

Kubernetes

[return to the Linux Foundation](#)

Note: The following is an excerpt from the official [Kubernetes \(K8\) home page](#). It is provided here as a convenience and is not authoritative. Refer to the original document as the authoritative reference.

Table 1: Data sheet for Linux Kubernetes

Title	Kubernetes
Acronym	K8s
Version	v1.18
Operating Systems	Darwin, Linux, Windows
Downloads	https://kubernetes.io/docs/setup/release/notes/#downloads-for-v1180
Repository	https://github.com/kubernetes/kubernetes
Supported Languages	
License	Apache License 2.0
Reference	https://kubernetes.io/docs/reference/

Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications. It groups containers that make up an [application](#) into logical units for easy management and [discovery](#). Kubernetes builds upon 15 years of experience of running production workloads at Google, combined with best-of-breed ideas and practices from the community

It is based on the principles of:

- **Planet Scale:** Designed on the same principles that allows Google to run billions of containers a week, Kubernetes can scale without increasing your ops team.
- **Never Outgrow:** Whether testing locally or running a global enterprise, Kubernetes flexibility grows with you to deliver your applications consistently and easily no matter how complex your need is.
- **Run Anywhere:** Kubernetes is open source giving you the freedom to take advantage of on-premises, hybrid, or public cloud infrastructure, letting you effortlessly move workloads to where it matters to you.

Kubernetes features:

- **Service discovery and load balancing:** no need to use an unfamiliar service discovery mechanism.
- **Storage Orchestration:** ability to automatically mount storage mechanisms of choice.
- **Automated rollouts and rollbacks:** progressive change rollouts including application health monitoring.
- **Automatic bin packing:** automatically places containers based on resource requirements and other constraints.
- **Batch execution:** manages batch and CI workloads.
- **Service Topology:** Route service traffic based on cluster topology.
- **Self-healing:** manage failed and failing containers and dead nodes.

- **Secret and configuration management:** Deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.
- **IPv4/IPv6 dual-stack:** Allocate of IPv4 and IPv6 addresses to Pods and Services
- **Horizontal scaling:** Scale applications with simple CLI and/or UI or in automatically in response to CPU usage.

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:ra:xapend:xapend.b_stds:defact:linuxf:kubernetes

Last update: **2021/11/09 15:41**

