Ansible Network Automation

Kyle Benson
Global Alliance Architect | Red Hat
October 2018

Kevin Delgadillo
Principal Product Manager – Orchestration & Automation | F5
AGENDA

INTRODUCTION
1. Red Hat Is A Network Company(ish)

STRATEGY
2. Ansible Network

ACTIONS
3. Partner Highlight: F5
ANSIBLE
Evolving Network Operations

...is NOT about

- SDN
- Host Networking
- Virtual Networking
- Control Plane or Data Plane
Evolving Network Operations

... IS about:

- Operation
- Augmentation
- Consumption
- Simplification
MODERNIZE NETWORK OPERATIONS

TRADITIONAL NETWORK OPERATIONS

EVOLVED NETWORK OPERATIONS
Control Node: Any client system (server, laptop, VM) running Linux or Mac OSX

Managed Nodes (Inventory): A collection of endpoints being managed via SSH or API.

Modules: Handles execution of remote system commands
ANSIBLE NETWORK AUTOMATION

50 Network Platforms
700+ Network Modules
12* Galaxy Network Roles

ansible.com/for/networks
galaxy.ansible.com/ansible-network

Ansible Network modules comprise 1/3 of all modules that ship with Ansible Engine
NETWORK AUTOMATION PROGRESS

- **2.1** May 2016: 7 Platforms, 28 Modules
- **2.2** Oct 2016: 17 Platforms, 141 Modules
- **2.3** Apr 2017: 29 Platforms, 267 Modules
- **2.4** Sep 2017: 33 Platforms, 463 Modules
- **2.5** Mar 2018: 40 Platforms, 572 Modules
- **2.6** Jun 2018: 45 Platforms, 639 Modules
- **2.7** Oct 2018: 50 Platforms, 700 Modules
ANSIBLE NETWORK ENGINE
**Ansible Network Engine** is a set of consumable functions distributed as Ansible Roles that have been optimized for automating the bootstrap, provisioning and configuration management of *network infrastructure* and *multi cloud network connectivity*. 
- Ansible Role decoupled from mainline development branch
- Incubate new capabilities, features functions for interfacing with network devices and models
- Biweekly release cycle - ship early, ship often approach
- Extends the Network Engine role - It’s all just Ansible
- Data driven with focused implementation on operational use case
- Extensible by anyone for any platform / device
ANATOMY OF A FUNCTION

PLAYBOOK

ROLE

FUNCTION DEFINITION

Provider

Provider

Provider
ANSIBLE NETWORK STACK ARCHITECTURE
ANSIBLE NETWORK STACK ARCHITECTURE
ANSIBLE NETWORK STACK ARCHITECTURE

DEVICE | APPLICATION | CLOUD | NETWORK FUNCTIONS

CONFIGURATION | OPERATIONS | NETWORK ENGINE

NETWORK MODULES | CONNECTION PLUGINS (CLI, API, NETCONF) | ANSIBLE ENGINE

VIRTUAL NETWORK | SDN CONTROLLERS | PHYSICAL DEVICES
ANSIBLE NETWORK STACK ARCHITECTURE

DEVICE  APPLICATION  CLOUD  NETWORK FUNCTIONS

CONFIGURATION  OPERATIONS  NETWORK ENGINE

NETWORK MODULES  CONNECTION PLUGINS (CLI, API, NETCONF)  ANSIBLE ENGINE

VIRTUAL NETWORK  SDN CONTROLLERS  PHYSICAL DEVICES
ANSIBLE NETWORK STACK ARCHITECTURE

ANSIBLE TOWER
- Audit
- RBAC
- Credentials
- Inventory
- Web UI
- REST API
- Survey

DEVICE
APPLICATION
CLOUD

NETWORK FUNCTIONS

OPERATOR FOCUSED

CONFIGURATION
OPERATIONS

NETWORK ENGINE

DEVLOPER FOCUSED

NETWORK MODULES

CONNECTION PLUGINS (CLI, API, NETCONF)

ANSIBLE ENGINE

VIRTUAL NETWORK
SDN CONTROLLERS
PHYSICAL DEVICES

#redhat #rhnapp
DEVICE CENTRIC NETWORK AUTOMATION

Build and push device configurations

Automate tactical operations on network devices

APPLICATION CENTRIC NETWORK AUTOMATION

Automate network devices in support of applications

Support direct to device and controller based virtualization

CLOUD CENTRIC NETWORK AUTOMATION

Describe and deploy network connectivity between clouds

Support public/private and/or public/public clouds
DEVICE CENTRIC NETWORK AUTOMATION

- Primary use case for automating network devices today
- Push configurations / Pull facts from network devices
- Any Model, Any Encoding, Any Transport
USE CASE: OPENCONFIG

1. Make sure netconf is enabled
2. Validate schema name
3. Fetch schema and dependencies
4. Validate input against schema
5. Push change to device
Simplify automating the underlay network

Integrate network layering into cloud apps

Control Node

Automate application services - firewalls, load balancers
USE CASE: CLOUD VPN

---

cloud_vpn_name: openstack-vyos-to-aws-vpn
cloud_vpn_responder_type: aws_vpn
cloud_vpn_responder_aws_region: us-west-1
cloud_vpn_initiator_type: openstack_vyos
cloud_vpn_initiator_openstack_cloud: psprygad-cloud
cloud_vpn_initiator_cidr: 192.168.0.0/24
cloud_vpn_initiator_outside_interface: eth0
cloud_vpn_initiator_private_ip: 192.168.0.145
cloud_vpn_initiator_user: admin
cloud_vpn_initiator_ssh_private_key_file: tfd18.pem
cloud_vpn_initiator_key_name: tfd18
cloud_vpn_initiator_image_id: vyos-1.1.8
cloud_vpn_initiator_openstack_external_network: 38.145.32.0/22

---

- hosts: localhost
  connection: local
  gather_facts: no
  tasks:
    - include_role:
        name: cloud-vpn
- Application domain automates infrastructure using Ansible
- Ansible provides consistent abstraction of resources
- Integrate network changes as part of workload definitions
ANSIBLE CERTIFIED
ANSIBLE CERTIFIED

TRUSTED AUTOMATION

FULLY SUPPORTED
PARTNER HIGHLIGHT: F5
Objective: F5 & Ansible

Integrate seamlessly with the entire product line
- BIG-IP
- BIG-IQ

Get customers used to automating & orchestrating everything F5
- Configuration management
- Infrastructure As Code

No more hacky scripts
- Ad-hoc shell scripts are not a good substitute for automation frameworks
- Shell scripts not portable
- Ad-hoc scripts are not natively idempotent
Ansible Solution for F5 BIG-IP

Use Cases

• Deploying L4-L7 application services
• BIG-IP onboarding, including high availability
• Drive Infrastructure as Code migrations
---

- name: Virtual Server config on BIG-IP
  hosts: bigip
  gather_facts: false

  tasks:

  - name: Add Virtual Server on BIG-IP
    bigip_virtual_server:
      server: ""
      user: ""
      password: ""
      name: "http_vs"
      destination: "10.168.90.92"
      port: 80
      pool: "web-pool"
      snat: "automap"
      profiles: "http"
      validate_certs: False
      delegate_to: localhost
Ansible 2.7.0

- Security logging profiles for AFM (HSL)
- DDoS profile create, modify, & apply to virtual
- LTM Persistence Profile
- LTM DNS Monitor

Complete list: https://docs.ansible.com/ansible/latest/modules/list_of_network_modules.html#f5

- LTM DNS Nameservers
- Tunnel Configuration (VXLAN)
- AFM Firewall Rules & Policies
- BIG-IP Software Upgrades
- BIG-IP System Authentication
- And more…
Introducing Application Services 3 Extension (AS3)

- Mechanism for automating & managing L4-L7 configurations on BIG-IP
- Enables complete BIG-IP L4-L7 configuration with single REST API call
- Provides intent-based **declarative** APIs for common use cases (end-state driven)
- Abstracts away all aspects of BIG-IP configuration
- Minimizes required domain expertise
- Node.js iControl LX plug-in → TMOS-independent
- Built for DevOps & integration with CI/CD toolchains
- Moves configuration source-of-truth off of BIG-IP into orchestrator
- Free, supported, & available now: [https://github.com/F5Networks/f5-appsvcs-extension](https://github.com/F5Networks/f5-appsvcs-extension)
Configuring an L4 – L7 app service requires many REST API calls

**Without AS3:** Many Imperative Commands

**With AS3:** One Declarative Statement

Many Imperative Commands

© 2018 F5 Networks
Where Do I Begin?

Automation is not just a tool, it’s a journey it’s a strategy

Learn automation practices
- Super NetOps training courses can help
- Join existing Ansible network automation communities

Start small…
- Create Playbooks that read or check only
- Create simple jobs that eliminate the most annoying network tasks

Automate!
BIG-IP on Ansible Galaxy

Please contribute..
Your BIG-IP roles for community!

https://galaxy.ansible.com/list#/roles?page=1&page_size=10&autocomplete=bigip
COME LEARN MORE!

- ansible.com/products/network-automation
- F5 Documentation
THANKS!