Network Automation Foundations

Presented By CBTS & Arista Networks

8-6-2019

Craig Bruenderman
CCIE #46341

Rhetorical Questions

- Why are you here?
- What is your problem statement?
- Do you know how your business makes money?
  - What is your Unit of Value?
  - Knowing that will help you connect to your business
- What are your use cases?

Network Automation

- What?
  - “Network automation is the process of automating the configuration, management, testing, deployment, and operations of physical and virtual devices within a network. Every day network tasks and functions are performed automatically.” Juniper Networks
- Why?
  - Speed? Maybe.
  - Multiplex yourself
  - Predictability/Determinism
  - Drive heterogeneous environments with same toolkit
  - Reduce the mundane and get to workflows that move your career, and your business forward

Basic Toolkit

- Python Programming Language
- Fundamental Data Structures
- IDE
  - Combines a text editor with add-ons such as code completion, debugging, build tools, version control integration, integrated terminal, etc.
- API Concepts
- SDK Concepts

Python

- General-purpose object-oriented programming language developed since the 1980's by Guido van Rossum
- Open source
- Interpreted language
  - Internally, your source code is compiled into byte code
  - Byte code then runs in a platform virtual machine, similar to Java
- Approachable, easy(ish) to read
- Huge ecosystem of extension libraries
  - Text processing, Bioinformatics, Network Automation, Machine Learning…

Python Versions

- Python 2
  - Lots of code written for Python 2.7
  - 2.7 will be deprecated on Jan 1 2020
  - https://pythonclock.org/
- **Python 3**
  - Start here
  - Certainly differences, but minimal for our purposes when starting out
    - `print “Hello World” becomes print(“Hello World”)`

Python Run Modes

- Interactive
  - Run python from CLI
  - Do prototyping here
- Scripted
  - Capture your code in .py files

Python Modules

- These are Python libraries you can include to avoid reinventing the wheel
- PIP - Package manager to install/remove/upgrade Modules
- Netmiko
  - Module / library for interacting with network equipment

Lab 1 - Python Intro

aristagui@ip-10-33-5-81: ~

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~$ which python
/usr/bin/python
aristagui@ip-10-33-5-81:~$ python --version
Python 2.7.12
aristagui@ip-10-33-5-81:~$ which python3
/usr/bin/python3
aristagui@ip-10-33-5-81:~$ python3 --version
Python 3.5.2
aristagui@ip-10-33-5-81:~$ python3
Python 3.5.2 (default, Nov 12 2018, 13:43:14)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> exit
Use exit() or Ctrl-D (i.e. EOF) to exit
>>>
aristagui@ip-10-33-5-81:~$
```

Data Structures

- **Structured** - Pre-defined data types with predictable characteristics; easy to parse; usually text-based
  - Dates, IP addresses, E.164 telephone numbering plan
- **Unstructured** - No predefined data model
    - Text files, media files, screen scrapes
- **Semi-Structured** - Somewhere in between; uses internal tags and/or hierarchy to establish parent/child relationships
  - XML, JSON, YAML

Data Serialization Formats

- Data formats designed to be exchanged/consumed by software
- Largely storing and interacting on
  - Lists (Arrays)
  - Dictionaries (Associative Arrays)
- Many to choose from
  - XML
  - JSON
  - YAML

Lists

- Ordered sequence of values
- Entries are indexed (referenced) by their integer index
- Examples
  - In Python
  - In JSON
  - In YAML

Dictionaries

- Unordered collection of key - value pairs
- Entries are indexed by their key, which can be multiple types, but is often a string
- Examples
  - In Python
  - In JSON
  - In YAML

Lab 2.1 - Data Structures

- http://bit.ly/cbautomation

aristagui@ip-10-33-5-81: ~

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~$ git clone https://github.com/craigbruenderman/automation-class.git
Cloning into 'automation-class'...
remote: Enumerating objects: 35, done.
remote: Counting objects: 100% (35/35), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 35 (delta 4), reused 34 (delta 3), pack-reused 0
Unpacking objects: 100% (35/35), done.
Checking connectivity... done.
aristagui@ip-10-33-5-81:~$ tree automation-class/
automation-class/
├── code
│   ├── ansible
│   │   └── install-jq.yml
│   ├── lab2.1.py
│   ├── lab2.2.py
│   └── ports.yml
└── README.md

2 directories, 5 files
aristagui@ip-10-33-5-81:~$
```

Lab 2.2 - Data Structures

aristagui@ip-10-33-5-81: ~/automation-class/code

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~$ cd ~/automation-class/code/
aristagui@ip-10-33-5-81:~/automation-class/code$ python3 -i lab2.1.py
>>> type(dns_servers)
<class 'list'>
>>> dns_servers
['8.8.8.8', '8.8.4.4']
>>>
>>> type(core1)
<class 'dict'>
>>> core1
{'dns': ['8.8.8.8', '8.8.4.4'], 'vlans': [{'id': 10, 'name': 'Data'}, {'id': 50, 'name': 'Voice'}]
, 'hostname': 'core1.cbts.com', 'users': {'craigb': 0, 'admin': 15}, 'vendor': 'arista'}
>>>
>>> as_python()
------ As Python Native List ------
['8.8.8.8', '8.8.4.4']


------ As Python Native Dictionary ------
{'dns': ['8.8.8.8', '8.8.4.4'],
 'hostname': 'core1.cbts.com',
 'users': {'admin': 15, 'craigb': 0},
 'vendor': 'arista',
 'vlans': [{'id': 10, 'name': 'Data'}, {'id': 50, 'name': 'Voice'}]}


>>>
```

Lab 2.3 - Data Structures

ports.yml — ~/automation-class/code — Atom

File Edit View Selection Find Packages Help

Project

- code
  - ansible
  - lab2.1.py
  - lab2.2.py
  - ports.yml

ports.yml

```yaml
- interface: FastEthernet0/28
  description: AppleTV
  mode: access
  vlan: 10

- interface: FastEthernet0/29
  description: IP Camera
  mode: access
  vlan: 11

- interface: FastEthernet0/30
  description: ESXi Host
  mode: trunk
  vlan: 10
```

ports.yml 1:1 LF UTF-8 YAML master 0 files 1 update

Lab 2.3 - Convert Between Data Structures

aristagui@ip-10-33-5-81: ~/automation-class/code

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~/automation-class/code$ ls -l
total 16
drwxrwxr-x 2 aristagui aristagui 4096 Jul 23 17:31 ansible
-rw-rw-r-- 1 aristagui aristagui  817 Jul 23 17:31 lab2.1.py
-rw-rw-r-- 1 aristagui aristagui 1179 Jul 23 17:42 lab2.2.py
-rw-rw-r-- 1 aristagui aristagui  242 Jul 23 17:31 ports.yml
aristagui@ip-10-33-5-81:~/automation-class/code$
aristagui@ip-10-33-5-81:~/automation-class/code$ python3 lab2.2.py
<class 'list'>

[{'description': 'AppleTV',
  'interface': 'FastEthernet0/28',
  'mode': 'access',
  'vlan': 10},
 {'description': 'IP Camera',
  'interface': 'FastEthernet0/29',
  'mode': 'access',
  'vlan': 11},
 {'description': 'ESXi Host',
  'interface': 'FastEthernet0/30',
  'mode': 'trunk',
  'vlan': 10}]
aristagui@ip-10-33-5-81:~/automation-class/code$
aristagui@ip-10-33-5-81:~/automation-class/code$ ls -l
total 20
drwxrwxr-x 2 aristagui aristagui 4096 Jul 23 17:31 ansible
-rw-rw-r-- 1 aristagui aristagui  817 Jul 23 17:31 lab2.1.py
-rw-rw-r-- 1 aristagui aristagui 1179 Jul 23 17:42 lab2.2.py
-rw-rw-r-- 1 aristagui aristagui  276 Jul 23 17:44 ports.json
-rw-rw-r-- 1 aristagui aristagui  242 Jul 23 17:31 ports.yml
aristagui@ip-10-33-5-81:~/automation-class/code$
```

IDE Overview

- Integrated Development Environment
- Distinction between IDE and Code Editor not always clear
- Integrate several tools
  - Source code editor with syntax highlighting, auto-completion
  - Shortcuts
  - Plugin System
  - Version Control System Integration
  - Build, execution, debugging tools

IDE Options

- Visual Studio Code
  - Multi-platform, Open Source, developed by Microsoft
- Atom
  - Multi-platform, Open Source, developed by Google
- PyCharm
  - Multi-platform, Freemium, developed by JetBrains
- Sublime Text
  - Multi-platform, Commercial

API Overview

- Application Programming Interface
- In basic terms, APIs just allow applications to communicate with one another.
- When people speak of “an API”, they sometimes generalize and actually mean “a publicly available web-based API that returns data, likely in JSON or XML”
- The API is not the database or even the server, it is the code that governs the access point(s) for the server.

Web Services API

REST Overview

- An architectural style defined by a set of principles developed by Roy Fielding in his PhD work
  - Not a protocol
- Think of it as a set of design patterns for Web Service APIs that, when followed, make its use predictable and intuitive

Database	HTTP
Create	POST
Read	GET
Update	PUT
Delete	DELETE

Web Service API - Arista eAPI

- Arista eAPI is a method for simply interacting with Arista gear without screen scraping
- HTTP or HTTPS and uses JSON (JavaScript Object Notation)
- JSON Request and Response
- eAPI allows CLI Commands to be issued remotely
- eAPI returns output as JSON, generally in key-value pairs

Navigating eAPI

- Browser-based interaction
  - API docs
- Python interaction
- (pyeapi or other) SDK interaction
- Ansible interaction

Lab 3 - eAPI Command Explorer

SDK Overview

- Generally, a library add-on written for a given language

Lab 4.1 - Add a user with pyeapi

Expanded Toolkit

- Git
- Github
- Ansible

Git Overview

- **Version Control System -** System in which we store and maintain versioned repository of source code
  - Enables collaboration between multiple contributors
- Git is one such free source/version control system that is distributed by design
- Credit (blame) log
- Excellent free book https://git-scm.com/book/en/v2

Git Workflow

Lab 5.1 - Git Preferences Setup

aristagui@ip-10-33-5-81: ~

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~$ git --version
git version 2.7.4
aristagui@ip-10-33-5-81:~$ git config --global user.name "Craig Bruenderman"
aristagui@ip-10-33-5-81:~$ git config --global user.email "craig@bruenderman.org"
aristagui@ip-10-33-5-81:~$
aristagui@ip-10-33-5-81:~$ git config --list
user.email=craig@bruenderman.org
user.name=Craig Bruenderman
aristagui@ip-10-33-5-81:~$
```

Lab 5.2 - Git Create Your Own Repository

aristagui@ip-10-33-5-81: ~/muh-bad-codez

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~$ mkdir muh-bad-codez
aristagui@ip-10-33-5-81:~$ cd muh-bad-codez/
aristagui@ip-10-33-5-81:~/muh-bad-codez$ git init
Initialized empty Git repository in /home/aristagui/muh-bad-codez/.git/
aristagui@ip-10-33-5-81:~/muh-bad-codez$ echo "Info here" >> README.md
aristagui@ip-10-33-5-81:~/muh-bad-codez$ git add README.md
aristagui@ip-10-33-5-81:~/muh-bad-codez$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md

aristagui@ip-10-33-5-81:~/muh-bad-codez$ git commit -m "my first commit"
[master (root-commit) 6c2aa7d] my first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
aristagui@ip-10-33-5-81:~/muh-bad-codez$ git status
On branch master
nothing to commit, working directory clean
aristagui@ip-10-33-5-81:~/muh-bad-codez$
```

Git Branches

- Branching means you diverge from the main line of development and continue to do work without messing with that main line
- These are useful for playing with features, testing, multi-user
- You probably don’t want to commit incomplete or broken code, but you still want to keep it safe in the repo
- Git keeps a special pointer called HEAD that points to the current branch you’re on

Lab 5.3 - Branch and Merge

Github

- Popular hosting service for storing Git source repositories
- Allows public and private repos
- 100MM+ repos and growing
- Uses “freemium” model
- Bought by Microsoft June 2018

Lab 6 - View Code Diffs in Github

craigbruenderman committed 1 hour ago 1 parent 672424d commit f999e6fae0b729e77e7953039608b2cb6887e699

Showing **3 changed files** with **9 additions** and **2 deletions**. Unified | Split

BIN -33.1 KB (88%) Lab_Guide.pdf

Binary file not shown.

11 code/ansible/hosts

```
@@ -1,2 +1,9 @@
-[veos]
-192.168.0.14
+[webservers]
+web1 ansible_host=10.0.0.1
+web2 ansible_host=10.0.0.2
+
+[database_servers]
+db1 ansible_host=172.16.99.1
+
+#[veos]
+#switch1 ansible_host=192.168.0.14
```

0 code/ansible/install-jq.yml → code/ansible/lab8.1.yml

File renamed without changes.

0 comments on commit f999e6f

Ansible Overview

- **Ansible** is an open source project for software provisioning, configuration management, and application deployment
  - Infrastructure as Code Tool sponsored by Red Hat (now owned by IBM, fingers crossed)
- **AWX** - community project that enables users to better control their Ansible project use in IT environments
  - AWX is the upstream project from which the commercial Red Hat Ansible Tower offering is ultimately derived
- **Ansible Tower** - is produced by taking selected releases of AWX, hardening them for long-term supportability, and making them available to customers as the Ansible Tower offering
- **Galaxy**

Ansible

- Simple, yet extremely powerful automation
- Agent-less and self-contained
- Uses SSH or Web Service as transport
- Capable of automating network devices, servers, cloud providers, VMware, many OEMs and SaaS services
- Python based, so extendable
- YAML driven, for easier digestion
- **Idempotent** — ability to perform same action repeatedly while producing the same result, without harm; making multiple identical requests has the same effect as making a single request

Ansible Terminology

- **Tasks** – smallest unit of work in Ansible; combines a module action with a task name
- **Play** – mapping of task instructions to the set of hosts they will run against
- **Playbook** — collection of at least one play by which Ansible orchestrates, configures, administers, or deploys systems
- **Inventory** — file that defines hosts and groups in Ansible
- **Modules** — task plugins that control particular resources

Ansible Modules

- 1200+ built-in modules including:
  - apt, yum, copy, command, cron, dns, docker, easy_install, ec2 (amazon modules), file, filesystem, find, git, known_hosts, mysql, mongodb, nagios, npm, openstack, rax (rackspace), pip, shell, snmp_facts…
- Typical modules for me
  - vmware, eos, ios, nxos, ucsm
- Modules differ in completeness

Lab 7.1 - Ansible Ad Hoc Command

craigb@Craigs-MacBook-Pro: ~/GoogleDrive/CBTS/Automation Foundations/repo/code/ansible (zsh)

..ndations/repo (zsh) ⌘1 | ../code/ansible (zsh) ⌘2 | ..e/code/python (zsh) ⌘3

```
➜ ansible git:(master) ✗ ansible localhost -m raw -a "ping -c 3 127.0.0.1"
 [WARNING]: No inventory was parsed, only implicit localhost is available

localhost | CHANGED | rc=0 >>
PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=64 time=0.050 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.077 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.050/0.071/0.086/0.015 ms


➜ ansible git:(master) ✗
```

Lab 7.2 - Ad Hoc Package Installation

aristagui@ip-10-33-5-81: ~

File Edit Tabs Help

```
Reading database ... 179774 files and directories currently installed.)\r\nPreparing to unpack .../
yamllint_1.2.1-1_amd64.deb ...\r\nUnpacking yamllint (1.2.1-1) ...\r\nProcessing triggers for man-d
b (2.7.5-1) ...\r\nSetting up yamllint (1.2.1-1) ...\r\n",
    "stdout_lines": [
        "Reading package lists...",
        "Building dependency tree...",
        "Reading state information...",
        "The following NEW packages will be installed:",
        "  yamllint",
        "0 upgraded, 1 newly installed, 0 to remove and 204 not upgraded.",
        "Need to get 23.4 kB of archives.",
        "After this operation, 124 kB of additional disk space will be used.",
        "Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial/universe amd64 yamllint amd64
1.2.1-1 [23.4 kB]",
        "Fetched 23.4 kB in 0s (389 kB/s)",
        "Selecting previously unselected package yamllint.",
        "(Reading database ... ",
        "(Reading database ... 5%",
        "(Reading database ... 10%",
        "(Reading database ... 15%",
        "(Reading database ... 20%",
        "(Reading database ... 25%",
        "(Reading database ... 30%",
        "(Reading database ... 35%",
        "(Reading database ... 40%",
        "(Reading database ... 45%",
        "(Reading database ... 50%",
        "(Reading database ... 55%",
        "(Reading database ... 60%",
        "(Reading database ... 65%",
        "(Reading database ... 70%",
        "(Reading database ... 75%",
        "(Reading database ... 80%",
        "(Reading database ... 85%",
        "(Reading database ... 90%",
        "(Reading database ... 95%",
        "(Reading database ... 100%",
        "(Reading database ... 179774 files and directories currently installed.)",
        "Preparing to unpack .../yamllint_1.2.1-1_amd64.deb ...",
        "Unpacking yamllint (1.2.1-1) ...",
        "Processing triggers for man-db (2.7.5-1) ...",
        "Setting up yamllint (1.2.1-1) ..."
    ]
}
aristagui@ip-10-33-5-81:~$
```

Lab 7.3 - Ad Hoc Using An Inventory

aristagui@ip-10-33-5-81: ~/automation-class/code/ansible

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~/automation-class/code/ansible$ ansible veos -i hosts -m raw -a "show vers
ion" -u arista -k
SSH password:
switch1 | CHANGED | rc=0 >>
Arista vEOS
Hardware version:
Serial number:       leaf1
System MAC address:  2cc2.60fd.3d06

Software image version: 4.21.2F
Architecture:           i386
Internal build version: 4.21.2F-10430819.4212F
Internal build ID:      7b26fbef-3d08-4910-bb95-df08faaa5b13

Uptime:                 0 weeks, 0 days, 0 hours and 8 minutes
Total memory:           3977260 kB
Free memory:            3202276 kB

Warning: Permanently added '192.168.0.14' (ECDSA) to the list of known hosts.
Shared connection to 192.168.0.14 closed.


aristagui@ip-10-33-5-81:~/automation-class/code/ansible$
```

Ansible Inventory

- Contains a list of hosts which you want Ansible to interact with
- Can be as basic as containing a list of hostnames
- Can be more complex, containing groups, groups of groups, host variables, etc.
- General advice to keep inventory files minimal and put variables elsewhere

Ansible Variables

- Often we want to set the same parameter to a different value on a per-device or group basis
- DRY principle
- Multiple ways to define vars, depending on what we need
  - Directly within playbooks
  - Within inventory
  - In separate host variables and group variables files
- There is an order of precedence when vars are defined in more than one place, but try to avoid this

Ansible Group Variables

- Stored in group_vars directory
- Apply to all of the hosts which reside inside a group
- To configure group variables simply create a YAML file which matches the name of the group
- Also an **all** group variables file which specifies variables for all groups.

Ansible Host Variables

- Stored in host_vars directory
- Host variables work the same way as Group variables
- Instead of the filename matching group name, it needs to match the host name

Ansible Variables Example Layout

craigb@Craigs-Mac-mini: ~/Projects/automation-class/code (zsh) — 122×27

..on-class/code (zsh) ⌘1 | ~ (zsh) ⌘2

```
➜  code git:(master) ✗ tree
.
├── group_vars
│   ├── all.yml
│   └── webservers.yml
├── host_vars
│   └── db1.yml
├── hosts
├── lab2.2.py
├── lab2.3.py
├── lab4.1.py
├── lab7.1.yml
├── lab7.2.yml
├── lab8.1.yml
├── ports.json
├── ports.yml
└── reports
    ├── facts
    │   ├── 192.168.10.2.md
    │   └── 192.168.10.3.md
    ├── facts.j2
    ├── master.md
    └── master.pdf

4 directories, 17 files
➜  code git:(master) ✗
```

Anatomy of an Ansible Playbook

```yaml
---
# Name of the playbook
- name: Our first Ansible playbook
  # List of hosts or host group to run against
  hosts: localhost
  # Toggle to run with root privileges
  become: true

  # Variables we define to use within this playbook
  vars:
    # Password to use when running command as root
    ansible_become_password: "@rista123"
    # Name of the package we'll install
    package_name: "jq"

  # List of tasks to be executed
  tasks:
    # Name of one task
    - name: Install jq
      # Action of this task
      # In this case, the apt package manager module
      apt:
        # Actual name of the package to be installed
        # Referencing the defined variable 'package_name'
        name: "{{ package_name }}"

```

Ansible Challenges

- Very whitespace sensitive
- Documentation sometimes needs its own documentation
- Multiple connection methods aren’t always clear

Lab 8.1 - Use Ansible Playbook to Install jq

aristagui@ip-10-33-5-81: ~/automation-class/code/ansible

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~/automation-class/code$ jq
bash: /usr/bin/jq: No such file or directory
aristagui@ip-10-33-5-81:~/automation-class/code$ cd ansible/
aristagui@ip-10-33-5-81:~/automation-class/code/ansible$ ansible-playbook install-jq.yml
 [WARNING]: Unable to parse /etc/ansible/hosts as an inventory source

 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit
localhost does not match 'all'


PLAY [Our first Ansible playbook] *********************************************************

TASK [Gathering Facts] ********************************************************************
ok: [localhost]

TASK [Install jq] *************************************************************************
changed: [localhost]

PLAY RECAP ********************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0

aristagui@ip-10-33-5-81:~/automation-class/code/ansible$ jq . ../ports.json
[
  {
    "interface": "FastEthernet0/28",
    "mode": "access",
    "description": "AppleTV",
    "vlan": 10
  },
  {
    "interface": "FastEthernet0/29",
    "mode": "access",
    "description": "IP Camera",
    "vlan": 11
  },
  {
    "interface": "FastEthernet0/30",
    "mode": "trunk",
    "description": "ESXi Host",
    "vlan": 10
  }
]
aristagui@ip-10-33-5-81:~/automation-class/code/ansible$
```

Lab 8.2 - Use Ansible to interact with the Twilio REST API

- **Prize Alert**
- First one to use Ansible to send me an MMS with their email address, switch serial, and an Ansible heart wins a prize

```
craigb@Craigs-MacBook-Pro: ~/GoogleDrive/CBTS/Automation Foundations/repo/code/ansible (zsh)
..ndations/repo (zsh) ⌘1   ../code/ansible (zsh) ⌘2   ..e/code/python (zsh) ⌘3

➜ ansible git:(master) ✗ ansible-playbook lab8.2.yml
 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'


PLAY [Twilio API] *************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************
ok: [localhost]

TASK [Send a text, win a prize!] **********************************************************************************
ok: [localhost -> localhost]

PLAY RECAP ********************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

➜ ansible git:(master) ✗
```

Jinja Templating Engine

- Jinja is a Python-based templating engine that works with Ansible out of the box
- Jinja templates take variables from Ansible and then output text
- It's like mail merge for configuration
- Writing Ansible playbooks that issue sequential commands to configure all 48+ interfaces on a switch is inefficient
- Jinja is also present in other parts of Ansible, such as filters and tests
- Jinja2 is the actual name of the Python library

Lab 8 - Jinja Templating

aristagui@ip-10-33-5-81: ~/automation-class/code

File Edit Tabs Help

```
aristagui@ip-10-33-5-81:~/automation-class/code$ ansible-playbook -i hosts lab8.1.yml
Switch password?:

PLAY [Create Report with Ansible and Jinja] ************************************************************************

TASK [Collect facts for EOS] ***************************************************************************************
ok: [192.168.0.14]
ok: [192.168.0.10]
ok: [192.168.0.11]

TASK [Debug config] ************************************************************************************************
skipping: [192.168.0.10]
skipping: [192.168.0.11]
skipping: [192.168.0.14]

TASK [Debug neighbors] *********************************************************************************************
skipping: [192.168.0.10]
skipping: [192.168.0.11]
skipping: [192.168.0.14]

TASK [Create reports directory if it does not exist] ***************************************************************
changed: [192.168.0.11]
ok: [192.168.0.14]
ok: [192.168.0.10]

TASK [Generate Device Reports] *************************************************************************************
changed: [192.168.0.10]
changed: [192.168.0.14]
changed: [192.168.0.11]

TASK [Create Master Report] ****************************************************************************************
changed: [192.168.0.10]

TASK [Create PDF of Master] ****************************************************************************************
changed: [192.168.0.10 -> localhost]

PLAY RECAP *********************************************************************************************************
192.168.0.10               : ok=5    changed=3    unreachable=0    failed=0
192.168.0.11               : ok=3    changed=2    unreachable=0    failed=0
192.168.0.14               : ok=3    changed=1    unreachable=0    failed=0

aristagui@ip-10-33-5-81:~/automation-class/code$
```

aristagui@ip-10-... [GitHub - craigbr... [reports] 16:01

Use Cases

Use Case: Bulk Config Audit

meraki-dns-check.py

```python
#!/usr/local/bin/python3

from meraki import meraki
from tabulate import tabulate
from itertools import islice
import csv
import os

apikey = os.environ['MERAKI_API_KEY']
myOrgs = meraki.myorgaccess(apikey)
#print(myOrgs)

for i, entry in enumerate(myOrgs):
    if entry['name'] == "CustomerName":
        orgid = entry['id']

outputList = []
myNetworks = meraki.getnetworklist(apikey, orgid, suppressprint=True)
# Limiter for debugging
#limit = 3
limit = len(myNetworks)
print("%s Networks found" % limit)

for network in islice(myNetworks, limit):
    deviceList = meraki.getnetworkdevices(apikey, network["id"], suppressprint=True)
    for device in deviceList:
        deviceUplinks = meraki.getdeviceuplink(apikey, network["id"], device["serial"], suppressprint=True)
        for uplink in deviceUplinks:
            outputList.append({'Network Name': network.get("name", None), "Device Name" : device["name"], "Device Model" : device["model"], "Interface" : uplink.get("interface" ,
            None), "DNS" : uplink.get("dns", None), "Static IP" : uplink.get("usingStaticIp", None)})

print(tabulate(outputList, headers="keys"))

with open("pjs-devices.csv", "w") as csvfile:
    fields = outputList[0].keys()
    writer = csv.DictWriter(csvfile, fieldnames=fields)
    writer.writeheader()
    for row in outputList:
        writer.writerow(row)
    csvfile.close()
```

LF UTF-8 Python master Fetch GitHub Git (1)

Use Case: Bulk Config Audit

craigb@Craigs-Mac-mini: ~/Projects/meraki-tools (zsh)

..ons/repo/code (zsh) ⌘1 | ..ndations/repo (zsh) ⌘2 | ../meraki-tools (zsh) ⌘3

```
➜ meraki-tools git:(master) ✗ python3 meraki-dns-check.py
Organization Operation Successful - See returned data for results

707 Networks found
Network Name         Device Name              Device Model    Interface    DNS                              Static IP
-------------------  -----------------------  --------------  -----------  -------------------------------  -----------
000000-HUB1-LOU-PN   LOU-HQ-SEC-MX450-P01     MX450           WAN 1        208.67.222.222, 208.67.220.220   True
000000-HUB1-LOU-PN   LOU-HQ-SEC-MX450-P01     MX450           WAN 2        208.67.222.222, 208.67.220.220   True
000000-HUB1-LOU-PN   LOU-HQ-PRI-MX450-P01     MX450           WAN 1        208.67.222.222, 208.67.220.220   True
000000-HUB1-LOU-PN   LOU-HQ-PRI-MX450-P01     MX450           WAN 2        208.67.222.222, 208.67.220.220   True
000000-HUB1-NSH-PN   NSH-PT-SEC-MX450-P01     MX450           WAN 1        208.67.222.222, 208.67.220.220   True
000000-HUB1-NSH-PN   NSH-PT-SEC-MX450-P01     MX450           WAN 2
000000-HUB1-NSH-PN   NSH-PT-PRI-MX450-P01     MX450           WAN 1        208.67.222.222, 208.67.220.220   True
000000-HUB1-NSH-PN   NSH-PT-PRI-MX450-P01     MX450           WAN 2        208.67.222.222, 208.67.220.220   True
008670-KY-LOU-PN     008670-KY-LOU-PN         MX65W           WAN 1        209.18.47.63, 209.18.47.62       False
008670-KY-LOU-PN     008670-KY-LOU-PN         MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
008670-KY-LOU-PN     008670-KY-LOU-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
008670-KY-LOU-PN     008670-KY-LOU-SW-P02     MS225-24P       WAN 1
004857-TX-AUS-PN     004857-TX-AUS-MX-P01     MX65W           WAN 1        209.18.47.63, 209.18.47.61       False
004857-TX-AUS-PN     004857-TX-AUS-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
004857-TX-AUS-PN     004857-TX-AUS-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
000000-CBTS-TEST-PN  000000-OH-CIN-MX-P01     MX65W           WAN 1        192.168.250.1                    False
000000-CBTS-TEST-PN  000000-OH-CIN-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
000000-CBTS-TEST-PN  000000-CBTS-TEST-SW-P01  MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
004849-TX-COR-PN     004849-TX-COR-MX-P01     MX65W           WAN 1        71.10.216.1, 71.10.216.2         False
004849-TX-COR-PN     004849-TX-COR-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
004849-TX-COR-PN     004849-TX-COR-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
000017-KY-LOU-PN     000017-KY-LOU-MX-P01     MX65W           WAN 1        209.18.47.61, 209.18.47.63       False
000017-KY-LOU-PN     000017-KY-LOU-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
000017-KY-LOU-PN     000017-KY-LOU-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
000036-KY-LOU-PN     000036-KY-LOU-MX-P01     MX65W           WAN 1        209.18.47.61, 209.18.47.63       False
000036-KY-LOU-PN     000036-KY-LOU-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
000036-KY-LOU-PN     000036-KY-LOU-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
004914-SC-ROC-PN     004914-SC-ROC-MX-P01     MX65W           WAN 1        8.8.8.8, 208.104.2.36            False
004914-SC-ROC-PN     004914-SC-ROC-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
004914-SC-ROC-PN     004914-SC-ROC-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
004876-CA-RIV-PN     004876-CA-RIV-MX-P01     MX65W           WAN 1        71.10.216.1, 71.10.216.2         False
004876-CA-RIV-PN     004876-CA-RIV-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
004876-CA-RIV-PN     004876-CA-RIV-SW-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
000004-KY-LOU-PN     000004-KY-LOU-MX-P01     MX65W           WAN 1        209.18.47.62, 209.18.47.61       False
000004-KY-LOU-PN     000004-KY-LOU-MX-P01     MX65W           WAN 2        208.67.222.222, 208.67.220.220   True
000004-KY-LOU-PN     000004-KY-LOU-MS-P01     MS120-48LP      WAN 1        208.67.222.222, 208.67.220.220   False
```

Use Case: Bulk Security Policy Change

craigb@Craigs-Mac-mini: ~/Projects/meraki-tools (zsh)

..ons/repo/code (zsh) ⌘1 | ..ndations/repo (zsh) ⌘2 | ../meraki-tools (zsh) ⌘3 | python3 (Python) ⌘4

```
➜ meraki-tools git:(master) jq . removeCatGroupPolicy.json
{
  "firewallAndTrafficShaping": {
    "settings": "custom",
    "trafficShapingRules": [],
    "l3FirewallRules": [],
    "l7FirewallRules": [
      {
        "policy": "deny",
        "type": "applicationCategory",
        "value": {
          "id": "meraki:layer7/category/8",
          "name": "Peer-to-peer (P2P)"
        }
      },
      {
        "policy": "deny",
        "type": "applicationCategory",
        "value": {
          "id": "meraki:layer7/category/6",
          "name": "Gaming"
        }
      },
      {
        "policy": "deny",
        "type": "applicationCategory",
        "value": {
          "id": "meraki:layer7/category/9",
          "name": "Social web & photo sharing"
        }
      },
      {
        "policy": "deny",
        "type": "applicationCategory",
        "value": {
          "id": "meraki:layer7/category/12",
          "name": "Software & anti-virus updates"
        }
      },
      {
        "policy": "deny",
        "type": "applicationCategory",
```

Use Case: Bulk Security Policy Change

python3 bn-automation.py (Python)

..e/pop-ansible (zsh) ⌘1 | python3 (Python) ⌘2

```
2019-07-12 18:29:39.612 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2194mxr ID: L_583216151744487706
2019-07-12 18:29:41.123 | DEBUG    | __main__:<module>:109 - Network Name: s2226mxr ID: L_583216151744487707 matching Group Policy ID 101
2019-07-12 18:29:42.715 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2226mxr ID: L_583216151744487707
2019-07-12 18:29:44.035 | DEBUG    | __main__:<module>:109 - Network Name: s2542mxr ID: L_583216151744487708 matching Group Policy ID 101
2019-07-12 18:29:45.645 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2542mxr ID: L_583216151744487708
2019-07-12 18:29:46.916 | DEBUG    | __main__:<module>:109 - Network Name: s2197mxr ID: L_583216151744487709 matching Group Policy ID 101
2019-07-12 18:29:48.399 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2197mxr ID: L_583216151744487709
2019-07-12 18:29:50.107 | DEBUG    | __main__:<module>:109 - Network Name: s2653mxr ID: L_583216151744487710 matching Group Policy ID 101
2019-07-12 18:29:51.709 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2653mxr ID: L_583216151744487710
2019-07-12 18:29:53.104 | DEBUG    | __main__:<module>:109 - Network Name: s2566mxr ID: L_583216151744487790 matching Group Policy ID 101
2019-07-12 18:29:54.778 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2566mxr ID: L_583216151744487790
2019-07-12 18:29:56.564 | DEBUG    | __main__:<module>:109 - Network Name: s2574mxr ID: L_583216151744487791 matching Group Policy ID 101
2019-07-12 18:29:58.165 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2574mxr ID: L_583216151744487791
2019-07-12 18:29:59.607 | DEBUG    | __main__:<module>:109 - Network Name: s2229mxr ID: L_583216151744487792 matching Group Policy ID 101
2019-07-12 18:30:01.083 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2229mxr ID: L_583216151744487792
2019-07-12 18:30:02.441 | DEBUG    | __main__:<module>:109 - Network Name: s9982mxr ID: L_583216151744487854 matching Group Policy ID 101
2019-07-12 18:30:04.343 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s9982mxr ID: L_583216151744487854
2019-07-12 18:30:06.086 | DEBUG    | __main__:<module>:109 - Network Name: s2244mxr ID: L_583216151744487855 matching Group Policy ID 101
2019-07-12 18:30:07.816 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2244mxr ID: L_583216151744487855
2019-07-12 18:30:09.568 | DEBUG    | __main__:<module>:109 - Network Name: s2576mxr ID: L_583216151744487856 matching Group Policy ID 101
2019-07-12 18:30:11.365 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2576mxr ID: L_583216151744487856
2019-07-12 18:30:12.882 | DEBUG    | __main__:<module>:109 - Network Name: s2676mxr ID: L_583216151744487857 matching Group Policy ID 101
2019-07-12 18:30:14.443 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2676mxr ID: L_583216151744487857
2019-07-12 18:30:15.847 | DEBUG    | __main__:<module>:109 - Network Name: s2256mxr ID: L_583216151744487858 matching Group Policy ID 101
2019-07-12 18:30:17.449 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2256mxr ID: L_583216151744487858
2019-07-12 18:30:19.706 | DEBUG    | __main__:<module>:109 - Network Name: s2694mxr ID: L_583216151744487859 matching Group Policy ID 101
2019-07-12 18:30:21.547 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2694mxr ID: L_583216151744487859
2019-07-12 18:30:23.085 | DEBUG    | __main__:<module>:109 - Network Name: s2586mxr ID: L_583216151744487860 matching Group Policy ID 101
2019-07-12 18:30:24.926 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2586mxr ID: L_583216151744487860
2019-07-12 18:30:26.261 | DEBUG    | __main__:<module>:109 - Network Name: s2594mxr ID: L_583216151744487861 matching Group Policy ID 101
2019-07-12 18:30:28.305 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2594mxr ID: L_583216151744487861
2019-07-12 18:30:29.634 | DEBUG    | __main__:<module>:109 - Network Name: s2606mxr ID: L_583216151744487862 matching Group Policy ID 101
2019-07-12 18:30:31.768 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2606mxr ID: L_583216151744487862
2019-07-12 18:30:33.325 | DEBUG    | __main__:<module>:109 - Network Name: s2275mxr ID: L_583216151744487863 matching Group Policy ID 101
2019-07-12 18:30:35.268 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2275mxr ID: L_583216151744487863
2019-07-12 18:30:36.712 | DEBUG    | __main__:<module>:109 - Network Name: s2706mxr ID: L_583216151744487864 matching Group Policy ID 101
2019-07-12 18:30:38.597 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2706mxr ID: L_583216151744487864
2019-07-12 18:30:40.106 | DEBUG    | __main__:<module>:109 - Network Name: s2614mxr ID: L_583216151744487865 matching Group Policy ID 101
2019-07-12 18:30:41.720 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2614mxr ID: L_583216151744487865
2019-07-12 18:30:43.360 | DEBUG    | __main__:<module>:109 - Network Name: s2279mxr ID: L_583216151744487869 matching Group Policy ID 101
2019-07-12 18:30:45.196 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2279mxr ID: L_583216151744487869
2019-07-12 18:30:46.647 | DEBUG    | __main__:<module>:109 - Network Name: s2608mxr ID: L_583216151744487870 matching Group Policy ID 101
2019-07-12 18:30:48.376 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2608mxr ID: L_583216151744487870
2019-07-12 18:30:49.728 | DEBUG    | __main__:<module>:109 - Network Name: s2721mxr ID: L_583216151744487871 matching Group Policy ID 101
2019-07-12 18:30:51.550 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2721mxr ID: L_583216151744487871
2019-07-12 18:30:52.993 | DEBUG    | __main__:<module>:109 - Network Name: s2657mxr ID: L_583216151744487872 matching Group Policy ID 101
2019-07-12 18:30:54.612 | DEBUG    | __main__:updateGroupPolicies:94 - Updated Group Policy 101 for Network Name: s2657mxr ID: L_583216151744487872
```

Use Case: New Datacenter Build

- Full stack, compute/storage/FC switching/routing/switching/firewall/management etc.
- Remote site in Halifax, Nova Scotia
- Netbox DCIMs as a source of truth
  - Abstract the config specifics out of your automation code, and into a more universal DCIM tool
  - Interact with Netbox using its API & SDK

netbox | Organization | Racks | Devices | IPAM | Virtualization | Circuits | Power | Secrets | Search | john.bruenderman

Racks / HLFXNS02W02 / C2-2-2-1 (HFX1-H35)

Search racks

Previous Rack | Next Rack | Edit this rack | Delete this rack

Rack C2-2-2-1 (HFX1-H35)

Created March 7, 2019 · Updated 1 week, 6 days ago

Rack | Changelog

Rack

Site	HEUC Pods > HLFXNS02W02
Group	None
Facility ID	HFX1-H35
Tenant	None
Status	Active
Role	None
Serial Number	—
Asset Tag	—
Devices	16

Dimensions

Type	4-post cabinet
Width	19 inches
Height	42U (ascending)
Outer Width	—
Outer Depth	—

Tags

No tags assigned

Comments

None

Power Feeds

Panel	Feed	Status	Type	Utilization
HLFXNS02W02 Test Power Panel	C2-2-2-1 A Side	Active	Primary	0%
HLFXNS02W02 Test Power Panel	C2-2-2-1 B Side	Active	Redundant	0%

Images

None

Non-Racked Devices

Name	Role	Type	Parent
C2-2-2-1-PDU A	PDU	Generic Generic HEUC PDU	—
C2-2-2-1-PDU B	PDU	Generic Generic HEUC PDU	—

Add a non-racked device

netbox | Organization | Racks | Devices | IPAM | Virtualization | Circuits | Power | Secrets | Search | john.bruenderman

+ Add Components | Edit this device | Delete this device

HEUC-HAL-3850-01

Created March 26, 2019 · Updated 3 weeks, 6 days ago

Device | Inventory 0 | Status | LLDP Neighbors | Configuration | Config Context | Changelog

Device

Site	HEUC Pods > HLFXNS02W02
Rack	C2-2-2-1 (HFX1-H35)
Position	U28 / Front
Tenant	Customers-E > EHCS-HEUC
Device Type	Cisco WS-C3850-48T (1U)
Serial Number	—
Asset Tag	HEUC-HAL-3850-01

Management

Role	Management
Platform	Cisco IOS
Status	Active
Primary IPv4	172.30.80.1
Primary IPv6	—

Custom Fields

CRM Account Number	—

Tags

No tags assigned

Comments

OOB Mangement switch for HEUC environment

Console / Power

Console	HAL-959	HEUC-HAL-TS-01	Port 1

+ Add console port | + Add power port

Secrets

None found

+ Add secret

Services

None

+ Assign service

Images

None

+ Attach an image

Related Devices

None found

Interfaces

Show IPs

Name	LAG	Description	MTU	Mode	Cable	Connection		
Gi0/0		OOB Management Port	—	Access	HAL-1177	Meraki-OOB-HAL	LAN 11	
Gi1/0/1		—	—	—	HAL-960	HEUC-HAL-TS-01	LAN-1	
Gi1/0/2		—	—	—	#1170	HEUC-HAL-ASR-01	Gi0	
Gi1/0/3		—	—	—	#1171	HEUC-HAL-ASR-02	Gi0	
Gi1/0/4		—	—	—	HAL-1070	HEUC-HAL-FI-A	Mgmt0	
Gi1/0/5		—	—	—	HAL-1071	HEUC-HAL-FI-B	Mgmt0	

Driving Ansible with Netbox Data

```yaml
- name: Find ID of switch
  uri:
    url: "http://{{netbox_server}}/api/dcim/devices/?site={{ site_name }}&name={{ inventory_hostname }}"
    method: GET
    headers:
      Authorization: "{{ netbox_auth_header }}"
    body_format: json
    return_content: yes
    follow_redirects: yes
  register: json_response
  tags:
    - site_devices

- name: Find switch interfaces
  uri:
    url: "http://{{netbox_server}}/api/dcim/interfaces/?device_id={{ json_response.json.results[0].id }}"
    method: GET
    headers:
      Authorization: "{{ netbox_auth_header }}"
    body_format: json
    return_content: yes
    follow_redirects: yes
  register: json
  tags:
    - site_devices

- name: Label all used interfaces
  ios_interface:
    name: "{{ item.name }}"
    description: "[<- {{ item.connected_endpoint.device.name }} - {{ item.connected_endpoint.name }} ->]"
  loop: "{{ json.json.results }}"
  when: (
          (item.connected_endpoint)
          and
          (item.connected_endpoint.device.name)
          and
          (item.connected_endpoint.name)
        )
  tags: label-interfaces
```

Driving Ansible with Netbox Data

```
craigb@Craigs-Mac-mini: ~/Projects/pop-ansible (zsh)                                                    ⌥⌘1
..ons/repo/code (zsh) ⌘1   ..n Foundations (zsh) ⌘2   ..s/pop-ansible (zsh) ⌘3

➜ pop-ansible git:(master) ✗ ansible-playbook playbooks/management-switches.yml -t site_devices,label-interfaces

PLAY [3850 Management Switch Configuration] ***********************************************************************

TASK [Find ID of switch] ******************************************************************************************
 [WARNING]: The value True (type bool) in a string field was converted to 'True' (type string). If this does not look like what you
expect, quote the entire value to ensure it does not change.

ok: [HEUC-HAL-3850-01]

TASK [Find switch interfaces] *************************************************************************************
ok: [HEUC-HAL-3850-01]

TASK [Label all used interfaces] ******************************************************************************
changed: [HEUC-HAL-3850-01] => (item={'id': 11301, 'device': {'id': 491, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/491/', 'name':
 'HEUC-HAL-3850-01', 'display_name': 'HEUC-HAL-3850-01'}, 'name': 'Gi0/0', 'type': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'form_fact
or': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': True, 'desc
ription': 'OOB Management Port', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 11297, 'url': 'https://netbox.cb
adnms.com/api/dcim/interfaces/11297/', 'device': {'id': 483, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/483/', 'name': 'Meraki-OOB
-HAL', 'display_name': 'Meraki-OOB-HAL'}, 'name': 'LAN 11', 'cable': 1177, 'connection_status': {'value': True, 'label': 'Connected'}}, 'co
nnection_status': {'value': True, 'label': 'Connected'}, 'cable': {'id': 1177, 'url': 'https://netbox.cbadnms.com/api/dcim/cables/1177/', '
label': 'HAL-1177'}, 'mode': {'value': 100, 'label': 'Access'}, 'untagged_vlan': None, 'tagged_vlans': [], 'tags': [], 'count_ipaddresses':
 0})
changed: [HEUC-HAL-3850-01] => (item={'id': 9920, 'device': {'id': 491, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/491/', 'name':
'HEUC-HAL-3850-01', 'display_name': 'HEUC-HAL-3850-01'}, 'name': 'Gi1/0/1', 'type': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'form_fac
tor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'de
scription': '', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 9918, 'url': 'https://netbox.cbadnms.com/api/dcim
/interfaces/9918/', 'device': {'id': 492, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/492/', 'name': 'HEUC-HAL-TS-01', 'display_nam
e': 'HEUC-HAL-TS-01'}, 'name': 'LAN-1', 'cable': 960, 'connection_status': {'value': True, 'label': 'Connected'}}, 'connection_status': {'v
alue': True, 'label': 'Connected'}, 'cable': {'id': 960, 'url': 'https://netbox.cbadnms.com/api/dcim/cables/960/', 'label': 'HAL-960'}, 'mo
de': None, 'untagged_vlan': None, 'tagged_vlans': [], 'tags': [], 'count_ipaddresses': 0})
changed: [HEUC-HAL-3850-01] => (item={'id': 9921, 'device': {'id': 491, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/491/', 'name':
'HEUC-HAL-3850-01', 'display_name': 'HEUC-HAL-3850-01'}, 'name': 'Gi1/0/2', 'type': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'form_fac
tor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'de
scription': '', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 9803, 'url': 'https://netbox.cbadnms.com/api/dcim
/interfaces/9803/', 'device': {'id': 488, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/488/', 'name': 'HEUC-HAL-ASR-01', 'display_na
me': 'HEUC-HAL-ASR-01'}, 'name': 'Gi0', 'cable': 1170, 'connection_status': {'value': True, 'label': 'Connected'}}, 'connection_status': {'
value': True, 'label': 'Connected'}, 'cable': {'id': 1170, 'url': 'https://netbox.cbadnms.com/api/dcim/cables/1170/', 'label': ''}, 'mode':
 None, 'untagged_vlan': None, 'tagged_vlans': [], 'tags': [], 'count_ipaddresses': 0})
changed: [HEUC-HAL-3850-01] => (item={'id': 9922, 'device': {'id': 491, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/491/', 'name':
'HEUC-HAL-3850-01', 'display_name': 'HEUC-HAL-3850-01'}, 'name': 'Gi1/0/3', 'type': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'form_fac
tor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'de
scription': '', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 9786, 'url': 'https://netbox.cbadnms.com/api/dcim
```

Resulting Switch Config

HAL-3850

```
Interface                      Status         Protocol Description
Vl1                            admin down     down
Vl11                           up             up
Gi0/0                          up             up       [<- Meraki-OOB-HAL - LAN 11 ->]
Gi1/0/1                        up             up       [<- HEUC-HAL-TS-01 - LAN-1 ->]
Gi1/0/2                        up             up       [<- HEUC-HAL-ASR-01 - Gi0 ->]
Gi1/0/3                        up             up       [<- HEUC-HAL-ASR-02 - Gi0 ->]
Gi1/0/4                        up             up       [<- HEUC-HAL-FI-A - Mgmt0 ->]
Gi1/0/5                        up             up       [<- HEUC-HAL-FI-B - Mgmt0 ->]
Gi1/0/6                        up             up       [<- HEUC-HAL-MDS-A - Mgmt0 ->]
Gi1/0/7                        up             up       [<- HEUC-HAL-MDS-B - Mgmt0 ->]
Gi1/0/8                        up             up       [<- HEUC-HAL-ASA4110-01 - Mgmt0 ->]
Gi1/0/9                        up             up       [<- HEUC-HAL-ASA4110-02 - Mgmt0 ->]
Gi1/0/10                       up             up       [<- HEUC-HAL-N9K-01 - Mgmt0 ->]
Gi1/0/11                       up             up       [<- HEUC-HAL-N9K-02 - Mgmt0 ->]
Gi1/0/12                       down           down
Gi1/0/13                       up             up       [<- HEUC-HAL-ASA4110-01 - Eth1/8 ->]
Gi1/0/14                       up             up       [<- HEUC-HAL-ASA4110-02 - Eth1/8 ->]
Gi1/0/15                       down           down
```

Automatic Label Generation with Netbox Data

```python
#!/usr/local/bin/python3

import requests
import json
import csv

device_url = "https://netbox.cbadnms.com/api/dcim/interfaces/?device_id=491"

payload = ""
headers = {
    'Authorization': "Token ",
    'Content-Type': "application/json",
    'Accept': "application/json",
    'cache-control': "no-cache",
    'Postman-Token': "e46ffd75-f6f8-451c-a77f-d1d558968dd2"
    }

response = requests.request("GET", device_url, data=payload, headers=headers)
device_js = json.loads(str(response.text))

def getCables(device_js):
    with open('cable-data.txt', mode="w") as f:
        for interface in device_js['results']:
            if interface['cable']:
                print("Found cable " + str(interface['cable']['id']))
                cable_url = (interface['cable']['url'])
                response = requests.request("GET", cable_url, data=payload, headers=headers)
                cable = json.loads(str(response.text))
                f.write("HAL-" + str(cable['id']) + '\n')
                f.write(cable['termination_a']['device']['name'][5:] + '\n')
                f.write(cable['termination_a']['name'].upper() + '\n')
                f.write(cable['termination_b']['device']['name'][5:] + '\n')
                f.write(cable['termination_b']['name'].upper() + '\n')
                f.write('\n')

def makeCSV(cables):
    with open('cable-matrix.csv', mode='w') as f:
        fieldnames = ['Cable ID', 'Device A', 'Interface A', 'Device Z', 'Interface Z']
        writer = csv.writer(f)
        writer.writerows(cables)

getCables(device_js)
#makeCSV(device_js)
```

Resulting Patch Cable Labels

cable-label-15.pdf (1 page)

Search

HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11
HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11	HAL1177 HAL-3850-01 GI0/0 i-OOB-HAL LAN 11

Advanced Example: Velocloud SD-WAN on AWS Provisioning using API and Terraform

- YouTube

Please, No More Tools!

- This automation journey has an rapidly evolving, seemingly infinite toolchain
- Postman tool to interact with Web Service APIs
- Continuous Integration / Continuous Deployment (CI/CD) Tools
  - Travis, Jenkins, etc.
- Other Infrastructure as Code (IAC) Tools
  - Terraform, Cobbler, Chef, Puppet
- Python Libraries
  - Napalm, Paramiko, Netmiko
- Loads of SDKs by OEMs and community
- Terminals, shells, and add-ons

Do You Even Tool, Bro?

Thoughts on DevOps

- “DevOps is a cultural and professional movement, focused on how we build and operate high velocity organizations, born from the experiences of its practitioners” - Adam Jacob, CTO Chef
- There is no “The Business” - We’re all The Business
- How can these tools and strategies change my job and affect my business processes?

Where to Start?

- Evaluate your environment, and which components have mature tooling (APIs, SDKs, Ansible Modules, etc)
- Preferably, pick something you have domain knowledge of
- “To err is human; to propagate errors massively at scale is automation.”
- Read operations —> Config Generation —> Direct Config Manipulation
- Education / Community / Consulting & Professional Services
- Some low fruit
  - Auditing / Reporting
  - Labeling
  - Greenfield deployments, perhaps L2 switching

Resources

- Courses
  - Charles Severance Python for Everybody
  - A Cloud Guru - Git, Ansible, Chef, AWS, Azure…
  - https://www.tutorialspoint.com/ansible
- Books
  - Network Programmability and Automation: Skills for the Next-Generation Network Engineer
- Community
  - IRC Freenode #ansible, #ansible-network
  - Packet Pushers - Datanauts 166

Feedback

- Understandable?
- Useful?
- Labs?
- Less of…
- More of…

Future Topics

- Ansible 201
- Python 201
- Git 201
- Terraform
- Specific SDKs
- AWS networking
- Container networking
- Serverless
- CI/CD

Thanks for coming, I'll be here all week, be sure to tip the waitress