Agora Expert

ArubaOS-CX

Rafa Requena

Abril, 2019
Contenido

- Portfolio Actual
- Entorno VM ArubaOS-CX y GNS3
  - Laboratorio Virtual
  - Laboratorios
    - Escenario Base
    - Escenario de uso NAE
- NetEdit
  - Laboratorio NetEdit
- Aruba WorkBench
- Extra Bonus
  - Aruba CX Mobile App
  - Laboratorios
    - Escenario OSPF
    - Escenario VSX

Portfolio actual
Aruba Core & Datacenter Switching: Powered by CX Innovations

Mobile First Campus Networking  
Core and Aggregation

Aruba Core & Datacenter Switching: Future proof wired infrastructure, WLAN and IoT enabling

- Highly scalable, programmable automated Data Center solution
- User, device, server aware – ZTP ease of deployment

Aruba 8320

Future proof wired infrastructure, WLAN and IoT enabling

Aruba 8325

Aruba 8400

Aruba 8400 Series with ArubaOS-CX

Modernizing Campus Core, Aggregation and Data Center

Aruba 8400
- Highest reliability, flexibility, performance, port density
- 19.2 Tbps system, 8-slot chassis
- Redundancy everywhere: Mgmt. Module, Fabric, Power, Fans
- Up to 512 10GbE, 128 40GbE, 96 100GbE in a 2-chassis pair

Aruba 8320
- Workhorse for mid-size core/aggregation use cases
- 2.5 Tbps system, 1RU
- N+1 redundant hot swappable power supplies, fans
- Three models: 48 x 10GbE, 48 x 10GBASE-T, 32 x 40GbE

Aruba 8325
- Mid-size core/aggregation use cases and DC ToR or EoR
- 6.4 Tbps system, 1RU
- N+1 redundant hot swappable power supplies, fans
- 32 ports of 40/100 GbE or
- 48 ports of 10/25 GbE and 8 ports of 40/100 GbE
- Front to back or back to front airflow
### Features
ArubaOS-CX 10.2 release improvements

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ACL (Group, CP)</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>PBR (In/L2 ports)</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>NTP (To Any VRF, Master)</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Mirror (OSPF/GRE, Mirror CP)</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Security (Bytag TL4)</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>VXLAN</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>WOL</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>More (DHCP traits ..)</td>
<td></td>
</tr>
</tbody>
</table>

### High Level Selection Considerations

- **Traditional DC Requirements**
  - Software Feature Depth
  - Advanced Integrations

- **Aruba**

- **FlexFabric**

- **Consistency with Campus Analytics, Automation, and Simplicity CX Innovations**

- **Aruba AgoraTech19**
What’s new from last year?
Massive Success and Improvements since 10.0

<table>
<thead>
<tr>
<th>Successes</th>
<th>Improvements</th>
<th>Top customer asks</th>
</tr>
</thead>
<tbody>
<tr>
<td>• 500+ 8000 Series Customers</td>
<td>• Last Year – 10.0</td>
<td>• VXLAN and BGP-EVPN</td>
</tr>
<tr>
<td>• 8325 Launch</td>
<td>• Today – 10.2 and 10.3 Around the Corner</td>
<td>• ERPS</td>
</tr>
<tr>
<td>• Feedback continues to be extremely positive</td>
<td>• Massive Scale Improvements (1M Routes, more!)</td>
<td>• Split / Hydra Cable Support</td>
</tr>
<tr>
<td>• Campus customers deploying Aruba 8000 Series in Data Center</td>
<td>• Feature advancements everywhere in all areas</td>
<td>• MPLS L3 VPN</td>
</tr>
</tbody>
</table>

Management and Orchestration
Core, Aggregation and Data Center

- **IMC**: Advanced wired management
- **AirWave**: Unified multi-vendor wired + wireless network management
- **NAE**: Flexible troubleshooting and automated root cause analytics simplify and enhance visibility and control
- **NetEdit**: Scalable, Simple CLI-based Orchestration
NAE Overview

An end user reports an issue
• IT Ticket is Generated

IT needs to pinpoint the root cause
• Determine Service Impact and Time, Pull logs

Deploy fix

Monitor
• Prevent issue from occurring again

Introduction to Network Automation with ArubaOS-CX

ArubaOS-CX

Agent
Agent
Agent
Agent

NAE

Management Plane

Control Plane

Forwarding Plane

HTTPS
REST API

WebUI

Analytics dashboard

Python script/application

Postman

REST browser

Ansible
Monitoring & Troubleshooting Made Easy

Wide Monitoring Capabilities
Configuration • Protocol and System State
ASIC Counters • ACL’s

Real Time Network Visibility
Synchronization every 5 seconds
Realistic model of network behavior

Intelligence and Automation
Full power of Python
Parameters for customization
Variables for persistent policy state

Sandbox Isolation
Low system overhead
Application Robustness
Security

Flexible Actions
Alert Level
CLI command execution and output capture
Configuration checkpoint diff capture
Syslog generation
Script function callback
External REST calls

“It’s like having a 24/7 network technician as part of the network”

NAE - Current Capabilities

– General Networking Monitoring and Troubleshooting

– Configuration Monitoring

– Networking Protocol Monitoring
  – OSPF, STP, VRRPv2, VSX

– Interface Monitors
  – Link Flap, Link State, Queues, L3/L2 Interface, Tx/Rx Stats

– Physical Device Monitors
  – Fan Stats, Temperature, Power Supply, Transceiver, Line Card Health

– NAE Driven Features
  – Baselining
  – Analytics Data Collector
NAE – 10.2 improvements

- 10.2 Feature Improvements
  - UI: Capacities and Capabilities
  - Scripting: Encrypted Credentials
  - Scripting: Periodic Callbacks
- Use Case Driven NAE Script Highlights
  - Configuration Change Notification
  - IPSLA (New)
  - CoPP (New)
  - Protocol Driven Scripts: OSPFv2, VSX Health (New)
  - MAC/ARP/Neighbor Count and Rate (New)

NETEDIT FOR CONFIG AUTOMATION AND VERIFICATION

Anomaly Detection
Discover inconsistencies in the network configuration

Predictive Assistance
Analyze what you do, anticipate your needs, help complete tasks

Auto-Change Verification
Minimizing change windows and reduce errors

Continuous Validation
Continuously monitor for deviations from intended policy or design

Repetitive Task Automation
Single operation to complete tasks across multiple systems
Introducing Aruba NetEdit
Orchestration Using The CLI

- No knowledge of scripting required
- Easily track changes
- Tagging devices based on location (ToR or EoR) or role (spine and leaf)
- Or in the campus, at access, aggregation or core
- Configuration consistency across the network (data center or the campus)

Entorno VM ArubaOS-CX y GNS3
SETTING UP THE ENVIRONMENT VIRTUAL ARUBAOS-CX IN GNS3

– Download and install Oracle VM VirtualBox (Oracle, Licencia: Privativa / GPLv2)
– Download and install GNS3 (Licencia: GPLv3)
– Download the ArubaOS-CX OVA

https://afp.arubanetworks.com/afp/index.php/ArubaOS-CX_OVA
https://support.hpe.com/hpsc/doc/public/display?docId=a00052041en_us
ArubaOS-CX OVA Requirements

- 4 GB of Ram
- 8 GB Disk Space
- 2 CPU 2 sockets
- Supported Platforms
  - ESXi (5.2.8, release 121009 and above )
  - VirtualBox/GNS3 (2.1.4 and above)
  - VMWare ESXi (6.0.0, build 5224934)
  - VMWare Workstation Pro (12.5.8, build 7098237)
  - VMWare Fusion (10.0.1-6754183)

Lab:
- MacOS: VirtualBox 5.2.26, GNS3 2.1.15

ArubaOS-CX OVA

Unsupported Features

- CoPP
- QoS
- Classifier Policy
- Mirroring
- ECMP
- Changing MTU from default 1500
- Link detection (link is always on)
- MAC ACLs
- UDLD
- Firmware upgrade
- ADC (from NAE)
- native tagged VLAN on ISL

Known Bugs

- VSX active gateway is configurable via the CLI but cannot be pinged.
- Interfaces go down and stay down after a reboot.
- VRRP Virtual IP address is configurable via the CLI but cannot be pinged.
Enlazar GNS3 con VirtualBox

– VirtualBox
  – Importar la OVA
  – NICs not attached
  – Microsoft: Añadir adaptador Microsoft KM-TEST Loopback Adapter
  – No arrancarla !!!!

– GNS3
  – New VirtualBox VM -> Select ArubaOS-CX_Virtual
  – Use as a linked VM
  – Network Tap -> Name format: 1/1/(0)
  – Symbol -> Change

Entorno VM ArubaOS-CX y GNS3
Laboratorios
Escenario Base

- ArubaOS-CX 10.02 VM
- Conectada al MacOS a través de tap0

ArubaOS-CX
- vlan 1
- interface 5/1/6
- no shutdown
- ip address 192.168.1.2/24
- ip route 0.0.0.0/0 192.168.1.1
- ip dns server-address 8.8.8.8
- https-server rest access-mode read-write
- https-server vrf default

MacOS
- $ sudo ifconfig tap0 inet 192.168.1.1/24 up
- $ ifconfig tap0
tap0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
  ether 6a:6e:fe:c4:28:24
  inet 192.168.1.1 netmask 0xffffffff broadcast 192.168.1.255
  media: autoselect
  status: active
  open (pid 61750)

Demo Time ......
Entorno VM ArubaOS-CX y GNS3
Laboratorios - NAE

Aruba Network Analytics Engine: Built in Assurance

Intelligent monitoring

Automated data collection and analysis

Enabling rapid drill-down to root cause
Anatomy of a Script

Script components

Parameters → MONITOR → Time Series

Current State

Condition

Rule → ACTION
Header

- Legal Copyright and Header Information
- Apache 2 License
- Community Friendly License

```python
--- coding: utf-8 ---

# Copyright (c) 2017 Hewlett Packard Enterprise Development LP
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
# http://www.apache.org/licenses/LICENSE-2.0
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
```

Manifest

- Metadata
- Details that can be used in the NAE WebUI

```python
Manifest = {
    'Name': 'system_resource_monitor',
    'Description': 'System Resource (CPU/Memory) Monitoring agent',
    'Version': '1.0',
    'Author': 'Aruba Networks'
}
```
Parameters

- Enable easy modifications to script
- Parameters can be modified in the Agent Edit Wizard
- Recommend creating parameters for any user configurable option in the script

```json
ParameterDefinitions:
  'short-term-high-threshold': {
    'Name': 'Average CPU/Memory utilization in percentage',
    'Description': 'In a short period of offence to set Minor alert',
    'Type': 'integer',
    'Default': '90'
  },
  'short-term-normal-threshold': {
    'Name': 'Average CPU/Memory utilization in percentage in a short period of offence to unset Minor alert',
    'Description': 'When average CPU/Memory utilization is below this value, Minor status will be unset.',
    'Type': 'integer',
    'Default': '80'
  },
```

Monitors

- Network Analytic Engine monitors Data in the Time Series Database
- It monitors elements from the REST API, using URI variables
- Scripts use the "Monitor" function to indicate monitoring a URI
- Diagram: monitor URI maps to Request URL from Swagger API browser
  - /rest/v1/system/subsystems/system/base
  - attributes=resource_utilization
  - cpu indicates value from resource_utilization
  - Response (JSON)

```python
url = '/rest/v1/system/subsystems/system/base?attributes=resource_utilization,cpu'
s
self.sm = Monitor('CPU utilization (CPU/Memory utilization in %)')

Request URL

Response Body
{
  "name" : "utilization",
  "type" : "Integer",
  "attributes" : {
    "cpu" : 90
  }
}

Response Code
200
```
Rules, Conditions and Actions

- **Rule:**
  - Execute an action when condition is met.

- **Condition:**
  - Criteria for determining when a trigger needs to be created.

- **Action:**
  - Execute this action when condition is met.

```
self.rl = Rule('Short-Term High CPU')
self.rl.condition(
    'avg over {} minutes {} > {} pause 30 seconds',
    [self.params['short_term_time_period'],
     self.m1,
     self.params['short_term_high_threshold']])
self.rl.action(self.action_minor_cpu)
```

Advanced Callbacks and Actions

- **Example action_cpu method:**
  - Includes:
    - Custom Syslog messages
    - CLI commands
    - To Run at the time of the alert

```
def action_cpu(self, event):
    self.variables['cpu_minor'] = '1'
    cpu_status = int(self.variables['cpu_status'])
    overall_status = int(self.variables['overall_status'])
    if cpu_status == 0 and overall_status <= 1:
        self.variables['cpu_status'] = '1'
        self.action_cpu(
            event['Value'],
            self.params['short_term_high_threshold'],
            self.params['short_term_time_period'])
        self.set_agent_alert_level()
```

```
def action_minor_cpu(self, event):
    self.variables['cpu_minor'] = '1'
    cpu_status = int(self.variables['cpu_status'])
    overall_status = int(self.variables['overall_status'])
    if cpu_status == 0 and overall_status <= 1:
        self.variables['cpu_status'] = '1'
        self.action_cpu(
            event['Value'],
            self.params['short_term_high_threshold'],
            self.params['short_term_time_period'])
        self.set_agent_alert_level()
```

- **To manage ensuring that alerts aren’t issued every callback, advanced scripting can be employed.**
## Actions (Available in first release)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionCLI()</td>
<td>Execute a “CLI” command and store with Alert</td>
</tr>
<tr>
<td>ActionSyslog()</td>
<td>Issue a custom Syslog Event</td>
</tr>
<tr>
<td>ActionBASH()</td>
<td>Execute a command in BASH and store with Alert</td>
</tr>
<tr>
<td>ActionCustomReport()</td>
<td>• Generate a custom HTML Report and store with Alert</td>
</tr>
<tr>
<td></td>
<td>• Displayed in WebUI as “Analysis Report”</td>
</tr>
<tr>
<td>Set_ALERT_Level()</td>
<td>• Set Alert Level to Critical, Major and Minor</td>
</tr>
<tr>
<td>Remove_ALERT_Level()</td>
<td>• To revert to Normal, remove alert</td>
</tr>
<tr>
<td>Action() &gt; a.k.a. Call Back</td>
<td>• Call back to a NAE script to perform further processing</td>
</tr>
<tr>
<td></td>
<td>• REST Calls can be made via call back using Python</td>
</tr>
</tbody>
</table>
Demo Time …..

Enhancing the Impact w/ Community Participation
Easy to Access and Built to Scale

Primary Hub
• Primary Script Portal
• Designed for Network Engineers
• Public Solutions Integrate Directly with NAE UI
• Community Can Create Custom NAE Solutions

Developer Community
• All Aruba NAE Scripts will be Posted to GitHub
• Community can Fork and Enhance Aruba Scripts
• Embrace NAE Development in a Forum Familiar with Developers

Airheads
• Community to Glue Components Together
• NAE, Aruba Solutions Exchange and GitHub
• Dedicated Aruba Development Resources

NAE Ignite Program
NetEdit

State of the Network

- The network now must support a more complex and less predictable environment
- New security concerns are causing new network architectures to emerge
- IT staffing pressure can’t afford more staff or more experienced staff
Top 10 Causes of Outages

1. Faults, errors or discards in network devices
2. Device configuration changes
3. Operational human errors and mismanagement of devices
4. Link failure caused due to fiber cable cuts
5. Power outages
6. Server hardware failure
7. Security attacks such as denial of service (DoS)
8. Failed software and firmware upgrade or patches
9. Incompatibility between firmware and hardware device
10. Unprecedented natural disasters and ad hoc mishaps on the network such as a minor accidents, or even as unrelated as a rodent chewing through a network line, etc.

Network Changes and Remediation

Not easy to quickly rollback changes

Did I miss something?

Device to device verification
Can’t We Use API’s?

NetEdit

Device Configuration
- Editing
- Validating
- Deploying
- Validating

Software Image
- Uploading
- Deploying
- Rebooting
- Validating

Documenting
Auditing
Entorno VM NetEdit y GNS3
Laboratorios

Creating a NetEdit learning environment
Using VirtualBox and GNS3

Learning environment
- VMs:
  - NetEdit
- VM Platform: VirtualBox
- Virtual Network Platform: GNS3
Configuration management

Workflow

- Create PLAN
- EDIT
- Verify differences
- Validate configuration
- DEPLOY PLAN
- Validate changes
- Rollback
- COMMIT PLAN

Watch a video

NetEdit Configuration Management

Installing OVA en VirtualBox
GNS3 Escenario

- 2 x ArubaOS-CX 10.02 VM
  - 192.168.1.3
  - 192.168.1.4
- OVA NetEdit 1.0 en VirtualBOX
  - 192.168.1.10

MacOS
$ sudo ifconfig tap0 inet 192.168.1.1/24 up

Demo Time ......
NetEdit Device Management and configuration

- Device Management
  - NetEdit Plans. (Changes every 5')
  - Importing ArubaOS-CX Switches
  - Viewing Device Details
  - Analyzing Device’s history
  - Assigning Attributes to devices
    - VSX.false AND OSPF.false
- Configuration
  - Creation a configuration plan
  - Using de Multi-editor
  - Deploying a new configuration
    - Results
    - Rolling Back
    - Commiting

Aruba WorkBench
Recursos en el WorkBench (AWB)

The Aruba Workbench (AWB) is an on-demand hands-on technical hardware work bench that SEs can load and configure their own PoC configs from a "shopping cart" of equipment or select a predefined "recipe".

Conexión:

– Connect to SEEL through a standard demo RAP connected to tokyo.arubademo.net (AOS 6.X Environment) using your legacy Aruba credentials, or "Ondemand" credentials
– OR using VIA connect to fiji.arubademo.net using your "Ondemand" credentials

https://ondemand.arubademo.net account. IE (<username>@hpe.com or <username>@<partnerdomain.com and the password you set in Ondemand)
SEEL Demo Architecture

Resources Include
- Lync
- PBX
- ClearPass
- AirWave
- Palo Alto
- Mobile Iron
- Splunk
- MaaS360
- Spectrum Analysis
- Visual RF
- AppRF
- Remote Switching
- McAfee
- MFA

Overview
- The Layer 1 Fabric
- Current Inventory
- Accessing Workbench
  - SEEL Through VIA
  - SEEL Through RAP
- Interface Walkthrough
- Deployment Scenario
- Connect a RAP into a WB Controller
- Work On Your Own
- Future Release Plans
- Q&A
https://workbench.arubademo.net

Conexión

Deployment A la carté
Ejemplo

Demo Time ......
Extra Bonus

Aruba CX Mobile App

The Aruba CX mobile app automates connecting an ArubaOS-CX switch to the network. Use this mobile app to deploy and manage ArubaOS-CX switches from your mobile device. You can connect to the switch through Bluetooth or WiFi.

The Aruba CX Mobile App redefines and simplifies switch installation.

Installing switches into a campus network just got a lot easier with the Aruba CX mobile app. Running on a smart phone, the Aruba CX mobile app automates connecting the switch to the network.

Using the Aruba CX mobile app, you can use your mobile device to both deploy and manage the switch. The Aruba CX mobile app also automates importing the switch into Aruba NetEdit for intelligent configuration management and continuous conformance validation.

Download Aruba CX Mobile App

[Images of Aruba CX mobile app interface]
PowerShell

**PowerArubaSW**: PowerShell Module to use Aruba Switch API for Vlan, VlanPorts, LACP, LLDP... [More info]

**PowerArubaCP**: Powershell Module to use ClearPass API (create NAD, Guest...) [More info]

**PowerArubaCX**: Powershell Module to use ArubaCX API (get interface/vlan/ports info)

---

**Entorno VM ArubaOS-CX y GNS3**

Laboratorios II
Escenario L3 Access

- ArubaOS-CX 10.02 VM
- Conectada al MacOS a través de tap0

**ArubaOS-CX**
- CORE: 1/1/1 loopback 1.0.1.1
- AggLT: loopback 1.0.3.1
- AccLT: loopback 1.0.1.1
- AccRT: loopback 1.0.1.2
  - show interface brief
  - show lisp neighbor
  - show lisp aggregates
  - show lisp interfaces
  - show ip lisp neighbor
  - show ip lisp interfaces brief
  - ping 10.10.10.254

**MacOS**
- Conectada al MacOS a través de tap0 y la interfaz en 10.10.10.254/24

**Windows**
- Configure the LOOPBACK NIC with the 10.10.10.254 address
- Add routes to 10.0.0.0 and 1.0.0.0 with mask 255.255.0.0 and 10.10.10.1 as the gateway.

Entorno VM ArubaOS-CX y GNS3
Laboratorios II - VSX
Escenario VSX

- ArubaOS-CX 10.02 VM
- Conectada al MacOS a través de tap0

**ArubaOS-CX**
- CORE: loopback 1.1.1.1, 10.0.1.1
- Agg01: loopback 1.1.2.1, vlan 10: 10.0.10.2
- Agg02: loopback 1.1.2.2, vlan 10: 10.0.10.3
- ACCESS: vlan 10: 10.0.10.5

**show lldp neighbor**
**ping 10.0.1.1**
**show ip ospf neighbor**

**MacOS**
- $ sudo ifconfig tap0 inet 10.10.254/24 up

**Windows**
- Configure the LOOPBACK NIC with the 10.10.254 address
- Add routes to 10.0.0.0 and 1.0.0.0 with mask 255.255.0.0 and 10.10.10.1 as the gateway.

**Gracias**
rafael.requena@hpe.com