

# Abstracting Network Automation Tasks with NAPALM

---



**Nick Russo**

NETWORK ENGINEER

@nickrusso42518 [www.njrusmc.net](http://www.njrusmc.net)



# Agenda



Introducing NAPALM

Set theory refresher

Custom parsers to help with infra as code  
... with unit tests!

The grand finale demo



# Introducing NAPALM

**Multi-vendor  
abstraction**

**Advanced  
network  
operations**

**By David Barroso**  
**[github.com/napalm-automation/napalm](https://github.com/napalm-automation/napalm)**



# Demo



Utilizing NAPALM "getters" and merge



```
>>> letters = {"a", "b", "c", "d"}  
>>> numbers = {0, 1, 2, 3, 4}  
>>> ps_courses = {"icmp", "igmp/mld", "ansible", "ftp"}  
>>> type(letters)  
<class 'set'>
```

## What Is a Set?

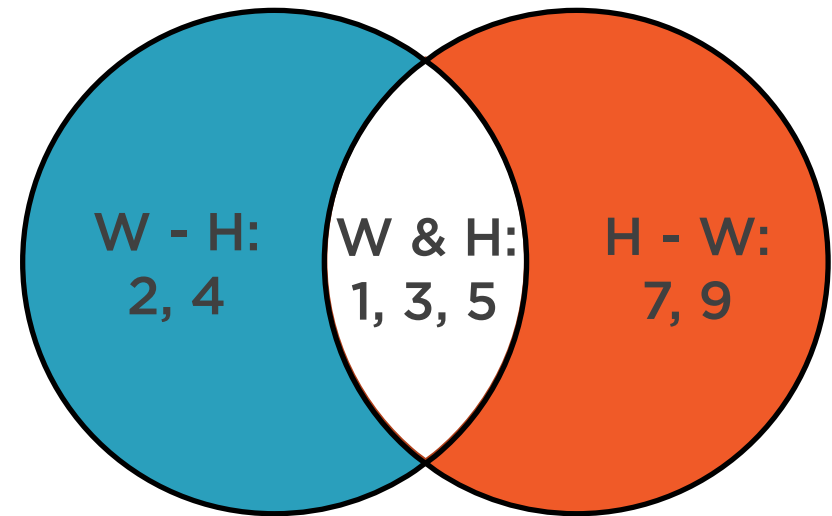
**An unsorted collection of unique elements**



```
>>> want = {1, 2, 3, 4, 5}
>>> have = {1, 3, 5, 7, 9}
>>> want - have
{2, 4}
>>> have - want
{9, 7}
>>> want & have
{1, 3, 5}
```

want =  
{1, 2, 3, 4, 5}

have =  
{1, 3, 5, 7, 9}



◀ Set intersection for verification



# Demo



## Set theory implementation in Python



# Solution Overview



Regex parsers



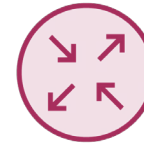
Inventory



Variables

For each host:

- Get config
- Parse RTs
- Set theory
- Merge/diff
- Commit diff



```
{% for each vrf %}  
  {% Add RT im %}  
  {% Del RT im %}  
  {% Add RT ex %}  
  {% Del RT ex %}
```





# Demo



## The grand finale



# What Else Can NAPALM Do?

**Config replace**

**Config rollback**

**Pythonic style;  
context manager**



```
driver = get_network_driver("ios")
with driver("hostname", "user", "pass") as conn:
    conn.load_replace_candidate(filename="new_config.txt")
    conn.commit_config()
    # oops, something went wrong
    conn.rollback()
```

NAPALM Alternative Implementation

<https://napalm.readthedocs.io/en/latest/>



# Midterm Review

More abstract

**NAPALM**

```
r = load_merge_candidate(config)
r.commit_config()
```

utilizes

**Netmiko**

```
r = conn.send_config_set(config)
```

utilizes

**Paramiko**

```
conn.send(config)
r = conn.recv(65535).decode(...)
```

Less Abstract



# Challenge: Too Slow!

