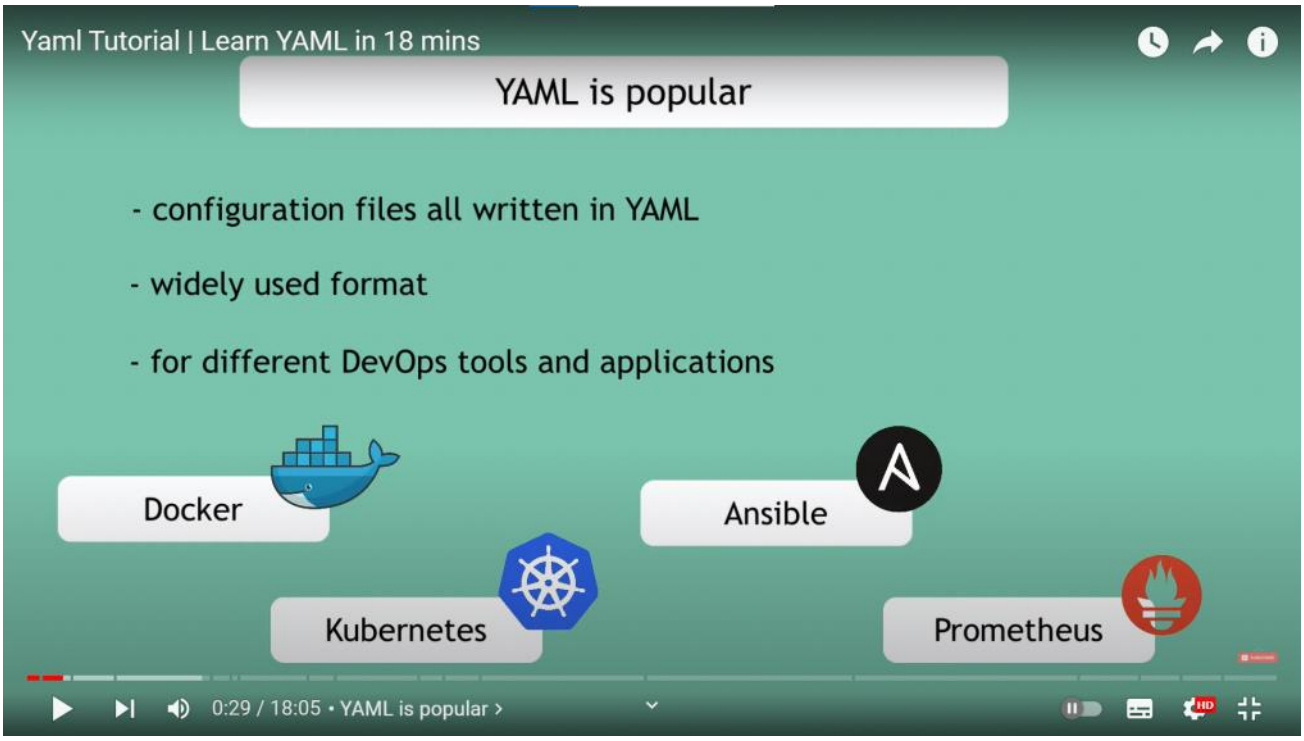
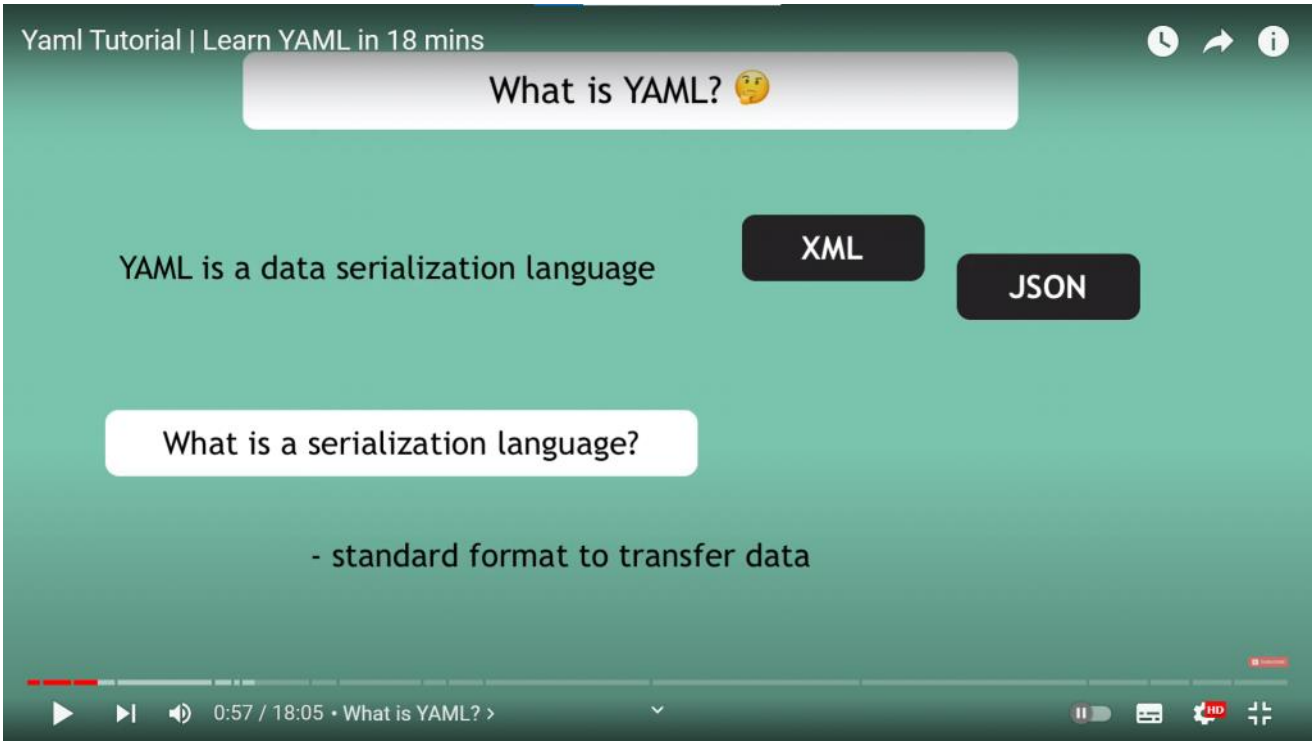
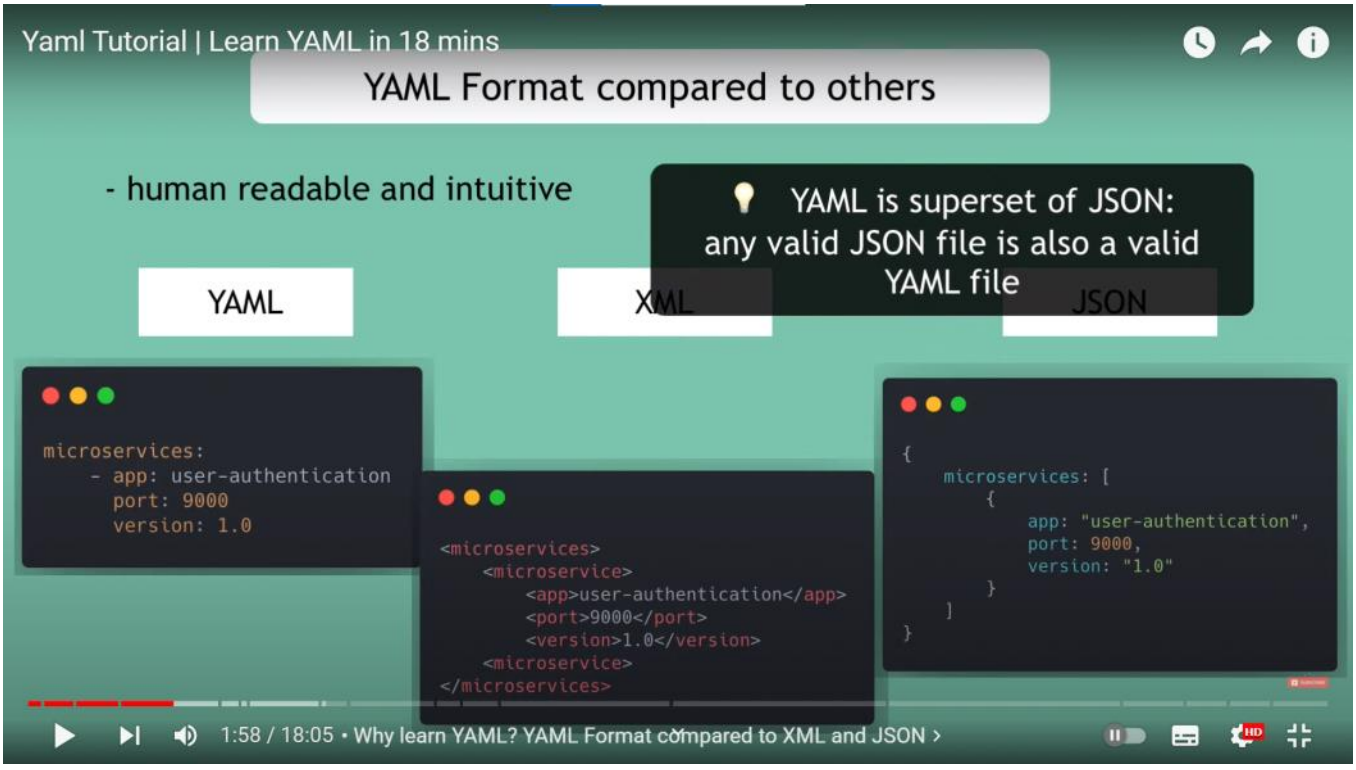
YAML

Thursday, October 12, 2023 4:44 PM





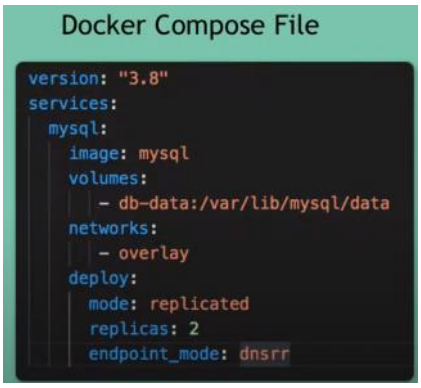

YAML Ain't Markup Language
File extension:
.yaml
.yml


Yaml Tutorial | Learn YAML in 18 mins
YAML Format compared to others
- human readable and intuitive
YAML is superset of JSON: any valid JSON file is also a valid YAML file
YAML
XML
JSON
microservices:
- app: user-authentication
port: 9000
version: 1.0
<microservices>
<microservice>
<app>user-authentication</app>
<port>9000</port>
<version>1.0</version>
<microservice>
</microservices>
{
microservices: [
{
app: "user-authentication",
port: 9000,
version: "1.0"
}
]
}
1:58 / 18:05 • Why learn YAML? YAML Format compared to XML and JSON >


YAML
microservices:
- app: user-authentication
port: 9000
version: 1.0
line separation
indentation


YAML Use Cases

Docker Compose File



```yaml
version: "3.8"
services:
  mysql:
    image: mysql
    volumes:
      - db-data:/var/lib/mysql/data
    networks:
      - overlay
    deploy:
      mode: replicated
      replicas: 2
      endpoint_mode: dnsrr
```

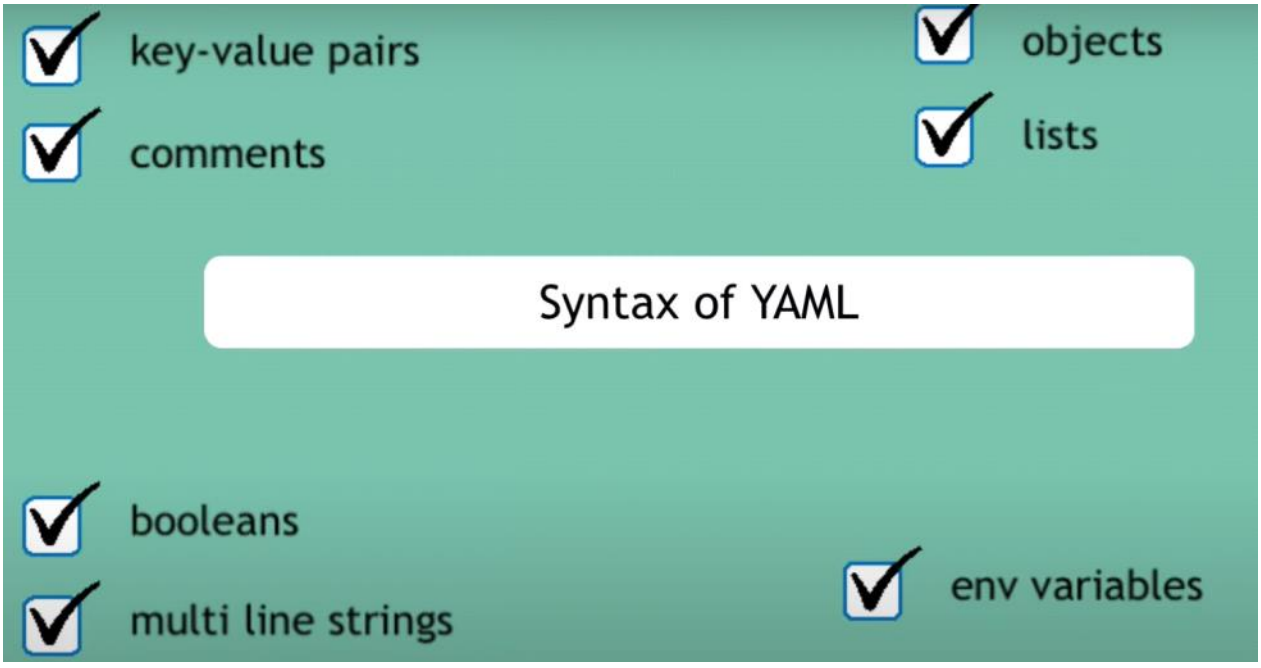
Ansible

```yaml
- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
      state: started
```

Kubernetes

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
```



KEY-VALUE PAIRS:

```
app: user-authentication
port: 9000
version: 1.7
```

COMMENTS:

```
# comment here
app: user-authentication
port: 9000
# comment here
```

OBJECTS:

```
microservice:
  app: user-authentication
  port: 9000
  version: 1.7
```

NOTE: HERE , microservice is object attribute

LISTS:

```
microservices:
  - app: user-authentication
    port: 9000
    version: 1.7
```

- IS used to indicate list of objects .

```
microservices:
  - app: user-authentication
    port: 9000
    version: 1.7
  - app: shopping-cart
    port: 9002
    version: 1.9
```

Shopping-cart /user-authentication are objects of lists

```
microservices:
  - user-authentication
  - shopping-cart
```

List of names

```
versions:
- 1.9
- 2.0
- 2.1
```

Or

```
- app: shopping-cart
  port: 9002
  versions: [1.9, 2.0, 2.1]
```

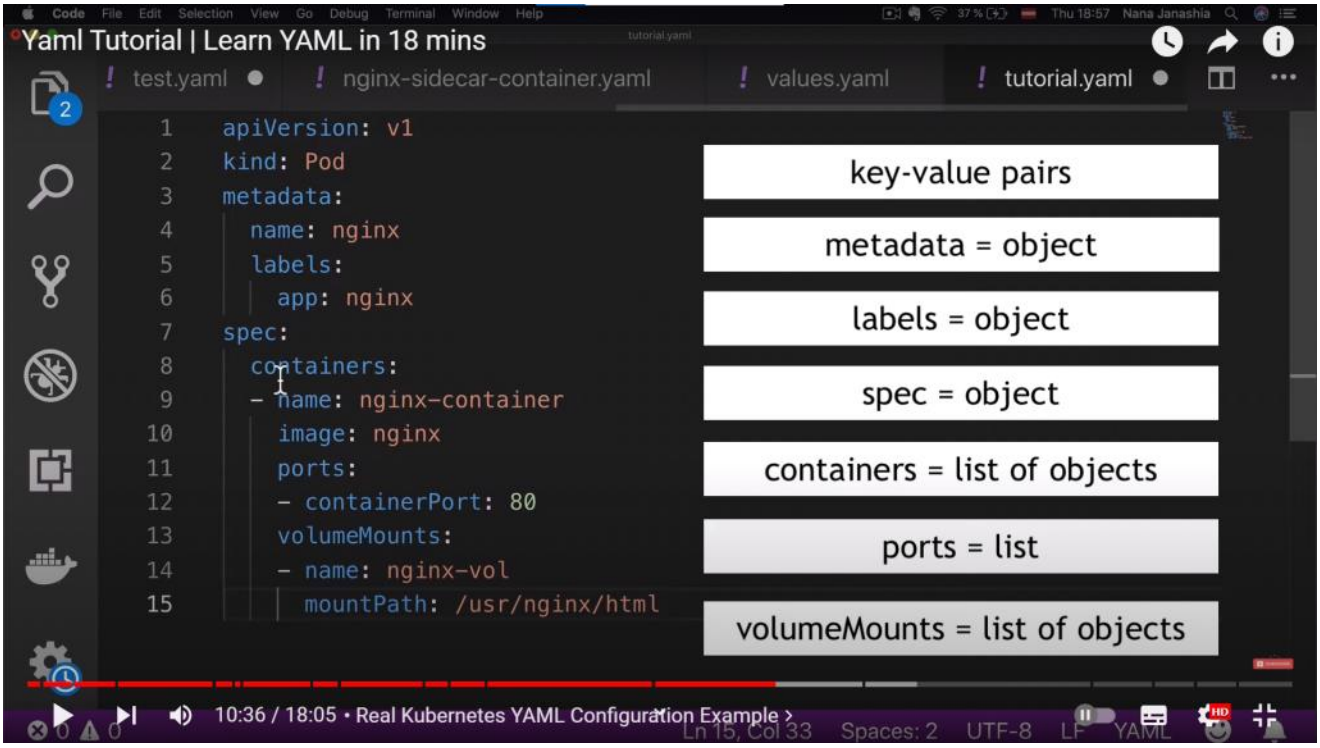
List of versions

BOOLEAN :

```
- app: user-authentication
  port: 9000
  version: 1.7
  deployed: true
```

Deployed: true/false or ON/OFF or YES/NO , here indicates boolean

KUBERNETES POD CONFIGURATION EXAMPLE:



MULTILINE STRING

```
multilineString: |
  this is a multiline string
  and this is the next line
  next line
```

REPRESENTING MULTILINE IN ONE LINE:

```
multilineString: >
  this is a single line string,
  that should be all on one line.
  some other stuff
```

KEEPING SHELL SCRIPT IN YAML CONFIG FILE AND EXECUTING USING MULTILINE:

```
command:
  - sh
  - -c
  - |
    #!/usr/bin/env bash -e
    http () {
        local path="${1}"
        set -- -XGET -s --fail
        # some more stuff here
        curl -k "$@" "http://localhost:5601${path}"
    }
    http "/app/kibana"
```

ENVIRON VARIABLE USING $ SIGN:

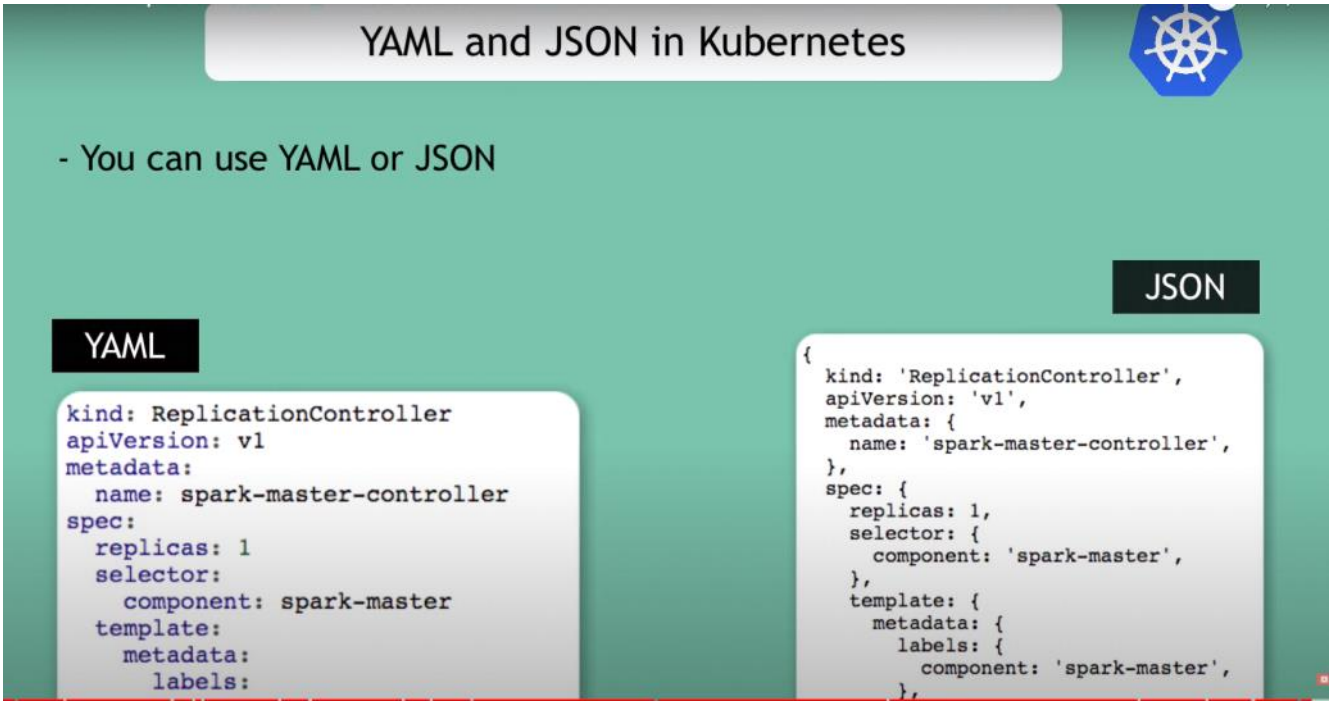
```
# readiness probe
command:
  - /bin/sh
  - -ec
  - >-
    mysql -h 127.0.0.1 -u root -p$MYSQL_ROOT_PASSWORD -e 'SELECT 1'
```

USING PLACEHOLDER IN YAML:

```
metadata:
  name: {{ .Values.service.name }}
spec:
```

MULTIPLE YAML DOCS IN SINGLE YAML FILE USING --- :

```
1   apiVersion: v1
2   kind: ConfigMap
3   metadata:
4     name: mosquitto-config-file
5   data:
6     mosquitto.conf: |
7       log_dest stdout
8       log_type all
9       log_timestamp true
0       listener 9001
1
2   ---
3
```


YAML and JSON in Kubernetes
- You can use YAML or JSON
YAML
kind: ReplicationController
apiVersion: v1
metadata:
name: spark-master-controller
spec:
replicas: 1
selector:
component: spark-master
template:
metadata:
labels:
JSON
{
kind: 'ReplicationController',
apiVersion: 'v1',
metadata: {
name: 'spark-master-controller',
},
spec: {
replicas: 1,
selector: {
component: 'spark-master',
},
template: {
metadata: {
labels: {
component: 'spark-master',
},