



## CUSTOM RESOURCE DEFINITIONS

## Custom Resource Definitions

### Custom Resource Definitions

As we have already learnt, the decoupled nature of Kubernetes depends on a collection of watcher loops, or controllers, interrogating the kube-apiserver to determine if a particular configuration is true. If the current state does not match the declared state, the controller makes API calls to modify the state until they do match. If you add a new API object and controller, you can use the existing kube-apiserver to monitor and control the object. The addition of a Custom Resource Definition will be added to the cluster API path, currently under **apiextensions.k8s.io/v1**.

While this is the easiest way to add a new object to the cluster, it may not be flexible enough for your needs. Only the existing API functionality can be used. Objects must respond to REST requests and have their configuration state validated and stored in the same manner as built-in objects. They would also need to exist with the protection rules of built-in objects.

A CRD allows the resource to be deployed in a namespace or be available in the entire cluster. The YAML file sets this with the **scope:** parameter, which can be set to **Namespaced** or **Cluster**.

Prior to v1.8, there was a resource type called ThirdPartyResource (TPR). This has been deprecated and is no longer available. All resources will need to be rebuilt as CRD. After upgrading, existing TPRs will need to be removed and replaced by CRDs such that the API URL points to functional objects.