



ANSTE

Advanced Network Service Testing Environment

2nd of February 2014

Context

Zentyal Server

- Zentyal is a drop-in replacement for Microsoft Small Business Server and Microsoft Exchange Server
- But not only that...
 - Gateway & UTM
 - Infrastructure Server
 - Office Server
 - Communications Server

Testing a complex stack

- Based on Ubuntu Server
- Integrate other packages and customize configuration
- Ubuntu updates can break integration
- Other external dependencies:
 - Squid/Dansguardian rules
 - Snort/Suricata rules
 - Spamassassin rules



Testing complex scenarios

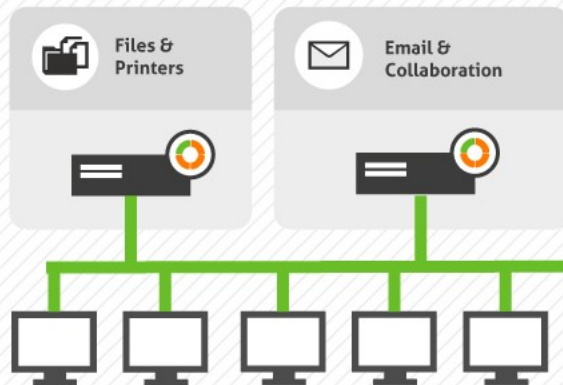
ENTERPRISE NETWORK INFRASTRUCTURE

300 users

Robust.
Easy to scale.

More complex to manage.
More hardware
More electricity costs.

OFFICE 1



Network infrastructure

Users

Internet access & Firewall

Network security

File sharing

User, Groups and Permissions management

Disaster recovery

Automatic server synchronization

Internet access & Firewall

Users, Files & Printers

Network infrastructure




Email & Collaboration

OFFICE 2

REMOTE ACCESS

Internet

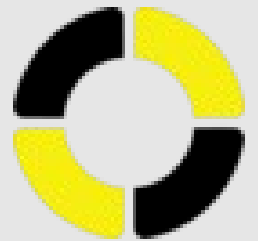
Testing in Zentyal

- Unit testing → 
- Integration testing → 
- Functional testing → 

Solution

ANSTE

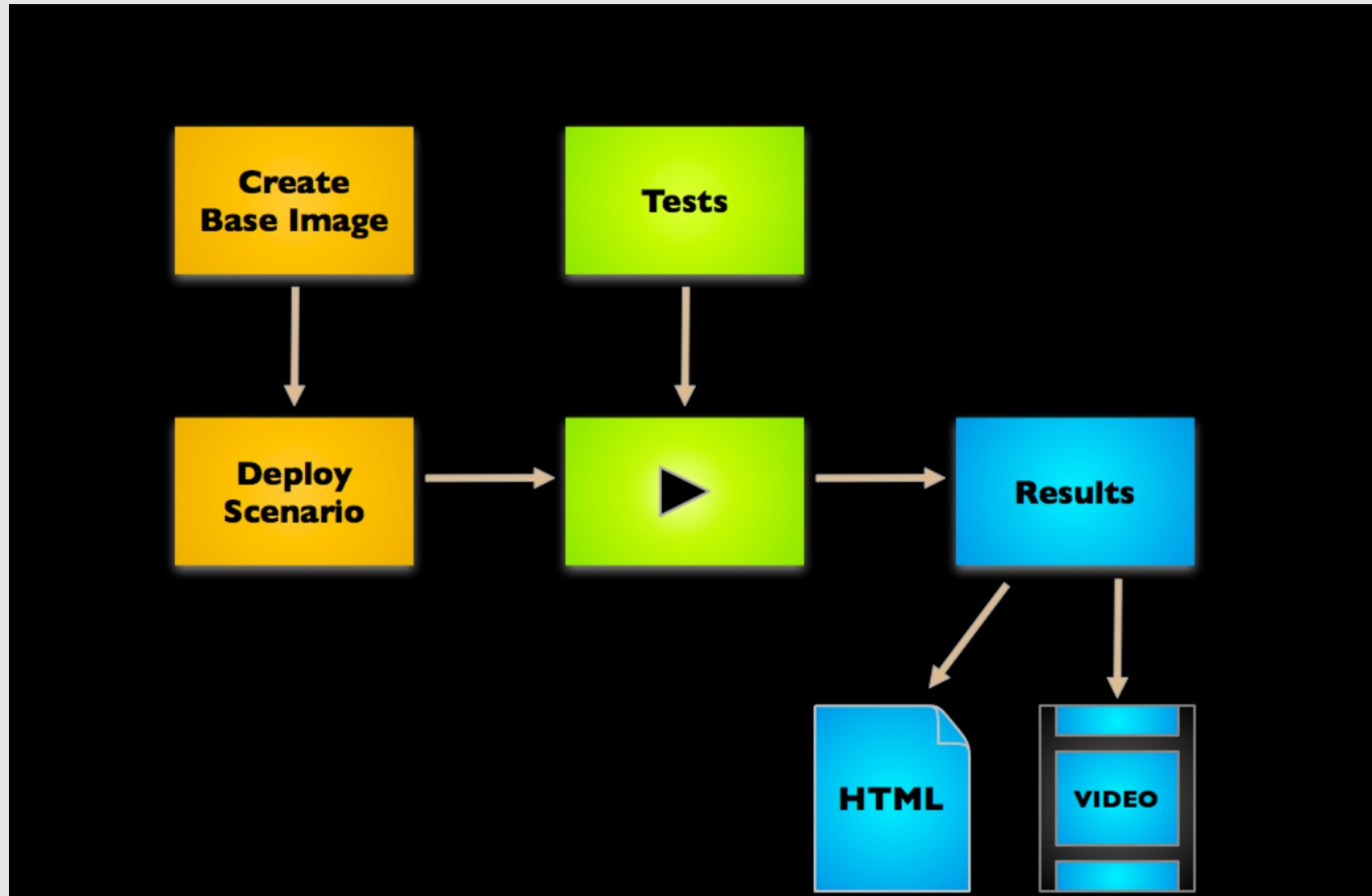
- Advanced Network Service Testing Environment
- 2007
- Open source tool - GPL
- Written in Perl
- Virtualization Backends
 - Libvirt (KVM)
- Scenarios/Tests Format - YAML
- Code - <https://github.com/Zentyal/anste>



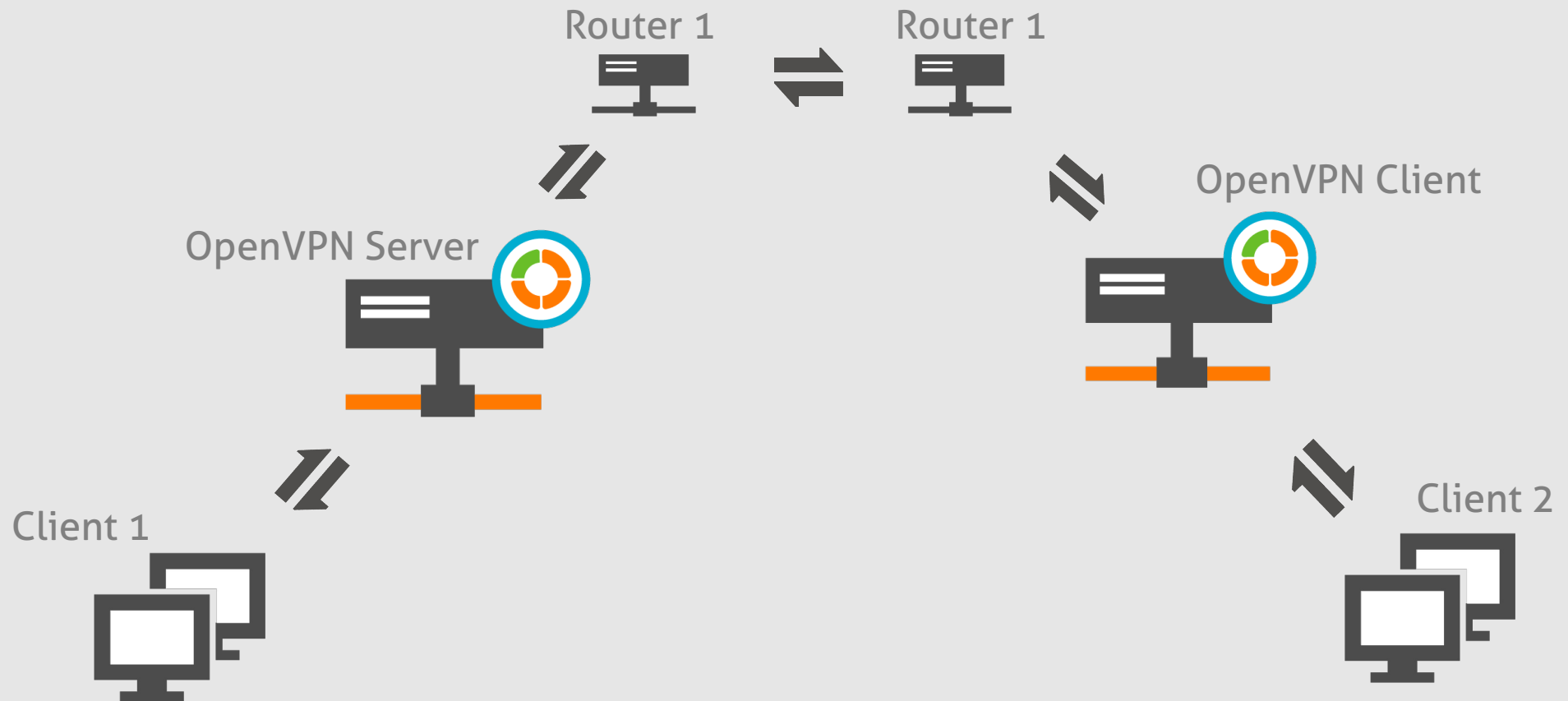
What ANSTE does?

- **Deploy Scenarios**
- **Run tests**
- **Gather results**

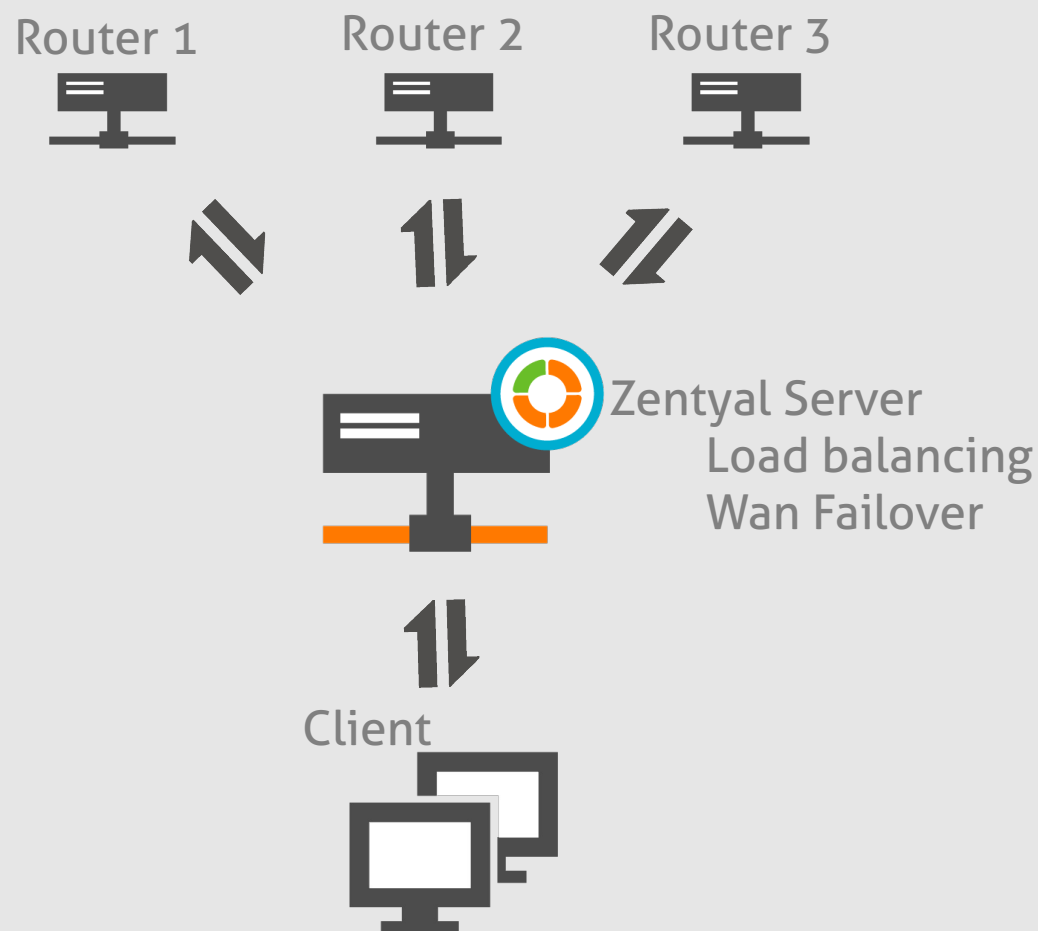
What ANSTE does?



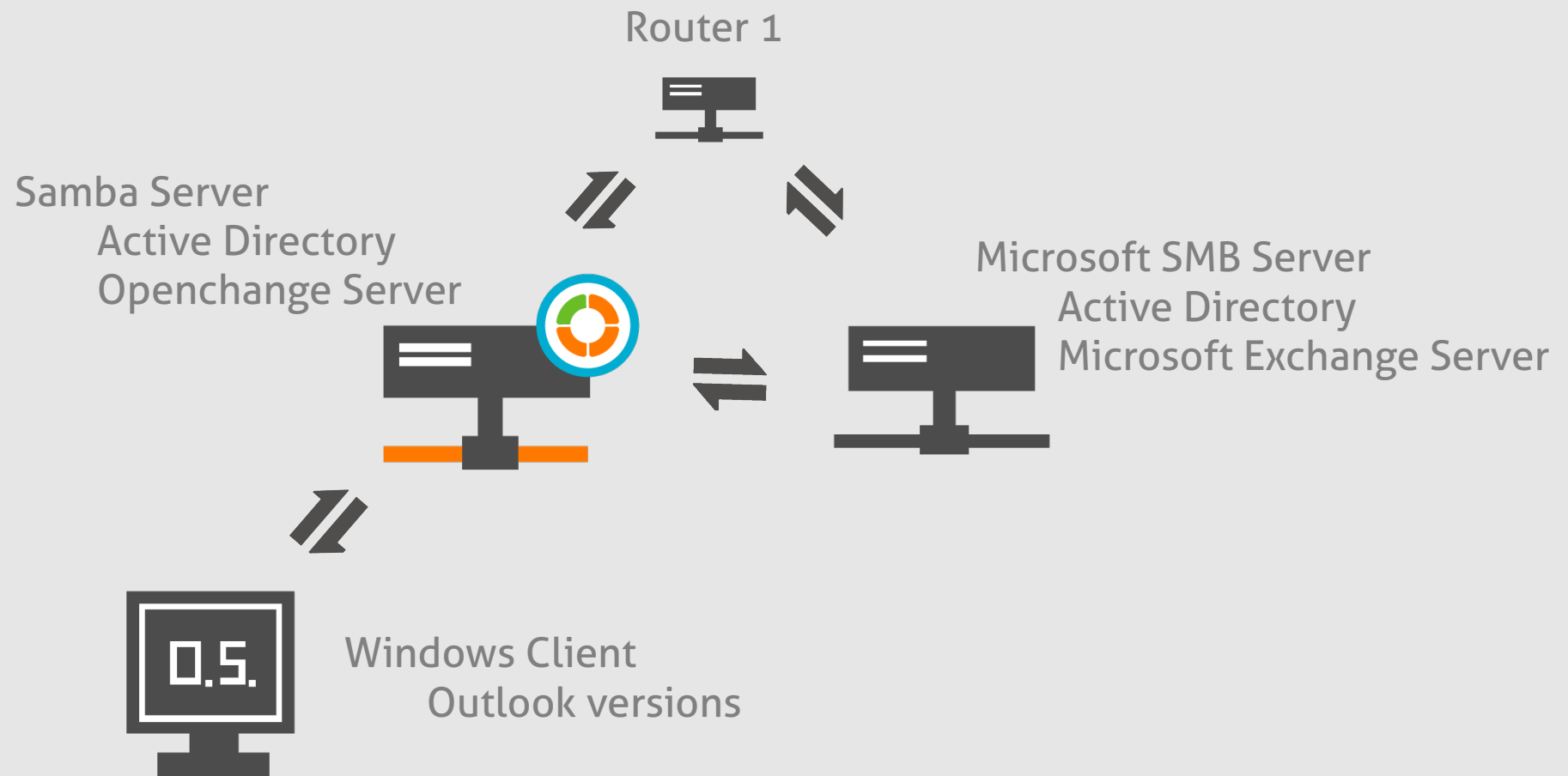
Example of testing scenarios



Example of testing scenarios



Example of testing scenarios



Internals

Create base Image

- **Ubuntu base images**
 - Created with **ubuntu-vm-builder**
 - Added a daemon to receive tests and check status
 - Package list installation
 - Configuration and provisioning via script
- **Support for “raw” images (ALPHA)**
 - Any other OS
 - Only launch
 - Already setup

Create base Image

```
name: precise-mini
desc: Ubuntu Precise
memory: 256
size: 1150
method: debootstrap
dist: precise
pre-install:
  - conf-apt-cacher.sh
  - write-precise-sources.sh
packages:
  - openssh-server
  - wget
  - smbclient
post-install:
  - install-last-wget-version.sh
```

```
name: precise-zentyal-3.3
desc: Ubuntu Precise with Zentyal
memory: 2000
size: 3000
method: debootstrap
dist: precise
packages:
  - vim
  - strace
  - apt-transport-https
  - apparmor
  - zentyal
  - debug
pre-install:
  - conf-apt-cacher.sh
  - write-precise-zentyal-sources-3.3.sh
  - install-mysql.sh
files:
  - webadmin.preservice
post-install:
  - set-english-language.sh
  - ssl-cert-workaround.sh
  - delete-first.sh
  - force-package-install.sh
  - add-anste-firewall-rule
  - ebox-same-cert.sh
  - config-auto-login.sh
  - ebox-set-debug-on.sh
  - zentyal-ignore-anste-iface.sh
post-tests:
  - check-zentyal-log
  - check-syslog-apparmor
```


Deploy Scenario

- **Create network**
 - **Complex network**
 - **Virtual interfaces**
- **Launch Images**
 - **Configuration scripts**
 - **Pre-populate data**
 - **Configure basic services**
 - **Special types:**
 - **Router → Special config (DHCP, PPPOE, ...)**

Deploy Scenario

```
name: Zentyal to Zentyal OpenVPN Test Scenario
desc: Two Zentyal with one router and one internal machine each.
hosts:
  - name: router-a
    type: router
    desc: Host that acts like a router
    baseimage: {$dist}-mini
    network:
      interfaces:
        - name: eth1
          type: static
          address: 192.168.3.2
          netmask: 255.255.255.0
        - name: eth2
          type: static
          address: 192.168.4.2
          netmask: 255.255.255.0

  - name: router-b
    type: router
    desc: Host that acts like a router
    baseimage: {$dist}-mini
    network:
      interfaces:
        - name: eth1
          type: static
          address: 192.168.5.2
          netmask: 255.255.255.0
        - name: eth2
          type: static
          address: 192.168.4.3
          netmask: 255.255.255.0
```

Deploy Scenario

```
name: Zentyal and client
desc: One Zentyal machine and a user cliente machine
manual-bridging: 1
bridges:
  - id: 2
    address: 192.168.2.254
  - id: 3
    address: 192.168.3.254
hosts:
  - name: zentyal-server
    desc: Zentyal Server
    baseimage: ${dist}-zentyal${$image}
    network:
      interfaces:
        - name: eth1
          type: static
          bridge: 2
          address: 192.168.2.1
          netmask: 255.255.255.0
        - name: eth2
          type: static
          bridge: 3
          address: 192.168.3.1
          netmask: 255.255.255.0
          gateway: 10.6.7.1
    pre-install:
      - conf-apt-cacher.sh
      - dist-upgrade-packages${$script}.sh
    post-install:
      - zentyal-import-network.pl
      - wait-start-apache.sh
```

Deploy Scenario

```
- name: dhcp-router
  type: dhcp-router
  desc: Host that acts like a DHCP router
  baseimage: {$dist}-mini
  network:
    interfaces:
      - name: eth1
        type: static
        address: 192.168.2.1
        netmask: 255.255.255.0
        bridge: 2

- name: pppoe-router
  type: pppoe-router
  desc: Host that acts like a PPPoE router
  baseimage: {$dist}-mini
  network:
    interfaces:
      - name: eth1
        type: static
        address: 192.168.2.1
        netmask: 255.255.255.0
        bridge: 2
```

Deploy Scenario

```
- name: WinXP
  desc: Windows XP
  baseimage: WinXP
  baseimage-type: raw

- name: Windows7
  desc: Windows 7
  baseimage: Windows7
  baseimage-type: raw

- name: Win2k3
  desc: Windows Server 2003
  baseimage: Win2k3
  baseimage-type: raw
```

Run Tests

- Any scripting language (Bash, Perl, Python, ...)
- Special types of tests: (selenium/webdriver, reboot, ...)
- Run in any of the machines
 - Even the host machine
- Different modes:
 - Step by step → Debugging
 - Breakpoints
 - Wait on failure

Run Tests

```
name: Zentyal to Zentyal tunnel OpenVPN tests suite
desc: Tests to ensure that Zentyal OpenVPN module works ok.
scenario: openvpn/zentyal-to-zentyal.yaml
tests:
  - name: BasicConfigServer
    type: web
    desc: Remove default gateway, set interface as external, enable modules, add new gateway and set certificates.
    host: zentyal-server
    script: basic-config
    vars:
      NAME: default
      IFACE: "eth2"
      MODULES: firewall logs openvpn
      IP: "192.168.3.2"
      DEFAULT: "True"

  - name: ConfigVPNServer
    type: web
    desc: Configure VPN server in the Zentyal server
    host: zentyal-server
    script: config-vpn-server

  - name: DeleteAdvertisedNetwork
    type: web
    desc: Delete advertised network in the Zentyal server
    host: zentyal-server
    script: delete-advertised-network
    vars:
      SERVER: "foobar-server"
      NETWORK: "10.6.7.0"
```

Run Tests

```
#!/usr/bin/env python

import zentyal

driver = zentyal.driver()
server_name = driver.var('SERVER', 1)
advertised_network = driver.var('NETWORK', 1)

driver.open('/CA/Index')
driver.go_to('VPN -> Servers')

driver.table_filter('Servers', server_name)
xpath_exposed_networks = "//div[@id='Servers']//a[contains(@href,'ExposedNetworks')]"
driver.click(xpath=xpath_exposed_networks)

driver.wait_for(name='ExposedNetworks')
driver.table_filter('ExposedNetworks', advertised_network)
driver.click(xpath="//div[@id='ExposedNetworks']//button[@name='del']")

deleted_xpath = "//div[@class='note' and text()='Advertised network deleted']"
found_deleted = driver.wait_for(xpath=deleted_xpath)
driver.assert_true(found_deleted)
```

```
#!/bin/bash

echo "Expecting foo on $HOST:1100"

for i in `seq 1 30`
do
    nc $HOST 1100 | grep "foo"
    if [ $? != 0 ]
    then
        echo "$i try"
        sleep 1
    else
        echo "Success"
        exit 0
    fi
done

exit 1
```


Gather results

- **Auto-generated reports**
 - Basics easy to read HTML reports
 - XML reports for CI integration (Jenkins)
- **Record video & Image capture**
 - For web UI tests
 - `recordmydesktop`

Gather results

Zentyal OpenChange tests

Contains a set of tests to check that the Zentyal OpenChange module works properly.

Test	Description	Result
InstallNonProfilePackages	Install additional packages	OK (script)
EnableModules	Enable modules.	OK (script)
CreateVDomain	Create the virtual mail domain	OK (script)
SaveChanges	Save the changes.	OK (script)
AddUser	Adds a user	OK (script)
Provision	OpenChange provision and enable all users account	OK (script)
LoginUser1	Log into roundcube with user1 and then logout	OK (script)
Mail	Access the main mail screen	OK (script)
ComposeMail	Access the compose mail screen	OK (script)
AddContactZentyal	Adds a contact in Zentyal	OK (script)
check-zentyal-log	PostTest added from the baseImage of the host zentyal-server	ERROR (script)

Gather results

Test Result : Zentyal Network Multi-Gateway Tests

0 failures (±0)

11 tests (±0)

Took 1 min 47 sec.

 [add description](#)

All Tests

Test name	Duration	Status
AddGateways	16 sec	Passed
ConfigMultiGateway	26 sec	Passed
ConfigNetwork	6 sec	Passed
DumpRoutes	1 sec	Passed
DumpRoutes2	1 sec	Passed
EnableBalance	6 sec	Passed
EnableModules	26 sec	Passed
TestBalanceRouterA	12 sec	Passed
TestBalanceRouterB	11 sec	Passed
check-syslog-apparmor	1 sec	Passed
check-zentyal-log	1 sec	Passed

What's next ?

What's next...

- **Integration with OpenStack**
 - **On its way...**
- **Improving Windows integration**
 - **Windows service for ANSTE**
 - **Sikuli integration**
 - **Windows automatic provision**

We are hiring!!!

- **Two open positions:**
 - **R&D Senior C/C++ Developer**
 - **R&D Junior C/C++ Developer**
- **www.zentyal.com/company/careers/**



Questions?

Julio J. Garcia Martin,

QA Engineer

jjgarcia@zentyal.com

www.zentyal.com