What are Cluster scoped resources  and namespace scoped resources in K8s?

## Cluster Scope resources

In Kubernetes, **cluster-scoped resources** are objects that **exist at the cluster level**, not within a specific namespace. These are resources that apply to the **entire cluster**, unlike namespaced resources (like Pods, Services, etc.) which are limited to a namespace.

## 1. Node

* **What it is:**  
  A Node is a physical or virtual machine in the Kubernetes cluster on which Pods run. Each node includes the container runtime, kubelet, and kube-proxy.
* **Why it's used:**  
  Nodes are the workhorses of the cluster—they actually run the workloads. The Kubernetes scheduler assigns Pods to Nodes based on resource availability and constraints.

## 2. ****Namespace****

* **What it is:**  
  A Namespace is a logical partition of cluster resources. Think of it as a virtual cluster within the Kubernetes cluster.
* **Why it's used:**  
  To separate workloads by teams, environments (dev/test/prod), or projects. Each namespace is isolated from others, allowing for multi-tenant usage in the same cluster.

## 3. ****PersistentVolume (PV)****

* **What it is:**  
  A PV is a piece of networked storage in the cluster that has been provisioned by an admin or automatically via a StorageClass. It exists independently of any Pod.
* **Why it's used:**  
  To provide durable, persistent storage that outlives Pod lifecycles—ideal for databases, file storage, and apps that require state retention.

## 4. ****CustomResourceDefinition (CRD)****

* **What it is:**  
  A way to define your own custom resource types in Kubernetes.
* **Why it's used:**  
  Extends Kubernetes to support new APIs (e.g., for Argo CD apps)

## 5. ****ClusterRole****

* **What it is:**  
  Defines a set of permissions **at the cluster level** (or for all namespaces).
* **Why it's used:**  
  Needed when you want to give access to resources **across all namespaces** or to cluster-scoped resources.

## 7. ****StorageClass****

* **What it is:**  
  Defines **how** a PersistentVolume is dynamically provisioned (e.g., with SSD, HDD, cloud storage, etc.).
* **Why it's used:**  
  Provides control over types and performance of storage used by apps

## 8. ****APIService****

* **What it is:**  
  A way to **register an external API** with the Kubernetes API server.
* **Why it's used:**  
  Enables extending Kubernetes with aggregated APIs (used by metrics server, custom APIs).

## 9. ****PodSecurityPolicy**** / ****PodSecurityAdmission****

* **What it is:**  
  Controls what pods are allowed to do (e.g., run as root, mount volumes).
* **Why it's used:**  
  To enforce security standards across the cluster.

## 10. ****Runtime Class****

* **What it is:**  
  Allows selecting different container runtimes (e.g., runc, gVisor).
* **Why it's used:**  
  To choose the right runtime for performance or security.

## ****Namespace-scoped resources****

**Namespace-scoped resources** are Kubernetes objects that **exist and operate within a specific namespace.**They are **isolated per namespace** and cannot be directly accessed from other namespaces.

## 1. Pod

**What it is:**  
A basic unit in Kubernetes that runs one or more containers.

**Why it's used:**  
To deploy and run containerized applications inside the cluster.

## 2. Deployment

**What it is:**  
Manages stateless pods with rolling updates and rollback support.

**Why it's used:**  
To ensure a specific number of replicas are running and to handle automated updates.

## 3. ReplicaSet

**What it is:**  
A controller that maintains a stable set of pod replicas.

**Why it's used:**  
To ensure high availability and fault tolerance by replicating pods.

## 4. DaemonSet

**What it is:**  
Ensures a copy of a specific pod runs on all (or selected) nodes.

**Why it's used:**  
To deploy node-specific services like monitoring agents or log collectors.

## 5. Job

**What it is:**  
Runs a pod or set of pods to perform a one-time task.

**Why it's used:**  
To execute batch or background jobs that need to complete and exit.

## 6. Ingress

**What it is:**  
A resource that manages external HTTP/HTTPS access to services.

**Why it's used:**  
To route external traffic to internal services using URLs, hostnames, etc.

## 7. ConfigMap

**What it is:**  
Stores non-sensitive key-value configuration data.

**Why it's used:**  
To inject app configuration without rebuilding container images.

## 8. Secret

**What it is:**  
Stores sensitive data like passwords, tokens, and TLS certs in Base64 format.

**Why it's used:**  
To safely inject secure data into pods.

## 9. ServiceAccount

**What it is:**  
An account identity used by pods to interact with the Kubernetes API.

**Why it's used:**  
To control access rights and associate pods with specific permissions.

## 10. HorizontalPodAutoscaler

**What it is:**  
A controller that automatically scales the number of pods in a deployment based on metrics.

**Why it's used:**  
To ensure optimal performance and resource efficiency under varying load.