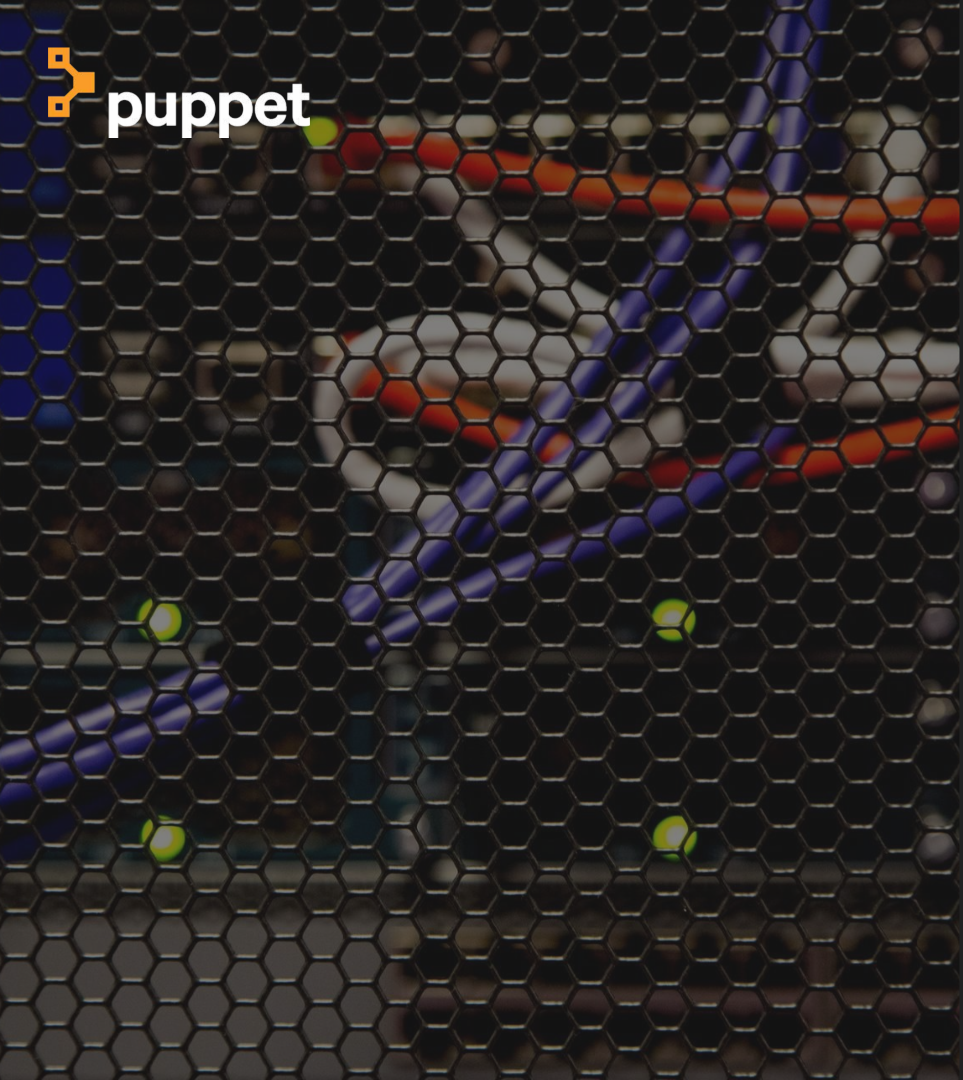
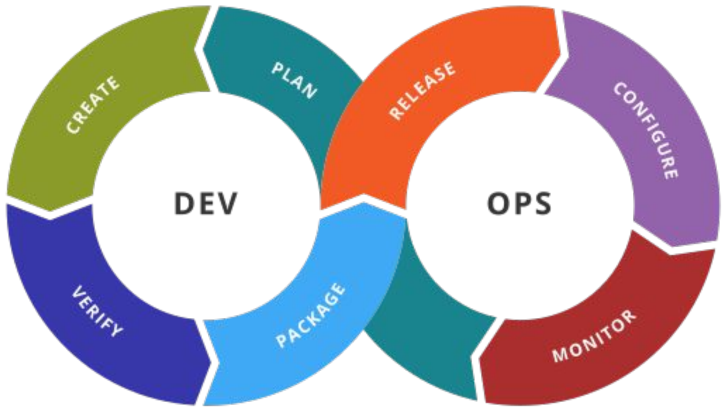


Network Operations as Code

September 2018

DevOps



Aim?	To reduce the time, cost and errors of software deployment, while maintaining compliance
Goal?	Of being more agile and responsive to business needs
How?	Defining state in code with configuration management and use automation to maintain state

NetOps

- NetOps has a similar aim to DevOps
  - i.e. being more responsive to business needs
- However, in networking, stability is critical
  - contradiction with the desire for agility
- For NetOps to be successful it must enable network management personnel to increase agility while ensuring compliance and reducing risk

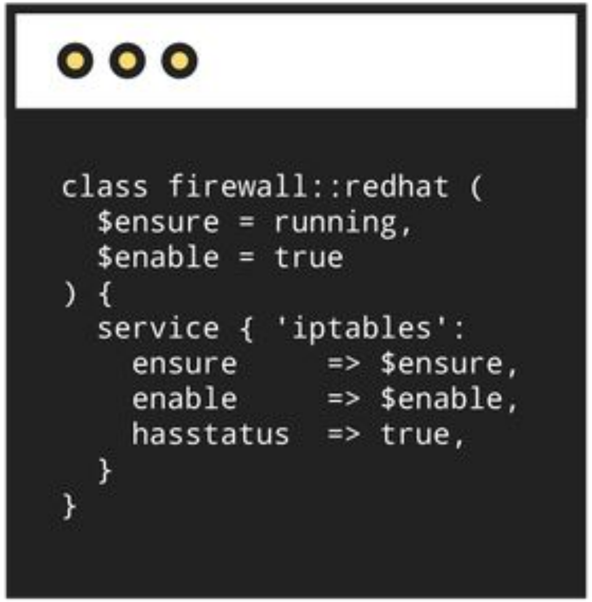
puppet

Puppet Overview

Who we are and what we do

Using a common language

Get a standard way to deliver & operate all of your software



```
class firewall::redhat (
  $ensure = running,
  $enable = true
) {
  service { 'iptables':
    ensure     => $ensure,
    enable     => $enable,
    hasstatus  => true,
  }
}
```

- Define once with an easy-to-understand language
- Improve collaboration by unifying processes and tooling
- Get started quickly by choosing from existing modules, or create your own
- Open-source provides scale for building out content

puppet

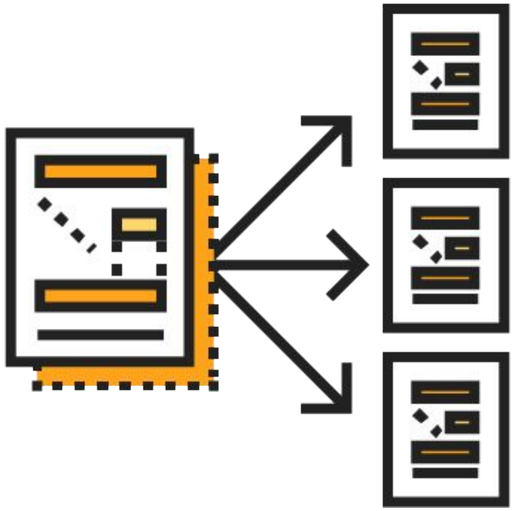
Infrastructure as Code

Describe the ideal environment with a simple, commonly understood language

```
building { 'home':
  ensure     => 'clean',
  front_door => 'closed',
  keys       => 'key_hook',
  jacket     => 'closet',
  floor      => 'vacuumed’,
  litter_box => 'empty',
  remote     => 'coffee_table',
}
```

Control & enforce consistency across your devices

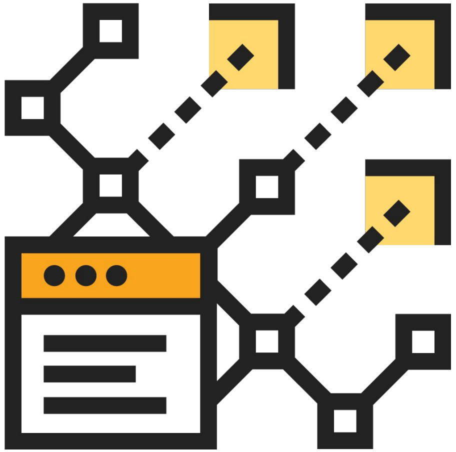
Make changes with confidence & deliver faster



- Orchestrate changes to infrastructure on-demand or on-schedule
- Simulate changes using no-op
- Continually enforce desired configurations
- Automatically remediate misconfigurations & unexpected changes
- Run ordered deployments based on dependencies you define

Simulation and no-op

Only change what you need to when you need to



- Puppet is idempotent
  - Config is only updated when it doesn't match the catalogue
- Simulation is possible and strongly advised
  - no-op: this is what will change if you run this command for real

Puppet Resources

```
package { 'openssh-server':
  ensure => installed,
}

file { '/etc/ssh/sshd_config':
  source  => 'puppet:///modules/sshd/sshd_config',
  owner   => 'root',
  group   => 'root',
  mode    => '0640',
  notify  => Service['sshd'], # sshd restarts whenever this file is changed.
  require => Package['openssh-server'],
}

service { 'sshd':
  ensure      => running,
  enable      => true,
}
```

Puppet Resources: Cisco

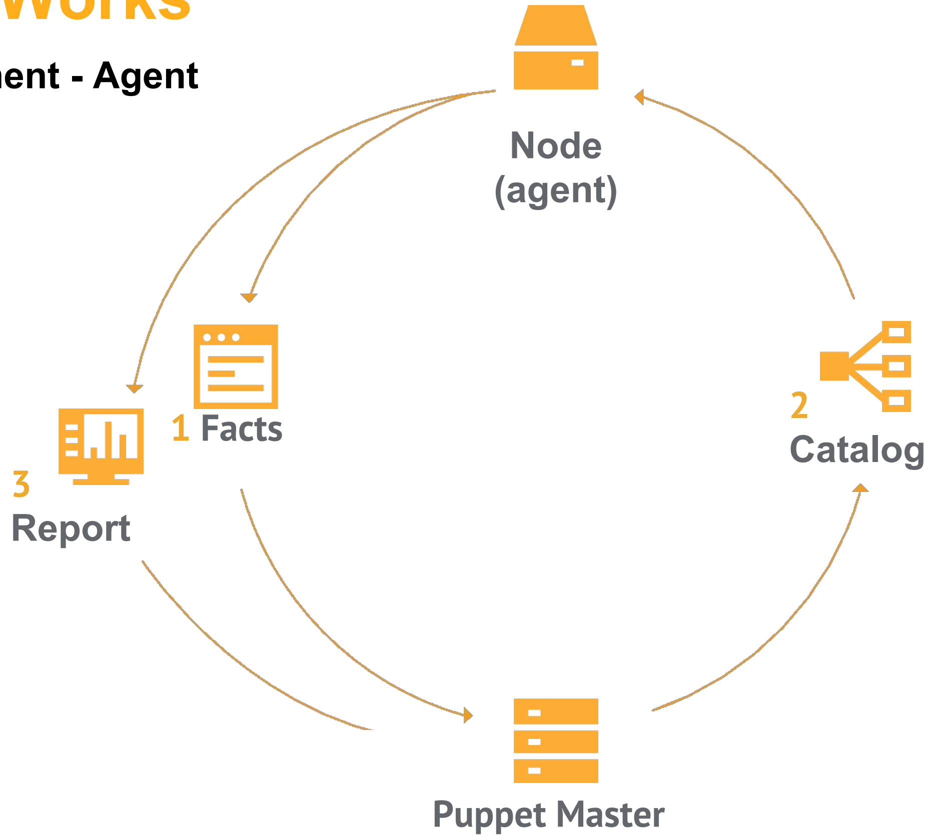
```
banner { 'default':
  motd => 'Hello, world!',
}

cisco_interface { 'ethernet1/1':
  ensure               => 'present',
  ipv4_address         => '192.168.1.1',
  ipv4_netmask_length => '24',
  mtu                  => '1600',
  shutdown             => false,
  access_vlan          => 1,
  switchport_mode      => disabled,
}

ios_config { $name:
  command           => $command,
  idempotent_regex => $regex,
}
```

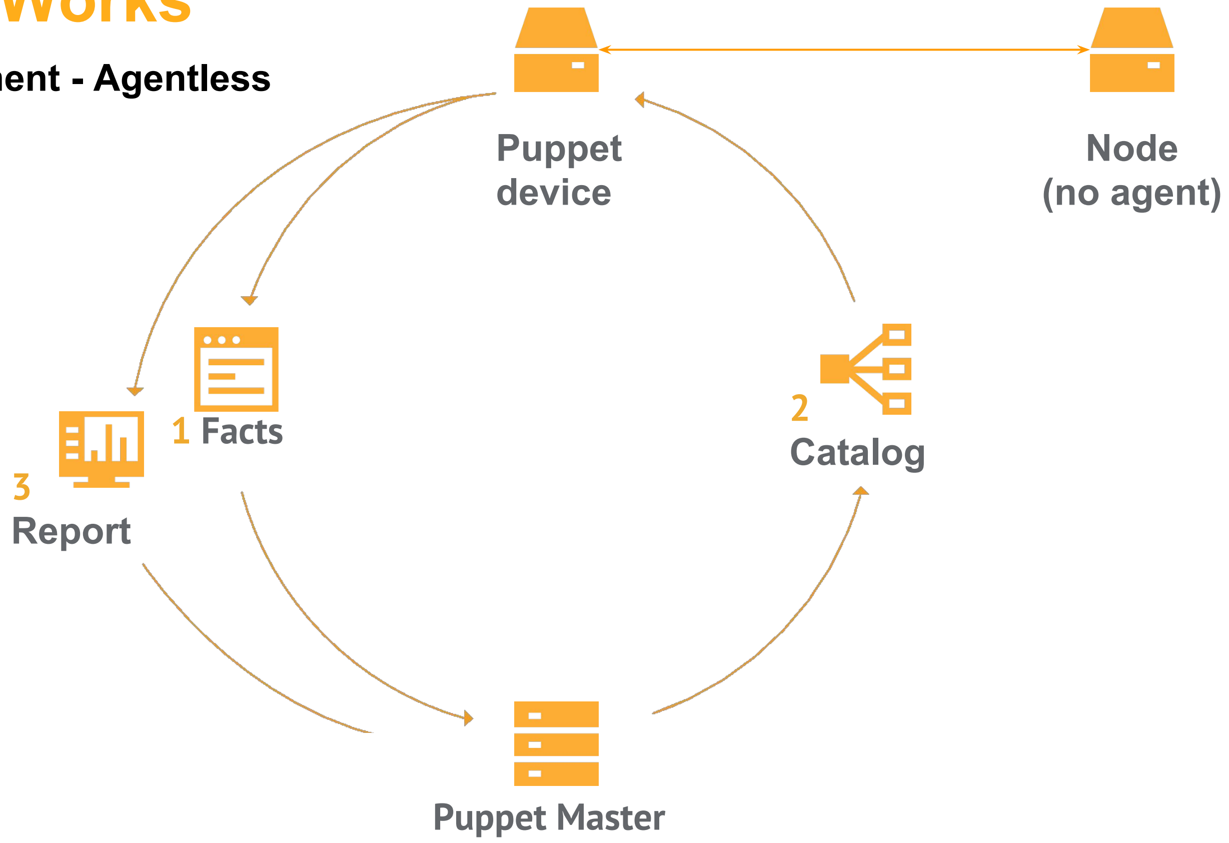
How Puppet Works

Continuous Enforcement - Agent



How Puppet Works

Continuous Enforcement - Agentless



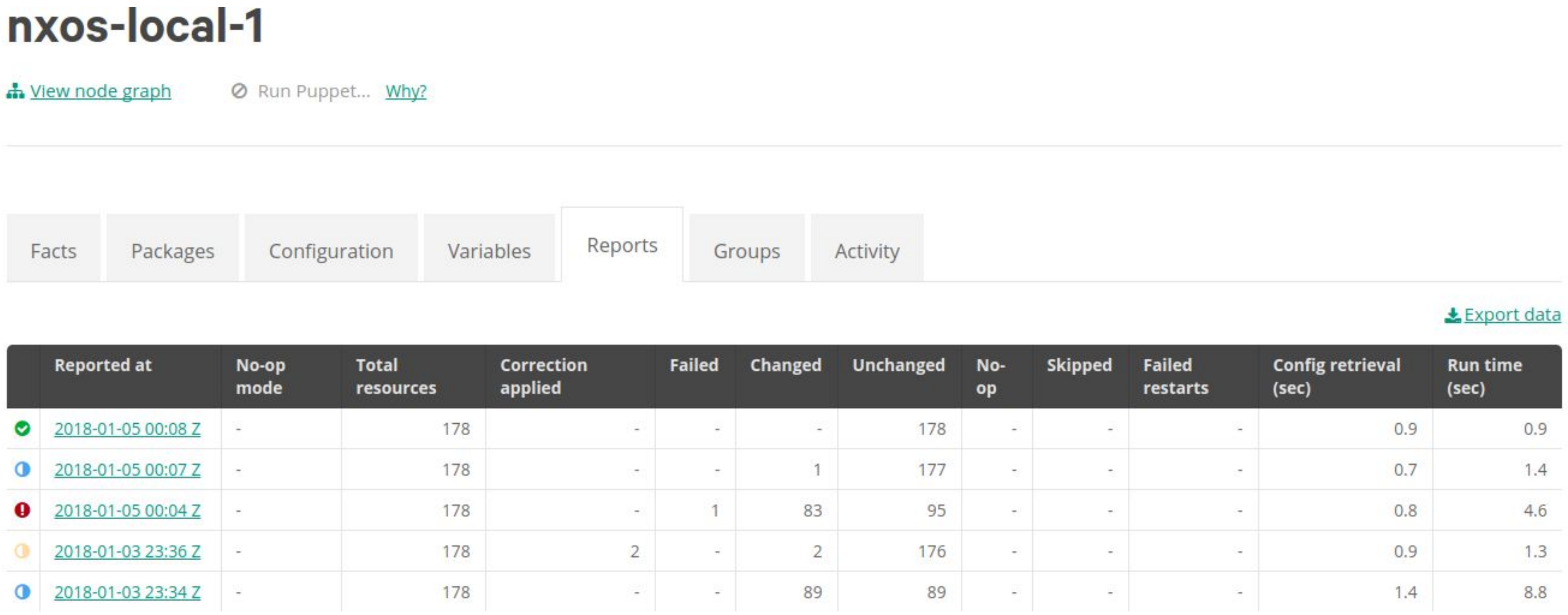
Know the types of changes

Status values indicating what happened during a Puppet run

- Failure
- Corrective change
- Intentional change
- Corrective no-op
- Intentional no-op
- Skip

Puppet Enterprise Reports

Know when changes occur and why



nxos-local-1

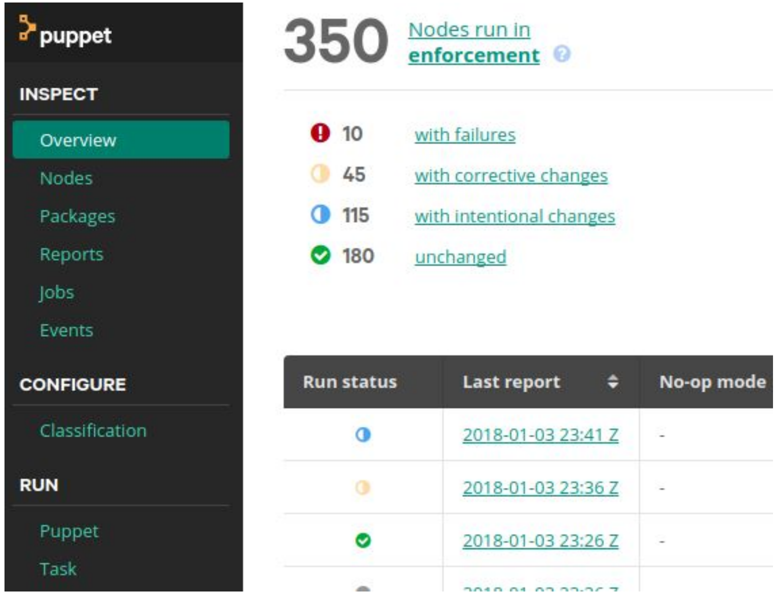
View node graph   Run Puppet... Why?

	Reported at	No-op mode	Total resources	Correction applied	Failed	Changed	Unchanged	No-op	Skipped	Failed restarts	Config retrieval (sec)	Run time (sec)
✓	2018-01-05 00:08 Z	-	178	-	-	-	178	-	-	-	0.9	0.9
◐	2018-01-05 00:07 Z	-	178	-	-	1	177	-	-	-	0.7	1.4
!	2018-01-05 00:04 Z	-	178	-	1	83	95	-	-	-	0.8	4.6
◐	2018-01-03 23:36 Z	-	178	2	-	2	176	-	-	-	0.9	1.3
◐	2018-01-03 23:34 Z	-	178	-	-	89	89	-	-	-	1.4	8.8

puppet

Know what you have

Gain situational awareness & understand exactly what's happening across your software



350 Nodes run in **enforcement**

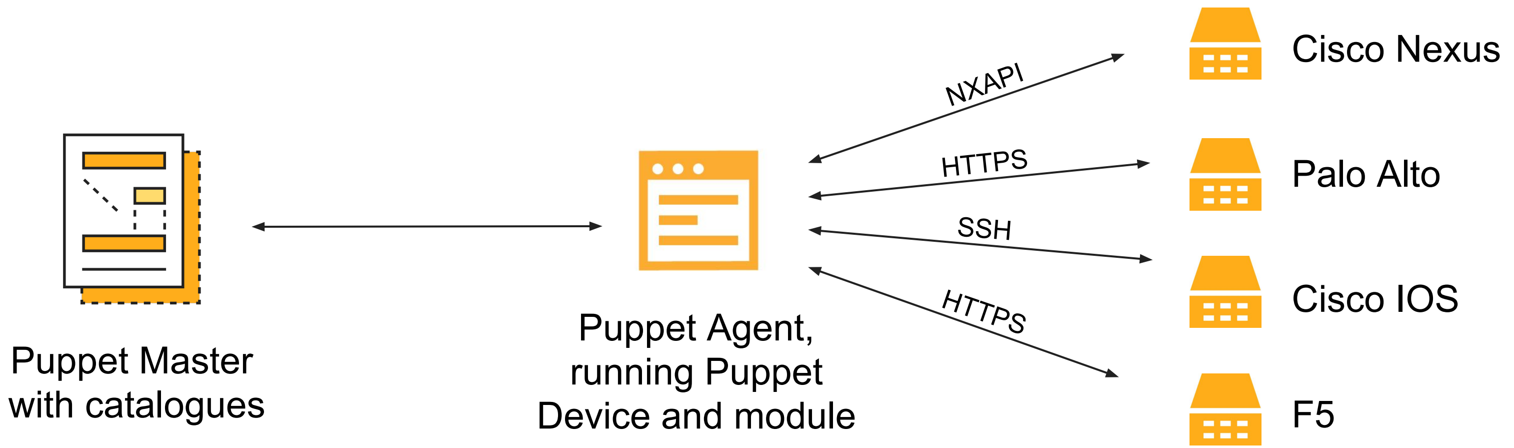
●	10	with failures
◐	45	with corrective changes
◐	115	with intentional changes
✓	180	unchanged

Run status	Last report	No-op mode
◐	2018-01-03 23:41 Z	-
◐	2018-01-03 23:36 Z	-
✓	2018-01-03 23:26 Z	-

- Monitor exactly what you have running across your data center & cloud
- View changes taking place in real-time and report on the cause of those changes
- Visualize dependencies across your infrastructure & apps to improve change success rate

Deployment Model

Perform multi-vendor device management at scale with a single language

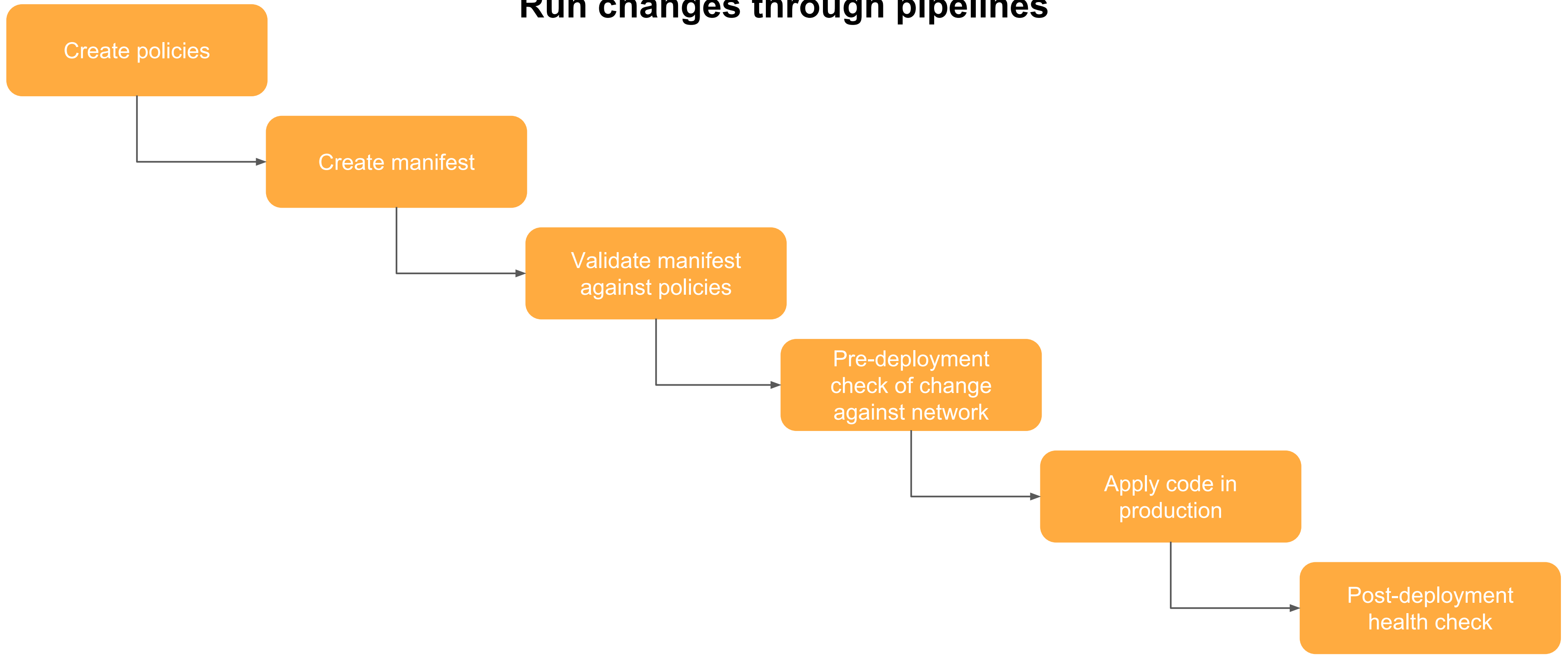


Puppet Master with catalogues
Puppet Agent, running Puppet Device and module
NXAPI
HTTPS
SSH
HTTPS
Cisco Nexus
Palo Alto
Cisco IOS
F5

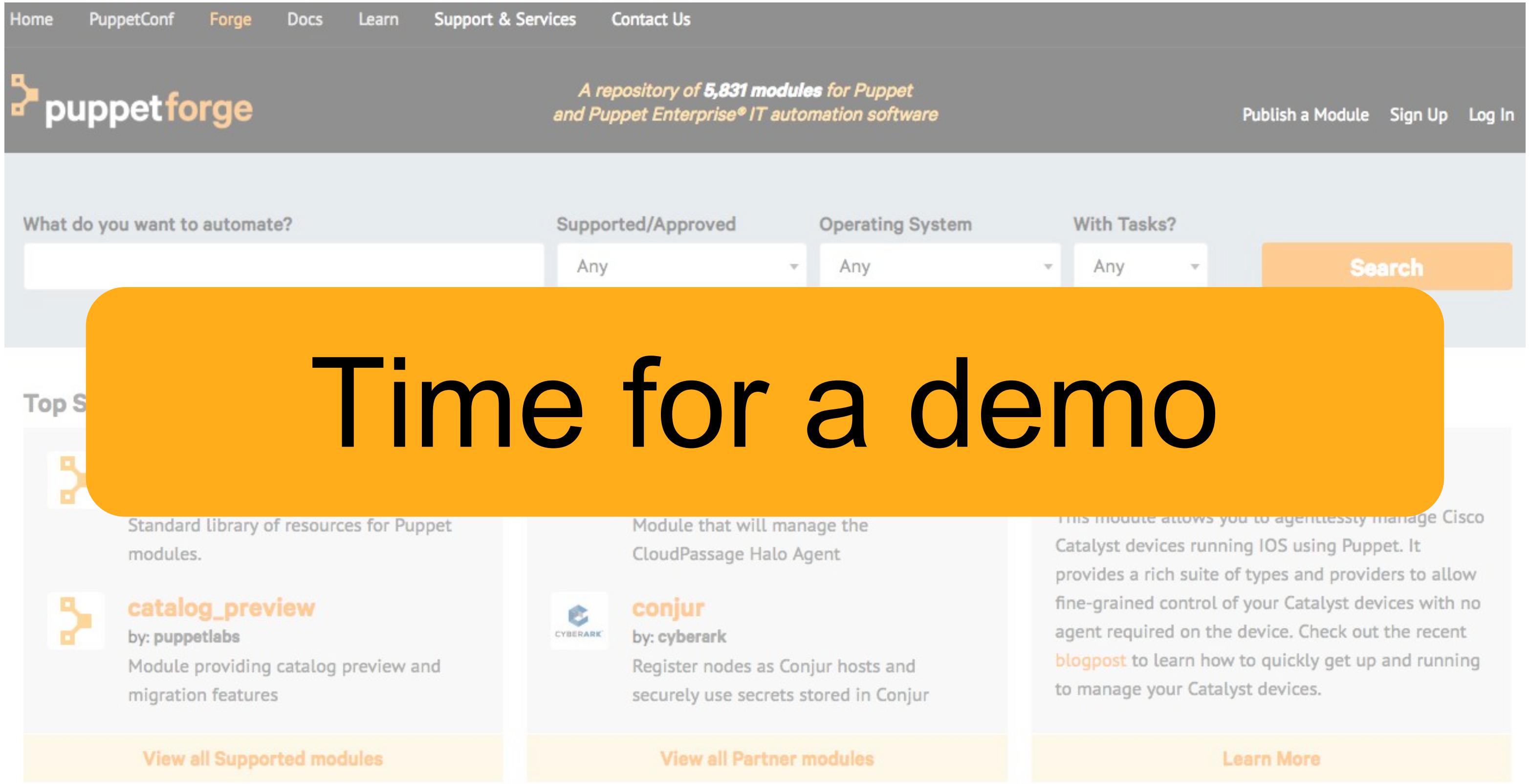
Modules from: Cisco Nexus, Palo Alto, Cisco IOS, F5 Big-IP, Cisco ACI, Cisco Meraki, Netscaler, NetApp, Huawei, Arista, Cumulus, Lenovo CNOS

Pipeline concept: the future for network automation?

Run changes through pipelines


Create policies
Create manifest
Validate manifest against policies
Pre-deployment check of change against network
Apply code in production
Post-deployment health check

puppet



Home PuppetConf Forge Docs Learn Support & Services Contact Us

puppetforge

A repository of ***5,831 modules*** *for Puppet and Puppet Enterprise® IT automation software*

Publish a Module Sign Up Log In

What do you want to automate? | Supported/Approved: Any | Operating System: Any | With Tasks?: Any | Search

Top S

Standard library of resources for Puppet modules.

catalog_preview
by: puppetlabs
Module providing catalog preview and migration features

Module that will manage the CloudPassage Halo Agent

conjur
by: cyberark
Register nodes as Conjur hosts and securely use secrets stored in Conjur

This module allows you to agentlessly manage Cisco Catalyst devices running IOS using Puppet. It provides a rich suite of types and providers to allow fine-grained control of your Catalyst devices with no agent required on the device. Check out the recent blogpost to learn how to quickly get up and running to manage your Catalyst devices.

View all Supported modules | View all Partner modules | Learn More

Time for a demo

NetOps Principles

- Automate - move away from the command line as much as possible
- Define state in code
- Manage compliance in code
- Use pipelines to run pre- and post-deployment checks
- Trust the tools
- Be open to change

Adopting NetOps

- Walk before running - take a single device type and try to automate common tasks
- Define policies and desired state in code
- Take a pipeline approach to test before deployment
- Use no-op to simulate before making the change
- Take an open-source approach
- Check out what other people are doing, like Netflix's Winston: https://bit.ly/2phEgTe

Thanks!

davin.hanlon@puppet.com

puppet

puppet