All PE meeting - Summer 2018 - Project Network Tools

1. Objective

The objective of this project is to "build" a ZTP (Zero Touch Provisioning) server functionality in your Pi that will "push" the minimal configuration to a switch/router.

This minimal configuration includes:

- Management IP
- User credentials for API access
- Enable API access for configuration changes

Once the minimal configuration is loaded in the device, the rest of the configuration will be sent using a python program that you will have to write using the available libraries for that purpose.

In addition, you will need to write a "route server" functionality in your Pi in order send the necessary information to the switches:

- Add new prefix networks (both IPv4 and IPv6)
- Modify parameters to existing ones (AS_PATH, MED, communities, next hop, etc)
- Remove prefix networks (both IPv4 and IPv6)
- Display the BGP table
- Display the routing table
- Display parameters of an specific prefix (IPv4 or IPv6)

Each switch/Router has 2 additional interfaces (plus management).

The rest of the configuration will include:

- Base configuration:
  - Hostname
  - Enable IPv4 and IPv6 routing
  - Additional user credentials/passwords
- Additional interfaces configuration (IP and IPv6 addresses, description, etc).
- Routing protocol configuration (BGP)

Details like IP addressing, password, hostnames, ASNs, flavor of BGP is up to your group to define.

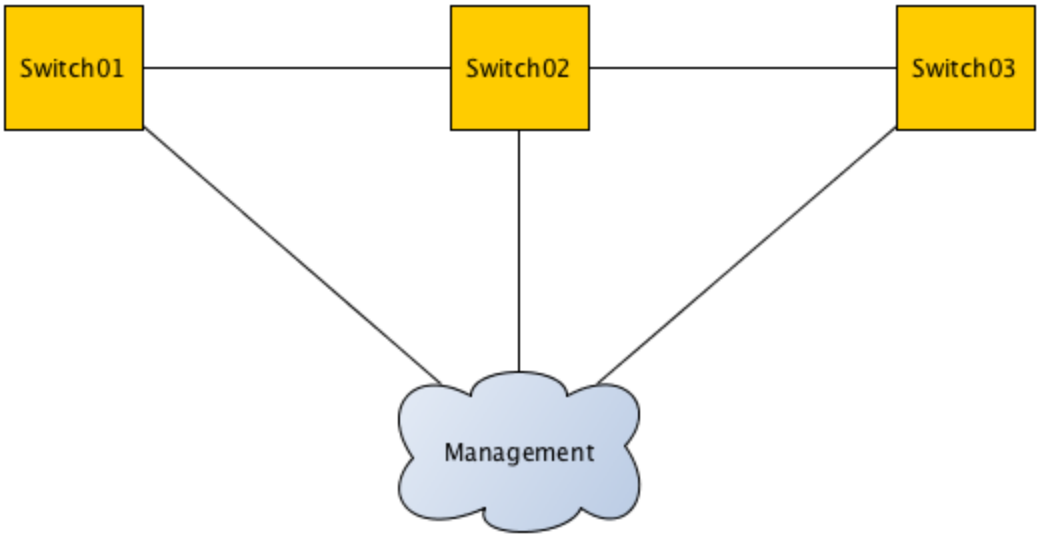
Your switch will be connected to 1 or 2 switches, and at the end of this stage, there should be full reachability between all the switches that conform your group.

To test reachability, you need to be able to execute basic commands like ping and traceroute,

2. Groups

Your group will be integrated by at least 3 or 4 engineers. There should be 6 groups/teams total.

Each group will have 3 switches/routers that will be connected as follows:



Each group of switches will have their own management network, so one ZTP server might be used to send the basic configuration of all the devices, and an additional Pi can be used to push any additional configuration.

How you divide the work is up to your team to decide.

3. Requirements

To access your PI, you should:

- Connect to the on-board LAN port of your Pi and ssh/telnet directly to it.
- Identify which IP address was assigned to the WLAN interface.
- Disconnect the cabled connection and try to access your Pi though the WLAN interface.
- The on-board LAN interface on your Pi will be used to establish connection with the vSwitches and be used by the ZTP and Route servers.

In order to be able to perform the base tasks of this project, the following has to be installed in your Raspberry Pi:

- DHCP server
- TFTP server
- pyeAPI: Python Client for Arista EOS API.

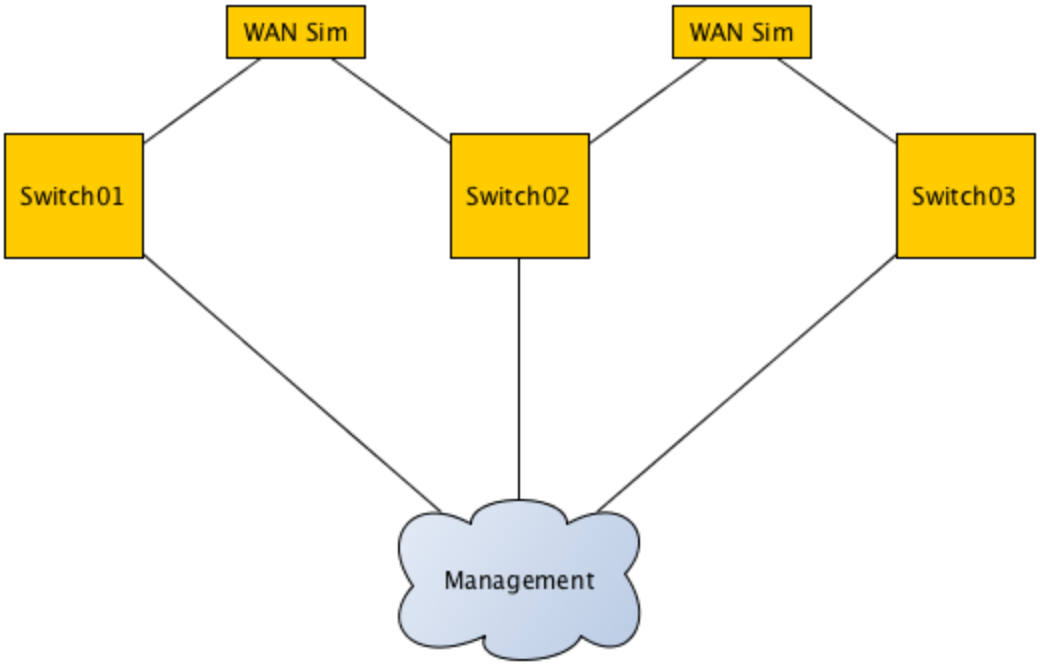
4. References

- ZTP basic setup guide: https://eos.arista.com/ztp-set-up-guide/
- Arista eAPI Python Library: https://github.com/arista-eosplus/pyeapi
- Python Client for eAPI documentation: http://pyeapi.readthedocs.io/en/latest/quickstart.html
- Robot Framework for Auto Test: https://eos.arista.com/robot-framework-for-auto-test/

5. Interaction with WAN Simulator project

Once your network is up and running, you will have to interact with one group on the WAN simulator project in order to add 1 or 2 WAN simulator in the links between your switches.

The WAN simulator will modify parameters in the link in order to simulate particular scenarios: satellite like delay, packet drops, etc. It is up to your team (including the WAN Simulator team) to decide what kind of testing do.



6. Challenges

Depending on your level of expertise on python, automation, linux and other topics, you might want to try the following challenges:

- Instead of using a DHCP server, use scapy to build your own server to intercept DHCP request and send your own responses. NOTE: scapy should be already installed in your Pi, but if it is not, you can find information about how to install it here.
- Use any Web Framework for Python (i.e Flask, Django, web2py, etc.) to provide a web-based interface for the Route Server.
- Instead accessing the switches directly and use CLI, try to use the Arista library for Robot Framework to build your own test suite.
- Build your own WAN simulator.