



Kubernetes

Cheat Sheet

Follow  [@techyoutbe](https://www.techyoutbe.com)

- **Cluster:** A group of connected computers (nodes) that run applications.
- **Node:** A single computer in a cluster that runs applications.
- **Pod:** The smallest unit in Kubernetes that can run one or more containers.
- **Namespace:** A way to divide resources in a cluster for different projects or teams.
- **Deployment:** Manages a set of identical pods to ensure the correct number are running.
- **ReplicaSet:** Ensures a specified number of pod copies are running at all times.
- **DaemonSet:** Ensures a pod runs on all or some nodes.
- **StatefulSet:** Manages stateful applications, keeping track of each pod's identity.
- **Job:** Runs a task until it completes successfully.
- **CronJob:** Runs tasks on a scheduled basis, like a cron job in Unix.
- **Service:** Exposes a set of pods as a network service.
- **Ingress:** Manages external access to services, usually HTTP.
- **ConfigMap:** Stores configuration data as key-value pairs.
- **Secret:** Stores sensitive data, like passwords and tokens.
- **Volume:** Provides storage for containers.
- **PersistentVolume (PV):** A piece of storage that an administrator sets up.
- **PersistentVolumeClaim (PVC):** A request for storage by a user.
- **Kubelet:** The agent that runs on each node to manage pods.
- **Kube-Proxy:** Manages network rules on nodes.
- **Controller Manager:** Manages controllers that regulate the state of the cluster.
- **Scheduler:** Decides which nodes will run new pods.
- **Etcd:** A key-value store that stores all cluster data.
- **Kubectl:** The command-line tool to interact with the Kubernetes API.
- **Helm:** A package manager for Kubernetes applications.
- **Horizontal Pod Autoscaler:** Automatically adjusts the number of pods based on resource usage.
- **Cluster Autoscaler:** Automatically adjusts the number of nodes in a cluster based on resource usage.
- **Label:** Key-value pairs attached to objects for organizing and selecting them.
- **Annotation:** Metadata attached to objects to provide additional information.
- **Taints:** Prevents specific pods from running on certain nodes.
- **Tolerations:** Allows pods to run on nodes with specific taints.
- **Affinity/Anti-Affinity:** Rules that specify which nodes can or cannot run specific pods.
- **Role-Based Access Control (RBAC):** Manages who can do what in the cluster.
- **ServiceAccount:** An identity for processes running in pods to interact with the Kubernetes API.
- **ClusterRole:** Defines permissions that apply across the entire cluster.
- **Role:** Defines permissions within a specific namespace.
- **RoleBinding:** Grants a Role's permissions to a user or group within a namespace.
- **ClusterRoleBinding:** Grants a ClusterRole's permissions to a user or group across the entire cluster.
- **NetworkPolicy:** Controls the traffic between pods in the cluster.
- **PodSecurityPolicy:** Defines security rules that pods must follow.
- **PodDisruptionBudget (PDB):** Limits the number of pods that can be unavailable during maintenance.
- **Ingress Controller:** Manages Ingress resources to provide HTTP and HTTPS routing.
- **CoreDNS:** A DNS server for the cluster, providing name resolution for services.
- **StorageClass:** Describes different types of storage available in the cluster.
- **Init Containers:** Special containers that run before the main containers in a pod start.
- **Sidecar Container:** A helper container that runs alongside the main container in a pod.
- **Readiness Probe:** Checks if a container is ready to start accepting traffic.
- **Liveness Probe:** Checks if a container is still running and should be restarted if not.
- **Headless Service:** A service without a cluster IP, used to directly access pods.
- **LoadBalancer Service:** Exposes a service externally using a cloud provider's load balancer.
- **ClusterIP Service:** Exposes a service internally within the cluster.



Kubernetes

Cheat Sheet (Part 2)

Follow  [@techyoutbe](#)

- **NodePort Service:** Exposes a service on a static port on each node.
- **Endpoints:** A list of IP addresses and ports that a service forwards traffic to.
- **Resource Quotas:** Limits the amount of resources a namespace can use.
- **LimitRange:** Defines resource usage limits for containers in a namespace.
- **Custom Resource Definition (CRD):** Extends Kubernetes to manage custom resources.
- **Operator:** A custom controller that manages complex applications on Kubernetes.
- **Admission Controller:** Intercepts requests to the Kubernetes API for validation and mutation.
- **Finalizer:** Ensures that specific cleanup steps are completed before an object is deleted.
- **Horizontal Pod Autoscaler (HPA):** Automatically scales the number of pods based on CPU/memory usage.
- **Vertical Pod Autoscaler (VPA):** Adjusts the resource limits and requests for running pods.
- **Cluster Autoscaler:** Automatically adjusts the size of the Kubernetes cluster by adding or removing nodes.
- **Affinity Rules:** Specify rules about which nodes can host a pod.
- **Anti-Affinity Rules:** Specify rules about which nodes should not host a pod.
- **Init Containers:** Special containers that run before the main containers in a pod start.
- **Sidecar Containers:** Helper containers that run alongside the main container in a pod.
- **Resource Requests:** Specify the minimum amount of resources a container needs.
- **Resource Limits:** Specify the maximum amount of resources a container can use.
- **PersistentVolumeClaim (PVC):** A request for storage by a user.
- **EmptyDir:** A temporary directory that is created when a pod is assigned to a node.
- **ConfigMap:** Provides configuration data to pods.
- **Secret:** Stores sensitive data, such as passwords and keys.
- **Security Context:** Defines security settings for a pod or container.
- **ServiceAccount:** Provides an identity for processes running in pods.
- **ClusterRoleBinding:** Binds a ClusterRole to a user or group for the entire cluster.
- **RoleBinding:** Binds a Role to a user or group within a namespace.
- **Pod Preset:** Injects certain information, like secrets or volume mounts, into pods at creation.
- **Priority Class:** Specifies the priority of pods to influence their scheduling.
- **Horizontal Pod Autoscaler:** Scales the number of pods based on observed CPU/memory utilization.
- **Vertical Pod Autoscaler:** Adjusts the CPU and memory requests/limits for pods.
- **Cluster Autoscaler:** Automatically adds/removes nodes based on cluster usage.
- **Self-healing:** Automatically replaces and reschedules failed containers.
- **Secrets Management:** Manages sensitive information like passwords and API keys.
- **Resource Quotas:** Limits the amount of resources that can be consumed in a namespace.
- **Default Namespace:** The default namespace for Kubernetes objects without a specified namespace.
- **Master Node:** Controls and manages the Kubernetes cluster.
- **Worker Node:** Runs applications and workloads in pods.
- **Controller Manager:** Runs controllers to regulate the state of the cluster.
- **Scheduler:** Assigns pods to nodes based on resource availability.
- **Etcd:** Stores all cluster data, ensuring data consistency and availability.
- **Kubelet:** Manages pod operations on each node.
- **Kube-Proxy:** Manages network rules and traffic routing for services.
- **Kubectl:** Command-line tool to interact with the Kubernetes API.
- **Helm:** Package manager for managing Kubernetes applications.
- **Helm Chart:** Pre-configured Kubernetes resources packaged for easy deployment.
- **Kustomize:** Tool for customizing Kubernetes YAML configurations.
- **Admission Controller:** Intercepts requests to the Kubernetes API for validation and mutation.
- **Custom Resource Definition (CRD):** Extends Kubernetes by defining custom resources.
- **Operator:** Custom controllers for managing complex applications.
- **Kubeadm:** Tool for initializing and managing Kubernetes clusters.
- **Minikube:** Tool for running a single-node Kubernetes cluster locally for testing and development.



[@techyoutbe](#)