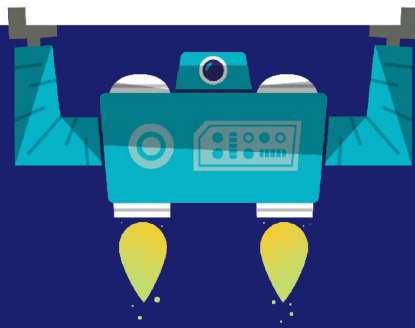


# AUTOMATION AT THE NETWORK LAYER

Rick “Shermdog” Sherman and Scott Garman  
Puppet Labs



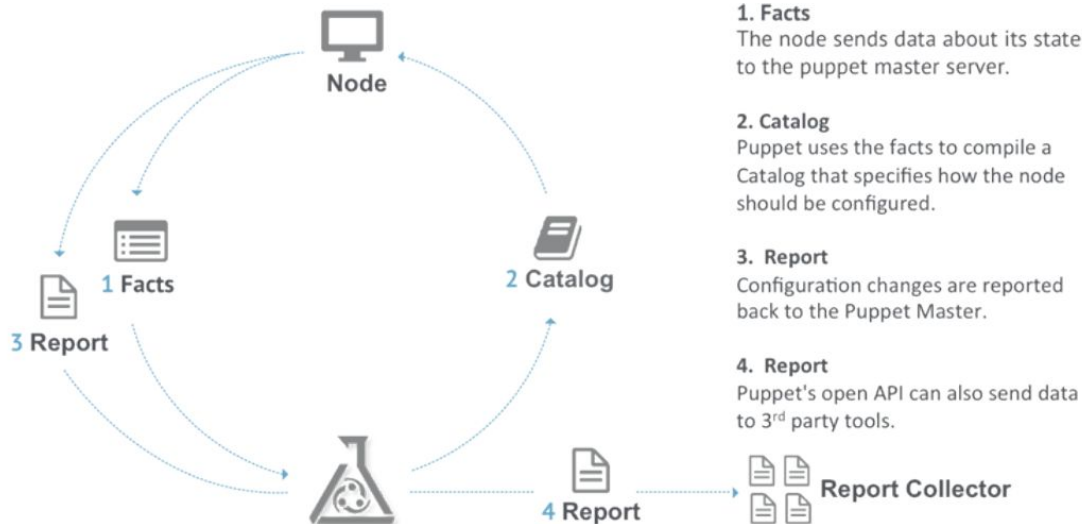
# Brief Intro to Puppet

- Puppet is a configuration management and application orchestration tool
- Automate IT deployments and increase velocity, consistency, reliability, and stability
- Puppet language is declarative - you describe the desired state of your servers and applications (vs. writing procedural scripts)
- Leverage inheritance and node classification to organize your infrastructure
- Available in Open Source and Puppet Enterprise editions
- Over 3,700 modules available on the PuppetForge ([forge.puppetlabs.com](https://forge.puppetlabs.com))

# Brief Intro to Puppet (cont)

## Lifecycle of a Puppet Agent Run

### Data Flow Between Puppet Components



# Try Our Learning VM

*<https://learn.puppetlabs.com>*

- Downloadable Virtual Machine with hands-on learning “quests” - an ideal learning environment for first-time Puppet users
- Learning site also has many resources to help you level-up your Puppet proficiency - including self-paced online courses, instructor-led training, and details on Puppet certification

# Life of a Network Engineer

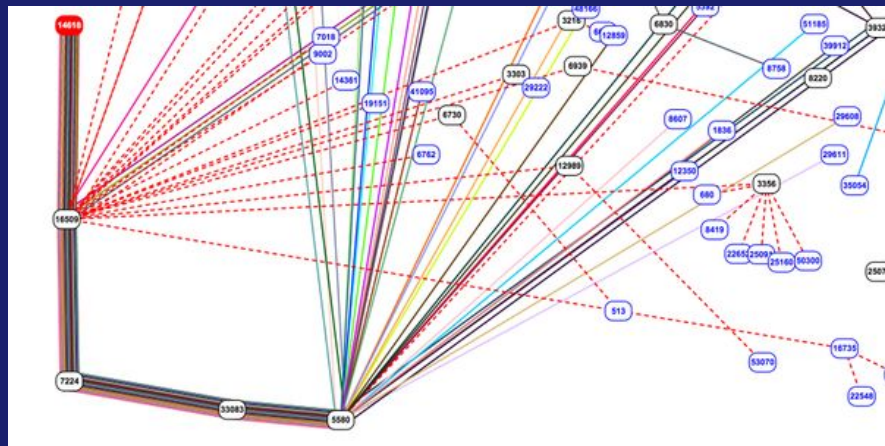
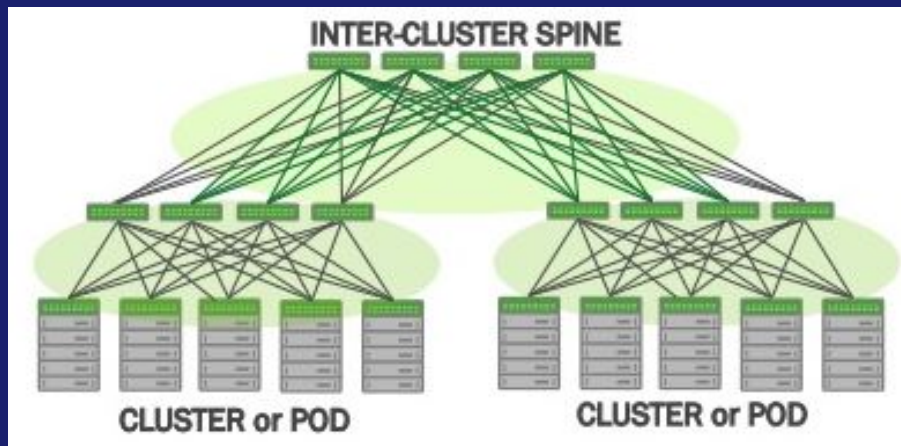
Let's make some generalizations (what could go wrong?)

- 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 CAN I DO?

# An Overview of Supported Network Devices

# Disclaimer

*The enclosed material represents Puppet Labs' current view of its product development cycle and future directions. It is intended for informational purposes only, and should not be interpreted as a commitment on the part of Puppet Labs. Puppet Labs makes no warranties, express or implied, in this document.*

# Arista Networks

## Current Support:

- 31 Network Switch Models
- Runs on Arista EOS 4.13+
- Compatible with Puppet Enterprise 3.7.2+
- Puppet Supported

## Puppet Modules:

- [aristanetworks/eos](#)
- [aristanetworks/netdev\\_stdlib\\_eos](#)

# Cisco

## Current Support:

- 17 Network Switch Models
- Nexus 9k and 3k
- Runs on Cisco NX-OS 7.0(3)I2(1) - released Sept 4, 2015
- Compatible with Puppet Enterprise 2015.2.0+
- Puppet Supported

## Puppet Modules:

- [puppetlabs/ciscopuppet](#)
- [github.com/datacenter/puppet-aci](#) (Cisco ACI Fabric Configuration, unsupported)

# Cisco

## Upcoming Support - Q1 2016:

- Nexus 5k/6k/7k Series Switches
- Puppet Enterprise with these switches will be a **Cisco Compatible** solution
- Cisco's Puppet web portal: [www.cisco.com/go/puppetlabs](http://www.cisco.com/go/puppetlabs)



Compatible

# Cumulus Networks

## Current Support:

- 13 Network Switch Models
- All x86 based switches on Cumulus' Hardware Compatibility List
- Runs on Cumulus Linux 2.2+
- Compatible with Puppet Enterprise 3.7.2+
- Puppet Supported Program

## Puppet Modules:

- `cumuluslinux/cumulus_interfaces`
- `cumuluslinux/cumulus_interface_policy`
- `cumuluslinux/cumulus_ports`
- `cumuluslinux/cumulus_license`

# F5 Networks

## Current Support:

- 6 BIG-IP Load Balancer Model Families
- Runs on F5 OS 11.6+
- Compatible with Puppet Enterprise 3.7.0+
- Uses Puppet Device proxy model (no agent required)
- Puppet Supported

## Puppet Modules:

- [puppetlabs/f5](#)



# Huawei

## Upcoming Support - Q1 2016

- CloudEngine 12800 Series Switches
- Will run on HuaweiOS 6.0+
- Will be compatible with Puppet Enterprise
- Puppet Agent will be OEM-installed, no need for initial deployment
- Puppet Supported

# OVERVIEW OF NETWORKING TYPES

# Netdev Stdlib Types

- domain\_name
- name\_server
- search\_domain
- network\_dns
- ntp\_server
- ntp\_config
- syslog\_server
- syslog\_settings
- network\_interface
- network\_vlan
- network\_trunk
- port\_channel
- continued...

# Netdev Stdlib Types

- `tacacs`
- `tacacs_global`
- `tacacs_server`
- `tacacs_server_group`
- `radius`
- `radius_global`
- `radius_server`
- `radius_server_group`
- `snmp_community`
- `snmp_user`
- `snmp_notification`
- `snmp_notification_receiver`

# Arista Types

- eos\_acl\_entry
- eos\_bgp\_config
- eos\_bgp\_neighbor
- eos\_bgp\_network
- eos\_command
- eos\_ethernet
- eos\_interface
- eos\_ipinterface
- eos\_mlag
- eos\_mlag\_interface
- eos\_ntp\_config
- eos\_ntp\_server
- eos\_portchannel
- eos\_snmp
- eos\_staticroute
- eos\_stp\_interface
- eos\_switchport
- eos\_system
- eos\_vlan
- eos\_vxlan
- eos\_vxlan\_vlan
- eos\_vxlan\_vtep

# Cisco NX-OS Types

- `cisco_bgp`
- `cisco_bgp_af`
- `cisco_bgp_neighbor`
- `cisco_bgp_neighbor_af`
- `cisco_command_config`
- `cisco_interface`
- `cisco_interface_ospf`
- `cisco_ospf`
- `cisco_ospf_vrf`
- `cisco_snmp_community`
- `cisco_snmp_group`
- `cisco_snmp_server`
- `cisco_snmp_user`
- `cisco_tacacs_server`
- `cisco_tacacs_server_host`
- `cisco_vlan`
- `cisco_vrf`
- `cisco_vtp`

# Cumulus Types

- cumulus\_ports
- cumulus\_interfaces
  - cumulus\_interface
  - cumulus\_bond
  - cumulus\_bridge
- cumulus\_interface\_policy
- cumulus\_license
- Debian based - stdlib types and forge

# WHERE TO BEGIN?



# Arista: NTP

```
class profile::network::base {
  require eos::install
  include profile::network::base::ntp

  service { ['puppet']:
    ensure => running,
    enable => true,
  }
}
```

```
class profile::network::base::ntp {
  eos_ntp_config { ['settings']:
    source_interface => 'Management1',
  }

  eos_ntp_server { ['10.14.99.10']:
    ensure => present,
  }

  eos_ntp_server { ['10.27.88.10']:
    ensure => present,
  }
}
```

# Cisco NX-OS: NTP

```
class profile::network::base {
  require ciscopuppet::install
  include profile::network::base::ntp

  service { ['puppet']:
    ensure => running,
    enable => true,
  }
}
```

```
class profile::network::base::ntp {
  cisco_command_config { 'corp-ntp':
    command => "
      ntp server 10.14.99.10 prefer
      ntp server 10.27.88.10
      ntp master 1
      ntp source-interface vlan99",
  }
}
```

# Cumulus: NTP

```
class profile::network::base {
  require ntp::install
  include ::ntp

  service { ['puppet']:
    ensure => running,
    enable => true,
  }
}
```

```
class { '::ntp':
  servers => [
    '10.14.99.10',
    '10.27.88.10'],
  interfaces => ['10.14.99.1'],
}
```

# Arista: VLAN Trunk

```
class profile::network::trunks {  
  eos_switchport { 'Eth48':  
    ensure           => present,  
    mode             => 'trunk',  
    trunk_allowed_vlans => ['60-62', '80'],  
    trunk_native_vlan => '20',  
  }  
}
```

# Cisco: VLAN Trunk

```
class profile::network::trunks {  
  cisco_interface { 'Ethernet1/48':  
    ensure           => present,  
    switchport_mode  => 'trunk',  
    switchport_trunk_allowed_vlan => ['60-62', '80'],  
    switchport_trunk_native_vlan  => '20',  
  }  
}
```

# Cumulus: VLAN Trunk

```
class profile::network::trunks {  
  cumulus_interface { 'swp48':  
    ensure => present,  
    mode   => 'trunk',  
    vids   => ['60-62', '80'],  
    pvid   => '20',  
  }  
}
```

# Ready to Get Started?

## Next Steps:

1. Check to make sure your networking device is supported
2. Download and install puppet agent for your device (if required)
3. Install the relevant module(s) for your device
4. Begin with a simple task, such as managing NTP or syslog, and then begin moving onto features such as managing network ports

# 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

HOW CAN I HELP?

# THANK YOU!

[shermdog@puppetlabs.com](mailto:shermdog@puppetlabs.com), [@shermdog01](https://twitter.com/shermdog01)  
[scott.garman@puppetlabs.com](mailto:scott.garman@puppetlabs.com)