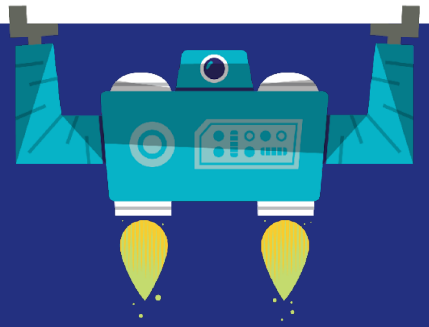
DevOps for NetOps

Rick “Shermdog” Sherman
Puppet Labs



puppet labs

Introduction

~9.5 years - Juniper Networks

- Professional Services
  - Identity and Policy Management
  - Workflow systems
- Security Business Unit
  - Cloud Architect
- Junos Manageability
  - PyEZ (Python micro-framework)
  - Ansible Modules
  - Onbox scripting
  - NetDev Evangelism

~3.5 months - Puppet Labs

- Release Engineering
  - Network Platform Expansion

Life of a Network Engineer

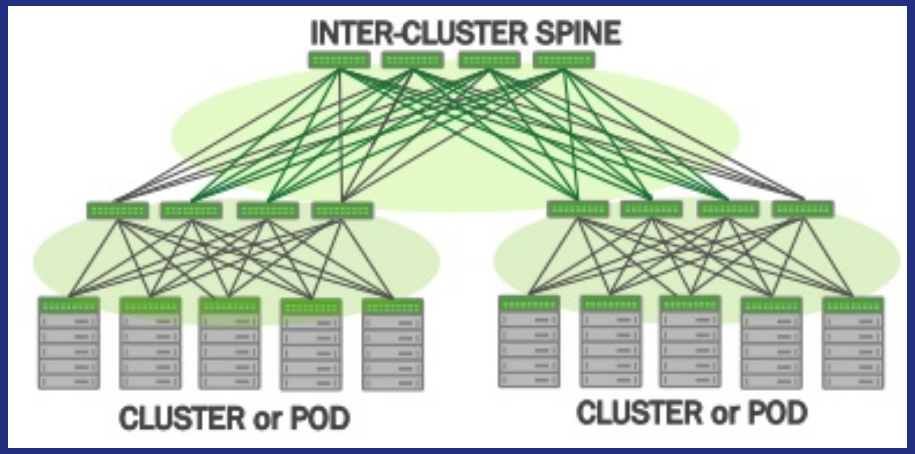
Let's make some generalizations (what could go wrong?)

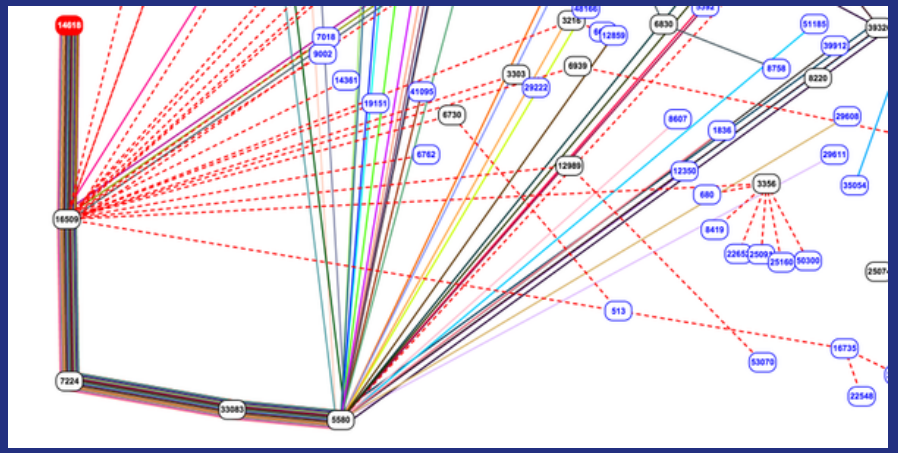
- Networks are a complex ecosystem inter-connected devices
  - Services are spread over multiple systems
  - Equipment is often heterogeneous
- Require a lot of planning, testing, and validation
- A lot of time is spent fire fighting
- Also a lot of mundane tasks

How does that differ from Sys Admins?

- Network devices have historically been closed systems with vendor specific CLIs
  - They often differ between the same vendor device types and versions
- Configurations are hundreds if not thousands of lines (per system)
- Configuration != Desired state
  - Often peering with other systems not under our control
- Vendors slow to introduce features, sometimes 18-24 months - upgrade cycle is just as long.
- Network Engineers typically do not have a Sys Admin or programming background

Inter-tubes? More like spaghetti o.O





Ad-hoc management is difficult

What is DevOps

- Collaborative
  - Tear down silos
    - We should all be working towards the same goal and have each other's back
- Systematic
  - Emphasis on the big picture. All the bandwidth and uptime in the world means nothing if the services fail
- Iterative
  - Work towards a series of goals
  - Don't have to boil the ocean - start small and get feedback often
- Automated
  - Build, Test, and Deliver at scale. Eliminate time sucks.
  - Infrastructure as Code

Change?

State of DevOps

https://puppetlabs.com/2015-devops-report

- High-performing IT organizations experience **60 times fewer failures and recover from failure 168 times faster** than their lower-performing peers. They also **deploy 30 times more frequently with 200 times shorter lead times**. Failures are unavoidable, but how quickly you detect and recover from failure can mean the difference between leading the market and struggling to catch up with the competition.
- Burnout can be prevented, and DevOps can help. **Burnout is associated with pathological cultures and unproductive, wasteful work.** The consequences of burnout are huge, both for individuals and for organizations. Organizations can fix the conditions that lead to burnout by fostering a supportive work environment and ensuring work is meaningful, and that **employees understand how their own work ties to strategic objectives**.

NetEng's "must become programmers"

You are not the CLI

Industry has rewarded memorizing CLI commands.

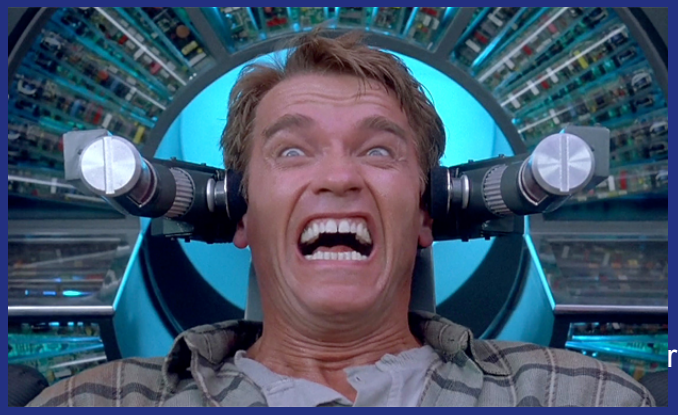
Network engineers are well versed in understanding complex problems and distributed systems.

Realize the value you can provide to your organizations - move beyond the CLI

Think like a programmer

- In basic terms, programming is the manipulation of data.
- You already know the core concepts of data types and how to manipulate them, the missing link is language and tools.

```
IT'S SHOWTIME
BECAUSE I'M GOING TO SAY PLEASE a
TALK TO THE HAND "a is true"
BULLSHIT
TALK TO THE HAND "a is not true"
YOU HAVE NO RESPECT FOR LOGIC
YOU HAVE BEEN TERMINATED
```



rnoldC

Separate the HOW from the WHAT

Regardless of the language you speak, you know what this is.

You know that you can open and close this object and you may also be able to lock and unlock it.

Hand crafted - artisanal configs

A tale of two configs

Cisco

```
hostname nanog
ip domain-name shermdog.com
ip name-server 10.0.0.1
ntp server 10.14.99.10
```

Juniper

```
system {
    host-name nanog;
    domain-name shermdog.com;
    name-server {
        10.0.0.1;
    }
    ntp {
        server 10.14.99.10;
    }
}
```

The How from the What

Cisco

```
hostname nanog
ip domain-name shermdog.com
ip name-server 10.0.0.1
ntp server 10.14.99.10
```

Juniper

```
system {
    host-name nanog;
    domain-name shermdog.com;
    name-server {
        10.0.0.1;
    }
    ntp {
        server 10.14.99.10;
    }
}
```

Where's the beef?

Data can come from a variety of sources - YAML, JSON, SQL, etc. Source control it!

```
---
host_name: nanog
domain: shermdog.com
dns: 10.0.0.1
ntp_server: 10.14.99.10
```

Templates

Cisco

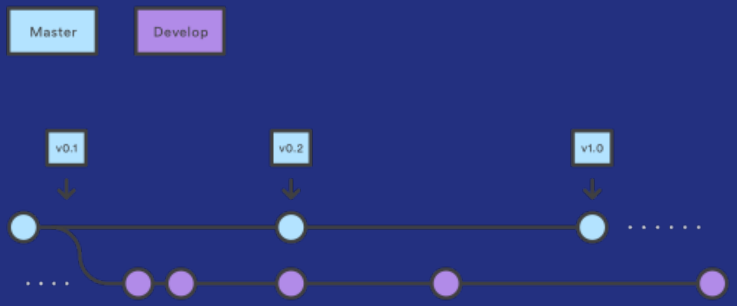
```
hostname {{ host_name }}
ip domain-name {{ domain }}
ip name-server {{ dns }}
ntp server {{ ntp_server }}
```

Juniper

```
system {
    host-name {{ host_name }};
    domain-name {{ domain }};
    name-server {
        {{ dns }};
    }
    ntp {
        server {{ ntp_server }};
    }
}
```

Git with the program

- Source control is *AMAZING*
- Git is a version control tool. It create a facility to store version history of files and folders (organized as projects). It has mechanism for teamwork and sharing with a foundation around file and history integrity.
- Unlike traditional source control where versions are stored as a set of diffs, Git stores a snapshot of the entire project – much like a file system. This gives users great flexibility to retrieve code throughout the history.



https://www.atlassian.com/git/tutorials/

Stop, Collaborate and Listen.

```
104     -  # can't autogen getters and setters because the default_<prop>
105     -  # functions are class functions
106     -  def deadtime
107     -    return :default if @resource[:deadtime] == :default &&
108     -                       @property_hash[:deadtime] ==
109     -                       Cisco::AaaServerGroup.default_deadtime
110     -    @property_hash[:deadtime]
111     -  end
112     -
113     -  def deadtime=(set_value)
114     -    set_value = Cisco::AaaServerGroup.default_deadtime if
115     -      set_value == :default
116     -    @property_flush[:deadtime] = set_value
117     -  end
118     -
        114 +  # can't autogen server_hosts, special array handling
119     115    def server_hosts
120     116      return [:default] if @resource[:server_hosts] &&
121     117                           @resource[:server_hosts][0] == :default
122     118                           @property_hash[:server_hosts] ==
123     -                           Cisco::AaaServerGroup.default_servers
        119 +                           @aaa_group.default_servers
```

chrisvanheuveln added a note 4 days ago

nit: this return (and others in this file) would be cleaner as:

```
return [:default] if
  @resource[:server_hosts] &&
  @resource[:server_hosts][0] == :default &&
  @property_hash[:server_hosts] == @aaa_group.default_servers
```

(PUP-3695) yum provider handling ... jhoblitt authored on Nov 24, 2014	fd58fa3	69 `def self.check_updates(enablerepo, disablerepo, c`
(PUP-4055) Make necessary chang... whopper authored on Sep 29	657ba78	70 `args = [command(:cmd), 'check-update']`
(PUP-1362) (PUP-1775) map yumhe... jhoblitt authored on Nov 24, 2014	f8853df	71 `args.concat(enablerepo.map { |repo| ["--enabler`
		72 `args.concat(disablerepo.map { |repo| ["--disabl`
(PUP-3695) yum provider handling ... jhoblitt authored on Nov 24, 2014	fd58fa3	73 `args.concat(disableexcludes.map { |repo| ["--di`
(PUP-1060) Respect yum enable an... adrienthebo authored on Apr 21, 2014	6dc0a0e	74
(PUP-1362) (PUP-1775) Use yum c... adrienthebo authored on May 16, 2014	101d1de	75 `output = Puppet::Util::Execution.execute(args, :failonfail => false, :combine => f`
		76
		77 `updates = {}`
		78 `if output.exitstatus == 100`
		79 `  updates = parse_updates(output)`
		80 `elsif output.exitstatus == 0`
(PUP-4055) Make necessary chang... whopper authored on Sep 29	657ba78	81 `  self.debug "#{command(:cmd)} check-update exited with 0; no package updates avai`
(PUP-1362) (PUP-1775) Use yum c... adrienthebo authored on May 16, 2014	101d1de	82 `else`
(PUP-4055) Make necessary chang... whopper authored on Sep 29	657ba78	83 `  self.warn "Could not check for updates, '#{command(:cmd)} check-update' exited w`
(PUP-1362) (PUP-1775) Use yum c... adrienthebo authored on May 16, 2014	101d1de	84 `end`
		85 `updates`
		86 `end`

Code | Pull requests 3

Branch: **master**

Switch branches/tags

Find a tag...

Branches | Tags

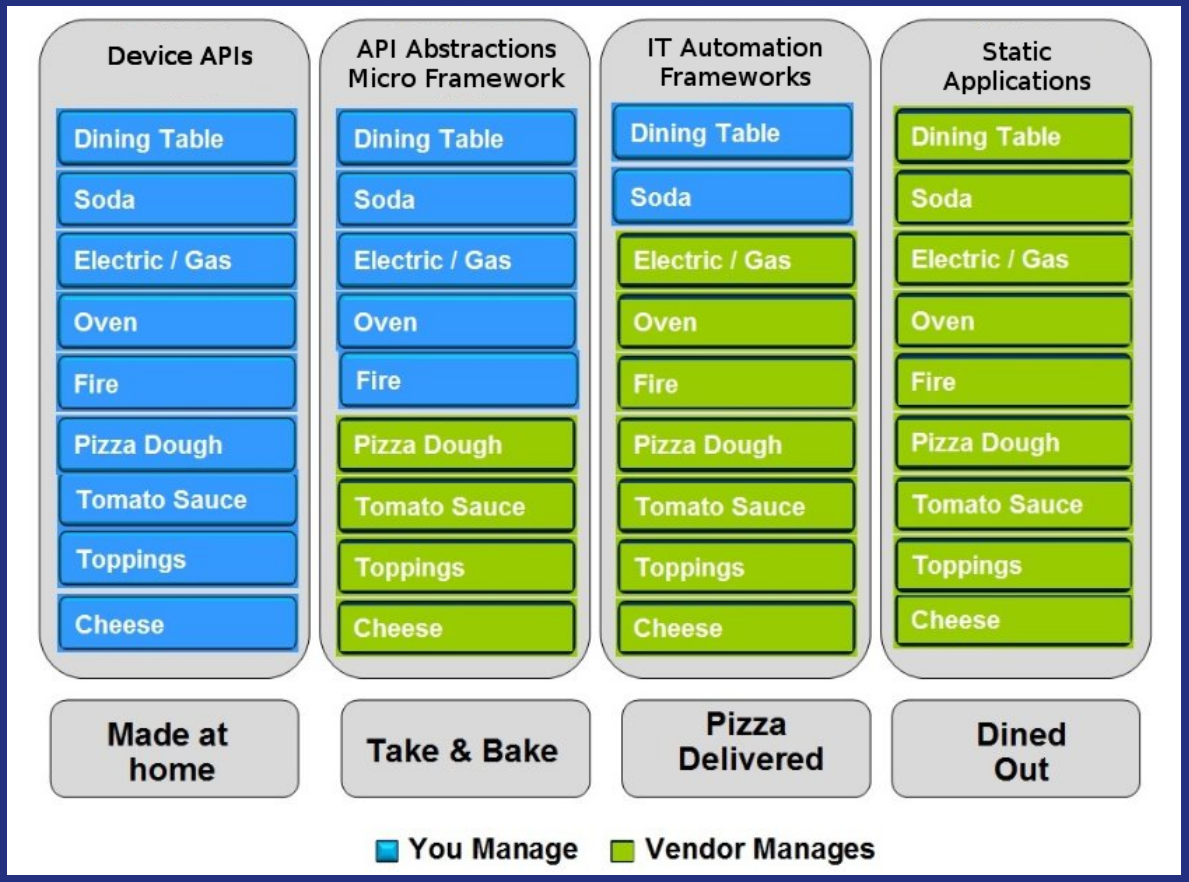
0.4.1

0.4.0

0.3.19

0.3.18

Network Automation as Pizza



Rise of the API

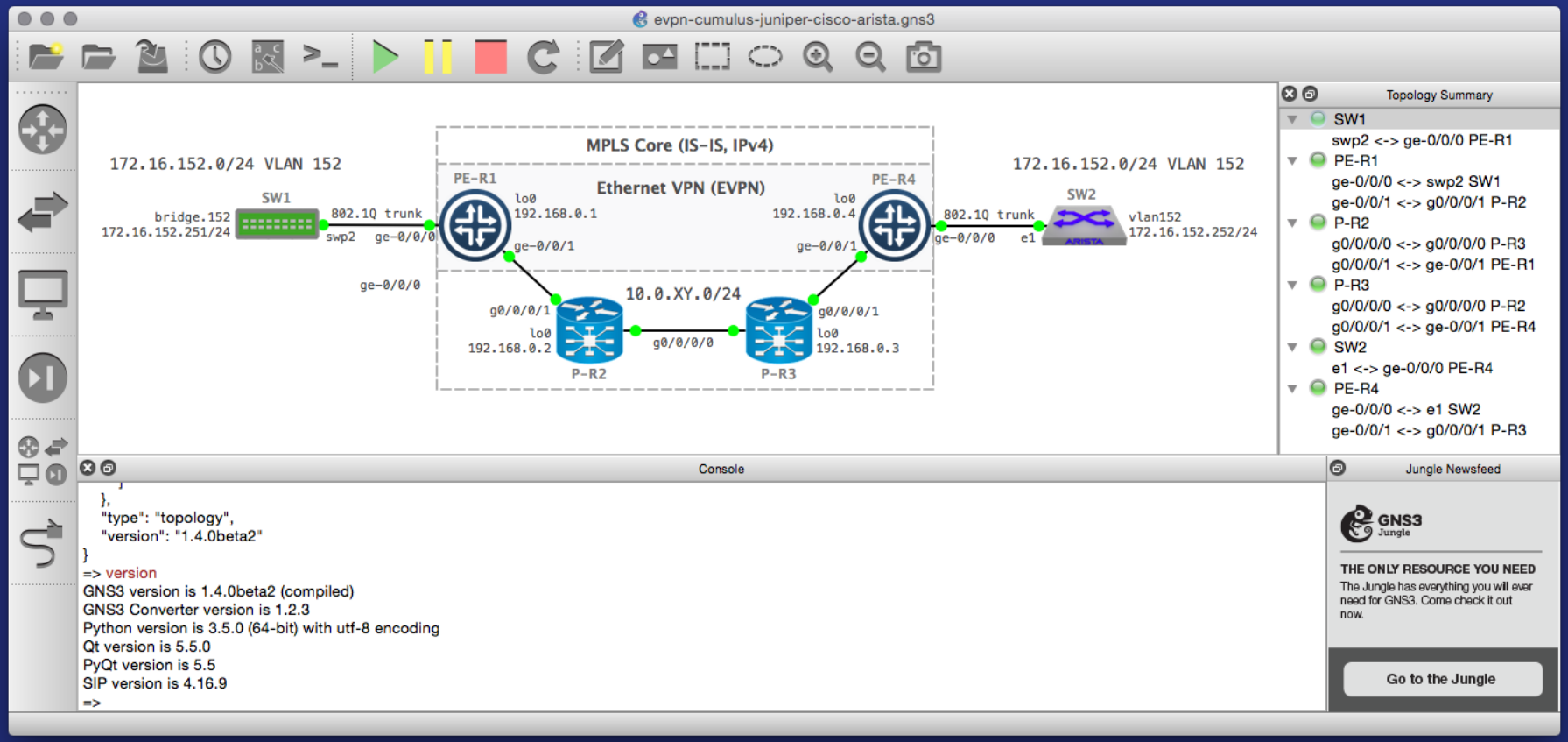
Vendors are opening up their platforms with a variety of API's and abstraction layers (highlights in no particular order)

- Cisco
  - NX-API, onePK
  - Python API
- Juniper
  - Python PyEZ
  - JET
- Arista
  - eAPI Python Library

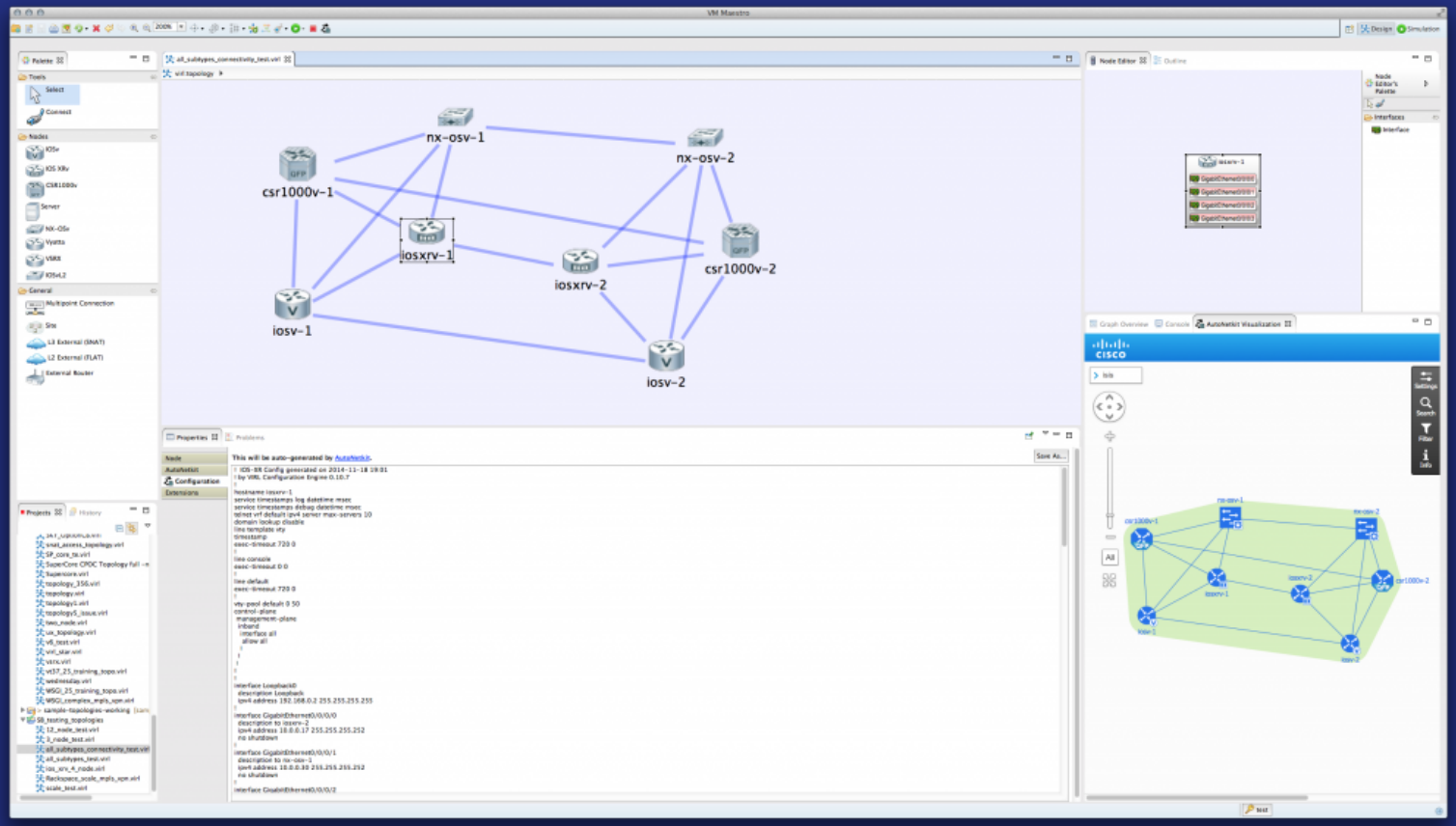
IT Automation Frameworks

Ruby	Python
• Agent Based (some agentless support) • Puppet DSL • Network Devices - Officially Supported • Large community • Mature commercial offering	• Agentless • YAML + Jinja2 Filters • Network Devices - Vendor/Community Supported • Growing community • Basic commercial offering
• Agent Based • Ruby DSL • Network Devices - Officially Supported • Large community • Mature commercial offering	• Agent Based (some agentless support) • YAML / Jinja • Minimal Network • Small community • Basic commercial offering

GNS3


evpn-cumulus-juniper-cisco-arista.gns3
MPLS Core (IS-IS, IPv4)
Ethernet VPN (EVPN)
172.16.152.0/24 VLAN 152
172.16.152.0/24 VLAN 152
SW1
bridge.152
172.16.152.251/24
802.1Q trunk
swp2
ge-0/0/0
PE-R1
lo0
192.168.0.1
ge-0/0/1
ge-0/0/0
10.0.XY.0/24
g0/0/0/1
lo0
192.168.0.2
P-R2
g0/0/0/0
g0/0/0/1
lo0
192.168.0.3
P-R3
PE-R4
lo0
192.168.0.4
ge-0/0/1
802.1Q trunk
ge-0/0/0
e1
SW2
vlan152
172.16.152.252/24
Topology Summary
SW1
swp2 <-> ge-0/0/0 PE-R1
PE-R1
ge-0/0/0 <-> swp2 SW1
ge-0/0/1 <-> g0/0/0/1 P-R2
P-R2
g0/0/0/0 <-> g0/0/0/0 P-R3
g0/0/0/1 <-> ge-0/0/1 PE-R1
P-R3
g0/0/0/0 <-> g0/0/0/0 P-R2
g0/0/0/1 <-> ge-0/0/1 PE-R4
SW2
e1 <-> ge-0/0/0 PE-R4
PE-R4
ge-0/0/0 <-> e1 SW2
ge-0/0/1 <-> g0/0/0/1 P-R3
Console
},
"type": "topology",
"version": "1.4.0beta2"
}
=> version
GNS3 version is 1.4.0beta2 (compiled)
GNS3 Converter version is 1.2.3
Python version is 3.5.0 (64-bit) with utf-8 encoding
Qt version is 5.5.0
PyQt version is 5.5
SIP version is 4.16.9
=>
Jungle Newsfeed
GNS3
Jungle
THE ONLY RESOURCE YOU NEED
The Jungle has everything you will ever need for GNS3. Come check it out now.
Go to the Jungle

Cisco - VIRL


nx-osv-1
nx-osv-2
csr1000v-1
iosxrv-1
iosxrv-2
csr1000v-2
iosv-1
iosv-2

The Unicorns

Google

Microsoft

Cross-Vendor Standards and the Future of Network Automation

NETCONF

NETCONF - IETF network management standard

- XML based encoding
  - Vendor specific data models and implementation
- Configuration RPCs
  - get-config, edit-config, copy-config, delete-config, lock, unlock
- Operational state RPCs
  - Generally map to CLI “show” commands
- Transport: SSH, HTTPS, TLS, BEEP

YANG

YANG - IETF Data Modeling Language for Netconf

- Human-readable representation of data
- Hierarchical data node representation
- Built-in data types
- Constraints can be placed on the data
- Extensible

Data is still vendor (or group) specific

WHERE TO BEGIN?

HOW CAN I HELP?

THANK YOU!

shermdog@puppetlabs.com
@shermdog01