

RED HAT
SUMMIT

SELF-HEALING NETWORKS USING ANSIBLE

Consulting Discovery Zone

Jon Bersuder/Walter Bentley
Automation Practice Architect/Lead
May 4th 2017

Jon Bersuder

Automation Practice Architect

GitHub: b3rn

- Automation Architect
- Automating environments for nearly 15 years
 - Hit parade: Cfengine, Puppet, Chef, dsh, rPath, and for loops and text files
- Ansible user for more than 5 years
- Worked on hundreds of automation engagements (some of which were even successful!)
- Raleigh native
- Been attending TriLUG meetings off and on for almost 15 years

Walter Bentley

Automation Practice Lead

Twitter: @djstayflypro

LinkedIn: <http://goo.gl/r2p21i>

GitHub: wbentley15

Blog: hitchnyc.com

- Over 18 years of IT experience
- New Yexan (a New Yorker living in Texas)
- Cloud Advocate (hybrid is my favorite)
- Author, Knowledge sharer
- Ansible & OpenStack believer
- Amateur Marksman & Motorcyclist
- Always about living life now!

AGENDA

Automation Approach to Identifying Network Failures

- Why Capturing Metrics from Your Network is Important
- Defining Failure States
- Analyzing Failure States in Order to Determine Optimal Remediation
- Auto-Remediation versus Remediation

How Are The Small Guys Doing It?

- Facebook approach
- Google approach
- Microsoft approach
- Netflix approach

Try Not to Boil the Ocean with Ansible

- Best Use Cases for Ansible Core and Ansible Tower
- How Can Consulting Services can help?

AUTOMATION APPROACH TO IDENTIFYING NETWORK FAILURES

Why Capturing Metrics from Your Network is Important

“How do you determine when something is broken? What is considered normal network wise?”

- Capture ‘Good’ state metrics (store them away for analysis)
- Simple network pings can do the trick
- Simple network pings can’t *always* do the trick
- Without benchmarks, you will never know when things are broken



Defining Failure States

- Everyone's 'Failure' states will be different!
- Should be determined based on overall network topology and acceptable network latency
- Seek to be proactive
- Make provisions for hiccups or expected abnormalities

Analyzing Failure States in Order to Determine Optimal Remediation

FAILURE  **GO FIX IT**
EVERYTIME

Automated Remediation versus Remediation

Automated Remediation or Self-Healing: *is when automation responds to alerts or events by executing actions that can prevent or fix the problem*

Remediation: *Human driven action of remedying something, in particular of reversing or stopping damage*

HOW ARE THE “SMALL GUYS” DOING IT?

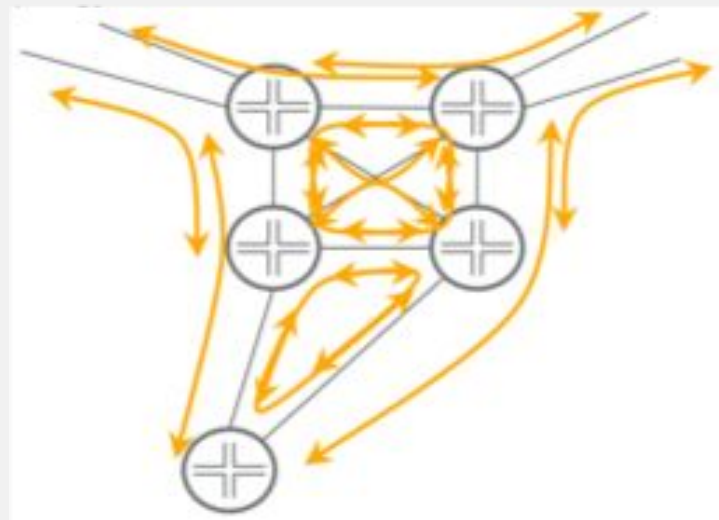
Facebook Approach



- Encountered cases known as gray failures, where either the problem is not detectable by traditional metrics, or the device cannot properly report its own malfunctioning
- Lead them to create **NetNORAD** - a system to treat the network like a “black box” and troubleshoot network problems independently of device polling
- Using UDP instead of TCP
- Breaks network down into clusters by DC, Region and Global
- Outlier Detection Logic and Proximity Tagging

Google Approach

- With traditional network monitoring – “too many times the device either 'lie' or does not tell you the whole picture.”
- Running an “IP SLA” process on each router
- Source route the test packets; allows them to target what gets monitored and ensure full Layer 3 coverage.
- Instead of testing simply interfaces and nodes, they test the ability for a node to forward a packet from every ingress to every egress interfaces.



Microsoft Approach

- Developed a system called, **Pingmesh**, a system for large scale data center network latency measurement and analysis
- It has been running in Microsoft data centers for more than four years, and it collects tens of terabytes of latency data per day
- **Pingmesh** leverages all the servers to launch TCP or HTTP pings to provide the maximum latency measurement coverage.
- Multiple levels of graphs are created consisting of:
 - ✓ servers within a rack
 - ✓ top-of-rack (ToR) switches as virtual nodes
 - ✓ across data centers, treating each data center as a virtual node

Netflix Approach



- Developed **Winston**: an event driven diagnostic and remediation platform
- Monitors for microservice level alerts, feeds diagnostic data into a state machine, takes action based upon the gathered data (DIAGNOSTICS! DIAGNOSTICS! DIAGNOSTICS!)
- Winston is designed to minimize human driven remediation.
- Build vs. Buy: A hybrid solution

TRY NOT TO BOIL THE OCEAN WITH ANSIBLE

ANSIBLE

- “The Great and Powerful”
 - When you should be using Ansible
 - How you should be using Ansible
- “Is Ansible the best solution for this problem?”
 - Let’s throw Ansible Tower into the mix
 - Scope creep (even in this talk!)
- The steps Ansible can handle
 - White Box metrics
 - Grey Box metrics
 - Collation of these metrics
 - Invocation of Automated remediation based on gathered metrics
 - Sending out alerts to human actors for final decision on remediation
 - Allowing for limited human decision making over automated remediation

RED HAT CONSULTING

Who We Are

- Global services organization within Red Hat
- Have the best skills on Red Hat technologies in the industry
- Bring together Red Hat technologies, best practices, and expertise to deliver Red Hat solutions
- We can do everything from initial strategy to hands-on keyboard



**RED HAT
CONSULTING**

RED HAT CONSULTING

What We Do

- Bring **subject matter expertise** around open source and Red Hat products and technologies into your organization
- Enable industry standards and **best practices** within your organization to improve process efficiencies and productivity
- **Mentor your staff** on Red Hat technologies and relevant concepts
- **Mitigate product and open source adoption risks** during enterprise migrations, modernization, and cutting-edge initiatives
- Get **direct access** to Red Hat business units and roadmaps

RED HAT
SUMMIT

THANK YOU



plus.google.com/+RedHat



facebook.com/redhatinc



linkedin.com/company/red-hat



twitter.com/RedHatNews



youtube.com/user/RedHatVideos

RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

Make sure not to miss
out on getting your
FREE BOOK!

Book Signing at the
DevZone @ 12PM

Titled: OpenStack
Administration with
Ansible 2

PLAYBOOK EXAMPLE

```
---
- name: run multiple commands and evaluate the output
  hosts: ios01
tasks:
  - name: show version and show interfaces
    ios_command:
      commands:
        - show version
        - show interfaces
    wait_for:
      - result[0] contains IOS
      - result[1] contains Loopback0
```

PLAYBOOK EXAMPLE – REAL WORLD

```
---
- name: configure ios interface
  hosts: ios01
tasks:
  - name: collect device running-config
    ios_command:
      commands: show running-config interface GigabitEthernet0/2
      provider: "{{ cli }}"
    register: config

  - name: administratively enable interface
    ios_config:
      lines: no shutdown
      parents: interface GigabitEthernet0/2
      provider: "{{ cli }}"
    when: ` "shutdown" in config.stdout[0] `
```

PLAYBOOK EXAMPLE – REAL WORLD (CONT.)

```
---
- name: verify operational status
  ios_command:
    commands:
      - show interfaces GigabitEthernet0/2
      - show cdp neighbors GigabitEthernet0/2 detail
    waitfor:
      - result[0] contains 'line protocol is up'
      - result[1] contains 'iosxr03'
      - result[1] contains '10.0.0.42'
    provider: "{{ cli }}"
```

PLAYBOOK EXAMPLE – REAL WORLD (RESULTS)

```
(ansible)[network]$ ansible-playbook ios_interface.yaml

PLAY [configure ios interface] *****

TASK [collect device running-config] *****
ok: [ios01]

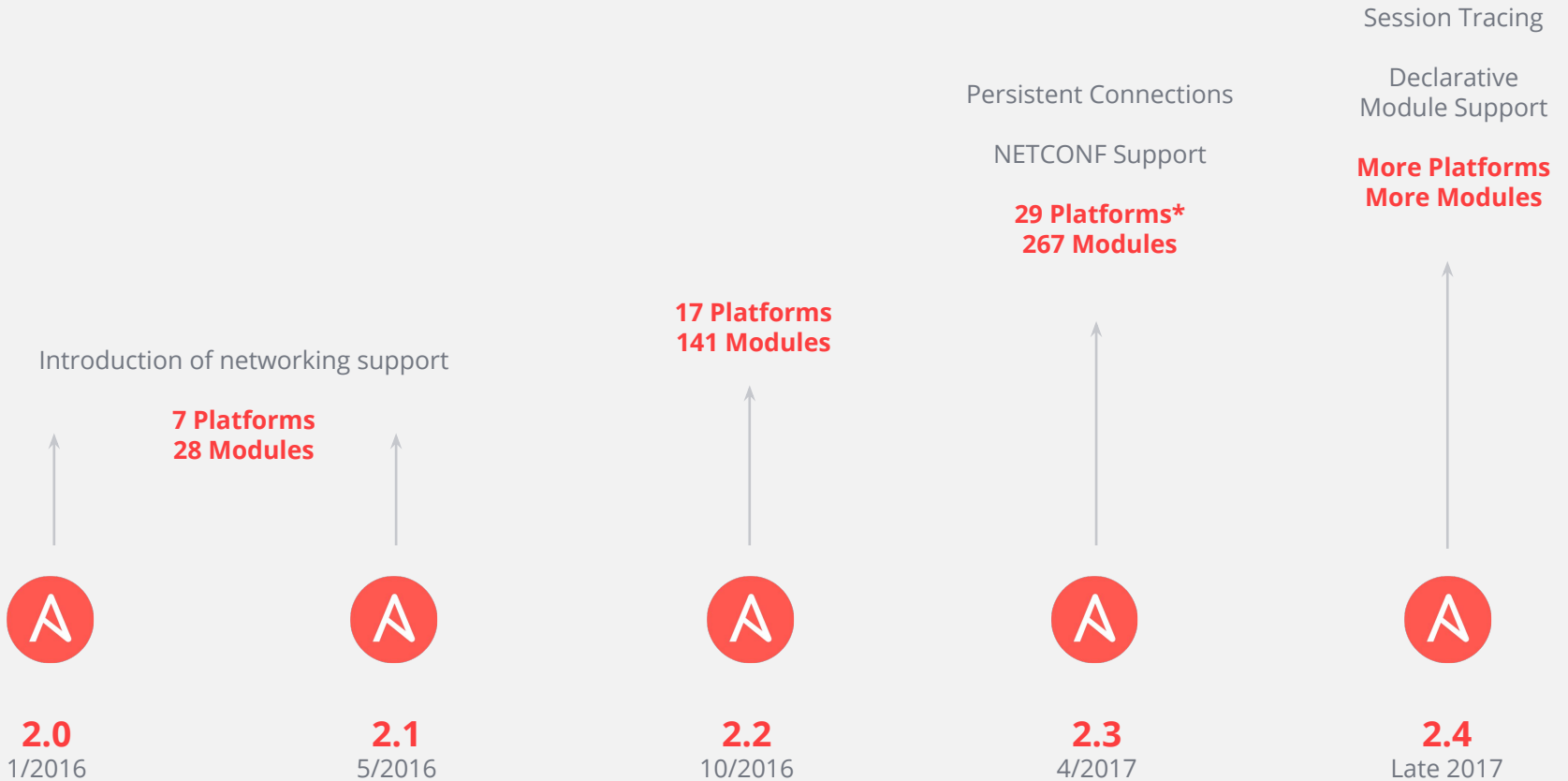
TASK [administratively enable interface] *****
changed: [ios01]

TASK [verify operational status] *****
ok: [ios01]

PLAY RECAP *****
ios01           : ok=3    changed=1    unreachable=0    failed=0
```


NETWORK MODULES: DEVICE SUPPORT

- A10
- Apstra
- Arista EOS (cli, eAPI)
- AVI Networks
- Big Switch Networks
- Cisco ASA, IOS, IOS-XR, NX-OS
- Citrix Netscaler
- Cumulus Linux
- Dell OS6, OS9, OS10
- Exoscale
- F5 BIG-IP
- Fortinet
- Huawei
- Illumos
- Juniper Junos
- Lenovo
- Ordnance
- NETCONF
- Netvisor
- Openswitch
- Open vSwitch (OVS)
- Palo Alto PAN-OS
- Nokia SR OS
- VyOS



*as of 3/2017



NAME DESCRIPTION ACTIVITY LABELS ACTIONS

Bug Check

Continuous Compliance

Daily Switch Report

Deploy Arista Pod

Ops Playbook

Run Ad-hoc Commands

Sanitize Demo

Troubleshoot Virtual M...

Upgrade EOS

LAUNCH JOB | RUN AD-HOC COMMANDS

SURVEY

*WHICH DEVICES TO RUN AGAINST?

Enter list of hosts, colon-delimited. You can use regex as well.

all

*PROVIDE A LIST OF COMMANDS

Enter the full command

```
show version | grep 'image version'  
show ip route | grep '0.0.0.0/0'  
show interfaces counters errors | nz
```

*OUTPUT FORMAT

Return the data in json or text

text

INVENTORY
HQ Data CenterCREDENTIAL
Demo Credential

CANCEL

LAUNCH



RESULTS ▾



STATUS	● Successful	TEMPLATE	Run Ad-hoc Commands
STARTED	1/16/2017 12:37:03 PM	JOB TYPE	Run
FINISHED	1/16/2017 12:37:17 PM	LAUNCHED BY	admin
ELAPSED	00:00:13	INVENTORY	HQ Data Center
PROJECT	DC Santa Clara	PLAYBOOK	run_commands.yml
MACHINE CREDENTIAL	Demo Credential	NETWORK CREDENTIAL	Arista Devices
VERBOSITY	Default		
EXTRA VARIABLES			

ernote

```
1 encoding: text
2 run_commands: "show version | grep 'image version'\nshow ip
3 run_hosts: all
4
```

STANDARD OUT



```

" S      0.0.0.0/0 [1/0] via 172.16.130.2, Management1",
"*****",
"* Command: show interfaces counters errors | nz",
"*****",
"Port          FCS  Align  Symbol  Rx  Runts
Giants        Tx"
]
}
ok: [dc1-tora] => {
  "output": [
    "*****",
    "* Command: show version | grep 'image version'",
    "*****",
    "Software image version: 4.16.7M",
    "*****",
    "* Command: show ip route | grep '0.0.0.0/0'",
    "*****",
    " S      0.0.0.0/0 [1/0] via 172.16.130.2, Management1",
    "*****",
    "* Command: show interfaces counters errors | nz",
    "*****",
    "Port          FCS  Align  Symbol  Rx  Runts
Giants        Tx"
]
}
ok: [dc1-spine2] => {
  "output": [

```