google_container_cluster

Note

Visit the Provision a GKE Cluster (Google Cloud) Learn tutorial to learn how to provision and interact with a GKE cluster.

Note

See the Using GKE with Terraform guide for more information about using GKE with Terraform.
Manages a Google Kubernetes Engine (GKE) cluster. For more information see the official documentation and the API reference.

Note

: On version 5.0.0+ of the provider, you must explicitly set `deletion_protection=false` (and run `terraform apply` to write the field to state) in order to destroy a cluster. It is recommended to not set this field (or set it to true) until you're ready to destroy.

Warning:

All arguments and attributes, including basic auth username and passwords as well as certificate outputs will be stored in the raw state as plaintext. Read more about sensitive data in state.

Example Usage - with a separately managed node pool (recommended)

```
resource "google_service_account" "default" {
  account_id   = "service-account-id"
  display_name = "Service Account"
}

resource "google_container_cluster" "primary" {
  name     = "my-gke-cluster"
  location = "us-central1"

  # We can't create a cluster with no node pool defined, but we want to only use
  # separately managed node pools. So we create the smallest possible default
  # node pool and immediately delete it.
  remove_default_node_pool = true
  initial_node_count       = 1
}

resource "google_container_node_pool" "primary_preemptible_nodes" {
  name       = "my-node-pool"
  location   = "us-central1"
  cluster    = google_container_cluster.primary.name
```

```
  node_count = 1

  node_config {
    preemptible  = true
    machine_type = "e2-medium"

    # Google recommends custom service accounts that have cloud-platform scope and
permissions granted via IAM Roles.
    service_account = google_service_account.default.email
    oauth_scopes    = [
      "https://www.googleapis.com/auth/cloud-platform"
    ]
  }
}
```

Note:

It is recommended that node pools be created and managed as separate resources as in the example above. This allows node pools to be added and removed without recreating the cluster. Node pools defined directly in the `google_container_cluster` resource cannot be removed without re-creating the cluster.

Example Usage - with the default node pool

```
resource "google_service_account" "default" {
  account_id   = "service-account-id"
  display_name = "Service Account"
}

resource "google_container_cluster" "primary" {
  name               = "marcellus-wallace"
  location           = "us-central1-a"
  initial_node_count = 3
  node_config {
    # Google recommends custom service accounts that have cloud-platform scope and
permissions granted via IAM Roles.
    service_account = google_service_account.default.email
    oauth_scopes = [
      "https://www.googleapis.com/auth/cloud-platform"
    ]
    labels = {
      foo = "bar"
    }
    tags = ["foo", "bar"]
  }
  timeouts {
    create = "30m"
    update = "40m"
  }
}
```

Argument Reference

- `name` - (Required) The name of the cluster, unique within the project and location.

- `location` - (Optional) The location (region or zone) in which the cluster master will be created, as well as the default node location. If you specify a zone (such as `us-central1-a`), the cluster will be a zonal cluster with a single cluster master. If you specify a region (such as `us-west1`), the cluster will be a regional cluster with multiple masters spread across zones in the region, and with default node locations in those zones as well
- `node_locations` - (Optional) The list of zones in which the cluster's nodes are located. Nodes must be in the region of their regional cluster or in the same region as their cluster's zone for zonal clusters. If this is specified for a zonal cluster, omit the cluster's zone.

Note

A "multi-zonal" cluster is a zonal cluster with at least one additional zone defined; in a multi-zonal cluster, the cluster master is only present in a single zone while nodes are present in each of the primary zone and the node locations. In contrast, in a regional cluster, cluster master nodes are present in multiple zones in the region. For that reason, regional clusters should be preferred.

- `deletion_protection` - (Optional) Whether or not to allow Terraform to destroy the cluster. Unless this field is set to false in Terraform state, a `terraform destroy` or `terraform apply` that would delete the cluster will fail.
- `addons_config` - (Optional) The configuration for addons supported by GKE. Structure is documented below.
- `allow_net_admin` - (Optional) Enable NET_ADMIN for the cluster. Defaults to `false`. This field should only be enabled for Autopilot clusters (`enable_autopilot` set to `true`).
- `cluster_ipv4_cidr` - (Optional) The IP address range of the Kubernetes pods in this cluster in CIDR notation (e.g. `10.96.0.0/14`). Leave blank to have one automatically chosen or specify a `/14` block in `10.0.0.0/8`. This field will default a new cluster to routes-based, where `ip_allocation_policy` is not defined.
- `cluster_autoscaling` - (Optional) Per-cluster configuration of Node Auto-Provisioning with Cluster Autoscaler to automatically adjust the size of the cluster and create/delete node pools based on the current needs of the cluster's

workload. See the guide to using Node Auto-Provisioning for more details. Structure is documented below.

- `binary_authorization` - (Optional) Configuration options for the Binary Authorization feature. Structure is documented below.
- `service_external_ips_config` - (Optional) Structure is documented below.
- `mesh_certificates` - (Optional) Structure is documented below.
- `database_encryption` - (Optional) Structure is documented below.
- `description` - (Optional) Description of the cluster.
- `default_max_pods_per_node` - (Optional) The default maximum number of pods per node in this cluster. This doesn't work on "routes-based" clusters, clusters that don't have IP Aliasing enabled. See the official documentation for more information.
- `enable_kubernetes_alpha` - (Optional) Whether to enable Kubernetes Alpha features for this cluster. Note that when this option is enabled, the cluster cannot be upgraded and will be automatically deleted after 30 days.
- `enable_k8s_beta_apis` - (Optional) Configuration for Kubernetes Beta APIs. Structure is documented below.
- `enable_tpu` - (Optional) Whether to enable Cloud TPU resources in this cluster. See the official documentation.
- `enable_legacy_abac` - (Optional) Whether the ABAC authorizer is enabled for this cluster. When enabled, identities in the system, including service accounts, nodes, and controllers, will have statically granted permissions beyond those provided by the RBAC configuration or IAM. Defaults to `false`
- `enable_shielded_nodes` - (Optional) Enable Shielded Nodes features on all nodes in this cluster. Defaults to `true`.
- `enable_autopilot` - (Optional) Enable Autopilot for this cluster. Defaults to `false`. Note that when this option is enabled, certain features of Standard GKE are not available. See the official documentation for available features.
- `initial_node_count` - (Optional) The number of nodes to create in this cluster's default node pool. In regional or multi-zonal clusters, this is the number of nodes per zone. Must be set if `node_pool` is not set. If you're using `google_container_node_pool` objects with no default node pool, you'll need to set this to a value of at least `1`, alongside setting `remove_default_node_pool` to `true`.

- `ip_allocation_policy` - (Optional) Configuration of cluster IP allocation for VPC-native clusters. If this block is unset during creation, it will be set by the GKE backend. Structure is documented below.
- `networking_mode` - (Optional) Determines whether alias IPs or routes will be used for pod IPs in the cluster. Options are `VPC_NATIVE` or `ROUTES`. `VPC_NATIVE` enables IP aliasing. Newly created clusters will default to `VPC_NATIVE`.
- `logging_config` - (Optional) Logging configuration for the cluster. Structure is documented below.
- `logging_service` - (Optional) The logging service that the cluster should write logs to. Available options include `logging.googleapis.com` (Legacy Stackdriver), `logging.googleapis.com/kubernetes` (Stackdriver Kubernetes Engine Logging), and `none`. Defaults to `logging.googleapis.com/kubernetes`
- `maintenance_policy` - (Optional) The maintenance policy to use for the cluster. Structure is documented below.
- `master_auth` - (Optional) The authentication information for accessing the Kubernetes master. Some values in this block are only returned by the API if your service account has permission to get credentials for your GKE cluster. If you see an unexpected diff unsetting your client cert, ensure you have the `container.clusters.getCredentials` permission. Structure is documented below.
- `master_authorized_networks_config` - (Optional) The desired configuration options for master authorized networks. Omit the nested `cidr_blocks` attribute to disallow external access (except the cluster node IPs, which GKE automatically whitelists). Structure is documented below.
- `min_master_version` - (Optional) The minimum version of the master. GKE will auto-update the master to new versions, so this does not guarantee the current master version--use the read-only `master_version` field to obtain that. If unset, the cluster's version will be set by GKE to the version of the most recent official release (which is not necessarily the latest version). Most users will find the `google_container_engine_versions` data source useful - it indicates which versions are available, and can be use to approximate fuzzy versions in a Terraform-compatible way. If you intend to specify versions manually, the docs describe the various acceptable formats for this field.

Note

If you are using the `google_container_engine_versions` datasource with a regional cluster, ensure that you have provided a `location` to the datasource. A region can have a different set of

supported versions than its corresponding zones, and not all zones in a region are guaranteed to support the same version.

- `monitoring_config` - (Optional) Monitoring configuration for the cluster. Structure is documented below.
- `monitoring_service` - (Optional) The monitoring service that the cluster should write metrics to. Automatically send metrics from pods in the cluster to the Google Cloud Monitoring API. VM metrics will be collected by Google Compute Engine regardless of this setting Available options include `monitoring.googleapis.com` (Legacy Stackdriver), `monitoring.googleapis.com/kubernetes` (Stackdriver Kubernetes Engine Monitoring), and `none`. Defaults to `monitoring.googleapis.com/kubernetes`
- `network` - (Optional) The name or self_link of the Google Compute Engine network to which the cluster is connected. For Shared VPC, set this to the self link of the shared network.
- `network_policy` - (Optional) Configuration options for the NetworkPolicy feature. Structure is documented below.
- `node_config` - (Optional) Parameters used in creating the default node pool. Generally, this field should not be used at the same time as a `google_container_node_pool` or a `node_pool` block; this configuration manages the default node pool, which isn't recommended to be used with Terraform. Structure is documented below.
- `node_pool` - (Optional) List of node pools associated with this cluster. See google_container_node_pool for schema. **Warning:** node pools defined inside a cluster can't be changed (or added/removed) after cluster creation without deleting and recreating the entire cluster. Unless you absolutely need the ability to say "these are the *only* node pools associated with this cluster", use the google_container_node_pool resource instead of this property.
- `node_pool_auto_config` - (Optional) Node pool configs that apply to auto-provisioned node pools in autopilot clusters and node auto-provisioning-enabled clusters. Structure is documented below.
- `node_pool_defaults` - (Optional) Default NodePool settings for the entire cluster. These settings are overridden if specified on the specific NodePool object. Structure is documented below.
- `node_version` - (Optional) The Kubernetes version on the nodes. Must either be unset or set to the same value as `min_master_version` on create. Defaults to the default version set by GKE which is not necessarily the latest version. This only affects nodes in the default node pool. While a fuzzy version can be specified, it's

recommended that you specify explicit versions as Terraform will see spurious diffs when fuzzy versions are used. See the `google_container_engine_versions` data source's `version_prefix` field to approximate fuzzy versions in a Terraform-compatible way. To update nodes in other node pools, use the `version` attribute on the node pool.

- `notification_config` - (Optional) Configuration for the cluster upgrade notifications feature. Structure is documented below.
- `confidential_nodes` - Configuration for Confidential Nodes feature. Structure is documented below documented below.
- `pod_security_policy_config` - (Optional, Beta) Configuration for the PodSecurityPolicy feature. Structure is documented below.
- `authenticator_groups_config` - (Optional) Configuration for the Google Groups for GKE feature. Structure is documented below.
- `private_cluster_config` - (Optional) Configuration for private clusters, clusters with private nodes. Structure is documented below.
- `cluster_telemetry` - (Optional, Beta) Configuration for ClusterTelemetry feature, Structure is documented below.
- `project` - (Optional) The ID of the project in which the resource belongs. If it is not provided, the provider project is used.
- `release_channel` - (Optional) Configuration options for the Release channel feature, which provide more control over automatic upgrades of your GKE clusters. When updating this field, GKE imposes specific version requirements. See Selecting a new release channel for more details; the `google_container_engine_versions` datasource can provide the default version for a channel. Note that removing the `release_channel` field from your config will cause Terraform to stop managing your cluster's release channel, but will not unenroll it. Instead, use the `"UNSPECIFIED"` channel. Structure is documented below.
- `remove_default_node_pool` - (Optional) If `true`, deletes the default node pool upon cluster creation. If you're using `google_container_node_pool` resources with no default node pool, this should be set to `true`, alongside setting `initial_node_count` to at least `1`.
- `resource_labels` - (Optional) The GCE resource labels (a map of key/value pairs) to be applied to the cluster.
- `cost_management_config` - (Optional) Configuration for the Cost Allocation feature. Structure is documented below.

- `resource_usage_export_config` - (Optional) Configuration for the ResourceUsageExportConfig feature. Structure is documented below.
- `subnetwork` - (Optional) The name or self_link of the Google Compute Engine subnetwork in which the cluster's instances are launched.
- `vertical_pod_autoscaling` - (Optional) Vertical Pod Autoscaling automatically adjusts the resources of pods controlled by it. Structure is documented below.
- `workload_identity_config` - (Optional) Workload Identity allows Kubernetes service accounts to act as a user-managed Google IAM Service Account. Structure is documented below.
- `enable_intranode_visibility` - (Optional) Whether Intra-node visibility is enabled for this cluster. This makes same node pod to pod traffic visible for VPC network.
- `enable_l4_ilb_subsetting` - (Optional, Beta) Whether L4ILB Subsetting is enabled for this cluster.
- `enable_multi_networking` - (Optional, Beta) Whether multi-networking is enabled for this cluster.
- `enable_fqdn_network_policy` - (Optional, Beta) Whether FQDN Network Policy is enabled on this cluster. Users who enable this feature for existing Standard clusters must restart the GKE Dataplane V2 `anetd` DaemonSet after enabling it. See the Enable FQDN Network Policy in an existing cluster for more information.
- `private_ipv6_google_access` - (Optional) The desired state of IPv6 connectivity to Google Services. By default, no private IPv6 access to or from Google Services (all access will be via IPv4).
- `datapath_provider` - (Optional) The desired datapath provider for this cluster. This is set to `LEGACY_DATAPATH` by default, which uses the IPTables-based kube-proxy implementation. Set to `ADVANCED_DATAPATH` to enable Dataplane v2.
- `default_snat_status` - (Optional) GKE SNAT DefaultSnatStatus contains the desired state of whether default sNAT should be disabled on the cluster, API doc. Structure is documented below
- `dns_config` - (Optional) Configuration for Using Cloud DNS for GKE. Structure is documented below.
- `gateway_api_config` - (Optional) Configuration for GKE Gateway API controller. Structure is documented below.
- `protect_config` - (Optional, Beta) Enable/Disable Protect API features for the cluster. Structure is documented below.
- `security_posture_config` - (Optional) Enable/Disable Security Posture API features for the cluster. Structure is documented below.

- `fleet` - (Optional) Fleet configuration for the cluster. Structure is documented below.

The `default_snat_status` block supports

- `disabled` - (Required) Whether the cluster disables default in-node sNAT rules. In-node sNAT rules will be disabled when defaultSnatStatus is disabled.When disabled is set to false, default IP masquerade rules will be applied to the nodes to prevent sNAT on cluster internal traffic

The `cluster_telemetry` block supports

- `type` - Telemetry integration for the cluster. Supported values (`ENABLED, DISABLED, SYSTEM_ONLY`); `SYSTEM_ONLY` (Only system components are monitored and logged) is only available in GKE versions 1.15 and later.

The `addons_config` block supports:

- `horizontal_pod_autoscaling` - (Optional) The status of the Horizontal Pod Autoscaling addon, which increases or decreases the number of replica pods a replication controller has based on the resource usage of the existing pods. It is enabled by default; set `disabled = true` to disable.
- `http_load_balancing` - (Optional) The status of the HTTP (L7) load balancing controller addon, which makes it easy to set up HTTP load balancers for services in a cluster. It is enabled by default; set `disabled = true` to disable.
- `network_policy_config` - (Optional) Whether we should enable the network policy addon for the master. This must be enabled in order to enable network policy for the nodes. To enable this, you must also define a `network_policy` block, otherwise nothing will happen. It can only be disabled if the nodes already do not have network policies enabled. Defaults to disabled; set `disabled = false` to enable.
- `gcp_filestore_csi_driver_config` - (Optional) The status of the Filestore CSI driver addon, which allows the usage of filestore instance as volumes. It is disabled by default; set `enabled = true` to enable.
- `gcs_fuse_csi_driver_config` - (Optional) The status of the GCSFuse CSI driver addon, which allows the usage of a gcs bucket as volumes. It is disabled by default for Standard clusters; set `enabled = true` to enable. It is enabled by default for Autopilot clusters with version 1.24 or later; set `enabled = true` to enable it explicitly. See Enable the Cloud Storage FUSE CSI driver for more information.
- `cloudrun_config` - (Optional). Structure is documented below.
- `istio_config` - (Optional, Beta). Structure is documented below.
- `identity_service_config` - (Optional). Structure is documented below.

- `dns_cache_config` - (Optional). The status of the NodeLocal DNSCache addon. It is disabled by default. Set `enabled = true` to enable.

  Enabling/Disabling NodeLocal DNSCache in an existing cluster is a disruptive operation. All cluster nodes running GKE 1.15 and higher are recreated.
- `gce_persistent_disk_csi_driver_config` - (Optional). Whether this cluster should enable the Google Compute Engine Persistent Disk Container Storage Interface (CSI) Driver. Set `enabled = true` to enable.

  Note: The Compute Engine persistent disk CSI Driver is enabled by default on newly created clusters for the following versions: Linux clusters: GKE version 1.18.10-gke.2100 or later, or 1.19.3-gke.2100 or later.
- `gke_backup_agent_config` - (Optional). The status of the Backup for GKE agent addon. It is disabled by default; Set `enabled = true` to enable.
- `kalm_config` - (Optional, Beta). Configuration for the KALM addon, which manages the lifecycle of k8s. It is disabled by default; Set `enabled = true` to enable.
- `config_connector_config` - (Optional). The status of the ConfigConnector addon. It is disabled by default; Set `enabled = true` to enable.

This example `addons_config` disables two addons:

```
addons_config {
  http_load_balancing {
    disabled = true
  }

  horizontal_pod_autoscaling {
    disabled = true
  }
}
```

The `binary_authorization` block supports:

- `enabled` - (DEPRECATED) Enable Binary Authorization for this cluster. Deprecated in favor of `evaluation_mode`.
- `evaluation_mode` - (Optional) Mode of operation for Binary Authorization policy evaluation. Valid values are `DISABLED` and `PROJECT_SINGLETON_POLICY_ENFORCE`.

The `service_external_ips_config` block supports:

- `enabled` - (Required) Controls whether external ips specified by a service will be allowed. It is enabled by default.

The `mesh_certificates` block supports:

- `enable_certificates` - (Required) Controls the issuance of workload mTLS certificates. It is enabled by default. Workload Identity is required, see workload_config.

The `database_encryption` block supports:

- `state` - (Required) `ENCRYPTED` or `DECRYPTED`
- `key_name` - (Required) the key to use to encrypt/decrypt secrets. See the DatabaseEncryption definition for more information.

The `enable_k8s_beta_apis` block supports:

- `enabled_apis` - (Required) Enabled Kubernetes Beta APIs. To list a Beta API resource, use the representation {group}/{version}/{resource}. The version must be a Beta version. Note that you cannot disable beta APIs that are already enabled on a cluster without recreating it. See the Configure beta APIs for more information.

The `cloudrun_config` block supports:

- `disabled` - (Optional) The status of the CloudRun addon. It is disabled by default. Set `disabled=false` to enable.
- `load_balancer_type` - (Optional) The load balancer type of CloudRun ingress service. It is external load balancer by default. Set `load_balancer_type=LOAD_BALANCER_TYPE_INTERNAL` to configure it as internal load balancer.

The `identity_service_config` block supports:

- `enabled` - (Optional) Whether to enable the Identity Service component. It is disabled by default. Set `enabled=true` to enable.

The `istio_config` block supports:

- `disabled` - (Optional) The status of the Istio addon, which makes it easy to set up Istio for services in a cluster. It is disabled by default. Set `disabled = false` to enable.
- `auth` - (Optional) The authentication type between services in Istio. Available options include `AUTH_MUTUAL_TLS`.

The `cluster_autoscaling` block supports:

- `enabled` - (Optional) Whether node auto-provisioning is enabled. Must be supplied for GKE Standard clusters, `true` is implied for autopilot clusters. Resource limits for `cpu` and `memory` must be defined to enable node auto-provisioning for GKE Standard.

- `resource_limits` - (Optional) Global constraints for machine resources in the cluster. Configuring the `cpu` and `memory` types is required if node auto-provisioning is enabled. These limits will apply to node pool autoscaling in addition to node auto-provisioning. Structure is documented below.
- `auto_provisioning_defaults` - (Optional) Contains defaults for a node pool created by NAP. A subset of fields also apply to GKE Autopilot clusters. Structure is documented below.
- `autoscaling_profile` - (Optional, Beta) Configuration options for the Autoscaling profile feature, which lets you choose whether the cluster autoscaler should optimize for resource utilization or resource availability when deciding to remove nodes from a cluster. Can be `BALANCED` or `OPTIMIZE_UTILIZATION`. Defaults to `BALANCED`.

The `resource_limits` block supports:

- `resource_type` - (Required) The type of the resource. For example, `cpu` and `memory`. See the guide to using Node Auto-Provisioning for a list of types.
- `minimum` - (Optional) Minimum amount of the resource in the cluster.
- `maximum` - (Optional) Maximum amount of the resource in the cluster.

The `auto_provisioning_defaults` block supports:

- `min_cpu_platform` - (Optional, Beta) Minimum CPU platform to be used for NAP created node pools. The instance may be scheduled on the specified or newer CPU platform. Applicable values are the friendly names of CPU platforms, such as "Intel Haswell" or "Intel Sandy Bridge".
- `oauth_scopes` - (Optional) Scopes that are used by NAP and GKE Autopilot when creating node pools. Use the "https://www.googleapis.com/auth/cloud-platform" scope to grant access to all APIs. It is recommended that you set `service_account` to a non-default service account and grant IAM roles to that service account for only the resources that it needs.

Note

`monitoring.write` is always enabled regardless of user input. `monitoring` and `logging.write` may also be enabled depending on the values for `monitoring_service` and `logging_service`.

- `service_account` - (Optional) The Google Cloud Platform Service Account to be used by the node VMs created by GKE Autopilot or NAP.
- `boot_disk_kms_key` - (Optional) The Customer Managed Encryption Key used to encrypt the boot disk attached to each node in the node pool. This should be of the form

projects/[KEY_PROJECT_ID]/locations/[LOCATION]/keyRings/[RING_NAME]/cryptoKeys/[KEY_NAME]. For more information about protecting resources with Cloud KMS Keys please see: https://cloud.google.com/compute/docs/disks/customer-managed-encryption

- `disk_size` - (Optional) Size of the disk attached to each node, specified in GB. The smallest allowed disk size is 10GB. Defaults to `100`
- `disk_type` - (Optional) Type of the disk attached to each node (e.g. 'pd-standard', 'pd-ssd' or 'pd-balanced'). Defaults to `pd-standard`
- `image_type` - (Optional) The default image type used by NAP once a new node pool is being created. Please note that according to the official documentation the value must be one of the [COS_CONTAINERD, COS, UBUNTU_CONTAINERD, UBUNTU]. **NOTE** : COS AND UBUNTU are deprecated as of `GKE 1.24`
- `shielded_instance_config` - (Optional) Shielded Instance options. Structure is documented below.
- `management` - (Optional) NodeManagement configuration for this NodePool. Structure is documented below.

The `management` block supports:

- `auto_upgrade` - (Optional) Specifies whether node auto-upgrade is enabled for the node pool. If enabled, node auto-upgrade helps keep the nodes in your node pool up to date with the latest release version of Kubernetes.
- `auto_repair` - (Optional) Specifies whether the node auto-repair is enabled for the node pool. If enabled, the nodes in this node pool will be monitored and, if they fail health checks too many times, an automatic repair action will be triggered.

This block also contains several computed attributes, documented below.

- `upgrade_settings` - (Optional) Specifies the upgrade settings for NAP created node pools. Structure is documented below.

The `upgrade_settings` block supports:

- `strategy` - (Optional) Strategy used for node pool update. Strategy can only be one of BLUE_GREEN or SURGE. The default is value is SURGE.
- `max_surge` - (Optional) The maximum number of nodes that can be created beyond the current size of the node pool during the upgrade process. To be used when strategy is set to SURGE. Default is 0.
- `max_unavailable` - (Optional) The maximum number of nodes that can be simultaneously unavailable during the upgrade process. To be used when strategy is set to SURGE. Default is 0.

- `blue_green_settings` - (Optional) Settings for blue-green upgrade strategy. To be specified when strategy is set to BLUE_GREEN. Structure is documented below.

The `blue_green_settings` block supports:

- `node_pool_soak_duration` - (Optional) Time needed after draining entire blue pool. After this period, blue pool will be cleaned up. A duration in seconds with up to nine fractional digits, ending with 's'. Example: "3.5s".
- `standard_rollout_policy`: (Optional) Standard policy for the blue-green upgrade. To be specified when strategy is set to BLUE_GREEN. Structure is documented below.

The `standard_rollout_policy` block supports:

- `batch_percentage`: (Optional) Percentage of the bool pool nodes to drain in a batch. The range of this field should be (0.0, 1.0). Only one of the batch_percentage or batch_node_count can be specified.
- `batch_node_count` - (Optional) Number of blue nodes to drain in a batch. Only one of the batch_percentage or batch_node_count can be specified.
- `batch_soak_duration` - (Optional) Soak time after each batch gets drained. A duration in seconds with up to nine fractional digits, ending with 's'. Example: "3.5s".`.

The `authenticator_groups_config` block supports:

- `security_group` - (Required) The name of the RBAC security group for use with Google security groups in Kubernetes RBAC. Group name must be in format `gke-security-groups@yourdomain.com`.

The `logging_config` block supports:

- `enable_components` - (Required) The GKE components exposing logs. Supported values include: `SYSTEM_COMPONENTS`, `APISERVER`, `CONTROLLER_MANAGER`, `SCHEDULER`, and `WORKLOADS`.

The `monitoring_config` block supports:

- `enable_components` - (Optional) The GKE components exposing metrics. Supported values include: `SYSTEM_COMPONENTS`, `APISERVER`, `SCHEDULER`, `CONTROLLER_MANAGER`, `STORAGE`, `HPA`, `POD`, `DAEMONSET`, `DEPLOYMENT` and `STATEFULSET`. In beta provider, `WORKLOADS` is supported on top of those 10 values. (`WORKLOADS` is deprecated and removed in GKE 1.24.)
- `managed_prometheus` - (Optional) Configuration for Managed Service for Prometheus. Structure is documented below.

- `advanced_datapath_observability_config` - (Optional) Configuration for Advanced Datapath Monitoring. Structure is documented below.

The `managed_prometheus` block supports:

- `enabled` - (Required) Whether or not the managed collection is enabled.

The `advanced_datapath_observability_config` block supports:

- `enable_metrics` - (Required) Whether or not to enable advanced datapath metrics.
- `relay_mode` - (Optional) Mode used to make Relay available.

The `maintenance_policy` block supports:

- `daily_maintenance_window` - (Optional) structure documented below.
- `recurring_window` - (Optional) structure documented below
- `maintenance_exclusion` - (Optional) structure documented below

In beta, one or the other of `recurring_window` and `daily_maintenance_window` is required if a `maintenance_policy` block is supplied.

- `daily_maintenance_window` - Time window specified for daily maintenance operations. Specify `start_time` in RFC3339 format "HH:MM", where HH : [00-23] and MM : [00-59] GMT. For example:

Examples:

```
maintenance_policy {
  daily_maintenance_window {
    start_time = "03:00"
  }
}
```

- `recurring_window` - Time window for recurring maintenance operations.

Specify `start_time` and `end_time` in RFC3339 "Zulu" date format. The start time's date is the initial date that the window starts, and the end time is used for calculating duration. Specify `recurrence` in RFC5545 RRULE format, to specify when this recurs. Note that GKE may accept other formats, but will return values in UTC, causing a permanent diff.

Examples:

```
maintenance_policy {
  recurring_window {
    start_time = "2019-08-01T02:00:00Z"
    end_time = "2019-08-01T06:00:00Z"
    recurrence = "FREQ=DAILY"
  }
}
maintenance_policy {
  recurring_window {
    start_time = "2019-01-01T09:00:00Z"
    end_time = "2019-01-01T17:00:00Z"
    recurrence = "FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR"
  }
}
```

- `maintenance_exclusion` - Exceptions to maintenance window. Non-emergency maintenance should not occur in these windows. A cluster can have up to 20 maintenance exclusions at a time Maintenance Window and Exclusions

The `maintenance_exclusion` block supports:

- `exclusion_options` - (Optional) MaintenanceExclusionOptions provides maintenance exclusion related options.

The `exclusion_options` block supports:

- `scope` - (Required) The scope of automatic upgrades to restrict in the exclusion window. One of: **NO_UPGRADES | NO_MINOR_UPGRADES | NO_MINOR_OR_NODE_UPGRADES**

Specify `start_time` and `end_time` in RFC3339 "Zulu" date format. The start time's date is the initial date that the window starts, and the end time is used for calculating duration.Specify `recurrence` in RFC5545 RRULE format, to specify when this recurs. Note that GKE may accept other formats, but will return values in UTC, causing a permanent diff.

Examples:

```
maintenance_policy {
  recurring_window {
    start_time = "2019-01-01T00:00:00Z"
    end_time = "2019-01-02T00:00:00Z"
    recurrence = "FREQ=DAILY"
  }
  maintenance_exclusion{
    exclusion_name = "batch job"
    start_time = "2019-01-01T00:00:00Z"
    end_time = "2019-01-02T00:00:00Z"
    exclusion_options {
      scope = "NO_UPGRADES"
    }
  }
  maintenance_exclusion{
    exclusion_name = "holiday data load"
    start_time = "2019-05-01T00:00:00Z"
    end_time = "2019-05-02T00:00:00Z"
    exclusion_options {
      scope = "NO_MINOR_UPGRADES"
    }
  }
}
```

The `ip_allocation_policy` block supports:

- `cluster_secondary_range_name` - (Optional) The name of the existing secondary range in the cluster's subnetwork to use for pod IP addresses. Alternatively, `cluster_ipv4_cidr_block` can be used to automatically create a GKE-managed one.
- `services_secondary_range_name` - (Optional) The name of the existing secondary range in the cluster's subnetwork to use for service `ClusterIP`s.

Alternatively, `services_ipv4_cidr_block` can be used to automatically create a GKE-managed one.

- `cluster_ipv4_cidr_block` - (Optional) The IP address range for the cluster pod IPs. Set to blank to have a range chosen with the default size. Set to /netmask (e.g. /14) to have a range chosen with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use.
- `services_ipv4_cidr_block` - (Optional) The IP address range of the services IPs in this cluster. Set to blank to have a range chosen with the default size. Set to /netmask (e.g. /14) to have a range chosen with a specific netmask. Set to a CIDR notation (e.g. 10.96.0.0/14) from the RFC-1918 private networks (e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16) to pick a specific range to use.
- `stack_type` - (Optional) The IP Stack Type of the cluster. Default value is `IPV4`. Possible values are `IPV4` and `IPV4_IPV6`.
- `additional_pod_ranges_config` - (Optional) The configuration for additional pod secondary ranges at the cluster level. Used for Autopilot clusters and Standard clusters with which control of the secondary Pod IP address assignment to node pools isn't needed. Structure is documented below.

The `additional_pod_ranges_config` block supports:

- `pod_range_names` - (Required) The names of the Pod ranges to add to the cluster.

The `master_auth` block supports:

- `client_certificate_config` - (Required) Whether client certificate authorization is enabled for this cluster. For example:

```
master_auth {
  client_certificate_config {
    issue_client_certificate = false
  }
}
```

This block also contains several computed attributes, documented below.

The `master_authorized_networks_config` block supports:

- `cidr_blocks` - (Optional) External networks that can access the Kubernetes cluster master through HTTPS.
- `gcp_public_cidrs_access_enabled` - (Optional) Whether Kubernetes master is accessible via Google Compute Engine Public IPs.

The `master_authorized_networks_config.cidr_blocks` block supports:

- `cidr_block` - (Optional) External network that can access Kubernetes master through HTTPS. Must be specified in CIDR notation.
- `display_name` - (Optional) Field for users to identify CIDR blocks.

The `network_policy` block supports:

- `provider` - (Optional) The selected network policy provider. Defaults to PROVIDER_UNSPECIFIED.
- `enabled` - (Required) Whether network policy is enabled on the cluster.

The `node_config` block supports:

- `disk_size_gb` - (Optional) Size of the disk attached to each node, specified in GB. The smallest allowed disk size is 10GB. Defaults to 100GB.
- `disk_type` - (Optional) Type of the disk attached to each node (e.g. 'pd-standard', 'pd-balanced' or 'pd-ssd'). If unspecified, the default disk type is 'pd-standard'
- `enable_confidential_storage` - (Optional, Beta) Enabling Confidential Storage will create boot disk with confidential mode. It is disabled by default.
- `ephemeral_storage_config` - (Optional, Beta) Parameters for the ephemeral storage filesystem. If unspecified, ephemeral storage is backed by the boot disk. Structure is documented below.

```
ephemeral_storage_config {
  local_ssd_count = 2
}
```

- `ephemeral_storage_local_ssd_config` - (Optional) Parameters for the ephemeral storage filesystem. If unspecified, ephemeral storage is backed by the boot disk. Structure is documented below.

```
ephemeral_storage_local_ssd_config {
  local_ssd_count = 2
}
```

- `fast_socket` - (Optional) Parameters for the NCCL Fast Socket feature. If unspecified, NCCL Fast Socket will not be enabled on the node pool. Node Pool must enable gvnic. GKE version 1.25.2-gke.1700 or later. Structure is documented below.
- `local_nvme_ssd_block_config` - (Optional) Parameters for the local NVMe SSDs. Structure is documented below.
- `logging_variant` (Optional) Parameter for specifying the type of logging agent used in a node pool. This will override any cluster-wide default value. Valid values include DEFAULT and MAX_THROUGHPUT. See Increasing logging agent throughput for more information.
- `gcfs_config` - (Optional) Parameters for the Google Container Filesystem (GCFS). If unspecified, GCFS will not be enabled on the node pool. When enabling this feature you must specify `image_type = "COS_CONTAINERD"` and `node_version` from GKE versions 1.19 or later to use it. For GKE versions 1.19, 1.20, and 1.21, the

recommended minimum `node_version` would be 1.19.15-gke.1300, 1.20.11-gke.1300, and 1.21.5-gke.1300 respectively. A `machine_type` that has more than 16 GiB of memory is also recommended. GCFS must be enabled in order to use image streaming. Structure is documented below.

```
gcfs_config {
  enabled = true
}
```

- `gvnic` - (Optional) Google Virtual NIC (gVNIC) is a virtual network interface. Installing the gVNIC driver allows for more efficient traffic transmission across the Google network infrastructure. gVNIC is an alternative to the virtIO-based ethernet driver. GKE nodes must use a Container-Optimized OS node image. GKE node version 1.15.11-gke.15 or later Structure is documented below.

```
gvnic {
  enabled = true
}
```

- `guest_accelerator` - (Optional) List of the type and count of accelerator cards attached to the instance. Structure documented below. To support removal of guest_accelerators in Terraform 0.12 this field is an Attribute as Block
- `image_type` - (Optional) The image type to use for this node. Note that changing the image type will delete and recreate all nodes in the node pool.
- `labels` - (Optional) The Kubernetes labels (key/value pairs) to be applied to each node. The kubernetes.io/ and k8s.io/ prefixes are reserved by Kubernetes Core components and cannot be specified.
- `resource_labels` - (Optional) The GCP labels (key/value pairs) to be applied to each node. Refer here for how these labels are applied to clusters, node pools and nodes.
- `local_ssd_count` - (Optional) The amount of local SSD disks that will be attached to each cluster node. Defaults to 0.
- `machine_type` - (Optional) The name of a Google Compute Engine machine type. Defaults to `e2-medium`. To create a custom machine type, value should be set as specified here.
- `metadata` - (Optional) The metadata key/value pairs assigned to instances in the cluster. From GKE `1.12` onwards, `disable-legacy-endpoints` is set to `true` by the API; if `metadata` is set but that default value is not included, Terraform will attempt to unset the value. To avoid this, set the value in your config.
- `min_cpu_platform` - (Optional) Minimum CPU platform to be used by this instance. The instance may be scheduled on the specified or newer CPU platform.

Applicable values are the friendly names of CPU platforms, such as `Intel Haswell`. See the official documentation for more information.

- `oauth_scopes` - (Optional) The set of Google API scopes to be made available on all of the node VMs under the "default" service account. Use the "https://www.googleapis.com/auth/cloud-platform" scope to grant access to all APIs. It is recommended that you set `service_account` to a non-default service account and grant IAM roles to that service account for only the resources that it needs.
  See the official documentation for information on migrating off of legacy access scopes.
- `preemptible` - (Optional) A boolean that represents whether or not the underlying node VMs are preemptible. See the official documentation for more information. Defaults to false.
- `reservation_affinity` (Optional) The configuration of the desired reservation which instances could take capacity from. Structure is documented below.
- `spot` - (Optional) A boolean that represents whether the underlying node VMs are spot. See the official documentation for more information. Defaults to false.
- `sandbox_config` - (Optional, Beta) GKE Sandbox configuration. When enabling this feature you must specify `image_type = "COS_CONTAINERD"` and `node_version = "1.12.7-gke.17"` or later to use it. Structure is documented below.
- `boot_disk_kms_key` - (Optional) The Customer Managed Encryption Key used to encrypt the boot disk attached to each node in the node pool. This should be of the form projects/[KEY_PROJECT_ID]/locations/[LOCATION]/keyRings/[RING_NAME]/cryptoKeys/[KEY_NAME]. For more information about protecting resources with Cloud KMS Keys please see: https://cloud.google.com/compute/docs/disks/customer-managed-encryption
- `service_account` - (Optional) The service account to be used by the Node VMs. If not specified, the "default" service account is used.
- `shielded_instance_config` - (Optional) Shielded Instance options. Structure is documented below.
- `tags` - (Optional) The list of instance tags applied to all nodes. Tags are used to identify valid sources or targets for network firewalls.
- `taint` - (Optional) A list of Kubernetes taints to apply to nodes. This field will only report drift on taint keys that are already managed with Terraform, use `effective_taints` to view the list of GKE-managed taints on the node pool from all sources. Importing this resource will not record any taints as being

Terraform-managed, and will cause drift with any configured taints. Structure is documented below.

- `workload_metadata_config` - (Optional) Metadata configuration to expose to workloads on the node pool. Structure is documented below.
- `kubelet_config` - (Optional) Kubelet configuration, currently supported attributes can be found here. Structure is documented below.

```
kubelet_config {
  cpu_manager_policy   = "static"
  cpu_cfs_quota        = true
  cpu_cfs_quota_period = "100us"
  pod_pids_limit       = 1024
}
```

- `linux_node_config` - (Optional) Parameters that can be configured on Linux nodes. Structure is documented below.
- `node_group` - (Optional) Setting this field will assign instances of this pool to run on the specified node group. This is useful for running workloads on sole tenant nodes.
- `sole_tenant_config` (Optional) Allows specifying multiple node affinities useful for running workloads on sole tenant nodes. `node_affinity` structure is documented below.

```
sole_tenant_config {
  node_affinity {
    key = "compute.googleapis.com/node-group-name"
    operator = "IN"
    values = ["node-group-name"]
  }
}
```

- `advanced_machine_features` - (Optional) Specifies options for controlling advanced machine features. Structure is documented below.

The `node_affinity` block supports:

- `key` (Required) - The default or custom node affinity label key name.
- `operator` (Required) - Specifies affinity or anti-affinity. Accepted values are `"IN"` or `"NOT_IN"`
- `values` (Required) - List of node affinity label values as strings.

The `advanced_machine_features` block supports:

- `threads_per_core` - (Required) The number of threads per physical core. To disable simultaneous multithreading (SMT) set this to 1. If unset, the maximum number of threads supported per core by the underlying processor is assumed.

The `ephemeral_storage_config` block supports:

- `local_ssd_count` (Required) - Number of local SSDs to use to back ephemeral storage. Uses NVMe interfaces. Each local SSD is 375 GB in size. If zero, it means to disable using local SSDs as ephemeral storage.

The `ephemeral_storage_local_ssd_config` block supports:

- `local_ssd_count` (Required) - Number of local SSDs to use to back ephemeral storage. Uses NVMe interfaces. Each local SSD is 375 GB in size. If zero, it means to disable using local SSDs as ephemeral storage.

The `fast_socket` block supports:

- `enabled` (Required) - Whether or not the NCCL Fast Socket is enabled

The `local_nvme_ssd_block_config` block supports:

- `local_ssd_count` (Required) - Number of raw-block local NVMe SSD disks to be attached to the node. Each local SSD is 375 GB in size. If zero, it means no raw-block local NVMe SSD disks to be attached to the node. -> Note: Local NVMe SSD storage available in GKE versions v1.25.3-gke.1800 and later.

The `gcfs_config` block supports:

- `enabled` (Required) - Whether or not the Google Container Filesystem (GCFS) is enabled

The `gvnic` block supports:

- `enabled` (Required) - Whether or not the Google Virtual NIC (gVNIC) is enabled

The `guest_accelerator` block supports:

- `type` (Required) - The accelerator type resource to expose to this instance. E.g. `nvidia-tesla-k80`.
- `count` (Required) - The number of the guest accelerator cards exposed to this instance.
- `gpu_driver_installation_config` (Optional) - Configuration for auto installation of GPU driver. Structure is documented below.
- `gpu_partition_size` (Optional) - Size of partitions to create on the GPU. Valid values are described in the NVIDIA mig user guide.
- `gpu_sharing_config` (Optional) - Configuration for GPU sharing. Structure is documented below.

The `gpu_driver_installation_config` block supports:

- `gpu_driver_version` (Required) - Mode for how the GPU driver is installed. Accepted values are:
  - `"GPU_DRIVER_VERSION_UNSPECIFIED"`: Default value is to not install any GPU driver.

    - `"INSTALLATION_DISABLED"`: Disable GPU driver auto installation and needs manual installation.
    - `"DEFAULT"`: "Default" GPU driver in COS and Ubuntu.
    - `"LATEST"`: "Latest" GPU driver in COS.

The `gpu_sharing_config` block supports:

- `gpu_sharing_strategy` (Required) - The type of GPU sharing strategy to enable on the GPU node. Accepted values are:
    - `"TIME_SHARING"`: Allow multiple containers to have time-shared access to a single GPU device.
- `max_shared_clients_per_gpu` (Required) - The maximum number of containers that can share a GPU.

The `workload_identity_config` block supports:

- `workload_pool` (Optional) - The workload pool to attach all Kubernetes service accounts to.

```
workload_identity_config {
  workload_pool = "${data.google_project.project.project_id}.svc.id.goog"
}
```

The `node_pool_auto_config` block supports:

- `network_tags` (Optional) - The network tag config for the cluster's automatically provisioned node pools.

The `network_tags` block supports:

- `tags` (Optional) - List of network tags applied to auto-provisioned node pools.

```
node_pool_auto_config {
  network_tags {
    tags = ["foo", "bar"]
  }
}
```

The `node_pool_defaults` block supports:

- `node_config_defaults` (Optional) - Subset of NodeConfig message that has defaults.

The `node_config_defaults` block supports:

- `logging_variant` (Optional) The type of logging agent that is deployed by default for newly created node pools in the cluster. Valid values include DEFAULT and MAX_THROUGHPUT. See Increasing logging agent throughput for more information.

- `gcfs_config` (Optional, Beta) The default Google Container Filesystem (GCFS) configuration at the cluster level. e.g. enable image streaming across all the node pools within the cluster. Structure is documented below.

The `notification_config` block supports:

- `pubsub` (Required) - The pubsub config for the cluster's upgrade notifications.

The `pubsub` block supports:

- `enabled` (Required) - Whether or not the notification config is enabled
- `topic` (Optional) - The pubsub topic to push upgrade notifications to. Must be in the same project as the cluster. Must be in the format: `projects/{project}/topics/{topic}`.
- `filter` (Optional) - Choose what type of notifications you want to receive. If no filters are applied, you'll receive all notification types. Structure is documented below.

```
notification_config {
  pubsub {
    enabled = true
    topic = google_pubsub_topic.notifications.id
  }
}
```

The `filter` block supports:

- `event_type` (Optional) - Can be used to filter what notifications are sent. Accepted values are `UPGRADE_AVAILABLE_EVENT`, `UPGRADE_EVENT` and `SECURITY_BULLETIN_EVENT`. See Filtering notifications for more details.

The `confidential_nodes` block supports:

- `enabled` (Required) - Enable Confidential GKE Nodes for this cluster, to enforce encryption of data in-use.

The `pod_security_policy_config` block supports:

- `enabled` (Required) - Enable the PodSecurityPolicy controller for this cluster. If enabled, pods must be valid under a PodSecurityPolicy to be created.

The `private_cluster_config` block supports:

- `enable_private_nodes` (Optional) - Enables the private cluster feature, creating a private endpoint on the cluster. In a private cluster, nodes only have RFC 1918 private addresses and communicate with the master's private endpoint via private networking.
- `enable_private_endpoint` (Optional) - When `true`, the cluster's private endpoint is used as the cluster endpoint and access through the public endpoint is disabled.

When `false`, either endpoint can be used. This field only applies to private clusters, when `enable_private_nodes` is `true`.

- `master_ipv4_cidr_block` (Optional) - The IP range in CIDR notation to use for the hosted master network. This range will be used for assigning private IP addresses to the cluster master(s) and the ILB VIP. This range must not overlap with any other ranges in use within the cluster's network, and it must be a /28 subnet. See Private Cluster Limitations for more details. This field only applies to private clusters, when `enable_private_nodes` is `true`.
- `master_global_access_config` (Optional) - Controls cluster master global access settings. If unset, Terraform will no longer manage this field and will not modify the previously-set value. Structure is documented below.

In addition, the `private_cluster_config` allows access to the following read-only fields:

- `peering_name` - The name of the peering between this cluster and the Google owned VPC.
- `private_endpoint` - The internal IP address of this cluster's master endpoint.
- `private_endpoint_subnetwork` - Subnetwork in cluster's network where master's endpoint will be provisioned.
- `public_endpoint` - The external IP address of this cluster's master endpoint.

Warning

The Google provider is unable to validate certain configurations of `private_cluster_config` when `enable_private_nodes` is `false`. It's recommended that you omit the block entirely if the field is not set to `true`.

The `private_cluster_config.master_global_access_config` block supports:

- `enabled` (Optional) - Whether the cluster master is accessible globally or not.

The `reservation_affinity` block supports:

- `consume_reservation_type` (Required) The type of reservation consumption Accepted values are:
  - `"UNSPECIFIED"`: Default value. This should not be used.
  - `"NO_RESERVATION"`: Do not consume from any reserved capacity.
  - `"ANY_RESERVATION"`: Consume any reservation available.
  - `"SPECIFIC_RESERVATION"`: Must consume from a specific reservation. Must specify key value fields for specifying the reservations.
- `key` (Optional) The label key of a reservation resource. To target a SPECIFIC_RESERVATION by name, specify "compute.googleapis.com/reservation-name" as the key and specify the name of your reservation as its value.

- `values` (Optional) The list of label values of reservation resources. For example: the name of the specific reservation when using a key of "compute.googleapis.com/reservation-name"

The `sandbox_config` block supports:

- `sandbox_type` (Required) Which sandbox to use for pods in the node pool. Accepted values are:
  - `"gvisor"`: Pods run within a gVisor sandbox.

The `release_channel` block supports:

- `channel` - (Required) The selected release channel. Accepted values are:
  - UNSPECIFIED: Not set.
  - RAPID: Weekly upgrade cadence; Early testers and developers who requires new features.
  - REGULAR: Multiple per month upgrade cadence; Production users who need features not yet offered in the Stable channel.
  - STABLE: Every few months upgrade cadence; Production users who need stability above all else, and for whom frequent upgrades are too risky.

The `cost_management_config` block supports:

- `enabled` (Optional) - Whether to enable the cost allocation feature.

The `resource_usage_export_config` block supports:

- `enable_network_egress_metering` (Optional) - Whether to enable network egress metering for this cluster. If enabled, a daemonset will be created in the cluster to meter network egress traffic.
- `enable_resource_consumption_metering` (Optional) - Whether to enable resource consumption metering on this cluster. When enabled, a table will be created in the resource export BigQuery dataset to store resource consumption data. The resulting table can be joined with the resource usage table or with BigQuery billing export. Defaults to `true`.
- `bigquery_destination` (Required) - Parameters for using BigQuery as the destination of resource usage export.
- `bigquery_destination.dataset_id` (Required) - The ID of a BigQuery Dataset. For Example:

```
resource_usage_export_config {
  enable_network_egress_metering = false
  enable_resource_consumption_metering = true

  bigquery_destination {
    dataset_id = "cluster_resource_usage"
  }
}
```

The `shielded_instance_config` block supports:

- `enable_secure_boot` (Optional) - Defines if the instance has Secure Boot enabled.

Secure Boot helps ensure that the system only runs authentic software by verifying the digital signature of all boot components, and halting the boot process if signature verification fails. Defaults to `false`.

- `enable_integrity_monitoring` (Optional) - Defines if the instance has integrity monitoring enabled.

Enables monitoring and attestation of the boot integrity of the instance. The attestation is performed against the integrity policy baseline. This baseline is initially derived from the implicitly trusted boot image when the instance is created. Defaults to `true`.

The `taint` block supports:

- `key` (Required) Key for taint.
- `value` (Required) Value for taint.
- `effect` (Required) Effect for taint. Accepted values are `NO_SCHEDULE`, `PREFER_NO_SCHEDULE`, and `NO_EXECUTE`.

The `workload_metadata_config` block supports:

- `mode` (Required) How to expose the node metadata to the workload running on the node. Accepted values are:
  - MODE_UNSPECIFIED: Not Set
  - GCE_METADATA: Expose all Compute Engine metadata to pods.
  - GKE_METADATA: Run the GKE Metadata Server on this node. The GKE Metadata Server exposes a metadata API to workloads that is compatible with the V1 Compute Metadata APIs exposed by the Compute Engine and App Engine Metadata Servers. This feature can only be enabled if workload identity is enabled at the cluster level.

The `kubelet_config` block supports:

- `cpu_manager_policy` - (Required) The CPU management policy on the node. See K8S CPU Management Policies. One of `"none"` or `"static"`. Defaults to `none` when `kubelet_config` is unset.
- `cpu_cfs_quota` - (Optional) If true, enables CPU CFS quota enforcement for containers that specify CPU limits.
- `cpu_cfs_quota_period` - (Optional) The CPU CFS quota period value. Specified as a sequence of decimal numbers, each with optional fraction and a unit suffix, such as `"300ms"`. Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h". The value must be a positive duration.

Note

Note: At the time of writing (2020/08/18) the GKE API rejects the `none` value and accepts an invalid `default` value instead. While this remains true, not specifying the `kubelet_config` block should be the equivalent of specifying `none`.

- `pod_pids_limit` - (Optional) Controls the maximum number of processes allowed to run in a pod. The value must be greater than or equal to 1024 and less than 4194304.

The `linux_node_config` block supports:

- `sysctls` - (Optional) The Linux kernel parameters to be applied to the nodes and all pods running on the nodes. Specified as a map from the key, such as `net.core.wmem_max`, to a string value. Currently supported attributes can be found here. Note that validations happen all server side. All attributes are optional.

```
linux_node_config {
  sysctls = {
    "net.core.netdev_max_backlog" = "10000"
    "net.core.rmem_max"           = "10000"
  }
}
```

- `cgroup_mode` - (Optional) Possible cgroup modes that can be used. Accepted values are:
  - `CGROUP_MODE_UNSPECIFIED`: CGROUP_MODE_UNSPECIFIED is when unspecified cgroup configuration is used. The default for the GKE node OS image will be used.
  - `CGROUP_MODE_V1`: CGROUP_MODE_V1 specifies to use cgroupv1 for the cgroup configuration on the node image.
  - `CGROUP_MODE_V2`: CGROUP_MODE_V2 specifies to use cgroupv2 for the cgroup configuration on the node image.

The `vertical_pod_autoscaling` block supports:

- `enabled` (Required) - Enables vertical pod autoscaling

The `dns_config` block supports:

- `cluster_dns` - (Optional) Which in-cluster DNS provider should be used. `PROVIDER_UNSPECIFIED` (default) or `PLATFORM_DEFAULT` or `CLOUD_DNS`.
- `cluster_dns_scope` - (Optional) The scope of access to cluster DNS records. `DNS_SCOPE_UNSPECIFIED` (default) or `CLUSTER_SCOPE` or `VPC_SCOPE`.
- `cluster_dns_domain` - (Optional) The suffix used for all cluster service records.

The `gateway_api_config` block supports:

- `channel` - (Required) Which Gateway Api channel should be used. `CHANNEL_DISABLED`, `CHANNEL_EXPERIMENTAL` or `CHANNEL_STANDARD`.

The `protect_config` block supports:

- `workload_config` - (Optional, Beta) WorkloadConfig defines which actions are enabled for a cluster's workload configurations. Structure is documented below
- `workload_vulnerability_mode` - (Optional, Beta) Sets which mode to use for Protect workload vulnerability scanning feature. Accepted values are DISABLED, BASIC.

The `protect_config.workload_config` block supports:

- `audit_mode` - (Optional, Beta) Sets which mode of auditing should be used for the cluster's workloads. Accepted values are DISABLED, BASIC.

The `security_posture_config` block supports:

- `mode` - (Optional) Sets the mode of the Kubernetes security posture API's off-cluster features. Available options include `DISABLED` and `BASIC`.
- `vulnerability_mode` - (Optional) Sets the mode of the Kubernetes security posture API's workload vulnerability scanning. Available options include `VULNERABILITY_DISABLED`, `VULNERABILITY_BASIC` and `VULNERABILITY_ENTERPRISE`.

The `fleet` block supports:

- `project` - (Optional) The name of the Fleet host project where this cluster will be registered.

Attributes Reference

In addition to the arguments listed above, the following computed attributes are exported:

- `id` - an identifier for the resource with format `projects/{{project}}/locations/{{zone}}/clusters/{{name}}`
- `self_link` - The server-defined URL for the resource.
- `endpoint` - The IP address of this cluster's Kubernetes master.
- `label_fingerprint` - The fingerprint of the set of labels for this cluster.
- `maintenance_policy.0.daily_maintenance_window.0.duration` - Duration of the time window, automatically chosen to be smallest possible in the given scenario. Duration will be in RFC3339 format "PTnHnMnS".
- `master_auth.0.client_certificate` - Base64 encoded public certificate used by clients to authenticate to the cluster endpoint.

- `master_auth.0.client_key` - Base64 encoded private key used by clients to authenticate to the cluster endpoint.
- `master_auth.0.cluster_ca_certificate` - Base64 encoded public certificate that is the root certificate of the cluster.
- `master_version` - The current version of the master in the cluster. This may be different than the `min_master_version` set in the config if the master has been updated by GKE.
- `tpu_ipv4_cidr_block` - The IP address range of the Cloud TPUs in this cluster, in CIDR notation (e.g. `1.2.3.4/29`).
- `services_ipv4_cidr` - The IP address range of the Kubernetes services in this cluster, in CIDR notation (e.g. `1.2.3.4/29`). Service addresses are typically put in the last `/16` from the container CIDR.
- `cluster_autoscaling.0.auto_provisioning_defaults.0.management.0.upgrade_options` - Specifies the Auto Upgrade knobs for the node pool.
- `node_config.0.effective_taints` - List of kubernetes taints applied to each node. Structure is documented above.
- `fleet.0.membership` - The resource name of the fleet Membership resource associated to this cluster with format `//gkehub.googleapis.com/projects/{{project}}/locations/{{location}}/membership/{{name}}`. See the official doc for fleet management.

Timeouts

This resource provides the following Timeouts configuration options: configuration options:

- `create` - Default is 40 minutes.
- `read` - Default is 40 minutes.
- `update` - Default is 60 minutes.
- `delete` - Default is 40 minutes.

Import

GKE clusters can be imported using the `project`, `location`, and `name`. If the project is omitted, the default provider value will be used. Examples:

- `projects/{{project_id}}/locations/{{location}}/clusters/{{cluster_id}}`
- `{{project_id}}/{{location}}/{{cluster_id}}`
- `{{location}}/{{cluster_id}}`

In Terraform v1.5.0 and later, use an `import` block to import GKE clusters using one of the formats above. For example:

```
import {
  id = "projects/{{project_id}}/locations/{{location}}/clusters/{{cluster_id}}"
  to = google_container_cluster.default
}
```

When using the `terraform import` command, GKE clusters can be imported using one of the formats above. For example:

```
$ terraform import google_container_cluster.default
projects/{{project_id}}/locations/{{location}}/clusters/{{cluster_id}}

$ terraform import google_container_cluster.default {{project_id}}/{{location}}/{{cluster_id}}

$ terraform import google_container_cluster.default {{location}}/{{cluster_id}}
```

Note:

This resource has several fields that control Terraform-specific behavior and aren't present in the API. If they are set in config and you import a cluster, Terraform may need to perform an update immediately after import. Most of these updates should be no-ops but some may modify your cluster if the imported state differs.

For example, the following fields will show diffs if set in config:

- `min_master_version`
- `remove_default_node_pool`

User Project Overrides

This resource supports User Project Overrides.