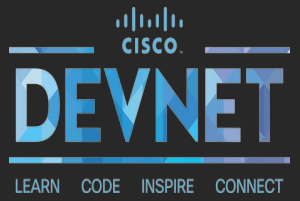
NetDevOps Style Configuration Management for the Network

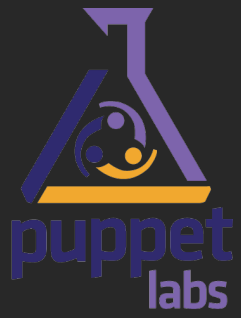
What it means for network engineers, and why we should care?

Stuart Clark Network Automation Evangelist
Cisco Devnet

Game Plan

- Virlutils Network Simulation
- What are Infrastructure as Code and Configuration Management?
- Benefits of Configuration Management
- Recipes, Manifests, Playbooks, and the Tools
- Configuration Management with Ansible Example





“virl up” Network Simulation Done Right

- virlutils makes it easy to start a network simulation
- virlutils is written in Python
- https://github.com/CiscoDevNet/virlutils

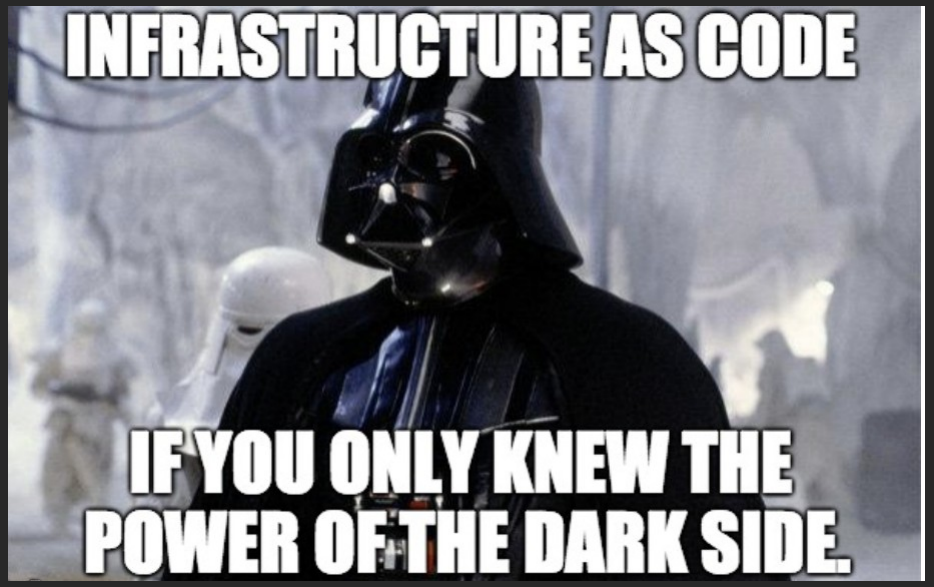
Why Ansible for the Network?

- Agentless
- Currently popular in network community
  - ie Lots of examples!
- Written in Python
- Simple to install and get started!

What are Infrastructure as Code and Configuration Management?

Infrastructure as Code... huh?

"**Infrastructure as Code** (IaC) is the process of managing and provisioning computer data centers through machine-readable definition files..."



https://en.wikipedia.org/wiki/Infrastructure_as_Code

Some Principals of ”Network as Code”

- Store network configuration in source control systems (ie git)
  - Use ”machine readable” formats like YAML, JSON, XML
- Treat the source control as single source of truth
  - Develop, test, and deploy to prod from same source
- Deploy configuration using programmatic APIs and tooling
  - Limit manual network configuration
  - Explore “Configuration Management” tooling.

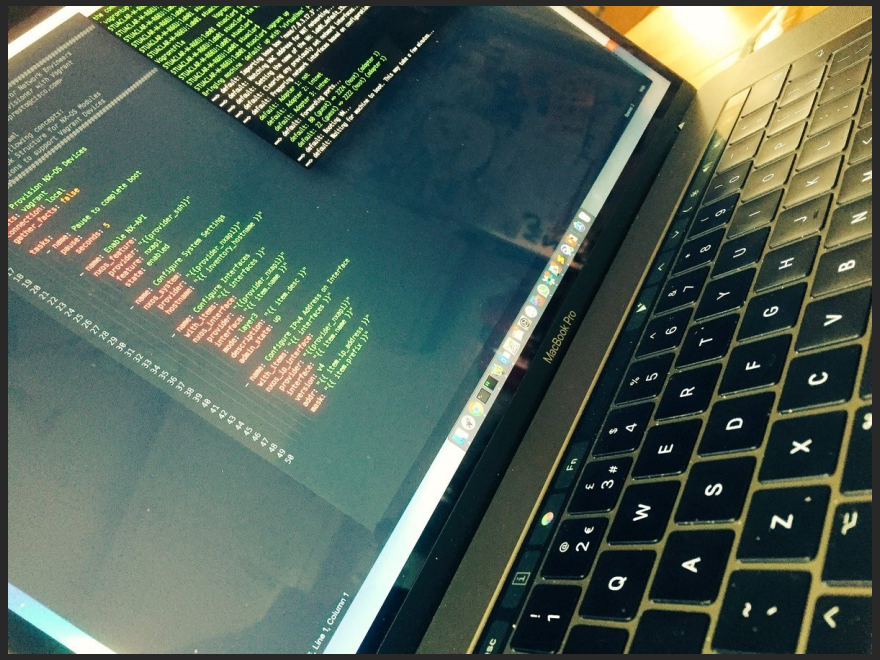
Mechanism = Automation

No more hand to hand combat configuration management

Configuration Management today is about the “tools”

Characteristics = Desired State

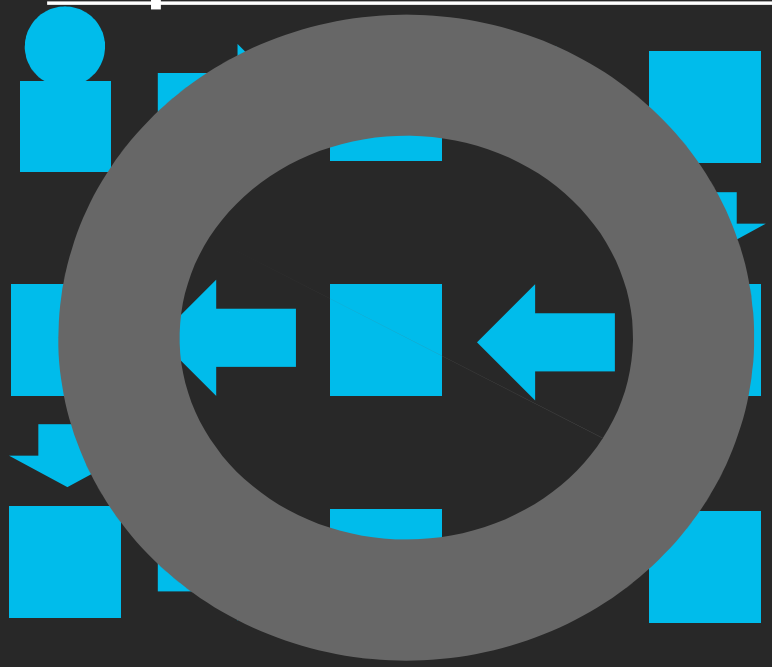
- The software and version installed
- System attributes like name, address, ownership, etc
- Feature specific configurations



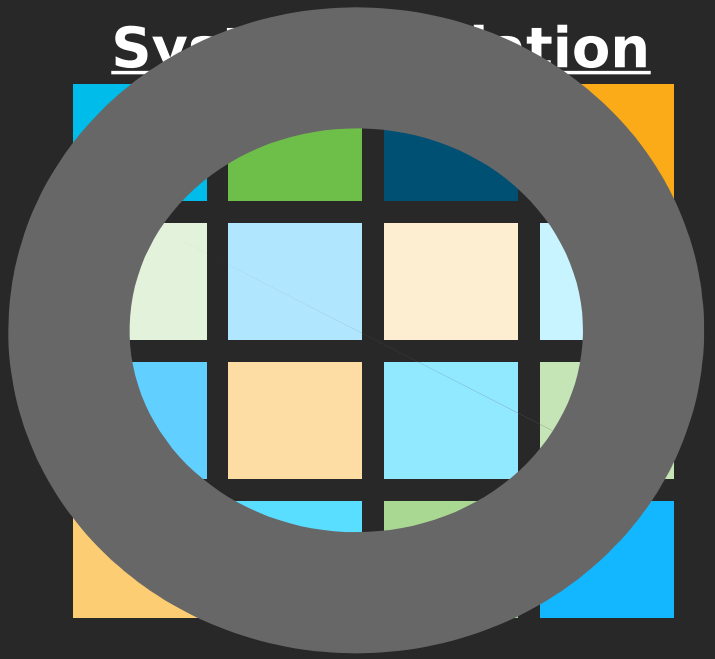
Benefits of Configuration Management

Quickly Provision Infrastructure

Sequential and Manual



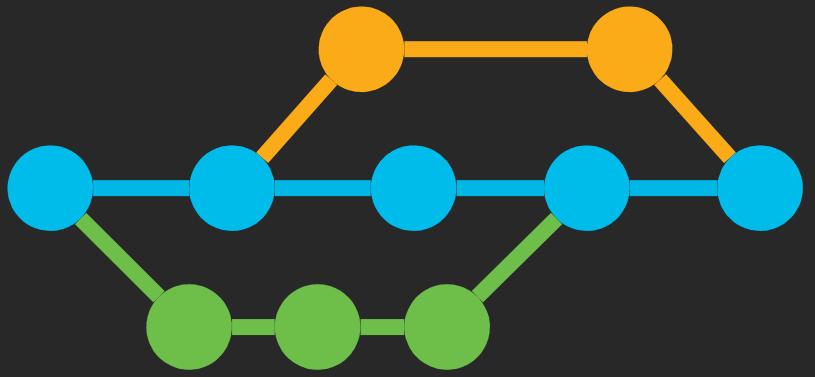
No More Snowflakes



Orga ion

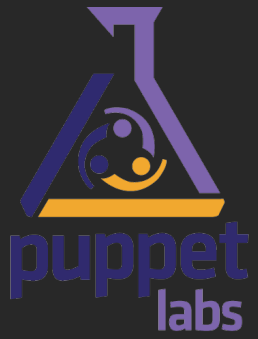
Version Controlled Infrastructure

Infrastructure as Code



Commonalities of Configuration Management Tools

- Open Source Foundation
- Automation and Orchestration
- Idempotent Behavior
- Facts, lots of facts
- Modules and Libraries



Matrix of Common Info and Terms

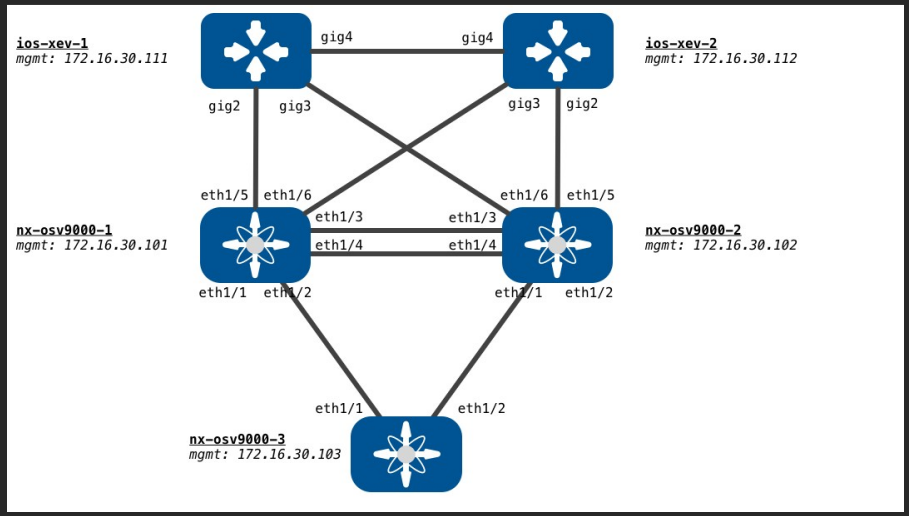
	Ansible	**Puppet**	**Chef**	**SaltStack**
Language	Python + YAML	Ruby Based	Ruby	Python
Managed Node Requirements	Agentless	Traditionally Agent Based	Agent Based	Agent Based “minions”
Centralized Management	Any computer can be “controller” *Optional “Tower”*	Puppet Master	Chef Server	Salt Master
What you create	Playbook / Roles	Manifest / Module	Recipe / Cookbook	Pillar / Include

”Network as Code” with Ansible for Configuration Management

- Ansible Playbook
  - Run roles against relevant groups
- Ansible Roles
  - Align to network roles
- Inventory File
  - List network devices
  - Logically group for configuration
- Variable Files
  - Device specific details
  - General group details

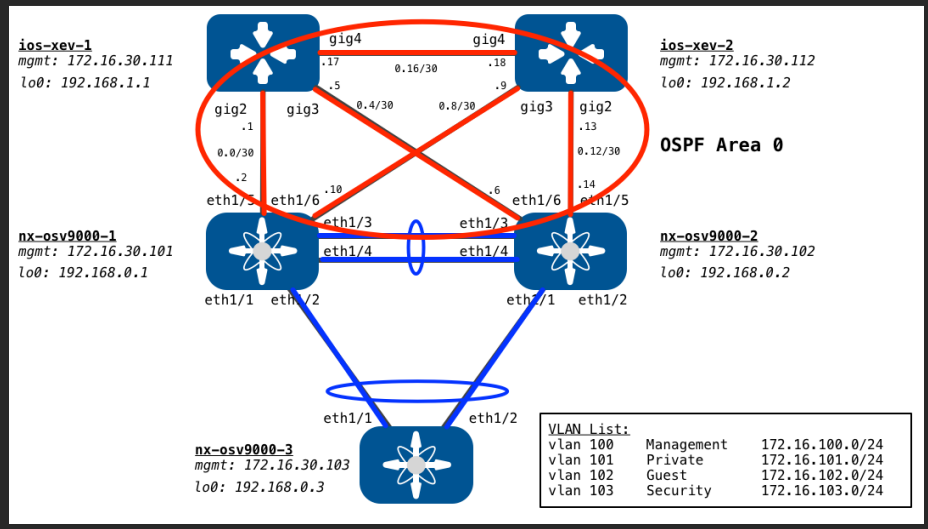
Starting Network Topology

- Physical Topology
  - “Core” – IOS XE Routers
  - “Distribution” – NX-OS Switches
  - “Access” – NX-OS Switches
- Network has been cabled already
- Management access to devices enabled
  - No other configuration completed



Desired Network Configuration

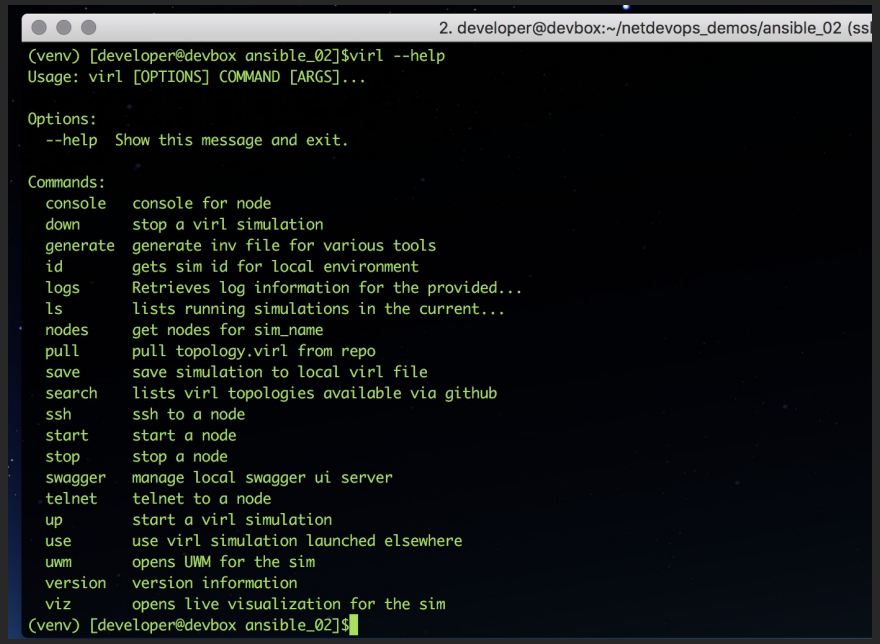
- Layer 3 Links between Core/Dist
  - OSPF Area 0 Routing Configured
- Distribution configured for VPC Domain
- Layer 2 port-channel trunk to access
- Set of VLANs Configured
  - SVIs at Distribution with HSRP Configured



Running the code

- Bring up the topology – ***virl up [virlfiles/name]***
  - Not specifying a virlfile name will import the default one from the current directory
- See the simulations are currently running on your server ***virl ls --all***
- Check the state of the nodes - ***virl nodes***
  - This will display the nodes, name, ip address and state
- Once the nodes are available - ***virl generate ansible***
  - Creates the default inventory file

Use ***virl --help*** to see the variety of commands that are available



Advantages of VIRL as the development environment

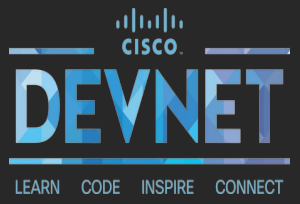
- Robust support for large topologies mimicking production
- Simulations can include servers and applications in addition to network
- Full data plane within the simulation to test traffic flows and protocol behaviour
- Off-load simulation to remote server

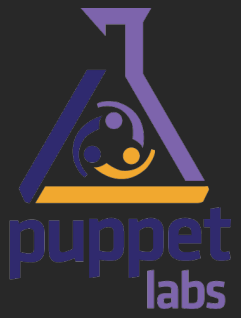
But there are definite caveats to consider as well.

- For typical uses, not a fully local dev environment
- Significant time to instantiate networks
- Not an insignificant resource requirement for large topologies

What did we cover?

- What are Infrastructure as Code and Configuration Management?
- Benefits of Configuration Management
- Recipes, Manifests, Playbooks, Oh My! The Tools
- Configuration Management with Ansible Example





What do do next?

Resources

Network Automation with Ansible (ansible.com)
Network Automation wtih Salt (saltstack.com)

Network Automation with Puppet (puppet.com)
Network Automation with CFEngine (cfengine.com)

DevNet Learning Labs

- Introduction to Ansible
- Using Ansible with NX-OS Devices
- Introduction to ACI and Ansible
- Home Lab: Using Ansible from your Desktop OS

Blogs and Videos

- What does "Network as Code" Mean?
- Automating Cisco ACI with Ansible Eliminates Repetitive Day-to-Day Tasks
- NetDevOps Video Module

Got more questions? Stay in touch!

Stuart Clark

stuaclar@cisco.com
@bigevilbeard
http://github.com/bigevilbeard

developer.cisco.com

@CiscoDevNet
facebook.com/ciscodevnet/
http://github.com/CiscoDevNet