YAML - STRUCTURES

Yaml Breakdown

- Mapping
- Sequences
- Scalars
- **Structures**
- Comments
- Tags
- Anchors

Structures

- Combining mappings, Sequences, and Scalars to represent complex data.
- Advanced data structures

What Are Structures?

- “Structures” refers to the organization and arrangement of data within a YAML document.
- YAML provides various structures for representing complex and hierarchical data, allowing you to organize information in a clear, readable, and structured manner.

The Main Structures In Yaml Are:

- Mappings (Dictionaries or Hashes)
- Sequences (Lists or Arrays)
- Scalars
- Anchors and Aliases
- Comments
- Directives

Represent Complex Data

- Combining mappings, sequences, and scalars in YAML allows you to represent complex and structured data effectively.
- You can create hierarchies, arrays of data, and individual values within a single YAML document.

Example:


```yaml
---
server_configuration:
  server_name: web_server
  ip_address: 192.168.1.100
  ports:
    - 80
    - 443
  services:
    - name: nginx
      version: 1.18.0
      enabled: true
    - name: mysql
      version: 8.0.23
      enabled: false
  environment:
    production: true
    backup_enabled: false
```

In this example related to Ansible:

Mapping (server_configuration):

- Represents the configuration for a server.
- Contains scalars such as server_name, ip_address.
- Contains a sequence (ports) representing the ports the server uses.
- Contains a sequence (services) representing the installed services with their respective versions and whether they are enabled or not.
- Contains another mapping (environment) representing the server's environment settings.

What Are Advanced Structures?

- In YAML, you can utilize advanced data structures beyond simple mappings, sequences, and scalars.
- These advanced structures allow for more complex and nested representations of data.

Some of The Advanced Structures Are:

- Nested Mappings and Sequences
- Set (Using Mappings with Null Values)
- Tree-like Structures
- Anchors and Aliases with Complex Data
- Multi-dimensional Arrays using Sequences

These advanced data structures can be utilized based on your specific use case to represent more complex, organized, and hierarchical data within YAML. It's important to choose the appropriate structure to best represent the relationships and organization of your data.

Nested Mappings And Sequences:

- Nesting mappings and sequences within each other to create complex hierarchical structures.

Example:

```
---
users:
  - name: John Doe
    roles:
      - role: admin
      - role: editor
  - name: Jane Smith
    roles:
      - role: viewer
```

Set (Using Mappings With Null Values):

- Simulating a set data structure using mappings with null values.

Example:

```
---
unique_values:
  item1: null
  item2: null
  item3: null
```

Tree-like Structures:

- Creating tree-like structures using mappings and sequences to represent hierarchical relationships.

Example:

```
---
tree:
  - node1:
      - subnode1
      - subnode2
  - node2:
      - subnode3
```

Anchors And Aliases With Complex Data:

- Utilizing anchors and aliases for referencing complex data within the document.

Example:

```
---
- &user1
  name: John Doe
  age: 30
- &user2
  name: Jane Smith
  age: 25
```

Referencing Aliases:

```
---
users:
  - *user1
  - *user2
```

Multi-dimensional Arrays using Sequences:

- Creating multi-dimensional arrays or matrices using sequences within sequences.

Example:

```
---
matrix:
  - [1, 2, 3]
  - [4, 5, 6]
```