YAML FILE USAGE

YAML files can be used in docker compose, Ansible, Kubernetes etc… which are helpful for configuration management and container orchestrations.

1. YAML in docker compose:-

The example application is composed of the following parts:

* 2 services, backed by Docker images: webapp and database.
* 1 secret (HTTPS certificate), injected into the frontend.
* 1 configuration (HTTP), injected into the frontend.
* 1 persistent volume, attached to the backend.
* 2 networks.

services:

frontend:

image: awesome/webapp

ports:

- "443:8043"

networks:

- front-tier

- back-tier

configs:

- httpd-config

secrets:

- server-certificate

backend:

image: awesome/database

volumes:

- db-data:/etc/data

networks:

- back-tier

volumes:

db-data:

driver: flocker

driver\_opts:

size: "10GiB"

configs:

httpd-config:

external: true

secrets:

server-certificate:

external: true

networks:

# The presence of these objects is sufficient to define them

front-tier: {}

back-tier: {}

1. YAML in Ansible: -

Ansible uses YAML syntax for expressing Ansible playbooks. Ansible uses YAML because it is very easy for humans to understand, read and write when compared to other data formats like XML and JSON.

Sample task and handler files in a function-based role:

Ansible loads any file called main.yml in a role sub-directory. This sample tasks/main.yml file is simple - it sets up NTP, but it could do more if we wanted:

---

# file: roles/common/tasks/main.yml

- name: be sure ntp is installed

yum:

name: ntp

state: present

tags: ntp

- name: be sure ntp is configured

template:

src: ntp.conf.j2

dest: /etc/ntp.conf

notify:

- restart ntpd

tags: ntp

- name: be sure ntpd is running and enabled

service:

name: ntpd

state: started

enabled: yes

tags: ntp

1. YAML in Kubernetes: -

How to create a kubernetes Pod using YAML ?

To create a Kubernetes pod with YAML, you first create an empty file, assign it the necessary access permissions, and then define the necessary key-value pairs. The important ones are the apiVersion, the kind (pod), name, and the containers within the pod.

For instance, below is the YAML code to create a pod named mywebapp1 that has 2 containers: One is a web server and the other is a database server. It is also assigned to a specific volume named websvr-storage:

Filename: /k8s/pods/pod1.YAML

apiVersion: v1

kind: Pod

metadata:

name: mywebapp1

labels:

role: webserver-role

app: nginx

spec:

containers:

- name: webserver1

image: nginx:1.6

ports:

- containerPort:80

- name: database-server

image: mysql-3.2

ports:

- containerPort:3306

volumes:

- name: websvr-storage

emptyDir: {}