GKE overview

Google Kubernetes Engine (GKE) provides a managed environment for deploying, managing, and scaling your containerized applications using Google infrastructure. The GKE environment consists of multiple machines (specifically, Compute Engine (/compute) instances) grouped together to form a cluster (/kubernetes-engine/docs/concepts/cluster-architecture).

Cluster orchestration with GKE

GKE clusters are powered by the Kubernetes (https://kubernetes.io) open source cluster management system. Kubernetes provides the mechanisms through which you interact with your cluster. You use Kubernetes commands and resources to deploy and manage your applications, perform administration tasks, set policies, and monitor the health of your deployed workloads.

Kubernetes draws on the same design principles that run popular Google services and provides the same benefits: automatic management, monitoring and liveness probes for application containers, automatic scaling, rolling updates, and more. When you run your applications on a cluster, you're using technology based on Google's 10+ years of experience running production workloads in containers.

Kubernetes on Google Cloud

When you run a GKE cluster, you also gain the benefit of advanced cluster management features that Google Cloud provides. These include:

- Google Cloud's load-balancing (/compute/docs/load-balancing-and-autoscaling) for Compute Engine instances
- Node pools (/kubernetes-engine/docs/concepts/node-pools) to designate subsets of nodes within a cluster for additional flexibility
- Automatic scaling (/kubernetes-engine/docs/cluster-autoscaler) of your cluster's node instance count
- Automatic upgrades (/kubernetes-engine/docs/concepts/node-auto-upgrades) for your cluster's node software
- Node auto-repair (/kubernetes-engine/docs/concepts/node-auto-repair) to maintain node health and availability
- Logging and monitoring (/monitoring/kubernetes-engine) with Google Cloud's operations suite for visibility into your cluster

Kubernetes versions and features

GKE cluster control planes are automatically upgraded to run new versions of Kubernetes as those versions become stable, so you can take advantage of newer features from the open source Kubernetes project.

Note: You can opt-in to newer versions of Kubernetes than those scheduled for automatic upgrades by manually initiating a control plane upgrade (/kubernetes-engine/docs/how-to/upgrading-a-cluster#upgrade_cp). For more information on what Kubernetes versions are available for upgrades, refer to the release notes (/kubernetes-engine/docs/release-notes) and documentation on versioning and upgrades (/kubernetes-engine/versioning-and-upgrades).

New features in Kubernetes are listed as **Alpha**, **Beta**, or **Stable**, depending upon their status in development. In most cases, Kubernetes features that are listed as **Beta** or **Stable** are included with GKE. Kubernetes **Alpha** features are available in special GKE alpha clusters (/kubernetes-engine/docs/concepts/alpha-clusters).

GKE workloads

GKE works with containerized applications. These are applications packaged into platform independent, isolated user-space instances, for example by using Docker (https://www.docker.com). In GKE and Kubernetes, these containers, whether for applications or batch jobs, are collectively called *workloads*. Before you deploy a workload on a GKE cluster, you must first package the workload into a container.

Note: Use Migrate for GKE (/kubernetes-engine/docs/concepts/migration) to containerize existing VM-based applications to run on GKE. Migrate for GKE provides a simple way to migrate your existing applications to containers without requiring access to your source code or rewriting the applications.

GKE supports the use of Docker containers. To learn more about the node images that GKE supports for your workloads, see Node images (/kubernetes-engine/docs/concepts/node-images).

Google Cloud provides continuous integration and continuous delivery tools to help you build and serve application containers. You can use Cloud Build (/build) to build container images (such as Docker) from a variety of source code repositories, and Artifact Registry (/artifact-registry) or Container Registry (/container-registry) to store and serve your container images.

Modes of operation

The level of flexibility, responsibility, and control that you require for your clusters determines the mode of operation to use in GKE. GKE clusters have two modes of operation to choose from:

- **Autopilot**: Manages the entire cluster and node infrastructure for you. Autopilot provides a hands-off Kubernetes experience so that you can focus on your workloads and only pay for the resources required to run your applications. Autopilot clusters are pre-configured with an optimized cluster configuration that is ready for production workloads.
- **Standard**: Provides you with node configuration flexibility and full control over managing your clusters and s created using the Standard mode, you determine the configurations needed

for your production workloads, and you pay for the nodes that you use.

For more information about these modes, and to learn more about Autopilot, see the Autopilot overview (/kubernetes-engine/docs/concepts/autopilot-overview).

What's next

- Explore the GKE tutorials (https://cloud.google.com/kubernetes-engine/docs/tutorials).
- Learn how to deploy a containerized application in GKE (/kubernetes-engine/docs/quickstart).
- Learn more about types of clusters (/kubernetes-engine/docs/concepts/types-of-clusters).
- Learn more about Kubernetes (https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/).

Last updated 2021-12-10 UTC.