This page explains how regional clusters work in Google Kubernetes Engine. You can create a regional cluster (/kubernetes-engine/docs/how-to/creating-a-regional-cluster) or learn more about the different types of clusters (/kubernetes-engine/docs/concepts/types-of-clusters).

By default, a cluster's control plane (master) and nodes all run in a single compute zone (/compute/docs/regions-zones/#available) that you specify when you create the cluster. Regional clusters increase the availability of both a cluster's control plane (master) and its nodes by replicating them across multiple zones of a region (/compute/docs/regions-zones/regions-zones#available). This provides the advantages of multi-zonal clusters (/kubernetes-engine/docs/concepts/types-of-clusters#multi-zonal_clusters), with the following additional benefits:

- If one or more (but not all) zones in a region experience an outage, the cluster's control plane remains accessible as long as one replica of the control plane available.
- During cluster maintenance such as a cluster upgrade, only one replica of the control plane is unavailable at a time, and the cluster is still operational.

By default, the control plane and each node pool is replicated across three zones of a region, but you can customize the number of replicas.

You cannot modify whether a cluster is zonal, multi-zonal, or regional after creating the cluster.

Regional clusters replicate cluster masters and nodes across multiple zones within in a single region (/compute/docs/regions-zones/regions-zones#available). For example, a regional cluster in the `us-east1` region creates replicas of the control plane and nodes in three `us-east1` zones: `us-east1-b`, `us-east1-c`, and `us-east1-d`. In the event of an infrastructure outage, your workloads continue to run, and nodes can be rebalanced manually or using the cluster autoscaler (/kubernetes-engine/docs/concepts/cluster-autoscaler).

Benefits of using regional clusters include:

- **Resilience from single zone failure.** Regional clusters are available across a *region* rather than a single zone within a region. If a single zone becomes unavailable, your Kubernetes control plane and your resources are not impacted.
- **Zero downtime master upgrades, master resize, and reduced downtime from master failures**. Regional clusters provide a high availability control plane, so you can access your control plane even during upgrades.

- By default, regional clusters consist of nine nodes spread evenly across three zones in a region. This consumes nine IP addresses. You can reduce the number of nodes down to one per zone, if desired. Newly-created Google Cloud accounts are granted only eight IP addresses per region, so you may need to request an increase in your quotas (/compute/quotas) for regional in-use IP addresses, depending on the size of your regional cluster. If you have too few available in-use IP addresses, cluster creation fails.
- For regional clusters that run GPUs (/kubernetes-engine/docs/concepts/gpus), you must either choose a region that has GPUs in three zones, or specify zones using the `--node-locations` flag. Otherwise, you may see an error like the following:

  For a complete list of regions and zones where GPUs are available, refer to GPUs on Compute Engine (/compute/docs/gpus).
- You can't create node pools in zones outside of the cluster's zones. However, you can change a cluster's zones (/kubernetes-engine/docs/how-to/managing-clusters#add_or_remove_zones), which causes all new and existing nodes to span those zones.

Regional clusters are offered at no additional charge (/kubernetes-engine/pricing).

Using regional clusters requires more of your project's regional quotas (/kubernetes-engine/quotas) than a similar zonal or multi-zonal cluster. Ensure that you understand your quotas and Google Kubernetes Engine pricing before using regional clusters. If you encounter an `Insufficient regional quota to satisfy request for resource` error, your request exceeds your available quota in the current region.

Additionally, you are charged for node-to-node traffic across zones. For example, if a workload running in one zone needs to communicate with a workload in a different zone, the cross-zone traffic incurs cost. For more information, see Egress between zones in the same region (per GB) (/compute/network-pricing#general) in the Compute Engine pricing page.

Persistent storage disks are zonal resources. When you add persistent storage (/kubernetes-engine/docs/how-to/stateful-apps#requesting_persistent_storage_in_a_statefulset) to your cluster, unless a zone is specified, GKE assigns the disk to a single zone. GKE chooses the zone at random. When using a StatefulSet, the provisioned persistent disks for each replica are spread across zones.

Once a persistent disk is provisioned, any Pods referencing the disk are scheduled to the same zone as the disk.

A read-write persistent disk cannot be attached to multiple nodes.

Keep the following considerations in mind when using the cluster autoscaler (/kubernetes-engine/docs/concepts/cluster-autoscaler) to automatically scale node pools in regional clusters.

You can also learn more about Autoscaling limits (/kubernetes-engine/docs/concepts/cluster-autoscaler#autoscaling_limits) for regional clusters or about how Cluster Autoscaler balances across zones (/kubernetes-engine/docs/concepts/cluster-autoscaler#balancing_across_zones).

To maintain capacity in the unlikely event of zonal failure, you can allow GKE to overprovision your scaling limits, to guarantee a minimum level of availability even when some zones are unavailable.

For example, if you overprovision a three-zone cluster to 150% (50% excess capacity), you can ensure that 100% of traffic is routed to available zones if one-third of the cluster's capacity is lost. In the above example, you would accomplish this by specifying a maximum of six nodes per zone rather than four. If one zone fails, the cluster scales to twelve nodes in the remaining zones.

Similarly, if you overprovision a two-zone cluster to 200%, you can ensure that 100% of traffic is rerouted if half of the cluster's capacity is lost.

You can learn more about the cluster autoscaler (/kubernetes-engine/docs/concepts/cluster-autoscaler) or read the FAQ for autoscaling (https://github.com/kubernetes/autoscaler/blob/master/cluster-autoscaler/FAQ.md) in the Kubernetes documentation.

- Create a regional cluster (/kubernetes-engine/docs/how-to/creating-a-regional-cluster).
- Learn more about the different types of clusters (/kubernetes-engine/docs/concepts/types-of-clusters).
- Learn more about node pools (/kubernetes-engine/docs/concepts/node-pools).
- Learn more about the cluster architecture (/kubernetes-engine/docs/concepts/cluster-architecture).