

# Glossary: Kubernetes Basics

Term	Definition
<b>Automated bin packing</b>	Increases resource utilization and cost savings using a mix of critical and best-effort workloads.
<b>Batch execution</b>	Manages batch and continuous integration workloads and automatically replaces failed containers, if configured.
<b>Cloud Controller Manager</b>	A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster.
<b>Cluster</b>	A set of worker machines, called nodes, that run containerized applications. Every cluster has at least one worker node.
<b>Container Orchestration</b>	Container orchestration is a process that automates the container lifecycle of containerized applications.
<b>Container Runtime</b>	The container runtime is the software that is responsible for running containers.
<b>Control Loop</b>	A non-terminating loop that regulates the state of a system. A thermostat is an example of a control loop.
<b>Control plane</b>	The container orchestration layer that exposes the API and interfaces to define, deploy, and manage the lifecycle of containers.
<b>Controller</b>	In Kubernetes, controllers are control loops that watch the state of your cluster, then make or request changes where needed. Each controller tries to move the current cluster state closer to the desired state.
<b>Data (Worker) Plane</b>	The layer that provides capacity such as CPU, memory, network, and storage so that the containers can run and connect to a network.
<b>DaemonSet</b>	Ensures a copy of a Pod is running across a set of nodes in a cluster.
<b>Declarative Management</b>	A desired state that can be expressed (for example, the number of replicas of a specific application), and Kubernetes will actively work to ensure that the observed state matches the desired state.
<b>Deployment</b>	An object that provides updates for both Pods and ReplicaSets. Deployments run multiple replicas of an application by creating ReplicaSets and offering additional management capabilities on top of those ReplicaSets. In addition, deployments are suitable for stateless applications.
<b>Designed for extensibility</b>	Adds features to your cluster without adding or modifying source code.
<b>Docker Swarm</b>	Automates the deployment of containerized applications but was designed specifically to work with Docker Engine and other Docker tools making it a popular choice for teams already working in Docker environments.
<b>Ecosystem</b>	A composition of services, support and tools that are widely available. The Kubernetes ecosystem is a large, rapidly growing ecosystem where its services, support, and tools are widely available.
<b>etcd</b>	A highly available key value store that contains all the cluster data. For any deployment, the deployment configuration is stored in etcd. It is the source of truth for the state in a Kubernetes cluster, and the system works to bring the cluster state into line with what is stored in etcd.
<b>Eviction</b>	Process of terminating one or more Pods on Nodes.
<b>Imperative commands</b>	Create, update, and delete live objects directly.
<b>Imperative Management</b>	Defining steps and actions to get to a desired state.
<b>Ingress</b>	An API object that manages external access to the services in a cluster, typically HTTP.
<b>IPv4/IPv6 dual stack</b>	Assigns both IPv4 and IPv6 addresses to Pods and Services.
<b>Job</b>	A finite or batch task that runs to completion.
<b>Kubecttl</b>	Also known as kubecttl Command line tool for communicating with a Kubernetes cluster's control plane, using the Kubernetes API.
<b>Kubelet</b>	The kubelet is the primary "node agent" that runs on each node. The kubelet takes a set of PodSpecs (a YAML or JSON object that describes a pod) provided primarily through the apiserver and ensures that the containers described in those PodSpecs are running and healthy. The kubelet doesn't manage containers which were not created by Kubernetes.
<b>Kubernetes</b>	Is the de facto open-source platform standard for container orchestration. It was developed by Google and is maintained by the Cloud Native Computing Foundation (CNCF). Kubernetes automates container management tasks, like deployment, storage provisioning, load balancing and scaling, service discovery, and fixing failed containers. Its open-source toolset and wide array of functionalities are very attractive to leading cloud providers, who both support it, and in some cases, also offer fully managed Kubernetes services.
<b>Kubernetes API</b>	The application that serves Kubernetes functionality through a RESTful interface and stores the state of the cluster.

<b>Kubernetes API Server</b>	The Kubernetes API server validates and configures data for the api objects which include pods, services, replication controllers, and others. The API Server services REST operations and provides the frontend to the cluster's shared state through which all other components interact.
<b>Kubernetes Controller Manager</b>	Runs all the controller processes that monitor the cluster state and ensures that the actual state of a cluster matches the desired state. Examples of controllers that ship with Kubernetes are the replication controller, endpoints controller, namespace controller, and service accounts controller.
<b>Kubernetes Cloud Controller Manager</b>	A Kubernetes control plane component that embeds cloud-specific control logic. The cloud controller manager lets you link your cluster into your cloud provider's API, and separates out the components that interact with that cloud platform from components that only interact with your cluster.
<b>Kubernetes Proxy</b>	A network proxy that runs on each node in a cluster. This proxy maintains network rules that allow communication to Pods running on nodes—in other words, communication to workloads running on the cluster. The user must create a service with the apiserver API to configure the proxy.
<b>kube-scheduler</b>	Control plane component that watches for newly created Pods with no assigned node, and selects a node for them to run on.
<b>Label Selector</b>	Allows users to filter a list of resources based on labels.
<b>Labels</b>	Tags objects with identifying attributes that are meaningful and relevant to users.
<b>Load balancing</b>	Balances traffic across Pods for better performance and high availability.
<b>Marathon</b>	Is an Apache Mesos framework. Apache Mesos is an open-source cluster manager developed by UC Berkeley. It lets users scale container infrastructure through the automaton of most management and monitoring tasks.
<b>Namespace</b>	An abstraction used by Kubernetes to support isolation of groups of resources within a single cluster.
<b>Node</b>	The worker machine in a Kubernetes cluster. User applications are run on nodes. Nodes can be virtual or physical machines. Each node is managed by the control plane and is able to run Pods.
<b>Nomad</b>	(Hashicorp) is a free and open-source cluster management and scheduling tool that supports Docker and other applications on all major operating systems across all infrastructure, whether on-premises or in the cloud. This flexibility lets teams work with any type and level of workload.
<b>Object</b>	An entity in the Kubernetes system. The Kubernetes API uses these entities to represent the state of your cluster.
<b>Persistence</b>	Ensures that an object exists in the system, until the object is modified or removed.
<b>Preemption</b>	Logic in Kubernetes helps a pending Pod to find a suitable Node by evicting low priority Pods existing on that Node.
<b>Self-healing</b>	Restarts, replaces, reschedules, and kills failing or unresponsive containers.
<b>Service</b>	An abstract way to expose an application running on a set of Pods as a network service.
<b>Service Discovery</b>	Discovers Pods using their IP addresses or a single DNS name.
<b>StatefulSet</b>	Manages the deployment and scaling of a set of Pods, and provides guarantees about the ordering and uniqueness of these Pods.
<b>Storage</b>	A data store that supports persistent and temporary storage for Pods.
<b>Storage Orchestration</b>	Automatically mounts your chosen storage system whether from local storage, network storage, or public cloud.
<b>Pod</b>	The smallest and simplest Kubernetes object. Represents a process running in a cluster; it also represents a single instance of an application running in a cluster. Usually, a Pod wraps a single container but, in some cases encapsulates multiple tightly coupled containers that share resources.
<b>Proxy</b>	In computing, a proxy is a server that acts as an intermediary for a remote service.
<b>ReplicaSet</b>	A ReplicaSet (aims to) maintain a set of replica Pods running at any given time.
<b>Workload</b>	A workload is an application running on Kubernetes.



# **Skills** Network