

Network Automation Foundations

Presented By CBTS & Arista Networks

8-6-2019
Craig Bruenderman
CCIE #46341

Rhetorical Questions

- Why are you here?
 - Do you have use cases in mind?
- Do you know how your business makes money?
 - What is your Unit of Value?
 - Knowing that will help you connect to your business
- What about automation makes you uncomfortable?

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
 - Anyone purchased tools in the past that didn’t deliver or were abandoned?
 - Reduce the mundane and get to workflows that move your career, and your business forward

Basic Toolkit

- Python Programming Language
- Fundamental Data Structures
- IDE
- API Concepts
- Python Extensibility Concepts



Python



- General-purpose object-oriented programming language developed since the 1980's by Guido van Rossum
- Open source, itself written in C
- Interpreted
 - Internally, your source code is translated into machine code
 - Byte code then runs in a platform virtual machine, similar to Java
- High-level language, so approachable, easy(ish) to read by humans
- 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")`

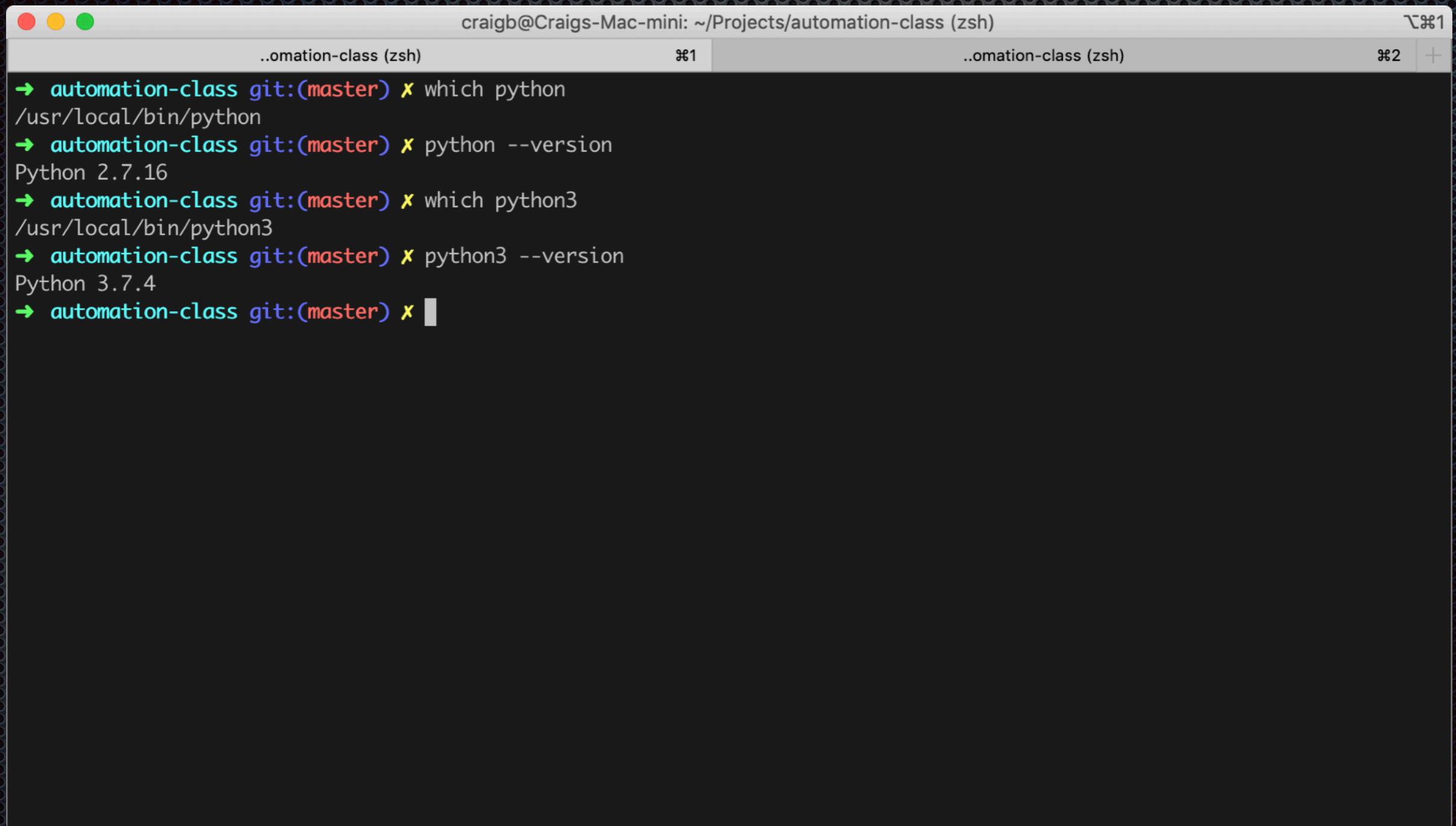
Python Run Modes

- Interactive
 - Run python from CLI
 - Do prototyping here
 - Can't do this with compiled languages
- Scripted
 - Capture your code in .py files

Python Extensibility

- 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 1.1 - Python Paths and Versions



craigb@Craigs-Mac-mini: ~/Projects/automation-class (zsh)

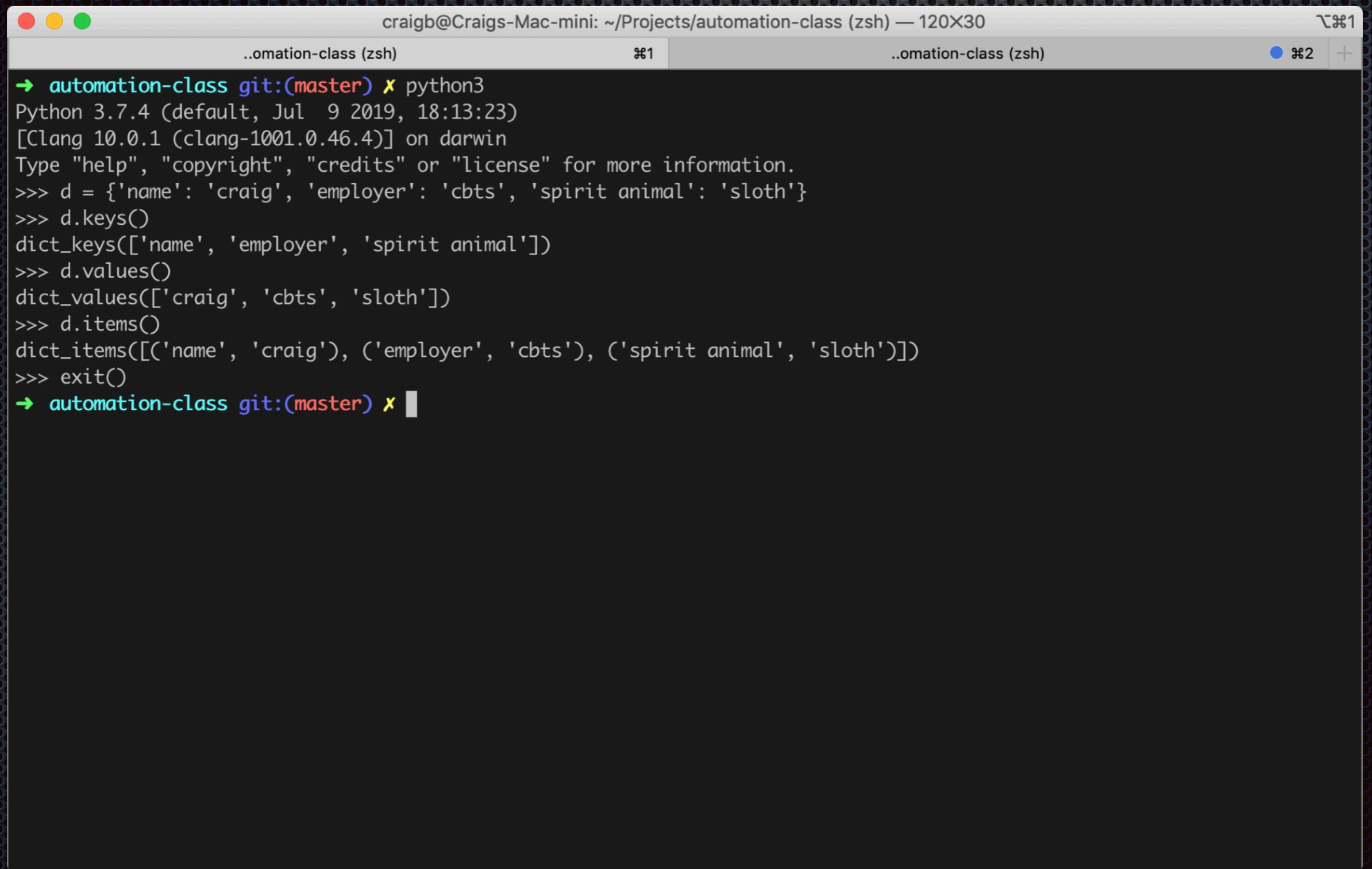
..omation-class (zsh) #1 ..omation-class (zsh) #2 +

```
→ automation-class git:(master) ✘ which python
/usr/local/bin/python
→ automation-class git:(master) ✘ python --version
Python 2.7.16
→ automation-class git:(master) ✘ which python3
/usr/local/bin/python3
→ automation-class git:(master) ✘ python3 --version
Python 3.7.4
→ automation-class git:(master) ✘
```

Lab 1.2 - Pip Package List

```
craigb@Craigs-Mac-mini: ~/Projects/automation-class (zsh)
..omation-class (zsh) #1 ..omation-class (zsh) #2 +
→ automation-class git:(master) ✘ pip list
DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 won't be maintained after that date. A future version of pip will drop support for Python 2.7.
Package           Version
-----
asn1crypto        0.24.0
astroid           1.6.5
backports.functools-lru-cache 1.5
bcrypt            3.1.7
certifi           2019.3.9
cffi              1.12.3
Click              7.0
configparser      3.7.4
cryptography     2.7
docutils          0.14
enum34            1.1.6
futures           3.2.0
ipaddress         1.0.22
isort              4.3.16
lazy-object-proxy 1.3.1
mccabe            0.6.1
meraki            0.34
netmiko           2.4.0
paramiko          2.6.0
pathspec          0.5.9
pip               19.0.3
pip-tools         3.8.0
pycparser         2.19
pylint             1.9.4
PyNaCl            1.3.0
pyparsing         2.4.0
pyserial          3.4
python-dateutil   2.8.0
PyYAML            5.1.1
scp               0.13.2
setuptools        40.8.0
singledispatch    3.4.0.3
six               1.12.0
textfsm           0.4.1
ucsmsdk          0.9.8
```

Lab 1.3 - Interactive Python



The screenshot shows a macOS terminal window with two tabs open. The active tab, labeled '#1', displays an interactive Python session. The session starts with the Python interpreter's banner:

```
→ automation-class git:(master) ✘ python3
Python 3.7.4 (default, Jul  9 2019, 18:13:23)
[Clang 10.0.1 (clang-1001.0.46.4)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
```

Then, the user enters several commands to demonstrate dictionary methods:

```
>>> d = {'name': 'craig', 'employer': 'cbts', 'spirit animal': 'sloth'}
>>> d.keys()
dict_keys(['name', 'employer', 'spirit animal'])
>>> d.values()
dict_values(['craig', 'cbts', 'sloth'])
>>> d.items()
dict_items([('name', 'craig'), ('employer', 'cbts'), ('spirit animal', 'sloth')])
```

Finally, the user exits the session:

```
>>> exit()
```

The terminal window has a dark theme and includes standard macOS window controls (red, yellow, green buttons) and a tab bar with another tab labeled '#2'.

Data Objects

- Everything in Python is an Object
 - Objects are just pieces of memory with values and sets of associated operations
- Python has several built-in data object types
 - Numbers, Strings, booleans
 - Lists, Dictionaries, Tuples
- More to this topic, but that will get us started

Lists

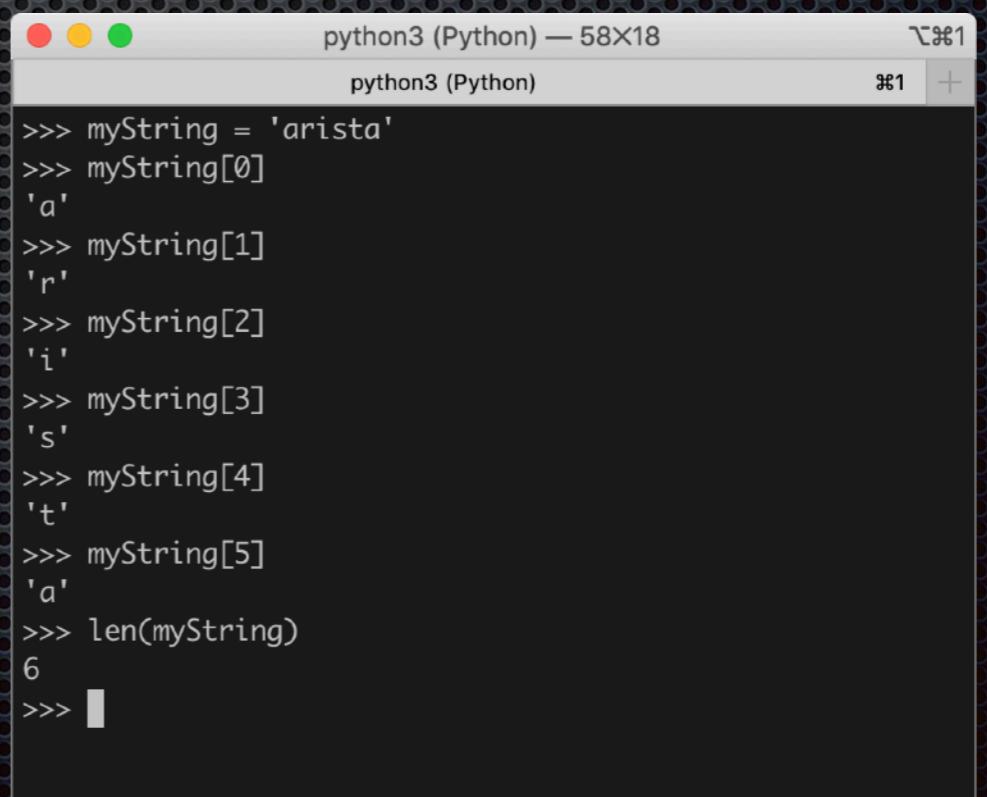
- Positionally ordered sequence of data objects
- Entries are indexed (referenced) by their integer index, which starts at 0
- In Python, a string is technically a list of single-character strings

- Examples

- In Python

- In JSON

- In YAML



A screenshot of a macOS terminal window titled "python3 (Python) — 58×18". The window shows a Python session where a string "arista" is assigned to a variable "myString". The session then prints each character of the string at its respective index from 0 to 5, followed by the length of the string (6). The terminal has a dark background with light-colored text.

```
>>> myString = 'arista'
>>> myString[0]
'a'
>>> myString[1]
'r'
>>> myString[2]
'i'
>>> myString[3]
's'
>>> myString[4]
't'
>>> myString[5]
'a'
>>> len(myString)
6
>>> █
```

Dictionaries

- Unordered mappings 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

Data Structure

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

Data Exchange Formats

- Designed to be exchanged/consumed by software
- Several to choose from
 - XML — Heavier weight
 - JSON (Javascript Object Notation) — Lighter weight, maps closely to native data structures in programming languages
 - YAML (YAML Ain't Markup Language) — Easiest to read
- Largely used for storing and exchanging:
 - Lists (Arrays)
 - Dictionaries (Associative Arrays)

Lab 2.1 - Lab Prep

- <http://bit.ly/cbautomation>

The screenshot shows a macOS terminal window with two tabs. The active tab, labeled '#1', displays the command `git clone https://github.com/craigbruenderman/automation-class.git` and its output, which shows the repository has been cloned successfully. The second tab, labeled '#2', shows the command `tree automation-class` and its output, which lists the contents of the cloned repository, including a `Lab_Guide.md` file, a `Lab_Guide.pdf`, a `Network_Foundations_Preso.key`, a `Network_Foundations_Preso.pdf`, a `README.md` file, and a `code` directory containing various configuration files and Python scripts.

```
craigb@Craigs-Mac-mini: ~ (zsh)
~ (zsh)                                     ..omation-class (zsh)
→ ~ git clone https://github.com/craigbruenderman/automation-class.git
Cloning into 'automation-class'...
remote: Enumerating objects: 115, done.
remote: Total 115 (delta 0), reused 0 (delta 0), pack-reused 115
Receiving objects: 100% (115/115), 49.82 MiB | 26.01 MiB/s, done.
Resolving deltas: 100% (64/64), done.
→ ~ tree automation-class
automation-class
├── Lab_Guide.md
├── Lab_Guide.pdf
├── Network_Foundations_Preso.key
├── Network_Foundations_Preso.pdf
└── README.md
└── code
    ├── 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.j2
        └── master.md

4 directories, 19 files
→ ~
```

Lab 2.2 - Data Structures

```
python -i lab2.2.py (Python) — 156x47
python (Python)
python git:(master) ✘ python -i lab2.2.py
>>> type(dns_servers)
<type 'list'>
>>> dns_servers
['8.8.8.8', '8.8.4.4']
>>> core1
{'vlans': [{"name": "Data", "id": 10}, {"name": "Voice", "id": 50}], "hostname": "core1.cbts.com", "vendor": "arista", "users": {"admin": 15, "craigb": 0}, "dns": ["8.8.8.8", "8.8.4.4"]}
>>> as_python()
----- As Python Native List -----
<type 'list'>
['8.8.8.8', '8.8.4.4']

----- As Python Native Dictionary -----
<type 'dict'>
{'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"}]}

>>> as_yaml(core1)
----- As YAML -----
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

>>> as_json(core1)
----- As JSON -----
[{"vlans": [{"name": "Data", "id": 10}, {"name": "Voice", "id": 50}], "hostname": "core1.cbts.com", "vendor": "arista", "users": {"admin": 15, "craigb": 0}, "dns": ["8.8.8.8", "8.8.4.4"]}

>>> █
```

Lab 2.3 - Convert Between Data Structures

The screenshot shows a terminal window with a file browser on the left and a code editor on the right.

File Browser (Left):

- automation-class
 - .git
 - code
 - group_vars
 - host_vars
 - reports
 - facts
 - facts.j2
 - master.md
 - master.pdf
 - hosts
 - lab2.2.py
 - lab2.3.py
 - lab4.1.py
 - lab7.1.yml
 - lab7.2.yml
 - lab8.1.yml
 - ports.json
 - ports.yml
 - .DS_Store

Code Editor (Right):

ports.yml — ~/Projects/automation-class

	ports.yml	Lab_Guide.md	lab2.2.py	lab4.1.py
1	- interface: FastEthernet0/28			
2	description: AppleTV			
3	mode: access			
4	vlan: 10			
5				
6	- interface: FastEthernet0/29			
7	description: IP Camera			
8	mode: access			
9	vlan: 11			
10				
11	- interface: FastEthernet0/30			
12	description: ESXi Host			
13	mode: trunk			
14				
15	- interface: FastEthernet0/31			
16	description: IP Phone			
17	mode: access			
18	vlan: 12			
19				

LF UTF-8 YAML ⌂ master ⌂ Fetch ⌂ GitHub ⌂ Git (4)

Lab 2.3 - Convert Between Data Structures

```
craigb@Craigs-Mac-mini: ~/Projects/automation-class/code (zsh)
..on-class/code (zsh) #1 ..omation-class (zsh) #2 +
➔ code git:(master) ✘ ls
group_vars hosts lab2.3.py lab7.1.yml lab8.1.yml ports.yml
host_vars lab2.2.py lab4.1.py lab7.2.yml ports.json reports
➔ code git:(master) ✘ python3 -i lab2.3.py
<class 'list'> 4

<class 'dict'>
<class 'dict'>
<class 'dict'>
<class 'dict'>

[{'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'},
 {'description': 'IP Phone',
 'interface': 'FastEthernet0/31',
 'mode': 'access',
 'vlan': 12}]
>>>
➔ code git:(master) ✘ ls
group_vars hosts lab2.3.py lab7.1.yml lab8.1.yml ports.yml
host_vars lab2.2.py lab4.1.py lab7.2.yml ports.json reports
➔ code git:(master) ✘
```

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

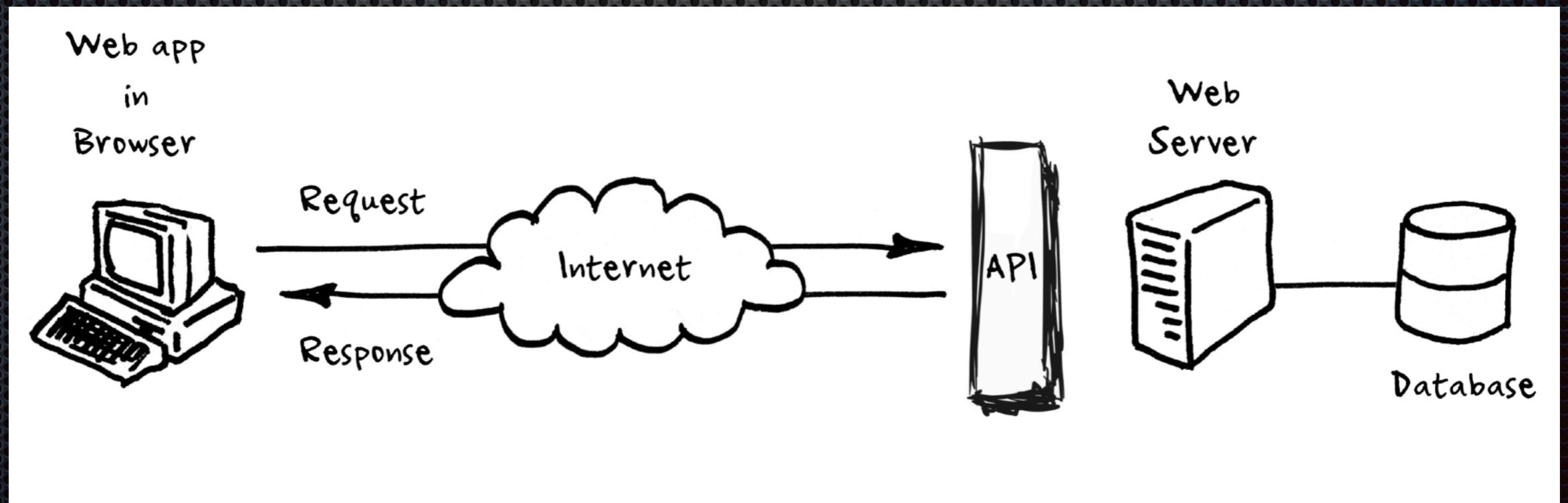
- Visual Studio Code
 - Multi-platform, Open Source, developed by Microsoft
- Atom
 - Multi-platform, Open Source, developed by Google
- PyCharm
 - Multi-platform, Freemium, developed by JetBrains
- Sublime Text
 - Multi-platform, Commercial

API Overview

- Application Programming Interface
- In basic terms, APIs just allow applications to communicate with one another
- Concept of Client and Server still exist, except Client can be a piece of software
- As an approximation, APIs are to software what CLIs are to humans
- 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”

Web Services API

- Data exchange format is XML or JSON
- This lets us represent complex data structures
- Web Services APIs exchange data using HTTP(s) as transport
- The Client can be a literal web browser, a tool like Postman, or a piece of software like a Python script



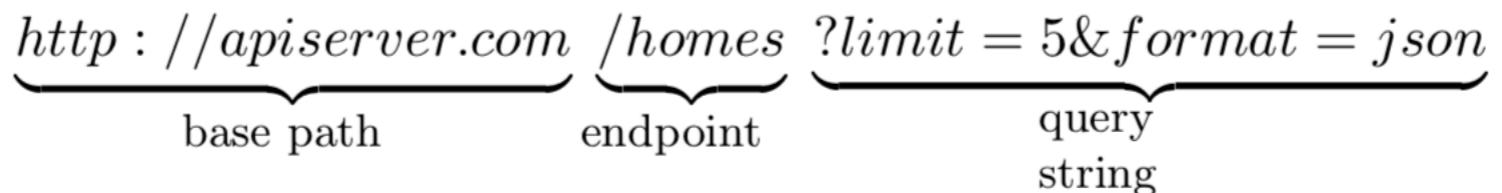
REST Overview

- An architectural style defined by a set of principles developed by Roy Fielding in his PhD work
 - REST is a style, not a standard or a protocol
- Focuses on resources (that is, things, rather than actions) and ways to access the resources
- Resources are different types of information accessed through URLs
- Just like going to a URL in your browser retrieves an information resource
- The URLs are accompanied by an HTTP method that specifies how you want to interact with the resource

Database	HTTP Method
Create	POST
Read	GET
Update	PUT
Delete	DELETE

REST Overview

- The base path (or base URL or host) refers to the common path for the API
- The endpoint refers to the end path of the endpoint
- The query string contains parameters for the endpoint
- The Client would also apply one of the HTTP method verbs (GET, PUT, POST, DELETE are the common ones)



http : //apiserver.com */homes* *?limit = 5&format = json*

base path endpoint query string

Web Service API - Arista eAPI

- eAPI is Arista's Web Service API
 - Provides a method for simply interacting with Arista gear without CLI interaction
- HTTP(s) transport
- JSON Request and Response
- eAPI allows CLI Commands to be issued remotely
- eAPI returns output as JSON, generally in key-value pairs

Navigating Web Service APIs

- Browser-based interaction
- Postman interaction
- Python REST interaction with *requests* library
- Python specific client library interaction (like pyeapi)
- Ansible interaction

Lab 3 - eAPI Command Explorer

The screenshot shows two windows side-by-side. On the left is the 'Command API Explorer' window, which includes a 'Simple Request' tab and a 'Script Editor' tab. The 'Simple Request' tab is active, displaying fields for 'API Endpoint' (https://192.168.0.14/command-api), 'Version' (1), 'Format' (json), 'Timestamps' (false), 'AutoComplete' (false), 'ExpandAliases' (false), and an 'ID' field (EapiExplorer-1). Below these fields is a 'Submit POST request' button. At the bottom of the window are two tabs: 'Request Viewer' and 'Response Viewer'. The 'Request Viewer' tab contains the command '1'. The 'Response Viewer' tab is currently empty. On the right is a 'Response Viewer' window titled 'Response Viewer' containing a JSON response. The JSON output is as follows:

```
1 {  
2   "jsonrpc": "2.0",  
3   "id": "EapiExplorer-1",  
4   "result": [  
5     {},  
6     {  
7       "interfaces": {  
8         "Ethernet2": {  
9           "lastStatusChangeTimestamp": 1498704273.1386933,  
10          "name": "Ethernet2",  
11          "interfaceStatus": "connected",  
12          "autoNegotiate": "unknown",  
13          "burnedInAddress": "00:0c:29:e7:43:09",  
14          "loopbackMode": "loopbackNone",  
15          "interfaceStatistics": {  
16            "inBitsRate": 0,  
17            "inPktsRate": 0,  
18            "outBitsRate": 0,  
19            "updateInterval": 300,  
20            "outPktsRate": 0  
21          },  
22          "mtu": 9214,  
23          "hardware": "ethernet",  
24          "duplex": "duplexFull",  
25          "bandwidth": 0,  
26        }  
27      }  
28    }  
29  }  
30 }
```

Lab 4.1 - Add a user with pyeapi

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh)
arista@ip-10-33-5-52: ~/automation-class/code (ssh) ⌘1 ..omation-class (zsh) ⌘2 +
arista@ip-10-33-5-52:~/automation-class/code$ python lab4.1.py
Users on the device

{'admin': {'format': '5',
            'nopassword': False,
            'privilege': '15',
            'role': 'network-admin',
            'secret': '$1$5085YVn$HrXcf0ivJEnISTMb6xrJc.',
            'sshkey': ''},
 'arista': {'format': '5',
            'nopassword': False,
            'privilege': '15',
            'role': '',
            'secret': '$1$4VjIjfd1$XkUVulbNDESHFzcxDU.Tk1',
            'sshkey': 'ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQC6bJB3TkBEQZ9jNy01kbdU0P20gZ1D72CvsPNZ5S4bbcIBNTT/MHX8GwyLmM9k+ih
HK2JtRhWFcdsm9MojRgjAuzw4wn/6pa92y/93GvaYL//d0BXrHctZsX3PX7TZFL9VVBVA8aFp5iXxEM8uyKWhxnBo/D0eR25Jed4gHVHQMi6Hyh7eKRpE3E6kvRlSkh
NikZ5EwdoM7lg2i6rfj7+o3G6isGtxliMZD98N6qWW79U6euS072qkK/q3dfgyHdd8a8MD5VLWbYR9ikhKwpXAmxcFn5aRllqXJ++QAW0N078noI91ICRxpAuQSzgrn
tdwXdyFWiqyiD3AxK28qWZ arista@labaccess'},
 'testuser': {'format': 'sha512',
              'nopassword': False,
              'privilege': '15',
              'role': 'network-admin',
              'secret': '$6$/ZIghqg7TS/LDY0J$yz/RMPf29PBfv71mSkQjcx0gmoolmchTLtEdX1dio1.HubQEM090JwbcQ6kcxhquEHzRQzszMlybCnaMEA
oxf/',
              'sshkey': ''}}}

arista@ip-10-33-5-52:~/automation-class/code$
```

Lab 4.2 - Get VLAN data with pyeapi

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh) ✘ 1
arista@ip-10-33-5-52: ~/automation-class/code (ssh) ✘ 1
..omation-class (zsh) ✘ 2 +
```

d is a dictionary of VLANs

```
VLAN0012
>>>
>>>
>>> type(d)
<type 'dict'>
>>> d
{'1': {'state': 'active', 'name': 'default', 'vlan_id': '1', 'trunk_groups': []}, '12': {'state': 'active', 'name': 'V
LANS0012', 'vlan_id': '12', 'trunk_groups': []}, '34': {'state': 'active', 'name': 'VLAN0034', 'vlan_id': '34', 'trunk_
groups': []}}
>>> d.keys()
['1', '12', '34']
>>> d['12']['name']
'VLAN0012'
>>> d['12']['state']
'active'
>>> █
```

Expanded Toolkit

- Git
- Github
- Ansible



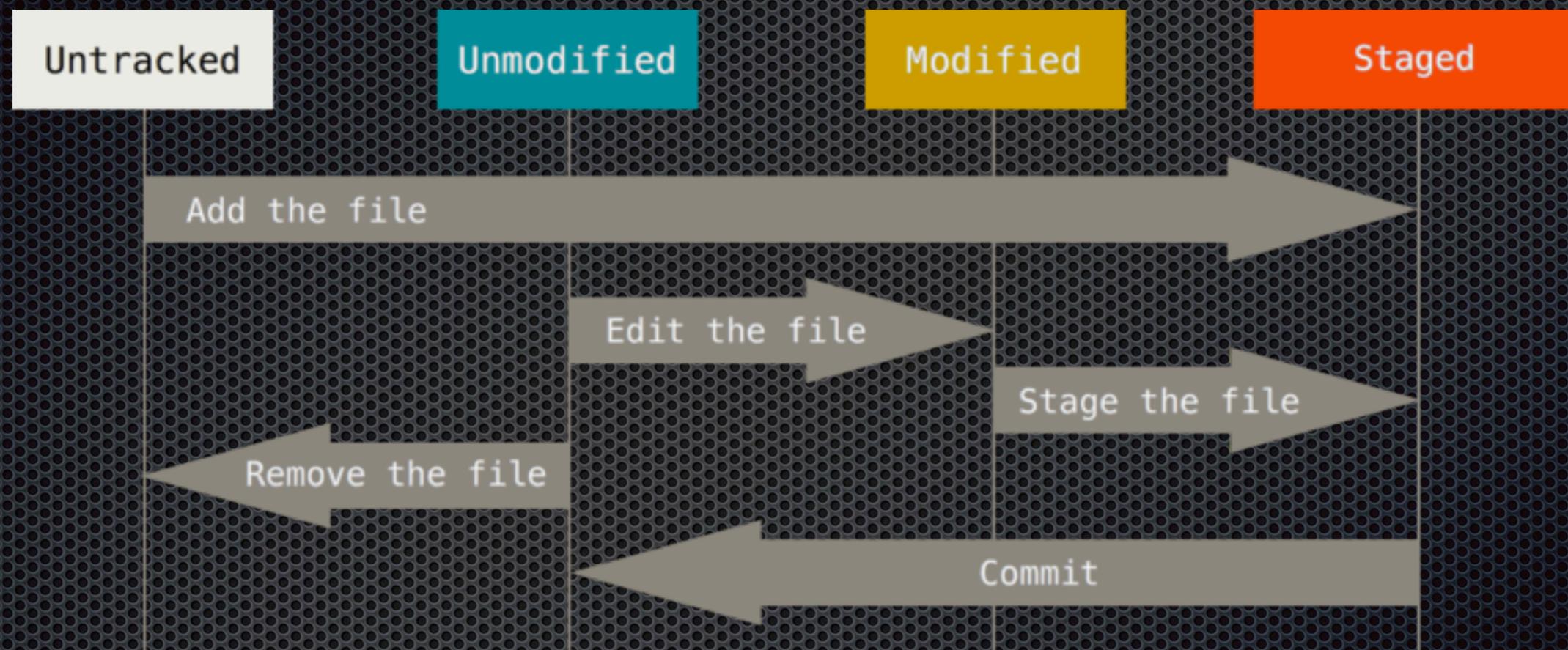
Git Overview

- **Version Control System** - System in which we store and maintain *versioned* repository of source code
 - Point in time snapshots of the repository are maintained
 - Enables collaboration between multiple contributors
- Git is one such free source/version control system that is distributed by design
- Built for non-linear development
- Excellent free book <https://git-scm.com/book/en/v2>

Git Overview

- **Repository** - database containing all of a projects files and metadata
- **Commit** - An entry in the Git repository recording metadata for each change introduced to the repo with a description
 - Captures the exact state of the repo
 - Each commit gets a SHA1 hash to identify it
 - Acts as a credit (blame) log

Life of a file in Git



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
- The main line of the repo is called *master*, *branches can be named anything meaningful*
- 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

- This is a bit of an advanced topic and often confusing
- This lab is Very optional

Github



- ❖ Since Git is a distributed VCS, where is the central source for the repo?
 - Run your own centralized Git server
 - Use a SaaS option (Github, Bitbucket, Code Commit, etc)
- ❖ 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 5.4 - 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 @@
1 -[veos]
2 -192.168.0.14
1 +[webservers]
2 +web1 ansible_host=10.0.0.1
3 +web2 ansible_host=10.0.0.2
4 +
5 +[database_servers]
6 +db1 ansible_host=172.16.99.1
7 +
8 +#[veos]
9 +#[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

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

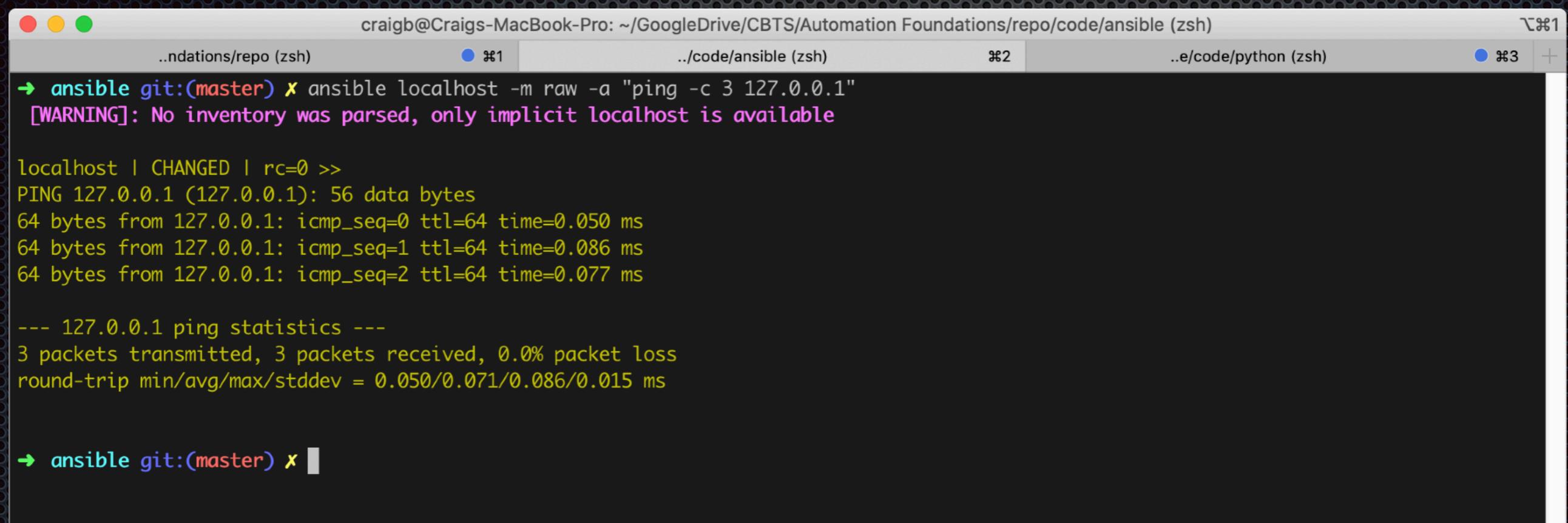
Ansible Terminology

- **Tasks** – Combines a module action with a task name
 - Basic unit of work in Ansible
- **Play** – mapping of task instructions to the target(s) they will run against
- **Playbook** — collection of at play(s) by which Ansible orchestrates, configures, administers, or deploys systems
- **Inventory** — file that defines hosts and groups targeted 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 6.1 - Ansible Ad Hoc Command



The screenshot shows a macOS terminal window with three tabs open:

- ..ndations/repo (zsh)
- ..code/ansible (zsh) (selected tab)
- ..e/code/python (zsh)

The terminal output is as follows:

```
craigb@Craigs-MacBook-Pro: ~/GoogleDrive/CBTS/Automation Foundations/repo/code/ansible (zsh)
→ 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 6.2 - Ad Hoc Package Installation

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh) arista@ip-10-33-5-52: ~/automation-class/code (ssh) ..omation-class (zsh) arista@ip-10-33-5-52:~/automation-class/code$ ansible localhost -m apt -a "name=yamllint" --become -k -c local SSH password:  
[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'  
  
localhost | CHANGED => {  
    "cache_update_time": 1553007139,  
    "cache_updated": false,  
    "changed": true,  
    "stderr": "",  
    "stderr_lines": [],  
    "stdout": "Reading package lists...\nBuilding dependency tree...\nReading state information...\nThe following NEW packages will be installed:\nyamllint\n  upgraded, 1 newly installed, 0 to remove and 65 not upgraded.\nNeed to get 0 B/23.4 kB of archives.  
After this operation, 124 kB of additional disk space will be used.  
Selecting previously unselected package yamllint.\r\n(Reading database ... \r(Reading database ... 5%\r(Reading database ... 10%\r(Reading database ... 15%\r(Reading database ... 20%\r(Reading database ... 25%\r(Reading database ... 30%\r(Reading database ... 35%\r(Reading database ... 40%\r(Reading database ... 45%\r(Reading database ... 50%\r(Reading database ... 55%\r(Reading database ... 60%\r(Reading database ... 65%\r(Reading database ... 70%\r(Reading database ... 75%\r(Reading database ... 80%\r(Reading database ... 85%\r(Reading database ... 90%\r(Reading database ... 95%\r(Reading database ... 100%\r(Reading database ... 179755 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-db (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 65 not upgraded.",  
        "Need to get 0 B/23.4 kB of archives.",  
        "After this operation, 124 kB of additional disk space will be used.",  
        "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 ... 179755 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)",  
        ""
```

Lab 6.3 - Ansible Ad Hoc Against Inventory

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh)  #1 ..omation-class (zsh)  #2 +  
arista@ip-10-33-5-52:~/automation-class/code (ssh)  
arista@ip-10-33-5-52:~/automation-class/code (ssh)$ ansible veos -i hosts -m raw -a "show version" -u arista -k  
SSH password:  
192.168.0.14 | 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 29 minutes  
Total memory:          3977260 kB  
Free memory:           3197824 kB  
  
Warning: Permanently added '192.168.0.14' (ECDSA) to the list of known hosts.  
Shared connection to 192.168.0.14 closed.  
  
192.168.0.11 | CHANGED | rc=0 >>  
Arista vEOS  
Hardware version:  
Serial number:      spine2  
System MAC address: 2cc2.6094.d76c  
  
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 29 minutes  
Total memory:          3977260 kB  
Free memory:           3196868 kB  
  
Warning: Permanently added '192.168.0.11' (ECDSA) to the list of known hosts.  
Shared connection to 192.168.0.11 closed.  
  
192.168.0.10 | CHANGED | rc=0 >>  
Arista vEOS  
Hardware version:
```

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.
 - Can even be dynamically generated programmatically, from CMDB or the like
- 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 places 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) — 122X27
..on-class/code (zsh)         ⌘1      ~ (zsh)      ⌘2 + ⌘⌘1

➔ 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

```
tab7.1.yml x

1  ---
2  # Name of the playbook
3  - name: Our first Ansible playbook
4  # List of hosts or host group to run against
5  hosts: localhost
6  # Toggle to run with root privileges
7  become: true
8
9  # Variables we define to use within this playbook
10 vars:
11   # Password to use when running command as root
12   ansible_become_password: "@rista123"
13
14 # List of tasks to be executed
15 tasks:
16   # Name of one task
17   - name: Install some packages
18     # Action of this task - in this case, the apt package manager module
19     apt:
20       # This is an Ansible looping technique
21       # {{ item }} is an iterator variable which will take
22       # the value of each item in the list from with_items
23       name: "{{ item }}"
24       state: present
25     # List of the packages to be installed
26     with_items:
27       - pandoc
28       - jq
29
```

Ansible Challenges

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

Lab 7.1 - Use Ansible Playbook to Install Packages

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh) ⌘1 ..omation-class (zsh) ⌘2 +  
arista@ip-10-33-5-52:~/automation-class/code$ more ports.json  
[{"interface": "FastEthernet0/28", "vlan": 10, "description": "AppleTV", "mode": "access"}, {"interface": "FastEthernet0/29", "vlan": 11, "description": "IP Camera", "mode": "access"}, {"interface": "FastEthernet0/30", "description": "ESXi Host", "mode": "trunk"}, {"interface": "FastEthernet0/31", "vlan": 12, "description": "IP Phone", "mode": "access"}]  
arista@ip-10-33-5-52:~/automation-class/code$ ansible-playbook -i hosts lab7.1.yml  
  
PLAY [Our first Ansible playbook] *****  
  
TASK [Gathering Facts] *****  
ok: [localhost]  
  
TASK [Install some packages] *****  
changed: [localhost] => (item=[u'pandoc', u'jq'])  
  
PLAY RECAP *****  
localhost : ok=2    changed=1    unreachable=0    failed=0  
  
arista@ip-10-33-5-52:~/automation-class/code$ jq . ports.json  
[  
  {  
    "interface": "FastEthernet0/28",  
    "vlan": 10,  
    "description": "AppleTV",  
    "mode": "access"  
  },  
  {  
    "interface": "FastEthernet0/29",  
    "vlan": 11,  
    "description": "IP Camera",  
    "mode": "access"  
  },  
  {  
    "interface": "FastEthernet0/30",  
    "description": "ESXi Host",  
    "mode": "trunk"  
  },  
  {  
    "interface": "FastEthernet0/31",  
    "vlan": 12,  
    "description": "IP Phone",  
    "mode": "access"  
  }]  
arista@ip-10-33-5-52:~/automation-class/code$
```

Lab 7.2 - Use Ansible to interact with the Twilio 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



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 8.1 - Jinja Templating

```
arista@ip-10-33-5-52: ~/automation-class/code (ssh) TextEdit
arista@ip-10-33-5-52: ~/automation-class/code (ssh) ⌘1 ..omation-class (zsh) ⌘2 +
arista@ip-10-33-5-52:~/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: [192.168.0.11]
ok: [192.168.0.10]

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] ****
ok: [192.168.0.10]
ok: [192.168.0.14]
ok: [192.168.0.11]

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

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

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

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

arista@ip-10-33-5-52:~/automation-class/code$
```

Use Cases



Use Case: Bulk Config Audit

```
meraki-dns-check.py
```

```
1 #!/usr/local/bin/python3
2
3 from meraki import meraki
4 from tabulate import tabulate
5 from itertools import islice
6 import csv
7 import os
8
9 apikey = os.environ['MERAKI_API_KEY']
10 myOrgs = meraki.myorgaccess(apikey)
11 #print(myOrgs)
12
13 for i, entry in enumerate(myOrgs):
14     if entry['name'] == "CustomerName":
15         orgid = entry['id']
16
17 outputList = []
18 myNetworks = meraki.getnetworklist(apikey, orgid, suppressprint=True)
19 # Limiter for debugging
20 #limit = 3
21 limit = len(myNetworks)
22 print("%s Networks found" % limit)
23
24 for network in islice(myNetworks, limit):
25     deviceList = meraki.getnetworkdevices(apikey, network["id"], suppressprint=True)
26     for device in deviceList:
27         deviceUplinks = meraki.getdeviceuplink(apikey, network["id"], device["serial"], suppressprint=True)
28         for uplink in deviceUplinks:
29             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)})
30
31 print(tabulate(outputList, headers="keys"))
32
33 with open("pjs-devices.csv", "w") as csvfile:
34     fields = outputList[0].keys()
35     writer = csv.DictWriter(csvfile, fieldnames=fields)
36     writer.writeheader()
37     for row in outputList:
38         writer.writerow(row)
39 csvfile.close()
40
```

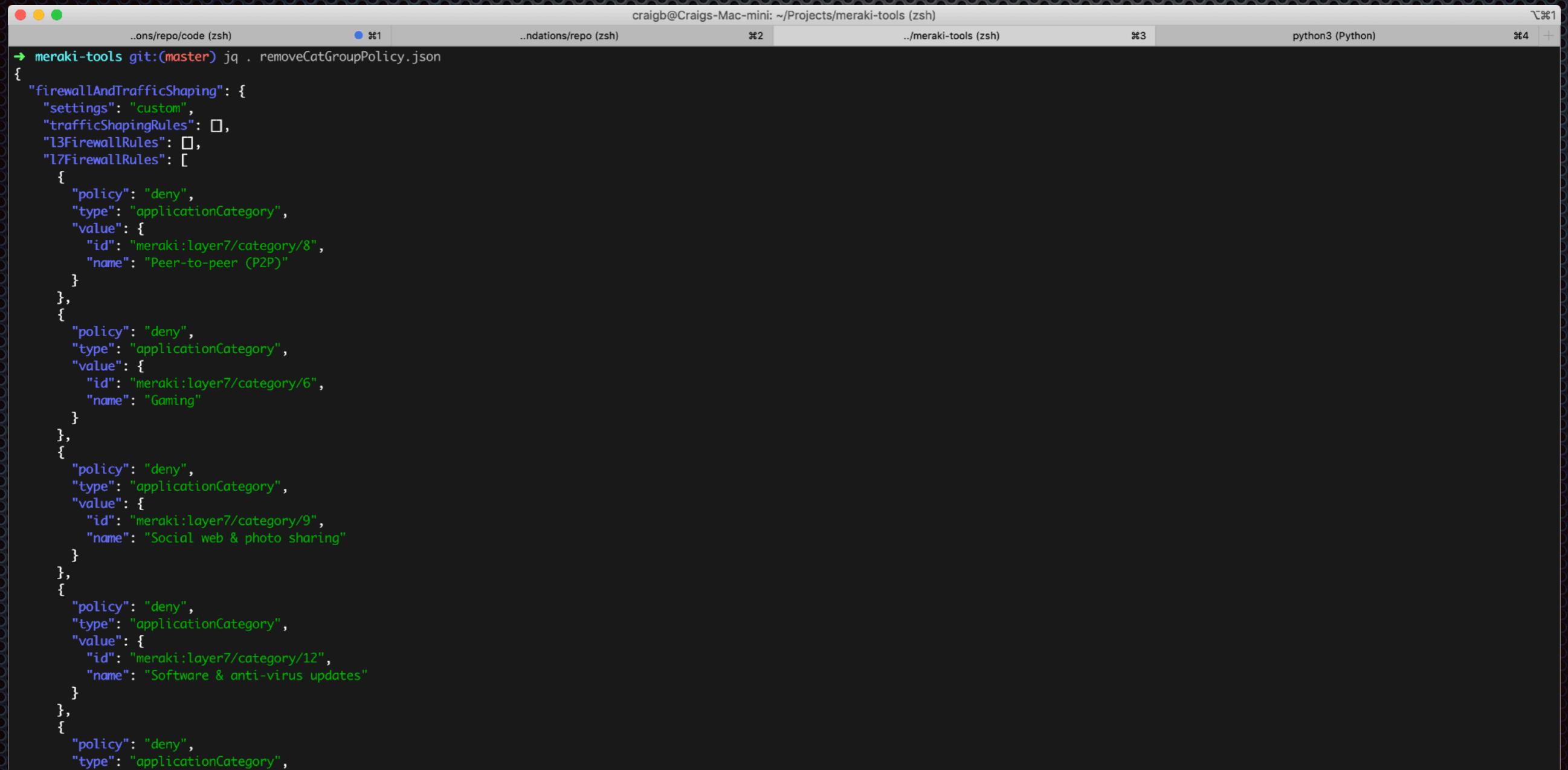
Use Case: Bulk Config Audit

The screenshot shows a terminal window with three tabs. The active tab (Tab 1) displays the output of a script named `meraki-dns-check.py`. The output indicates that 707 networks were found and lists their details, including Network Name, Device Name, Device Model, Interface, DNS, and Static IP. The script also reports that the organization operation was successful.

```
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          208.67.222.222, 208.67.220.220  True
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
```

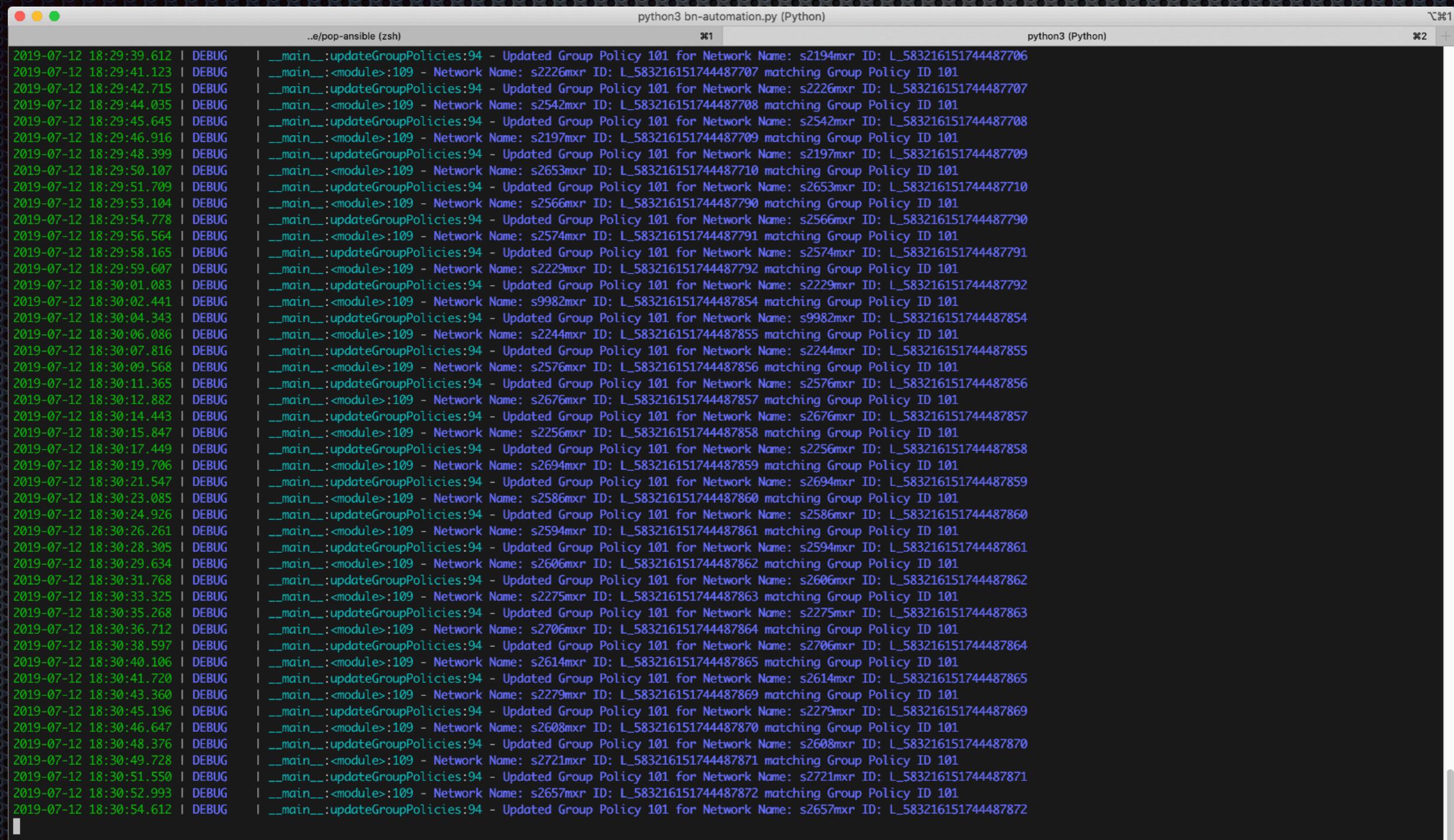
Use Case: Bulk Security Policy Change



The screenshot shows a macOS terminal window with five tabs open. The active tab, labeled #3, displays a JSON configuration file for Meraki security policies. The file defines several rules under the 'firewallAndTrafficShaping' section, specifically targeting application categories like P2P, Gaming, Social web & photo sharing, and Software & anti-virus updates.

```
..ons/repo/code (zsh)      #1 ..ndations/repo (zsh)      #2 craigb@Craigs-Mac-mini: ~/Projects/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",
        "value": {
          "id": "meraki:layer7/category/13",
          "name": "Cloud-based security"
        }
      }
    ]
  }
}
```

Use Case: Bulk Security Policy Change



```
python3 bn-automation.py (Python)  #1
..e/pop-ansible (zsh)  #2
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_583216151744487708
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.
- Twin sites in Toronto and Halifax
- Netbox DCIM 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

[◀ 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](#)

+ 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 Management switch for HEUC environment

Interfaces Show IPs

<input type="checkbox"/>	Name	LAG	Description	MTU	Mode	Cable	Connection	IPs
	Gi0/0		OOB Management Port	—	Access	HAL-1177	Meraki-OOB-HAL	
	Gi1/0/1			—	—	HAL-960	HEUC-HAL-TS-01	
	Gi1/0/2			—	—	#1170	HEUC-HAL-ASR-01	
	Gi1/0/3			—	—	#1171	HEUC-HAL-ASR-02	
	Gi1/0/4			—	—	HAL-1070	HEUC-HAL-FI-A	
	Gi1/0/5			—	—	HAL-1071	HEUC-HAL-FI-B	

Driving Ansible with Netbox Data

```
- 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) ..n Foundations (zsh) ..s/pop-ansible (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_factor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': True, 'description': 'OOB Management Port', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 11297, 'url': 'https://netbox.cbadnms.com/api/dcim/interfaces/11297/'}, 'device': {'id': 483, 'url': 'https://netbox.cbadnms.com/api/dcim/devices/483/', 'name': 'Meraki-00B-HAL', 'display_name': 'Meraki-00B-HAL'}, 'name': 'LAN 11', 'cable': 1177, 'connection_status': {'value': True, 'label': 'Connected'}}, 'connection_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_factor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'description': '', '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_name': 'HEUC-HAL-TS-01'}, 'name': 'LAN-1', 'cable': 960, 'connection_status': {'value': True, 'label': 'Connected'}}, 'connection_status': {'value': True, 'label': 'Connected'}, 'cable': {'id': 960, 'url': 'https://netbox.cbadnms.com/api/dcim/cables/960/'}, 'label': 'HAL-960'}, 'mode': 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_factor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'description': '', '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_name': '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_factor': {'value': 1000, 'label': '1000BASE-T (1GE)'}, 'enabled': True, 'lag': None, 'mtu': None, 'mac_address': None, 'mgmt_only': False, 'description': '', 'connected_endpoint_type': 'dcim.interface', 'connected_endpoint': {'id': 9786, 'url': 'https://netbox.cbadnms.com/api/dcim/interfaces/9786/'}})
```

Resulting Switch Config

Interface	Status	Protocol Description
Vl1	admin down	down
Vl11	up	up
Gi0/0	up	[<- Meraki-00B-HAL - LAN 11 ->]
Gi1/0/1	up	[<- HEUC-HAL-TS-01 - LAN-1 ->]
Gi1/0/2	up	[<- HEUC-HAL-ASR-01 - Gi0 ->]
Gi1/0/3	up	[<- HEUC-HAL-ASR-02 - Gi0 ->]
Gi1/0/4	up	[<- HEUC-HAL-FI-A - Mgmt0 ->]
Gi1/0/5	up	[<- HEUC-HAL-FI-B - Mgmt0 ->]
Gi1/0/6	up	[<- HEUC-HAL-MDS-A - Mgmt0 ->]
Gi1/0/7	up	[<- HEUC-HAL-MDS-B - Mgmt0 ->]
Gi1/0/8	up	[<- HEUC-HAL-ASA4110-01 - Mgmt0 ->]
Gi1/0/9	up	[<- HEUC-HAL-ASA4110-02 - Mgmt0 ->]
Gi1/0/10	up	[<- HEUC-HAL-N9K-01 - Mgmt0 ->]
Gi1/0/11	up	[<- HEUC-HAL-N9K-02 - Mgmt0 ->]
Gi1/0/12	down	down
Gi1/0/13	up	[<- HEUC-HAL-ASA4110-01 - Eth1/8 ->]
Gi1/0/14	up	[<- HEUC-HAL-ASA4110-02 - Eth1/8 ->]
Gi1/0/15	down	down

Cable Label Auto-Generation with Netbox

```
#!/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"
}

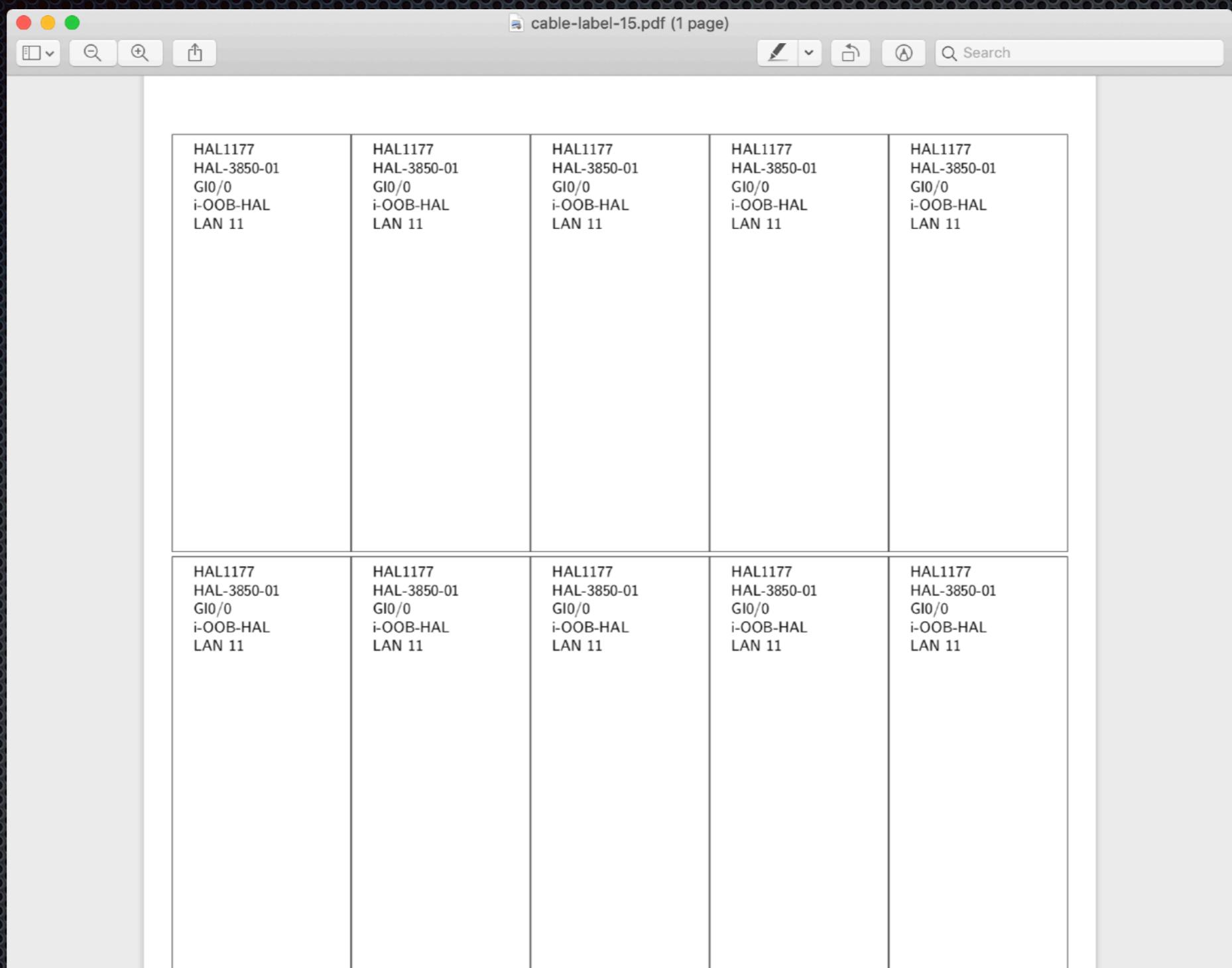
response = requests.request("GET", device_url, data=payload, headers=headers)
device_js = json.loads(str(response.text))

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.writerow(fieldnames)
        writer.writerows(cables)

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

Resulting Patch Cable Labels



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
 - E.g., Napalm, Paramiko, Netmiko
 - Loads more by OEMs and community
- Terminals, shells, and utilities



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
 - Embrace this if you want IT to be thought of as more than a cost center
- How can these tools and strategies change my job and affect my business processes?

Where to Start?

- Software engineering is a profession, just like network engineering, but that's ok
 - First build tools for yourself, then your team, then your customers
- Evaluate which components of your environment have automation tooling (APIs, SDKs, Ansible Modules, and pick something you have domain knowledge in
- “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
 - Greenfield deployments, perhaps L2 switching

Resources

- Labs
 - Arista ATD Labs
 - SRE Labs <https://labs.networkreliability.engineering/>
- Courses
 - Charles Severance [Python for Everybody](#)
 - [A Cloud Guru](#) - Git, Ansible, Chef, AWS, Azure...
 - <https://www.tutorialspoint.com/ansible>
 - <https://www.networktocode.com/>
- Books
 - [Network Programmability and Automation: Skills for the Next-Generation Network Engineer](#)
- Community
 - IRC Freenode #ansible, #ansible-network
 - Packet Pushers - [Datanauts 166](#)

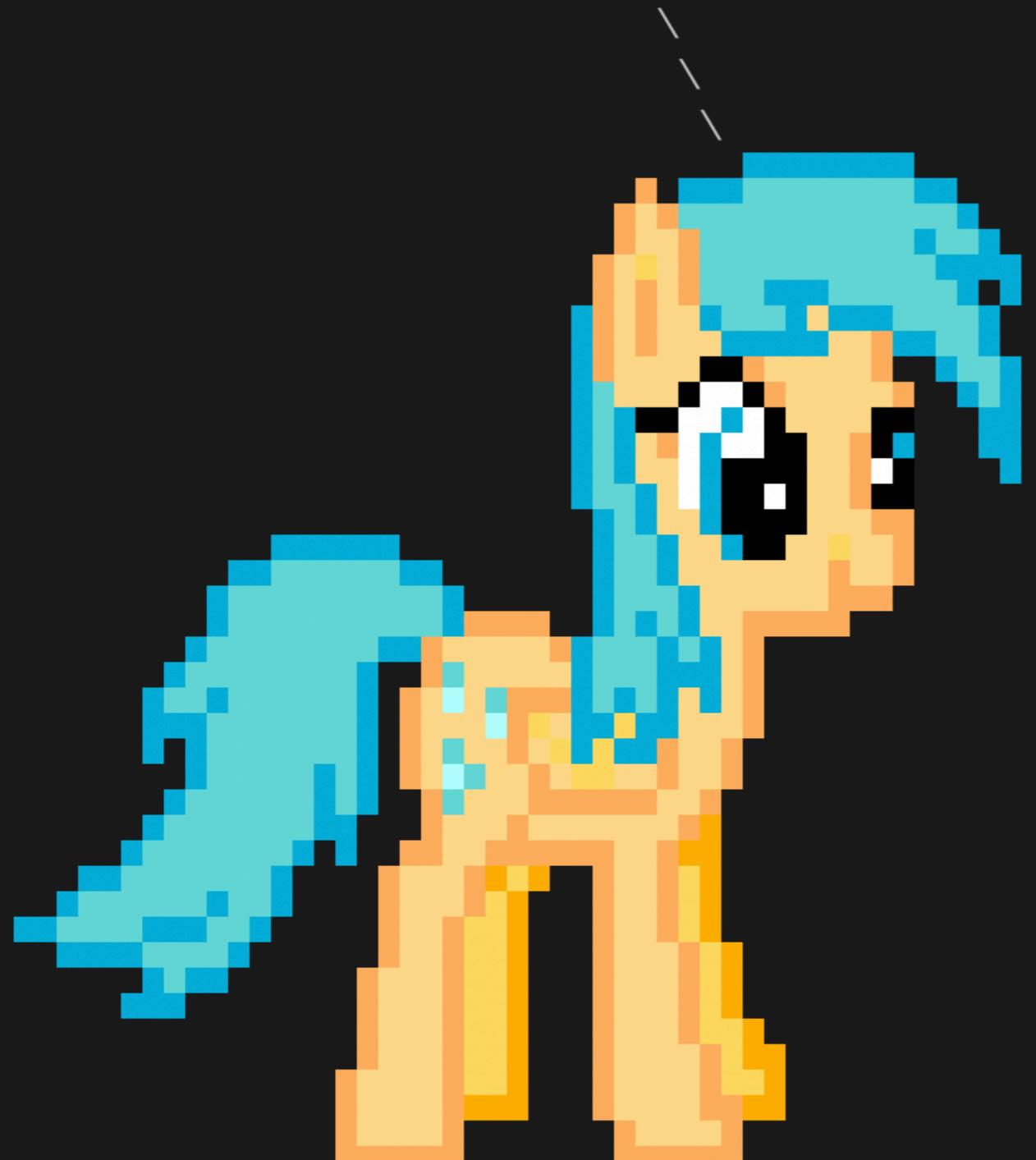
Feedback

- 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

< I'm here all week. Tip the wait staff >



→ ~ |

Happy Hour Down the Street at Mile
Wide Beer - 636 Barret Ave

