Aruba Network Automation Essentials

Igor Chinchak
Aruba Edge Expert & Certified Instructor
Computer Data Inc
Igor Chinchak, Aruba Expert and Certified Instructor

– Aruba-Certified Edge Expert #104
– Cisco Certified Internetwork Expert Wireless #61135
– Experienced in automation with Ansible and Python

– https://www.credly.com/users/igor-chinchak
Aruba Network Automation Essentials

What to expect

PART 1: Jan 24, 2022 | 9-11AM PST
• Part 1 Introduces network automation, Python, Ansible

PART 2: Jan 31, 2022 | 9-11AM PST
• Part 2 Explains how to use network automation with Aruba products
Overview / Summary

- Network Automation examples
- Different options for automation
- REST API
- Aruba platforms ready for automation
- Python for automation
- Ansible for automation
Network Automation

Examples
### Workflow Types

<table>
<thead>
<tr>
<th>Day 0 provisioning</th>
<th>Day 100 Maintenance</th>
<th>Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Generate configuration and push to device</td>
<td>• Regularly-scheduled configuration backups</td>
<td>• Backing up current configuration and deploying new configuration</td>
</tr>
<tr>
<td>• Multiple vendors/OS at a time</td>
<td>• audits</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• validations</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Updating Firmware</td>
<td></td>
</tr>
</tbody>
</table>
**API Use Case Example**

**Turn off APs in closed branch offices**

1. **Branch office user Jim arms the security system for the weekend**
2. **The security system sends a notification to the network, indicating branch office is armed**
3. **Network orchestration, queries switches for AP ports**
4. **Network orchestration turns off power to AP ports**
NAE Use Case Example

LAG Health Monitoring

Monitor LAG state

- Forwarding_state:
  - forwarding - Summarizes the state of all the contributors that can block the Port.
  - blocking_layer - Name of the layer that is blocking the forwarding_state.

Conditions:

- Conditions are defined to verify the transition of forwarding state of configured LAG from "true" to "false" and blocking layer from any state to "AGGREGATION"

Actions:

- Critical alert – When the monitoring condition is met, agent status is changed to Critical.
- Normal alert - When blocking layer and forwarding state is transitioned back to "NONE" and "true" respectively, then the agent status is set back to ‘Normal’
Different options for automation
Different options for automation with REST API

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Swagger UI</td>
<td></td>
</tr>
<tr>
<td>cURL</td>
<td></td>
</tr>
<tr>
<td>Postman</td>
<td></td>
</tr>
<tr>
<td>Python</td>
<td></td>
</tr>
<tr>
<td>Ansible</td>
<td></td>
</tr>
</tbody>
</table>
**Example Value** provides an example JSON body for this HTTP method and URI

**Model** describes the structure of the JSON body as well as each field.
API Tools - cURL

- Lightweight free command-line API client
- Supports HTTP and other protocols
- Used to make HTTP requests and test APIs
- Basic syntax: curl [options] <URI>
- Available on Linux, macOS, and Windows
API Tools - Postman

- Graphical API client
- Web app and desktop app
- Create and save requests
- Group saved requests in collections
- Execute entire collection of requests or individual requests
REST API
Overview
An Application Programming Interface (API) defines the ways users and other applications can interact with an application.

An API can use common web-based interactions or communication protocols.

Often used API types:
- RPC
- SOAP
- REST (Most popular)
REST - Overview
REpresentational State Transfer

REpresentational
• The client should be able to request that Data/Resource in different formats (or representations)

State Transfer
• Client should store the state its own application/session

Format
XML, JSON, HTTP

Request

 Currently viewing ABC

Client State
REST Basic Concepts

Four Key Components of a HTTP

Resource (Endpoint)
- URL (Universal Resource Locator)

Headers
- Meta data about what’s being sent and received (Request and Respond)

Method
- What you want to do with the data (e.g., GET, PUT, POST, etc.)

Body
- Data/Payload

---

GET https://10.2.3.4/rest/v10.04/vlans

Status: 200
Data: {"id": 10, "name": "vlan10"}
REST API Concepts (HTTP Request)

### Methods

1. **GET**
   - Retrieves a Resource
2. **POST**
   - Create a new Resource
3. **DELETE**
   - Removes a Resource
4. **PUT**
   - Updates an existing Resource

### Request Headers (some examples)
- **host**: the IP of client (or from where request originated)
- **accept-content**: content understandable by the client
- **user-agent**: data about client, operating system and vendor

### Request Body/Payload
- Also called body or message
- Contains info you want to send to the server
Aruba platforms
Ready for automation
Aruba Central – API Overview

- Central Subscriber ID
- App 1
- App N
- App N
- Token 1
- Token N
- Client ID
- Client Secret
- Token ID
- User ID
- Access Token
- Refresh Token

API entry

Example JSON:

```
{
  "access_token": "wcW55S16qy1gIsUHwXrmYClzB1eGrw",
  "app_name": "nms",
  "authenticated_userid": "kevin.zhu@hpe.com",
  "created_at": "1615114555900",
  "credential_id": "0c3d4f61-fcb0-4af8-bf4f-e749216b74f7",
  "expires_in": 7200,
  "id": "d26faaba-a3f6-41de-8df4-994b6a2343f",
  "refresh_token": "X45UN9FaZ8EUSd9K7epMahtc",
  "scope": "all",
  "token_type": "bearer"
}
```
Aruba AOSCX API Overview

Overview:
- Analytics:
  - Critical: 0
  - Major: 0
  - Minor: 0
  Out of a total of 1 agents
  Scripts: 1/50, Agents: 1/100, Monitors: 7/500

- Config:
  Most Recent Checkpoint:
  05/12/2021 12:09:37
  "CPC20210512190937"
  Total of 14 checkpoints

- Management Modules:
  Module: 1/1
  Login:

Parameters:
- username: admin
- password: ****

Execute
Responses
AOS8 API Documentation

Device itself
- https://<mcr-ip>:4343/api

Aruba Developer Hub - API Reference section
ClearPass API Documentation

The API clients you have defined are listed below.

<table>
<thead>
<tr>
<th>API</th>
<th>Services</th>
<th>Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentlessOnGuard</td>
<td>Settings, SubnetMapping</td>
<td>v1</td>
</tr>
<tr>
<td>ApiFramework</td>
<td>ApiAuthentication, ApiClient</td>
<td>v1</td>
</tr>
<tr>
<td>Authentication</td>
<td>AuthMethod, AuthSource</td>
<td>v1</td>
</tr>
<tr>
<td>Certificates</td>
<td>CertSignRequest, CertTrustList, CertTrustListDetails, ClientCert, RevocationList, SelfSignedCert, ServerCert, ServiceCert</td>
<td>v1</td>
</tr>
<tr>
<td>ClearPassPortal</td>
<td>ClearPassPortal</td>
<td>v1</td>
</tr>
<tr>
<td>Configuration</td>
<td>NetworkScan, Services</td>
<td>v1</td>
</tr>
</tbody>
</table>
DEMO 1

Swagger interface on Aruba platforms
Python
For network automation
Why Python

- Open-source
- Broad standard library
- Interactive Mode
- Cross-platform
- Object-Oriented

Python is an interpreted high-level programming language for general-purpose programming.
Pyaoscx modules that can be called upon to access the REST API and configure various features

Simplifies API usage

- Support both v1 and v10.0.4 APIs
- All functions in all modules fully documented

Open-Source and reliable through direct support from Aruba
AOS8.x API Reference Example

Create new nodes

```python
import requests
url = "https://mobility-master-ip:4343/v1/configuration/object/configuration_node"
querystring = {"config_path": "/md", "UIDARUBA": "XXXXXXXXXXXXX"}
payload = {"node-path": "/md/newnode"},
headers = {"Content-type": "application/json"}
response = requests.request("POST", url, json=payload, headers=headers, params=querystring)
print(response.text)
```
DEMO 2
Aruba Central automation
Ansible
For network automation
## Ansible Overview

An open-source IT automation tool maintained by RedHat

- YAML and Python based
- Free community version
- Enterprise solutions (Red Hat Ansible Tower or community AWX)

### Modular and extensible

- Scales to 100s and 1000s of devices

### Agentless

- nothing to install on the targets

### Secure

- Ansible uses standard transport protocols, including SSH, HTTPs to interact with devices.

### Vendor Support

- automate environments with numerous of systems including different device types such as firewalls, databases, networks
Ansible Concepts

Inventory contains all devices Ansible operates on

- Each device is called a host and has a name and IP address
- Inventory organizes devices into groups which tasks are applied on

Playbook consists of one or more plays in a list

Play is a collection of one or more tasks

Task is a single action to execute through Ansible

Module is a reusable, standalone script that Ansible runs on your behalf
Ansible Concept - Example

```
linux-bash$ ansible-playbook FILENAME -i INVENTORY_FILE
```
Ansible Tower and AWX

AWX is open-source (free) and not supported by RedHat

- Stability is supported through minimal testing
- Security is supported through minimal testing

Docker-based

Version upgrading is not supported

- May have to rebuild/reconfigure each time

Not recommended for production but great for lab use
DEMO 3
Aruba OS-CX automation
<table>
<thead>
<tr>
<th>Resources</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aruba Developer hub</td>
<td>developer.arubanetworks.com</td>
</tr>
<tr>
<td>Ansible &amp; Aruba</td>
<td><a href="http://www.ansible.com/integrations/networks/aruba">www.ansible.com/integrations/networks/aruba</a></td>
</tr>
<tr>
<td>Airheads Developer Community</td>
<td>community.arubanetworks.com/discussion</td>
</tr>
<tr>
<td>Aruba Solution Exchange</td>
<td>ase.arubanetworks.com</td>
</tr>
<tr>
<td>Github</td>
<td>github.com/aruba</td>
</tr>
</tbody>
</table>