April 25, 2018

Oracle Quarterly Technology Update: Ansible & F5

PRESENTED BY:

Chad Jenison, Senior Systems Engineer

WE MAKE APPS GO FASTER. SMARTER. SAFER.

Agenda

- **Ansible Intro**
- **Notes on Ansible F5 Specifics**
- **F5 Examples**
  - Set NTP Configuration
  - UCS Creation and archiving
  - Push new SSL Cert/Key/Profile to BIG-IP
  - ConfigSync
  - LTM Virtual/Pool
- **Paradigms for Ansible Adoption/Usage**
- **Conclusion**

Ansible (software)

From Wikipedia, the free encyclopedia

This article is about the software named Ansible. For other uses, see Ansible (disambiguation).

This article has multiple issues. Please help improve it or discuss these issues on the talk page. [show]

(Learn how and when to remove these template messages)

Ansible is software that automates software provisioning, configuration management, and application deployment.[2]

Michael DeHaan, the author of the provisioning server application Cobbler and co-author of the Func framework for remote administration, developed the platform.[3] It is included as part of the Fedora distribution of Linux, owned by Red Hat Inc., and is also available for Red Hat Enterprise Linux, CentOS, Scientific Linux and Oracle Linux via Extra Packages for Enterprise Linux (EPEL) as well as for other operating systems.[4]

Ansible, Inc. (originally AnsibleWorks, Inc.) was the company set up to commercially support and sponsor Ansible.[5][6] Red Hat acquired Ansible in October 2015.[7][8]

The name "Ansible" references a fictional instantaneous hyperspace communication system (as featured in Orson Scott Card's *Ender's Game* (1985),[9][10] and originally conceived by Ursula K. Le Guin for her novel *Rocannon's World* (1966)).[11]

Contents [hide]

Ansible

ANSIBLE

Original author(s)	Michael DeHaan
Developer(s)	Ansible Community / Ansible Inc. / Red Hat Inc.
Initial release	February 20, 2012; 6 years ago
Stable release	2.5.1 / April 19, 2018; 4 days ago[1]
Repository	https://github.com/ansible/ansible
Development status	Active
Written in	Python, PowerShell
Operating system	Linux, Unix-like, Windows
Available in	English
Type	Configuration management, Infrastructure as Code, Orchestration engine
License	GNU General Public License
Website	www.ansible.com

Ansible Intro

- Ansible is a configuration management solution
- Ansible is agent-less, but does execute some workload on hosts that it is managing; it copies and executes that workload via ssh.
- Ansible gets its power and ease of use from Modules

Debug Log of Ansible Execution (executing module on remote node)

Ansible Terms: Tasks, Plays, Playbooks

- **Playbooks: YAML file containing at least one “Play”**
- **Play: A set of tasks (at least one task in a play)**
- **Task: a specific step using an Ansible module**

```
- name: Example Play
  hosts: bigipve

  tasks:

      - name: Example Task
        bigip_facts:
            server: "{{ inventory_hostname }}"
            user: "admin"
            password: "admin"
            include: system_info,device
            validate_certs: "false"
        delegate_to: localhost
```

Ansible Modules

- Ansible's power comes from a library of modules
- Example core modules: "command", "copy", "fetch", "get_url"
- Example non-core modules: "bigip_command", "bigip_facts", "bigip_pool", "ios_facts", "docker_container"
- Broad coverage for numerous networking, infrastructure and cloud providers

http://docs.ansible.com/ansible/latest/modules/list_of_all_modules.html

bigip_virtual_server - Manage LTM virtu

New in version 2.1.

Synopsis

- Manage LTM virtual servers on a BIG-IP.

Requirements

The below requirements are needed on the host that executes this module.

- f5-sdk >= 3.0.9
- netaddr

Parameters

Parameter	Choices/Defaults	
default_persistence_profile		Default Profile which manages the session persistence. If you want to remove the existing default persistence profile, specify an empty value; `""`. See the documentation for an example.
description		Virtual server description.
destination required		Destination IP of the virtual server. Required when `state` is `present` and virtual server does not exist. aliases: address, ip
disabled_vlans (added in 2.5)		List of VLANs to be disabled. If the partition is not specified in the VLAN, then the `partition` option of this module will be used. This parameter is mutually exclusive with the `enabled_vlans` parameters.
enabled_vlans (added in 2.2)		List of VLANs to be enabled. When a VLAN named `all` is used, all VLANs will be allowed. VLANs can be specified with or without the leading partition. If the partition is not specified in the VLAN, then the `partition` option of this module will be used. This parameter is mutually exclusive with the `disabled_vlans` parameter.

Notes

Note

- Requires BIG-IP software version >= 11
- Requires the netaddr Python package on the host. This is as easy as pip install netaddr.
- For more information on using Ansible to manage F5 Networks devices see https://www.ansible.com/integrations/networks/f5.
- Requires the f5-sdk Python package on the host. This is as easy as *pip install f5-sdk*.

Examples

```
- name: Modify Port of the Virtual Server
  bigip_virtual_server:
    server: lb.mydomain.net
    user: admin
    password: secret
    state: present
    partition: Common
    name: my-virtual-server
    port: 8080
  delegate_to: localhost

- name: Delete virtual server
  bigip_virtual_server:
    server: lb.mydomain.net
    user: admin
    password: secret
    state: absent
    partition: Common
    name: my-virtual-server
  delegate_to: localhost
```

Idempotence

- Big word, pretty simple meaning.
- In configuration management context: a task is idempotent if executing it repeatedly (either the first time when no config is in place or on subsequent executions) produces the same result.
- In short, an Ansible task will get the system to the desired state whether or not the system already has the configuration in place and it is safe to execute repeatedly.
- Most modules provide idempotence; “command” modules usually do not

Inventory and Groups

- A critically important concept in Ansible is the “Inventory” which is usually a static text file (but can be dynamic) in either “INI” or “YAML” format
- Inventory contains the nodes/devices you’re managing with Ansible; referred to as “hosts”
- Examples:
  - all F5 devices in your organization
  - all F5 and other networking devices in your organization
  - all servers and networking devices in your organization
- The inventory will contain IP or hostnames for hosts
- Hosts can be in one or more groups and groups can be hierarchical (e.g. bigip group contains subgroups: appliance and ve with each subgroup having nodes in them)
- Variables can be assigned in the inventory file and they can be associated with groups (at any level) and hosts. During the variable merge process, generally, host variables would override sub-group and then group variables with the same name.

```
all:
  children:
    bigip:
      children:
        bigipve:
          hosts:
            192.168.133.133:
        bigip4200:
          hosts:
            10.192.87.20:
        bigip10200:
          hosts:
            10.192.87.60:
            10.192.87.61:
      vars:
        user: admin
        password: admin
        bigip_username: admin
        bigip_password: admin
        ansible_user: root
        ansible_ssh_pass: default
        validate_certs: "false"
```

Ad-Hoc Commands

```
[cjenison@localhost f5_ansible_demo]$ ansible -i inventory.yaml bigip -a "tmsh show sys clock"
192.168.133.133 | SUCCESS | rc=0 >>
------------------------------
Sys::Clock
------------------------------
Mon Apr 23 10:07:45 CDT 2018

10.192.87.20 | SUCCESS | rc=0 >>
------------------------------
Sys::Clock
------------------------------
Mon Apr 23 08:29:51 PDT 2018

10.192.87.60 | SUCCESS | rc=0 >>
------------------------------
Sys::Clock
------------------------------
Mon Apr 23 08:29:52 PDT 2018

10.192.87.61 | SUCCESS | rc=0 >>
------------------------------
Sys::Clock
------------------------------
Mon Apr 23 08:29:52 PDT 2018
```

Other Ansible Topics to Investigate

- Ansible Roles (another abstraction concept)
- Ansible Vault (solution for storing passwords and other sensitive data safely)
- Ansible Tower (web-based Ansible server) - https://www.ansible.com/products/tower
- Ansible Galaxy (Community developed roles)

Important Points about F5 Ansible Modules

- 100+ F5 Ansible Modules (covering BIG-IP Platform, LTM, GTM, ASM, AFM and BIG-IQ)
- Dependencies: F5 Ansible Modules are dependent on Python Modules including f5-sdk, suds, bigsuds and their pre-requisites
- Most Modules do not execute on BIG-IP, but are delegated – execute on Ansible controller
- You'll see "delegate_to: localhost" and "server: " key indicates which BIG-IP system is being controlled
- The f5-sdk module and bigsuds are communicating to the BIG-IP using HTTPS

Examples

Available at: https://github.com/cjenison/f5_ansible_demo

Making a global change on all BIG-IPs

- Using Ansible is a good fit and easy solution for something that you want to push out to all BIG-IP systems in your organization
- Examples: Device Settings (e.g. NTP configuration), LTM Profiles, Certs and Keys, Users

```
- name: Set NTP Server to pool.ntp.org
  hosts: bigip

  tasks:
      - name: Set NTP Server to pool.ntp.org
        bigip_device_ntp:
            server: "{{ inventory_hostname }}"
            ntp_servers:
              - pool.ntp.org
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            validate_certs: "false"
        delegate_to: localhost
~
```

Execution and Result

```
[cjenison@localhost f5_ansible_demo]$ ansible-playbook -i inventory.yaml set_ntp.yaml

PLAY [Set NTP Server to pool.ntp.org] ******************************************

TASK [Gathering Facts] *********************************************************
ok: [192.168.133.133]
ok: [10.192.87.20]
ok: [10.192.87.60]
ok: [10.192.87.61]

TASK [Set NTP Server to pool.ntp.org] ******************************************
[DEPRECATION WARNING]: Param 'user' is deprecated. See the module docs for more informa
 2.9. Deprecation warnings can be disabled by setting deprecation_warnings=False in ans
[DEPRECATION WARNING]: Param 'password' is deprecated. See the module docs for more inf
version 2.9. Deprecation warnings can be disabled by setting deprecation_warnings=False
[DEPRECATION WARNING]: Param 'validate_certs' is deprecated. See the module docs for mo
in version 2.9. Deprecation warnings can be disabled by setting deprecation_warnings=Fa
[DEPRECATION WARNING]: Param 'server' is deprecated. See the module docs for more infor
version 2.9. Deprecation warnings can be disabled by setting deprecation_warnings=False
ok: [192.168.133.133 -> localhost]
ok: [10.192.87.60 -> localhost]
changed: [10.192.87.20 -> localhost]
changed: [10.192.87.61 -> localhost]

PLAY RECAP *********************************************************************
10.192.87.20               : ok=2    changed=1    unreachable=0    failed=0
10.192.87.60               : ok=2    changed=0    unreachable=0    failed=0
10.192.87.61               : ok=2    changed=1    unreachable=0    failed=0
192.168.133.133            : ok=2    changed=0    unreachable=0    failed=0

[cjenison@localhost f5_ansible_demo]$
```

```
Apr 25 06:22:02 f5 notice icrd_child[29864]: 01420002:5: AUDIT - pid=29864 user=admin folder=/Common
module=(tmos)# status=[Command OK] cmd_data=modify sys ntp timezone America/Los_Angeles servers repla
ce-all-with { pool.ntp.org } description none include none
```

Create a UCS and store on Ansible Controller

```
- name: Create BIG-IP UCS File and Upload it to server; then delete on BIG-IP
  hosts: bigip

  tasks:
      - name: Get all of the facts from my BIG-IP
        bigip_facts:
            server: "{{ inventory_hostname }}"
            user: "admin"
            password: "admin"
            include: system_info,device
            validate_certs: "false"
        delegate_to: localhost

      - name: Create UCS on BIG-IP
        command: /usr/bin/tmsh save sys ucs {{ system_info.system_information.host_name }}_{{ ansible_date_time.date }}.ucs

      - name: Fetch UCS to Local Machine
        fetch:
          src: /var/local/ucs/{{ system_info.system_information.host_name }}_{{ ansible_date_time.date }}.ucs
          dest: ucs/
          flat: yes

      - name: Delete UCS on BIG-IP
        file:
          state: absent
          path: /var/local/ucs/{{ system_info.system_information.host_name }}_{{ ansible_date_time.date }}.ucs
```

Execution and Result

```
[cjenison@localhost f5_ansible_demo]$ ansible-playbook -i inventory.yaml create_and_upload_ucs.yaml

PLAY [Create BIG-IP UCS File and Upload it to server; then delete on BIG-IP] **********************

TASK [Gathering Facts] ***********************************************************************
ok: [192.168.133.133]
ok: [10.192.87.20]
ok: [10.192.87.60]
ok: [10.192.87.61]

TASK [Get all of the facts from my BIG-IP] ***************************************************
ok: [192.168.133.133 -> localhost]
ok: [10.192.87.60 -> localhost]
ok: [10.192.87.61 -> localhost]
ok: [10.192.87.20 -> localhost]

TASK [Create UCS on BIG-IP] ******************************************************************
changed: [10.192.87.60]
changed: [10.192.87.20]
changed: [10.192.87.61]
changed: [192.168.133.133]

TASK [Fetch UCS to Local Machine] ************************************************************
changed: [192.168.133.133]
changed: [10.192.87.20]
changed: [10.192.87.60]
changed: [10.192.87.61]

TASK [Delete UCS on BIG-IP] ******************************************************************
changed: [192.168.133.133]
changed: [10.192.87.20]
changed: [10.192.87.60]
changed: [10.192.87.61]

PLAY RECAP ***********************************************************************************
10.192.87.20               : ok=5    changed=3    unreachable=0    failed=0
10.192.87.60               : ok=5    changed=3    unreachable=0    failed=0
10.192.87.61               : ok=5    changed=3    unreachable=0    failed=0
192.168.133.133            : ok=5    changed=3    unreachable=0    failed=0

[cjenison@localhost f5_ansible_demo]$ ls ucs
bigip10200f1.sjc.f5net.net_2018-04-25.ucs  daytona.chadlab.net_2018-04-25.ucs
bigip10200f2.sjc.f5net.net_2018-04-25.ucs  f5.dhd.bd.f5.local_2018-04-25.ucs
[cjenison@localhost f5_ansible_demo]$
```

Pushing an SSL Key, Cert and Client SSL profile to a BIG-IP

```yaml
- name: Push SSL Key, Cert and Client-SSL profile to BIG-IP
  hosts: bigip10200

  tasks:
      - name: Push SSL Key
        bigip_ssl_key:
            name: chad
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            content: "{{ lookup('file', 'ssl/chad-key.pem') }}"
            validate_certs: "false"
        delegate_to: localhost

      - name: Push SSL Cert
        bigip_ssl_certificate:
            name: chad
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            content: "{{ lookup('file', 'ssl/chad-cert.pem') }}"
            validate_certs: "false"
        delegate_to: localhost

      - name: Create Client SSL Profile
        bigip_profile_client_ssl:
            name: chad-clientssl
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            ciphers: "DEFAULT"
            cert_key_chain:
              - cert: chad
                key: chad
            validate_certs: "false"
        delegate_to: localhost
```

Execution and Result

```
[cjenison@localhost f5_ansible_demo]$ ansible-playbook -i inventory.yaml key_cert_clientssl.yaml

PLAY [Push SSL Key, Cert and Client-SSL profile to BIG-IP] ************************************

TASK [Gathering Facts] ************************************************************************
ok: [192.168.133.133]
ok: [10.192.87.20]
ok: [10.192.87.61]
ok: [10.192.87.60]

TASK [Push SSL Key] ***************************
ok: [10.192.87.20 -> localhost]
changed: [192.168.133.133 -> localhost]
changed: [10.192.87.60 -> localhost]
changed: [10.192.87.61 -> localhost]

TASK [Push SSL Cert] **************************
changed: [192.168.133.133 -> localhost]
ok: [10.192.87.20 -> localhost]
changed: [10.192.87.61 -> localhost]
changed: [10.192.87.60 -> localhost]

TASK [Create Client SSL Profile] ******
changed: [192.168.133.133 -> localhost]
changed: [10.192.87.20 -> localhost]
changed: [10.192.87.60 -> localhost]
changed: [10.192.87.61 -> localhost]

PLAY RECAP ************************************************************************************
10.192.87.20               : ok=4    changed=1    unreachable=0    failed=0
10.192.87.60               : ok=4    changed=3    unreachable=0    failed=0
10.192.87.61               : ok=4    changed=3    unreachable=0    failed=0
192.168.133.133            : ok=4    changed=3    unreachable=0    failed=0
```

```
Apr 25 06:39:48 bigip10200f1 info rest(pam_audit)[13219]: 01070417:6: AUDIT - user admin - RAW: rest(pam_audit): user=admin(admin) partition=[Al
l] level=Administrator tty=(unknown) host=(unknown) attempts=1 start="Wed Apr 25 06:39:48 2018" end="Wed Apr 25 06:39:48 2018".
Apr 25 06:39:48 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7407278-2 - object 0 -
 create { certificate_key_file_object { certificate_key_file_object_name "/Common/chad.key" certificate_key_file_object_checksum "SHA1:1703:ad52
0723220bb73c46f0af86e279ffac7623d430" certificate_key_file_object_local_path "/var/system/tmp/tmsh/fh5gJi/data" certificate_key_file_object_sour
ce_path "file:///var/config/rest/downloads/chad.key" } } [Status=Command OK]
Apr 25 06:39:48 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK]
cmd_data=create sys file ssl-key /Common/chad.key { source-path file:///var/config/rest/downloads/chad.key }
Apr 25 06:39:51 bigip10200f1 info rest(pam_audit)[13230]: 01070417:6: AUDIT - user admin - RAW: rest(pam_audit): user=admin(admin) partition=[Al
l] level=Administrator tty=(unknown) host=(unknown) attempts=1 start="Wed Apr 25 06:39:51 2018" end="Wed Apr 25 06:39:51 2018".
Apr 25 06:39:51 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7407375-2 - object 0 -
 create { certificate_file_object { certificate_file_object_name "/Common/chad.crt" certificate_file_object_checksum "SHA1:1398:b41952f85a271908
8899005e28ae277c96412276" certificate_file_object_local_path "/var/system/tmp/tmsh/kAVmH1/data" certificate_file_object_source_path "file:///var
/config/rest/downloads/chad.crt" } } [Status=Command OK]
Apr 25 06:39:51 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK]
cmd_data=create sys file ssl-cert /Common/chad.crt { source-path file:///var/config/rest/downloads/chad.crt }
Apr 25 06:39:53 bigip10200f1 info rest(pam_audit)[13242]: 01070417:6: AUDIT - user admin - RAW: rest(pam_audit): user=admin(admin) partition=[Al
l] level=Administrator tty=(unknown) host=(unknown) attempts=1 start="Wed Apr 25 06:39:53 2018" end="Wed Apr 25 06:39:53 2018".
Apr 25 06:39:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7407472-2 - object 0 -
 create { profile_clientssl { profile_clientssl_name "/Common/chad-clientssl" profile_clientssl_ciphers "DEFAULT" profile_clientssl_config_sourc
e 0 } } [Status=Command OK]
Apr 25 06:39:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7407472-3 - object 0 -
 create { clientssl_certkeychain { clientssl_certkeychain_clientssl "/Common/chad-clientssl" clientssl_certkeychain_name "chad" clientssl_certke
ychain_cert "/Common/chad.crt" clientssl_certkeychain_key "/Common/chad.key" clientssl_certkeychain_chain "" } } [Status=Command OK]
Apr 25 06:39:53 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK]
cmd_data=create ltm profile client-ssl /Common/chad-clientssl { cert-key-chain add { chad { cert /Common/chad.crt chain none key /Common/chad.ke
y } } ciphers DEFAULT }
```

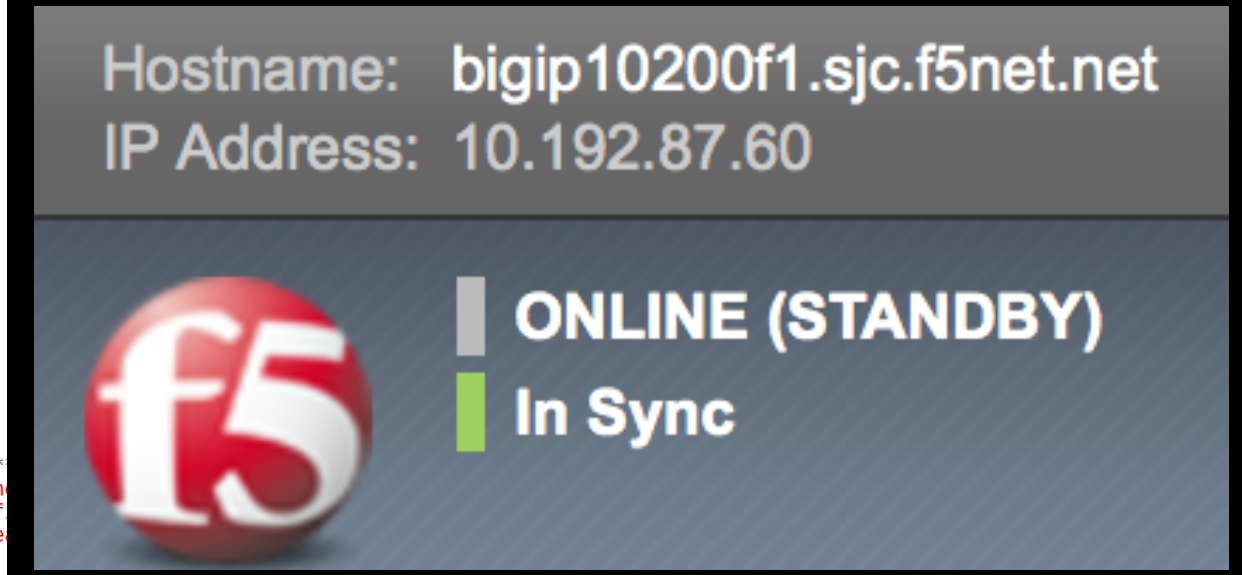
Performing a ConfigSync on BIG-IPs

```yaml
- name: Gather Facts and then pull most recent config
  hosts: bigip

  tasks:
      - name: Get all of the facts from my BIG-IP
        bigip_facts:
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            include: device_group
            validate_certs: "false"
        delegate_to: localhost

      - name: determine sync-failover device group
        set_fact:
          sync_failover_dg: "{{ item }}"
        when: device_group[item]['type'] == "DGT_FAILOVER"
        with_items: "{{ device_group.keys() }}"
        delegate_to: localhost

      - name: Pull Config From Group
        bigip_configsync_action:
            server: "{{ inventory_hostname }}"
            sync_most_recent_to_device: yes
            device_group: "{{ sync_failover_dg.split('/')[2] }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            validate_certs: "false"
        delegate_to: localhost
```

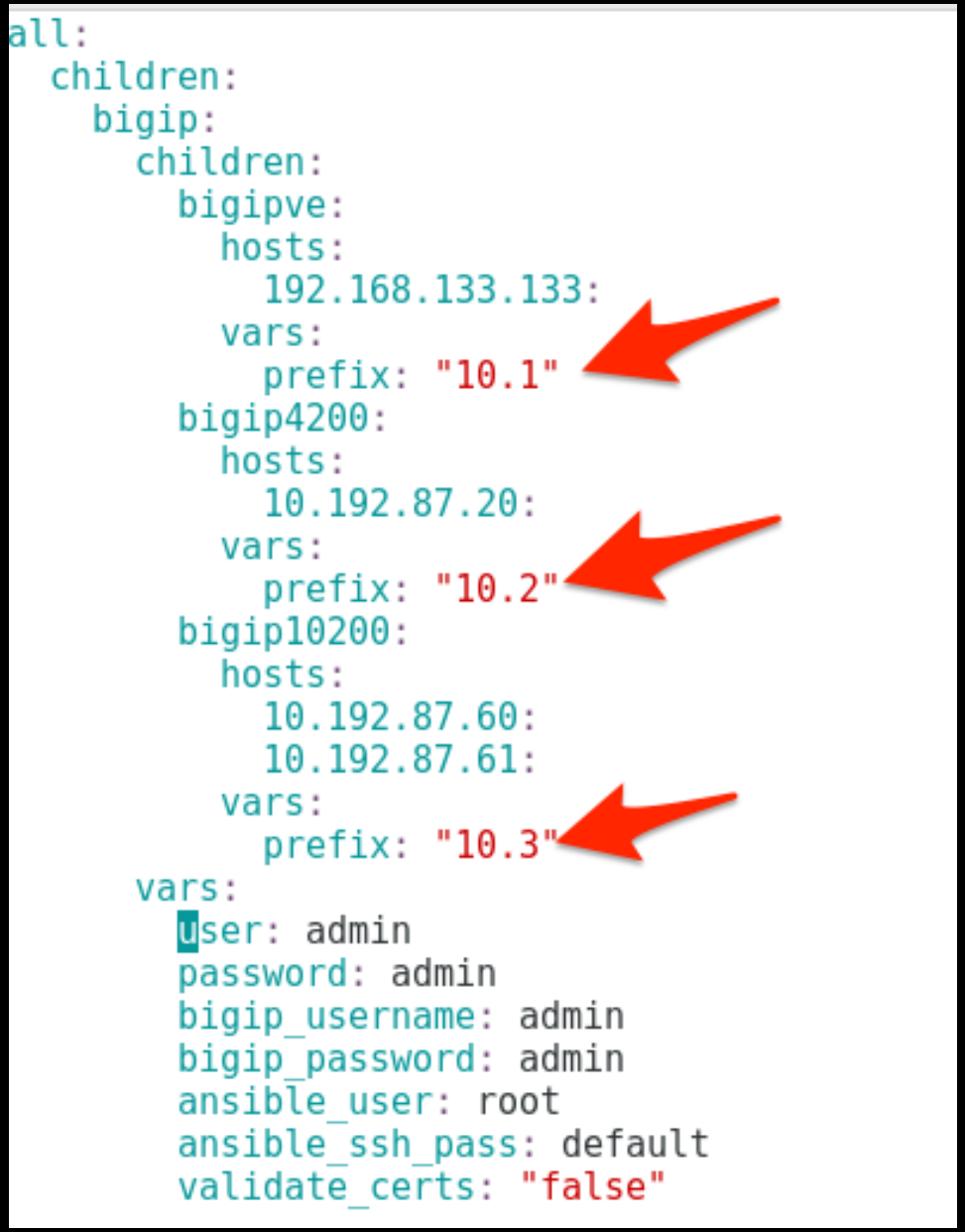
Execution and Result

```
[cjenison@localhost f5_ansible_demo]$ ansible-playbook -i inventory.yaml config_sync.yaml

PLAY [Gather Facts and then pull most recent config] ***********************************

TASK [Gathering Facts] *****************************************************************
ok: [192.168.133.133]
ok: [10.192.87.20]
ok: [10.192.87.60]
ok: [10.192.87.61]

TASK [Get all of the facts from my BIG-IP] *********************************************
ok: [192.168.133.133 -> localhost]
ok: [10.192.87.20 -> localhost]
ok: [10.192.87.60 -> localhost]
ok: [10.192.87.61 -> localhost]

TASK [determine sync-failover device group] ********************************************
skipping: [192.168.133.133] => (item=/Common/device_trust_group)
skipping: [192.168.133.133] => (item=/Common/datasync-device-daytona.chadlab.net-dg)
skipping: [192.168.133.133] => (item=/Common/gtm)
skipping: [192.168.133.133] => (item=/Common/datasync-global-dg)
skipping: [10.192.87.20] => (item=/Common/device_trust_group)
skipping: [10.192.87.20] => (item=/Common/gtm)
skipping: [10.192.87.20] => (item=/Common/datasync-device-f5.dhd.bd.f5.local-dg)
skipping: [10.192.87.20] => (item=/Common/datasync-global-dg)
skipping: [10.192.87.60] => (item=/Common/device_trust_group)
skipping: [10.192.87.60] => (item=/Common/datasync-device-bigip10200f2.sjc.f5net.net-dg)
skipping: [10.192.87.60] => (item=/Common/datasync-device-bigip10200f1.sjc.f5net.net-dg)
ok: [10.192.87.60 -> localhost] => (item=/Common/sync-failover)
skipping: [10.192.87.61] => (item=/Common/device_trust_group)
skipping: [10.192.87.61] => (item=/Common/datasync-device-bigip10200f2.sjc.f5net.net-dg)
skipping: [10.192.87.61] => (item=/Common/datasync-device-bigip10200f1.sjc.f5net.net-dg)
ok: [10.192.87.61 -> localhost] => (item=/Common/sync-failover)
skipping: [10.192.87.61] => (item=/Common/gtm)
skipping: [10.192.87.61] => (item=/Common/datasync-global-dg)
skipping: [10.192.87.60] => (item=/Common/gtm)
skipping: [10.192.87.60] => (item=/Common/datasync-global-dg)

TASK [Pull Config From Group] **********************************************************
fatal: [192.168.133.133]: FAILED! => {"msg": "The task includes an option with an undefin
g' is undefined\n\nThe error appears to have been in '/home/cjenison/f5_ansible_demo/conf
e elsewhere in the file depending on the exact syntax problem.\n\nThe offending line appe
om Group\n        ^ here\n"}
fatal: [10.192.87.20]: FAILED! => {"msg": "The task includes an option with an undefined
is undefined\n\nThe error appears to have been in '/home/cjenison/f5_ansible_demo/config_
lsewhere in the file depending on the exact syntax problem.\n\nThe offending line appears
Group\n        ^ here\n"}
changed: [10.192.87.61 -> localhost]
changed: [10.192.87.60 -> localhost]
```

```
Apr 25 06:50:34 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK]
cmd_data=run cm config-sync from-group sync-failover
Apr 25 06:51:23 bigip10200f1 info sshd(pam_audit)[13806]: 01070417:6: AUDIT - user root - RAW: sshd(pam_audit): user=root(root) partition=[All]
```



Add Network Prefix (First two octets) as variables in inventory



Creating LTM Pools and Virtuals (Playbook)

```yaml
- name: Create LTM Pool, Add 2 Members and then create 2 LTM Virtuals (80/443) that reference pool
  hosts: bigip

  tasks:
      - name: Create Pool example-pool
        bigip_pool:
            name: example-pool
            partition: Common
            lb_method: round-robin
            monitors:
              - tcp_half_open
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            validate_certs: "false"
        delegate_to: localhost

      - name: Add Members to Pool
        bigip_pool_member:
            pool: example-pool
            partition: Common
            host: "{{ hostvars[inventory_hostname].prefix }}.{{ item.addr }}"
            port: 80
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            validate_certs: "false"
        with_items:
          - addr: "2.1"
          - addr: "2.2"
        delegate_to: localhost

      - name: Create port 80 virtual server
        bigip_virtual_server:
            name: example-vs-http
            partition: Common
            destination: "{{ hostvars[inventory_hostname].prefix }}.1.1"
            port: 80
            profiles:
              - http
              - tcp
            irules:
              - _sys_https_redirect
            server: "{{ inventory_hostname }}"
            user: "{{ hostvars[inventory_hostname].user }}"
            password: "{{ hostvars[inventory_hostname].password }}"
            state: present
            validate_certs: "false"
        delegate_to: localhost
```

Variable Substitution for Prefix and Host Addresses

Simple "Looping" construct: "with_items"

Execution and Result

```
[cjenison@localhost f5_ansible_demo]$ ansible-playbook -i inventory-with-nets.yaml ltm_pool_and_virtual.yaml

PLAY [Create LTM Pool, Add 2 Members and then create 2 LTM Virtuals (80/443) that reference pool] **********

TASK [Gathering Facts] *********************************************************************************
ok: [192.168.133.133]
ok: [10.192.87.20]
ok: [10.192.87.61]
ok: [10.192.87.60]

TASK [Create Pool example-pool] ************************************************
changed: [192.168.133.133 -> localhost]
changed: [10.192.87.60 -> localhost]
changed: [10.192.87.61 -> localhost]
changed: [10.192.87.20 -> localhost]

TASK [Add Members to Pool] *****************************************************
changed: [192.168.133.133 -> localhost] => (item={u'addr': u'2.1'})
changed: [10.192.87.60 -> localhost] => (item={u'addr': u'2.1'})
changed: [10.192.87.20 -> localhost] => (item={u'addr': u'2.1'})
changed: [10.192.87.61 -> localhost] => (item={u'addr': u'2.1'})
changed: [192.168.133.133 -> localhost] => (item={u'addr': u'2.2'})
changed: [10.192.87.20 -> localhost] => (item={u'addr': u'2.2'})
changed: [10.192.87.60 -> localhost] => (item={u'addr': u'2.2'})
changed: [10.192.87.61 -> localhost] => (item={u'addr': u'2.2'})

TASK [Create port 80 virtual server] *******************************************
changed: [192.168.133.133 -> localhost]
changed: [10.192.87.60 -> localhost]
changed: [10.192.87.20 -> localhost]
changed: [10.192.87.61 -> localhost]

TASK [Create port 443 virtual server] ******************************************
changed: [192.168.133.133 -> localhost]
changed: [10.192.87.20 -> localhost]
changed: [10.192.87.61 -> localhost]
changed: [10.192.87.60 -> localhost]

PLAY RECAP *********************************************************************
10.192.87.20               : ok=5    changed=4    unreachable=0    failed=0
10.192.87.60               : ok=5    changed=4    unreachable=0    failed=0
10.192.87.61               : ok=5    changed=4    unreachable=0    failed=0
192.168.133.133            : ok=5    changed=4    unreachable=0    failed=0

[cjenison@localhost f5_ansible_demo]$
```

```
Apr 25 06:55:30 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428417-2 - object 0 - create { pool { pool_name "/Common/example-pool" pool_lb_mode 0 pool_monitor_rule "/Common/tcp_half_open" } } [Status=Command OK]
Apr 25 06:55:30 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK] cmd_data=create ltm pool /Common/example-pool { load-balancing-mode round-robin monitor /Common/tcp_half_open }
Apr 25 06:55:41 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client iControlSOAP, user admin - transaction #7428611-3 - object 0 - create { pool_member { pool_member_pool_name "/Common/example-pool" pool_member_node_name "/Common/10.3.2.1" pool_member_port 80 } } [Status=Command OK]
Apr 25 06:55:47 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client iControlSOAP, user admin - transaction #7428730-3 - object 0 - create { pool_member { pool_member_pool_name "/Common/example-pool" pool_member_node_name "/Common/10.3.2.2" pool_member_port 80 } } [Status=Command OK]
Apr 25 06:55:50 bigip10200f1 info rest(pam_audit)[14194]: 01070417:6: AUDIT - user admin - RAW: rest(pam_audit): user=admin(admin) partition=[All] level=Administrator tty=(unknown) host=(unknown) attempts=1 start="Wed Apr 25 06:55:50 2018" end="Wed Apr 25 06:55:50 2018".
Apr 25 06:55:50 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428875-2 - object 0 - create { virtual_server { virtual_server_name "/Common/example-vs-http" virtual_server_va_name "/Common/10.3.1.1" virtual_server_port 80 } } [Status=Command OK]
Apr 25 06:55:50 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428875-3 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name "/Common/example-vs-http" virtual_server_profile_profile_name "/Common/http" virtual_server_profile_profile_context 0 } } [Status=Command OK]
Apr 25 06:55:50 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428875-4 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name "/Common/example-vs-http" virtual_server_profile_profile_name "/Common/tcp" virtual_server_profile_profile_context 0 } } [Status=Command OK]
Apr 25 06:55:50 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428875-5 - object 0 - obj_delete { virtual_server_rule { virtual_server_rule_virtual_server_name "/Common/example-vs-http" } } [Status=Command OK]
Apr 25 06:55:50 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7428875-6 - object 0 - create_if { virtual_server_rule { virtual_server_rule_virtual_server_name "/Common/example-vs-http" virtual_server_rule_rule_name "/Common/_sys_https_redirect" virtual_server_rule_priority 0 } } [Status=Command OK]
Apr 25 06:55:51 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK] cmd_data=create ltm virtual /Common/example-vs-http { rules { /Common/_sys_https_redirect } destination /Common/10.3.1.1:80 profiles add { http { context all } tcp { context all } } }
Apr 25 06:55:53 bigip10200f1 info rest(pam_audit)[14205]: 01070417:6: AUDIT - user admin - RAW: rest(pam_audit): user=admin(admin) partition=[All] level=Administrator tty=(unknown) host=(unknown) attempts=1 start="Wed Apr 25 06:55:53 2018" end="Wed Apr 25 06:55:53 2018".
Apr 25 06:55:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7429042-2 - object 0 - create { virtual_server { virtual_server_name "/Common/example-vs-https" virtual_server_va_name "/Common/10.3.1.1" virtual_server_port 443 virtual_server_default_pool "/Common/example-pool" virtual_server_source_address_translation_type 3 } } [Status=Command OK]
Apr 25 06:55:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7429042-3 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name "/Common/example-vs-https" virtual_server_profile_profile_name "/Common/clientssl" virtual_server_profile_profile_context 1 } } [Status=Command OK]
Apr 25 06:55:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7429042-4 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name "/Common/example-vs-https" virtual_server_profile_profile_name "/Common/http" virtual_server_profile_profile_context 0 } } [Status=Command OK]
Apr 25 06:55:53 bigip10200f1 notice mcpd[9539]: 01070417:5: AUDIT - client tmsh, tmsh-pid-7555, user admin - transaction #7429042-5 - object 0 - create { virtual_server_profile { virtual_server_profile_vs_name "/Common/example-vs-https" virtual_server_profile_profile_name "/Common/tcp" virtual_server_profile_profile_context 0 } } [Status=Command OK]
Apr 25 06:55:53 bigip10200f1 notice icrd_child[7555]: 01420002:5: AUDIT - pid=7555 user=admin folder=/Common module=(tmos)# status=[Command OK] cmd_data=create ltm virtual /Common/example-vs-https { destination /Common/10.3.1.1:443 profiles add { clientssl { context clientside } http { context all } tcp { context all } } source-address-translation { type automap } pool /Common/example-pool }
Apr 25 06:56:27 bigip10200f1 info sshd(pam_audit)[14058]: 01070417:6: AUDIT - user root - RAW: sshd(pam_audit): user=root(root) partition=[All] level=Administrator tty=ssh host=172.18.77.254 attempts=1 start="Wed Apr 25 06:55:23 2018" end="Wed Apr 25 06:56:27 2018".
```

Pool

Pool Members

Create Virtual Server 1

Create Virtual Server 2

Ways of Using Ansible

As a Productivity Tool

- An individual engineer can use Ansible as a tool for making one-time changes or recurring tasks that would otherwise be done manually
- Bulk operations across many BIG-IP systems
- Good for “standardized” collections of configuration changes
- Bigip_command module can be helpful if you are already familiar with TMSH syntax

Ansible as Source of Truth (but with static inventory and playbooks)

- Ansible is “mandatory” for housing configuration
- ”Manual” config changes outside of Ansible break the paradigm
- Ansible Inventory, Roles and Playbooks would hold all information that exists in BIG-IP configurations
- Ansible files would typically be stored on a centralized server and managed with revision control

Ansible for managing devices, informed by dynamic/upstream source of truth

- Ansible would consume or be called by upstream system (e.g. customer or service provisioning portal)
- Standardization of configuration is important
- This approach represents a high degree of automation and will generally address F5 and other infrastructure components (whether on-prem or cloud-based)

Conclusion

- Ansible is a powerful tool with broad support from F5 and other vendors
- Intent is to simplify automation and avoid need for programming skills
- Does have a learning curve and represents a change from doing things “manually” or “directly”
- Developing Ansible competencies can pave the way for higher levels of Ansible adoption and automation

QUESTIONS?

Thank You

WE MAKE APPS

FASTER. SMARTER. SAFER.