CISCO

IMAGINE

INTUITIVE

Cisco live!

June 10-14, 2018 · Orlando, FL

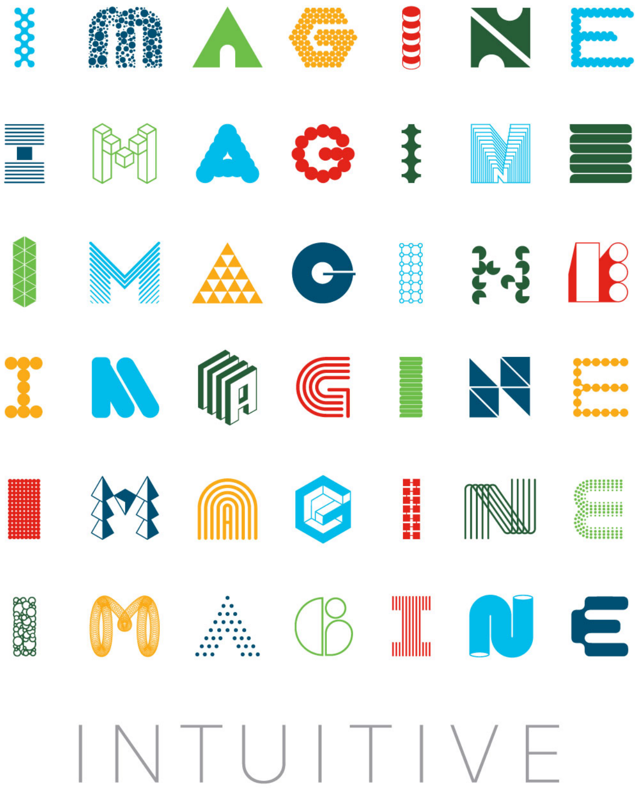
#CLUS

NetDevOps for the Network Dude

How to get started with API's, Ansible and Python

François Caen, Systems Engineer
Kevin Kuhls, Technical Solutions Architect - @sdn_dude

DEVNET-1002



Agenda

- Introduction
- Automation Motivation
- Ansible for CLI automation
- Better machine communication with APIs
- Configuration Abstraction
- Conclusion

Cisco Webex Teams

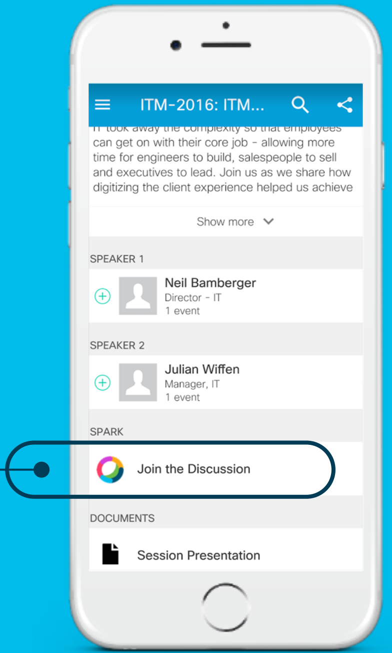
Questions?

Use Cisco Webex Teams (formerly Cisco Spark)
to chat with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install Webex Teams or go directly to the team space
4. Enter messages/questions in the team space

Webex Teams will be moderated
by the speaker until June 18, 2018.



ITM-2016: ITM...

IT took away the complexity so that employees can get on with their core job - allowing more time for engineers to build, salespeople to sell and executives to lead. Join us as we share how digitizing the client experience helped us achieve

Show more

SPEAKER 1

Neil Bamberger
Director - IT
1 event

SPEAKER 2

Julian Wiffen
Manager, IT
1 event

SPARK

Join the Discussion

DOCUMENTS

Session Presentation

cs.co/ciscolivebot#DEVNET-1002

Who are these guys?

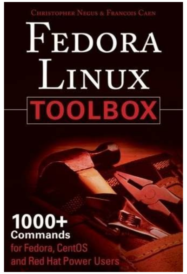
…and should I listen or look at my phone?

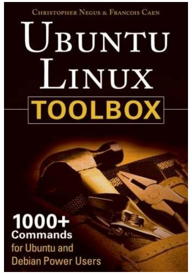
Kevin Kuhls

- 1998 – Cisco Router
- 2002 – PIX Firewall
- BIG LULL
- 2012 – DC Technologies (UCS, Nexus, VMWare)
- 2014 – OpenStack, ACI
- 2015 – Network Programmability, SDN
- Old Dog learning new tricks

François Caen

- 1999 – Linux sysadmin
- 2001 – FLOSS advocate / author


Christopher Negus & Francois Caen
Fedora Linux
TOOLBOX
1000+ Commands
for Fedora, CentOS and Red Hat Power Users

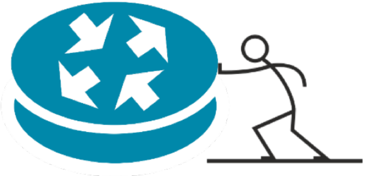

Christopher Negus & Francois Caen
Ubuntu Linux
TOOLBOX
1000+ Commands
for Ubuntu and Debian Power Users

- 2004 – Network Engineer
- 2015 – Cisco SE

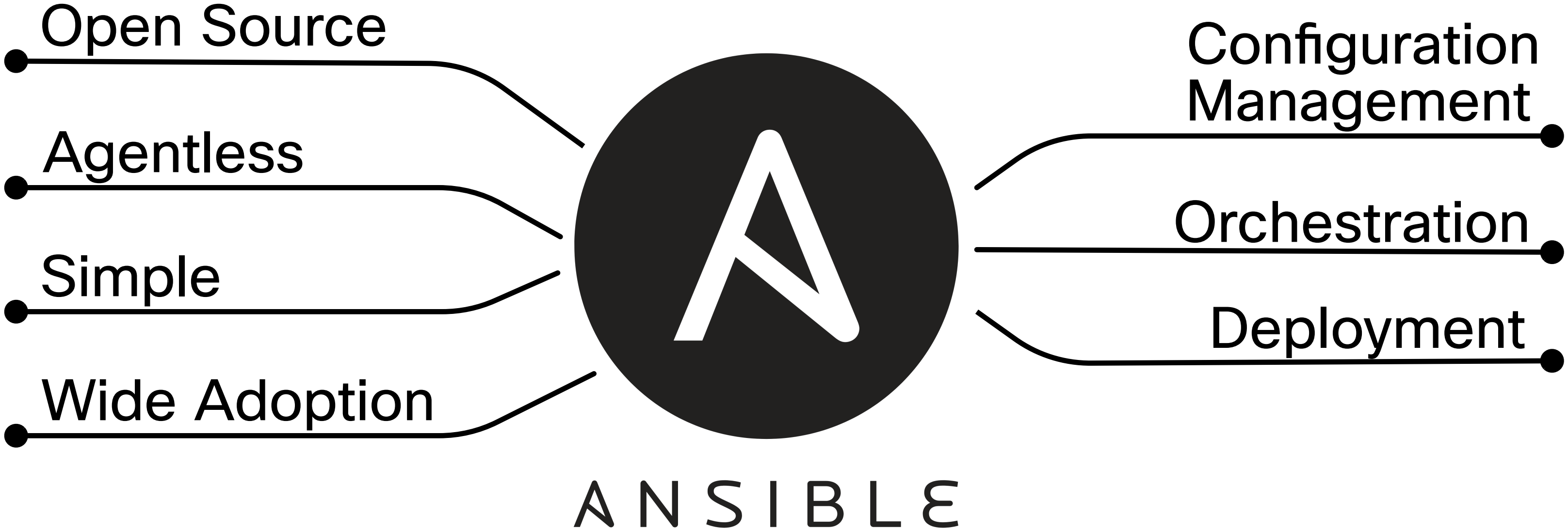
Motivators for Automation

- Cloud-scale:
  - Lots of Equipment: 1000 Network Devices
  - Multiple Operating Systems:
    - IOS, IOSXR, IOSXE, NXOS, ASA OS
    - Multivendor Security Appliances (WAF, DDoS, LB)
  - Small team: 6 people
  - Rapid Deployment
    - Several new Datacenters per year
    - Several Service Deployments requiring changes

- Enterprise-scale:
- Daily repetitive tasks:
  - New device configuration
  - 3rd party NMS config
  - Change one config line on all your devices (NF collector,…)
- Monitoring:
  - Be alerted when a route goes away



What is Ansible?


Open Source

Agentless

Simple

Wide Adoption

Configuration Management

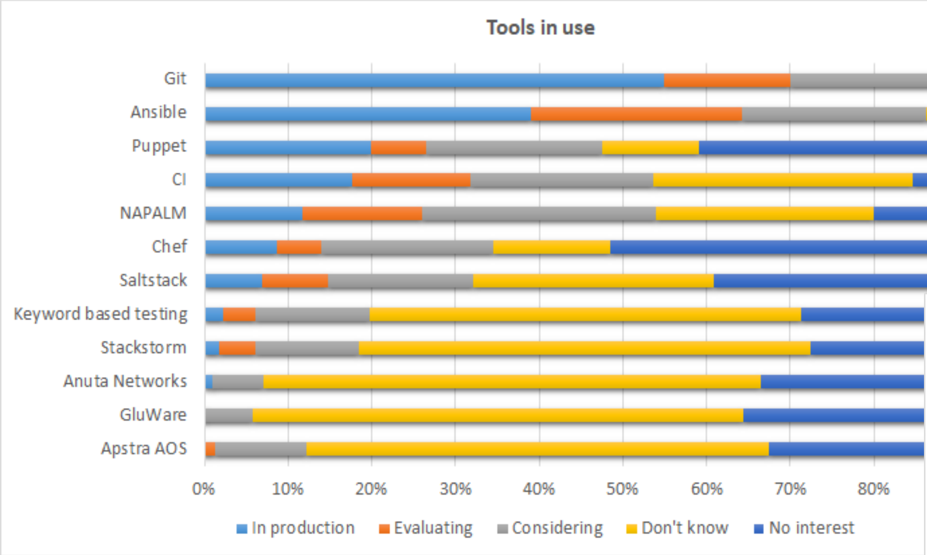
Orchestration

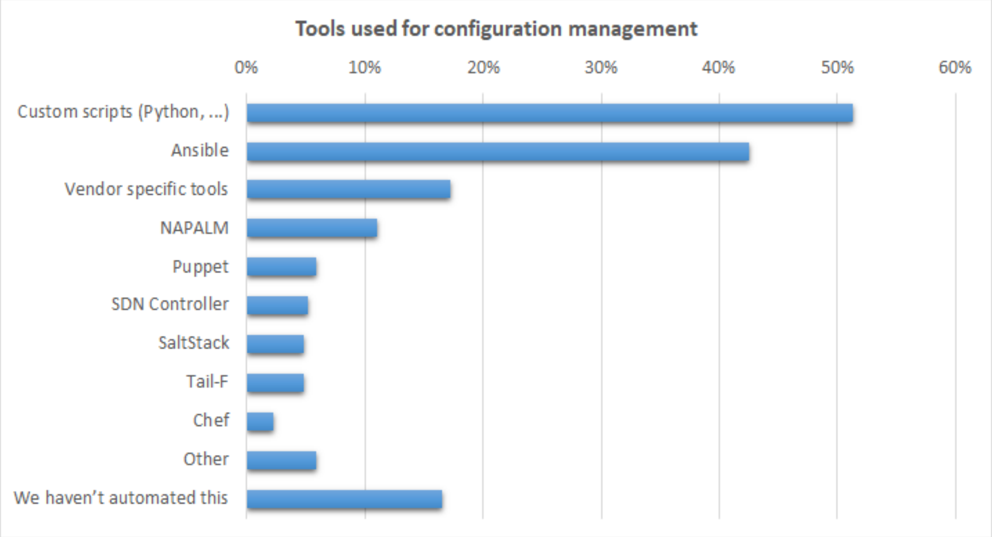
Deployment

Why choose Ansible?

- Agentless
- Server and support teams already using Ansible
- Infrastructure as code
- Simple to use and learn
- Community and vendor driven
- Modular framework, easily modified
- Leverage many common programming languages

NetDevOps Survey

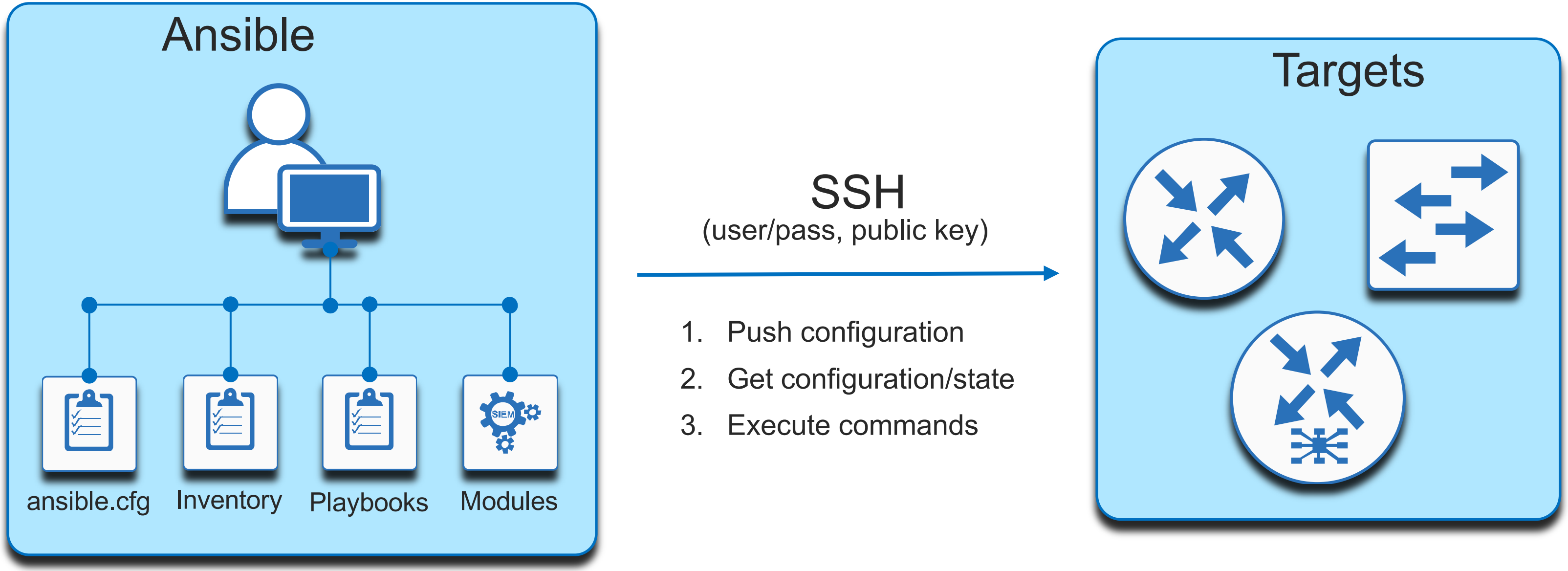




Source:

https://interestingtraffic.nl/2017/03/27/insights-from-the-netdevops-fall-2016-survey/

Ansible in a nutshell



Ansible

ansible.cfg

Inventory

Playbooks

Modules

SSH

(user/pass, public key)

1. Push configuration
2. Get configuration/state
3. Execute commands

Targets

What can I automate out of the box?

It might be faster to say what you can’t…

- Cisco Network Devices
  - IOS
  - XR
  - NX-OS (CLI and NXAPI)
  - ASA
  - ACI
  - UCS
- Other networking vendors
- NETCONF
- Generic network modules
- NSO
- Make your own – its open source

- RHEL
- Ubuntu
- Openstack
- Containers
- Files
- Package Managers

Ansible for Networking

ANSIBLE
2.0

```
- name: load new acl into device
  ios_config:
    lines:
      - 10 permit ip host 1.1.1.1 any log
      - 20 permit ip host 2.2.2.2 any log
      - 30 permit ip host 3.3.3.3 any log
      - 40 permit ip host 4.4.4.4 any log
      - 50 permit ip host 5.5.5.5 any log
    parents: ip access-list extended test
    before: no ip access-list extended test
    match: exact
    provider: "{{ cli }}"
```

Jinja Template

Contains **variables** and/or **expressions** which get replaced with values when *rendered*

```
# Simple Variable Replacment
hostname {{sitecode}}-fw

# Variable Replacement based on Dictionary
route outside 0.0.0.0 0.0.0.0 {{config['vlan101']['ip'][1]}}

# Variable Replacement by Filter
route outside 0.0.0.0 0.0.0.0 {{ external_net_cidr | ipaddr(’1') | ipaddr('address') }}"

# Loop Through set of data to create multiple lines
{%for route in config['routes'] %}
route oob-vpn {{config['routes'][route]['network']}} {{config['routes'][route]['mask']}} {{config['vlan90']['ip'][1]}}
{% endfor %}

# Conditional Statements
{% if config[‘vlan41’] is defined %}
route dmzext {{config['vlan41']['ip'][0]}} {{config['vlan41']['ip'].netmask}} {{config['vlan102']['ip'][1]}}
{endif %}
```

Yaml

- Structure to define:
  - dictionary (unordered set of key value pairs, lists)
  - list of items
  - key value pair

```
# A sample employee record
name: Francois Caen
job: Systems Engineer
employed: True
languages:
    French: Native
    English: Fluent
    German: Novice
    python: Novice
education: Maitrise
favorite drinks:
    - beer
    - gin
```

Ansible 2.x Exercise

Configuration Management Today: CLI

- Human Friendly
- Task Oriented
- Easy To Replay
- No Special Tools

- Syntax format changes
- No Structured output
- No Error Reporting
- No Transaction management

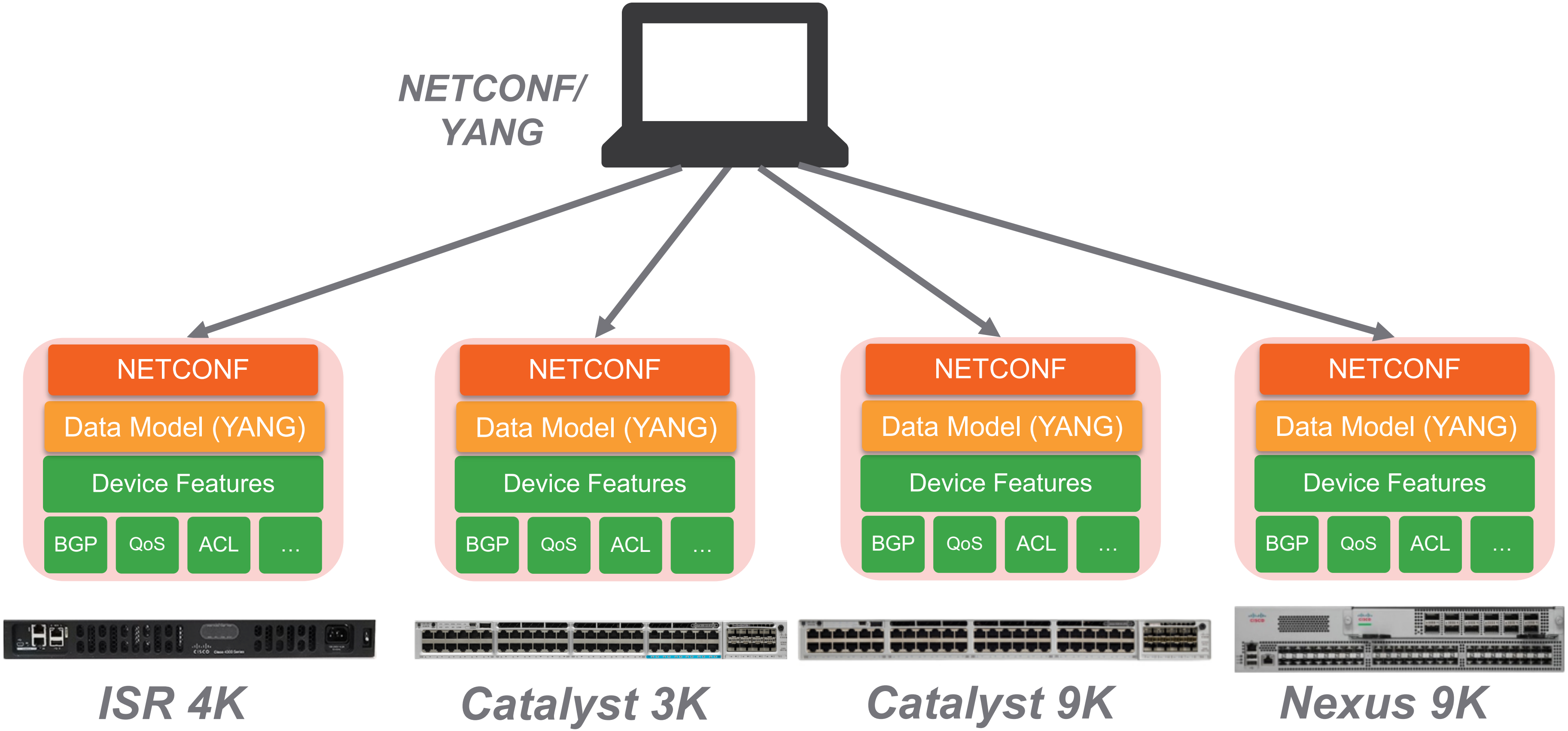
Alternatives to CLI automation?

APIs – Application Programming Interfaces

“A ***set of Function Calls*** that allow talking to a system”

- Programming Building block
- APIs can have various **Properties**
  - Transport (SSH, HTTP)
  - Encoding (XML, JSON, ProtoBuffer)
  - Data structure (Data Models)
- Some **Examples** of APIs
  - The Twitter API
  - The Java API

NETCONF: Consistency across Cisco platforms



NETCONF operations: <get-config>

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <get-config>
  <source>
   <running/>
  </source>
  <filter>
   <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
    <interface>
    <name>GigabitEthernet0</name>
    </interface>
   </interfaces>
  </filter>
 </get-config>
</rpc>
```

NETCONF ansible module

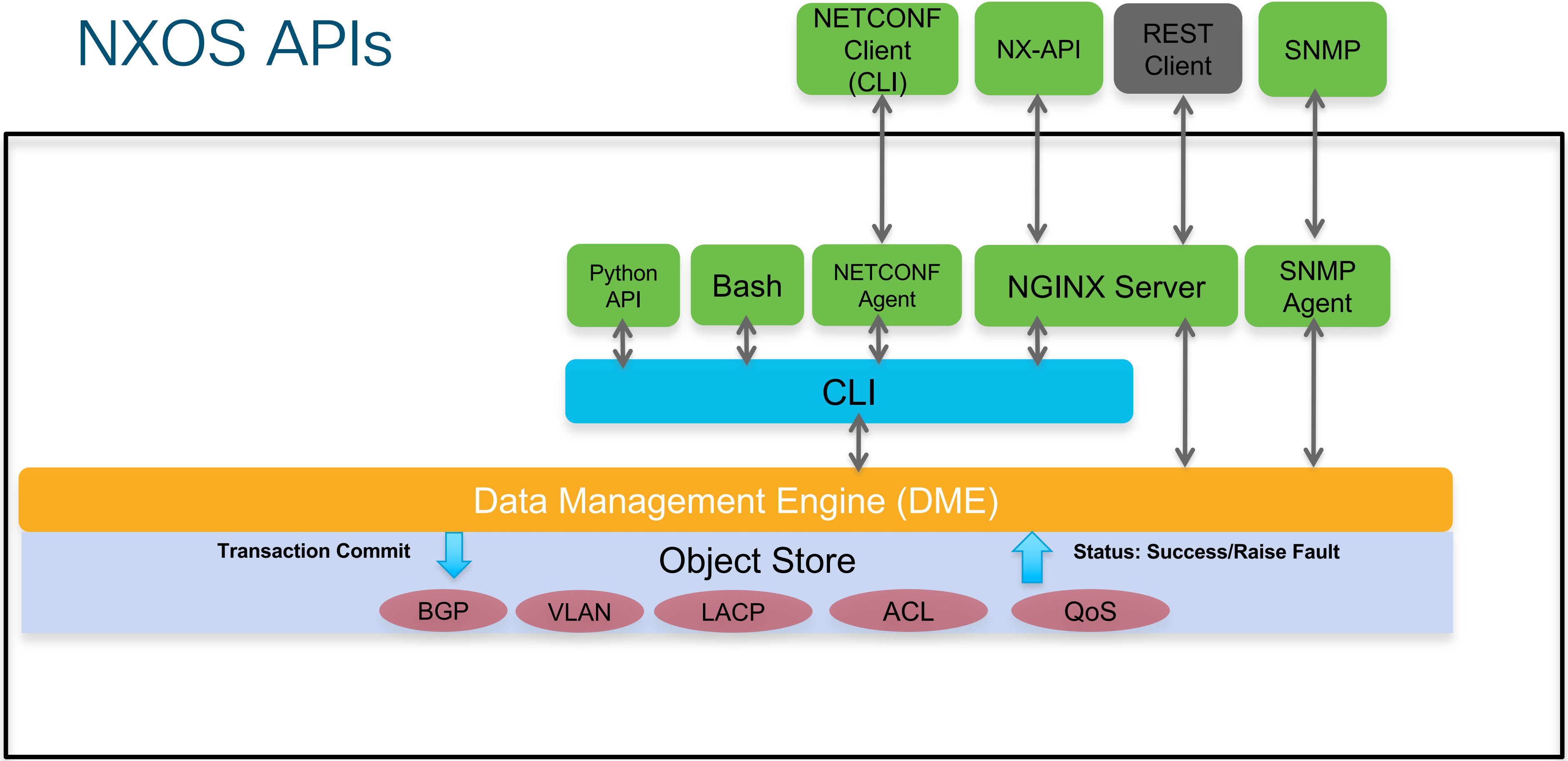
```
- name: configure new ntp server
  netconf_config:
  xml: |
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
    <system xmlns="urn:ietf:params:xml:ns:yang:ietf-system">
      <ntp>
        <enabled>true</enabled>
          <server>
            <name>ntp1</name>
            <udp><address>127.0.0.1</address></udp>
          </server>
        </ntp>
      </system>
    </config>
```

ACI

- 31 modules making specific REST calls
- aci_rest module to send JSON configuration over generic REST POST

```
- name: Add Bridge Domain
  aci_bd:
    host: "{{apic_ip}}”
    username: "{{username}}”
    password: "{{password}}”
    validate_certs: false
    state: present
    tenant: prod
    bd: web_servers
    vrf: prod_vrf
```

NXOS APIs


NETCONF Client (CLI)
NX-API
REST Client
SNMP
Python API
Bash
NETCONF Agent
NGINX Server
SNMP Agent
CLI
Data Management Engine (DME)
Transaction Commit
Object Store
Status: Success/Raise Fault
BGP
VLAN
LACP
ACL
QoS

NXOS

```
# create a new neighbor
- nxos_bgp_neighbor:
    asn: 65535
    neighbor: 3.3.3.3
    local_as: 20
    remote_as: 30
    description: "just a description”
    update_source: Ethernet1/3
    state: present
    transport: cli #or nxapi
```

DEVNET-1002

NETCONF & NX-OS Exercise

Configuration Abstraction

Infrastructure as Code Example

Variable structure to represent a multi-tenant fabric

```
fabric:
  - tenant_name: DEVELOPMENT
    tenant_id: 103
    ints:
      - vlan_id: 3240
        name: "10_103_240_0-DATA"
        subnet: " 10.103.240.0/24"
  - tenant_name: EMPLOYEE
    tenant_id: 101
    ints:
      - vlan_id: 1240
        name: "10_101_240_0-DATA"
        subnet: " 10.103.240.0/24”
      - vlan_id: 1241
        name: "10_101_241_0-VOICE"
        subnet: " 10.101.241.0/24”
```

Infrastructure as code Exercise

Complete your online session evaluation

Give us your feedback to be entered into a Daily Survey Drawing.

Complete your session surveys through the Cisco Live mobile app or on www.CiscoLive.com/us.

Don’t forget: Cisco Live sessions will be available for viewing on demand after the event at www.CiscoLive.com/Online.

References

Ansible – http://www.Ansible.com
Jinja - https://kontrolissues.net/2016/01/14/intro-to-jinja2/
YAML - http://www.yaml.org/start.html

Source code in Github:

- Clone exercises from session: git clone https://github.com/kuhlskev/devnet1002
- Ansible Networking - https://github.com/ansible/ansible-modules-core/tree/stable-2.2/network

Blogs:

- https://pynet.twb-tech.com/
- http://jedelman.com
- https://networklore.com/

Continue Your Education

- Demos in the Cisco campus and DevNet Zone
- Walk-in Self-Paced Labs
- Lunch & Learn
- Meet the Engineer 1:1 meetings
- Related sessions

Network Programmability Cisco Education Offerings

Course	Description	Cisco Certification
Developing with Cisco Network Programmability (NPDEV)	Provides Application Developers with comprehensive curriculum to develop infrastructure programming skills; Addresses needs of software engineers who automate network infrastructure and/or utilize APIs and toolkits to interface with SDN controllers and individual devices	Cisco Network Programmability Developer (NPDEV) Specialist Certification
Designing and Implementing Cisco Network Programmability (NPDESI)	Provides network engineers with comprehensive soup-to-nuts curriculum to develop and validate automation and programming skills; Directly addresses the evolving role of network engineers towards more programmability, automation and orchestration	Cisco Network Programmability Design and Implementation (NPDESI) Specialist Certification
Programming for Network Engineers (PRNE)	Learn the fundamentals of Python programming – within the context of performing functions relevant to network engineers. Use Network Programming to simplify or automate tasks	Recommended pre-requisite for NPDESI and NPDEV Specialist Certifications
Cisco Digital Network Architecture Implementation Essentials (DNAIE)	This training provides students with the guiding principles and core elements of Cisco's Digital Network Architecture (DNA) architecture and its solution components including; APIC-EM, NFV, Analytics, Security and Fabric.	None

For more details, please visit: http://learningnetwork.cisco.com
Questions? Visit the Learning@Cisco Booth

DEVNET-1002

Thank you

INTUITIVE

cisco

IMAGINE

INTUITIVE

Cisco live!

June 10-14, 2018 · Orlando, FL

#CLUS

Building the Environment

This is a rough guideline of how to bring up the CSR 1000v environment.

- Git client
- VirtualBox 5.2.6
- Vagrant 2.0.1
- Docker 17.12
- cdrtools (in particular mkisofs)
- a build environment (e.g. compiler, make, ...), suggest to use MacPorts or Brew if running on a Mac
- Clone the iso-xrv-x64-vbox repository from GitHub
- IOS XE image from Cisco.com (e.g. here, then go to IOS XE Software and download the Everest-16.5.2 .iso file in the Latest tree branch, ~350MB in size)

Building the Environment (cont)

Building the Vagrant Box

- Go to the directory where you cloned the iso-xrv-x64-vbox repository. Start the Vagrant box image build by running the following command:
  iosxe_iso2vbox.py -v ~/Downloads/csr1000v-universalk9.16.05.02.iso

- This will take a while. When done, you need to install the resulting box into Vagrant:
  vagrant box add --name iosxe csr1000v-universalk9.16.05.02.box
  (See the output at the end of the script. It has the exact location of the generated box file and also the command to add / replace the Vagrant box file).

Configure and Start Routers

The next steps are required to prepare configuration disks for the routers

- Clone this repo from GitHub into a new directory: https://github.com/kuhlskev/devnet1002
- Make sure that the Vagrant box name matches the one configured in the Vagrant file
- Ensure you have the required tools installed
- run make to create the ISO files with the router configurations
- Bring up the routers using vagrant up (brings up both) or vagrant up rtr1 to only start rtr1

cisco

IMAGINE

INTUITIVE

Cisco live!

June 10-14, 2018 · Orlando, FL

#CLUS