The bridge to possible

Programmability Skills for Engineers Who Don't Want to Code

Hank Preston, Principal Engineer Learning and Certifications
ccie 38336 | devnet expert 20220001

BRKOPS-1237

Reception
To Do
IN PROGRESS
PENDING
DONE
ALARM
ALARM
ALARM
ALARM
ALARM
DON'T
PANIC
OK
SecOps
DevNet
GCP
AWS
CLOUD
RFP
aws
© DCgubbins.com
DCgubbins

Cisco Webex App

Questions?

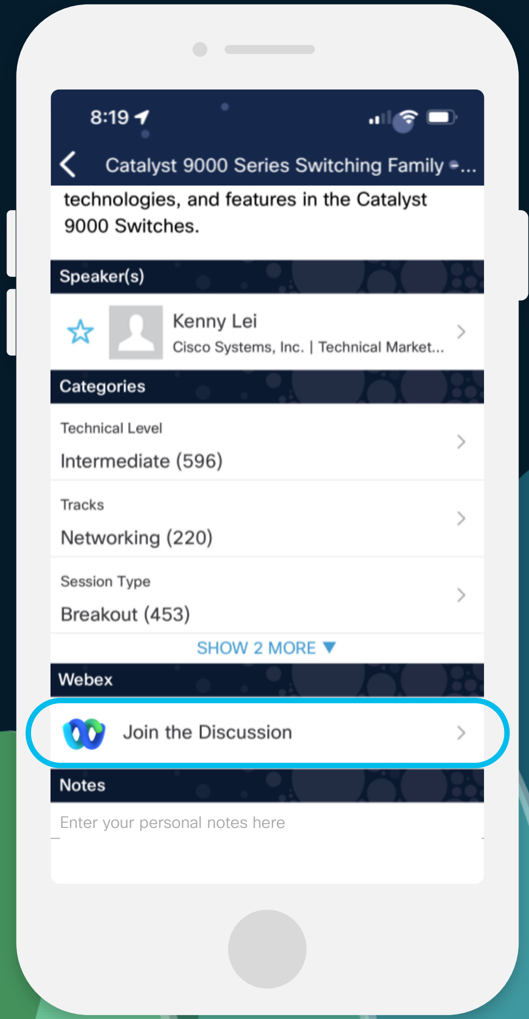
Use Cisco Webex App to chat
with the speaker after the session

How

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

Webex spaces will be moderated by the speaker until June 7, 2024.

https://ciscolive.ciscoevents.com/
ciscolivebot/#BRKOPS-1237



8:19

Catalyst 9000 Series Switching Family -...

technologies, and features in the Catalyst 9000 Switches.

Speaker(s)

Kenny Lei
Cisco Systems, Inc. | Technical Market...

Categories

Technical Level
Intermediate (596)

Tracks
Networking (220)

Session Type
Breakout (453)

SHOW 2 MORE ▼

Webex

Join the Discussion

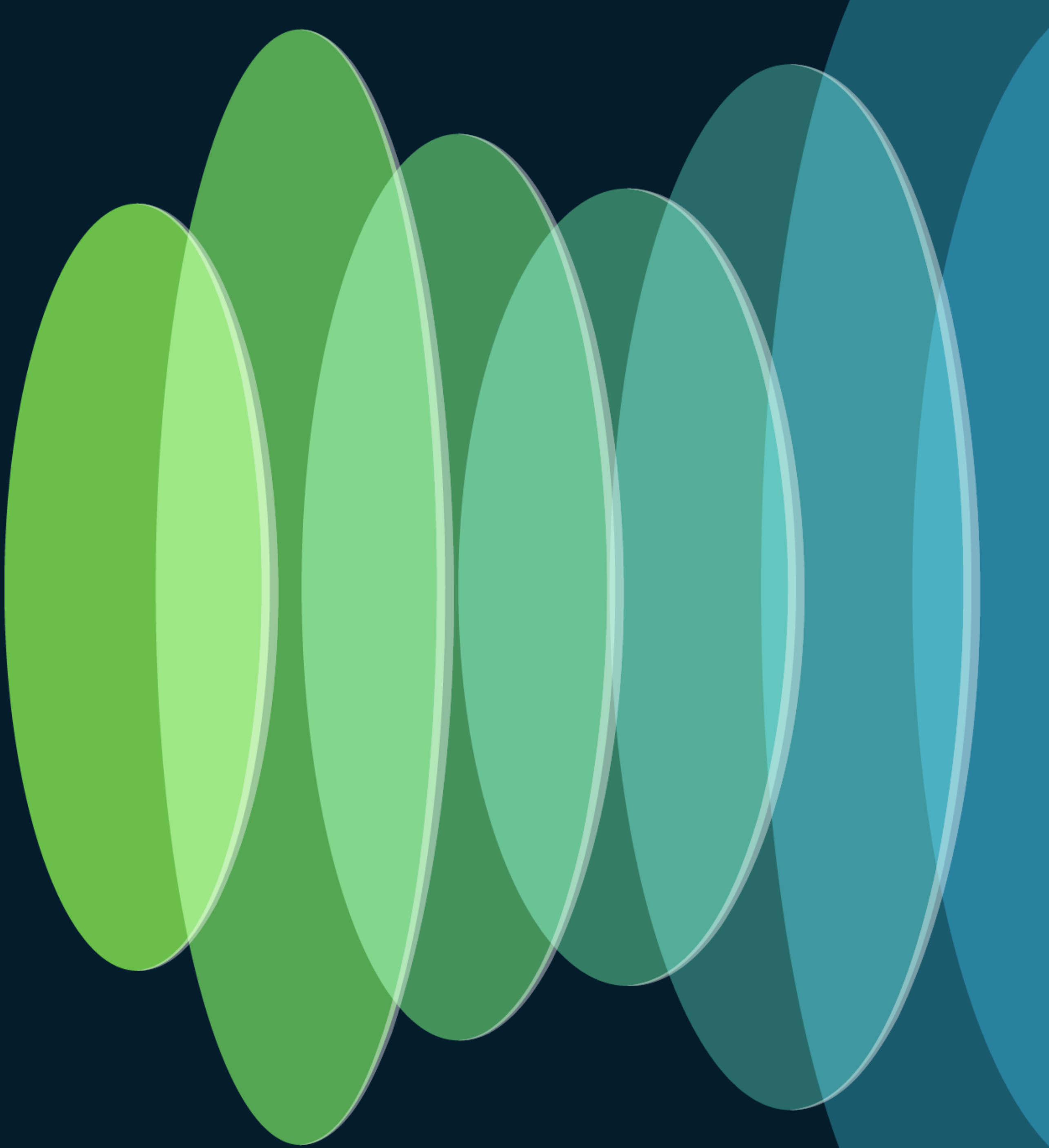
Notes

Enter your personal notes here

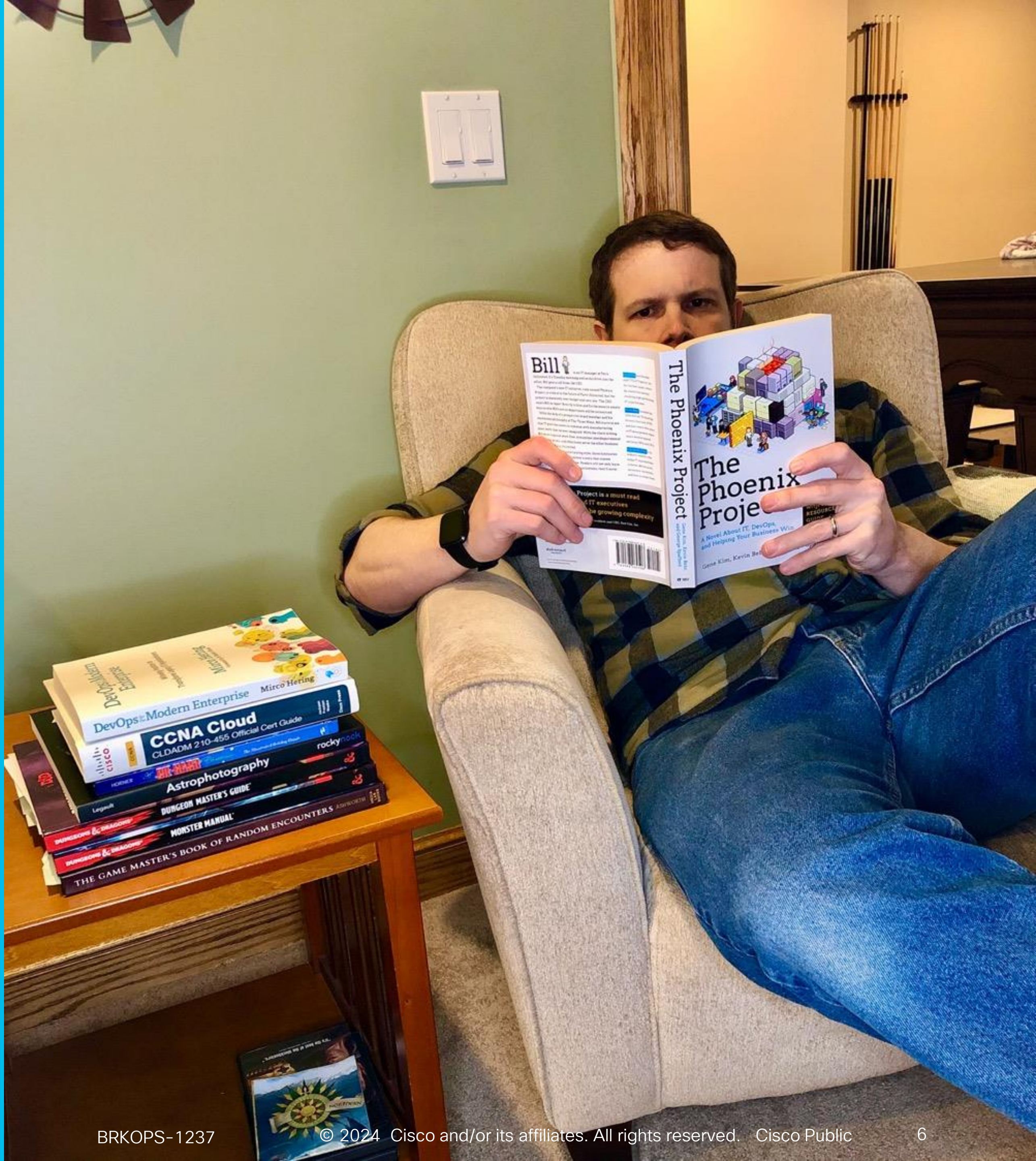
Agenda

- Spreadsheet driven automation
- Aw CRUD, we gotta talk a little about APIs
- Source of Truth, it doesn’t have to be you anymore
- pyATS - How to no-code and automate networks
- Ansible the Hammer of Network Automation

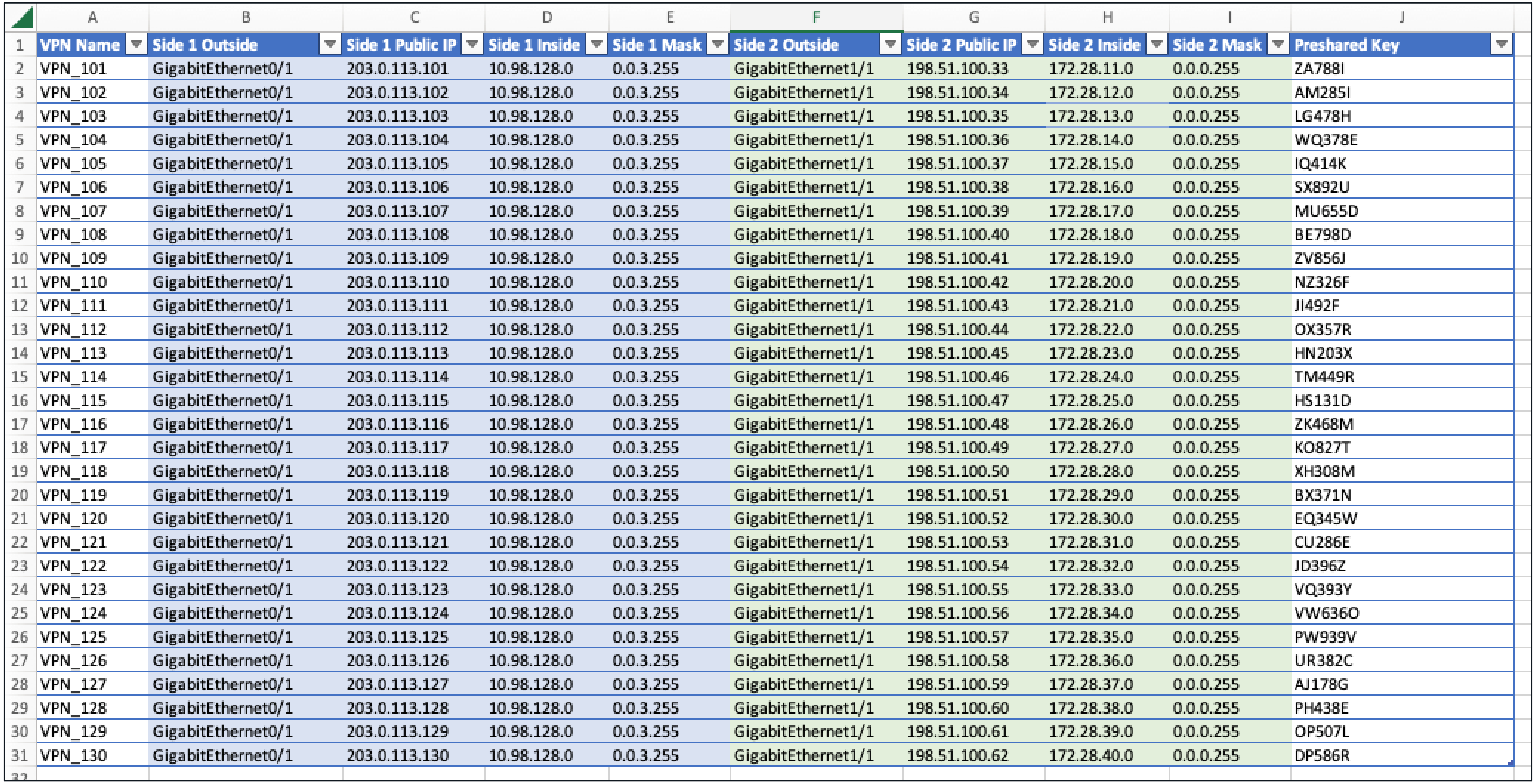
Spreadsheet driven automation



Story Time with Hank


The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
Gene Kim, Kevin Behr
DevOps for the Modern Enterprise
Mirco Hering
CCNA Cloud
CLDADM 210-455 Official Cert Guide
Astrophotography
Dungeon Master's Guide
Monster Manual
The Game Master's Book of Random Encounters

“Network Engineering Runs on Spreadsheets”

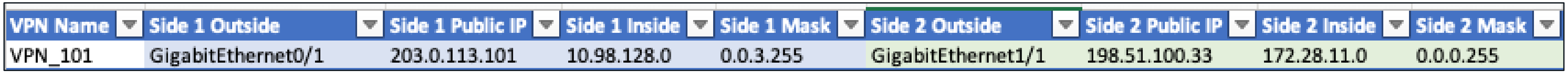


	A	B	C	D	E	F	G	H	I	J
1	VPN Name	Side 1 Outside	Side 1 Public IP	Side 1 Inside	Side 1 Mask	Side 2 Outside	Side 2 Public IP	Side 2 Inside	Side 2 Mask	Preshared Key
2	VPN_101	GigabitEthernet0/1	203.0.113.101	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.33	172.28.11.0	0.0.0.255	ZA788I
3	VPN_102	GigabitEthernet0/1	203.0.113.102	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.34	172.28.12.0	0.0.0.255	AM285I
4	VPN_103	GigabitEthernet0/1	203.0.113.103	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.35	172.28.13.0	0.0.0.255	LG478H
5	VPN_104	GigabitEthernet0/1	203.0.113.104	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.36	172.28.14.0	0.0.0.255	WQ378E
6	VPN_105	GigabitEthernet0/1	203.0.113.105	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.37	172.28.15.0	0.0.0.255	IQ414K
7	VPN_106	GigabitEthernet0/1	203.0.113.106	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.38	172.28.16.0	0.0.0.255	SX892U
8	VPN_107	GigabitEthernet0/1	203.0.113.107	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.39	172.28.17.0	0.0.0.255	MU655D
9	VPN_108	GigabitEthernet0/1	203.0.113.108	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.40	172.28.18.0	0.0.0.255	BE798D
10	VPN_109	GigabitEthernet0/1	203.0.113.109	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.41	172.28.19.0	0.0.0.255	ZV856J
11	VPN_110	GigabitEthernet0/1	203.0.113.110	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.42	172.28.20.0	0.0.0.255	NZ326F
12	VPN_111	GigabitEthernet0/1	203.0.113.111	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.43	172.28.21.0	0.0.0.255	JI492F
13	VPN_112	GigabitEthernet0/1	203.0.113.112	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.44	172.28.22.0	0.0.0.255	OX357R
14	VPN_113	GigabitEthernet0/1	203.0.113.113	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.45	172.28.23.0	0.0.0.255	HN203X
15	VPN_114	GigabitEthernet0/1	203.0.113.114	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.46	172.28.24.0	0.0.0.255	TM449R
16	VPN_115	GigabitEthernet0/1	203.0.113.115	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.47	172.28.25.0	0.0.0.255	HS131D
17	VPN_116	GigabitEthernet0/1	203.0.113.116	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.48	172.28.26.0	0.0.0.255	ZK468M
18	VPN_117	GigabitEthernet0/1	203.0.113.117	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.49	172.28.27.0	0.0.0.255	KO827T
19	VPN_118	GigabitEthernet0/1	203.0.113.118	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.50	172.28.28.0	0.0.0.255	XH308M
20	VPN_119	GigabitEthernet0/1	203.0.113.119	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.51	172.28.29.0	0.0.0.255	BX371N
21	VPN_120	GigabitEthernet0/1	203.0.113.120	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.52	172.28.30.0	0.0.0.255	EQ345W
22	VPN_121	GigabitEthernet0/1	203.0.113.121	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.53	172.28.31.0	0.0.0.255	CU286E
23	VPN_122	GigabitEthernet0/1	203.0.113.122	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.54	172.28.32.0	0.0.0.255	JD396Z
24	VPN_123	GigabitEthernet0/1	203.0.113.123	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.55	172.28.33.0	0.0.0.255	VQ393Y
25	VPN_124	GigabitEthernet0/1	203.0.113.124	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.56	172.28.34.0	0.0.0.255	VW636O
26	VPN_125	GigabitEthernet0/1	203.0.113.125	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.57	172.28.35.0	0.0.0.255	PW939V
27	VPN_126	GigabitEthernet0/1	203.0.113.126	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.58	172.28.36.0	0.0.0.255	UR382C
28	VPN_127	GigabitEthernet0/1	203.0.113.127	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.59	172.28.37.0	0.0.0.255	AJ178G
29	VPN_128	GigabitEthernet0/1	203.0.113.128	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.60	172.28.38.0	0.0.0.255	PH438E
30	VPN_129	GigabitEthernet0/1	203.0.113.129	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.61	172.28.39.0	0.0.0.255	OP507L
31	VPN_130	GigabitEthernet0/1	203.0.113.130	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.62	172.28.40.0	0.0.0.255	DP586R

Device Configuration Standard

```
crypto isakmp policy 10
  encryption aes
  hash sha256
  authentication pre-share
  group 14

crypto ipsec transform-set <TRANS_SET> esp-aes esp-sha256-hmac

crypto isakmp key <PRESHARED_KEY> address <PEER_ADDRESS>

access-list <ACL_NUM> permit ip <LOCAL_NET> <LOCAL_MASK> <REMOTE_NET> <REMOTE_MASK>

crypto map <MAP_NAME> 10 ipsec-isakmp
  set peer <PEER_ADDRESS>
  set transform-set <TRANS_SET>
  match address <ACL_NUM>

interface <OUTSIDE_INT>
  crypto map <MAP_NAME>
```

Creating the Access List Configuration

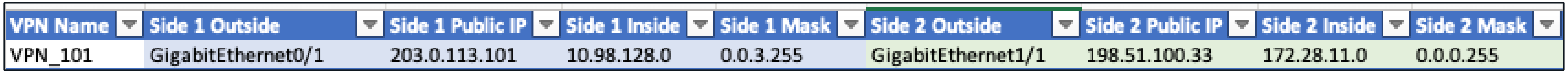


VPN Name	Side 1 Outside	Side 1 Public IP	Side 1 Inside	Side 1 Mask	Side 2 Outside	Side 2 Public IP	Side 2 Inside	Side 2 Mask
VPN_101	GigabitEthernet0/1	203.0.113.101	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.33	172.28.11.0	0.0.0.255

```
! Site 1 Access List
access-list <ACL_NUM> permit ip <LOCAL_NET> <LOCAL_MASK> <REMOTE_NET> <REMOTE_MASK>
```

```
! Site 1 Access List
access-list 100 permit ip 10.98.128.0 0.0.3.255 172.28.11.0 0.0.0.255
```

Creating the Access List Configuration

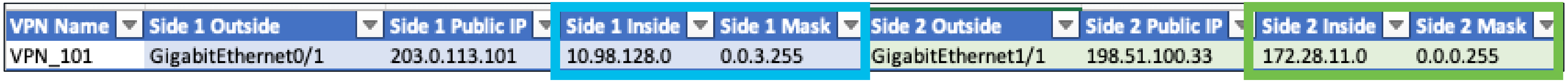


VPN Name	Side 1 Outside	Side 1 Public IP	Side 1 Inside	Side 1 Mask	Side 2 Outside	Side 2 Public IP	Side 2 Inside	Side 2 Mask
VPN_101	GigabitEthernet0/1	203.0.113.101	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.33	172.28.11.0	0.0.0.255

```
! Site 2 Access List
access-list <ACL_NUM> permit ip <LOCAL_NET> <LOCAL_MASK> <REMOTE_NET> <REMOTE_MASK>
```

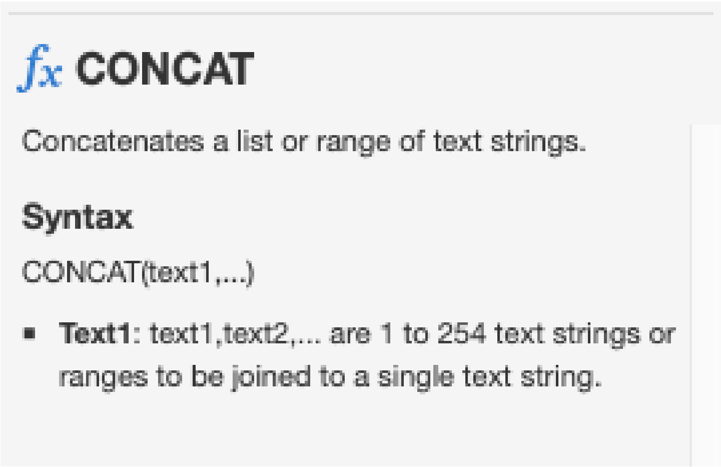
```
! Site 2 Access List
access-list 100 permit ip 172.28.11.0 0.0.0.255 10.98.128.0 0.0.3.255
```

Using CONCAT Function in Excel to Create Strings



VPN Name	Side 1 Outside	Side 1 Public IP	Side 1 Inside	Side 1 Mask	Side 2 Outside	Side 2 Public IP	Side 2 Inside	Side 2 Mask
VPN_101	GigabitEthernet0/1	203.0.113.101	10.98.128.0	0.0.3.255	GigabitEthernet1/1	198.51.100.33	172.28.11.0	0.0.0.255

```
=CONCAT("access-list 100 permit ip ",
        VPNs[@[Side 1 Inside]]," ", VPNs[@[Side 1 Mask]], " ",
        VPNs[@[Side 2 Inside]]," ", VPNs[@[Side 2 Mask]])
```



fx **CONCAT**

Concatenates a list or range of text strings.

Syntax

CONCAT(text1,...)

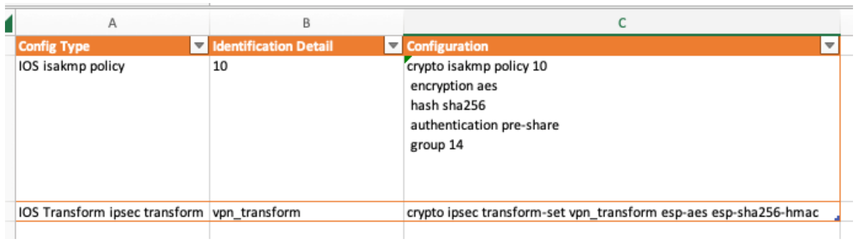
- **Text1**: text1,text2,... are 1 to 254 text strings or ranges to be joined to a single text string.

- Fill in the blank configuration creation
- Once formulas created, ”fill down” for other rows / sites

Help info on CONCAT Function

CONCAT your CONCATs for Full Configs

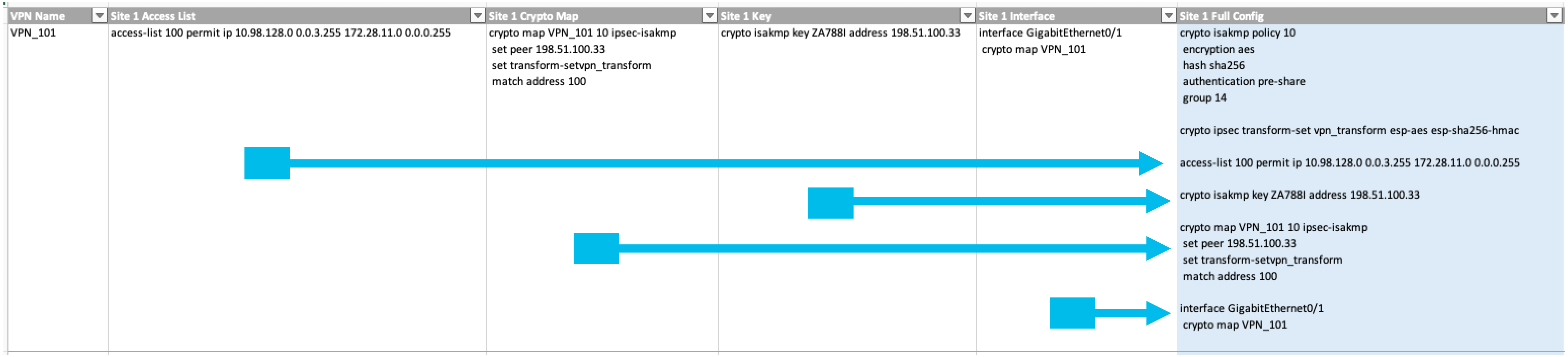
Common Config Elements



Config Type	Identification Detail	Configuration
IOS isakmp policy	10	crypto isakmp policy 10 encryption aes hash sha256 authentication pre-share group 14
IOS Transform ipsec transform	vpn_transform	crypto ipsec transform-set vpn_transform esp-aes esp-sha256-hmac

```
=CONCAT('Common Info'!$C$2,
      'Common Info'!$C$3,
      B2, D2, C2, E2)
```

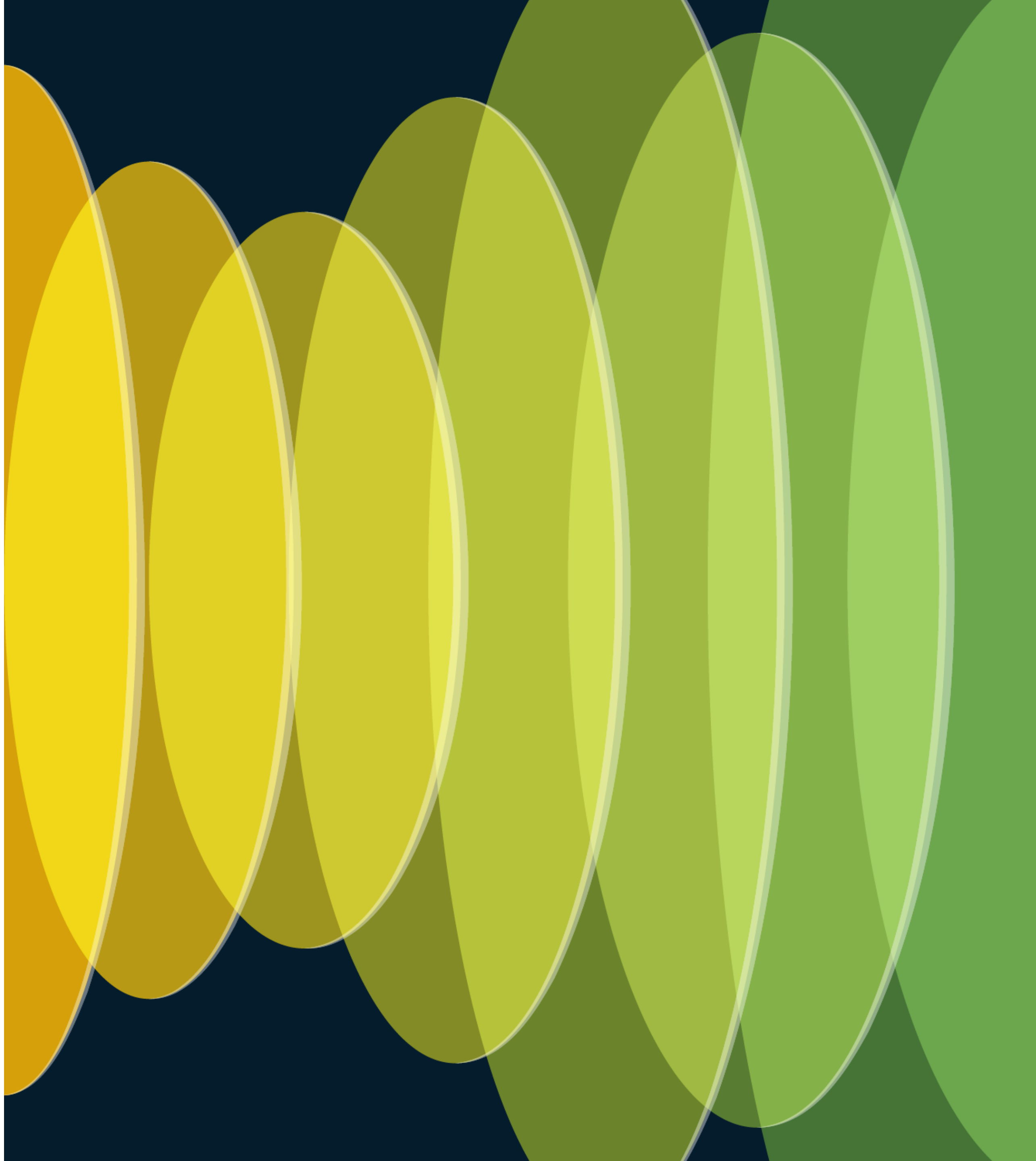
Per VPN Configurations



VPN Name	Site 1 Access List	Site 1 Crypto Map	Site 1 Key	Site 1 Interface	Site 1 Full Config
VPN_101	access-list 100 permit ip 10.98.128.0 0.0.3.255 172.28.11.0 0.0.0.255	crypto map VPN_101 10 ipsec-isakmp set peer 198.51.100.33 set transform-setvpn_transform match address 100	crypto isakmp key ZA788I address 198.51.100.33	interface GigabitEthernet0/1 crypto map VPN_101	crypto isakmp policy 10 encryption aes hash sha256 authentication pre-share group 14 crypto ipsec transform-set vpn_transform esp-aes esp-sha256-hmac access-list 100 permit ip 10.98.128.0 0.0.3.255 172.28.11.0 0.0.0.255 crypto isakmp key ZA788I address 198.51.100.33 crypto map VPN_101 10 ipsec-isakmp set peer 198.51.100.33 set transform-setvpn_transform match address 100 interface GigabitEthernet0/1 crypto map VPN_101

Tip! Don’t forget to add spaces and new lines where needed

Demo!



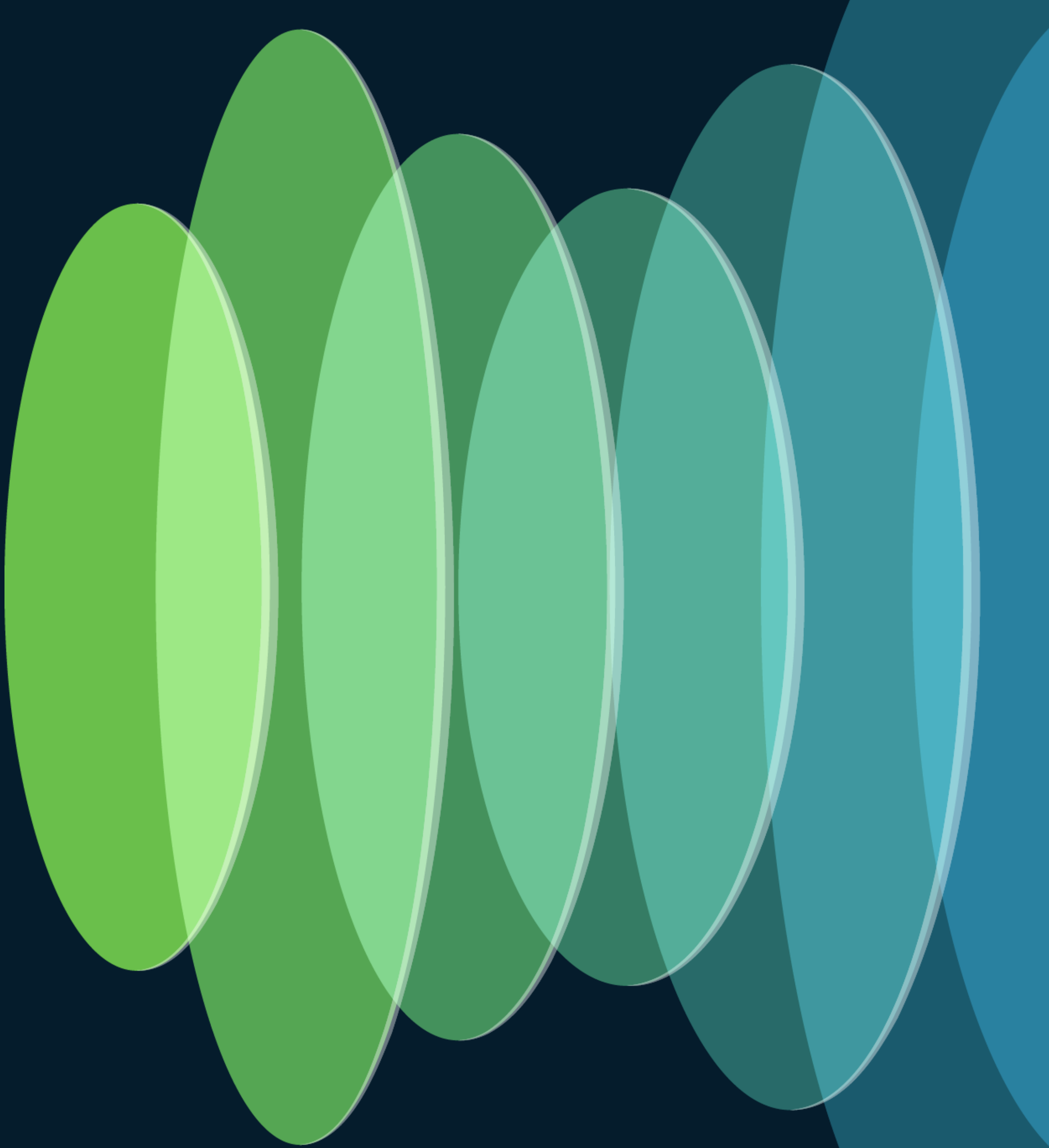
Other Useful Functions

- MID – Pull characters out of a string
- REPLACE – Change characters in a string
- FIND – Locate a character in a string
- TRIM – Remove white space
- RAND / RANDBETWEEN – Provide a random number
- Lots more!

Why should I care?

- Business runs on spreadsheets… not just networking
- Formulas are WAY faster and consistent than manual
  - Double check, then triple check results!
- But be careful you don't go too far…

Aw CRUD, we gotta talk a little about APIs



In the beginning... Humans were the only users

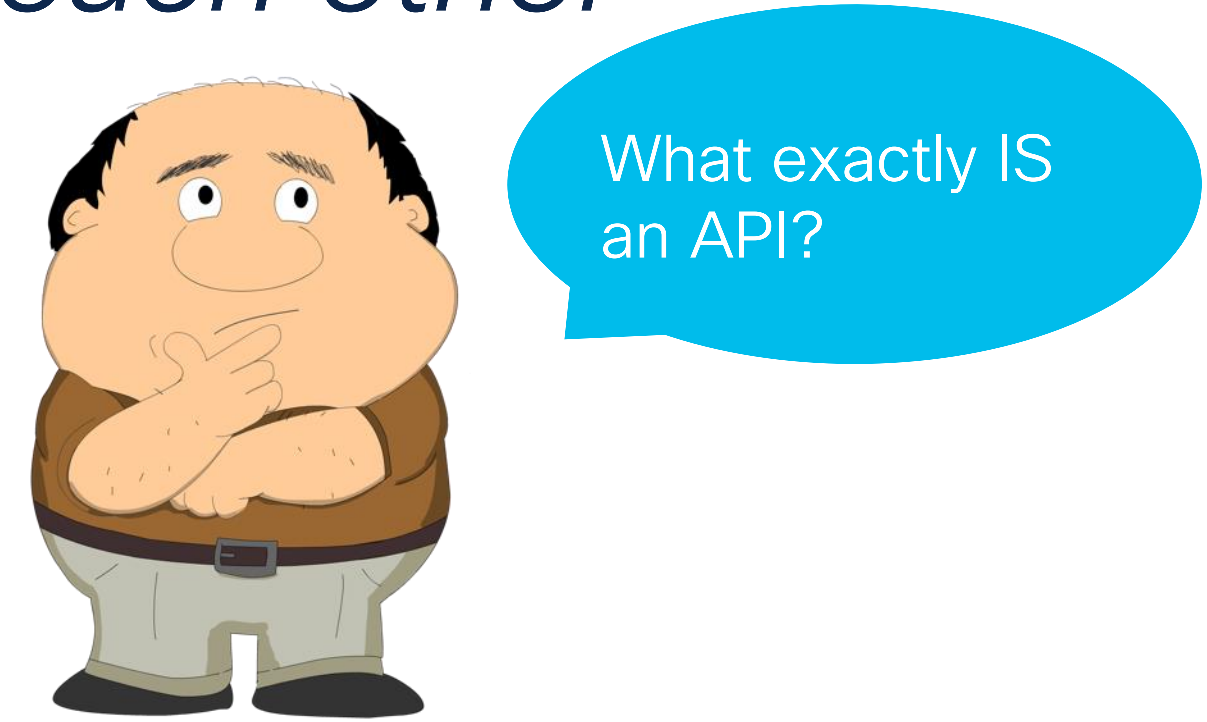
In the beginning… Humans were the only users

Software displays results in User Interface (UI)

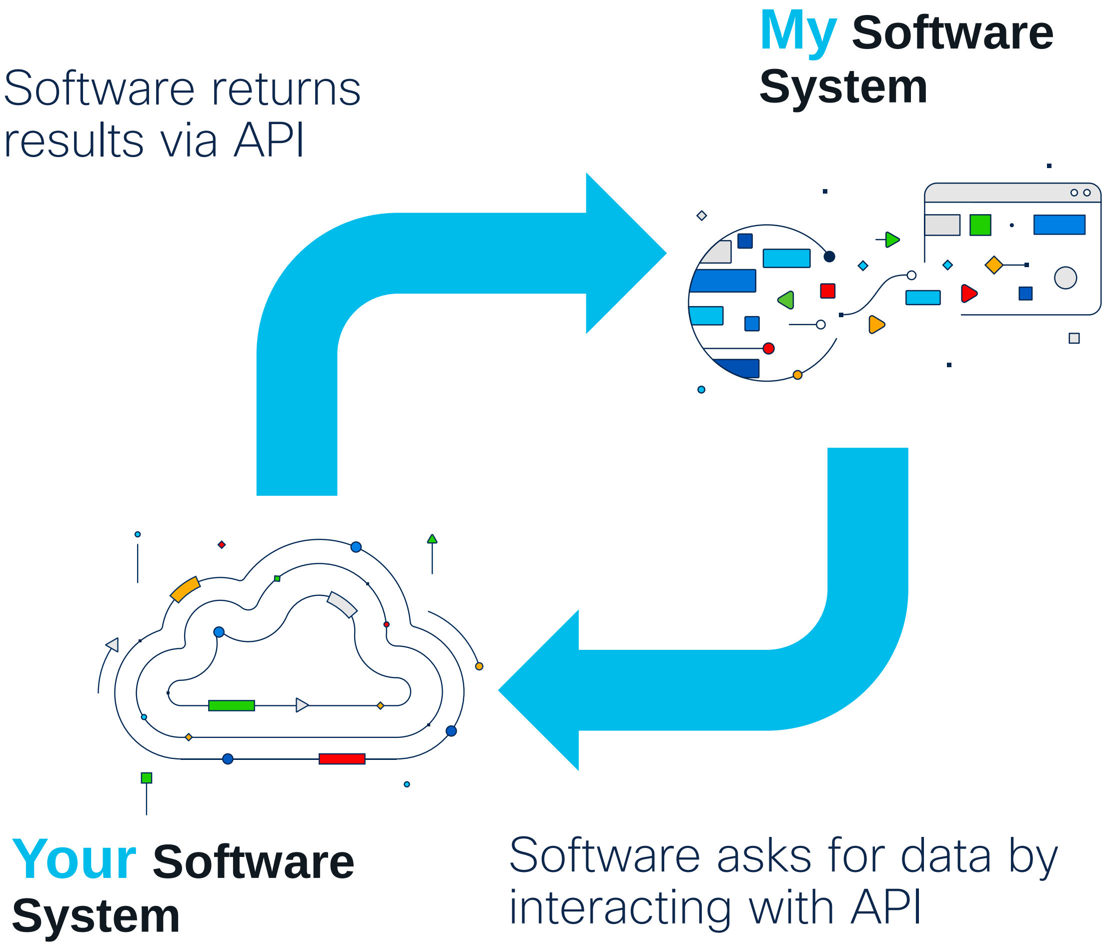
User asks for data or takes action by interacting with UI

cisco Live!

“It’s a way for two pieces of software to talk to each other”



Now software talks to software



Software returns results via API

My Software System

Your Software System

Software asks for data by interacting with API

cisco Live!

APIs aren’t scary... you already use them

Command Line Interface (CLI)

Designed for Humans... so more a UI than API

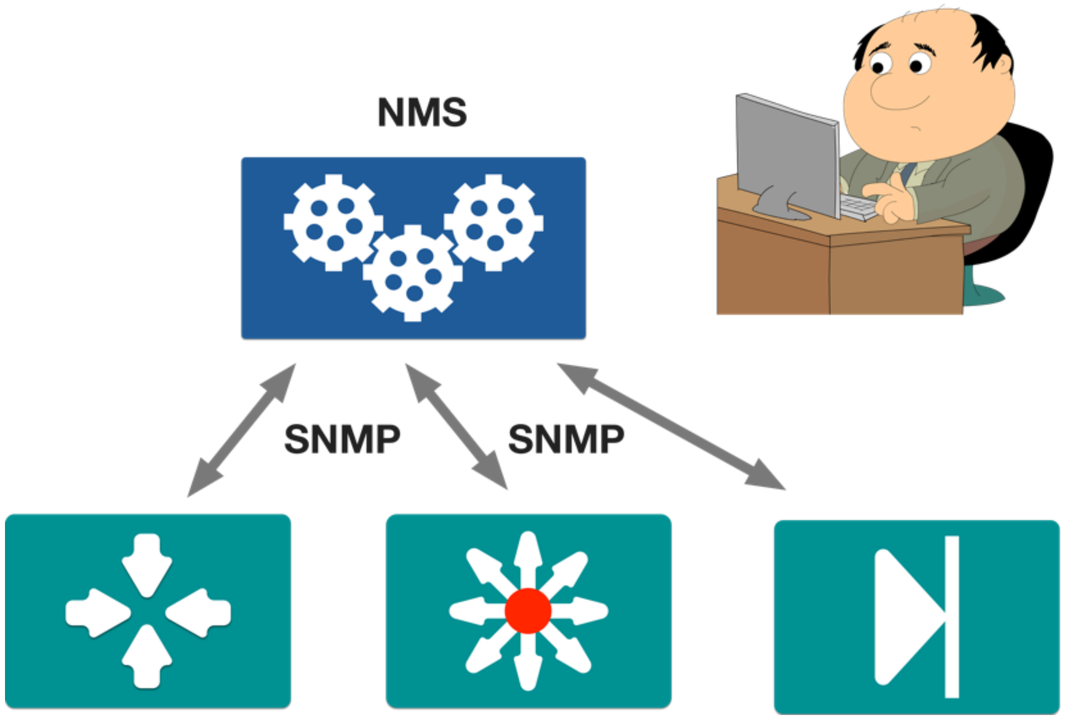
but...

- Network Management Systems
- Expect Scripts
- Paramiko/Netmiko
- NAPALM

```
#!/usr/bin/expect -f

send "conf t\n"
expect "(config)#"

send ”hostname my_switch\n"
expect "(config)#"
send "ntp server 10.10.10.101\n"
expect "(config)#"
send "ip domain-name domain.intra\n"
expect "(config)#"

send "end\n"
expect "#"
send "write mem\n"
expect "#"
```

Simple Network Management Protocol (SNMP)

> *“designed as a programmatic interface between management applications and devices”*



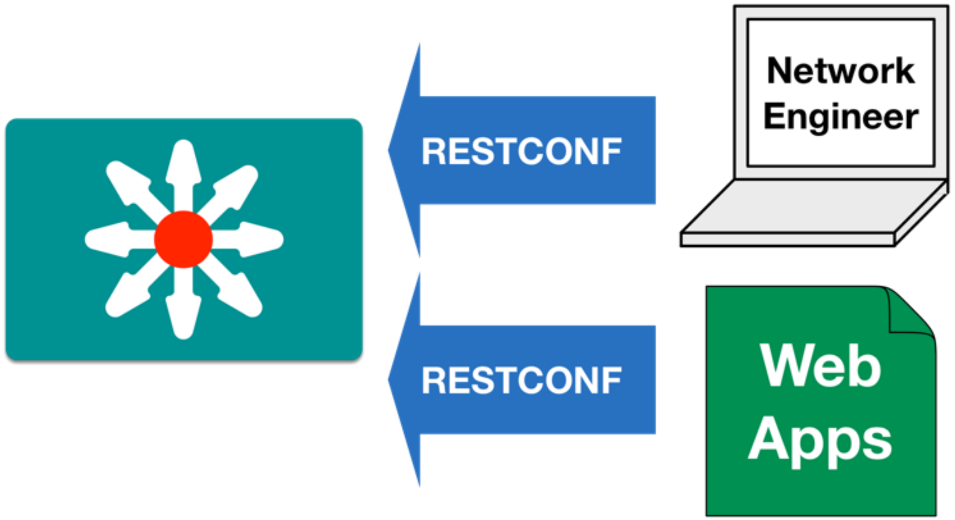
https://tools.ietf.org/html/rfc3535

Representational State Transfer (REST)

- API framework for simple web services
- Another use for the HTTP protocol
- Popular due to performance, scale, simplicity, and reliability

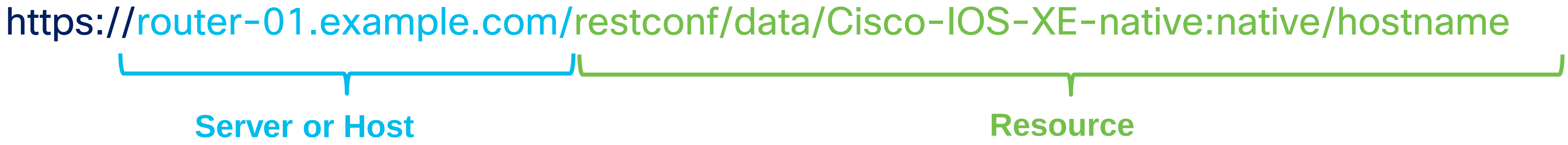
GET
POST
PUT
DELETE
{REST}

RESTCONF



Content	Configuration / Operational Data	`XML or JSON`
Operations	Actions to Take	`GET, POST, PUT, PATCH, DELETE`
Transport	TCP/IP Method	HTTP

The URI: What are you Requesting?



https://router-01.example.com/restconf/data/Cisco-IOS-XE-native:native/hostname

- Server or Host
  - Resolves to the IP and port to connect to
- Resource
  - The location of the data or object of interest on the server

Response

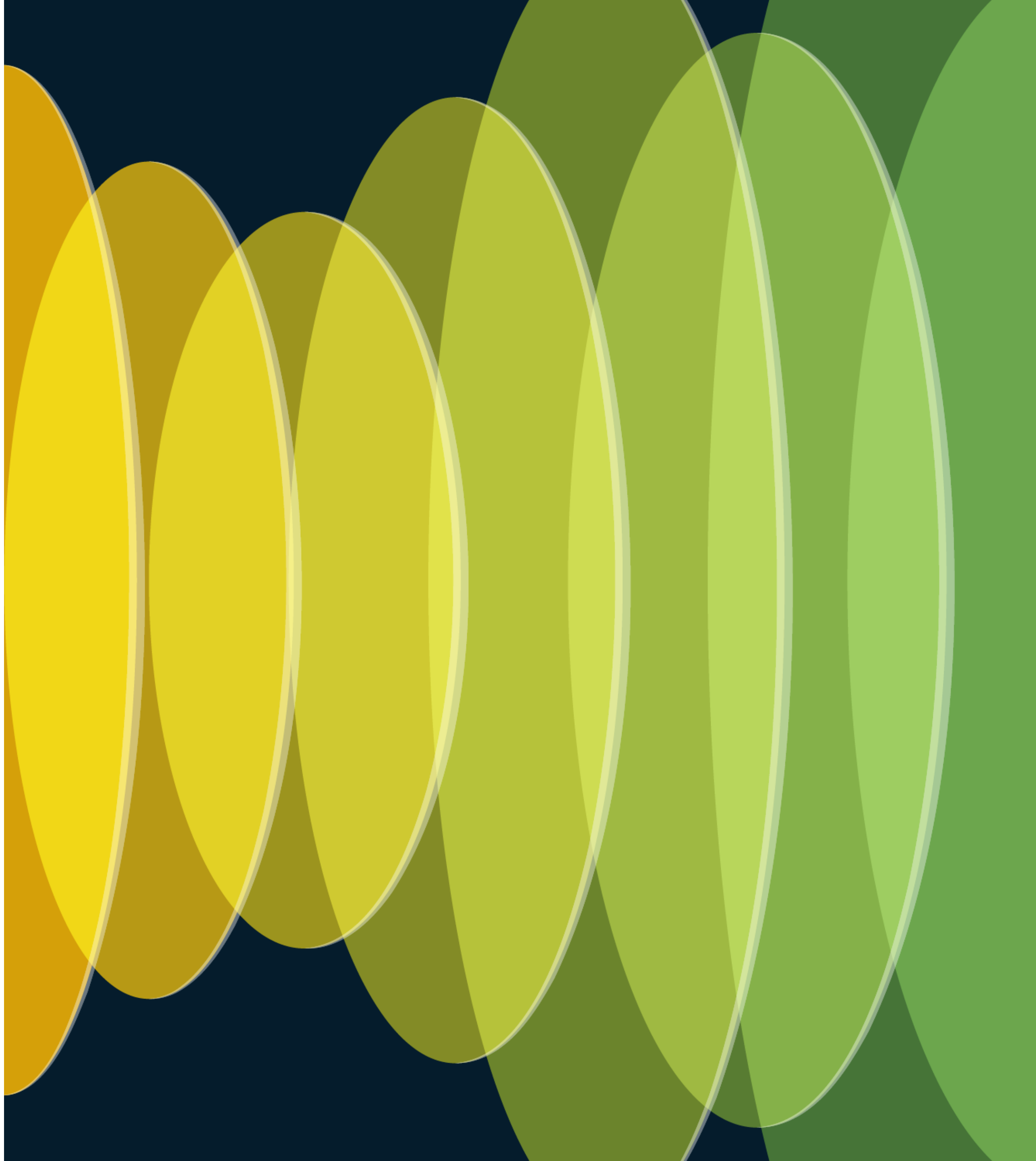
```
{
  "Cisco-IOS-XE-native:hostname":
     "csr1000v-1"
}
```

- JSON data returned
- Based on YANG model

HTTP Methods: What to do?

HTTP Verb	Typical Purpose (CRUD)	Description
POST	Create	Used to create a new object, or resource. *Example: Add new book to library*
GET	Read	Retrieve resource details from the system. *Example: Get list of books from the library*
PUT	Update	Typically used to replace or update a resource. Can be used to modify or create. *Example: Update the borrower details for a book*
DELETE	Delete	Remove a resource from the system. *Example: Delete a book from the library.*

Demo!



Why should I care?

- IT System Integrations
- Understand and work with automation peers
- Evaluate APIs from your products and vendors
- APIs are cool, add it to your resume!

Resources

Cisco U.

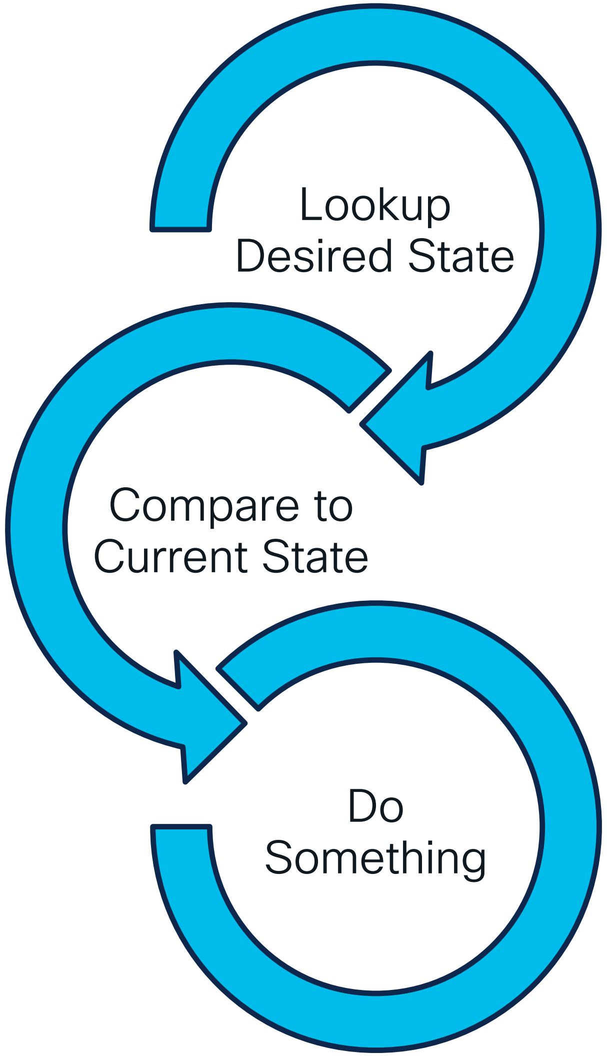
Free!

Understanding Network Automation Essentials Learning Path

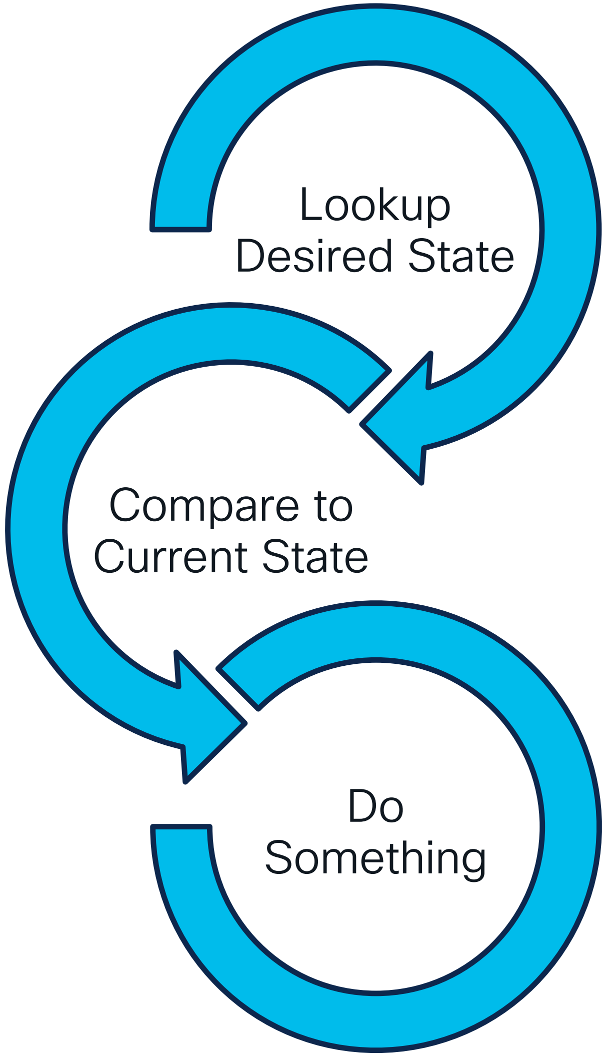
- Introduction to APIs
- NETCONF and RESTCONF

Source of Truth, it doesn’t have to be you anymore

How does automation work?



How does ~~automation~~ *network engineering* work?



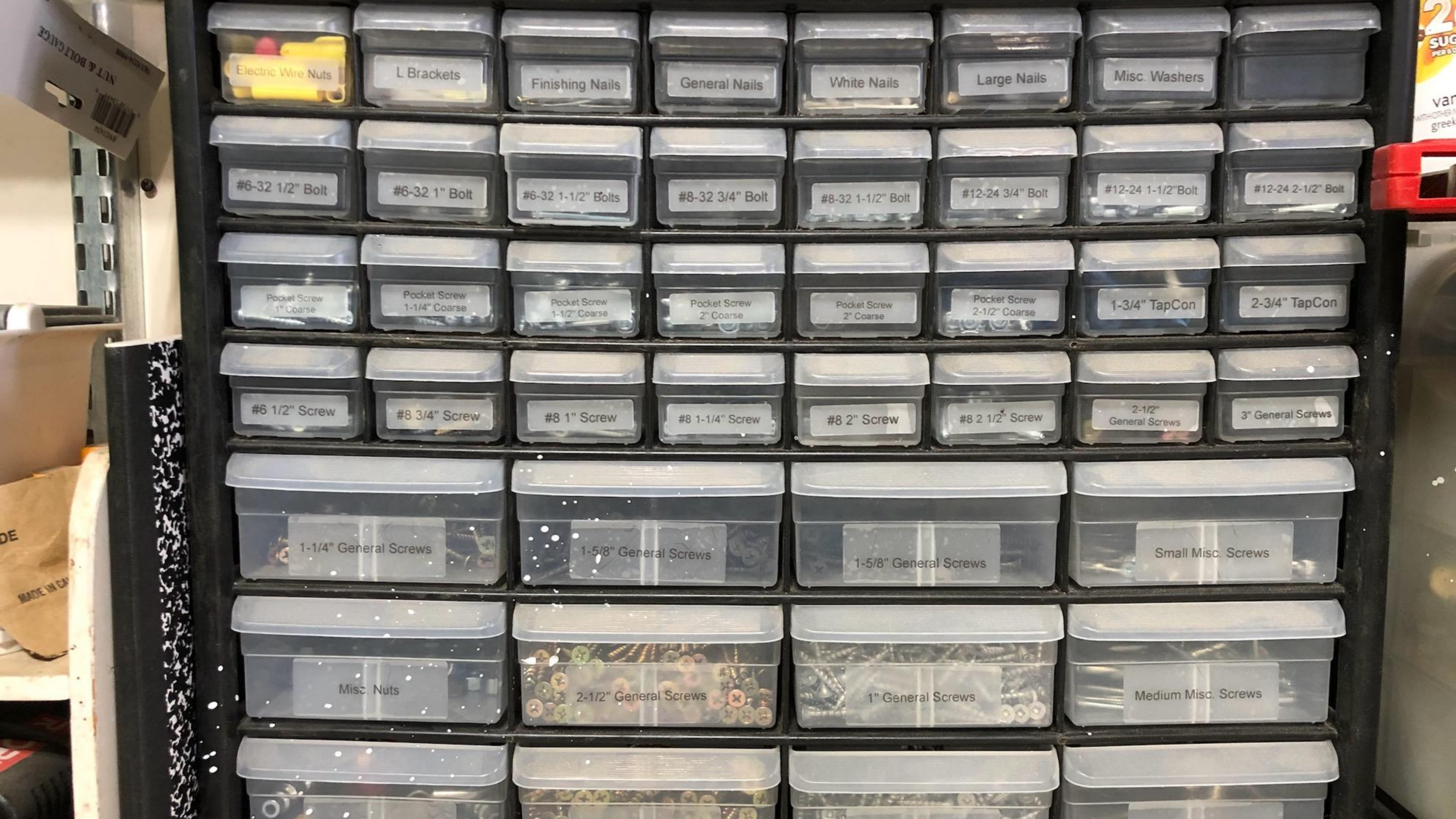
What types of data make up “Desired State”

- IP Addresses
- Interface settings
- VLANs
- Software Versions
- Credentials
- Serial / Asset Numbers
- Neighbors

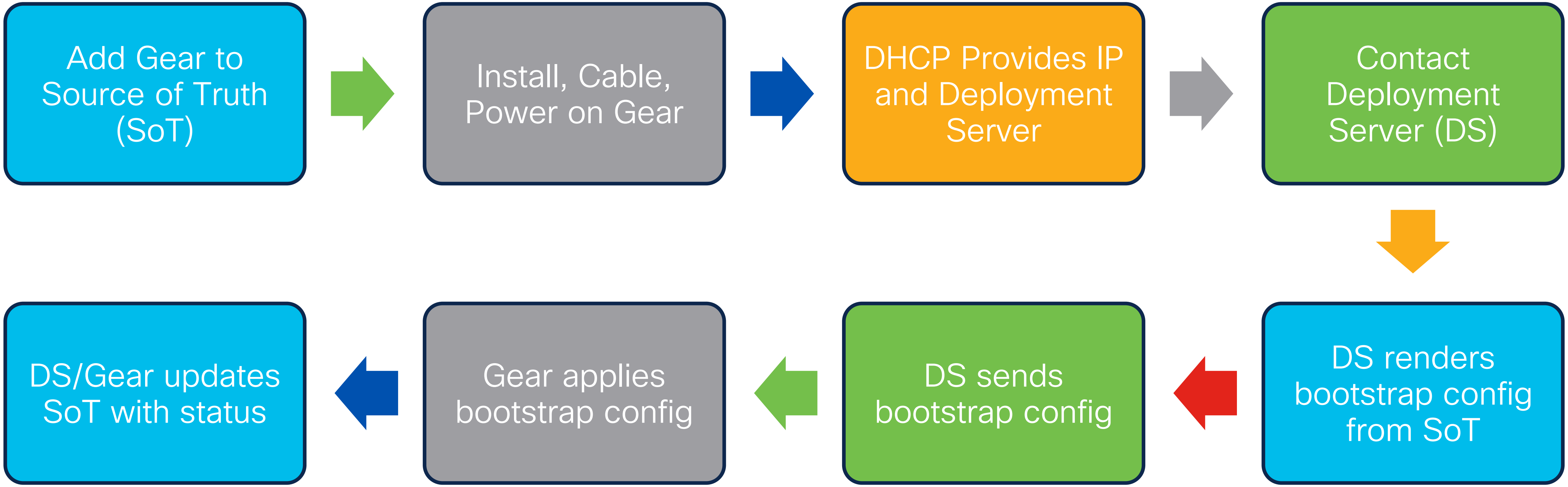
- Cables / Circuits
- Management Access
- Status
- Standards
- Application / Service Details
- Host Details

Where is the data found?

- In your head
- In someone else's head
- On a spreadsheet
- On a network diagram
- Text docs and notes apps
- Paper notes and scrap paper
- Within the network


NUT & BOLT GAUGE
Electric Wire Nuts
L Brackets
Finishing Nails
General Nails
White Nails
Large Nails
Misc. Washers
#6-32 1/2" Bolt
#6-32 1" Bolt
#6-32 1-1/2" Bolts
#8-32 3/4" Bolt
#8-32 1-1/2" Bolt
#12-24 3/4" Bolt
#12-24 1-1/2"Bolt
#12-24 2-1/2" Bolt
Pocket Screw 1" Coarse
Pocket Screw 1-1/4" Coarse
Pocket Screw 1-1/2" Coarse
Pocket Screw 2" Coarse
Pocket Screw 2" Coarse
Pocket Screw 2-1/2" Coarse
1-3/4" TapCon
2-3/4" TapCon
#6 1/2" Screw
#8 3/4" Screw
#8 1" Screw
#8 1-1/4" Screw
#8 2" Screw
#8 2 1/2" Screw
2-1/2" General Screws
3" General Screws
1-1/4" General Screws
1-5/8" General Screws
1-5/8" General Screws
Small Misc. Screws
Misc. Nuts
2-1/2" General Screws
1" General Screws
Medium Misc. Screws

Device Installation Workflow with Source of Truth



Add Gear to Source of Truth (SoT)

Install, Cable, Power on Gear

DHCP Provides IP and Deployment Server

Contact Deployment Server (DS)

DS renders bootstrap config from SoT

DS sends bootstrap config

Gear applies bootstrap config

DS/Gear updates SoT with status

Common Automation Source of Truth Options

- CSV Files
- YAML / JSON Files
- Git Repositories
- APIs into Other Applications
  - IPAM / DCIM
  - Secret Management
  - Service Desk / CDB

“Single” Source of Truth Realities

- “Single” is a fantasy
- Goal should be ”single per domain”
  - IPAM, DCIM, Secrets, etc
- It takes discipline to maintain
- Culture change – “Document First”
- Go in phases and celebrate success

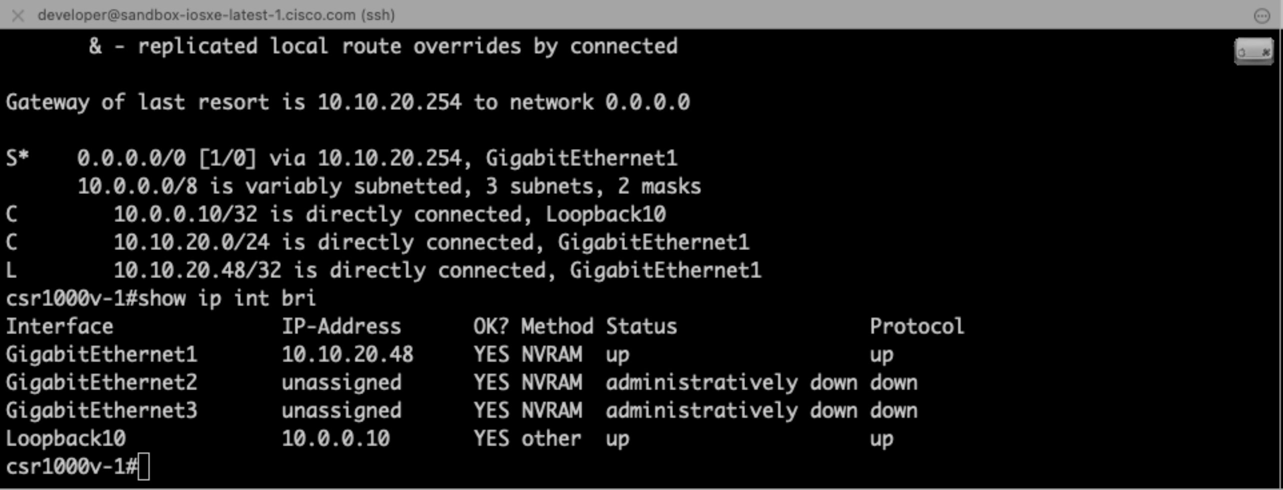
pyATS - How to no-code and automate networks

“Could you get me this info from all of the devices in the network?”

“Could you get me this info from all of the devices in the network?”

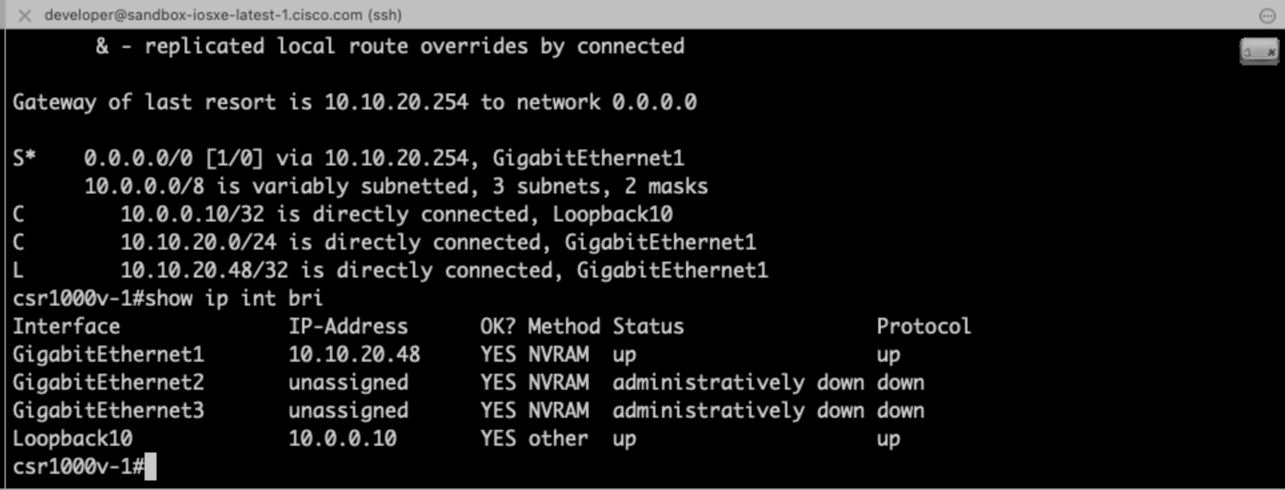
- Serial Numbers
- Interface status
- Routing table
- ARP / MAC tables
- Logs
- Etc
- Etc
- etc

We all have our cheats...



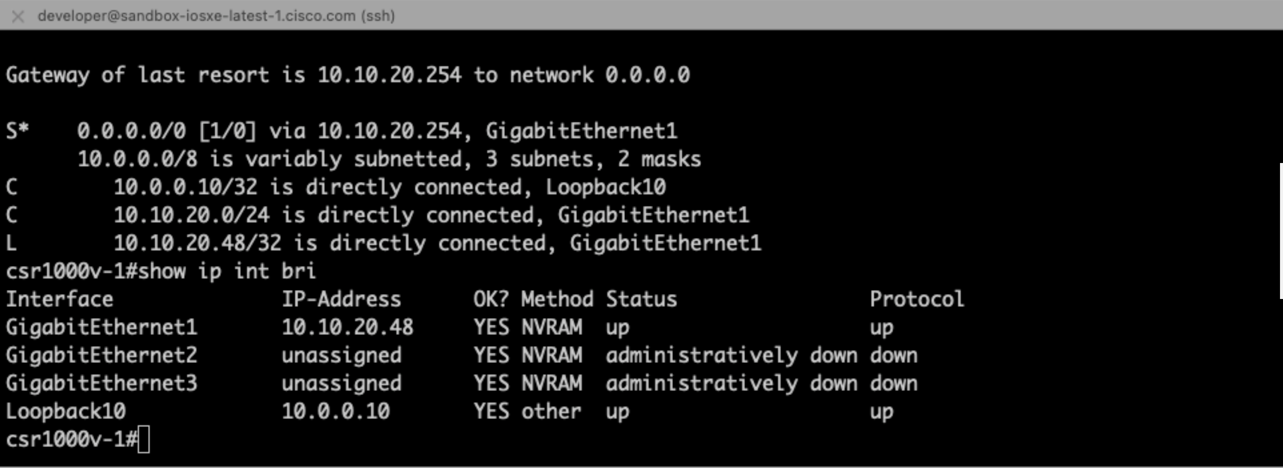
developer@sandbox-iosxe-latest-1.cisco.com (ssh)

```
      & - replicated local route overrides by connected

Gateway of last resort is 10.10.20.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.20.254, GigabitEthernet1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.10/32 is directly connected, Loopback10
C        10.10.20.0/24 is directly connected, GigabitEthernet1
L        10.10.20.48/32 is directly connected, GigabitEthernet1
csr1000v-1#show ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.10.20.48     YES NVRAM  up                    up
GigabitEthernet2       unassigned      YES NVRAM  administratively down down
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
Loopback10             10.0.0.10       YES other  up                    up
csr1000v-1#
```



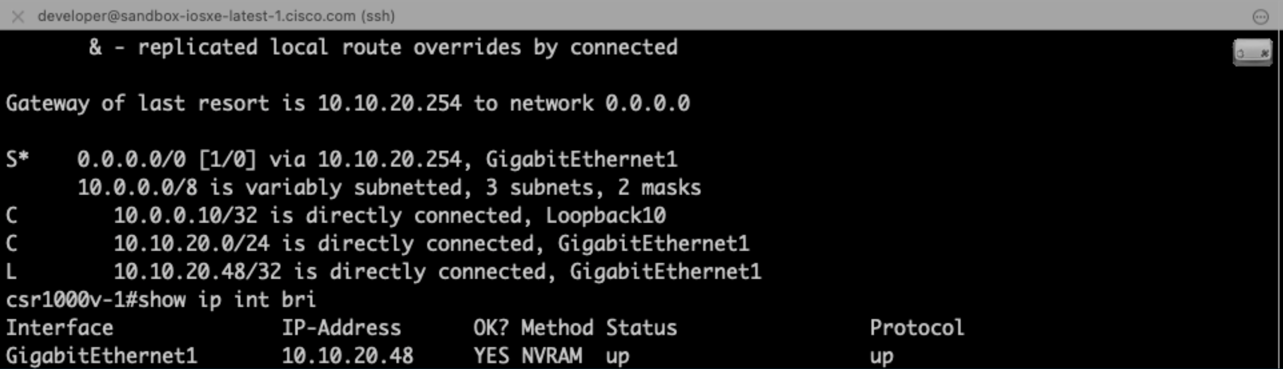
developer@sandbox-iosxe-latest-1.cisco.com (ssh)

```
      & - replicated local route overrides by connected

Gateway of last resort is 10.10.20.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.20.254, GigabitEthernet1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.10/32 is directly connected, Loopback10
C        10.10.20.0/24 is directly connected, GigabitEthernet1
L        10.10.20.48/32 is directly connected, GigabitEthernet1
csr1000v-1#show ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.10.20.48     YES NVRAM  up                    up
GigabitEthernet2       unassigned      YES NVRAM  administratively down down
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
Loopback10             10.0.0.10       YES other  up                    up
csr1000v-1#
```



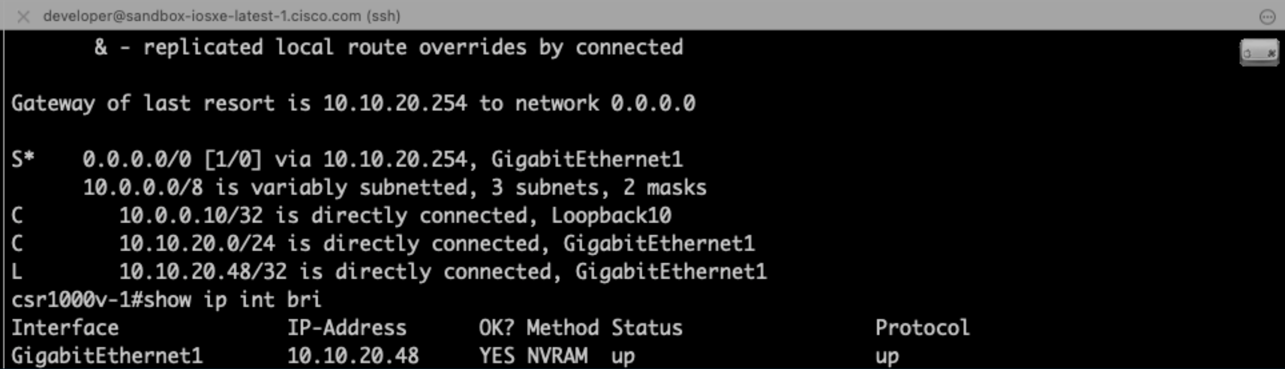
developer@sandbox-iosxe-latest-1.cisco.com (ssh)

```
Gateway of last resort is 10.10.20.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.20.254, GigabitEthernet1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.10/32 is directly connected, Loopback10
C        10.10.20.0/24 is directly connected, GigabitEthernet1
L        10.10.20.48/32 is directly connected, GigabitEthernet1
csr1000v-1#show ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.10.20.48     YES NVRAM  up                    up
GigabitEthernet2       unassigned      YES NVRAM  administratively down down
GigabitEthernet3       unassigned      YES NVRAM  administratively down down
Loopback10             10.0.0.10       YES other  up                    up
csr1000v-1#
```



developer@sandbox-iosxe-latest-1.cisco.com (ssh)

```
      & - replicated local route overrides by connected

Gateway of last resort is 10.10.20.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.20.254, GigabitEthernet1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.10/32 is directly connected, Loopback10
C        10.10.20.0/24 is directly connected, GigabitEthernet1
L        10.10.20.48/32 is directly connected, GigabitEthernet1
csr1000v-1#show ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.10.20.48     YES NVRAM  up                    up
```



developer@sandbox-iosxe-latest-1.cisco.com (ssh)

```
      & - replicated local route overrides by connected

Gateway of last resort is 10.10.20.254 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 10.10.20.254, GigabitEthernet1
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C        10.0.0.10/32 is directly connected, Loopback10
C        10.10.20.0/24 is directly connected, GigabitEthernet1
L        10.10.20.48/32 is directly connected, GigabitEthernet1
csr1000v-1#show ip int bri
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet1       10.10.20.48     YES NVRAM  up                    up
```

Per device results

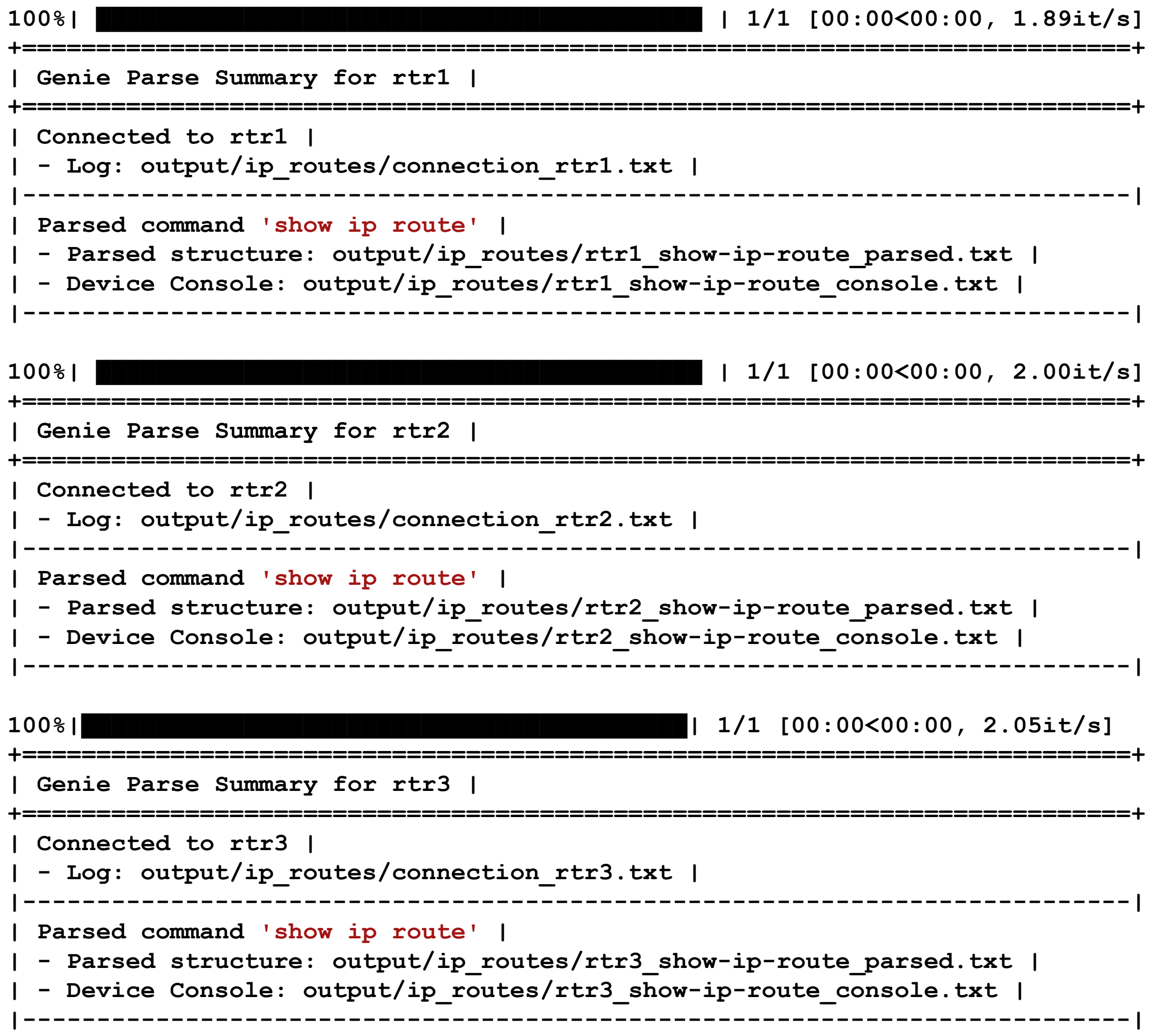
Command to Run

```
pyats parse "show ip route" \
  --testbed-file testbed.yaml \
  --output output/ip_routes
```

Network Inventory

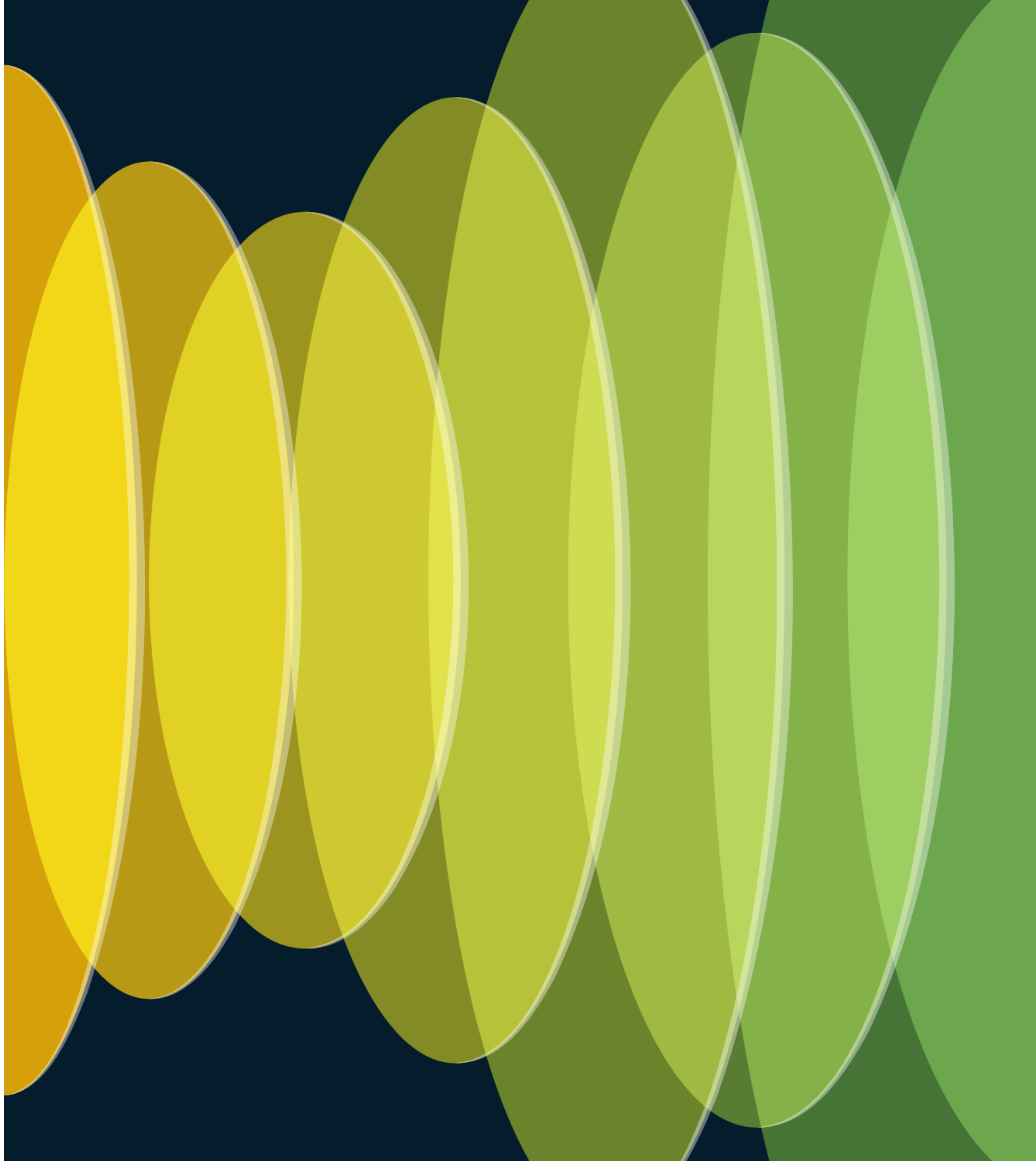
Captured Output

Per Device Results
- Parse = JSON Data
- Console = Raw Data



```
100%|████████████████████████████████████████| 1/1 [00:00<00:00, 1.89it/s]
+==============================================================================+
| Genie Parse Summary for rtr1 |
+==============================================================================+
| Connected to rtr1 |
| - Log: output/ip_routes/connection_rtr1.txt |
|------------------------------------------------------------------------------|
| Parsed command 'show ip route' |
| - Parsed structure: output/ip_routes/rtr1_show-ip-route_parsed.txt |
| - Device Console: output/ip_routes/rtr1_show-ip-route_console.txt |
|------------------------------------------------------------------------------|

100%|████████████████████████████████████████| 1/1 [00:00<00:00, 2.00it/s]
+==============================================================================+
| Genie Parse Summary for rtr2 |
+==============================================================================+
| Connected to rtr2 |
| - Log: output/ip_routes/connection_rtr2.txt |
|------------------------------------------------------------------------------|
| Parsed command 'show ip route' |
| - Parsed structure: output/ip_routes/rtr2_show-ip-route_parsed.txt |
| - Device Console: output/ip_routes/rtr2_show-ip-route_console.txt |
|------------------------------------------------------------------------------|

100%|████████████████████████████████████████| 1/1 [00:00<00:00, 2.05it/s]
+==============================================================================+
| Genie Parse Summary for rtr3 |
+==============================================================================+
| Connected to rtr3 |
| - Log: output/ip_routes/connection_rtr3.txt |
|------------------------------------------------------------------------------|
| Parsed command 'show ip route' |
| - Parsed structure: output/ip_routes/rtr3_show-ip-route_parsed.txt |
| - Device Console: output/ip_routes/rtr3_show-ip-route_console.txt |
|------------------------------------------------------------------------------|
```

Demo



pyATS can do much more

- Collect network baselines
- Quickly find differences
- Create robust network tests (does require coding)

Resources

Cisco U.

Free!

- Network Automation Testing with pyATS
- Test-Driven Automation with Cisco pyATS Using SSH
- pyATS home on DevNet
- pyATS CLI Hands On Lab

Ansible - The Hammer of Network Automation

Why Ansible?

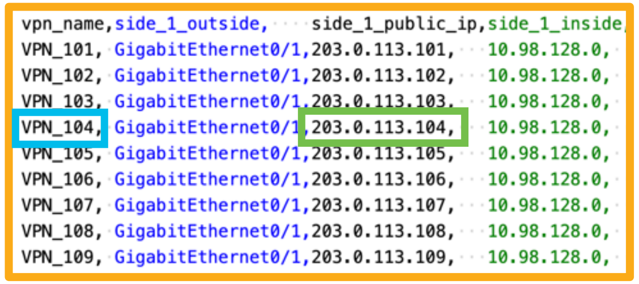
- OpenSource Infrastructure as Code tool
- Low barrier of entry into automation
- Coding skills not needed
- Very popular -> Lots of examples
- Broad network automation use cases

Ansible, it isn’t “code”… it’s just YAML ☺

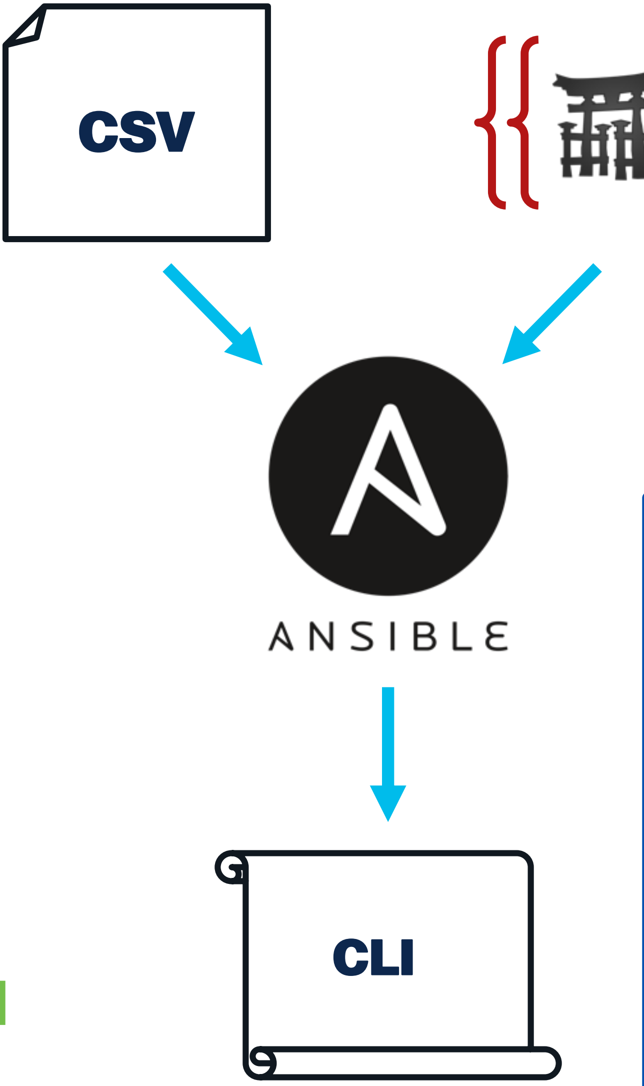
```
- name: Side 2 Config
  loop: "{{ vpns.list }}"
  loop_control:
    label: "{{ item.vpn_name }} Side 2"

  vars:
    vpn_name: "{{ item.vpn_name }}"
    preshared_key: "{{ item.preshared_key }} "
    source_network: "{{ item.side_2_inside }}"
    source_mask: "{{ item.side_2_mask }} "
    destination_network: "{{ item.side_1_inside }}"
    destination_mask: "{{ item.side_1_mask }}"
    tunnel_destination: "{{ item.side_1_public_ip }}"
    tunnel_interface: "{{ item.side_2_outside }} "

  ansible.builtin.template:
    src: ios-vpn.j2
    dest: configs/{{ item.vpn_name }}-side2-config.txt
```

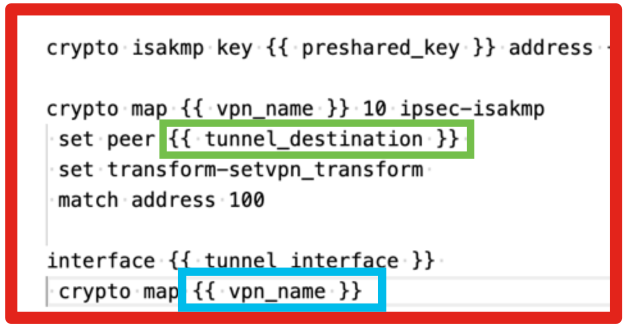
Another approach to the Spreadsheet Driven Automation

Maybe I’m ready to try something new...

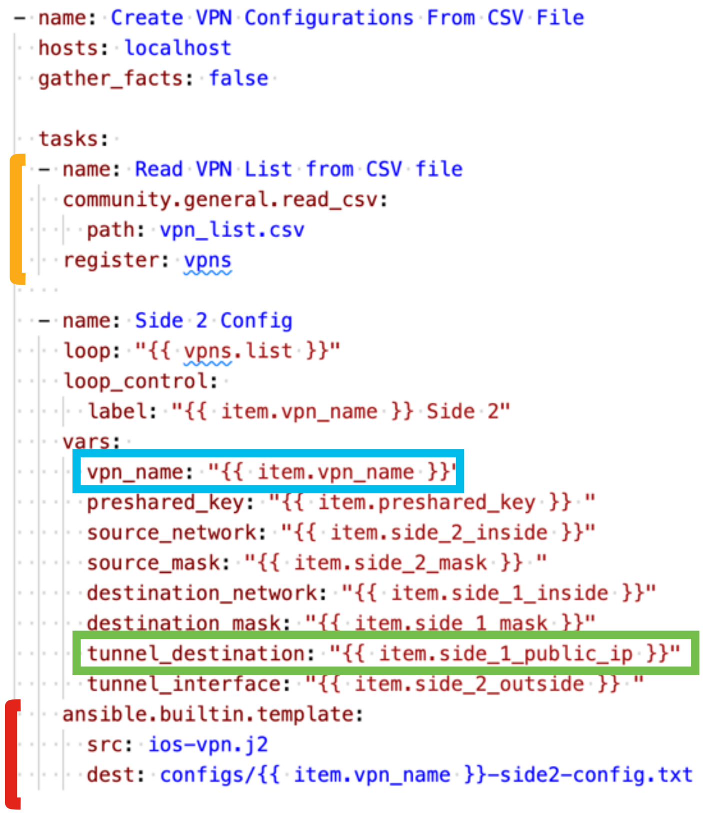


```
vpn_name,side_1_outside,    side_1_public_ip,side_1_inside,
VPN_101, GigabitEthernet0/1,203.0.113.101,   10.98.128.0,
VPN_102, GigabitEthernet0/1,203.0.113.102,   10.98.128.0,
VPN_103, GigabitEthernet0/1,203.0.113.103,   10.98.128.0,
VPN_104, GigabitEthernet0/1,203.0.113.104,   10.98.128.0,
VPN_105, GigabitEthernet0/1,203.0.113.105,   10.98.128.0,
VPN_106, GigabitEthernet0/1,203.0.113.106,   10.98.128.0,
VPN_107, GigabitEthernet0/1,203.0.113.107,   10.98.128.0,
VPN_108, GigabitEthernet0/1,203.0.113.108,   10.98.128.0,
VPN_109, GigabitEthernet0/1,203.0.113.109,   10.98.128.0,
```

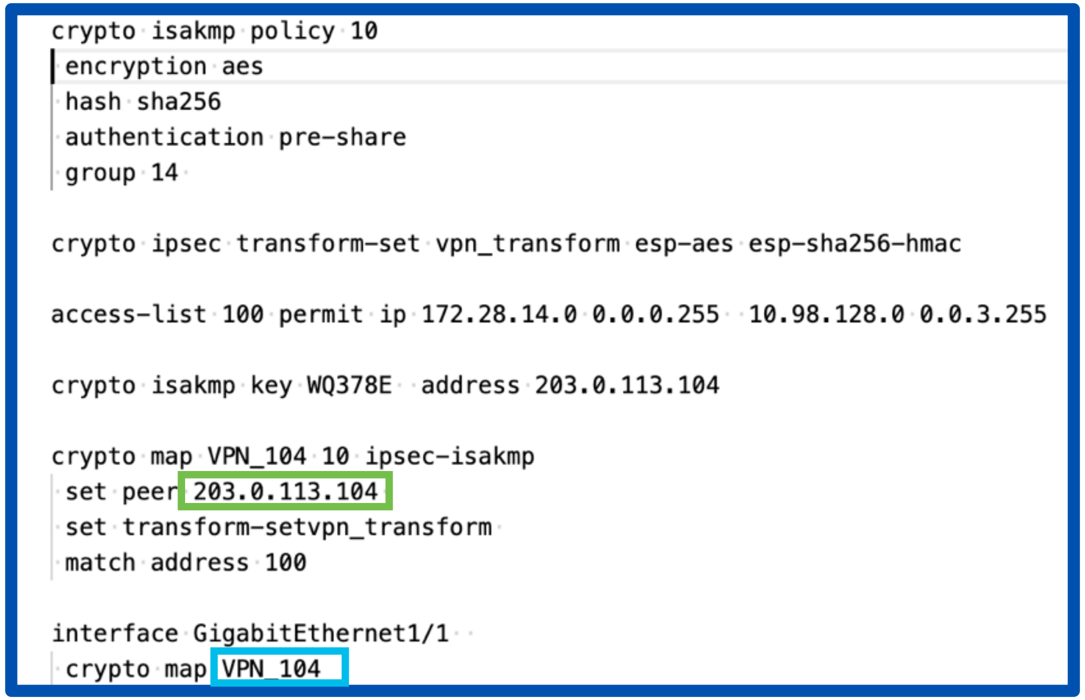




```
crypto isakmp key {{ preshared_key }} address

crypto map {{ vpn_name }} 10 ipsec-isakmp
 set peer {{ tunnel_destination }}
 set transform-setvpn_transform
 match address 100

interface {{ tunnel_interface }}
 crypto map {{ vpn_name }}
```

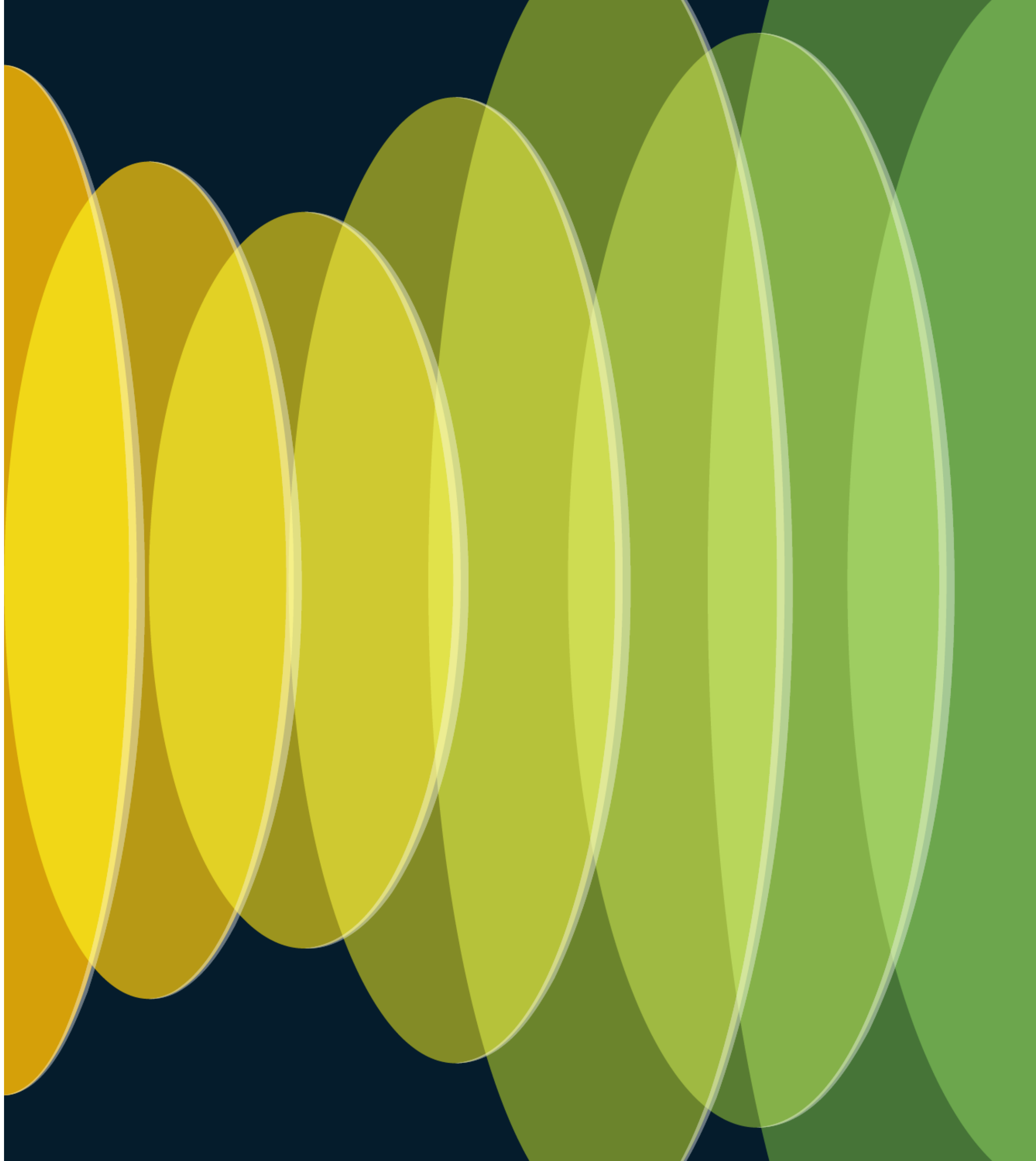


```
- name: Create VPN Configurations From CSV File
  hosts: localhost
  gather_facts: false

  tasks:
  - name: Read VPN List from CSV file
    community.general.read_csv:
      path: vpn_list.csv
    register: vpns

  - name: Side 2 Config
    loop: "{{ vpns.list }}"
    loop_control:
      label: "{{ item.vpn_name }} Side 2"
    vars:
      vpn_name: "{{ item.vpn_name }}"
      preshared_key: "{{ item.preshared_key }} "
      source_network: "{{ item.side_2_inside }}"
      source_mask: "{{ item.side_2_mask }} "
      destination_network: "{{ item.side_1_inside }}"
      destination_mask: "{{ item.side_1_mask }}"
      tunnel_destination: "{{ item.side_1_public_ip }}"
      tunnel_interface: "{{ item.side_2_outside }} "
    ansible.builtin.template:
      src: ios-vpn.j2
      dest: configs/{{ item.vpn_name }}-side2-config.txt
```



```
crypto isakmp policy 10
 encryption aes
 hash sha256
 authentication pre-share
 group 14

crypto ipsec transform-set vpn_transform esp-aes esp-sha256-hmac

access-list 100 permit ip 172.28.14.0 0.0.0.255  10.98.128.0 0.0.3.255

crypto isakmp key WQ378E  address 203.0.113.104

crypto map VPN_104 10 ipsec-isakmp
 set peer 203.0.113.104
 set transform-setvpn_transform
 match address 100

interface GigabitEthernet1/1
 crypto map VPN_104
```

Demo!



Ansible can do much more

- Apply configurations directly to devices
- Complex workflows
- Manage network, compute, storage, cloud, etc at once

Resources
(Training)

Cisco U.

Free!

- Network Automation with Ansible
- An Introduction to Ansible Roles
- Configuring IOS XE with Ansible

- Demo Example Code

Resources
(Documentation)

- Ansible on DevNet
- Ansible Core Docs
  - Reading from CSV
  - Using Jinja Templates
- Jinja Template Documentation

In closing...
Programmability... Nothing to be afraid of!

```
(std) → Downloads ./execute.py
Execute Command? yes or no: yes
```

What did we cover?

- Spreadsheet driven automation
- Aw CRUD, we gotta talk a little about APIs
- Source of Truth, it doesn’t have to be you anymore
- pyATS - How to no-code and automate networks
- Ansible the Hammer of Network Automation

Hank Preston

- Stay in touch:
  - Webex/Email: hapresto@cisco.com
  - Blogs: https://blogs.cisco.com/author/hankpreston

Complete Your Session Evaluations

Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to **win 1 of 5 full conference passes** to Cisco Live 2025.

Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.

Level up and earn **exclusive prizes!**

Complete your surveys in the **Cisco Live mobile app.**

Continue your education

- Visit the Cisco Showcase for related demos
- Book your one-on-one Meet the Engineer meeting
- Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs
- Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

cisco The bridge to possible

Thank you

cisco Live!

#CiscoLive