Kubernetes uses YAML files as inputs for the creation of objects such as PODs, replicas, deployments, services etc. All of these follow similar structure, a Kubernetes definition file always contains 4 top level fields:

1. The API version
2. Kind
3. Metadata
4. Spec

These are the top level or root level properties and these are also required fields so you must have them in your configuration file. Let’s look at each of them.

1. **The API Version:** This is the version of the Kubernetes API you’re using to create the objects. Depending on what we are trying to create we must use the right API version. For now, since we are working on POD, we will set the API version as we want.
2. **Kind:** Kind refers to the type of object we are trying to create which in this case happens to be a POD. Other possible values could be replica set, deployment or service.
3. **Metadata:** The Metadata is the data about the object like its name, labels etc. Unlike the first two where you have specified a string value, this is in the form of a dictionary. So, everything under Metadata is intended to the right a little bit. Metadata has following properties:

**name:** my-pod

labels:

**app:** name

**type:** front-end

With the help of labels, I can label my apps as front-end, back-end to easily filter my apps running on production environment. Metadata cannot contain any custom property; it can only consist of only those properties that are expected by the Kubernetes. But under label, we can have any key-value pair property. So, it’s important to understand what each of these parameters expect.

1. **Spec:** Depending on the object we are going to create this is where we are going to provide additional information to Kubernetes pertaining to that object. This is going to be different for different object. Spec is a dictionary so add a property under it called containers. Containers is a list or an array. The reason this property is a list because the pods can have multiple containers within them.

