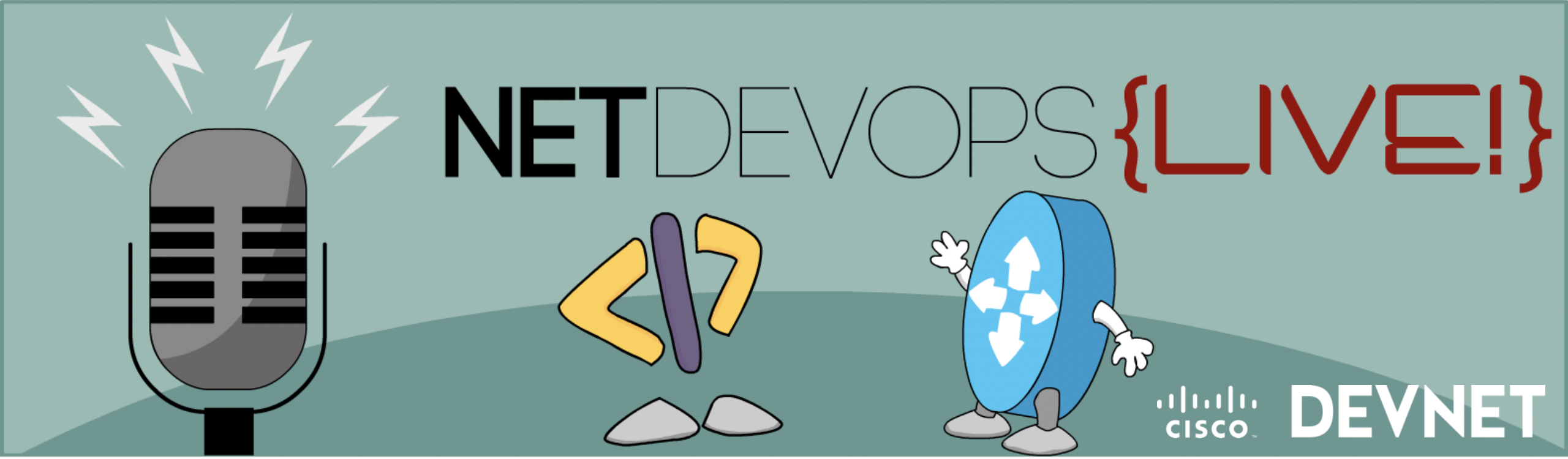


Picking your First Network Automation Project

Eric Thiel, CCIE 11956 DC/Security
Developer Advocacy, Cisco DevNet
Twitter: @securenetwrk

Season 2, Talk 2

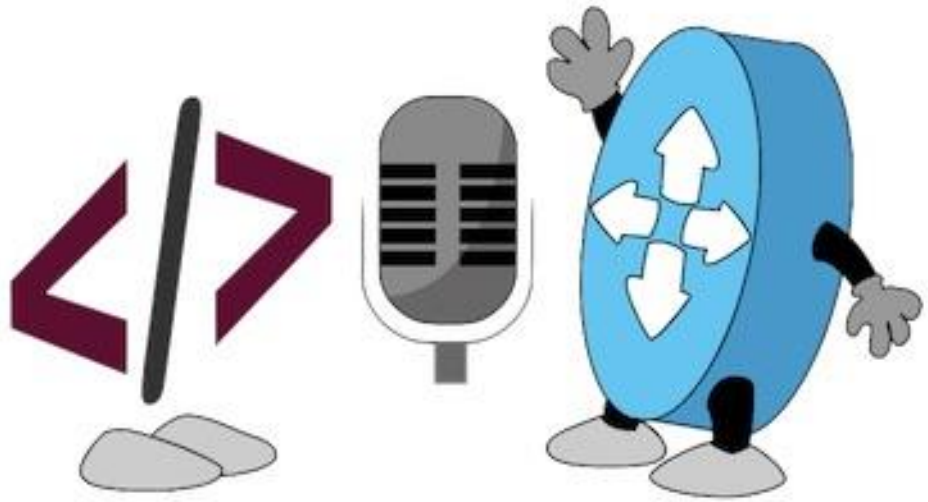
https://developer.cisco.com/netdevops/live

http://cs.co/ndl

Help us track NetDevOps Live Interest!

What are we going to talk about?

- Brainstorming ideas
  - Whiteboard 101
- Selecting your top candidates
  - Everyone gets a say
  - Weight results based on ease and risk
- Map out the workflow
- Choose your tools
- Demo

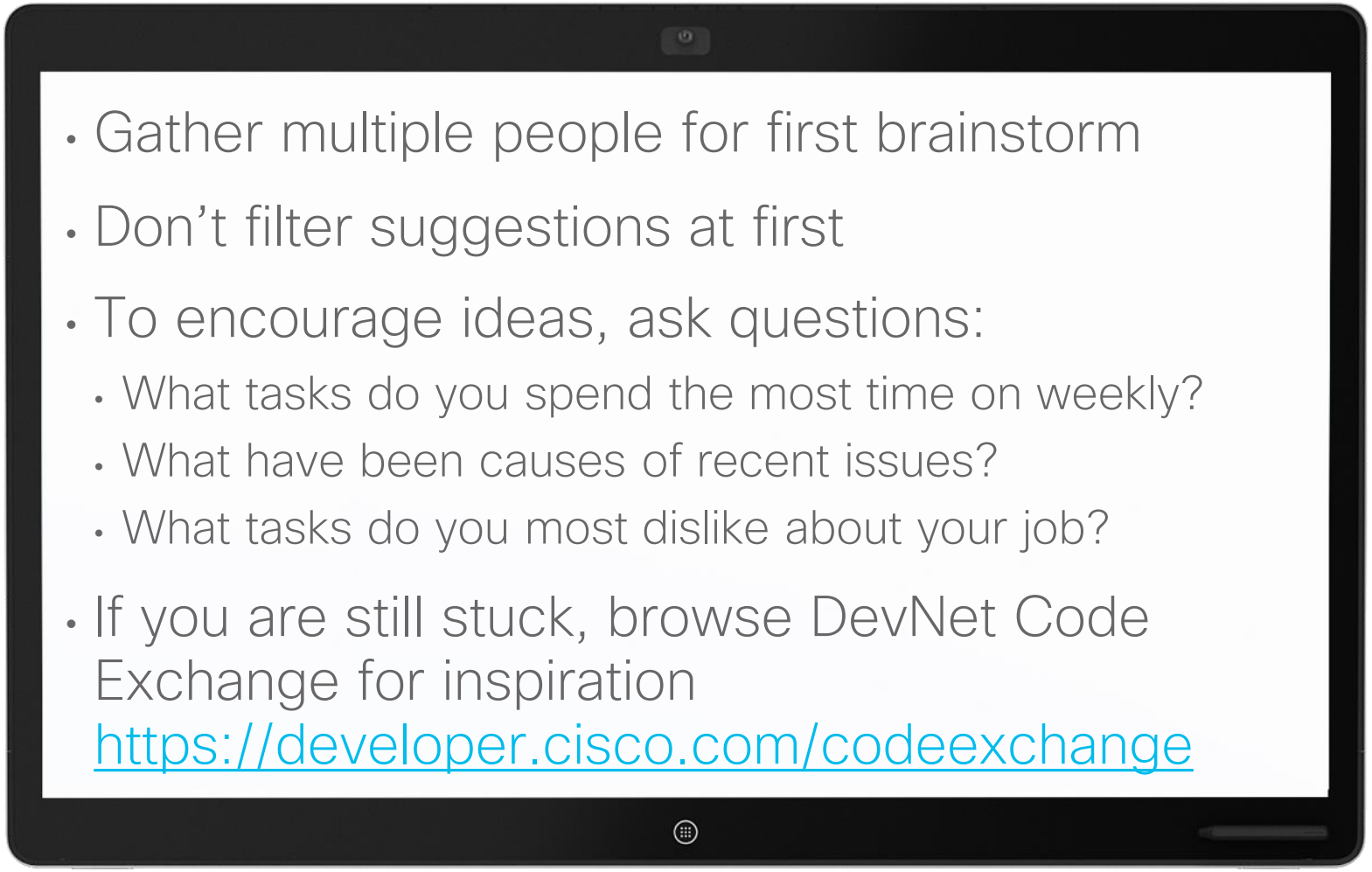


Before we start...

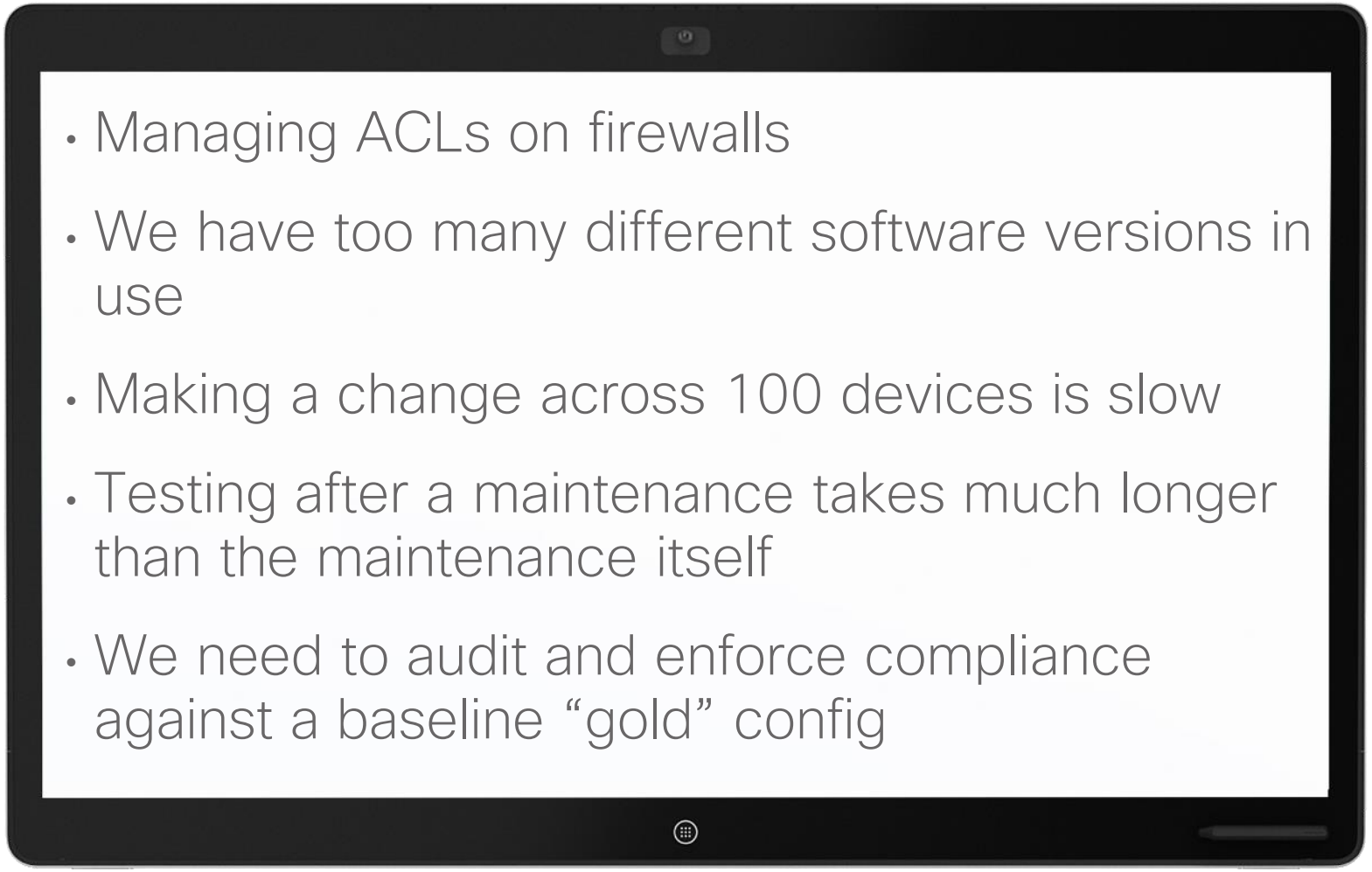
- ✓ “Perfect is the enemy of good”
- ✓ Choose the best tool for the job… and your skills

Brainstorming ideas

Whiteboard 101

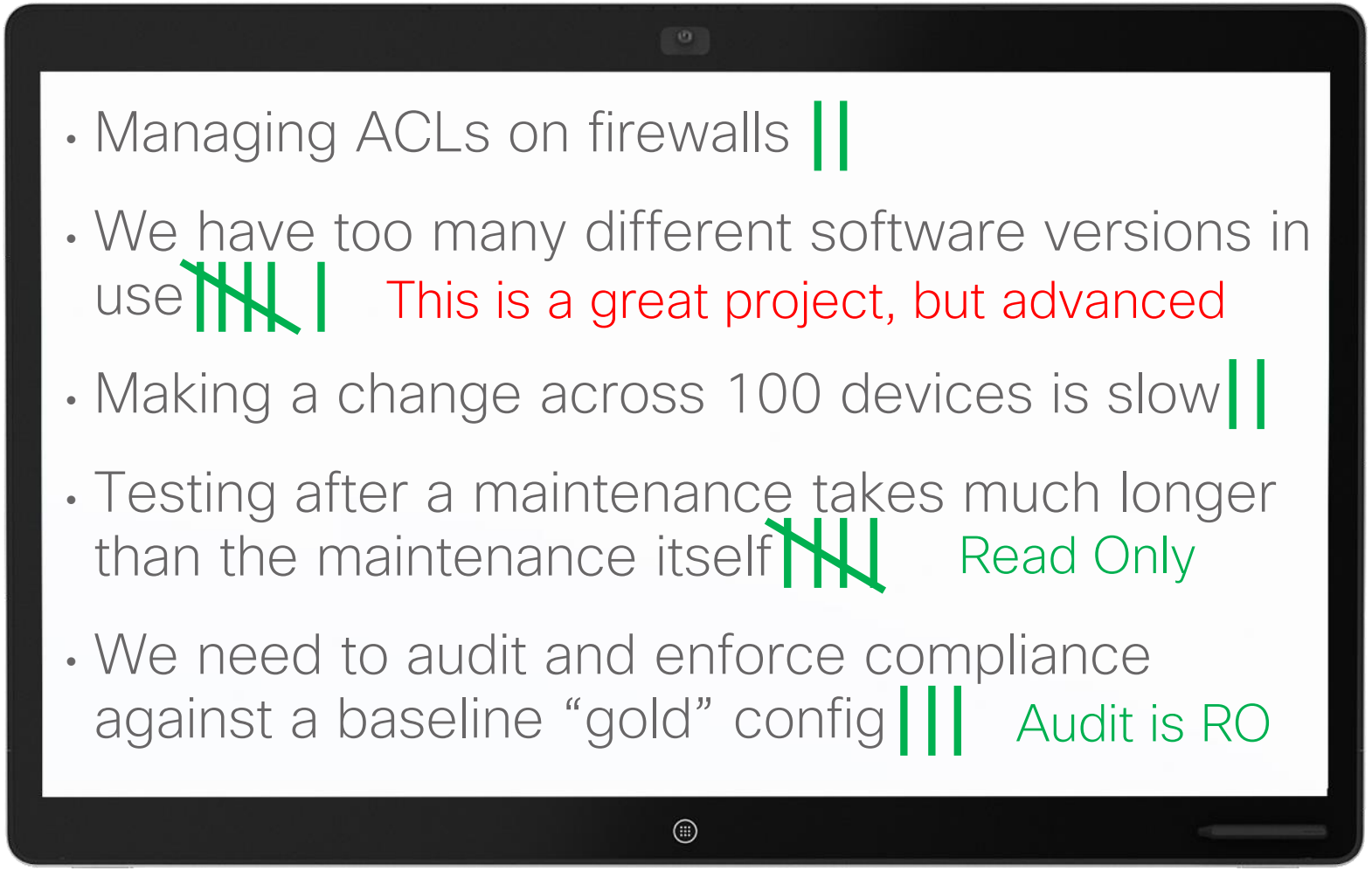


Whiteboard ideas



Selecting your top candidates

Whiteboard ideas



Map out the workflow

Ask “How would I do this manually”

- Testing after a maintenance takes much longer than the maintenance itself:
- SSH to each router at the site, look at OSPF/BGP and make sure it “looks about right”
- Ping from a handful of devices to a handful of other devices
- “Show log” and look for any recent unexpected entries

Ask “How would I do this manually”

- We need to audit and enforce compliance against a baseline “gold” config
- Build list of baseline config commands in notepad
- Build list of devices to connect to, and connect to each one at a time
- Show running config, and manually compare each line in notepad against existing config
- Look for any additional configs that should not be allowed (such as extra users or logging servers)
- Build list of changes needed in a text file and open a ticket to make changes during next change window

Choose your tools

Which vehicle would win in a race?

So what tools are in your toolbelt?

- Network controllers (DNAC/APIC) – Network automation out of the box
- Genie CLI – Run as a free-standing app with no code required.
- Ansible – A relatively easy to use configuration management tool with no Python experience needed
- NAPALM – A convenient Python library that abstracts different devices into a common set of functions
- NSO – Cisco Network Services Orchestrator CLI based device templates are a great way to get started automating across diverse platforms
- Python + requests – Using the requests module within Python allows easy interaction with REST based APIs

Now, what protocol would you like to use?

- As with tools, there are multiple protocols available for automation.
- RESTCONF – Newer platforms tend to offer a RESTCONF API. When a tool supports RESTCONF, that option will often outperform the other options, and could be a great choice
- NETCONF – Even more common than RESTCONF, NETCONF sends payloads via SSH, but in XML format, making it a reliable and robust option.
- CLI – The tools above generally offer at a minimum CLI-based automation. In that case, the tool must parse the text based CLI responses and present them to the script for processing

Let's choose a tool

- Testing after a maintenance takes much longer than the maintenance itself:
- This use case turns out to be the easiest by far. It is also low risk since it is only reading information. To solve it, you could leverage a tool like Genie command line tool.
- Genie leverages pyATS behind the scenes, but you can use it to gather operational state of the network without writing any code
- If you do know Python and wish to extend this use case further, you could use the Genie Python library to handle data gathering as part of a broader script

Netmiko vs. Genie CLI

Netmiko:

```
#!/usr/bin/env python
from netmiko import ConnectHandler
from getpass import getpass

ip_addr = raw_input("Enter IP Address: ")

device = {
    'device_type': 'cisco_ios',
    'ip': ip_addr,
    'username': 'admin',
    'password': getpass(),
    'port': 22,
}

net_connect = ConnectHandler(**device)
output = net_connect.send_command_expect("show version")

print
print '#' * 50
print output
print '#' * 50
print
```

Genie CLI:

```
genie learn config routing --testbed-file lab1.yaml --output
myout/output1
```

Let’s try it out

Let's choose a tool

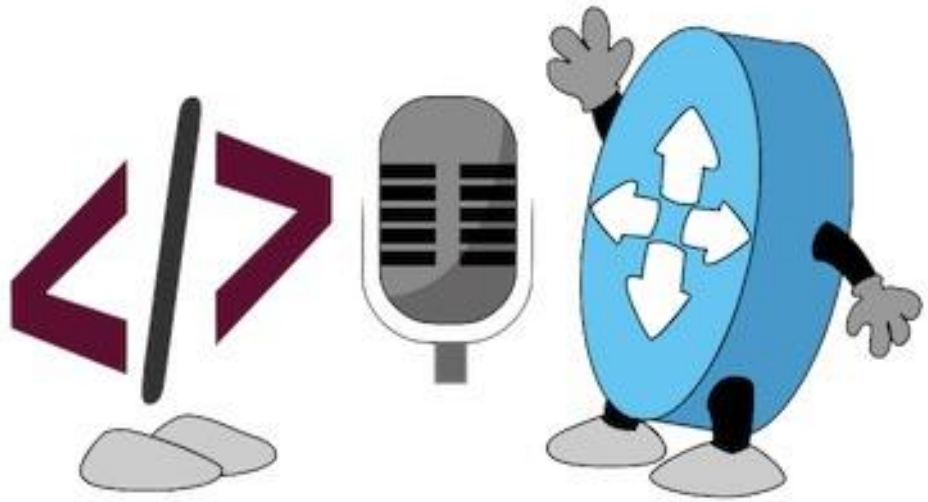
- We need to audit and enforce compliance against a baseline "gold" config
- Start with read-only (audit), and worry about enforcement later
- If you know Python, there are several options such as NAPALM
- If you don't know Python, or if you do but want a simple solution, there are a number of configuration management tools that do this very well. We will use Ansible for our example
- In Ansible, you can define all of the config information you wish to enforce, but run it in "check mode" to audit compliance

Let’s try it out

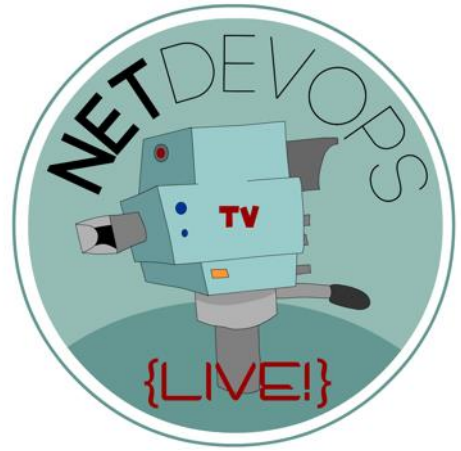
Summing up

What did we talk about?

- Brainstorming ideas
  - Whiteboard 101
- Selecting your top candidates
  - Everyone gets a say
  - Weight results based on ease and risk
- Map out the workflow
- Choose your tools
- Demo



Webinar Resource List



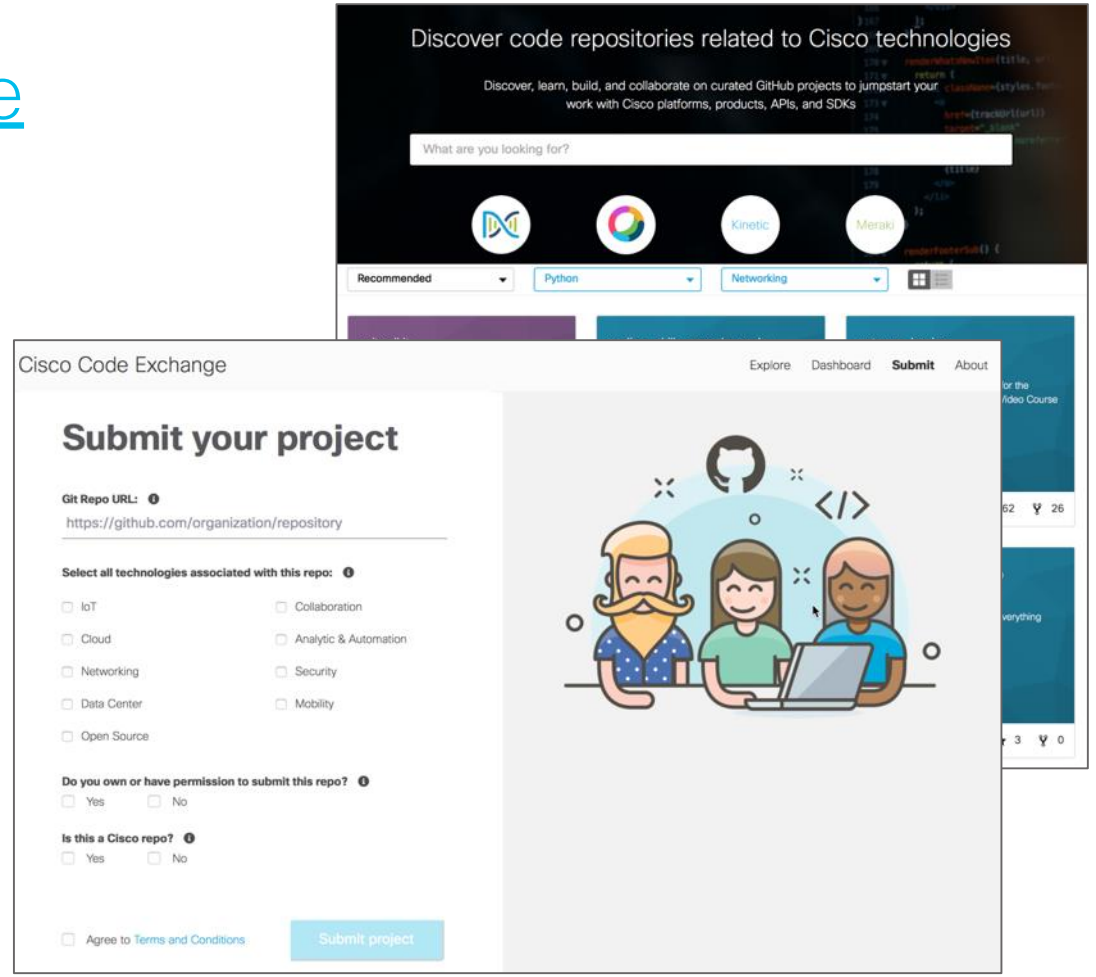
- Docs and Links
  - https://developer.cisco.com/python
- Learning Labs
  - Laptop Setup http://cs.co/lab-dev-setup
  - Ansible for IOS XE https://developer.cisco.com/learning/modules/intro-ansible-iosxe
  - pyATS and Genie https://developer.cisco.com/learning/modules/pyats-genie
- DevNet Sandboxes
  - Cisco DNA Center Always On http://cs.co/sbx-dnac-ao
  - IOS Always On http://cs.co/sbx-iosxe
- Code Samples
  - http://cs.co/etcode

NetDevOps Live! Code Exchange Challenge

developer.cisco.com/codeexchange

Hold a whiteboard session for your team. Pick one task that involves only reading information, and test it using Genie CLI, Ansible, or the tool of your choice

Example: Back up configs from your IOS XE devices to a local file using Ansible.



Looking for more about NetDevOps?

- NetDevOps on DevNet
  developer.cisco.com/netdevops
- NetDevOps Live!
  developer.cisco.com/netdevops/live
- NetDevOps Blogs
  blogs.cisco.com/tag/netdevops
- Network Programmability Basics Video Course
  developer.cisco.com/video/net-prog-basics/

Got more questions? Stay in touch!

Eric D. Thiel


erthiel@cisco.com


@securenetwrk


http://github.com/securenetwrk

CISCO DEVNET

LEARN CODE INSPIRE CONNECT

developer.cisco.com

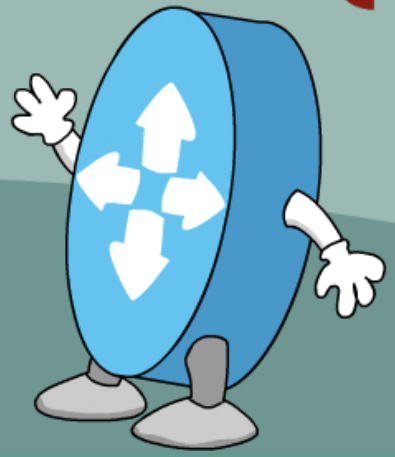

@CiscoDevNet


facebook.com/ciscodevnet/


http://github.com/CiscoDevNet

NETDEVOPS {LIVE!}

cisco DEVNET



https://developer.cisco.com/netdevops/live

@netdevopslive