

SwiNOG 32

NETWORK UNIT TESTING

Nuts (Network Unit Testing System) with SaltStack

Urs Baumann

Network Engineering

Bern, 09. November 2017

Urs Baumann
Network Engineer
Python Developer

**SDN, Cloud Infrastructure, OpenStack,
Python, Cisco and Brocade Instructor**

urs.baumann@ins.hsr.ch

 @ubaumann_ch

<http://ins.hsr.ch>

 @INSHSR



- NUTS
- NAPALM
- SaltStack



■ How do you test your network after a change?

■ Today

- ping 8.8.8.8 ✓
- facebook.com ✓
- youtube.com ✓
- e-mail to my private address ✓
- print next train connection ✓

■ Rest of my network life

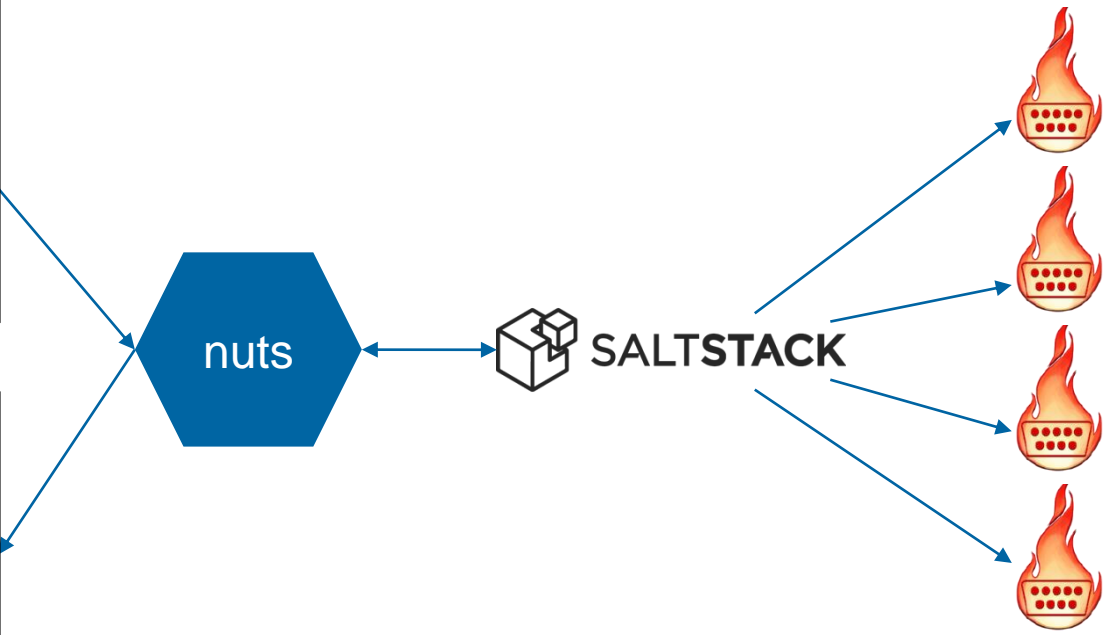
- Nuts (Network Unit Test System)

Nuts (Network Unit Testing System)

```
(nuts)[ubaumann@localhost nuts]$ cat ../mytest.yml
- name: bandwidth_ping
  command: bandwidth
  devices: 'git*'
  parameter: ['{{ ip[0] }}']
  operator: '<'
  expected: 100000000
  setup:
  - command: network.ip_addrs
    devices: 'git*'
    save: ip
  - command: cmd.run
    devices: 'git*'
    parameter: ['iperf3 -s -D -1']
  teardown:
  - command: cmd.run
    parameter: ['pkill iperf3']
    devices: 'git*'
- name: versionchk_01
  command: checkversion
  devices: 'c*'
  parameter: []
  operator: '='
  expected: 'CSR1000V Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 15.5(2)S, RELEASE SOFTWARE (fc3)'
```

```
(nuts)[ubaumann@localhost nuts]$ python main.py ../mytest.yml -c /home/ubaumann/config_nuts.yml
TestCases:
Async Tests:
Name: versionchk_01, Command: checkversion, Devices: c*. Parameter: [], Operator: =, Expected: CSR1000V Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 15.5(2)S, RELEASE SOFTWARE (fc3)
Sync Tests:
Name: bandwidth_ping, Command: bandwidth, Devices: git*, Parameter: ['{{ ip[0] }}'], Operator: <, Expected: 100000000
Start test versionchk_01
Started test 1 of 1
-----Started all tests-----
CollectResult of Test versionchk_01
Collected results from 1 of 1 tests
-----Collected all results-----
Start Test bandwidth_ping

versionchk_01: Test passed -----
bandwidth_ping: Test passed -----
-----Summary-----
2 out of 2 tests passed
```



<https://github.com/HSRNetwork/Nuts>

■ Command for network devices (= < > not)

- connectivity
- traceroute
- interfacestatus
- interfacespeed
- arp
- checkversion
- checkuser

■ Comming soon

- lldp_neighbor
- bgp_neighbor
- ospf_neighbor
- route_to
- stp_root

■ Command for debian systems (= < > not)

- connectivity
- traceroute
- dnscheck
- dhcpcheck
- webresponse
- portresponse
- bandwidth

arp

- name: example_arp
command: arp
devices: cisco.csr.1000v
parameter: [192.168.16.128]
operator: '='
expected: '00:0C:29:EA:D1:68'

interfacestatus

- name: example_interfacestatus
command: interfacestatus
devices: cisco.csr.1000v
parameter: [GigabitEthernet1]
operator: '='
expected: True

bandwidth

- name: bandwith_ping
command: bandwidth
devices: srvlnx0001
parameter: ['{{ ip[0] }}']
operator: '<'
expected: 100000000
setup:
 - command: network.ip_addrs
devices: srvlnx0099
save: ip
 - command: cmd.run
devices: srvlnx0099
parameter: ['iperf3 -s -D -1']
- teardown:
 - command: cmd.run
parameter: ['pkill iperf3']
devices: srvlnx0099

dhcpcheck

- name: dhcp_test
command: dhcpcheck
devices: srvlnx0014
parameter: ['10.10.10.11']
operator: '='
expected: True

- **Network Automation and Programmability Abstraction Layer with Multivendor support**
- <https://napalm.readthedocs.io/>
- <https://napalm-automation.net/>
- **Support matrix**
 - <https://napalm.readthedocs.io/en/latest/support/index.html>



```
from napalm import get_network_driver

driver = get_network_driver('ios')
device = driver('152.96.9.201', 'cisco', 'cisco')

device.open()
arp = device.get_arp_table()
device.close()

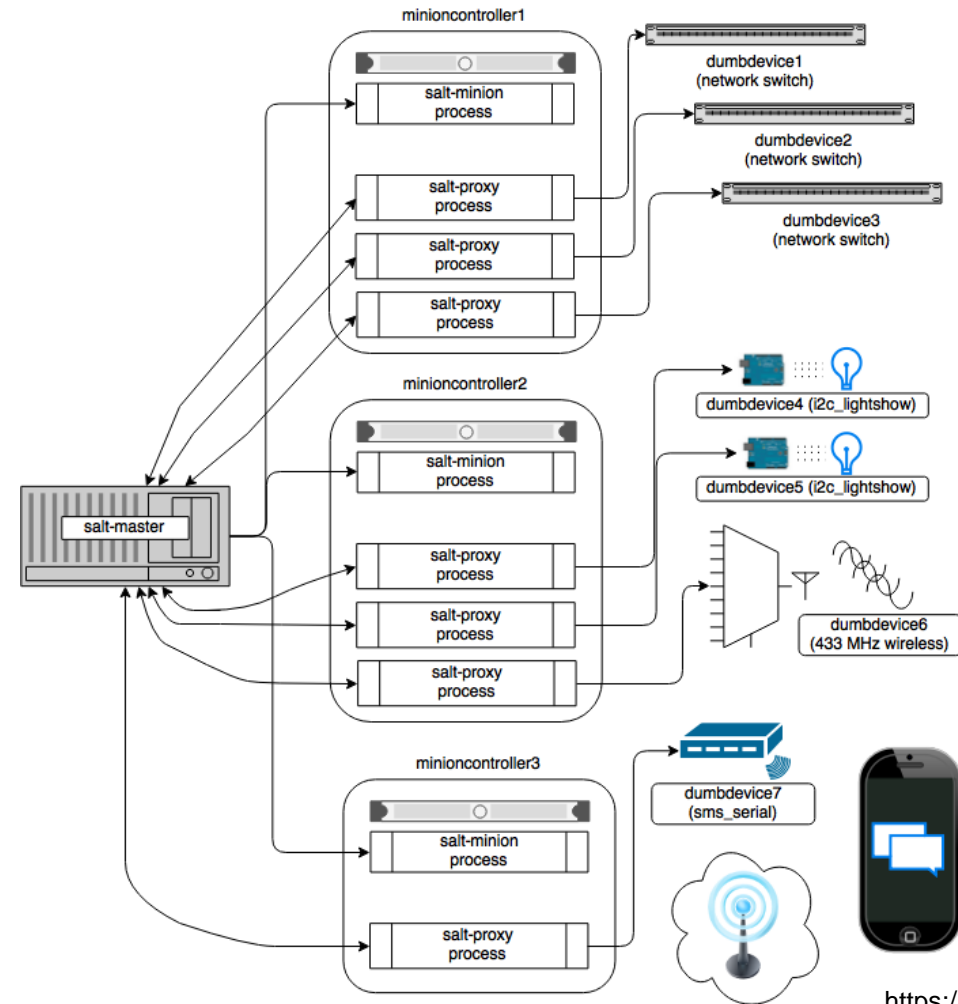
print(arp)
```

Support Matrix

	EOS	IOS	IOSXR	JUNOS	NXOS	NXOS_SSH
get_arp_table	✓	✓	✓	✓	✓	✓
get_bgp_config	✓	✗	✓	✓	✗	✗
get_bgp_neighbors	✓	✓	✓	✓	✓	✓
get_bgp_neighbors_detail	✓	✗	✓	✓	✗	✗
get_config	✓	✓	✓	✓	✓	✓
get_environment	✓	✓	✓	✓	✗	✗
get_facts	✓	✓	✓	✓	✓	✓
get_firewall_policies	✗	✗	✗	✗	✗	✗
get_interfaces	✓	✓	✓	✓	✓	✓
get_interfaces_counters	✓	✓	✓	✓	✗	✗
get_interfaces_ip	✓	✓	✓	✓	✓	✓
get_ipv6_neighbors_table	✗	✗	✗	✗	✗	✗
get_lldp_neighbors	✓	✓	✓	✓	✓	✓
get_lldp_neighbors_detail	✓	✓	✓	✓	✓	✓

get_mac_address_table	✓	✓	✓	✓	✓	✓
get_network_instances	✓	✗	✗	✓	✗	✗
get_ntp_peers	✗	✗	✓	✓	✓	✓
get_ntp_servers	✓	✓	✓	✓	✓	✓
get_ntp_stats	✓	✓	✓	✓	✓	✗
get_optics	✓	✓	✗	✓	✗	✗
get_probes_config	✗	✗	✓	✓	✗	✗
get_probes_results	✗	✗	✓	✓	✗	✗
get_route_to	✓	✗	✓	✓	✗	✗
get_snmp_information	✓	✓	✓	✓	✓	✓
get_users	✓	✗	✓	✓	✓	✓
is_alive	✓	✓	✓	✓	✓	✓
ping	✓	✓	✗	✓	✗	✗
traceroute	✓	✓	✓	✓	✓	✓

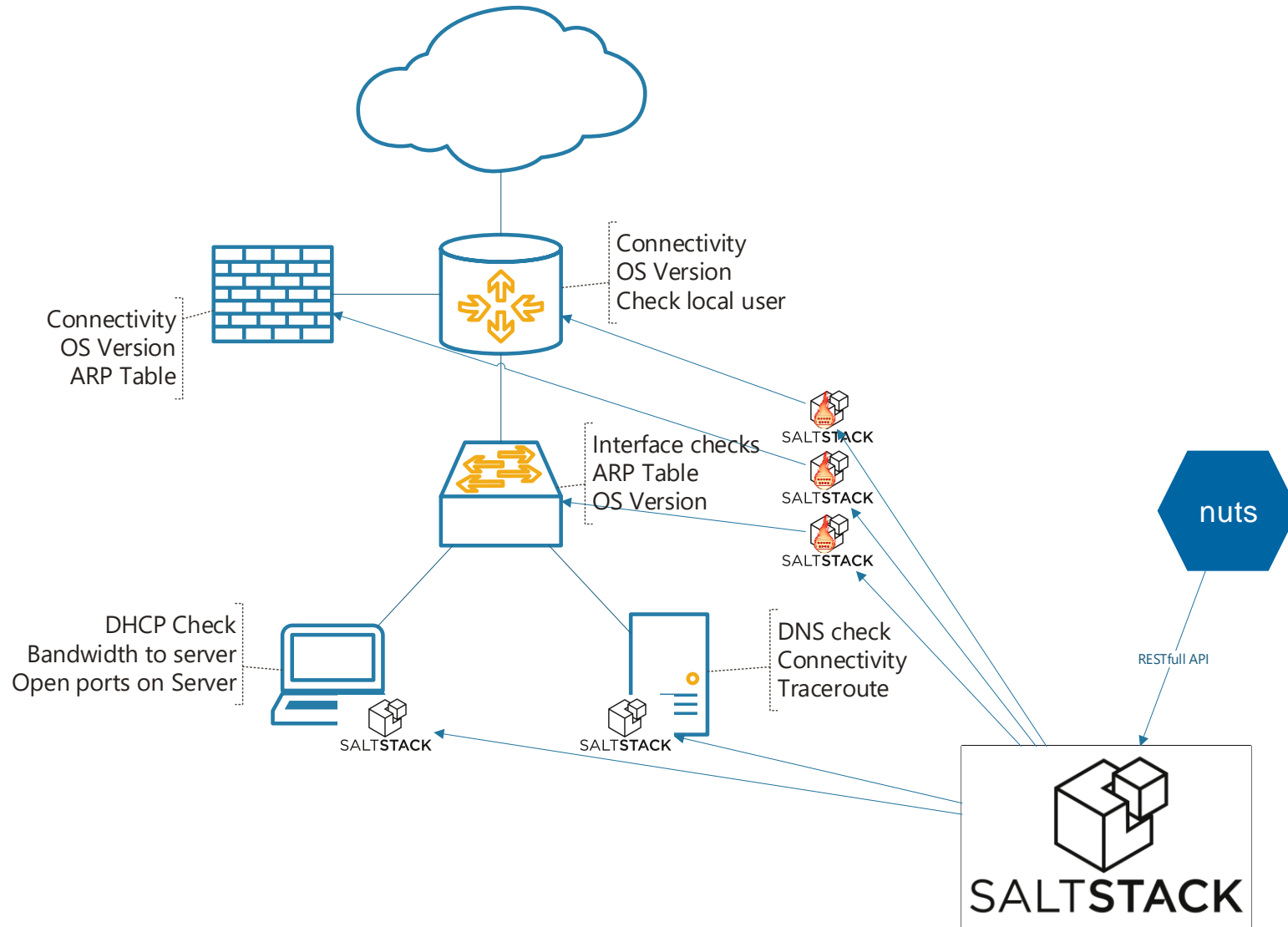
- Python
- YAML
- Jinja2
- Easy to extend
- API
- Event bus
- Agent-based / agent-less
- Proxy minions



<https://docs.saltstack.com>

```
root@sr-000001:~# salt 'sw*' ntp.servers
sw8262-d:
-----
comment:
out:
  - time0.ins.hsr.ch
  - 152.96.120.60
  - 152.96.120.53
result:
  True
sw8262-e:
-----
comment:
out:
  - time0.ins.hsr.ch
  - 152.96.120.60
  - 152.96.120.53
result:
  True
sw8262-b:
-----
comment:
out:
```

```
root@sr-000001:~# salt 'sw*' ntp.set_servers time1.ins.hsr.ch
sw8262-b:
-----
already_configured:
  False
comment:
diff:
  +ntp server time1.ins.hsr.ch
loaded_config:
result:
  True
sw8262-e:
-----
already_configured:
  False
comment:
diff:
  +ntp server time1.ins.hsr.ch
loaded_config:
result:
  True
sw8262-a:
-----
```

■ NUTS

- <https://nuts.readthedocs.io/>
- <https://github.com/HSRNetwork/Nuts/>

■ NAPALM

- <https://napalm.readthedocs.io/>
- <https://napalm-automation.net/>

■ SaltStack

- <https://docs.saltstack.com/>

■ Slack <https://networktocode.slack.com/>

- #nuts
- #napalm
- #saltstack

