

# Automation Practices

## Lab Guide

Version 1.0

## Table of Contents

Table of Contents .....	2
Executive Summary .....	3
Required hardware and software .....	3
Components in UDF .....	3
Lab 1 BIGIP New Device Configuration Exercises .....	4
Scenario 1 - New BIGIP Device Configuration (TMSH) .....	4
Scenario 2 - New BIGIP Device Configuration (DO) .....	7
Lab 2 BIGIP Device SW Upgrade Exercises .....	10
Scenario 1 - BIGIP Device Software Upgrade (TMSH) .....	10
Lab 3 AS3 Exercises .....	13
Scenario 1 - Application DO with AS3 (Direct) .....	13
Scenario 2 - Application DO with AS3 (via BIGIQ) .....	17
Lab 4 BIGIQ Exercises .....	20
Scenario 1 - Device Management .....	20
Scenario 2 - License Management .....	24
Appendices .....	27
A.    Ansible Tower Project Files .....	27
B.    Reset BIG-IP Configures .....	27
C.    References .....	28

## 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+
- 3<sup>rd</sup> 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)

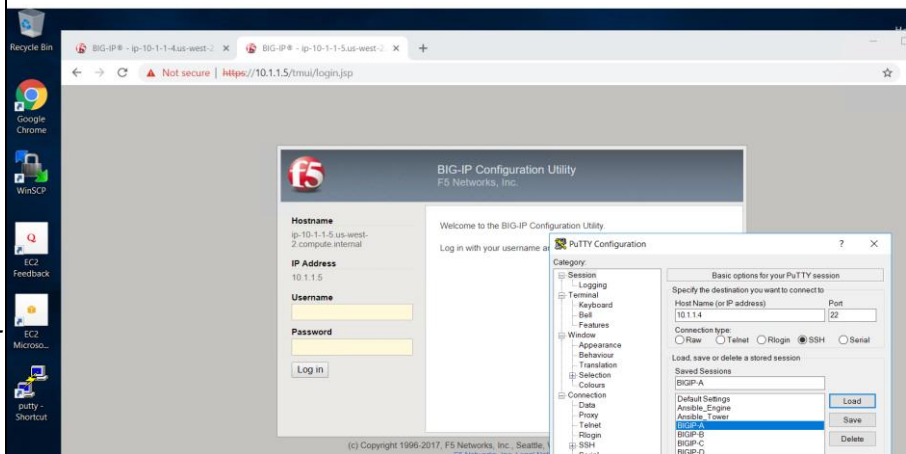
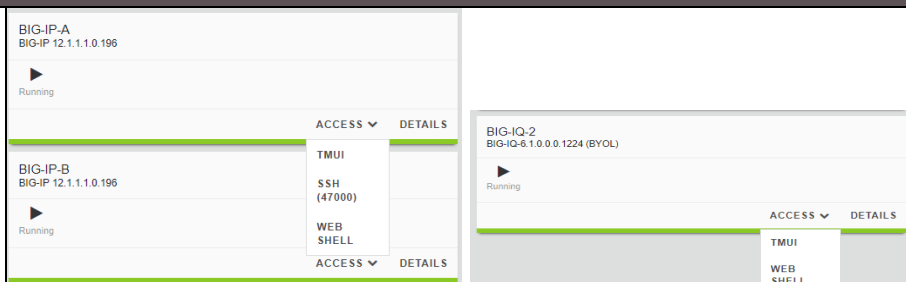
Open BIG-IP-A, BIG-IP-B and BIG-IQ-2 TMUI Consoles in UDF lab

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.

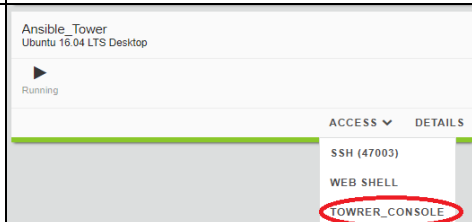
Notes:

- 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.

NAME	TYPE	ORGANIZATION	ACTIONS
all_pair1	Inventory	Default	[edit] [clone] [delete]
all_pair2	Inventory	Default	[edit] [clone] [delete]
bigip_pair_1	Inventory	Default	[edit] [clone] [delete]
bigip_pair_2	Inventory	Default	[edit] [clone] [delete]
bigip	Inventory	Default	[edit] [clone] [delete]
Demo Inventory	Inventory	Default	[edit] [clone] [delete]

Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the following 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\_variables.xlsx

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

bigip\_config\_initial\_template\_pair1

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

\* NAME bigip\_config\_initial\_template\_pair1 DESCRIPTION Initial device configurations with tmsh module \* JOB TYPE Run

\* INVENTORY all\_pair1 \* PROJECT bigip\_device\_config \* PLAYBOOK playbooks/bigip\_device\_initial\_config.yml

CREDENTIAL bigip\_root FORKS 1 LIMIT

\* VERBOSE 0 (Normal) JOB TAGS SKIP TAGS upload\_image, ns\_test, install\_ip, wait\_install\_ip

LABELS INSTANCE GROUPS JOB SLICING 1

SHOW CHANGES OPTIONS

- Enable Privilege Escalation
- Allow Provisioning Callbacks
- Enable Concurrent Jobs
- Use Fact Cache

EXTRA VARIABLES YML JSON

```
1: ---
2: # Common/shared parameters
3: threads: 2
4:
```

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.

bigip\_config\_initial\_template\_pair1 Job Template

ACTIVITY [progress bar]

INVENTORY all\_pair1

PROJECT bigip\_device\_config

CREDENTIALS bigip\_root

LAST MODIFIED 5/27/2019 5:47:53 PM by admin

Start a job using this template

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.

The screenshot displays the Ansible Tower interface for a job named 'bigip\_config\_initial\_template\_pair1'. The left pane shows job details: STATUS is 'Successful', STARTED at 5/24/2019 2:22:31 PM, FINISHED at 5/24/2019 2:38:39 PM, JOB TEMPLATE is 'bigip\_config\_initial\_template\_pair1', JOB TYPE is 'Run', LAUNCHED BY is 'admin', INVENTORY is 'all\_pair1', PROJECT is 'bigip\_device\_config', PLAYBOOK is 'playbooks/bigip\_device\_initial\_config.yml', CREDENTIAL is 'bigip\_root', FORKS is 1, and INSTANCE GROUP is 'tower'. The 'EXTRA VARIABLES' section shows a list of variables: 1. action\_type: add, 2. adhoc\_scripts\_names: test1.sh, test2.sh, test3.sh. The right pane shows the job progress with 4 plays, 72 tasks, 3 hosts, and 00:16:07 elapsed. A search bar is present. Below the progress bar, a task log for a 'debug' task is visible, showing messages for devices 10.1.1.4 and 10.1.1.5. At the bottom, a 'PLAY RECAP' table summarizes the results:

Host	ok	changed	unreachable	failed
10.1.1.4	44	32	0	0
10.1.1.5	34	28	0	0
10.1.1.6	18	2	0	0

## Scenario 2 - New BIGIP Device Configuration (DO)

Perform new device pair configurations with DO (Ansible Tower)

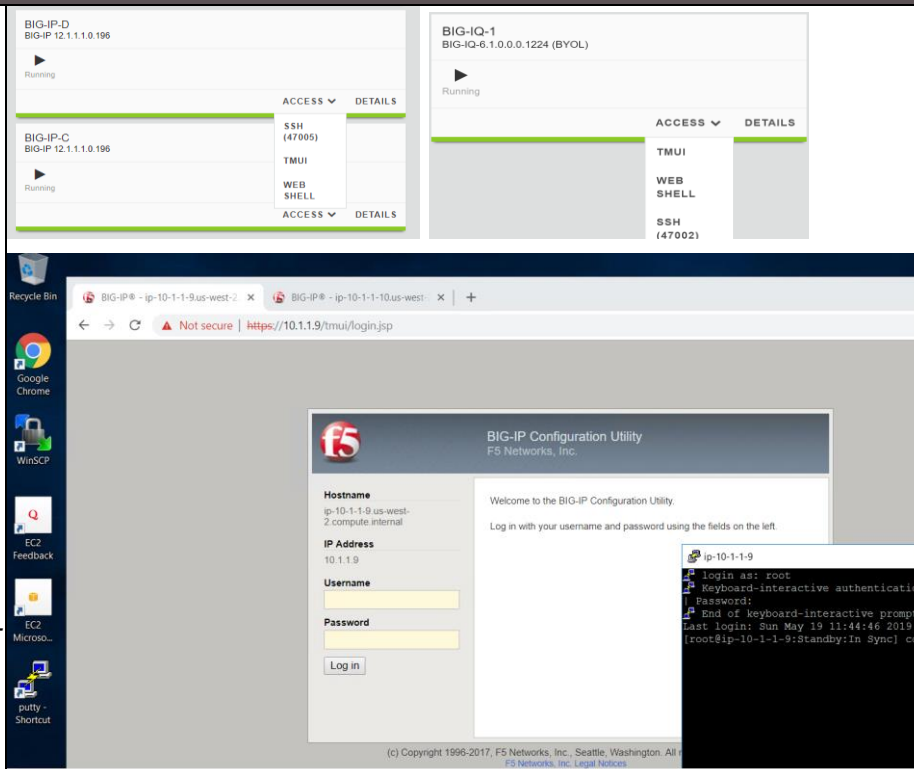
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab

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.

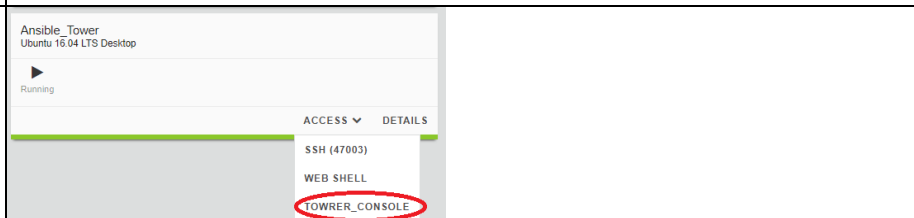
Notes:

- 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

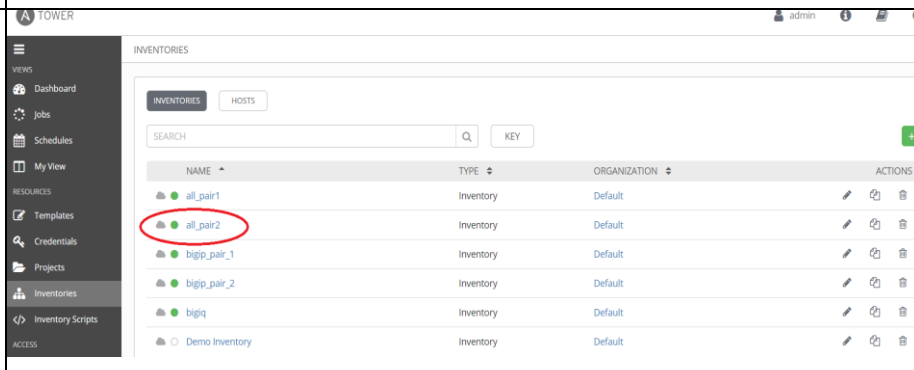
Login the Tower GUI with credential: [admin/default](#)



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\_v  
ars.xlsx

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

TEMPLATES / bigip\_config\_initial\_template\_pair2

bigip\_config\_initial\_template\_pair2

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

\* NAME: bigip\_config\_initial\_template\_pair2

DESCRIPTION: Initial device configurations with DO

\* JOB TYPE: Run

\* INVENTORY: all\_pair2

\* PROJECT: bigip\_device\_config

\* PLAYBOOK: playbooks/bigip\_device\_initial\_config\_do.yml

\* CREDENTIAL: bigip\_root

\* FORKS: 2

\* LIMIT: 1

\* VERBOSITY: 3 (Debug)

\* JOB TAGS: upload\_image, rm, boot, install, os, wait\_install, job

\* LABELS:

\* INSTANCE GROUPS:

\* JOB SLICING: 1

\* SHOW CHANGES: OFF

\* OPTIONS: Enable Privilege Escalation, Allow Provisioning Callbacks, Enable Concurrent Jobs, Use Fact Cache

EXTRA VARIABLES: Yaml, JSON

```
1: ---
2: # General/shared parameters
3: threads: 2
4:
5: project_base: "/var/lib/aux/projects/device_configure"
```

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.

bigip\_config\_initial\_template\_pair2 Job Template

ACTIVITY	Progress bar
INVENTORY	all_pair2
PROJECT	bigip_device_config
CREDENTIALS	bigip_root
LAST MODIFIED	5/27/2019 5:21:49 PM by admin
LAST RAN	5/27/2019 5:21:49 PM

Start a job using this template



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.

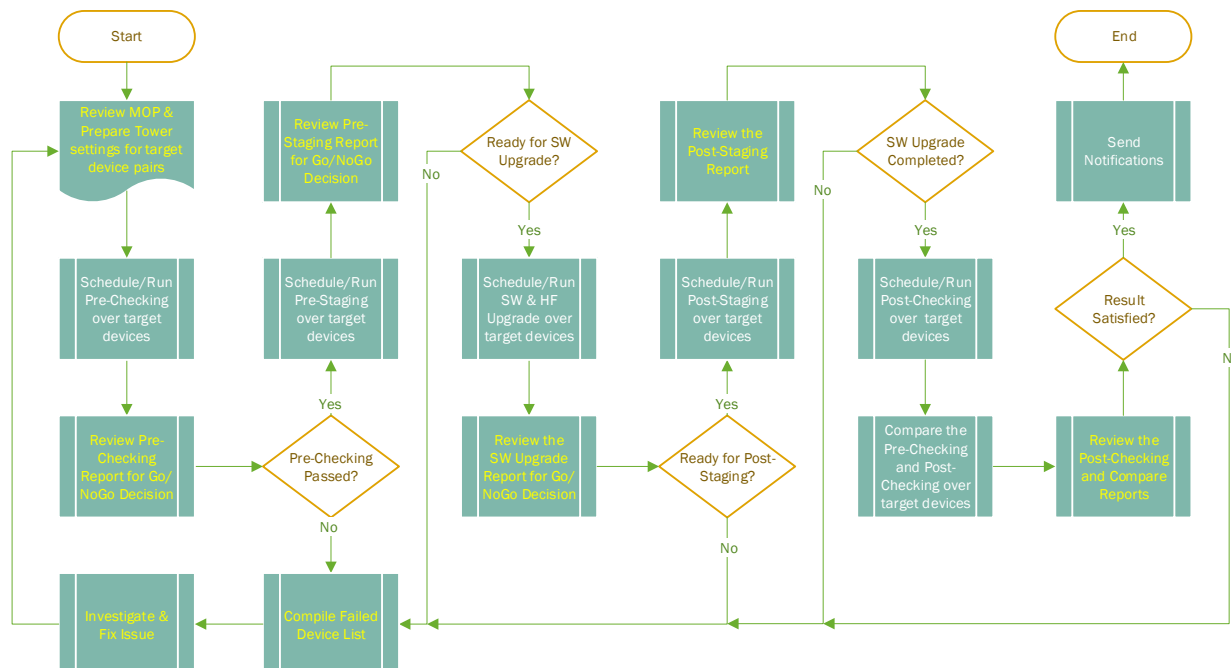
The screenshot displays the Ansible Tower interface for a job execution. On the left, the 'DETAILS' panel shows the job status as 'Successful', started at 5/24/2019 8:19:58 AM, and finished at 5/24/2019 8:41:18 AM. The job template is 'bigip\_config\_initial\_template\_pair1', launched by 'admin' using the 'all\_pair1' inventory. The project is 'bigip\_device\_config' and the playbook is 'playbooks/bigip\_device\_initial\_config.yml'. The credential is 'bigip\_root'. The job has 2 forks, a verbosity of 3 (Debug), and is part of the 'tower' instance group. The 'EXTRA VARIABLES' section shows a list with three items: 'action\_type: add', 'ad hoc\_scripts\_names:', and '- test1.sh'. On the right, the 'bigip\_config\_initial\_template\_pair1' job details are shown, including a search bar and a list of tasks. The tasks are grouped by host: 10.1.1.4, 10.1.1.5, and 10.1.1.6. The tasks include 'ok: [10.1.1.4] => {', 'msg: "Device 10.1.1.4 initial configuration has been completed successfully! Please review the detailed reports(within Stdout tab) in above Detailed Checking report task by clicking the link of each target device."', 'ok: [10.1.1.5] => {', 'msg: "Device 10.1.1.5 initial configuration has been completed successfully! Please review the detailed reports(within Stdout tab) in above Detailed Checking report task by clicking the link of each target device."', 'META: ran handlers', and 'PLAY RECAP'. The 'PLAY RECAP' table shows the following results:

Host	ok	changed	unreachable	failed
10.1.1.4	38	33	0	0
10.1.1.5	40	29	0	0
10.1.1.6	18	2	0	0

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)

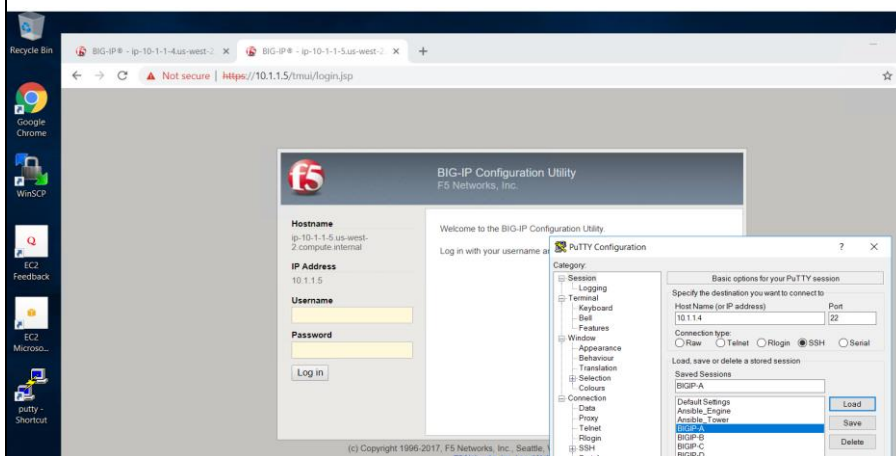
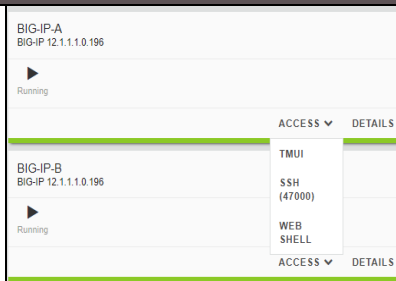
Perform device pair software upgrade (Ansible Tower)

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.

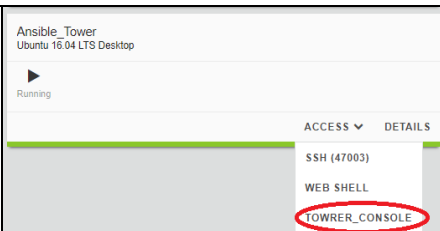
Notes:

- 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.

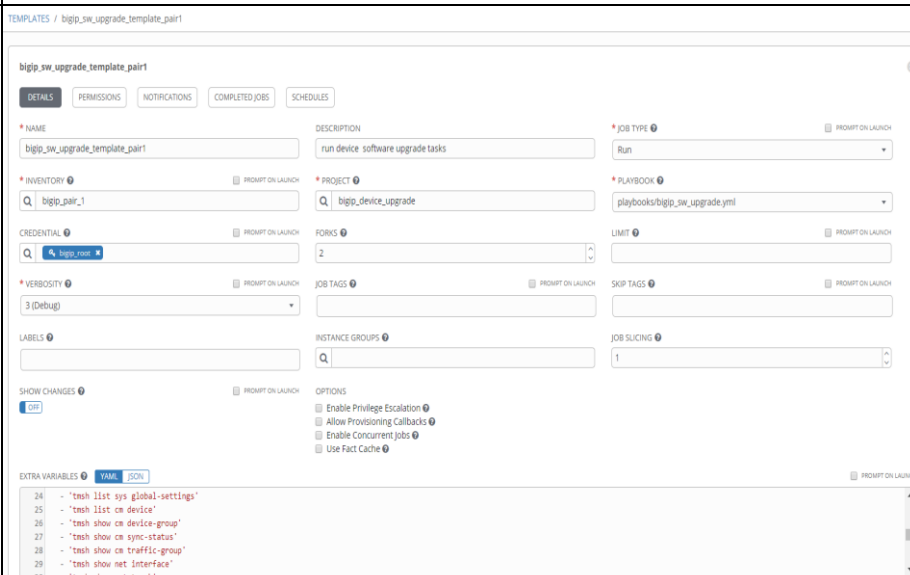
Notes:

- 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\_v  
ars.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.

The screenshot displays the Ansible Tower web interface. The top section shows the job details for 'bigip\_sw\_upgrade\_template\_pair1', which is in a 'Successful' state. It lists various metadata such as start/finish times, job template, and inventory. Below this, the 'EXTRA VARIABLES' section is visible. The main part of the interface is divided into two panes. The left pane shows the 'STANDARD OUT' tab for the task '10.1.1.4', displaying the output of the 'hostname' command for two target devices. The right pane shows the 'STANDARD ERROR' tab, which contains debug messages and a 'PLAY RECAP' summary indicating that all tasks were successful.

JOBS / 220 - bigip\_sw\_upgrade\_template\_pair1

DETAILS

STATUS: Successful

STARTED: 5/12/2019 8:06:17 PM

FINISHED: 5/12/2019 9:16:57 PM

JOB TEMPLATE: bigip\_sw\_upgrade\_template\_pair1

JOB TYPE: Run

LAUNCHED BY: admin

INVENTORY: bigip\_pair\_1

PROJECT: bigip\_device\_upgrade

PLAYBOOK: playbooks/bigip\_sw\_upgrade.yml

CREDENTIAL: % bigip\_node

FORKS: 2

VERBOSE: 3 (Debug)

INSTANCE GROUP: tower

EXTRA VARIABLES

1 adhoc\_scripts\_names:

2 - test1.sh

3 - test2.sh

4 adhoc\_scripts\_path: '{{ project\_base }}/scripts

5 license\_year: 2019

6 new\_device\_total #oldapi: {{ new\_device\_total }}

bigip\_sw\_upgrade\_template\_pair1

PLAYS: 0 TASKS: 123 HOSTS: 0 ELAPSED: 00:00:00

SEARCH

15115

15116

15117

15118 TASK [debug] \*\*\*\*\* 21:10:56

15119 task path: /var/lib/ansible/projects/device\_upgrade/playbooks/bigip\_postcheck.yml:93

15120 ok: [10.1.1.5] => {

15121 "msg": "Device 10.1.1.5 Post-checking has Passed successfully! Please review the above

15122 summary and detailed reports"

15123 }

15124 ok: [10.1.1.4] => {

15125 "msg": "Device 10.1.1.4 Post-checking has Passed successfully! Please review the above

15126 summary and detailed reports"

15127 }

15128 META: ran handlers

15129 META: ran handlers

15130

15131 PLAY RECAP \*\*\*\*\* 21:10:56

15132 10.1.1.4 : ok=91 changed=62 unreachable=0 failed=0

15133 10.1.1.5 : ok=82 changed=62 unreachable=0 failed=0

15134

15135

10.1.1.4

CREATED: 5/24/2019 3:52:23 PM

ID: 58063

PLAY: Perform Device Post-Checking

TASK: Task 9 - Go/No-Go decision summary report

MODULE: raw

JSON STANDARD OUT STANDARD ERROR

1 [Target Device Hostname]

hostname ip-10-1-1-4.us-west-2.compute.internal

[Target Devices]

failover-state active

hostname ip-10-1-1-4.us-west-2.compute.internal

failover-state standby

hostname ip-10-1-1-5.us-west-2.compute.internal

CLOSE

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)

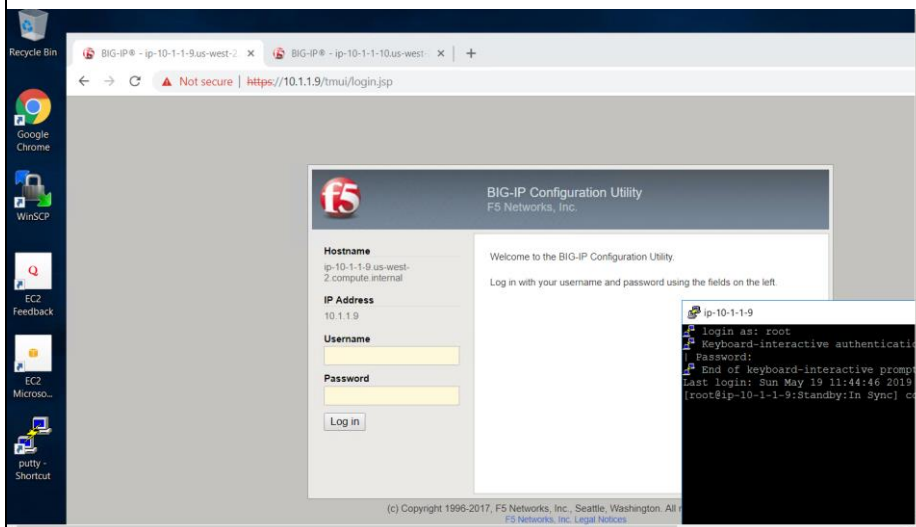
Perform application configurations with AS3 directly to BIGIP (Ansible Tower)

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.

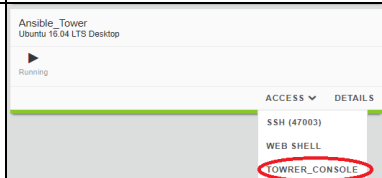
Notes:

- 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

TEMPLATES / bigip\_as3\_rpm\_manage\_template\_pair2

bigip\_as3\_rpm\_manage\_template\_pair2

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

\* NAME: bigip\_as3\_rpm\_manage\_template\_pair2

DESCRIPTION: Perform AS3 RPM package management tasks

\* JOB TYPE: Run

\* INVENTORY: bigip\_pair\_2

\* PROJECT: bigip\_app\_config

\* PLAYBOOK: playbooks/device\_as3\_rpm\_manage.yml

CREDENTIAL: bigip\_admin

FORKS: 2

\* VERBOSITY: 0 (Normal)

JOB TAGS:

Skip TAGS:

LABELS:

INSTANCE GROUPS:

JOB SLICING: 1

SHOW CHANGES: OFF

OPTIONS:

- Enable Privilege Escalation
- Allow Provisioning Callbacks
- Enable Concurrent Jobs
- Use Fact Cache

EXTRA VARIABLES: yaml json

```
1 ---
2
3 # Job Parameters
4 threads: 2
5 project_base: "/var/lib/ansible/projects/app_config"
6 templates_path: "[[ project_base ]]/templates"
```

(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.

bigip\_as3\_rpm\_manage\_template\_pair2 Job Template

ACTIVITY: [Progress Bar]

INVENTORY: bigip\_pair\_2

PROJECT: bigip\_app\_config

CREDENTIALS: bigip\_admin

LAST MODIFIED: 5/15/2019 2:38:04 PM by admin

LAST RAN: 5/15/2019 2:38:04 PM

Start a job using this template

(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.

Main Help About iApps Package Management LX

filter packages:

Name	Version	Build	Package
i5-appsvcs	3.10.0	5	i5-appsvcs-3.10.0-5.noarch
i5-declarative-onboarding	1.4.0	1	i5-declarative-onboarding-1.4.0-1.noarch
i5-service-discovery	1.0.0	1	i5-service-discovery-1.0.0-1.noarch

Uninstall

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.

TEMPLATES / bigip\_as3\_app\_config\_template\_pair2

bigip\_as3\_app\_config\_template\_pair2

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

\* NAME bigip\_as3\_app\_config\_template\_pair2 DESCRIPTION Configure/Modify application configurations \* JOB TYPE Run

\* INVENTORY bigip\_pair\_2 \* PROJECT bigip\_app\_config \* PLAYBOOK playbooks/bigip\_as3\_app\_config.yml

CREDENTIAL bigip\_admin \* FORMS 2 LIMIT master

\* VERBOSITY 0 (Normal) \* JOB TAGS \* SKIP TAGS

LABELS \* INSTANCE GROUPS \* JOB SLICING 1

SHOW CHANGES COPY \* OPTIONS

EXTRA VARIABLES

```

1 ---
2
3 # Job Parameters
4 threads: 2
5 project_base: "/var/lib/awx/projects/app_configure"

```

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.

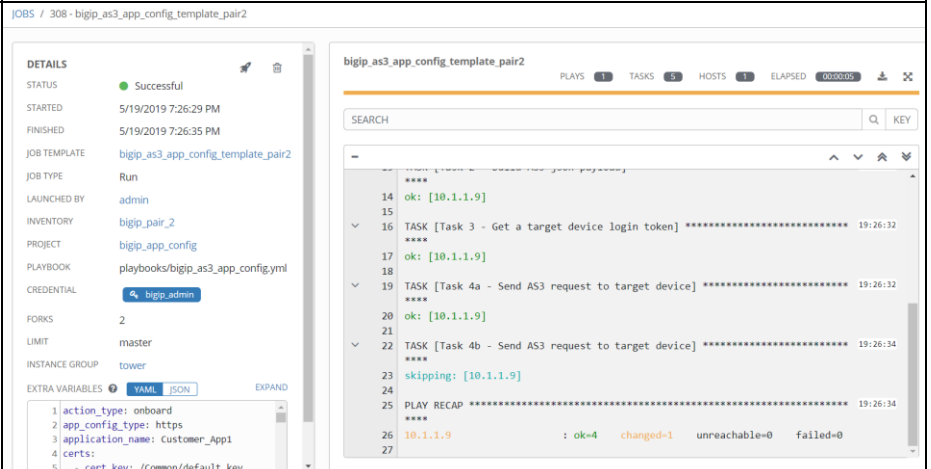


Monitor the job execution status in the right hand-side status window.  
Review each task executed in the playbook.

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.



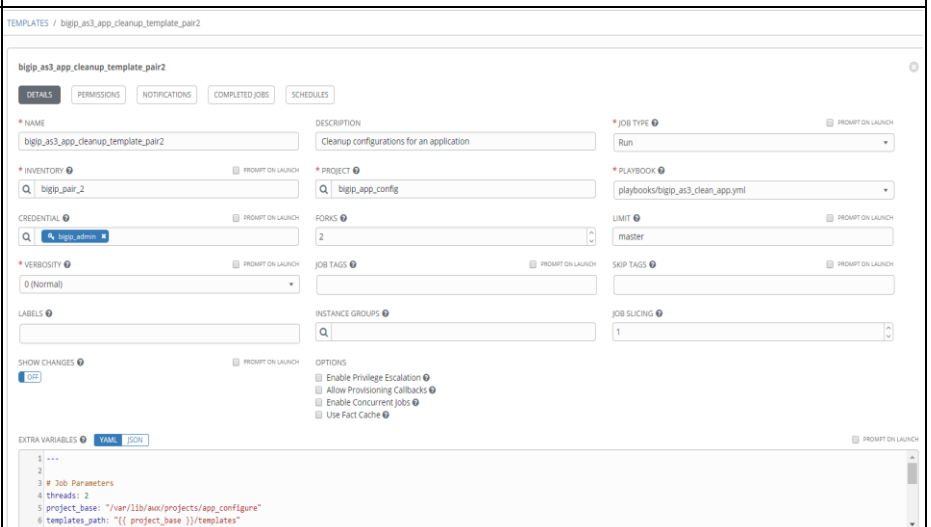
Click the "Templates" under "Resources" in the left-hand side menu panel. Locate the follow sample job template from the list:  
[bigip\\_as3\\_app\\_cleanup\\_template\\_pair2](#)

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)

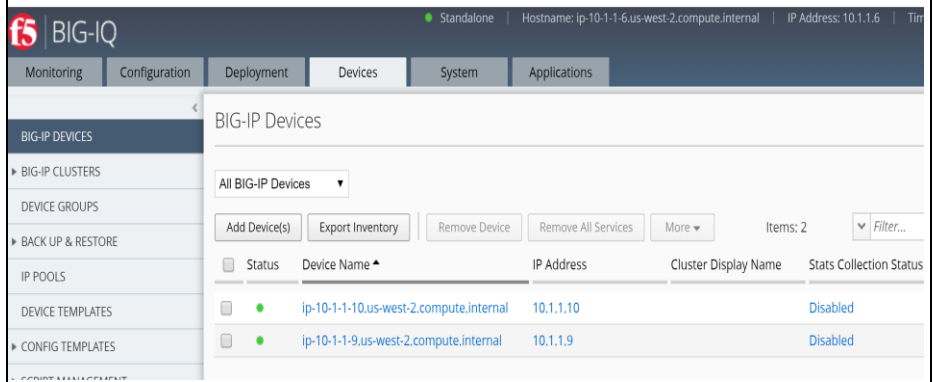
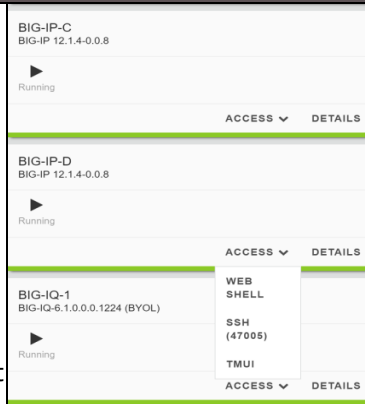
Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab.

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.

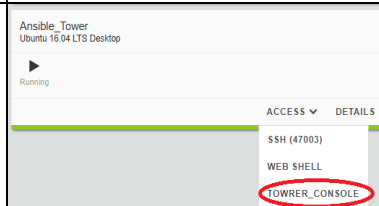
Notes:

- 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

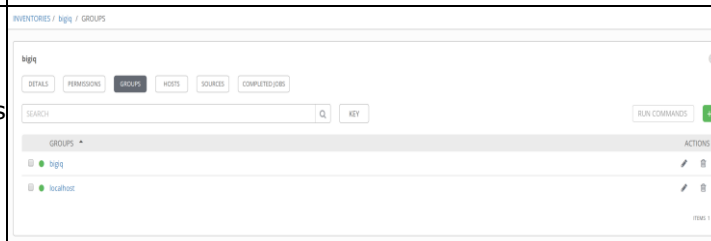
Login the Tower GUI with credential: [admin/default](#)



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\_conf  
g\_biq\_vars.xlsx

TEMPLATES / bigiq\_as3\_app\_config\_template\_pair2

bigiq\_as3\_app\_config\_template\_pair2

**DETAILS** PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES

**\* NAME** bigiq\_as3\_app\_config\_template\_pair2 **DESCRIPTION** Configure/Modify application configurations **\* JOB TYPE** Run

**\* INVENTORY** bigiq **\* PROJECT** bigiq\_app\_config **\* PLAYBOOK** playbooks/bigiq\_as3\_app\_config.yml

**\* CREDENTIAL** % bigiq\_admin **\* FORKS** 1 **\* LIMIT**

**\* VERBOSITY** 0 (Normal) **\* JOB TAGS** **\* SKIP TAGS**

**\* LABELS** **\* INSTANCE GROUPS** **\* JOB SLICING** 1

**SHOW CHANGES** **OPTIONS**

- ☐ Enable Privilege Escalation
- ☐ Allow Provisioning Callbacks
- ☐ Enable Concurrent Jobs
- ☐ Use Fact Cache

**EXTRA VARIABLES** **TOTAL** 320K

1 ---  
2  
3 1.4 TB Download

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*

The screenshot displays the BIG-IQ Configuration page. The left sidebar shows the navigation menu with 'APPLICATIONS' selected. The main content area shows the 'Applications' list with two entries: 'Customer\_App2\_Dom...' and 'Customer\_App1\_Dom...'. Below this, the 'Application Overview' for 'Customer\_App1\_Domain\_Customer\_App1' is shown. It includes a diagram with nodes for APPLICATION, CLIENT, ENVIRONMENT, APPLICATION SERVICES (Traffic Management), and SERVERS. The 'ANALYTICS' tab is active, showing the 'App Declaration JSON' with the following content:

```
1 {
2   "default": {
3     "certificate": {
4       "bigip": "/Common/default.crt"
5     },
6     "privateKey": {
7       "bigip": "/Common/default.key"
8     }
9   }
10 }
```

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)																			
<p>Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab.</p> <p>Login the BIG-IQ GUI with credential: admin/admin</p> <p>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.</p> <p><b>Notes:</b></p> <ul style="list-style-type: none"><li>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.</li></ul>	<div><div><div><div>BIG-IP-C BIG-IP 12.1.4-0.0.8</div><div>▶</div><div>Running</div></div><div>ACCESS ▾DETAILS</div></div><div><div><div>BIG-IP-D BIG-IP 12.1.4-0.0.8</div><div>▶</div><div>Running</div></div><div>ACCESS ▾DETAILS</div></div><div><div><div>BIG-IQ-1 BIG-IQ-6.1.0.0.0.1224 (BYOL)</div><div>▶</div><div>Running</div></div><div>WEB SHELL SSH (47005) TMUI ACCESS ▾DETAILS</div></div></div> <div><div><div><div><div><div><div>fs</div><div>BIG-IQ</div></div><div>Standalone</div><div>Hostname: ip-10-1-1-6.us-west-2.compute.internal</div><div>IP Address: 10.1.1.6</div><div>Tim</div></div><div>MonitoringConfigurationDeploymentDevicesSystemApplications</div><div><div>BIG-IP DEVICES</div><div>▶ BIG-IP CLUSTERS</div><div>DEVICE GROUPS</div><div>▶ BACK UP &amp; RESTORE</div><div>IP POOLS</div><div>DEVICE TEMPLATES</div><div>▶ CONFIG TEMPLATES</div></div><div><div>BIG-IP Devices</div><div>All BIG-IP Devices ▾</div><div>Add Device(s)Export InventoryRemove DeviceRemove All ServicesMore ▾Items: 2Filter...</div><table><tr><th><input type="checkbox"/></th><th>Status</th><th>Device Name ^</th><th>IP Address</th><th>Cluster Display Name</th><th>Stats Collection Status</th></tr><tr><td><input type="checkbox"/></td><td>●</td><td>ip-10-1-1-10.us-west-2.compute.internal</td><td>10.1.1.10</td><td></td><td>Disabled</td></tr><tr><td><input type="checkbox"/></td><td>●</td><td>ip-10-1-1-9.us-west-2.compute.internal</td><td>10.1.1.9</td><td></td><td>Disabled</td></tr></table></div></div></div></div><div><div><div><div>Ansible_Tower Ubuntu 16 04 LTS Desktop</div><div>▶</div><div>Running</div></div><div>ACCESS ▾DETAILS</div><div>SSH (47003) WEB SHELL TOWER_CONSOLE</div></div></div></div>	<input type="checkbox"/>	Status	Device Name ^	IP Address	Cluster Display Name	Stats Collection Status	<input type="checkbox"/>	●	ip-10-1-1-10.us-west-2.compute.internal	10.1.1.10		Disabled	<input type="checkbox"/>	●	ip-10-1-1-9.us-west-2.compute.internal	10.1.1.9		Disabled
<input type="checkbox"/>	Status	Device Name ^	IP Address	Cluster Display Name	Stats Collection Status														
<input type="checkbox"/>	●	ip-10-1-1-10.us-west-2.compute.internal	10.1.1.10		Disabled														
<input type="checkbox"/>	●	ip-10-1-1-9.us-west-2.compute.internal	10.1.1.9		Disabled														
<p>Open Ansible Tower Console in UDF lab</p> <p>Login the Tower GUI with credential: admin/default</p>																			

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”

#### 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

BIG-IP Devices

All BIG-IP Devices ▾

Items: 2

<input type="checkbox"/>	Status	Device Name ▲	IP Address	Cluster Display Name	Stats Collection Status
<input type="checkbox"/>	●	ip-10-1-1-10.us-west-2.compute.internal	10.1.1.10		Disabled
<input type="checkbox"/>	●	ip-10-1-1-9.us-west-2.compute.internal	10.1.1.9		Disabled

#### 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.

← ... / ip-10-1-1-9.us-west-2.compute.internal

PROPERTIES

HEALTH

STATISTICS COLLECTION

SERVICES

Local Traffic (LTM)

Configuration Discovery	Discovery last completed - Jun 02, 2019 22:04:49(EDT).	<input type="button" value="Re-discover"/>
Configuration Import ⓘ	Import last completed - Jun 02, 2019 22:06:47(EDT). <input type="checkbox"/> Create a snapshot of the current configuration before importing.	<input type="button" value="Re-import"/>

Access (APM)

Configuration Discovery	Discover to manage APM configuration.	<input type="button" value="Discover"/>
Configuration Import ⓘ	Configuration Discovery required before Configuration Import. <input type="checkbox"/> Create a snapshot of the current configuration before importing.	<input type="button" value="Import"/>

Web Application Security (ASM)

Configuration Discovery	Discover to manage ASM configuration.	<input type="button" value="Discover"/>
-------------------------	---------------------------------------	---

## 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

### Backup Files

<div> <div>Create</div> <div>Compare</div> <div>Download</div> <div>Restore</div> <div>Delete</div> </div> <div>Items: 9</div> <div>Filter...</div>					
<input type="checkbox"/>	Status	Name	Device Name	Date/Time	Expiration
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-10.us-west-2.comp...	ip-10-1-1-10.us-west-2.compute.i...	Jun 01, 2019 00:09:44(EDT)	Jun 15, 2019 00:09:44(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-9.us-west-2.comp...	ip-10-1-1-9.us-west-2.compute.in...	Jun 01, 2019 00:09:43(EDT)	Jun 15, 2019 00:09:43(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-10.us-west-2.comp...	ip-10-1-1-10.us-west-2.compute.i...	Jun 01, 2019 00:02:01(EDT)	Jun 15, 2019 00:02:01(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-9.us-west-2.comp...	ip-10-1-1-9.us-west-2.compute.in...	Jun 01, 2019 00:02:00(EDT)	Jun 15, 2019 00:02:00(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-10.us-west-2.comp...	ip-10-1-1-10.us-west-2.compute.i...	May 31, 2019 23:42:27(EDT)	Jun 14, 2019 23:42:27(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-9.us-west-2.comp...	ip-10-1-1-9.us-west-2.compute.in...	May 31, 2019 23:42:27(EDT)	Jun 14, 2019 23:42:27(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-10.us-west-2.comp...	ip-10-1-1-10.us-west-2.compute.i...	May 31, 2019 23:40:49(EDT)	Jun 14, 2019 23:40:49(EDT)
<input type="checkbox"/>	Backup Finished	pre-upgrade_ip-10-1-1-9.us-west-2.comp...	ip-10-1-1-9.us-west-2.compute.in...	May 31, 2019 23:40:49(EDT)	Jun 14, 2019 23:40:49(EDT)
<input type="checkbox"/>	Backup Finished	uiway2_ip-10-1-1-10.us-west-2.compute.in...	ip-10-1-1-10.us-west-2.compute.i...	May 31, 2019 16:43:18(EDT)	Jun 01, 2019 16:43:18(EDT)

## 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

### Deployment

### Devices

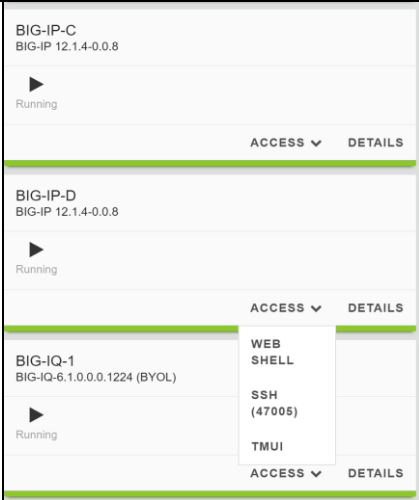
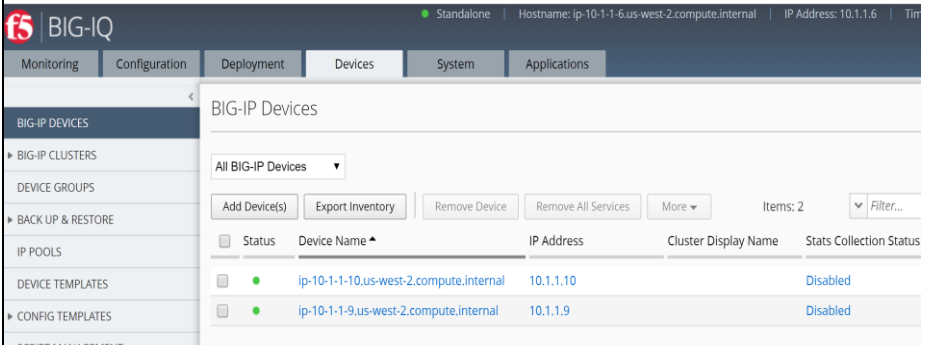
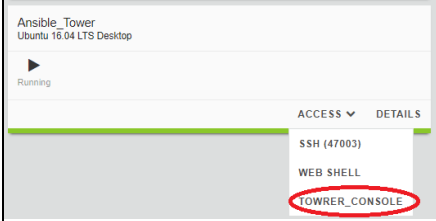
### System

### Applications

### Script Log

<div> <div>Run Script</div> <div>Delete</div> </div> <div>Items: 4</div> <div>Filter...</div>					
<input type="checkbox"/>	Status	Name	Description	Device Count	Script
<input type="checkbox"/>	Finished	Run Script		1	test1.sh
<input type="checkbox"/>	Finished	Run Script		1	test1.sh
<input type="checkbox"/>	Finished	Run Script		1	test1.sh
<input type="checkbox"/>	Finished	Run Script		1	test1.sh

Scenario 2 – License Management

Perform application configurations with AS3 via BIG-IQ (Ansible Tower)																	
<p>Open BIG-IP-C, BIG-IP-D and BIG-IQ-1 TMUI Consoles in UDF lab.</p> <p>Login the BIG-IQ GUI with credential: admin/admin</p> <p>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.</p> <p>Notes:</p> <ul style="list-style-type: none"><li>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.</li></ul>	  <table><caption>BIG-IP Devices</caption><tr><th>Status</th><th>Device Name</th><th>IP Address</th><th>Cluster Display Name</th><th>Stats Collection Status</th></tr><tr><td>●</td><td>ip-10-1-1-10.us-west-2.compute.internal</td><td>10.1.1.10</td><td></td><td>Disabled</td></tr><tr><td>●</td><td>ip-10-1-1-9.us-west-2.compute.internal</td><td>10.1.1.9</td><td></td><td>Disabled</td></tr></table>	Status	Device Name	IP Address	Cluster Display Name	Stats Collection Status	●	ip-10-1-1-10.us-west-2.compute.internal	10.1.1.10		Disabled	●	ip-10-1-1-9.us-west-2.compute.internal	10.1.1.9		Disabled	
Status	Device Name	IP Address	Cluster Display Name	Stats Collection Status													
●	ip-10-1-1-10.us-west-2.compute.internal	10.1.1.10		Disabled													
●	ip-10-1-1-9.us-west-2.compute.internal	10.1.1.9		Disabled													
<p>Open Tower Console in UDF lab</p> <p>Login the Tower GUI with credential: admin/default</p>																	



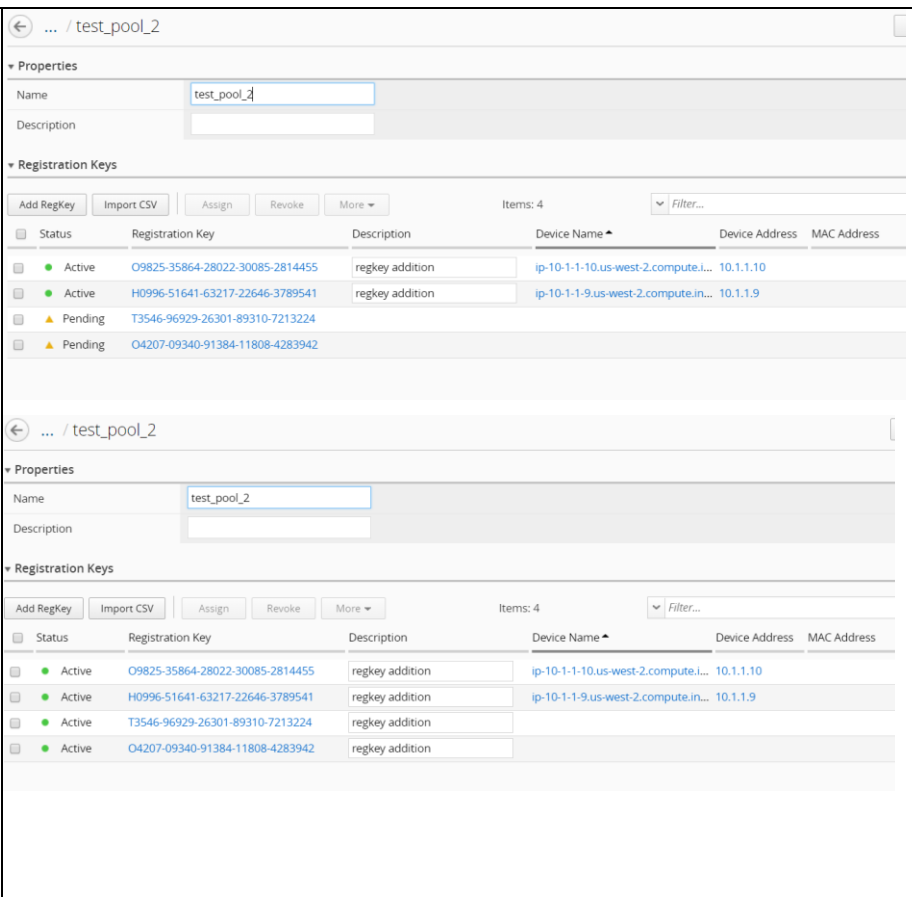
Import a list of new reg keys into an existing license pool in BIG-IQ:

- Modify in "extra variables" window of the job template:  
`regkey_pool: "test_pool_2"`  
`regkey_list:`  
`- '04207-09340-91384-11808-4283942'`  
`- '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\\_active\\_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.



... / test\_pool\_2

Properties

Name: test\_pool\_2

Description:

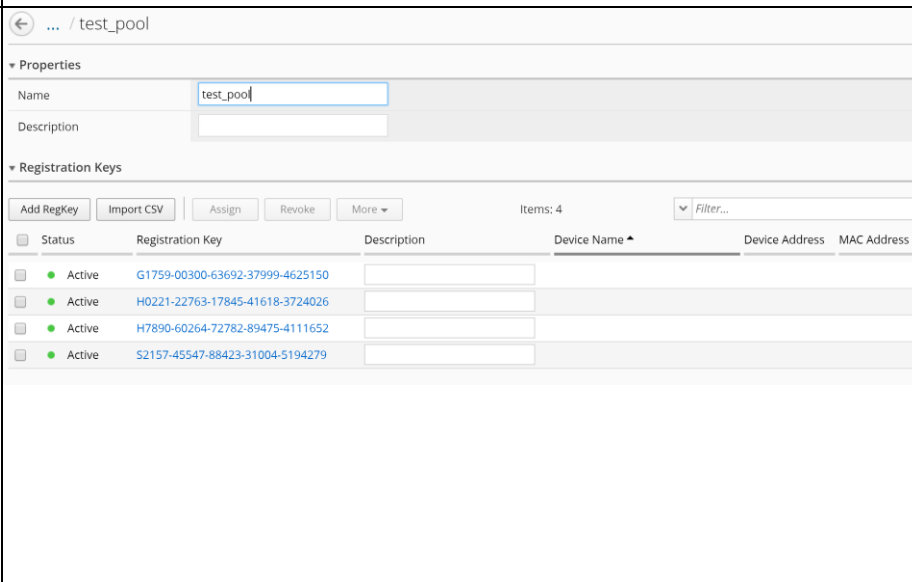
Registration Keys

Add RegKey Import CSV Assign Revoke More ▾ Items: 4 Filter...

Status	Registration Key	Description	Device Name	Device Address	MAC Address
Active	09825-35864-28022-30085-2814455	regkey addition	ip-10-1-1-10.us-west-2.compute.i...	10.1.1.10	
Active	H0996-51641-63217-22646-3789541	regkey addition	ip-10-1-1-9.us-west-2.compute.in...	10.1.1.9	
Pending	T3546-96929-26301-89310-7213224				
Pending	04207-09340-91384-11808-4283942				

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



... / test\_pool

Properties

Name: test\_pool

Description:

Registration Keys

Add RegKey Import CSV Assign Revoke More ▾ Items: 4 Filter...

Status	Registration Key	Description	Device Name	Device Address	MAC Address
Active	G1759-00300-63692-37999-4625150				
Active	H0221-22763-17845-41618-3724026				
Active	H7890-60264-72782-89475-4111652				
Active	S2157-45547-88423-31004-5194279				

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

← ... / test\_pool\_2

▼ Properties

Name: test\_pool\_2

Description:

▼ Registration Keys

Add RegKey Import CSV Assign Revoke More ▾ Items: 4 Filter...

<input type="checkbox"/>	Status	Registration Key	Description	Device Name ▲	Device Address	MAC Address
<input type="checkbox"/>	Active	O9825-35864-28022-30085-2814455	regkey addition	ip-10-1-1-10.us-west-2.compute.i...	10.1.1.10	
<input type="checkbox"/>	Active	H0996-51641-63217-22646-3789541	regkey addition	ip-10-1-1-9.us-west-2.compute.in...	10.1.1.9	
<input type="checkbox"/>	Active	T3546-96929-26301-89310-7213224	regkey addition			
<input type="checkbox"/>	Active	O4207-09340-91384-11808-4283942	regkey addition			

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](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>