```
    name: custom-api-server
version: v1alpha1
```

Besides the Service and Pod implementation, we need some additional security configuration for setting up the ServiceAccount under which the Pod is running.

Once it is set up, every request to the API Server `https://<api server ip>/apis/sample-api.k8spatterns.io/v1alpha1/namespaces/<ns>/...` is directed to our custom Service implementation. It's up to this custom Service's implementation to handle these requests, including persisting the resources managed via this API. This approach is different from the preceding CRD case, where Kubernetes itself completely manages the custom resources.

With a custom API Server, you have many more degrees of freedom, which allows going beyond watching resource lifecycle events. On the other hand, you also have to implement much more logic, so for typical use cases, an operator dealing with plain CRDs is often good enough.

A detailed exploration of the API Server capabilities is beyond the scope of this chapter. The official documentation as well as a complete sample-apiserver have more detailed information. Also, you can use the apiserver-builder library, which helps with implementing API Server aggregation.

Now, let's see how we can develop and deploy our operators with CRDs.

Operator Development and Deployment

At the time of this writing (2019), operator development is an area of Kubernetes that is actively evolving, with several toolkits and frameworks available for writing operators. The three main projects aiding in the creation of operators are as follows:

- CoreOS Operator Framework
- Kubebuilder developed under the SIG API Machinery of Kubernetes itself
- Metacontroller from Google Cloud Platform

We touch on them very briefly next, but be aware that all of these projects are quite young and might change over time or even get merged.

Operator framework

The Operator Framework provides extensive support for developing Golang-based operators. It provides several subcomponents:

- The *Operator SDK* provides a high-level API for accessing a Kubernetes cluster and a scaffolding to start up an operator project