**Kubernetes resources**

**Cluster Scoped Resources**

* What are Cluster Scoped Resources
* Cluster-scoped resources are not associated with any namespace.
* They represent global configuration or objects that apply across the entire Kubernetes cluster.
* Why Cluster Scoped Resources are necessary

1. Global Scope Configuration

* Some resources are relevant to the whole cluster, not a specific namespace. For example: Nodes, Persistent Volumes (PVs) etc.

1. Centralized Access Control (RBAC)

* ClusterRoles and ClusterRoleBindings allow defining and applying permissions across all namespaces or to non-namespaced resources.
* Example: Granting read-only access to all resources in the cluster to an admin.

1. Shared Infrastructure & Services

* Resources like StorageClasses, IngressClasses, and CRDs define behavior or capabilities that are shared across workloads in many namespaces.
* This avoids duplication and ensures consistency.

1. Custom Resource Definitions (CRDs)

* CRDs define new types of resources. They need to be available cluster-wide so all parts of the cluster can recognize and use them.
* Once installed, any namespace can create instances of the custom resource.

1. Cluster-Level Webhooks & APIs

* MutatingWebhookConfiguration and ValidatingWebhookConfiguration apply to requests across the entire cluster.
* APIService enables extending the Kubernetes API with aggregated APIs, which must be globally accessible.

1. Better Control and Visibility

* Centralizing control of global resources simplifies auditing, monitoring, and management of the cluster.
* Common Cluster Scoped Resources
* **Node:** Represents a worker node in the cluster
* **Namespace:** A logical partition of the cluster
* **PersistentVolume (PV):** A cluster-wide pool of storage volumes
* **CustomResourceDefinition (CRD):** Defines custom resource types at the cluster level
* **ClusterRole:** Defines cluster-wide permissions (used with RBAC)
* **ClusterRoleBinding:** Binds ClusterRole to a user, group, or service account at cluster scope
* **StorageClass:** Defines storage provisioning strategies
* **VolumeAttachment:** Represents attachment of a volume to a node
* **MutatingWebhookConfiguration:** Webhooks that modify requests at the cluster level
* **ValidatingWebhookConfiguration:** Webhooks that validate requests at the cluster level
* **APIService:** Registers an API group to be served by the API aggregator

**Namespace Scoped Resources**

* What are Namespace Scoped Resources
* Namespace-scoped resources are tied to a specific namespace.
* They only exist and operate within that namespace, and they cannot interact with resources in other namespaces unless explicitly allowed.
* Why Namespace Scoped Resources are Important

1. Logical Isolation Within a Cluster

* Namespaces provide a way to divide cluster resources between multiple users, teams, or applications without spinning up multiple clusters.

1. Access Control and Security (RBAC)

* You can grant fine-grained access to resources per namespace using Role and RoleBinding.
* This limits what users or service accounts can see or do in specific namespaces.

1. Resource Quotas and Limits

* Namespace allows you to define ResourceQuotas (Limit the CPU, memory, and object count (pods, PVCs, etc.) a namespace can use).
* Namespace allows you to apply LimitRanges (Set default resource requests/limits for pods and containers in a namespace).

1. Simplified Management and Organization

* It's easier to manage related resources (e.g., a web app's Pods, Services, ConfigMaps, etc.)
* Cleanup is easy: delete the namespace to remove everything inside it.

1. Supports CI/CD, Environments, and Multitenancy

By using Namespace you can:

* Separate **environments**: dev, staging, production
* Isolate **CI/CD pipelines** for testing without affecting production
* Allow **multiple tenants** or clients to share the same cluster securely

1. Prevents Naming Collisions

* Resources in different namespaces can have the same name, namespace avoids conflicts and simplifies naming conventions.
* Common Namespace Scoped Resources
* **Pod:** A running container or group of containers
* **Deployment:** Manages ReplicaSets and Pods
* **ReplicaSet:** Ensures a specified number of pod replicas are running
* **StatefulSet:** Like Deployment, but for stateful apps
* **DaemonSet:** Ensures a copy of a pod runs on each node (in namespace)
* **Service:** Exposes a set of Pods in the same namespace
* **ConfigMap:** Stores configuration data in key-value pairs
* **Secret:** Stores sensitive data like passwords and tokens
* **PersistentVolumeClaim (PVC):** Requests storage from a PersistentVolume
* **Role:** Defines permissions for resources within the namespace
* **RoleBinding:** Binds a Role to a user/service account in the namespace
* **Job / CronJob:** For running batch or scheduled tasks
* **Ingress:** Manages external access to services in a namespace
* **ServiceAccount:** Identity for pods to interact with the API