Core Concepts

- **Cluster** – A set of nodes (machines) running Kubernetes.
- **Node** – A single machine in a cluster (can be virtual or physical).
- **Pod** – The smallest deployable unit; can contain one or more containers.
- **Container** – A lightweight unit of software that runs in a pod.
- **Deployment** – Manages a set of identical pods; ensures the desired state.
- **ReplicaSet** – Ensures a specified number of pod replicas are running.
- **Namespace** – Virtual clusters within a Kubernetes cluster to isolate resources.

Workloads & Controllers

- **StatefulSet** – Manages stateful applications (persistent identity and storage).
- **DaemonSet** – Ensures a copy of a pod runs on all (or some) nodes.
- **Job** – Runs a pod to completion (for batch processing).
- **CronJob** – Runs jobs on a schedule.
- **HorizontalPodAutoscaler (HPA)** – Scales pods based on CPU/memory usage.

Networking & Services

- **Service** – Exposes pods to network (ClusterIP, NodePort, LoadBalancer).
- **Ingress** – Manages external access to services (like HTTP routing).
- **Ingress Controller** – Implements the Ingress rules.
- **ClusterIP** – Default service type; internal access only.
- **NodePort** – Exposes service on each node's IP at a static port.
- **LoadBalancer** – Exposes service externally via a cloud load balancer.
- **DNS** – Internal name resolution between services and pods.
- **CNI (Container Network Interface)** – Plugin system for networking.

Security

- **ServiceAccount** – Identity for processes that run in a pod.
- **RBAC (Role-Based Access Control)** – Permissions system for users and services.
- **NetworkPolicy** – Controls traffic between pods.

- **PodSecurityPolicy** (Deprecated) – Sets security rules for pod specs.
- **Secrets** – Stores sensitive data like passwords or API keys.
- **ConfigMap** – Stores non-confidential configuration data.

💾 Storage

- **Volume** – Abstracted storage that a pod can access.
- **PersistentVolume (PV)** – A piece of storage provisioned in the cluster.
- **PersistentVolumeClaim (PVC)** – Request for storage by a user.
- **StorageClass** – Describes different types of storage (e.g., SSD, HDD).
- **CSI (Container Storage Interface)** – Standard for integrating storage systems.

📦 Configuration & Package Management

- **Helm** – Package manager for Kubernetes.
- **Chart** – A Helm package containing YAML files and templates.
- **Manifest** – YAML file that defines a Kubernetes resource.
- **kustomize** – Native tool to customize YAML configurations.

🧩 Observability & Debugging

- **kubectl** – Command-line tool to interact with the Kubernetes API.
- **Logs** – Logs from containers for debugging.
- **Metrics Server** – Collects resource usage data.
- **Probes**:
    - **Liveness Probe** – Checks if the app is alive.
    - **Readiness Probe** – Checks if the app is ready to receive traffic.
    - **Startup Probe** – Checks if the app has started correctly.

🧠 Advanced Concepts

- **Custom Resource Definition (CRD)** – Extend Kubernetes with your own API objects.
- **Operator** – A controller for managing custom resources and application logic.
- **Admission Controller** – Intercepts API requests before persistence.
- **Taints and Tolerations** – Controls which pods can be scheduled on which nodes.

- **Affinity & Anti-affinity** – Rules for scheduling pods on nodes.

☁️ Cloud-Native & Ecosystem

- **Kubelet** – Runs on each node, ensures containers are running.
- **Kube-proxy** – Handles networking rules on nodes.
- **API Server** – Central management point (exposes Kubernetes API).
- **Controller Manager** – Runs background controllers to regulate cluster state.
- **etcd** – Key-value store for all cluster data.
- **Scheduler** – Assigns pods to nodes based on requirements and constraints.