

Automated Testing Using pyATS and Genie



PyATS

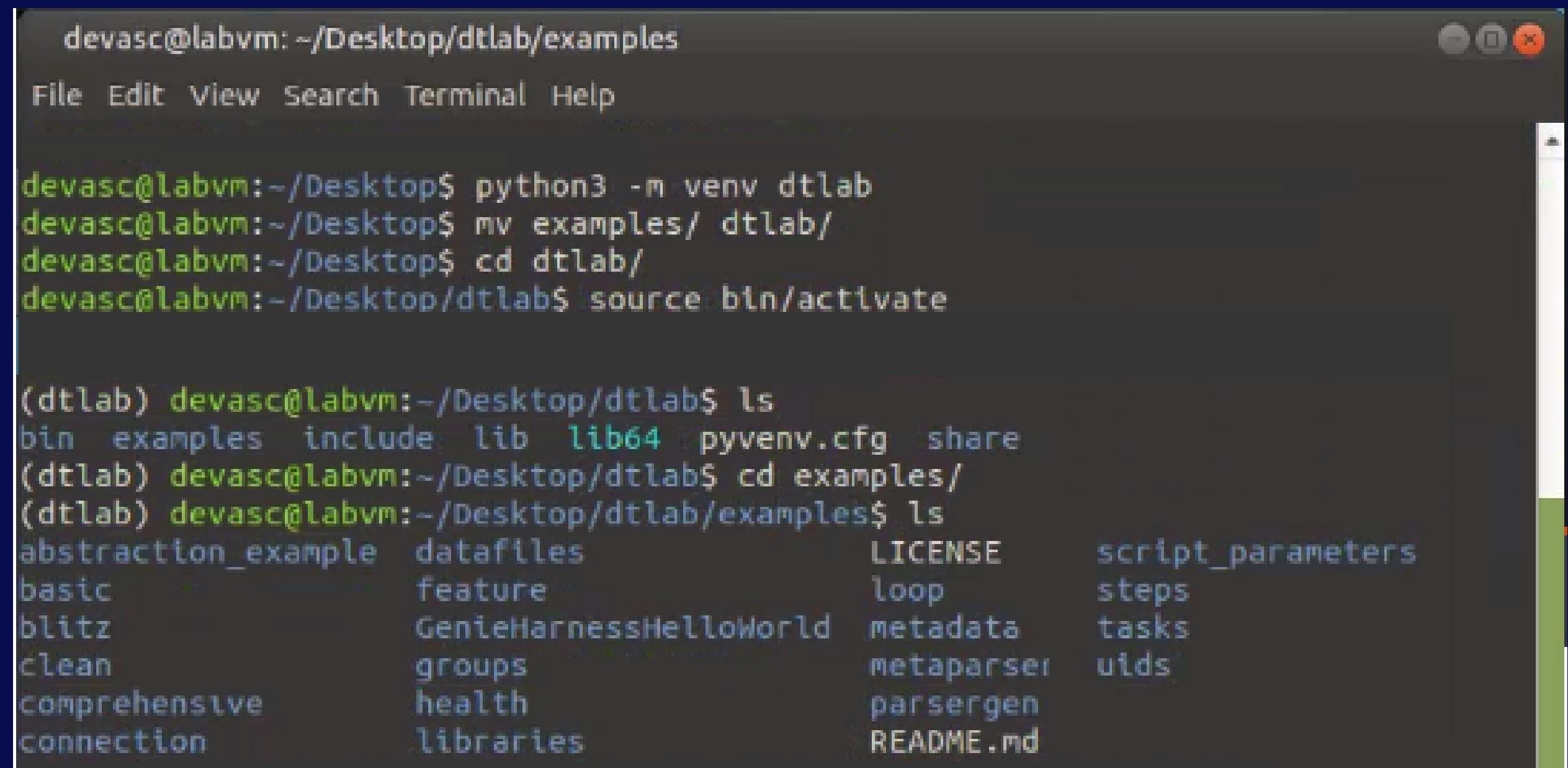
- The pyATS tool is an end-to-end testing ecosystem, specializing in data-driven and reusable testing, and engineered to be suitable for Agile, rapid development iterations
- ↓
- pyATS enables developers start with small, simple, and linear test cases, and scale towards large, complex, and asynchronous test suites.

Genie

- Genie extends and builds on pyATS to be used in a networking environment

Passo 1

1. Launch the DEVASC VM
2. Create a Python Virtual Environment



The screenshot shows a terminal window with a dark background and light-colored text. The title bar reads "devasc@labvm: ~/Desktop/dtlab/examples". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal prompt is "devasc@labvm:~/Desktop\$". The user runs several commands to set up a virtual environment:

```
devasc@labvm:~/Desktop$ python3 -m venv dtlab
devasc@labvm:~/Desktop$ mv examples/ dtlab/
devasc@labvm:~/Desktop$ cd dtlab/
devasc@labvm:~/Desktop/dtlab$ source bin/activate
```

After activating the environment, the prompt changes to "(dtlab)" and the user lists the contents of the "examples" directory:

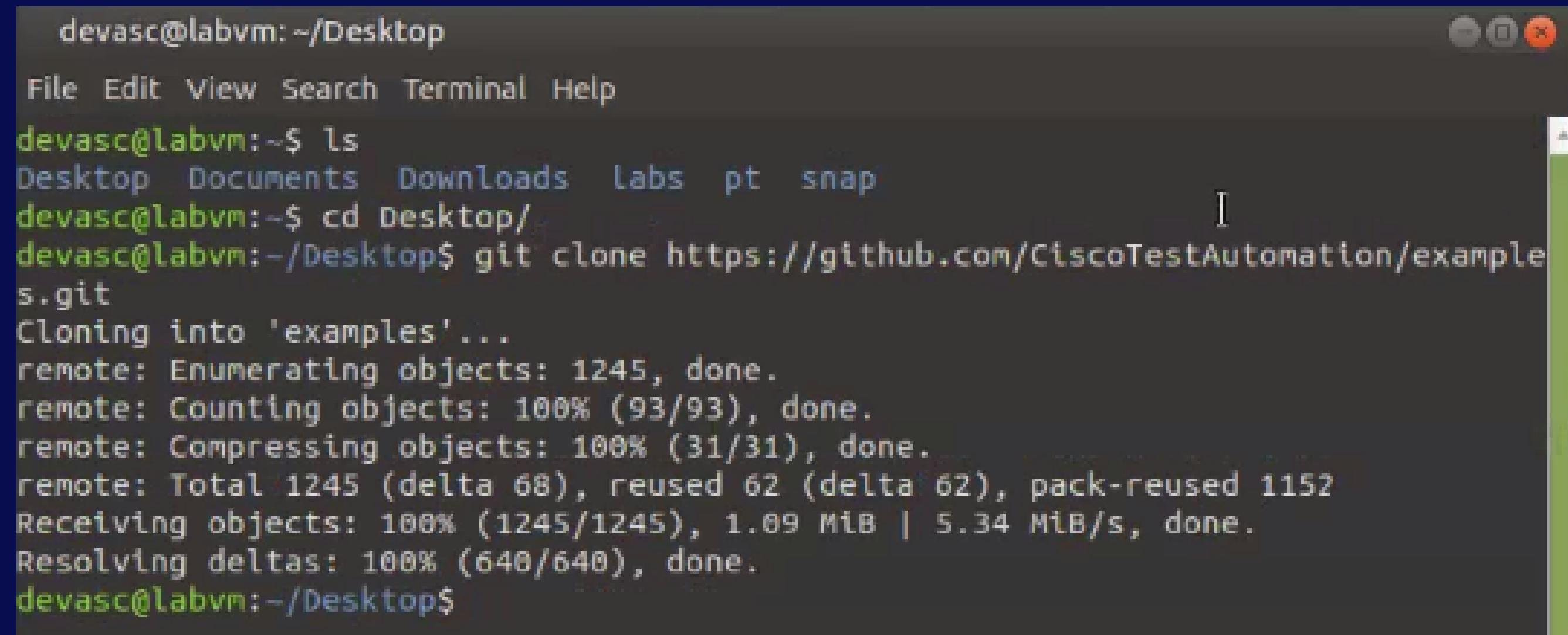
```
(dtlab) devasc@labvm:~/Desktop/dtlab$ ls
bin  examples  include  lib  lib64  pyvenv.cfg  share
(dtlab) devasc@labvm:~/Desktop/dtlab$ cd examples/
(dtlab) devasc@labvm:~/Desktop/dtlab/examples$ ls
abstraction_example  datafiles          LICENSE      script_parameters
basic                 feature           loop        steps
blitz                 GenieHarnessHelloWorld  metadata    tasks
clean                 groups            metaparser  uids
comprehensive        health            parsersgen README.md
connection
```

Passo 2

1. Use the pyATS Testing Library

```
devasc@labvm: ~/Desktop/dtlab/examples
File Edit View Search Terminal Help
(dtlab) devasc@labvm:~/Desktop/dtlab/examples$ pip3 install pyats
Collecting pyats.reporter<22.5.0,>=22.4.0
    Downloading pyats.reporter-22.4-cp38-cp38-manylinux1_x86_64.whl (3.4 MB)
        |████████| 3.4 MB 10.2 MB/s
Collecting pyats.connections<22.5.0,>=22.4.0
    Downloading pyats.connections-22.4-cp38-cp38-manylinux1_x86_64.whl (1.1 MB)
        |████████| 1.1 MB 5.9 MB/s
Collecting pyats.utils<22.5.0,>=22.4.0
    Downloading pyats.utils-22.4-cp38-cp38-manylinux1_x86_64.whl (8.5 MB)
        |████████| 8.5 MB 8.2 MB/s
Collecting pyats.topology<22.5.0,>=22.4.0
    Downloading pyats.topology-22.4-cp38-cp38-manylinux1_x86_64.whl (3.4 MB)
        |████████| 3.4 MB 7.7 MB/s
Collecting pyats.easypy<22.5.0,>=22.4.0
    Downloading pyats.easypy-22.4-cp38-cp38-manylinux1_x86_64.whl (6.8 MB)
        |████████| 6.8 MB 9.4 MB/s
Collecting pyats.async<22.5.0,>=22.4.0
    Downloading pyats.async-22.4-cp38-cp38-manylinux1_x86_64.whl (692 kB)
        |████████| 692 kB 9.1 MB/s
Collecting pyats.results<22.5.0,>=22.4.0
    Downloading pyats.results-22.4-cp38-cp38-manylinux1_x86_64.whl (603 kB)
        |████████| 603 kB 9.6 MB/s
Collecting pyats.datastructures<22.5.0,>=22.4.0
    Downloading pyats.datastructures-22.4-cp38-cp38-manylinux1_x86_64.whl (1.6 MB)
        |████████| 1.6 MB 7.5 MB/s eta 0:00:01
```

2. Clone and examine the pyATS sample scripts from GitHub.



A screenshot of a terminal window titled "Terminal". The window has a dark background and light-colored text. At the top, it shows the user's name "devasc@labvm: ~/Desktop" and a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". Below the menu, there are several icons in the top right corner. The main area of the terminal shows the following command-line session:

```
devasc@labvm:~/Desktop
File Edit View Search Terminal Help
devasc@labvm:~$ ls
Desktop Documents Downloads Labs pt snap
devasc@labvm:~$ cd Desktop/
devasc@labvm:~/Desktop$ git clone https://github.com/CiscoTestAutomation/examples.git
Cloning into 'examples'...
remote: Enumerating objects: 1245, done.
remote: Counting objects: 100% (93/93), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 1245 (delta 68), reused 62 (delta 62), pack-reused 1152
Receiving objects: 100% (1245/1245), 1.09 MiB | 5.34 MiB/s, done.
Resolving deltas: 100% (640/640), done.
devasc@labvm:~/Desktop$
```

3. Run pyATS manually to invoke the basic test case.

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ pyats run job examples/basic/blitz/
(dtlab) devasc@labvm:~/Desktop/dtlab$ pyats run job examples/basic/basic_example_job.py
2022-05-07T15:23:24: %EASYPY-INFO: Starting job run: basic_example_job
2022-05-07T15:23:24: %EASYPY-INFO: Runinfo directory: /home/devasc/.pyats/runinfo/basic_example_job.2022May07_15:23:23.857152
2022-05-07T15:23:24: %EASYPY-INFO: -----
----- PASSED
2022-05-07T15:23:30: %EASYPY-INFO: Sending report email...
2022-05-07T15:23:30: %EASYPY-INFO: Missing SMTP server configuration, or failed to reach/authenticate/send mail. Result notification email failed to send.
2022-05-07T15:23:30: %EASYPY-INFO: Done!

Pro Tip
-----
Try the following command to view your logs:
    pyats logs view
```

Passo 3 Use Genie to Parse IOS Command Output

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ genie
Usage:
  genie <command> [options]

Commands:
  create          Create Testbed, parser, triggers, ...
  develop         Puts desired pyATS packages into development mode
  diff            Command to diff two snapshots saved to file or directory
  dnac            Command to learn DNAC features and save to file (Prototype)
  learn           Command to learn device features and save to file
  parse           Command to parse show commands
  run             Run Genie triggers & verifications in pyATS runtime environment
  shell           enter Python shell, loading a pyATS testbed file and/or pickled data
  undevelop       Removes desired pyATS packages from development mode

General Options:
  -h, --help        Show help

Run 'genie <command> --help' for more information on a command.
```

1. Create a testbed YAML file

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ genie create testbed interactive --output yaml/testbed.yml -
-encode-password
Start creating Testbed yaml file ...
Do all of the devices have the same username? [y/n] n
Do all of the devices have the same default password? [y/n] n
Do all of the devices have the same enable password? [y/n] n

Device hostname: CSR1kv
  IP (ip, or ip:port): 192.168.56.101
  Username: cisco
Default Password (leave blank if you want to enter on demand):
Enable Password (leave blank if you want to enter on demand):
  Protocol (ssh, telnet, ...): ssh -o KexAlgorithms=diffie-hellman-group14-sha1
  OS (iosxr, iosxe, ios, nxos, linux, ...): iosxe
More devices to add ? [y/n] n
Testbed file generated:
yaml/testbed.yml
```

2. Use cat to view the testbed.yml file in the yaml directory

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ cat yaml/testbed.yml
devices:
  CSR1kv:
    connections:
      cli:
        ip: 192.168.56.101
        protocol: ssh -o KexAlgorithms=diffie-hellman-group14-sha1
    credentials:
      default:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmMKH}%
        username: cisco
      enable:
        password: '%ENC{w5PDos0Uw5fDosKQwpbCmMKH}%
    os: iosxe
    type: iosxe
(dtlab) devasc@labvm:~/Desktop/dtlab$
```

3. Using your testbed YAML file, invoke Genie to parse unstructured output from the show ip interface brief command into structured JSON

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ genie parse "show ip interface brief" --testbed-file yaml/testbed.yml --devices CSR1kv
Using the default YAML encoding key since no key was specified in configuration.
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN `pyats secret keygen` AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALUES.
2022-05-07T15:44:05: %UNICON-WARNING: Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring
Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring
  0%| | 0/1 [00:00<?, ?it/s]
{
  "interface": {
    "GigabitEthernet1": {
      "interface_is_ok": "YES",
      "ip_address": "192.168.56.101",
      "method": "DHCP",
      "protocol": "up",
      "status": "up"
    }
  }
}
100%| | 1/1 [00:01<00:00, 1.07s/it]
```

4. Use Genie to parse output from the show version command into JSON

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ genie parse "show version" --testbed-file yaml/testbed.yml  
--devices CSR1kv  
Using the default YAML encoding key since no key was specified in configuration.  
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN 'pyats secret keygen' AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALUES.  
2022-05-07T15:46:17: %UNICON-WARNING: Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring  
Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring  
| 0x | 0/1 [00:00<?, ?it/s]  
{  
    "version": {  
        "chassis": "CSR1000V",  
        "chassis_sn": "960TYDEHPIE",  
        "compiled_by": "mcpred",  
        "compiled_date": "Thu 30-Jan-20 18:48",  
        "copyright_years": "1986-2020",  
        "curr_config_register": "0x2102",  
        "disks": {  
            "bootflash..": {  
                "disk_size": "7774207",  
                "type_of_disk": "virtual hard disk"  
            },  
            "nvram": {  
                "disk_size": "1048576",  
                "type_of_disk": "NVRAM"  
            }  
        },  
        "mac": "00:0C:29:00:00:00",  
        "model": "CSR1000V",  
        "os": "ios",  
        "os_version": "15.1(4)M",  
        "serial": "960TYDEHPIE",  
        "sw_ver": "15.1(4)M",  
        "type": "CSR1000V",  
        "uptime": "0 days, 0 hours, 0 minutes, 0 seconds"  
    },  
    "vrf": {  
        "management": {  
            "ip": "10.0.0.1",  
            "mask": "255.255.255.0",  
            "name": "management",  
            "rd": null,  
            "rt": null  
        }  
    },  
    "interfaces": {  
        "GigabitEthernet0/0": {  
            "ip": "10.0.0.1",  
            "mask": "255.255.255.0",  
            "name": "GigabitEthernet0/0",  
            "status": "Up",  
            "type": "Physical",  
            "vrf": "management"  
        }  
    },  
    "ospf": {  
        "processes": {  
            "0": {  
                "area": "0.0.0.0",  
                "cost": 1, "dead": 40, "hello": 10, "mtu": 1500, "name": "0",  
                "networks": {  
                    "10.0.0.0/24": {  
                        "area": "0.0.0.0",  
                        "cost": 1, "dead": 40, "hello": 10, "mtu": 1500, "name": "0",  
                        "network": "10.0.0.0/24",  
                        "status": "Up",  
                        "type": "BROADCAST",  
                        "vif": "GigabitEthernet0/0",  
                        "vif_ip": "10.0.0.1",  
                        "vif_name": "GigabitEthernet0/0",  
                        "vif_vrf": "management"  
                    }  
                },  
                "name": "0",  
                "status": "Up",  
                "type": "OSPFv2",  
                "vrf": "management"  
            }  
        }  
    },  
    "routed_interfaces": {  
        "GigabitEthernet0/0": {  
            "ip": "10.0.0.1",  
            "mask": "255.255.255.0",  
            "name": "GigabitEthernet0/0",  
            "status": "Up",  
            "type": "Physical",  
            "vrf": "management"  
        }  
    },  
    "arp_table": {  
        "00:0c:29:00:00:00": {  
            "ip": "10.0.0.1",  
            "mac": "00:0c:29:00:00:00",  
            "name": "CSR1kv",  
            "status": "Up",  
            "type": "Host",  
            "vrf": "management"  
        }  
    },  
    "bgp": {  
        "processes": {}  
    },  
    "l2vpn": {  
        "processes": {}  
    },  
    "isis": {  
        "processes": {}  
    },  
    "lacp": {  
        "processes": {}  
    },  
    "ldp": {  
        "processes": {}  
    },  
    "mpls": {  
        "processes": {}  
    },  
    "rsvp": {  
        "processes": {}  
    },  
    "vxlan": {  
        "processes": {}  
    }  
}
```

Passo 4 Use Genie to Compare Configurations

1. Add an IPv6 address to CSR1kv

```
CSR1kv(config)#interface gig 1
CSR1kv(config-if)#ipv6 address 2001:db8:acad:56::101/64
CSR1kv(config-if)#
*May  7 15:56:22.761: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: cisco] [source: 192.168.56.1] [localport: 22] at 15:56:22 UTC Sat May  7 2022
CSR1kv(config-if)#
*May  7 15:56:24.833: %SYS-6-LOGOUT: User cisco has exited tty session 1(192.16.56.1)
CSR1kv(config-if)#
*May  7 16:01:02.319: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: cisco] [source: 192.168.56.1] [localport: 22] at 16:01:02 UTC Sat May  7 2022
CSR1kv(config-if)#
*May  7 16:01:04.517: %SYS-6-LOGOUT: User cisco has exited tty session 1(192.16.56.1)
```

2. Use Genie to verify configuration and parse output in JSON

3. Use cat to examine the contents of the _console.txt file

```
(dtlab) devasc@labvm:~/Desktop/dtlab/verify_ipv6_1$ cat CSR1kv_show-ipv6-interface-gig-1_console.txt
+++ CSR1kv with via 'cli': executing command 'show ipv6 interface gig 1' ***
show ipv6 interface gig 1
GigabitEthernet1 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::A00:27FF:FE0A:9E1
    No Virtual link-local address(es):
    Description: VBox
    Global unicast address(es):
        2001:DB8:ACAD:56::101, subnet is 2001:DB8:ACAD:56::/64
    Joined group address(es):
        FF02::1
        FF02::1:FF00:101
        FF02::1:FF0A:9E1
    MTU is 1500 bytes
    ICMP error messages limited to one every 100 milliseconds
    ICMP redirects are enabled
    ICMP unreachables are sent
    ND DAD is enabled, number of DAD attempts: 1
    ND reachable time is 30000 milliseconds (using 30000)
    ND NS retransmit interval is 1000 milliseconds
CSR1kv#
(dtlab) devasc@labvm:~/Desktop/dtlab/verify_ipv6_1$
```

```
(dtlab) devasc@labvm:~/Desktop/dtlabs$ cat verify_ipv6_1/*  
  
2022-05-07 16:00:54,377: %UNICON-INFO: +++ CSR1kv logfile verify_ipv6_1/connection_CSR1kv.txt +++  
  
2022-05-07 16:00:54,377: %UNICON-INFO: +++ Unicon plugin iosxe (unicon.plugins-iosxe) +++  
Password:  
  
2022-05-07 16:00:54,445: %UNICON-INFO: +++ connection to spawn: ssh -l cisco -o KexAlgorithms=diffie-hellman-group14-sha1  
192.168.56.101, id: 140325986952480 +++  
  
2022-05-07 16:00:54,445: %UNICON-INFO: connection to CSR1kv  
  
*  
**  
***  
*** Cisco Networking Academy  
***  
*** This software is provided for  
*** Educational Purposes  
*** Only in Networking Academies  
***  
**  
*  
  
CSR1kv#  
  
2022-05-07 16:00:54,505: %UNICON-INFO: +++ initializing handle +++
```

4. Modify the IPv6 Link-Local address.

```
CSR1kv(config-if)# ipv6 address fe80::56:1 link-local
```

5. Use Genie to verify configuration and parse output in JSON

```
(dtlab) devasc@labvm:~/Desktop/dtla$ genie parse "show ipv6 interface gig 1" --testbed-file yaml/testbed.yml --devices CSR1kv --output verify_ipv6_2
Using the default YAML encoding key since no key was specified in configuration.
THIS IS A SHARED KEY AND IS NOT SECURE, PLEASE RUN `pyats secret keygen` AND ADD TO YOUR pyats.conf FILE BEFORE ENCODING ANY VALUES.
2022-05-07T16:04:42: %UNICON-WARNING: Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring
Device 'CSR1kv' connection 'cli' does not have IP and/or port specified, ignoring
100% | 1/1 [00:01<00:00, 1.00s/it]
+-----+
| Genie Parse Summary for CSR1kv
+-----+
| Connected to CSR1kv
| - Log: verify_ipv6_2/connection_CSR1kv.txt
| -----
| Parsed command 'show ipv6 interface gig 1'
| - Parsed structure: verify_ipv6_2/CSR1kv_show-ipv6-interface-gig-1_parsed.txt
| - Device Console: verify_ipv6_2/CSR1kv_show-ipv6-interface-gig-1_console.txt
| -----
```

6. Use cat to examine the contents of the _console.txt file and _parsed.txt

```
devasc@labvm:~/Desktop/dtlab
File Edit View Search Terminal Help
CSR1kv#  
  
2022-05-07 16:04:44,691: %UNICON-INFO: +++ CSR1kv with via 'cli': executing command 'show ipv6 interface gig 1' +++
show ipv6 interface gig 1
GigabitEthernet1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::56:1
  No Virtual link-local address(es):
  Description: VBox
  Global unicast address(es):
    2001:DB8:ACAD:56::101, subnet is 2001:DB8:ACAD:56::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:101
    FF02::1:FF56:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachable messages are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND NS retransmit interval is 1000 milliseconds
CSR1kv#
+++ CSR1kv with via 'cli': executing command 'show ipv6 interface gig 1' +++
show ipv6 interface gig 1
GigabitEthernet1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::56:1
  No Virtual link-local address(es):
  Description: VBox
  Global unicast address(es):
    2001:DB8:ACAD:56::101, subnet is 2001:DB8:ACAD:56::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:101
    FF02::1:FF56:1
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ICMP unreachable messages are sent
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds (using 30000)
  ND NS retransmit interval is 1000 milliseconds
CSR1kv#
{
  "GigabitEthernet1": {
```

```
    "enabled": true,
    "ipv6": {
      "2001:DB8:ACAD:56::101/64": {
        "ip": "2001:DB8:ACAD:56::101",
        "prefix_length": "64",
        "status": "valid"
      },
      "FE80::56:1": {
        "ip": "FE80::56:1",
        "origin": "link_layer",
        "status": "valid"
      }
    },
    "enabled": true,
    "icmp": {
      "error_messages_limited": 100,
      "redirects": true,
      "unreachables": "sent"
    },
    "nd": {
      "dad_attempts": 1,
      "dad_enabled": true,
      "ns_retransmit_interval": 1000,
      "reachable_time": 30000,
      "suppress": false,
      "using_time": 30000
    }
  },
  "joined_group_addresses": [
    "FF02::1",
    "FF02::1:FF00:101",
    "FF02::1:FF56:1"
  ],
  "mtu": 1500,
  "oper_status": "up"
},
"_exclude": []
```

7. Use Genie to compare the difference between the configurations

```
[dtlab] devasc@labvm:~/Desktop/dtlab$ genie diff verify_ipv6_1 verify_ipv6_2  
lit [00:00, 760.53it/s]  
+=====+  
| Genie Diff Summary between directories verify_ipv6_1/ and verify_ipv6_2/ |  
+=====+  
| File: CSR1kv_show-ipv6-interface-gig-1_parsed.txt |  
| - Diff can be found at ./diff_CSR1kv_show-ipv6-interface-gig-1_parsed.txt |  
+-----+
```

8. Use cat to view the contents of the file with the differences

```
(dtlab) devasc@labvm:~/Desktop/dtlab$ cat ./diff_CSR1kv_show-ipv6-interface-gig-1_parsed.txt && echo "\n"  
--- verify_ipv6_1/CSR1kv_show-ipv6-interface-gig-1_parsed.txt  
+++ verify_ipv6_2/CSR1kv_show-ipv6-interface-gig-1_parsed.txt  
GigabitEthernet1:  
    ipv6:  
    + FE80::56:1:  
    + ip: FE80::56:1  
    + origin: link_layer  
    + status: valid  
    - FE80::A00:27FF:FE0A:9E1:  
    - ip: FE80::A00:27FF:FE0A:9E1  
    - origin: link_layer  
    - status: valid  
    joined_group_addresses:  
    - index[2]: FF02::1:FF0A:9E1  
    + index[2]: FF02::1:FF56:1\n
```

Passo 5

1. Lab Cleanup and Further Investigation

```
(dtlab) devasc@LabVM:~/Desktop/dtLab$ deactivate  
devasc@LabVM:~/Desktop/dtLab$ sudo shutdown now
```

Thank for your Attention

