**Kubernetes CRDs, Helm, and Operators: A Simple Guide**

**What are CRDs?**

In Kubernetes, **Custom Resource Definitions (CRDs)** allow you to extend the Kubernetes API with custom resource types. This means that Kubernetes can understand and manage resources that are not part of the standard Kubernetes API (e.g., Pod, Service, Deployment). CRDs are used to add new resource types that Kubernetes doesn't know about out-of-the-box.

For example, when you deploy tools like **Prometheus**, **Velero**, or **ArgoCD**, you are actually using CRDs to define custom resources that these tools need to function properly.

**What is an Operator?**

An **Operator** is a Kubernetes controller that manages the lifecycle of applications or resources. Operators watch and manage custom resources defined by CRDs. They ensure that the desired state of these resources (like Prometheus or Velero) is always maintained.

For instance, the **Prometheus Operator** manages the Prometheus instance, scaling, and configuration based on the custom resources you define.

**How CRDs are Deployed in Kubernetes (via Helm)**

**1. Helm Charts and CRDs**

Helm is a package manager for Kubernetes, and it simplifies the process of deploying applications. When you install an application using a **Helm chart**, it may include CRDs as part of the package.

When you install Helm charts for tools like **Velero**, **ArgoCD**, or **Prometheus**, the CRDs required for those tools are automatically installed into your Kubernetes cluster. This means you can immediately start using the custom resources defined by those CRDs.

Here’s how it works:

* **Helm** installs both the application components (like Pods, Deployments) and **CRDs**.
* The CRDs are then available in your cluster, allowing you to use the corresponding **custom resources** like Prometheus, VeleroBackup, ArgoCDApplication, etc.
* These custom resources are managed by the **Operator** associated with the application.

**Tools Example: Velero, ArgoCD, Prometheus**

**1. Prometheus**

* **What is Prometheus?**
  + Prometheus is an open-source monitoring and alerting toolkit designed for Kubernetes.
* **How Does Prometheus Use CRDs?**
  + **Prometheus CRDs** allow you to define and manage Prometheus servers in your cluster. The CRDs include Prometheus, Alertmanager, ServiceMonitor, and PodMonitor.
  + **Prometheus Operator** manages these CRDs to automatically deploy, configure, and scale Prometheus instances based on the custom resources you create.
* **How Do You Deploy Prometheus?**
  + When you install the **Prometheus Operator** using a Helm chart, it automatically installs the CRDs for Prometheus (Prometheus, Alertmanager, etc.).
  + You can then create custom resources (like Prometheus) to deploy Prometheus instances.

Example:

yaml

Copy code

apiVersion: monitoring.coreos.com/v1

kind: Prometheus

metadata:

name: example-prometheus

spec:

replicas: 2

**2. Velero**

* **What is Velero?**
  + Velero is a tool to back up and restore Kubernetes resources and persistent volumes.
* **How Does Velero Use CRDs?**
  + **Velero CRDs** allow you to define backup and restore operations in a Kubernetes-native way. It includes resources like Backup, BackupStorageLocation, Restore, etc.
* **How Do You Deploy Velero?**
  + When you install Velero using a Helm chart, the CRDs required by Velero (like Backup, Restore, etc.) are automatically installed in your cluster.
  + You can then create resources like Backup to back up your Kubernetes workloads.

Example:

yaml

Copy code

apiVersion: velero.io/v1

kind: Backup

metadata:

name: example-backup

spec:

includedNamespaces:

- default

storageLocation: default

**3. ArgoCD**

* **What is ArgoCD?**
  + ArgoCD is a GitOps tool for continuous delivery, allowing you to deploy and manage applications in Kubernetes using Git repositories.
* **How Does ArgoCD Use CRDs?**
  + **ArgoCD CRDs** allow you to define applications, sync policies, and deployment configurations. It includes custom resources like Application, AppProject, and SyncPolicy.
* **How Do You Deploy ArgoCD?**
  + When you install ArgoCD using a Helm chart, the CRDs for ArgoCD (Application, AppProject) are installed into the cluster.
  + You can then create Application resources to define which Git repositories ArgoCD should sync and deploy.

Example:

yaml

Copy code

apiVersion: argoproj.io/v1alpha1

kind: Application

metadata:

name: example-application

spec:

project: default

source:

repoURL: https://github.com/your-repo.git

path: path/to/k8s/manifests

destination:

server: https://kubernetes.default.svc

namespace: default

**Summary: Key Concepts**

1. **CRDs** (Custom Resource Definitions) define new resource types in Kubernetes that are not natively part of the Kubernetes API.
2. **Helm** is used to deploy tools (like Prometheus, Velero, ArgoCD) along with their CRDs.
3. After installing the Helm chart, you can create and manage custom resources (like Prometheus, Backup, Application) defined by the CRDs.
4. **Operators** (like Prometheus Operator, Velero Operator) watch and manage these custom resources, ensuring the desired state is maintained.

**Why CRDs Are Important:**

* CRDs allow tools to define their own Kubernetes-native resources and make management easy and declarative.
* By installing CRDs, you enable Kubernetes to understand and manage resources for applications that extend Kubernetes' core functionality (like monitoring, backups, continuous delivery).

This document explains the basics of CRDs, how they are deployed with Helm, and how tools like **Prometheus**, **Velero**, and **ArgoCD** rely on CRDs to extend Kubernetes' capabilities. By using CRDs and Operators, you can manage complex workloads in a Kubernetes-native way.