

Quick look at YAML and JSON

YAML

- Human readable data serialization language
- Heavily used for configuration files
- Relies heavily on indentation
- 2 space indent is common
- Superset of JSON

YAML Basics

YAML documents start with 3 hyphens (`---`)

Basic Key-Value Pairs

YAML

```
---
hostname: switch1
snmp_ro: public
snmp_rw: private
snmp_location: "nyc"

# integer
vlan_id: 100

# string
vlan_id: "101"
```

JSON

```
{
  hostname: switch1,
  snmp_ro: public,
  snmp_rw: private,
  snmp_location: "nyc",
  vlan_id: 100,
  vlan_id: "101"
}
```

Note: You can comment YAML but not JSON

YAML Basics

List of Strings / Numbers

YAML

```
---
snmp_ro_communities:
  - public
  - public123

vlangs:
  - 100
  - 101
  - 102
  - 103
  - 104
```

JSON

```
{
  "snmp_ro_communities": [
    "public",
    "public123"
  ],
  "vlangs": [
    100,
    101,
    102,
    103,
    104
  ]
}
```

YAML Basics

List of dictionaries

YAML

```
---
- vlan_name: web
  vlan_id: '10'
  vlan_state: active
- vlan_name: app
  vlan_id: '20'
  vlan_state: active
- vlan_name: DB
  vlan_id: '30'
  vlan_state: active
```

JSON

```
[
  {
    "vlan_name": "web",
    "vlan_id": "10",
    "vlan_state": "active"
  },
  {
    "vlan_name": "app",
    "vlan_id": "20",
    "vlan_state": "active"
  },
  {
    "vlan_name": "DB",
    "vlan_id": "30",
    "vlan_state": "active"
  }
]
```

YAML Advanced Data Types

Dictionaries

YAML

```
---
snmp:
  ro: public
  rw: private
  info:
    location: nyc
    contact: bob
vlangs:
  10:
    name: web
  20:
    name: app
```

JSON

```
{
  "snmp": {
    "ro": "public",
    "rw": "private",
    "info": {
      "location": "nyc",
      "contact": "bob"
    }
  },
  "vlangs": {
    "10": {
      "name": "web"
    },
    "20": {
      "name": "app"
    }
  }
}
```

YAML Advanced Data Types

Dictionaries that are lists of dictionaries

YAML

```
---
vlangs:
  - id: 10
    name: web
  - id: 20
    name: app
snmp_community_strings:
  - type: ro
    community: public
  - type: ro
    community: networktocode
  - type: rw
    community: private
```

JSON

```
{
  "vlangs": [
    {
      "id": 10,
      "name": "web"
    },
    {
      "id": 20,
      "name": "app"
    }
  ],
  "snmp_community_strings": [
    {
      "type": "ro",
      "community": "public"
    },
    {
      "type": "ro",
      "community": "networktocode"
    },
    {
      "type": "rw",
      "community": "private"
    }
  ]
}
```

YAML Advanced Data Types

YAML is a superset of JSON

YAML

```
---
ned:Loopback:
  #YAML supports comments
  name: 200
  ip:
    address:
      primary:
        address: 100.200.2.2
        mask: 255.255.255.0
      secondary:
        - address: 100.200.20.20
        - address: 100.200.200.200
```

JSON

```
{
  "ned:Loopback": {
    "name": 200,
    "ip": {
      "address": {
        "primary": {
          "address": "100.200.2.2",
          "mask": "255.255.255.0"
        },
        "secondary": [
          {
            "address": "100.200.20.20"
          },
          {
            "address": "100.200.200.200"
          }
        ]
      }
    }
  }
}
```


Data Types - Summary

- For most automation tasks YAML and JSON have 1-1 mapping
- They both tie back to dictionaries
- A lot of initial automation tasks revolve around parsing return data, therefore it is important to understand:
 - Lists of lists
 - Lists of dictionaries
 - Dictionaries with lists
 - Complex nested objects
- Always remember to traverse a complex object from left to right

Demo

- Validate YAML
- <http://yamllint.com/>
- YAML to JSON Conversion
- JSON to YAML Conversion
- <https://www.json2yaml.com>
- Understand how to model network configuration data in YAML (for use in Ansible)
- Compare/Contrast Data Models on different platforms