Network Automation:
Do I Need Expensive Vendor Tools To Do Meaningful Automation?

P. Moore
NANOG 72
February 20, 2018
Agenda

• Orchestration Domains
• An Onramp To Automation
• Open Source Tools
• Use Cases Considered
• Case Studies
• Conclusions
4. Services
   – Model-based Service Management
3. Policy
   – Model-based Policy Management
2. Maintenance
   – Leverage Device Management to automate MOPs
1. Device (Foundational)
   – Configuration Management
## Domains Build Upon Each Other

<table>
<thead>
<tr>
<th>Data Model</th>
<th>Automation Level</th>
<th>Operations Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOSCA</td>
<td></td>
<td>Device Images</td>
</tr>
<tr>
<td>YANG</td>
<td></td>
<td>Service Lifecycle</td>
</tr>
<tr>
<td>YAML</td>
<td></td>
<td>MANO</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stage 0</th>
<th>Manual / Script</th>
<th>CLI/Scripts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 1</td>
<td>Day 0, Day 1...</td>
<td>Templates</td>
</tr>
<tr>
<td>Stage 2</td>
<td>OpenConfig</td>
<td>Models</td>
</tr>
<tr>
<td>Stage 3</td>
<td>Service &amp; Policy Lifecycle</td>
<td>Orchestration</td>
</tr>
<tr>
<td>Stage 4</td>
<td>OpenStack VMWare Clouds</td>
<td>VNF – VMs</td>
</tr>
<tr>
<td>Stage 5</td>
<td>Kubernetes Docker</td>
<td>VNF – Cloud Native</td>
</tr>
</tbody>
</table>
Automation Onramp

• Address this on 3 fronts:
  – **People**: who will own network automation?
  – **Process**: define how you will manage the automation work
  – **Platform**: define the tools you will use

• Define your use cases thoroughly – “It is ALL about the use case!”

• Crawl > Walk > Run – start simple and expand

• “Evolve and Accelerate!”
Evolve then Accelerate
People

Who will own network automation?

• Dedicated Group? Not necessary unless you are looking to put a very formal program in place

• Roles Required:
  – **Network Automation Lead** – owns the automation efforts and works to remove roadblocks with other departments, vendors, etc.
  – **Automation Designer** – defines the work to be done, tools to use, workflow/steps of automation, and acts as technical lead
  – **Engineer** – works with the Designer to build the automation
  – **Subject Matter Experts (SME)** – provides knowledge in specific technology areas

• All roles may be filled by a single person in some cases, or may be 4 or more people in larger operations
Process: The Automation Factory

Define how you will manage the automation work
- Submission of automation requests
- Prioritization of which efforts to undertake
- Execution of automation efforts
Platform

Define the tools you will use

• Use tools you already have
• Leverage open source tools
  – Ansible & AWX
  – OpenDaylight, ONAP, etc.
  – Puppet, Chef, Salt, etc.
• Leverage vendors where the value makes sense
Examples: Tools

- **Ansible, Salt, Chef, Puppet, OpenDaylight**
  - Playbook scripting
  - YAML, YANG, NETCONF

- **AWX, Tower, ONAP**
  - Playbook Management
  - Workflow

- **Bitbucket, Github, etc.**
  - Playbook versioning
  - Config versioning (including diff)
Use Cases for Examples

- **Config Management**
  - Backup
  - Config Diff

- **MOP Automation**
  - Sub interface turn up
  - OS Upgrade
Examples: Tool Architecture

AWX provides GUI based:
- Simple Workflow
- Playbook Management
- Job Management
- Simple Inventory

Bitbucket manages:
- Playbooks
- Configs

Ansible is the execution engine underneath AWX to communicate with devices
Example: Config Backup

Leveraging Bitbucket:
- Repository for configs
- History of changes to configs
- Ability to view previous configs
- Ability to see diffs between current version and previous versions
Example: Config Diff

Diff examples showing items removed from config, as well as inserted or changed.
Sub-interface Turn Up: Playbook

Your Playbooks should be:
• Variablized for **reuse** purposes
• Specific to a use case
• Broken into smaller executable “chunks” – even if you could combine more functions into the single playbook – for **reuse** purposes
Sub-interface Turn Up: Job Template

AWX allows for:
- Definition of Templates for jobs
- Management of credentials for network access
- Management of simple inventory of devices
Device OS Upgrade: Playbook

Example of a Playbook for OS Upgrade:

- This Playbook leverages the NTC-Ansible module that can be found at:
  - https://github.com/networktocode/ntc-ansible
- The example Playbook, and more detail, can be found at:
Extensible Architecture

Network Automation Platform

- APIs
- Orchestrators (NFV, Data Center, Cloud, etc.)
- AWX
- Ansible
- Bitbucket (Playbooks, Configs)
- Controllers (ODL, Contrail, etc.)
- Devices
- OSS & Other
More advanced automation platforms allow for:

- Sophisticated workflows
- Cross tool workflows (e.g. Ansible, Puppet, Chef, Cisco NSO, etc.)
- Custom forms and apps to enable more complex automations
Questions?
References

- Network to Code Slack Channel: [https://networktocode.herokuapp.com/](https://networktocode.herokuapp.com/)
- Network To Code Ansible Module: [https://github.com/networktocode/ntc-ansible](https://github.com/networktocode/ntc-ansible)