

What is container orchestration?

Container orchestration is the process of automating the deployment, management, scaling, and networking of containers throughout their lifecycle. This helps you deploy the same application across different environments without having to redesign it, which is especially important for enterprises that need to deploy and manage hundreds or thousands of [Linux® containers](#) and hosts.

Managing the lifecycle of containers with orchestration also supports DevOps teams who integrate it into continuous integration and continuous delivery (CI/CD) workflows. Along with [application programming interfaces \(APIs\)](#) and DevOps teams, containerized [microservices](#) are the foundation for [cloud-native applications](#).

What is container orchestration used for?

Use container orchestration to automate and manage tasks such as:

- Provisioning and deployment
- Configuration and scheduling
- Resource allocation
- Container availability
- Scaling or removing containers based on balancing workloads across your infrastructure
- Load balancing and traffic routing
- Monitoring container health
- Configuring applications based on the container in which they will run
- Keeping interactions between containers secure



kubernetes



Amazon EKS



AWS ECS



HashiCorp
Nomad



Apache
MESOS™



Main components of Kubernetes:

- **Cluster:** A control plane and one or more compute machines, or nodes.
- **Control plane:** The collection of processes that control Kubernetes nodes. This is where all task assignments originate.
- **Kubelet:** This service runs on nodes and reads the container manifests and ensures the defined containers are started and running.
- **Pod:** A group of one or more containers deployed to a single node. All containers in a pod share an IP address, IPC, hostname, and other resources.

How does container orchestration work?

When you use a container orchestration tool, such as Kubernetes, you will describe the configuration of an application using either a YAML or JSON file. The configuration file tells the configuration management tool where to find the container images, how to establish a network, and where to store logs.

When deploying a new container, the container management tool automatically schedules the deployment to a cluster and finds the right host, considering any defined requirements or restrictions. The orchestration tool then manages the container's lifecycle based on the specifications that were determined in the compose file.

<https://www.cncf.io/training/certification/software-conformance/#>

Certified Kubernetes Software Conformance

Certified Kubernetes Conformance Program benefits























































1. **Consistency:** Users want consistency when interacting with any installation of Kubernetes.
2. **Timely updates:** To remain certified, vendors need to provide the latest version of Kubernetes yearly or more frequently, so you can be sure that you'll always have access to the latest features the community has been working hard to deliver.
3. **Confirmability:** Any end user can confirm that their distribution or platform remains conformant by running the identical open-source conformance application (Sonobuoy) that was used to certify.

www.linkedin.com/in/ragasudha-m

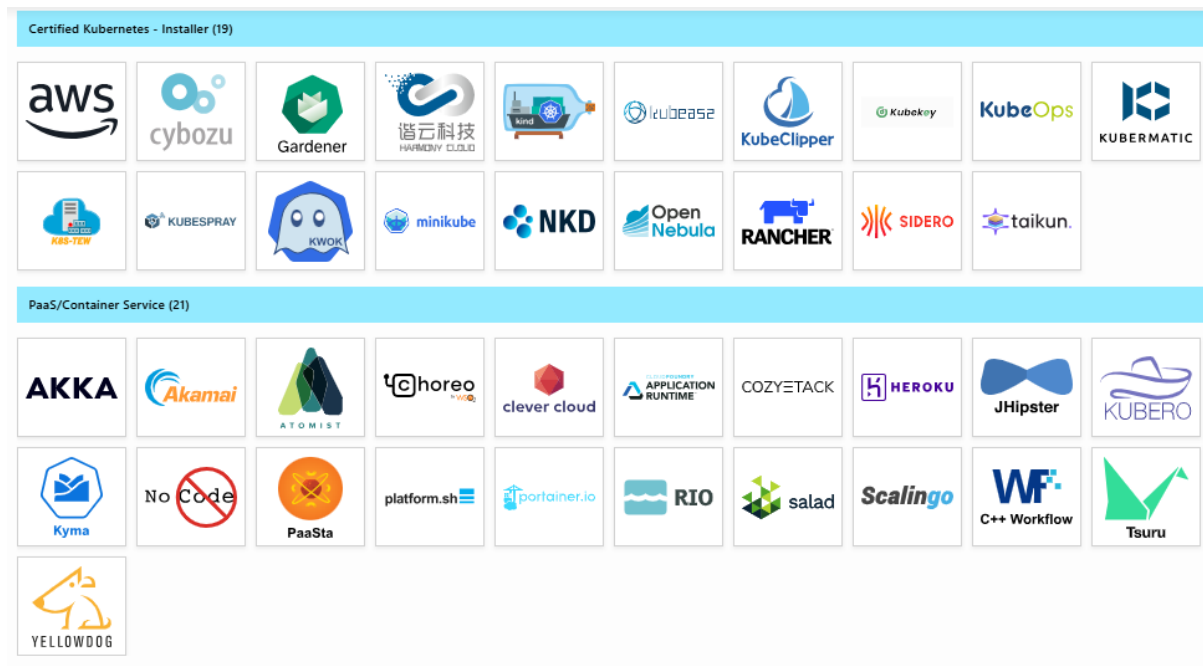
Look for products that proudly display the Certified Kubernetes logo

The following products have earned Certified Kubernetes status:

Platform

Certified Kubernetes - Distribution (60)									
									
									
									
									
									
									
Certified Kubernetes - Hosted (58)									
									
									
									
									
									
									

www.linkedin.com/in/ragasudha-m



In Kubernetes, the terms **distribution**, **hosted**, and **installer** refer to different methods and tools for deploying, managing, and running Kubernetes clusters. These terms reflect the different ways users can choose to set up and operate their Kubernetes environment depending on their needs, expertise, and resources.

Kubernetes Distribution

A **Kubernetes distribution** refers to a specific version or flavour of Kubernetes that comes with additional tools, configurations, and features to facilitate the setup and operation of Kubernetes clusters. These distributions often include pre-packaged components like networking, storage solutions, and monitoring systems that are tested and optimized for certain use cases.

Popular Kubernetes Distributions:

- **Kubeadm:** A tool provided by the Kubernetes community that simplifies the process of installing Kubernetes clusters manually. It's an installer and configuration tool but requires you to set up the infrastructure, networking, and storage.
- **OpenShift:** A Kubernetes distribution developed by Red Hat that includes additional features, like a web-based user interface, CI/CD pipelines, and security enhancements. OpenShift is often favoured by enterprises for its extra tools and integration with Red Hat's ecosystem.
- **Rancher:** A complete platform for managing Kubernetes clusters. Rancher provides a simplified interface for managing multi-cluster Kubernetes deployments, and it includes its own distribution of Kubernetes.
- **Amazon EKS (Elastic Kubernetes Service):** Amazon's managed Kubernetes service, based on a distribution of Kubernetes optimized for AWS.
- **Google GKE (Google Kubernetes Engine):** Google's managed Kubernetes service, based on the Kubernetes distribution optimized for the Google Cloud Platform.

www.linkedin.com/in/ragasudha-m

- **VMware Tanzu:** A Kubernetes distribution tailored for VMware environments, providing enhanced tools for deploying and managing Kubernetes in virtualized environments.

2. Hosted Kubernetes

A **hosted Kubernetes** service refers to a managed Kubernetes environment where the cloud provider or third-party vendor takes care of the infrastructure, maintenance, and management of Kubernetes clusters. With hosted Kubernetes, users can focus on deploying applications without worrying about the complexity of setting up and maintaining the Kubernetes control plane and underlying infrastructure.

Examples of Hosted Kubernetes Services:

- **Amazon EKS:** A fully managed Kubernetes service on AWS. Amazon takes care of the control plane, ensuring high availability, security patches, and scalability, while users manage only their workloads and application containers.
- **Google GKE:** Google's managed Kubernetes offering that simplifies the cluster setup process, offering deep integration with Google Cloud services.
- **Azure AKS (Azure Kubernetes Service):** Microsoft's managed Kubernetes service on Azure, offering similar benefits to EKS and GKE.
- **IBM Cloud Kubernetes Service:** A managed Kubernetes service from IBM Cloud, optimized for enterprise workloads.
- **DigitalOcean Kubernetes:** A simpler managed Kubernetes service with an easy-to-use interface for smaller-scale deployments.

With **hosted Kubernetes**, users do not need to manage the master nodes (control plane), as these are fully managed by the service provider. Users are only responsible for the worker nodes and the workloads running on the cluster.

3. Kubernetes Installer

A **Kubernetes installer** is a tool that simplifies the process of setting up a Kubernetes cluster, either on a local machine or across a distributed set of machines. These tools automate much of the complexity associated with setting up Kubernetes, handling tasks like configuring control plane components, worker nodes, networking, and security.

Popular Kubernetes Installers:

- **Kubeadm:** A tool from the Kubernetes community that provides a simple way to install and configure Kubernetes clusters. It is ideal for on-premises or virtualized environments. Kubeadm sets up the control plane and worker nodes, but other tools may be needed for network configuration and storage management.
- **Minikube:** A tool that creates a local Kubernetes cluster on a single machine (ideal for testing and development). Minikube runs Kubernetes in a virtual machine or container and is great for learning and experimentation.
- **Kind (Kubernetes IN Docker):** A tool to run Kubernetes clusters in Docker containers, useful for testing Kubernetes in a local environment. It's lightweight and easy to set up.

- **Kops:** A Kubernetes installer for cloud environments (primarily AWS). Kops automates the creation, configuration, and management of Kubernetes clusters in AWS, making it suitable for production-grade clusters.
- **Helm:** While not strictly an installer for Kubernetes itself, Helm is a package manager that simplifies the deployment and management of applications on a Kubernetes cluster, allowing users to install, upgrade, and manage complex applications with ease.
- **Cloud Provider-Specific Installers:**
 - **AWS CloudFormation:** AWS provides CloudFormation templates to deploy Kubernetes clusters on AWS, including Amazon EKS.
 - **Google Cloud Deployment Manager:** Like CloudFormation, Google provides templates to automate the creation of GKE clusters and associated resources.

Key Differences:

- **Distributions** are pre-configured versions of Kubernetes with added components or configurations tailored for specific use cases or environments (e.g., Red Hat OpenShift, Rancher, EKS).
- **Hosted Kubernetes** refers to managed services where the cloud provider takes care of the infrastructure and Kubernetes management, so users only need to focus on their workloads (e.g., GKE, AKS, EKS).
- **Installers** are tools that help you set up and configure a Kubernetes cluster manually, either locally or in the cloud. Examples include Kubeadm, Minikube, and Kops.

Choosing the Right Approach:

- **For learning and development:** Use **Minikube** or **Kind** to set up a local cluster.
- **For production environments:** Use **Kubeadm**, **Kops**, or a **hosted solution** like **Amazon EKS** or **Google GKE**, depending on your cloud preference and requirements.
- **For enterprises:** Consider **OpenShift** or **Rancher** for enhanced security, management, and support, or use a hosted service like **Azure AKS** or **IBM Cloud Kubernetes** for a managed environment.