Essential properties in a kubernetes YAML file

Every YAML resource definition file follows a standard structure with 4 essential top-level properties.

1. apiVersion

- Tells Kubernetes which API group and version to use when parsing and processing the object.
- Example
  - v1 : for core objects like Pod, Service, ConfigMap
  - apps/v1 : for Deployment, StatefulSet, ReplicaSet
  - batch/v1 : for Job, CronJob

2. kind

- Define what we are creating.
- Example: Pod, Deployment, Service, Configmap

3. metadata

- Contains identifying information
  - name: Unique name of the object
  - namespace: (optional) logical grouping
  - labels: key-value pairs to tag the object
  - annotations: metadata for tools/scripts

4. spec

- The core configuration of the object. Varies depending on the kind.
- For a Pod, it includes containers, images, ports, volumes.
- For a Deployment, it includes replicas, selector, and Pod template.

Example1 : **Kubernetes Pod YAML**

```yaml
apiVersion: v1                  # 1. API version
kind: Pod                       # 2. Resource type
metadata:                       # 3. Object metadata
  name: nginx-pod
  labels:
    app: nginx
spec:                           # 4. Desired configuration
  containers:
    - name: nginx
      image: nginx:latest
      ports:
        - containerPort: 80
```

Example2: **Kubernetes Deployment YAML**

```yaml
apiVersion: apps/v1             # Notice the different API version for Deployment
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - name: nginx
          image: nginx:1.25
          ports:
            - containerPort: 80
```

** The 4 essential properties in a Kubernetes YAML file**

Property	Description
`apiVersion`	Defines the **Kubernetes API version** to use for this resource (e.g. `v1`, `apps/v1`, `batch/v1`)
`kind`	Specifies the **type of Kubernetes object** being created (e.g. `Pod`, `Deployment`, `Service`)
`metadata`	Contains **data about the object**, such as `name`, `namespace`, `labels`, and `annotations`
`spec`	Describes the **desired state/configuration** of the object (e.g. container images, replicas, ports)