



You make **possible**



Network Assurance

Cisco pyATS/Genie for Network Engineers

Siming Yuan, Technical Leader, Engineering, @simingy

DEVNET-1204



Barcelona | January 27-31, 2020



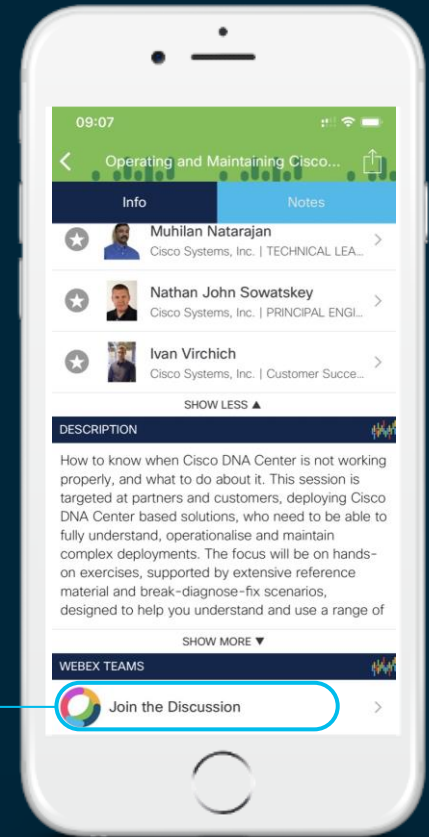
Cisco Webex Teams

Questions?

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

How

- 1 Find this session in the Cisco Events 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



Agenda

- Introduction to NetDevOps: Automation & Validation
- What is the Cisco pyATS | Genie, how it can help you
- Use Case Analysis & Examples
 - Network profiling
 - Stateful validation & certification testing
 - Chaos Monkey: preemptively catch potential network issues
- Demo
 - Shell/Command-line interface [no programming!]
 - RobotFramework – English-like scripting
 - Integration with pyATS Dashboard: XPRESSO
- Endless Possibilities: What's Next?

Network Engineering: Modern Age

- Software Defined [Network, Access, WAN]
- YANG: NETCONF, RESTCONF, gRPC
- Network Function Virtualization (NFV)
- Cloud

Automation is no longer a luxury,
It is now a *NECESSITY*.



CLI & Text

```
show interfaces
show vrf detail
show ip interface
show ipv6 interface
show interface switchport
show etherchannel summary
show interfaces [intf] accounting
```

CLI & Text

```
show interfaces
sh show ip protocols
sh show ip ospf
sh show ip ospf mpls ldp interface
sh show ip ospf mpls trafficeng link
sh show ip ospf virtuellinks
sh show ip ospf shamlinks
show ip ospf interface
show ip ospf database topology
show ip ospf database router
show ip ospf database network
show ip ospf database summary
show ip ospf database external
show ip ospf database opaquearea
show ip ospf database opaqueas
show ip ospf database opaquelink
show ip ospf neighbor detail
```


CLI & Text

```
show interfaces
show ip protocols
show ip ospf
show bgp all detail
show bgp all neighbor
show bgp all summary
show bgp all clusterids
show bgp all
show ip bgp template peersession <WORD>
show ip bgp template peerpolicy <WORD>
show ip bgp all dampening parameters
show ip bgp <af_name> [ vrf <vrf_id> ] <ipv4prefix>
show bgp vrf [vrf_id] <af_name> <ipv6prefix>
show bgp <af_name> <ipv6prefix>
show bgp all neighbors <neighbor> policy
show ip route bgp
show ip route vrf <WORD> bgp
show ipv6 route bgp
show ipv6 route vrf <WORD> bgp
show vrf detail
```

CLI & Text

```
show interfaces
```

```
sh show ip protocols
```

```
sh show ip ospf
```

```
sh show show bgp all detail
```

```
sh show show bgp all neighbor
```

```
sh show show bgp all summary
```

```
sh show show bgp all clusterids
```

```
show show bgp all
```

```
show show ip bgp template peersession <WORD>
```

```
show show ip bgp template peerpolicy <WORD>
```

```
show show ip bgp all dampening parameters
```

```
show show ip bgp <af_name> [ vrf <vrf_id> ] <ipv4pre
```

```
show show bgp vrf [vrf_id] <af_name> <ipv6prefix>
```

```
show show bgp <af_name> <ipv6prefix>
```

```
show show bgp all neighbors <neighbor> policy
```

```
show show ip route bgp
```

```
show show ip route vrf <WORD> bgp
```

```
show show ipv6 route bgp
```

```
show ipv6 route vrf <WORD> bgp
```

```
show vrf detail
```

```
N93_2# show interface
Ethernet1/1 is up
admin state is up, Dedicated Interface
  Hardware: 100/1000/10000 Ethernet, address: 5e00.8003.0007 (bia 5e00.8003.0008)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, medium is broadcast
  full-duplex, 1000 Mb/s
  Beacon is turned off
  Auto-Negotiation is turned on FEC mode is Auto
  Input flow-control is off, output flow-control is off
  Auto-mdix is turned off
  Switchport monitor is off
  EtherType is 0x8100
  EEE (efficient-ethernet) : n/a
  Last link flapped 01:52:50
  Last clearing of "show interface" counters never
  2 interface resets
  Load-Interval #1: 30 seconds
    30 seconds input rate 0 bits/sec, 0 packets/sec
    30 seconds output rate 0 bits/sec, 0 packets/sec
    input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  Load-Interval #2: 5 minute (300 seconds)
    300 seconds input rate 0 bits/sec, 0 packets/sec
    300 seconds output rate 0 bits/sec, 0 packets/sec
    input rate 0 bps, 0 pps; output rate 0 bps, 0 pps
  RX
    0 unicast packets  0 multicast packets  0 broadcast packets
    0 input packets  0 bytes
    0 jumbo packets  0 storm suppression packets
    0 runs  0 giants  0 CRC  0 no buffer
    0 input error  0 short frame  0 overrun  0 underrun  0 ignored
    0 watchdog  0 bad etype drop  0 bad proto drop  0 if down drop
    0 input with dribble  0 input discard
    0 Rx pause
```

CLI & Text

```
show interfaces
```

```
show ip protocols
```

```
show ip ospf
```

```
show bgp all detail
```

```
show bgp all neighbor
```

```
show bgp all summary
```

```
show bgp all clusterids
```

```
show bgp all
```

```
show ip bgp template peersession <WORD>
```

```
show ip bgp template peerpolicy <WORD>
```

```
show ip bgp all dampening parameters
```

```
show ip bgp <af_name> [ vrf <vrf_id> ] <ipv4pre
```

```
show bgp vrf [vrf_id] <af_name> <ipv6prefix>
```

```
show bgp <af_name> <ipv6prefix>
```

```
show bgp all neighbors <neighbor> policy
```

```
show ip route bgp
```

```
show ip route vrf <WORD> bgp
```

```
show ipv6 route bgp
```

```
show ipv6 route vrf <WORD> bgp
```

```
show vrf detail
```

```
N93_2# show interface
```

```
Etherne
```

```
admin : N93_2# show bgp vrf all all
```

```
BGP routing table information for VRF default, address family IPv4 Unicast
```

```
BGP table version is 24, Local Router ID is 210.1.1.1
```

```
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
```

```
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
Input	*>e83.0.0.0/16	100.1.1.1			0	100 i
Auto-	*>i83.0.0.0/24	110.1.1.1	0	100	0	300 ?
Switc	*>i83.0.1.0/24	110.1.1.1	0	100	0	300 ?
Ether	*>i83.0.2.0/24	110.1.1.1	0	100	0	300 ?
EEE	*>i83.0.3.0/24	110.1.1.1	0	100	0	300 ?
Last	*>i83.0.4.0/24	110.1.1.1	0	100	0	300 ?
Last	*>e84.0.0.0/24	100.1.1.1			0	100 101 300 ?
2 in	*>e84.0.1.0/24	100.1.1.1			0	100 101 300 ?
Load-	*>e84.0.2.0/24	100.1.1.1			0	100 101 300 ?
30	*>e84.0.3.0/24	100.1.1.1			0	100 101 300 ?
30	*>e84.0.4.0/24	100.1.1.1			0	100 101 300 ?
in	*>e85.0.0.0/24	100.1.1.1	0		0	100 ?
Load-	*>i88.0.0.0/24	110.1.1.1	0	100	0	300 ?
30	*>i88.0.1.0/24	110.1.1.1	0	100	0	300 ?
30	*>i88.0.2.0/24	110.1.1.1	0	100	0	300 ?
in	*>i88.0.3.0/24	110.1.1.1	0	100	0	300 ?
RX	*>i88.0.4.0/24	110.1.1.1	0	100	0	300 ?
0	*>e100.1.1.1/32	100.1.1.1			0	100 101 300 ?
0	*>i110.1.1.1/32	110.1.1.1	0	100	0	300 ?

```
BGP routing table information for VRF default, address family IPv6 Unicast
```

```
BGP table version is 17, Local Router ID is 210.1.1.1
```

```
Status: s-suppressed, x-deleted, S-stale, d-dampened, h-history, *-valid, >-best
```

```
Path type: i-internal, e-external, c-confed, l-local, a-aggregate, r-redist, I-injected
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete, | - multipath, & - backup
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
	*>i83::/112	110:1::1:1	0	100	0	300 ?

CLI & Text

```
show interfaces
```

```
show ip protocols
```

```
show ip ospf
```

```
show bgp all detail
```

```
show bgp all neighbor
```

```
show bgp all summary
```

```
show bgp all clusterids
```

```
show bgp all
```

```
show ip bgp template peersession <WORD>
```

```
show ip bgp template peerpolicy <WORD>
```

```
show ip bgp all dampening parameters
```

```
show ip bgp <af_name> [ vrf <vrf_id> ] <ipv4pre>
```

```
show bgp vrf [vrf_id] <af_name> <ipv6prefix>
```

```
show bgp <af_name> <ipv6prefix>
```

```
show bgp all neighbors <neighbor> policy
```

```
show ip route bgp
```

```
show ip route vrf <WORD> bgp
```

```
show ipv6 route bgp
```

```
show ipv6 route vrf <WORD> bgp
```

```
show vrf detail
```

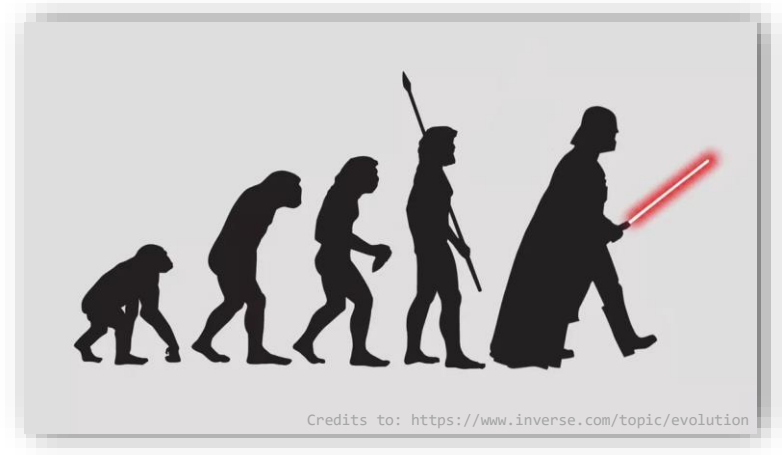
```
N93_2# show interface
Ethernet1/1.1, Interface status: protocol-up/link-up/admin-up, iof: 134,
IP address: 201.0.11.2, IP subnet: 201.0.11.0/24 route-preference: 0, tag: 0
IP broadcast address: 255.255.255.255
IP multicast groups locally joined:
  224.0.0.6  224.0.0.5  224.0.0.2  224.0.0.1  224.0.1.40  224.0.1.39
  224.0.0.13
IP MTU: 1500 bytes (using link MTU)
IP primary address route-preference: 0, tag: 0
IP proxy ARP : disabled
IP Local Proxy ARP : disabled
IP multicast routing: enabled
IP icmp redirects: enabled
IP directed-broadcast: disabled
IP Forwarding: disabled
IP icmp unreachable (except port): disabled
IP icmp port-unreachable: enabled
IP unicast reverse path forwarding: none
IP load sharing: none
IP interface statistics last reset: never
IP interface software stats: (sent/received/forwarded/originated/consumed)
  Unicast packets   : 238/357/232/6/12
  Unicast bytes     : 17220/24461/16592/628/1328
  Multicast packets  : 1298/1518/0/1298/2975
  Multicast bytes    : 110456/95636/0/110456/94466
  Broadcast packets  : 0/0/0/0/0
  Broadcast bytes    : 0/0/0/0/0
  Labeled packets    : 0/0/0/0/0
  Labeled bytes      : 0/0/0/0/0
WCCP Redirect outbound: disabled
WCCP Redirect inbound: disabled
WCCP Redirect exclude: disabled
Ethernet1/1.2, Interface status: protocol-up/link-up/admin-up, iof: 135,
IP address: 201.1.11.2, IP subnet: 201.1.11.0/24 route-preference: 0, tag: 0
IP broadcast address: 255.255.255.255
IP multicast groups locally joined: none
IP MTU: 1500 bytes (using link MTU)
IP primary address route-preference: 0, tag: 0
IP proxy ARP : disabled
IP Local Proxy ARP : disabled
IP multicast routing: disabled
```

To Err is Human ...

```
2 show interface
3 mgmt0 is up
4 admin state is up,
5 Hardware: Ethernet, address: 5e00.8003.0000 (bia 5e00.8003.0000)
6 Internet Address is 172.16.1.82/24
7 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
8 reliability 255/255, txload 1/255, rxload 1/255
9 Encapsulation ARPA, medium is broadcast
10 full-duplex, 1000 Mb/s
11 Auto-Negotiation is turned on
12 Auto-mdix is turned off
13 EtherType is 0x0000
14 1 minute input rate 1425896 bits/sec, 1347 packets/sec
15 1 minute output rate 144 bits/sec, 0 packets/sec
16 Rx
17 15153308 input packets 2314 unicast packets 15137206 multicast packets
18 13788 broadcast packets 2010655640 bytes
19 Tx
20 2582 output packets 2315 unicast packets 258 multicast packets
21 9 broadcast packets 296264 bytes
22
23 Ethernet1/1 is up
24 admin state is up, Dedicated Interface
25 Hardware: 100/1000/10000 Ethernet, address: 5e00.8003.0007 (bia 5e00.8003.0008)
26 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
27 reliability 255/255, txload 1/255, rxload 1/255
28 Encapsulation ARPA, medium is broadcast
29 full-duplex, 1000 Mb/s
30 Beacon is turned off
31 Auto-Negotiation is turned on FEC mode is Auto
32 Input flow-control is off, output flow-control is off
33 Auto-mdix is turned off
34 Switchport monitor is off
35 EtherType is 0x8100
36 EEE (efficient-ethernet) : n/a
37 Last link flapped 04:14:34
38 Last clearing of "show interface" counters never
39 2 interface resets
```

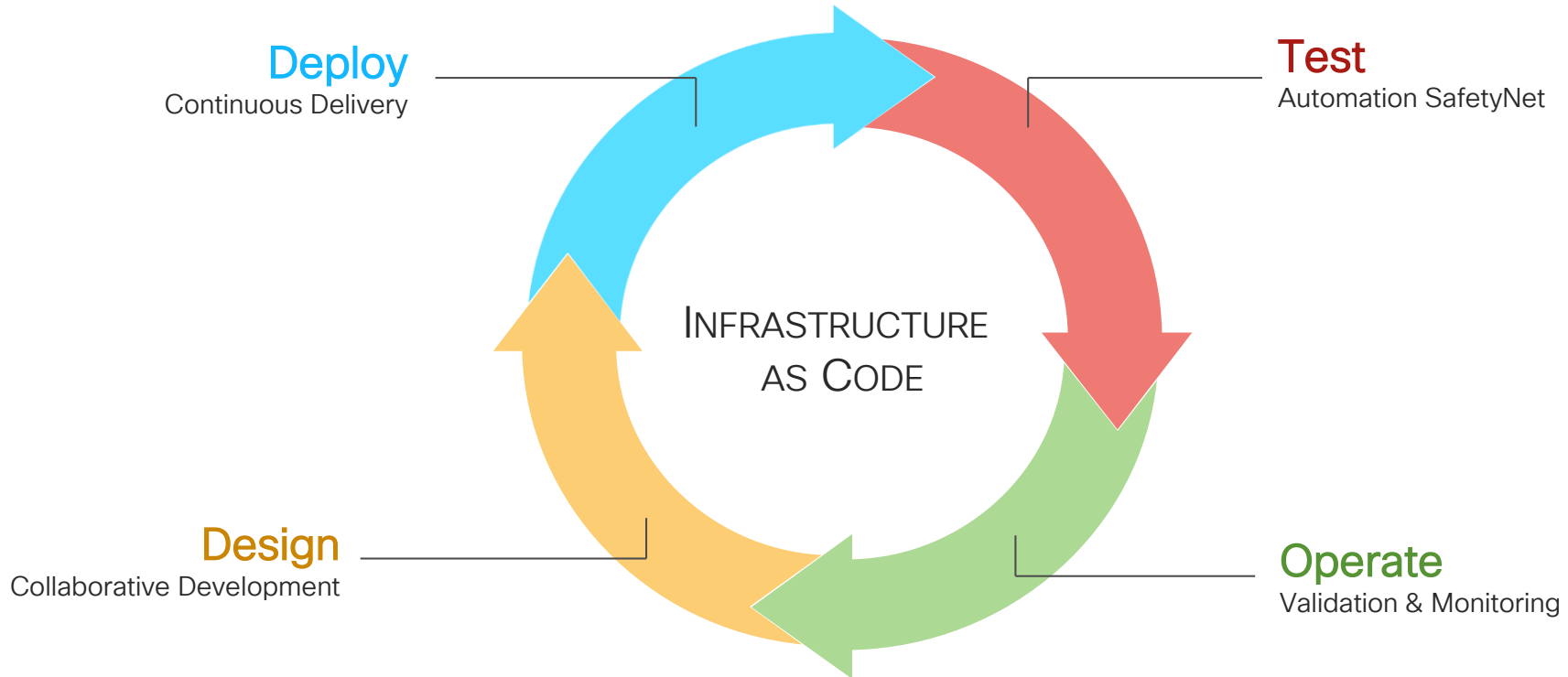
```
2 show interface
3 mgmt0 is up
4 admin state is up,
5 Hardware: Ethernet, address: 5e00.8003.0000 (bia 5e00.8003.0000)
6 Internet Address is 172.16.1.82/24
7 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
8 reliability 255/255, txload 1/255, rxload 1/255
9 Encapsulation ARPA, medium is broadcast
10 full-duplex, 1000 Mb/s
11 Auto-Negotiation is turned on
12 Auto-mdix is turned off
13 EtherType is 0x0000
14 1 minute input rate 1390384 bits/sec, 1311 packets/sec
15 1 minute output rate 136 bits/sec, 0 packets/sec
16 Rx
17 15256760 input packets 2329 unicast packets 15240578 multicast packets
18 13853 broadcast packets 2024379737 bytes
19 Tx
20 2598 output packets 2330 unicast packets 259 multicast packets
21 9 broadcast packets 298025 bytes
22
23 Ethernet1/1 is down (Link not connected)
24 admin state is up, Dedicated Interface
25 Hardware: 100/1000/10000 Ethernet, address: 5e00.8003.0007 (bia 5e00.8003.0008)
26 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec
27 reliability 255/255, txload 1/255, rxload 1/255
28 Encapsulation ARPA, medium is broadcast
29 auto-duplex, 1000 Mb/s
30 Beacon is turned off
31 Auto-Negotiation is turned on FEC mode is Auto
32 Input flow-control is off, output flow-control is off
33 Auto-mdix is turned off
34 Switchport monitor is off
35 EtherType is 0x8100
36 EEE (efficient-ethernet) : n/a
37 Last link flapped 00:00:17
38 Last clearing of "show interface" counters never
39 2 interface resets
```

Network DevOps Evolution

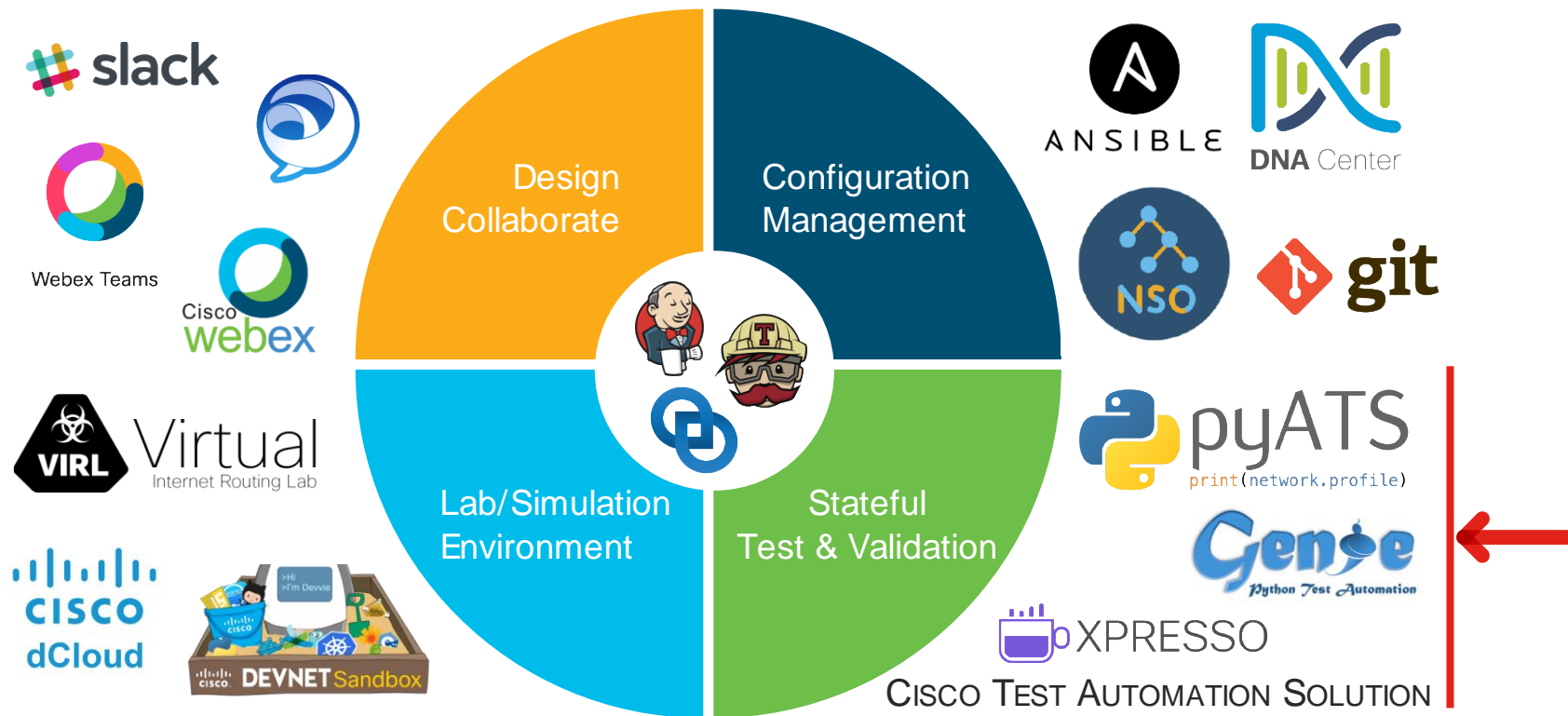


NetDevOps

“bringing all that is good in Software DevOps and applied towards Networking”



NetDevOps Tools



cisco *Live!*

pyATS | Genie: Test Automation Framework

- DevX test infrastructure for Cisco engineering since 2012.
- CI/CD, test, validation and automation across various platforms and teams
 - IOS, IOSXE, IOSXR, NXOS
 - Security, Cable, IOT, ACI, Cisco DNA Center...



~3500 developers



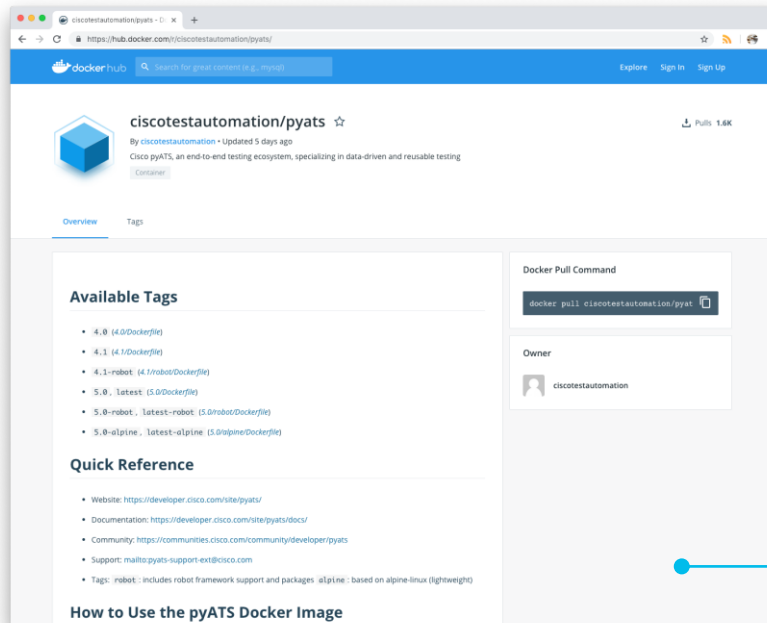
15,000,000+ LoC



3,500,000+ runs
month

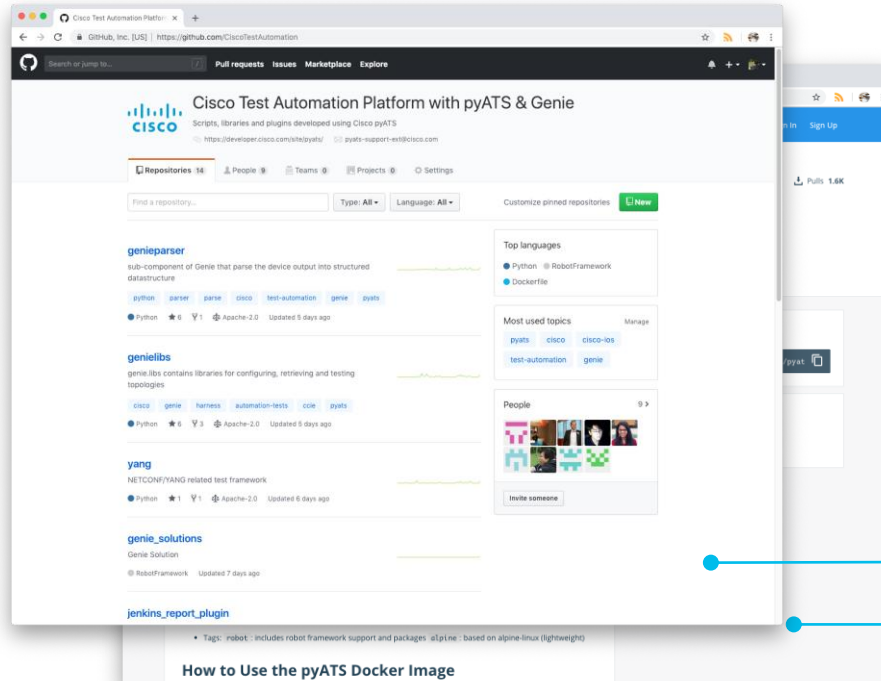
pyATS | Genie: Available to Everyone

pyATS | Genie: Available to Everyone



hub.docker.com/u/ciscotestautomation

pyATS | Genie: Available to Everyone



github.com/CiscoTestAutomation

hub.docker.com/u/ciscotestautomation

pyATS | Genie: Available to Everyone

The image shows a composite of two web pages. The top page is the Cisco DevNet landing page for pyATS & Genie, featuring a blue header with the DevNet logo and navigation links. The main content area has a blue background with the text "Accelerating your DevOps with pyATS & Genie" and a "Learn More" button. Below this, there are three icons representing "Certification Tests", "CI/CD Safety Net", and "Network Automation", each with a brief description and a "Learn more" link. The bottom page is a Docker Hub page for the "jenkins_report_plugin" image, showing the "How to Use the pyATS Docker Image" section.

Accelerating your DevOps with pyATS & Genie

THE Cisco Test Automation & Validation Solution: a 2018 Cisco Pioneer Award Winner.

pyATS & Genie is an end-to-end DevOps automation ecosystem. Agnostic by design, pyATS & Genie enables network engineers automate their day-to-day DevOps activities, performing stateful validation of their device operational status, and build a safety net of scalable, data-driven and reusable test cases around their network.

[Learn More](#)

pyATS & Genie In Your Network

- Certification Tests**
Assess whether new products, releases & changesets work with the rest of your network using pyATS & Genie as your gatekeeper.
[Learn more](#)
- CI/CD Safety Net**
Profile your network, save. Make changes as you will, and compare the state before and after. The framework will tell you the exact side effects of your changes.
- Network Automation**
Build your own business logic on top of the libraries that models after your devices and features. Control how you want to monitor the state of your network.
[Chat with Us!](#)

[jenkins_report_plugin](#)

Tags: robot: includes robot framework support and packages alpine: based on alpine linux (lightweight)

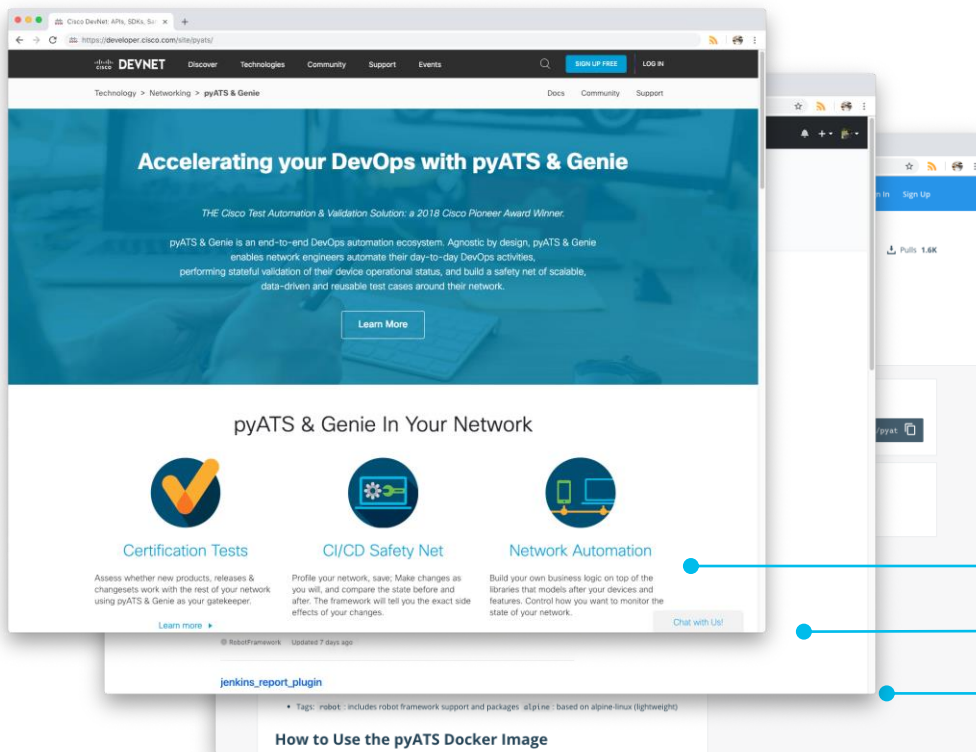
How to Use the pyATS Docker Image

developer.cisco.com/pyats/

github.com/CiscoTestAutomation

hub.docker.com/u/ciscotestautomation

pyATS | Genie: Available to Everyone



6000+
Monthly
Downloads

developer.cisco.com/pyats/

github.com/CiscoTestAutomation

hub.docker.com/u/ciscotestautomation

cisco *Live!*

Solution Layout

Business Logic

Integration

- XPRESSO, Ansible, RobotFramework
- Jenkins, CI/CD pipelines, CLI, other tooling, etc

SDK & Library

Genie Libs

- Parsers, Feature/Protocol Models
- Reusable Testcases: Triggers, Verifications

Genie Library Framework

- Basis for agnostic automation libraries
- Boilerplate library foundation & engine

Toolbox

pyATS Core Test Infrastructure

- Topology & Test definition
- Execution & Reporting

Open-Source Libraries

- The core of pyATS is vendor, platform, feature, and protocol agnostic
- All implementations are achieved through open-source libraries:
 - Connection plugins & extensions
 - Libraries - feature models, parsers, data-driven testcases
<https://github.com/CiscoTestAutomation/>
- Example:
 - IOSXE (eg asr1k, cat9k)
 - Junos
 - NXOS (eg n9k, n7k)
 - NSO
 - IOSXR (eg asr9k)
 - NGFW (eg, asa)

...

Automate Everything: Test, Validate, DevOps

Test & Validation

- Pre-commit, Sanity, Regression
- HA, Scale, Stability, Solution
- Certification
- Test as-a service
- Glue other test solutions and frameworks together

DevOps: Day-to-day

- Network monitoring
- Keep tabs on what changed in your network over time
- Pin-point the cause of failure
- Preemptive and proactive network error/failure detection
- Chaos Monkey

Snapshot, Save, Compare

```
$ pyats learn ospf interface  
--testbed-file tb.yaml  
--output today
```

```
$ pyats diff today/ yesterday/
```

```
*****  
Comparison between ./06_08_19/snapshot and ./06_08_20/snapshot is different for feature 'config' for device:  
  
'nx-osv-1'  
interface Ethernet2/1  
- no shutdown  
+ shutdown  
  
*****  
Comparison between ./06_08_19/snapshot and ./06_08_20/snapshot is different for feature 'ospf' for device:  
  
'nx-osv-1'  
info:  
vrf:  
  default:  
    address_family:  
      ipv4:  
        instance:  
          1:  
            areas:  
              0.0.0.0:  
                interfaces:  
                  Ethernet2/1:  
+                 enable: False  
-                 enable: True  
-                 neighbors:  
-                   10.1.1.1:  
-                     address: 10.0.1.1  
-                     bdr_ip_addr: 10.0.1.1  
-                     dead_timer: 00:00:35  
-                     dr_ip_addr: 10.0.1.2  
-                     last_state_change: 18:23:54  
-                     neighbor_router_id: 10.1.1.1  
-                     state: full  
-                     statistics:  
-                       nbr_event_count: 5
```

Parsers

converting/formatting command output into Python data structures

Parsers

converting/formatting command output into Python data structures

```
show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 2.2.2.2, remote AS 65000, internal link
BGP version 4, remote router ID 2.2.2.2
BGP state = Established, up for 4d00h
Last read 00:00:50, last write 00:00:44, hold time is 180, keepalive interval is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised
  Multisession Capability:
  Stateful switchover support enabled: NO for session 1
Message statistics:
  InQ depth is 0
  OutQ depth is 0

      Sent          Rcvd
Opens:          1          1
Notifications:  0          0
Updates:         2          2
Keepalives:    6369       5780
Route Refresh:  0          0
Totals:        6372       5783
Do log neighbor state changes (via global configuration)
Default minimum time between advertisement runs is 0 seconds

Address tracking is enabled, the RIB does have a route to 2.2.2.2
Route to peer address reachability Up: 3; Down: 0
  Last notification 4d00h
Connections established 1; dropped 0
Last reset never
Interface associated: (none) (peering address NOT in same link)
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 1.1.1.1, Local port: 179
Foreign host: 2.2.2.2, Foreign port: 36402
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x14B4F1B9):
Timer      Starts    Wakeups    Next
Retrans    6371      0          0x0
TimeWait   0         0          0x0
AckHold    5782      5667       0x0
```

Parsers

converting/formatting command output into Python data structures

```
show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 2.2.2.2, remote AS 65000, internal link
BGP version 4, remote router ID 2.2.2.2
BGP state = Established, up for 4d00h
Last read 00:00:50, last write 00:00:44, hold time is 180, keepalive interval is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised
  Multisession Capability:
  Stateful switchover support enabled: NO for session 1
Message statistics:
  InQ depth is 0
  OutQ depth is 0

      Sent          Rcvd
Opens:          1           1
Notifications:  0           0
Updates:         2           2
Keepalives:    6369        5780
Route Refresh:  0           0
Total:         6372        5783
Do log neighbor state changes (via global configuration)
Default minimum time between advertisement runs is 0 seconds

Address tracking is enabled, the RIB does have a route to 2.2.2.2
Route to peer address reachability Up: 3; Down: 0
  Last notification 4d00h
Connections established 1; dropped 0
Last reset never
Interface associated: (none) (peering address NOT in same link)
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 1.1.1.1, Local port: 179
Foreign host: 2.2.2.2, Foreign port: 36402
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x14B4F1B9):
Timer      Starts    Wakeups    Next
Retrans    6371      0          0x0
TimeWait   0         0          0x0
AckHold    5782      5667      0x0
```



Parsing Function

Parsers

converting/formatting command output into Python data structures

```
show bgp all neighbors
For address family: IPv4 Unicast
BGP neighbor is 2.2.2.2, remote AS 65000, internal link
BGP version 4, remote router ID 2.2.2.2
BGP state = Established, up for 4d00h
Last read 00:00:50, last write 00:00:44, hold time is 180, keepalive interval is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised
  Multisession Capability:
  Stateful switchover support enabled: NO for session 1
Message statistics:
  InQ depth is 0
  OutQ depth is 0

Opens:          1          1
Notifications:  0          0
Updates:        2          2
Keepalives:     6369      5780
Route Refresh:  0          0
Total:          6372      5783
Do log neighbor state changes (via global configuration)
Default minimum time between advertisement runs is 0 seconds

Address tracking is enabled, the RIB does have a route to 2.2.2.2
Route to peer address reachability Up: 3; Down: 0
  Last notification 4d00h
Connections established 1; dropped 0
Last reset never
Interface associated: (none) (peering address NOT in same link)
Transport(tcp) path-mtu-discovery is enabled
Graceful-Restart is disabled
SSO is disabled
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Connection is ECN Disabled, Minimum incoming TTL 0, Outgoing TTL 255
Local host: 1.1.1.1, Local port: 179
Foreign host: 2.2.2.2, Foreign port: 36402
Connection tableid (VRF): 0
Maximum output segment queue size: 50

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x14B4F1B9):
Timer      Starts    Wakeups    Next
Retrans    6371      0          0x0
TimeWait   0         0          0x0
AckHold    5782      5667      0x0
```



Parsing Function

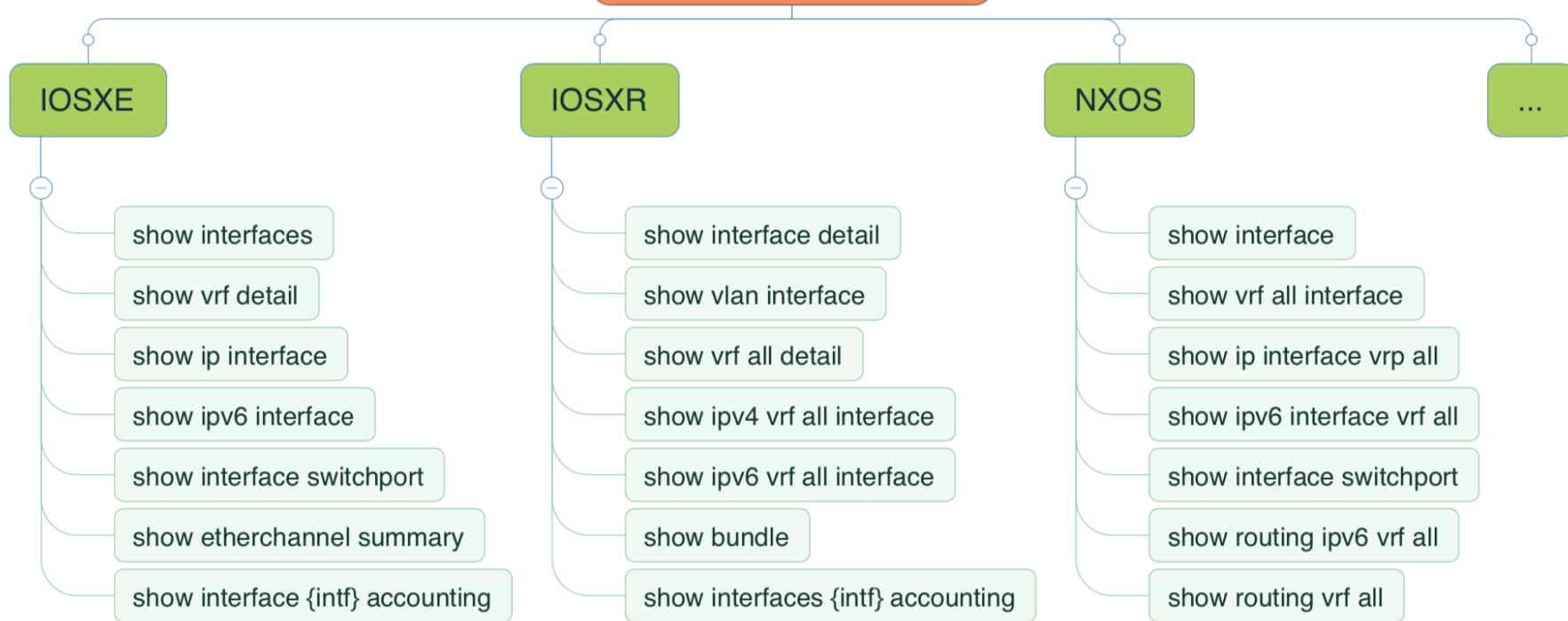
```
{
  "list_of_neighbors": [
    "2.2.2.2",
    "3.3.3.3",
    "2001:2:2:2::2",
    "2001:3:3:3::3",
    "2.2.2.2",
    "3.3.3.3",
    "2001:2:2:2::2",
    "2001:3:3:3::3"
  ],
  "vrf": {
    "VRF1": {
      "neighbor": {
        "2.2.2.2": {
          "address_family": {
            "vpngv4 unicast": {
              "current_time": "0x1E778CE1",
              "last_read": "00:00:51",
              "last_write": "00:00:43",
              "session_state": "Established",
              "up_time": "5d21h"
            }
          }
        },
        "bgp_event_timer": {
          "next": {
            "ackhold": "0x0",
            "deadwait": "0x0",
            "giveup": "0x0",
            "keepalive": "0x0",
            "linger": "0x0",
            "pmtuager": "0x0",
            "processq": "0x0",
            "retrans": "0x0",
            "sendwnd": "0x0",
            "timewait": "0x0"
          }
        }
      }
    }
  }
}
```

Feature/Protocol Model

- YANG-Inspired Python classes
- Represents a whole feature/protocol agnostically
 - Implement a use case for one platform – see it work on another seamlessly
- Human-Friendly, Programmer Friendly, Pythonic

```
{
  interface: {
    'description': description,          # 'Ethernet1/1'|'Vlan10'|'Ethernet1/1.10'
    'type': intf_type,                   # 'To R2'
    'oper_status': oper_status,          # '10/100/1000 Ethernet'
    'last_change': last_change,           # 'up'|'down'
    'phys_address': phys_address,         # '00:00:04'|'never'
    'mtu': mtu,                           # '5254.009c.f2e6'
    'enabled': enabled,                   # '1500 bytes'
    'vlan_id': vlan_id,                   # True(no shut)|False(shut)
    'access_vlan': access_vlan,           # '10'
    'trunk_vlans': trunk_vlans,            # '100'
    'mac_address': mac_address,           # '100200'
    'auto_negotiate': auto_negotiate,     # 'AAAA.BBBB.CCCC'
    'duplex_mode': duplex_mode,           # True|False
    'port_speed': port_speed,             # 'full'|'half'
    'switchport_enable': switchport_enable, # Boolean
    'switchport_mode': switchport_mode,   # '100'
    'medium': medium,                     # 'trunk'|'access'
    'delay': delay,                       # Enum('p2p','broadcast')
    'port_channel': {
      'port_channel_member': port_channel_member, # Boolean
      'port_channel_int': port_channel_int,       # 'Pol'
      'port_channel_member_intfs': port_channel_member_intfs, # List['Eth1/1','Eth1/2']
    },
    'flow_control': {
      'receive': flow_control_receive, # Boolean
      'send': flow_control_send,       # Boolean
    }
    'bandwidth': bandwidth,             # '1000000' Kbit
    'link_status': link_status,          # True|False (logging)
    'vrf': vrf,                          # 'VRF1'
    'vrf_downstream': vrf_downstream,    # 'VRF2'
    'accounting': {
      protocol: { # str, 'arp'
        'pkts_in': pkts_in, # int, 9
        'pkts_out': pkts_out, # int, 9
        'chars_in': chars_in, # int, 378
        'chars_out': chars_out, # int, 378
      },
      'counters': {
        'rate': {
          'load_interval': load_interval, # '5 minutes'
        }
      }
    }
  },
  # ...
}
```

Interface Model



Genie: The pyATS Standard Library

<https://pubhub.devnetcloud.com/media/genie-feature-browser/docs/#/>

- Agnostic framework: multiple OS/Platform/Release/etc support
- Multi-mgmt. protocol: support for parsing CLI/NETCONF/XML/etc
- Self-testing: each parser features its own schema

> 1500 parser on various platforms

> 30 feature/protocol models

> 1000 triggers & verifications (reusable, data-driven testcases)

> 500 convenience APIs

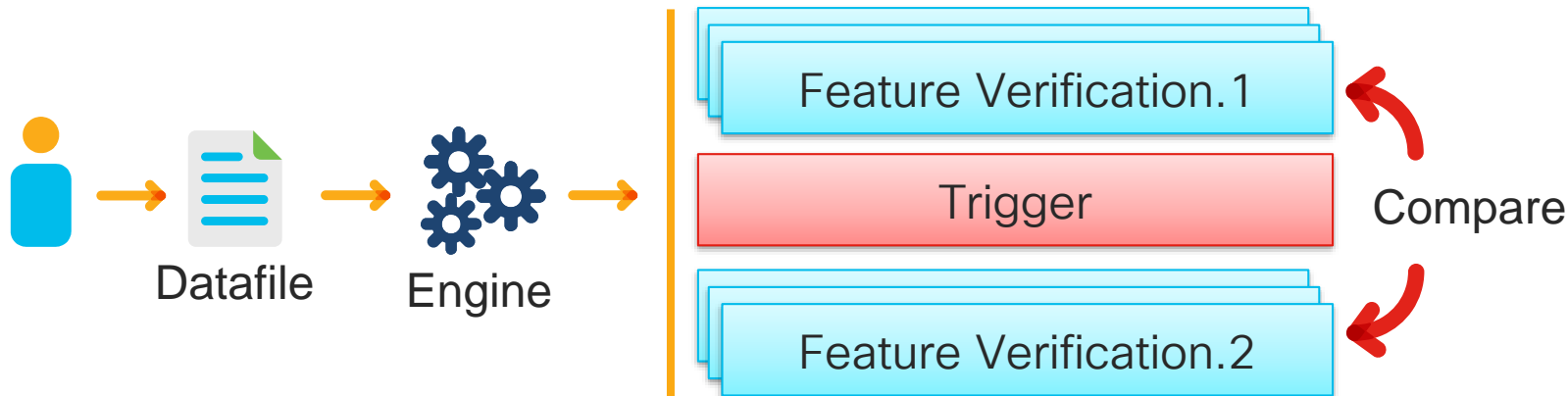
Perform ..., Verify ...

Trigger:

- perform incremental change
- check change took effect
- Roll-back configuration if necessary

• Verification:

- retrieve current state of device
- compare against a known state
 - is the difference expected?
- Verify traffic, usage, statistics, syslogs, etc





Demo

```
bash$ genie run --pts-features platform bgp interface  
--verification-uids "Verify_IpInterfaceBrief Verify_IpRoute_vrf_all"  
--trigger-uids "TriggerUnconfigConfigBgp.uut  
                TriggerShutNoShutBgpNeighbors  
                TriggerModifyLoopbackInterfaceIp.uut  
                TriggerShutNoShutEthernetInterface"  
--testbed-file testbed.yaml
```

ROBOT Framework

- Generic TDD/A-TDD Test Framework
- Uses **keyword-driven approach** to define test actions & testcases
- Test capabilities are extended by **test libraries**
- Anyone can create new **higher-level keywords** from existing ones
- OS/Application independent, based on Python

```
*** Settings ***
Library      pyats.robot.pyATSRobot
Library      genie.libs.robot.GenieRobot
Library      unicon.robot.UniconRobot

*** Variables ***
${pts_file}  ${CURDIR}/pts
${testbed}   %${TESTBED}

*** Test Cases ***
Initialize
    # initialize testbed by loading it through Genie
    use genie testbed "${testbed}"

Connect to devices
    # establish connection to all devices
    connect to devices

Learn current BGP state
    # save current BGP state by calling bgp summary parser, and saving it
    # internally for future reference
    run verification "Verify_BgpAll" on device "uut"

Trigger UnconfigConfig BGP
    # toggle bgp configuration, and verify that bgp returns to normal
    run trigger "TriggerUnconfigConfigBgp" on device "uut" using alias "cli"

Verify BGP state after trigger
    # verify bgp state against the first time this verification was called
    # ensure nothing changed
    run verification "Verify_BgpAll" on device "uut"

Verify BGP neighbor count
    # check specific bgp neighbor count
    verify count "1" "bgp neighbors" on device "uut"

Verify BGP routes
    # check specific bgp route counts
    verify count "2" "bgp routes" on device "uut"
```

Ansible Module

Ansible Blog: <https://www.ansible.com/blog/ansible-servicenow-part-2-parsing-facts-from-network-devices-using-pyats/genie>

Example Use Case:

- Using pyATS/Genie to parse device output as facts
- Take the facts and uploading them as data to ServiceNow using API

Automation: Reactive vs. Preemptive

Reactive	Preemptive
user/application experiences an issue, error or outage	24/7 monitoring of the system and looking for early failure symptoms
user contact support organization, ticket is created	automatically alert maintenance/support team about about potential and/or developing issue
escalation, as problem/outage widens and/or drags on	proactively disable, redirect, and or attempt to mitigate issue, while contacting impacted users
impacts production services	issues caught early and goes unnoticed by end user



Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```


Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices
- Check all interfaces for CRC counts > 0

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices
- Check all interfaces for CRC counts > 0
- Generate report email

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices
- Check all interfaces for CRC counts > 0
- Generate report email

Can we do better?

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices
- Check all interfaces for CRC counts > 0
- Generate report email

Can we do better?

- Keep track of error rate over time

Preemptive Network Automation Example

Interface CRC Error Counts – do you check them often?

```
R1_xe#show interfaces | inc CRC
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

- Connect to all your devices
- Check all interfaces for CRC counts > 0
- Generate report email

Can we do better?

- Keep track of error rate over time

https://github.com/CiscoTestAutomation/solutions_examples/tree/master/crc_errors

Chaos Monkey

- Randomly invoking a subset of triggers to introduce change on targeted testbed devices
- Verify whether your system continues to function as expected

“the only limitation is your comfort level”

```
from genie.harness.main import gRun

# chaos only knows..
import random

POSSIBLE_TRIGGERS = [
    'TriggerClearBgpAll',
    'TriggerClearIpOspfNeighborVrfAll',
    'TriggerClearIpRoute',
    'TriggerUnconfigConfigEvpnVni',
    'TriggerUnconfigConfigVlanInterface',
    'TriggerUnconfigConfigVlanVnsegment',
]

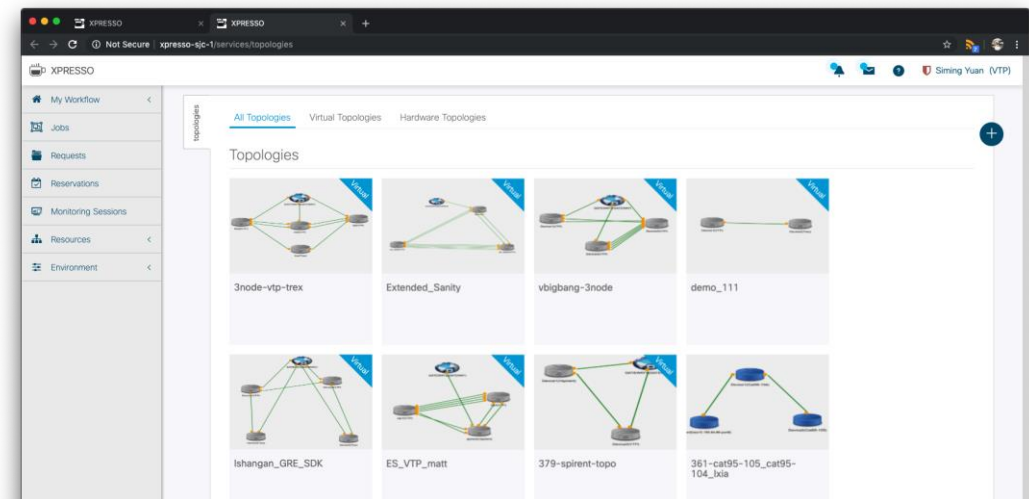
def main():
    'pyATS job entry point'

    # launch genie
    # profile ospf and bgp
    # run the above triggers randomly
    gRun(pts_features = ['ospf', 'bgp'],
        trigger_uids = random.choice(POSSIBLE_TRIGGERS))
```

https://github.com/CiscoTestAutomation/solutions_examples/tree/master/netchaos

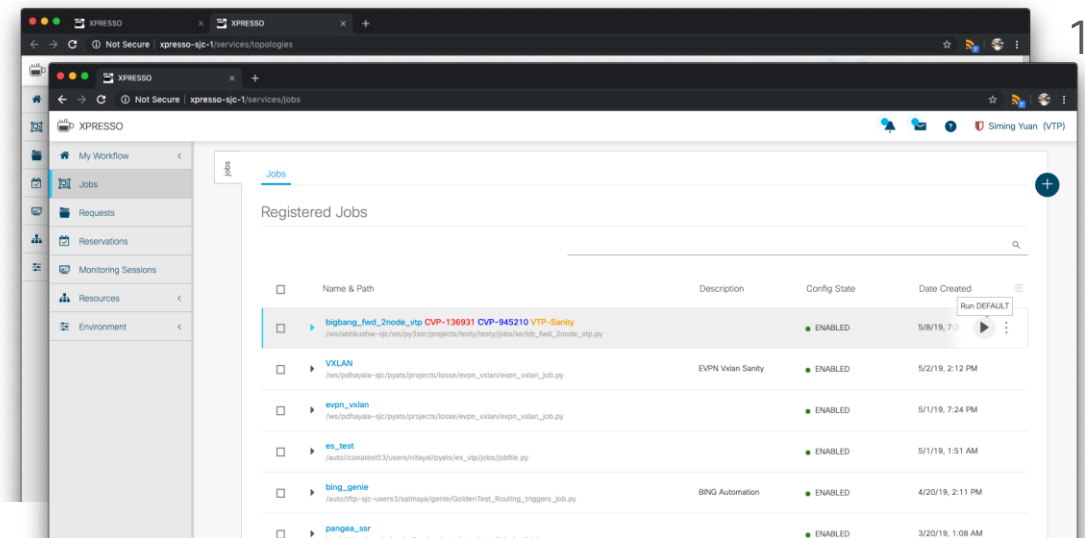
pyATS Dashboard: XPRESSO

pyATS Dashboard: XPRESSO



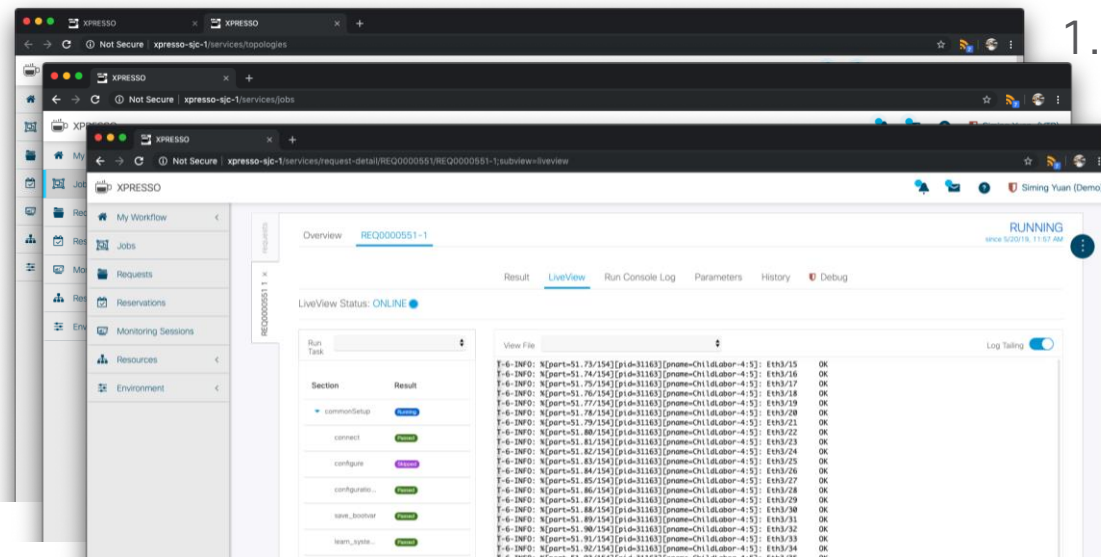
1. Reserve Topology

pyATS Dashboard: XPRESSO



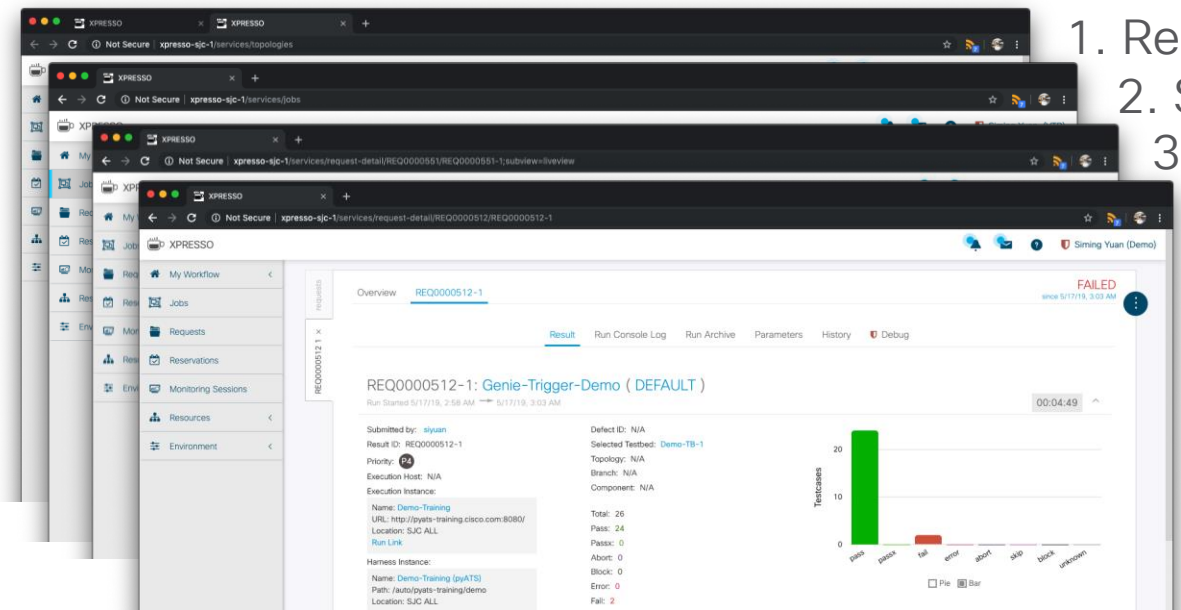
1. Reserve Topology
2. Select suite to run

pyATS Dashboard: XPRESSO



1. Reserve Topology
2. Select suite to run
3. See run in action

pyATS Dashboard: XPRESSO



1. Reserve Topology
2. Select suite to run
3. See run in action
4. Analyze results

pyATS Dashboard: XPRESSO

The image displays a stack of four browser windows showing the XPRESSO dashboard. The top window shows the 'Overview' page for request REQ0000512-1, detailing its submission by 'siyuan', its priority, and its execution instance. The bottom window shows the 'Result' page for the same request, displaying a detailed log of the test run, including the VPN session setup and VNC session details.

1. Reserve Topology
2. Select suite to run
3. See run in action
4. Analyze results

5. VPN to testbed for manual tests

Environment

- Linux/macOS/WSL/Docker
- Python 3.5-3.7 (3.8 soon)
- Your preferred editor:
 - vim, VSCode, Atom, Sublime Text, PyCharm... etc
- Connectivity to your devices



```
bash$ python3 -m venv ~/pyats
bash$ source ~/pyats/bin/activate
(pyats) bash$ pip install pyats[full]
```

Recap

- **Lightweight:** intuitively Python, simple concepts
- **Scalable & Reusable:** start with one device, scale to hundreds of devices
- **Data-driven, configuration-independent**
- **Agnostic:** library works cross NOS, concept works cross all networking devices and protocols
- **Open-Source Library:** can see immediately why something works or not, easy to contribute back, easy to tinker with
- **Caters to Everyone:** works for non-programmers, and scales to sophisticated software/solutions

Resources

- DevNet: pyATS/Genie - <https://developer.cisco.com/pyats/>
- Framework Documentation: <https://developer.cisco.com/site/pyats/docs/>
- Package Documentation:
<https://developer.cisco.com/site/pyats/docs/packages/>
- GitHub Folder: <https://github.com/CiscoTestAutomation>
- DockerHub: <https://hub.docker.com/r/ciscotestautomation/pyats/>

Learn more about the new DevNet Certifications and how you can prepare now!

Associate Level

Specialist Level

Professional Level

Expert Level

Engineering



Software



Future
Offering

Start Here | Upcoming Cisco DevNet Certifications

- Start at **Meet DevNet**

DEVNET-2864: Getting ready for Cisco DevNet Certifications

Offered daily at 9am, 1pm & 4pm at Meet DevNet

- Attend a **brownbag session**

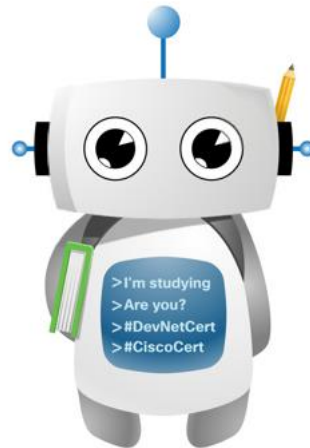
DEVNET-4099: DevNet Certifications: Bringing software practices & software skills to networking

Offered daily 12:15-12:45 in the DevNet Zone Theater

- Visit the **Learning@Cisco** booth
- Scan this code to **sign up** for the latest updates or go to <http://cs.co/20eur02>

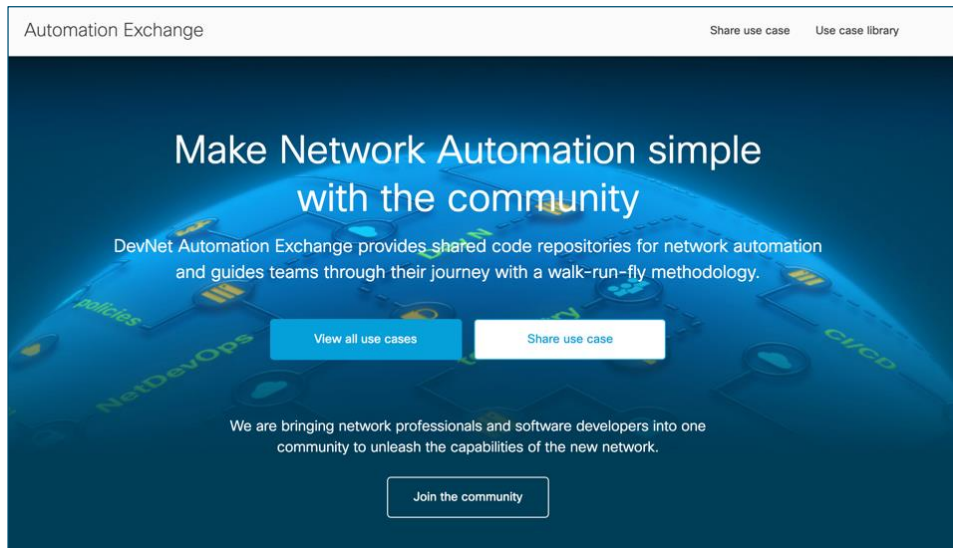


cisco *Live!*



Find shared code repositories of use cases for network automation & more!

Don't miss our 5
Automate Infrastructure
demos in the
DevNet Zone!



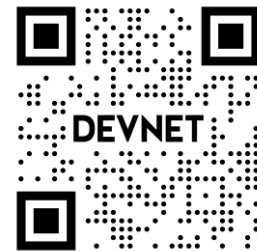
Start at **Meet DevNet**

DEVNET-3010 [a-j] Learn how to make Network Automation Simple with the Community

Offered Monday 2pm & 5pm, Tuesday & Wednesday 10am, 2pm & 5pm, and Thursday 10am & 5pm at Meet DevNet

cisco *Live!*

Scan this code or go to the URL to **learn more**



<http://cs.co/20eur01>

Complete your online session survey



- Please complete your session survey after each session. Your feedback is very important.
- Complete a minimum of 4 session surveys and the Overall Conference survey (starting on Thursday) to receive your Cisco Live t-shirt.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Content Catalog on ciscolive.com/emea.

Cisco Live sessions will be available for viewing on demand after the event at ciscolive.com.

Continue your education



Demos in the
Cisco Showcase



Walk-In Labs



Meet the Engineer
1:1 meetings



Related sessions



Thank you





You make **possible**