, review the intermediate report (in 'Standard Out' tab) by clicking each target device link at the end of each sub play for status.*
- *User can cancel the job during the "pause" interval between the sub plays, the next sub play would not execute.*

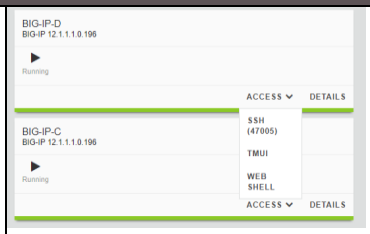
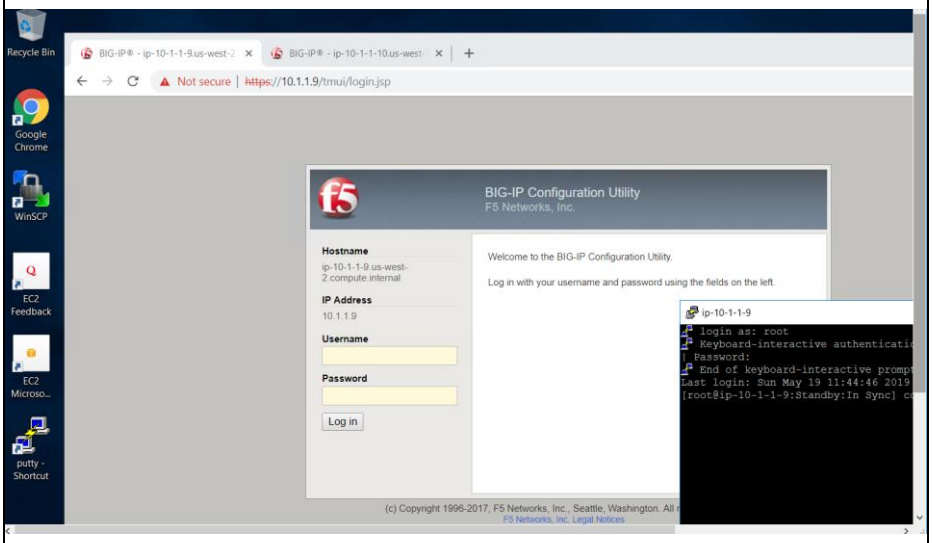
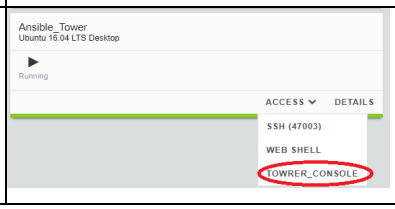


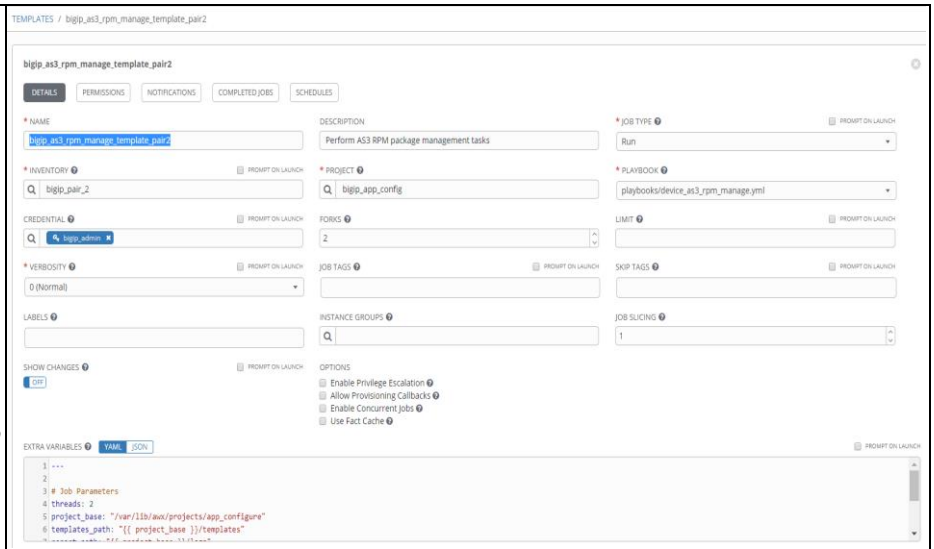
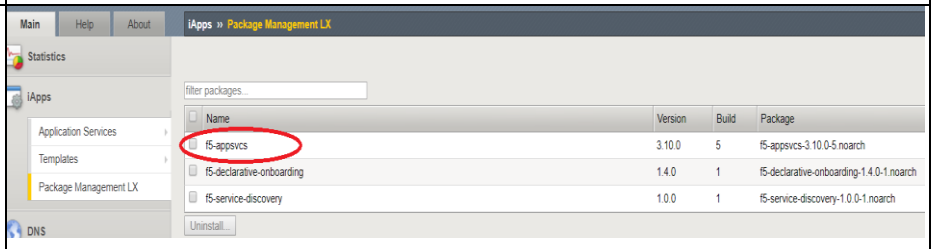
Note: Estimated time for the lab is 60~70 minutes, with one device upgraded at a time, considering 0 offline time in real production case.

Lab 3 AS3 Exercises

Sample applications declarative configurations with AS3.

Scenario 1 - Application DO with AS3 (Direct)

Perform application configurations with AS3 directly to BIGIP (Ansible Tower)	
Open BIG-IP-C and BIG-IP-D TMUI Consoles in UDF lab. Login the BIG-IP devices GUIs with credential: admin/admin, and confirm the basic objects are already configured by Lab1 Scenario 2. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab and access the BIG-IP devices Web GUIs and CLIs.*	 
Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	

(Optional) Click the "Templates" menu under "Resources" in the left panel. Locate the follow sample job template from the list: bigip_as3_rpm_manage_template_pair2 Click on & open the job template. Review the template configurations and parameters. *Notes:* • *This job is optional and only required once to install the AS3 rpm package or upgrade the rpm package onto target devices.* • *User can create a new job template by copying the existing sample template, then modify/personalize the new template as needed.* • *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template.* device_as3_rpm_vars.xlsx	
(Optional) Trigger the job by clicking on the Start icon in the right hand-side of the selected job template. The job status window will open.	
(Optional) Monitor the job execution status in the right hand-side status window. Review each task executed in the playbook. After the job completed successfully, login the target BIG-IP device pair GUIs or CLIs, confirm all the AS3 package has been installed on the devices.	

Click the "Templates under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list:

bigip_as3_app_config_template_pair2

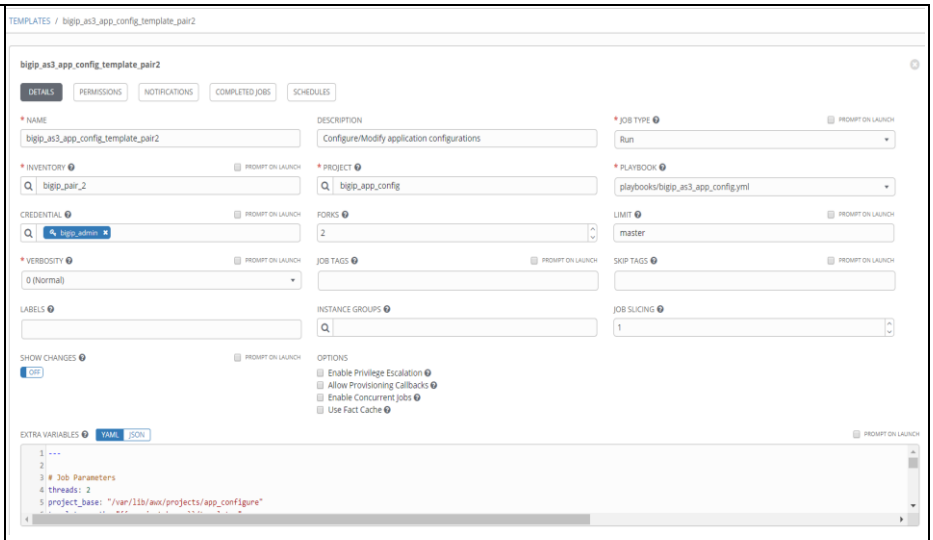
Click on & open the job template. Review the template configurations and parameters.

Notes:

- *User can clone a new job template by copying the existing sample template, then modify/personalize the new template as needed.*
- *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template.*

device_as3_app_config_vars.xlsx

- *This is a "patch" scenario, which modifies the application already configured in Lab1 scenario 2, by adding pool members.*



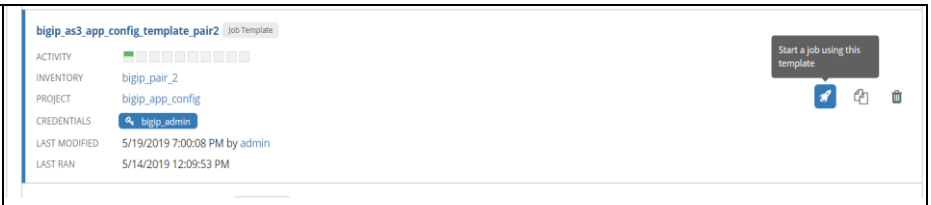
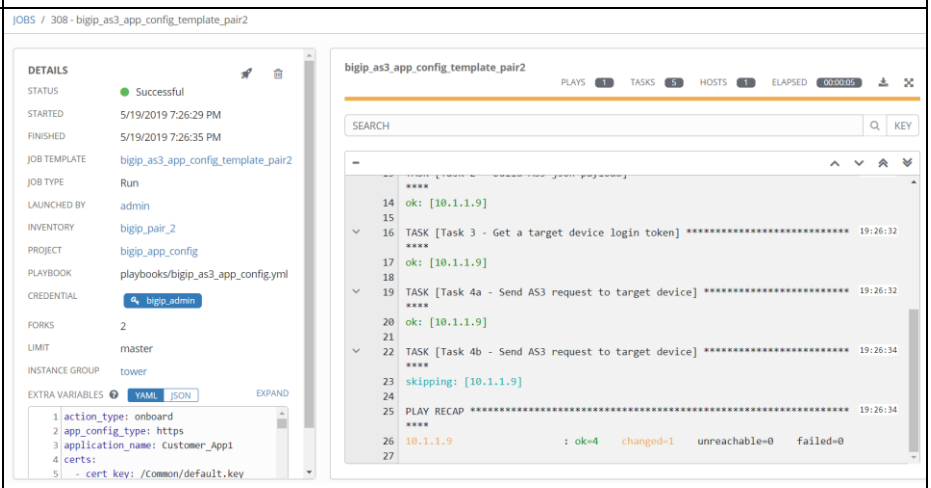
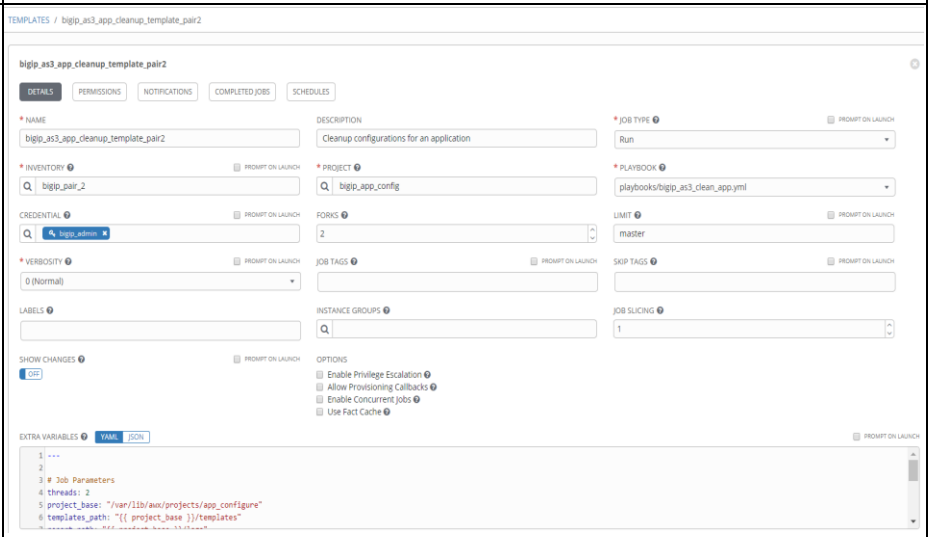
Modify the job template bigip_as3_app_config_template_pair2 in case modifications required to existing AS3 declared app configurations in Lab1 scenario 2. Within the "extra variables" section of the job template:

- Modify the action_type variable value as following:

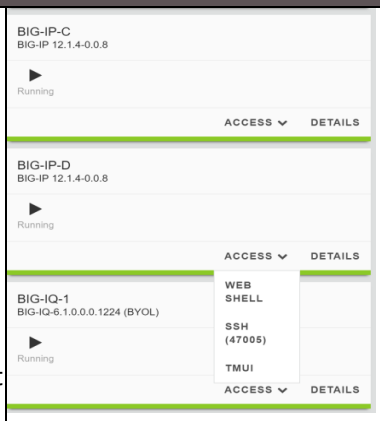
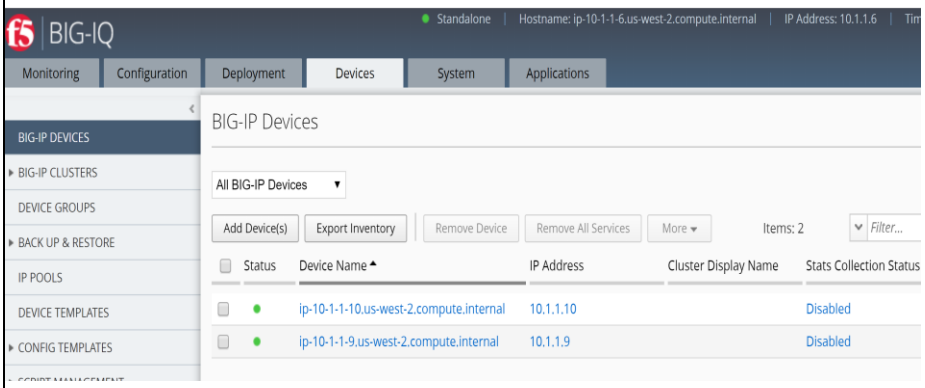
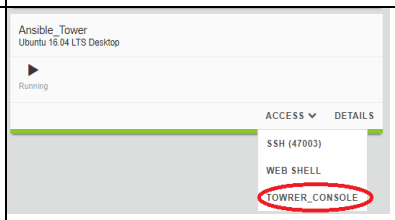
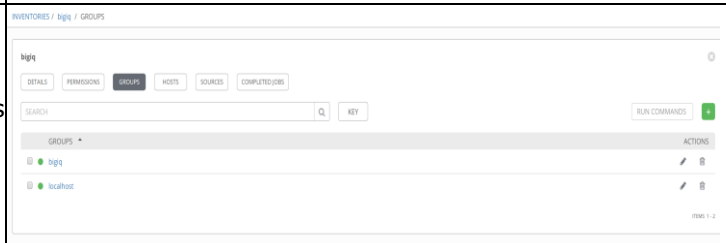
  action_type: "patch"

- Modify the configuration objects parameter values, for instance, add additional virtual servers or pool members, etc.

Trigger the job and confirm that the objects have been modified as expected on target BIGIP devices.

Trigger the job by clicking on the Start icon in the right hand-side of the selected job template. The job status window will open.	
Monitor the job execution status in the right hand-side status window. Review each task executed in the playbook. After the job completed successfully, login the target BIG-IP device pair GUIs or CLIs, verify the AS3 declared objects on the devices by switching to the "tenant_name" partition, and check the virtual servers and pools modification under LTM. Note: • *This example is to declare and "patch" a simple HTTPs application with SSL offload already configured in Lab 1 scenario 2.* • *The job only run on "master" BIGIP device, then verify the configurations have been synced to the "second" BIGIP device.*	
Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list: bigip_as3_app_cleanup_template_pair2 Click on & open the job template. Review the template configurations and parameters. Make sure the "tenant_name" value is the target tenant to be removed from the devices, and the "target_type" value is "tenant". For instance: *tenant_name: Customer_App1_Domain* *target_type: "tenant"* Trigger the job and verify if the app tenant and related objects have been removed from target devices.	

Scenario 2 - Application DO with AS3 (via BIGIQ)

Perform application configurations with AS3 via BIG-IQ (Ansible Tower)	
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab. Login the BIG-IP devices GUIs with credential: admin/admin, and confirm the application components have been removed by Lab3 Scenario 1. Login the BIG-IQ GUI with credential: admin/admin, and add the BIG-IP devices and DSC group from the "Devices" menu (this could be done by an automated task) if not currently managed. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab and access the BIG-IP devices and BIG-IQ Web GUIs and CLIs.*	 
Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	
Click the "Inventories" menu under "Resources" in the left panel. Locate and select the "bigiq" from the list. This is the target BIG-IQ used by this scenario. *Notes:* *Pay attention to the parameters defined at inventory level and group level.*	

Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list:

bigiq_as3_app_config_template_pair2

Within the "extra variables" section of the job template:

- Verify the action_type variable value as following:

  action_type: "onboard"

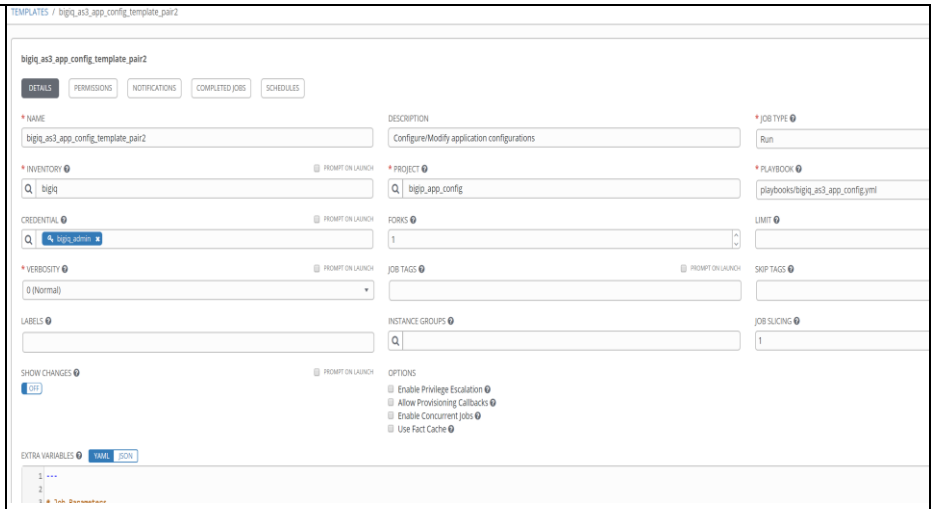
- Verify the tenant and application name to onboard the new app. Make sure no duplicated IPs used in the objects (vs, pool, etc.) if any app already configured previously.

Save the changes to the template.

Notes:

- *User can create a new job template by copying the existing sample template, then modify/personalize the new template as needed.*
- *Refer to the following attached spreadsheet for the variables defined in the "extra variables" yml of the sample job template*

device_as3_app_config_bigiq_vars.xlsx



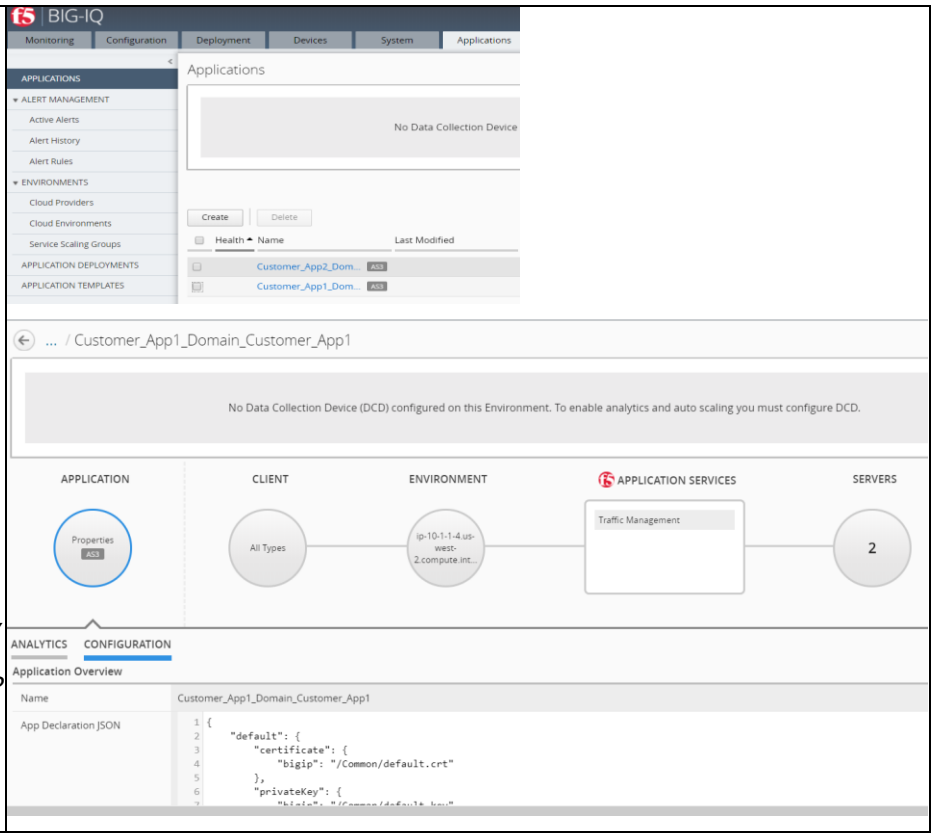
Trigger the job and verify:

- The app configuration entry appears in BIG-IQ under "Application" menu. Click on the app entry and verify the objects.
- The app objects have been configured as expected on target BIGIP devices by change the partition to "tenant_name" and verify the virtual servers and pools under LTM.

Notes:

- *This example is to declare and onboard a simple HTTPs application with SSL offload.*

  The job only needs to run on "master" BIGIP device, verify the configurations have been synced to the "second" BIGIP device

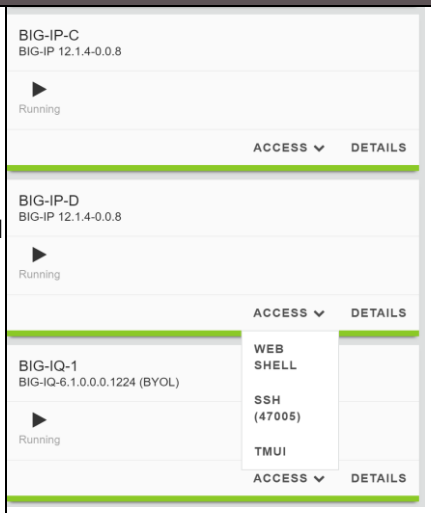
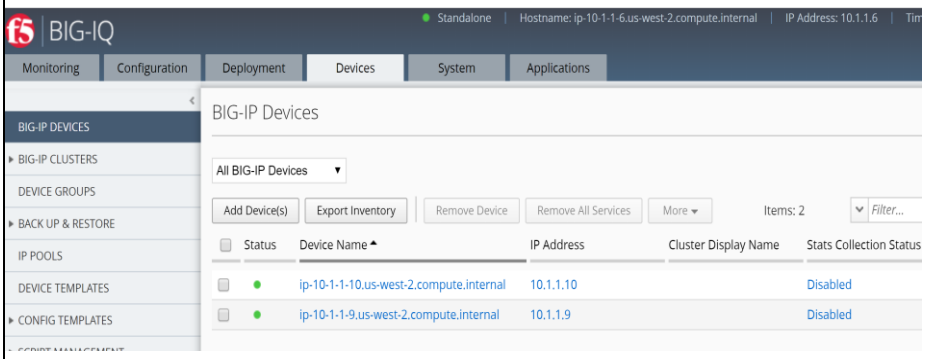
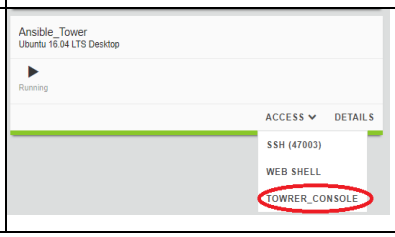


Note: Estimated time for the lab is about 10 minutes.

Lab 4 BIGIQ Exercises

Sample BIGIQ OAM tasks automations. Due to time restriction, only included few scenarios/tasks in this lab.

Scenario 1 – Device Management

Perform application configurations with AS3 directly to BIGIP (Ansible Tower)	
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab. Login the BIG-IQ GUI with credential: admin/admin Confirm the target BIG-IP devices have been managed by BIG-IQ, licensed and with services discovered and imported, which resulted from Lab 1 scenario 2. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab and access the BIG-IP devices and BIG-IQ Web GUIs and CLIs.*	 
Open Ansible Tower Console in UDF lab Login the Tower GUI with credential: admin/default	

Click the "Templates" menu under "Resources" in the left panel. Locate the follow sample job template from the list:

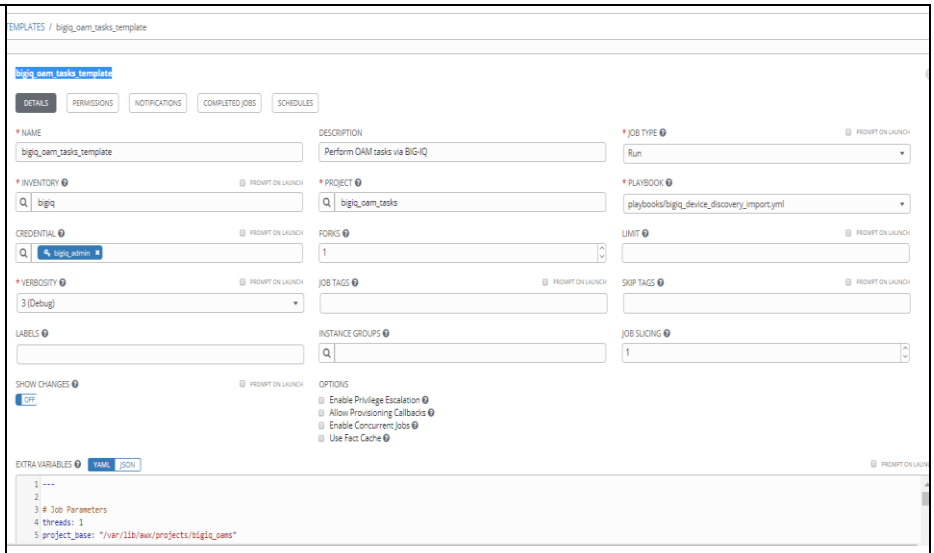
bigiq_oam_tasks_template

Click to open the job template. Review the template configurations and parameters.

Notes:

- *User can create a new job template by copying the existing sample template, then modify/personalize the new template as needed.*
- *Refer to attached spreadsheet for the variables/parameters required by the job template.*

bigiq_oams_params.xlsx



Remove BIG-IP devices from BIG-IQ:

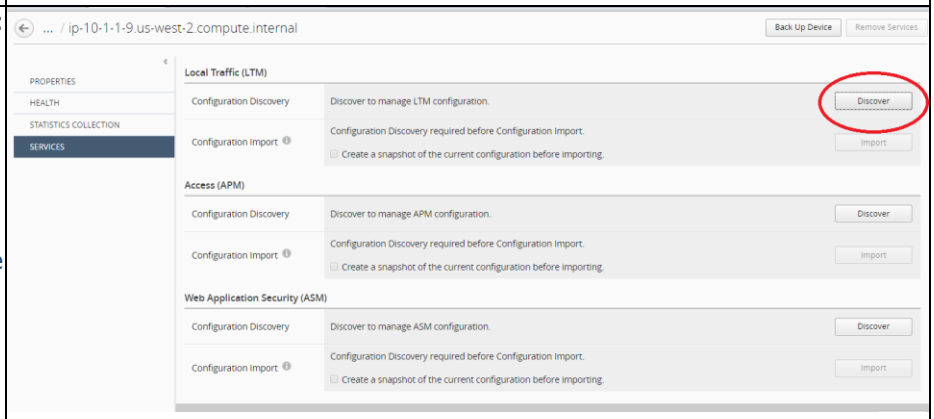
- Modify in "extra variables" window of the job template:

  iq_action_type: "remove"

- Select from "playbook" drop down list:

  bigiq_device_service_remove.yml

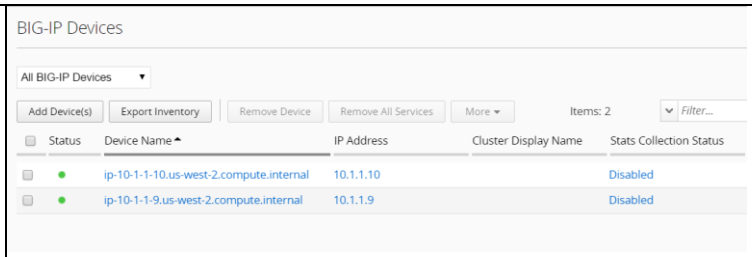
- Save the job template.
- Trigger the job and monitor the job status.
- Verify in BIG-IQ GUI:
  - Firstly, the services (LTM, DNS) have been removed for all target BIG-IP devices
  - Secondly, all target BIG-IP devices are removed from BIG-IQ "Devices" > "BIG-IP Devices"



<table>
<tr>
<td>Add BIG-IP devices to BIG-IQ:

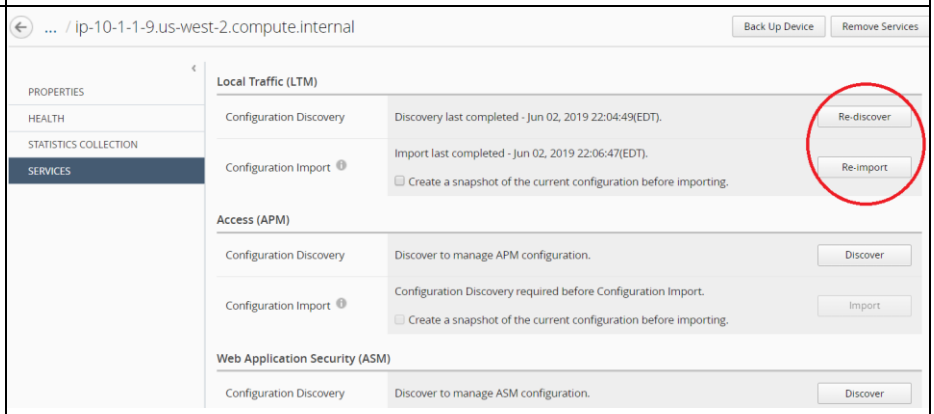
Modify in “extra variables” window of the job template:
iq_action_type: "add"
Select from “playbook” drop down list:
bigiq_device_add_only.yml
Save the job template.
Trigger the job and monitor the job status.
Verify in BIG-IQ GUI:

All target BIG-IP devices are added under BIG-IQ “Devices” > “BIG-IP Devices” list
The target BIG-IP devices have no services discovered at the moment

</td>
<td> 
</td>
</tr>
<tr>
<td>Discover & Import BIG-IP devices configurations into BIG-IQ:

Modify in “extra variables” window of the job template:
iq_action_type: "discover"
Select from “playbook” drop down list:
bigiq_device_discovery_import.yml
Save the job template.
Trigger the job and monitor the job status.
Verify in BIG-IQ GUI:

The target BIG-IP devices have services (LTM & DNS) discovered and imported

Note: The Re-discovery & Re-import services will use this same playbook and procedure.</td>
<td> 
</td>
</tr>
</table>

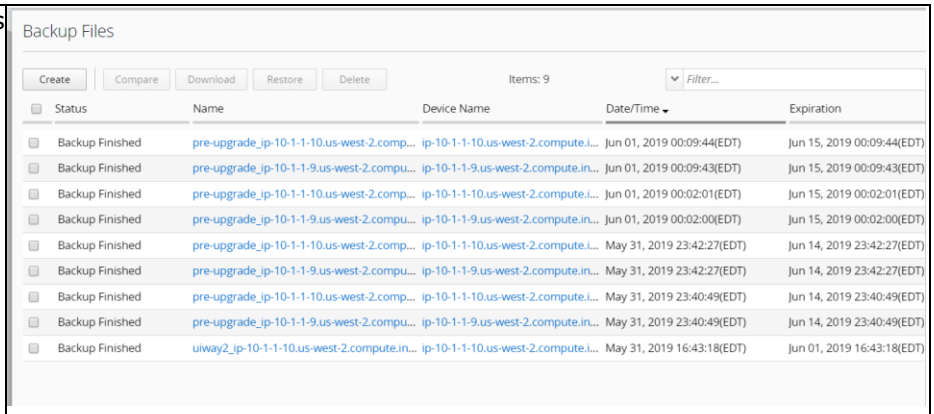
Backup BIG-IP devices configurations in BIG-IQ:

- Select from "playbook" drop down list:

  bigiq_get_bigip_ucs.yml

- Save the job template.
- Trigger the job and monitor the job status.
- Verify in BIG-IQ GUI:

  Backups have been created for the target BIG-IP devices under "Back Up & Restore" > "Backup Files" list



Upload and Run adhoc script on target BIG-IP devices from BIG-IQ:

(1) Upload:

- Modify in "extra variables" window of the job template:

  adhoc_script: 'test1.sh'

- Select from "playbook" drop down list:

  bigiq_script_upload.yml

- Save the job template.
- Trigger the job and monitor the job status.
- Verify in BIG-IQ GUI:

  The adhoc script has been upload to BIG-IQ under "Script Management" > "Scripts" list

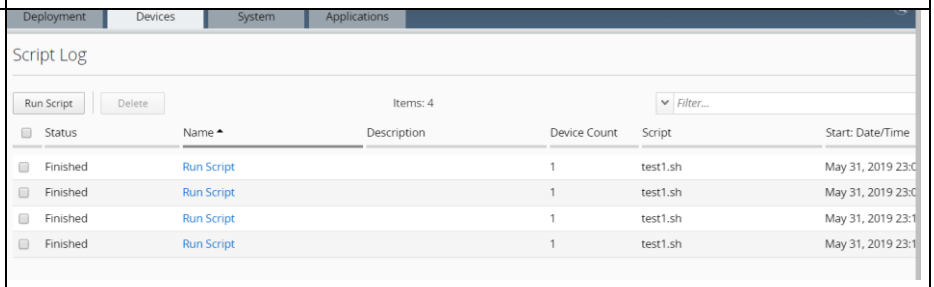
(2) Execute:

- Select from "playbook" drop down list:

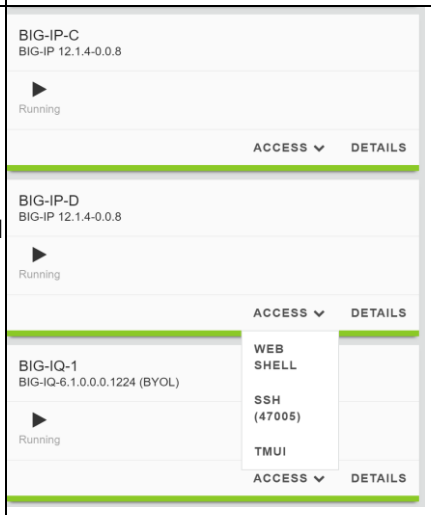
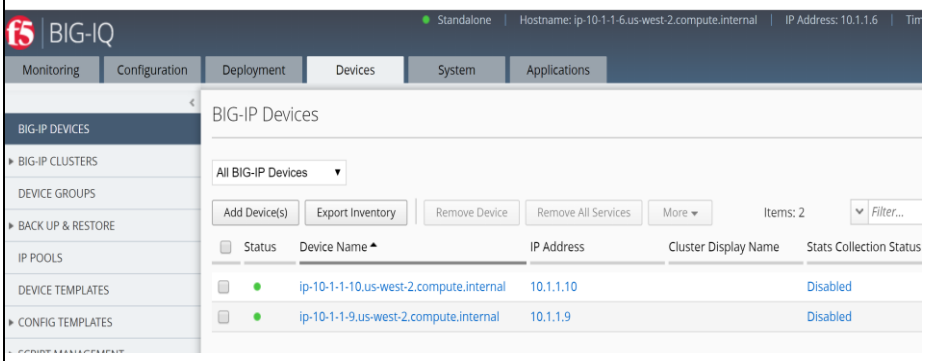
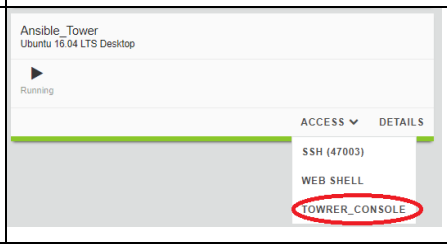
  bigiq_script_execute.yml

- Save the job template.
- Trigger the job and monitor the job status.
- Verify in BIG-IQ GUI:

  The adhoc script execution status for all target devices under "Script Management" > "Script Log" list



Scenario 2 – License Management

Perform application configurations with AS3 via BIG-IQ (Ansible Tower)	
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab. Login the BIG-IQ GUI with credential: admin/admin Confirm the target BIG-IP devices have been managed by BIG-IQ, licensed and with services discovered and imported, which resulted from Lab 1 scenario 2. *Notes:* • *User can also login the RDP of "Windows Client" in UDF lab and access the BIG-IP devices and BIG-IQ Web GUIs and CLIs.*	 
Open Tower Console in UDF lab Login the Tower GUI with credential: admin/default	

<table>
<tr>
<td>Import a list of new reg keys into an existing license pool in BIG-IQ:

• Modify in “extra variables” window of the job template:

regkey_pool: "test_pool_2"

regkey_list:

- 'O4207-09340-91384-11808-4283942'

- 'T3546-96929-26301-89310-7213224'

Note: Above are sample keys. User needs to generate a couple of new evaluation keys.

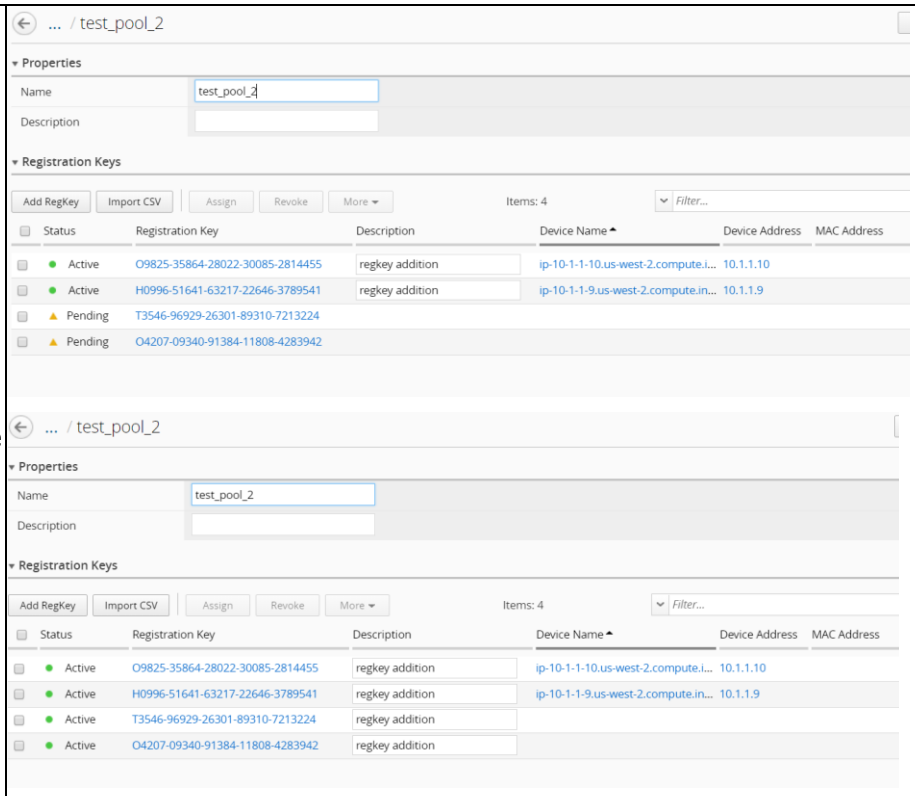
• Select from “playbook” drop down list:

bigiq_regkeys_import_activate_pool.yml

• Save the job template.

• Trigger the job and monitor the job status.

• Verify in BIG-IQ GUI:

The list of reg keys are imported and activated in the target pool.</td>
<td> 
</td>
</tr>
<tr>
<td>Revoke BIG-IP devices licenses from BIG-IQ:

• Modify in “extra variables” window of the job template:

license_action: "revoke"

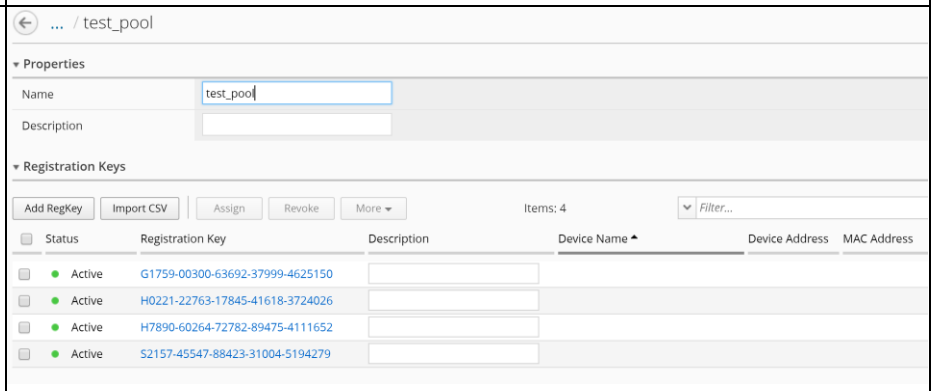
• Select from “playbook” drop down list:

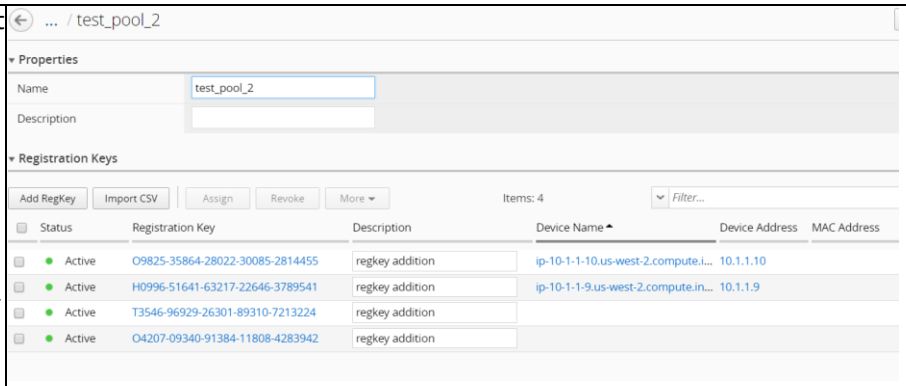
bigiq_device_license_revoke.yml

• Save the job template.

• Trigger the job and monitor the job status.

• Verify in BIG-IQ GUI:

The target BIG-IP devices license have been revoked</td>
<td> 
</td>
</tr>
</table>

Assign licenses from BIG-IQ to target BIG-IP devices: • Modify in "extra variables" window of the job template: license_action: "assign" • Select from "playbook" drop down list: bigiq_device_license_assign_only.yml • Save the job template. • Trigger the job and monitor the job status. • Verify in BIG-IQ GUI: The target BIG-IP devices have been licensed	

Note: Estimated time for the lab is about 30 minutes.

Appendices

A. Ansible Tower Project Files

The default Ansible Tower project base path is /var/lib/awx/projects/. The following sub directories contain specific files for each project.

(1) ~/files sub directory
Contains files used by the project, for instance, rpm, image files

(2) ~/playbooks sub directory

Contains all the playbooks for the project.

(3) ~/library sub directory
Ansible modules put under this sub directory override the ansible modules installed along with Ansible installation.

(4) ~/logs sub directory
This directory contains all the logs/reports. For instance, each device has its own directory containing the checking, staging, SW upgrade reports and temp files (ucs, qkview, dossier, etc). These files are only accesses and reviewed in case troubleshooting needed.

(5) ~/scripts sub directory

This directory contains certain utility scripts, called by the playbooks as needed to perform certain tasks.

(6) ~/vars sub directory
This directory contains extra var files referenced by the playbooks.

(7) ~/templates sub directory
This directory usually contains pre-defined templates used by playbooks. For instance, jinja2 templates to define a specific application, or templated configurations.

(8) ~/roles sub directory
This directory usually contains pre-defined roles used by playbooks, which simplify the playbooks and procedures by packaging common tasks and procedures before hands. which task to resume the procedure.

B. Reset BIG-IP Configures

Users can reset the UDF lab BIG-IP devices settings with the following steps, in case re-running the “new device config” scenarios:

- Remove the BIG-IP devices (pair1 or pair2) from BIG-IQ GUI (manually or use playbook)

- Load default config via BIG-IP CLIs:

> tmsh load sys config default

- Modify the root and admin password via BIG-IP CLIs:
  > tmsh modify auth password root (default)
  > tmsh modify auth password admin (admin)
  > tmsh save sys config

- Login BIG-IP GUI, enable root login manually in Platform (only for the 1st time)

- Reboot the device

Note: New UDF lab devices are instantiated with management port, hostname, and license configured, after re-licensed with BIG-IQ by the playbook, restarting the UDF lab/devices might cause license signature unmatched error. To work around the issue: Manually remove the BIG-IP devices with their services from BIG-IQ GUI, then re-add the BIG-IP devices and re-assign them licenses from the pool.

C. References

1. Ansible:
F5 GitHub: https://github.com/F5Networks/f5-ansible
F5 Ansible Modules:
https://docs.ansible.com/ansible/latest/modules/list_of_network_modules.html#f5
Ansible Distributions: https://pypi.org/project/ansible
Ansible Tower Documentation: https://docs.ansible.com/ansible-tower/
YAML Syntax: https://learn.getgrav.org/advanced/yaml

2. AS3:
AS3 Repository: https://github.com/F5Networks/f5-appsvcs-extension

Download: https://github.com/F5Networks/f5-appsvcs-extension/tree/master/dist

Bugs and Issues report: https://github.com/F5Networks/f5-appsvcs-extension/issues

User guide: https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/latest/

Schema Reference: https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/3/refguide/schema-reference.html

Declaration Example: https://clouddocs.f5.com/products/extensions/f5-appsvcs-extension/3/userguide/examples.html

JSON Schema: http://json-schema.org/

JSON Patch: http://jsonpatch.com/

Support Policy: https://www.f5.com/services/support/support-offerings/support-policies

3. DO:
DO Repository: https://github.com/F5Networks/f5-declarative-onboarding/tree/master/dist
User guide: https://clouddocs.f5.com/products/extensions/f5-declarative-onboarding/latest/
Declaration Example: https://clouddocs.f5.com/products/extensions/f5-declarative-onboarding/latest/examples.html