### **YAML - Primer (for Ansible learning)**

### **Ansible YAML Syntax Guide with Data Types**

#### **Basic YAML Data Types**

**Strings - Text values enclosed in quotes (single or double) or without quotes if they don't contain special characters.**

message: "Welcome to Ansible"

#this is a comment

**Integers - Whole numbers without quotes.**

max\_connections: 150

**Booleans - True or False values without quotes**

debug\_mode: true

**Floating Points - Numbers with decimals without quotes**

salt\_vaule: 5.599999

**Lists - Sequences of items, each item starting with a dash -.**

packages:

- git

- curl

- vim

**Dictionaries - Key-value pairs, with each key followed by a colon and the value**

database:

name: mydb

user: admin

password: “secret”

port: 3606

**Multiline Strings - Literal Block (|): Keeps newlines.**

certificate: |

-----BEGIN CERTIFICATE-----

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQE...

-----END CERTIFICATE-----

**Folded Block (>): Folds newlines into spaces:**

description: >

This is a description

that spans multiple lines

but will be shown as a single line.

**Basic Nested Structures**

**Nested Dictionaries:**

database:

name: mydb

settings:

user: admin

password: secret

host: localhost

**List of Dictionaries:**

employees:

- name: John Doe

role: Developer

skills:

- Python

- Django

- name: Jane Smith

role: Designer

skills:

- Photoshop

- Illustrator

**Dictionaries within Lists:**

servers:

web:

- name: web1

ip: 192.168.1.1

- name: web2

ip: 192.168.1.2

db:

- name: db1

ip: 192.168.2.1

- name: db2

ip: 192.168.2.2

**Complex Example with Nested Structures:**

application:

name: MyApp

version: 1.0

components:

frontend:

language: JavaScript

framework: React

servers:

- name: frontend1

ip: 192.168.1.10

- name: frontend2

ip: 192.168.1.11

backend:

language: Python

framework: Django

databases:

- name: db1

type: PostgreSQL

host: 192.168.2.10

- name: db2

type: MySQL

host: 192.168.2.11

#### 

#### 

#### **Now lets look at an example YAML file for Ansible use case:**

**Playbook with Various Data Types to illustrate Apache server installation:**

- name: Setup Web Server

hosts: webservers

become: yes

vars:

max\_clients: 200 # Integer

site\_name: "My Awesome Site" # String

debug\_mode: true # Boolean

timeout: 5.0 # Float

packages: # List

- apache2

- php

- mysql-server

config: # Dictionary

user: webadmin

group: webgroup

document\_root: /var/www/html

tasks:

- name: Install packages

apt:

name: "{{ packages }}"

state: present

- name: Configure Apache

template:

src: templates/apache.conf.j2

dest: /etc/apache2/sites-available/000-default.conf

notify:

- Restart Apache

- name: Set up document root

file:

path: "{{ config.document\_root }}"

state: directory

owner: "{{ config.user }}"

group: "{{ config.group }}"

handlers:

- name: Restart Apache

service:

name: apache2

state: restarted

### **Troubleshooting Tips**

1. Use VS Code Editor or any other online editors while writing ansible. These editors will help highlight the syntax errors and help speed up preparing the file.
   1. Example editor - <https://onlineyamltools.com/edit-yaml>
   2. Try copy pasting the code from this document and check for syntax errors.
   3. Once written correctly, copy to clipboard or download the file.
2. Check Indentation: YAML relies heavily on indentation. Ensure consistent use of spaces (no tabs).
3. Quotes for Special Characters: Use quotes for strings containing special characters like :, -, or #.
4. Proper Syntax: Verify key-value pairs are correctly formatted. Dictionaries use key: value, and lists start with -.
5. Error Messages: Read error messages carefully; they often indicate the line number and type of error.