» kubernetes_secret

The resource provides mechanisms to inject containers with sensitive information, such as passwords, while keeping containers agnostic of Kubernetes. Secrets can be used to store sensitive information either as individual properties or coarse-grained entries like entire files or JSON blobs. The resource will by default create a secret which is available to any pod in the specified (or default) namespace.

Read more about security properties and risks involved with using Kubernetes secrets: Kubernetes reference

Note: All arguments including the secret data will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
data "kubernetes_secret" "example" {
  metadata {
    name = "basic-auth"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard secret's metadata. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `name` - (Required) Name of the secret, must be unique. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the secret must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.

- `resource_version` - An opaque value that represents the internal version of this secret that can be used by clients to determine when secret has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this secret.
- `uid` - The unique in time and space value for this secret. For more info see Kubernetes reference

» Attribute Reference

- `data` - A map of the secret data.
- `type` - The secret type. Defaults to `Opaque`. For more info see Kubernetes reference

» kubernetes_service

A Service is an abstraction which defines a logical set of pods and a policy by which to access them - sometimes called a micro-service. This data source allows you to pull data about such service.

» Example Usage

```
data "kubernetes_service" "example" {
  metadata {
    name = "terraform-example"
  }
}

resource "aws_route53_record" "example" {
  zone_id = "${data.aws_route53_zone.k8.zone_id}"
  name    = "example"
  type    = "CNAME"
  ttl     = "300"
  records = ["${data.kubernetes_service.example.load_balancer_ingress.0.hostname}"]
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard service's metadata. For more info see Kubernetes reference

» Attributes

- `spec` - Spec defines the behavior of a service. Kubernetes reference
- `load_balancer_ingress` - A list containing ingress points for the load-balancer (only valid if `type = "LoadBalancer"`)

» Nested Blocks

» `metadata`

» Arguments

- `name` - (Optional) Name of the service, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the service must be unique.

» Attributes

- `annotations` - (Optional) An unstructured key value map stored with the service that may be used to store arbitrary metadata. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service. May match selectors of replication controllers and services. For more info see Kubernetes reference
- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service that can be used by clients to determine when service has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this service.
- `uid` - The unique in time and space value for this service. For more info see Kubernetes reference

» `port`

» Attributes

- `name` - The name of this port within the service. All ports within the service must have unique names. Optional if only one ServicePort is defined on this service.
- `node_port` - The port on each node on which this service is exposed when `type` is `NodePort` or `LoadBalancer`. Usually assigned by the system. If specified, it will be allocated to the service if unused or else creation of

the service will fail. Default is to auto-allocate a port if the `type` of this service requires one. For more info see Kubernetes reference
- `port` - The port that will be exposed by this service.
- `protocol` - The IP protocol for this port. Supports `TCP` and `UDP`. Default is `TCP`.
- `target_port` - Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. This field is ignored for services with `cluster_ip = "None"`. For more info see Kubernetes reference

» `spec`

» Attributes

- `cluster_ip` - The IP address of the service. It is usually assigned randomly by the master. If an address is specified manually and is not in use by others, it will be allocated to the service; otherwise, creation of the service will fail. `None` can be specified for headless services when proxying is not required. Ignored if type is `ExternalName`. For more info see Kubernetes reference
- `external_ips` - A list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.
- `external_name` - The external reference that kubedns or equivalent will return as a CNAME record for this service. No proxying will be involved. Must be a valid DNS name and requires `type` to be `ExternalName`.
- `external_traffic_policy` - (Optional) Denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints. `Local` preserves the client source IP and avoids a second hop for LoadBalancer and Nodeport type services, but risks potentially imbalanced traffic spreading. `Cluster` obscures the client source IP and may cause a second hop to another node, but should have good overall load-spreading. More info: https://kubernetes.io/docs/tutorials/services/source-ip/
- `load_balancer_ip` - Only applies to `type = LoadBalancer`. LoadBalancer will get created with the IP specified in this field. This feature depends on whether the underlying cloud-provider supports specifying this field when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.
- `load_balancer_source_ranges` - If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature. For more info see Kubernetes reference

- `port` - The list of ports that are exposed by this service. For more info see Kubernetes reference
- `selector` - Route service traffic to pods with label keys and values matching this selector. Only applies to types `ClusterIP`, `NodePort`, and `LoadBalancer`. For more info see Kubernetes reference
- `session_affinity` - Used to maintain session affinity. Supports `ClientIP` and `None`. Defaults to `None`. For more info see Kubernetes reference
- `type` - Determines how the service is exposed. Defaults to `ClusterIP`. Valid options are `ExternalName`, `ClusterIP`, `NodePort`, and `LoadBalancer`. `ExternalName` maps to the specified `external_name`. For more info see Kubernetes reference

» `load_balancer_ingress`

» Attributes

- `hostname` - Hostname which is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)
- `ip` - IP which is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)

» kubernetes_storage_class

Storage class is the foundation of dynamic provisioning, allowing cluster administrators to define abstractions for the underlying storage platform.

Read more at https://kubernetes.io/blog/2017/03/dynamic-provisioning-and-storage-classes-kubernetes/

» Example Usage

```
data "kubernetes_storage_class" "example" {
  metadata {
    name = "terraform-example"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard storage class's metadata. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `name` - (Required) Name of the storage class, must be unique. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this storage class that can be used by clients to determine when storage class has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this storage class.
- `uid` - The unique in time and space value for this storage class. For more info see Kubernetes reference

» Argument Reference

The following attributes are exported:

- `parameters` - The parameters for the provisioner that creates volume of this storage class. Read more about available parameters.
- `storage_provisioner` - Indicates the type of the provisioner this storage class represents
- `reclaim_policy` - Indicates the reclaim policy used.
- `volume_binding_mode` - Indicates when volume binding and dynamic provisioning should occur.
- `allow_volume_expansion` - Indicates whether the storage class allow volume expand.

» kubernetes_api_service

An API Service is an abstraction which defines for locating and communicating with servers.

» Example Usage

```
resource "kubernetes_api_service" "example" {
  metadata {
```

```
    name = "terraform-example"
  }
  spec {
    selector {
      app = "${kubernetes_pod.example.metadata.0.labels.app}"
    }
    session_affinity = "ClientIP"
    port {
      port = 8080
      target_port = 80
    }

    type = "LoadBalancer"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard API service's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec contains information for locating and communicating with a server. Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the API service that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference

- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the API service. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the API service, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this API service that can be used by clients to determine when API service has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this API service.
- `uid` - The unique in time and space value for this API service. For more info see Kubernetes reference

» `spec`

» Arguments

- `ca_bundle` - (Optional) CABundle is a PEM encoded CA bundle which will be used to validate an API server's serving certificate. If unspecified, system trust roots on the apiserver are used.
- `group` - (Required) Group is the API group name this server hosts.
- `group_priority_minimum` - (Required) GroupPriorityMininum is the priority this group should have at least. Higher priority means that the group is preferred by clients over lower priority ones. Note that other versions of this group might specify even higher GroupPriorityMininum values such that the whole group gets a higher priority. The primary sort is based on GroupPriorityMinimum, ordered highest number to lowest (20 before 10). The secondary sort is based on the alphabetical comparison of the name of the object. (v1.bar before v1.foo) We'd recommend something like: *.k8s.io (except extensions) at 18000 and PaaSes (OpenShift, Deis) are recommended to be in the 2000s.
- `insecure_skip_tls_verify` - (Required) InsecureSkipTLSVerify disables TLS certificate verification when communicating with this server. This is strongly discouraged. You should use the CABundle instead.

- `service` - (Optional) Service is a reference to the service for this API server. It must communicate on port 443. If the Service is nil, that means the handling for the API groupversion is handled locally on this server. The call will simply delegate to the normal handler chain to be fulfilled. See `service` block attributes below.
- `version` - (Required) Version is the API version this server hosts. For example, `v1`.
- `version_priority` - (Required) VersionPriority controls the ordering of this API version inside of its group. Must be greater than zero. The primary sort is based on VersionPriority, ordered highest to lowest (20 before 10). Since it's inside of a group, the number can be small, probably in the 10s. In case of equal version priorities, the version string will be used to compute the order inside a group. If the version string is `kube-like`, it will sort above non `kube-like` version strings, which are ordered lexicographically. `Kube-like` versions start with a `v`, then are followed by a number (the major version), then optionally the string `alpha` or `beta` and another number (the minor version). These are sorted first by GA > `beta` > `alpha` (where GA is a version with no suffix such as `beta` or `alpha`), and then by comparing major version, then minor version. An example sorted list of versions: `v10`, `v2`, `v1`, `v11beta2`, `v10beta3`, `v3beta1`, `v12alpha1`, `v11alpha2`, `foo1`, `foo10`..

» `service`

» Arguments

- `name` - (Required) Name is the name of the service.
- `namespace` - (Required) Namespace is the namespace of the service.

» Import

API service can be imported using its name, e.g.

```
$ terraform import kubernetes_api_service.example v1.terraform-name.k8s.io
```

» kubernetes_cluster_role

A ClusterRole creates a role at the cluster level and in all namespaces.

» Example Usage

```
resource "kubernetes_cluster_role" "example" {
  metadata {
```

```
    name = "terraform-example"
  }

  rule {
    api_groups = [""]
    resources  = ["namespaces", "pods"]
    verbs      = ["get", "list", "watch"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard kubernetes metadata. For more info see Kubernetes reference
- `rule` - (Required) The PolicyRoles for this ClusterRole. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the cluster role binding that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the cluster role binding. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause**

a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). For more info see Kubernetes reference

- `name` - (Optional) Name of the cluster role binding, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when the object has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this cluster role binding.
- `uid` - The unique in time and space value for this cluster role binding. For more info see Kubernetes reference

» `rule`

» Arguments

- `api_groups` - (Optional) APIGroups is the name of the APIGroup that contains the resources. If multiple API groups are specified, any action requested against one of the enumerated resources in any API group will be allowed.
- `non_resource_urls` - (Optional) NonResourceURLs is a set of partial urls that a user should have access to. *s are allowed, but only as the full, final step in the path Since non-resource URLs are not namespaced, this field is only applicable for ClusterRoles referenced from a ClusterRoleBinding. Rules can either apply to API resources (such as "pods" or "secrets") or non-resource URL paths (such as "/api"), but not both.
- `resource_names` - (Optional) ResourceNames is an optional white list of names that the rule applies to. An empty set means that everything is allowed.
- `resources` - (Optional) Resources is a list of resources this rule applies to. ResourceAll represents all resources.
- `verbs` - (Required) Verbs is a list of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule. VerbAll represents all kinds.

» Import

ClusterRole can be imported using the name, e.g.

```
$ terraform import kubernetes_cluster_role.example terraform-name
```

» kubernetes_cluster_role_binding

A ClusterRoleBinding may be used to grant permission at the cluster level and in all namespaces

» Example Usage

```
resource "kubernetes_cluster_role_binding" "example" {
  metadata {
    name = "terraform-example"
  }
  role_ref {
    api_group = "rbac.authorization.k8s.io"
    kind      = "ClusterRole"
    name      = "cluster-admin"
  }
  subject {
    kind      = "User"
    name      = "admin"
    api_group = "rbac.authorization.k8s.io"
  }
  subject {
    kind      = "ServiceAccount"
    name      = "default"
    namespace = "kube-system"
  }
  subject {
    kind      = "Group"
    name      = "system:masters"
    api_group = "rbac.authorization.k8s.io"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard kubernetes metadata. For more info see Kubernetes reference
- `role_ref` - (Required) The ClusterRole to bind Subjects to. For more info see Kubernetes reference
- `subject` - (Required) The Users, Groups, or ServiceAccounts to grant permissions to. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the cluster role binding that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the cluster role binding. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the cluster role binding, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.

- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when the object has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this cluster role binding.
- `uid` - The unique in time and space value for this cluster role binding. For more info see Kubernetes reference

» `role_ref`

» Arguments

- `name` - (Required) The name of this ClusterRole to bind Subjects to.
- `kind` - (Required) The type of binding to use. This value must be and defaults to `ClusterRole`
- `api_group` - (Optional) The API group to drive authorization decisions. This value must be and defaults to `rbac.authorization.k8s.io`

» `subject`

» Arguments

- `name` - (Required) The name of this ClusterRole to bind Subjects to.
- `namespace` - (Optional) Namespace defines the namespace of the ServiceAccount to bind to. This value only applies to kind `ServiceAccount`
- `kind` - (Required) The type of binding to use. This value must be `ServiceAccount`, `User` or `Group`
- `api_group` - (Optional) The API group to drive authorization decisions. This value only applies to kind `User` and `Group`. It must be `rbac.authorization.k8s.io`

» Import

ClusterRoleBinding can be imported using the name, e.g.

```
$ terraform import kubernetes_cluster_role_binding.example terraform-name
```

» kubernetes_config_map

The resource provides mechanisms to inject containers with configuration data while keeping containers agnostic of Kubernetes. Config Map can be used to store fine-grained information like individual properties or coarse-grained information like entire config files or JSON blobs.

» Example Usage

```
resource "kubernetes_config_map" "example" {
  metadata {
    name = "my-config"
  }

  data = {
    api_host             = "myhost:443"
    db_host              = "dbhost:5432"
    "my_config_file.yml" = "${file("${path.module}/my_config_file.yml")}"
  }

  binary_data = {
    "my_payload.bin" = "${filebase64("${path.module}/my_payload.bin")}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `binary_data` - (Optional) BinaryData contains the binary data. Each key must consist of alphanumeric characters, '-', '_' or '.'. BinaryData can contain byte sequences that are not in the UTF-8 range. The keys stored in BinaryData must not overlap with the ones in the Data field, this is enforced during validation process. Using this field will require 1.10+ apiserver and kubelet. This field only accepts base64-encoded payloads that will be decoded/received before being sent/received to the apiserver.
- `data` - (Optional) Data contains the configuration data. Each key must consist of alphanumeric characters, '-', '_' or '.'. Values with non-UTF-8 byte sequences must use the BinaryData field. The keys stored in Data must not overlap with the keys in the BinaryData field, this is enforced during validation process.
- `metadata` - (Required) Standard config map's metadata. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the config map that may be used to store arbitrary metadata. **By default,**

the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the config map. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the config map, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the config map must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this config map that can be used by clients to determine when config map has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this config map.
- `uid` - The unique in time and space value for this config map. For more info see Kubernetes reference

» Import

Config Map can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_config_map.example default/my-config
```

» kubernetes_cron_job

A Cron Job creates Jobs on a time-based schedule.

One CronJob object is like one line of a crontab (cron table) file. It runs a job periodically on a given schedule, written in Cron format.

Note: All CronJob `schedule` times are based on the timezone of the master where the job is initiated. For instructions on creating and working with cron jobs, and for an example of a spec file for a cron job, see Running automated tasks with cron jobs.

» Example Usage

```
resource "kubernetes_cron_job" "demo" {
  metadata {
    name = "demo"
  }
  spec {
    concurrency_policy            = "Replace"
    failed_jobs_history_limit     = 5
    schedule                      = "1 0 * * *"
    starting_deadline_seconds     = 10
    successful_jobs_history_limit = 10
    suspend                       = true
    job_template {
      metadata {}
      spec {
        backoff_limit = 2
        template {
          metadata {}
          spec {
            container {
              name    = "hello"
              image   = "busybox"
              command = ["/bin/sh", "-c", "date; echo Hello from the Kubernetes cluster"]
            }
            restart_policy = "OnFailure"
          }
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard resource's metadata. More info: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#spec-and-status
- `spec` - (Required) Spec defines the behavior of a CronJob. https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#spec-and-status

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the resource that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/annotations
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. Read more: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#idempotency
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/labels
- `name` - (Optional) Name of the service, must be unique. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#

names
- `namespace` - (Optional) Namespace defines the space within which name of the service must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service that can be used by clients to determine when service has changed. Read more: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#concurrency-control-and-consistency
- `self_link` - A URL representing this service.
- `uid` - The unique in time and space value for this service. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

» `spec`

» Arguments

- `concurrency_policy` - (Optional) Specifies how to treat concurrent executions of a Job. Valid values are: - "Allow" (default): allows CronJobs to run concurrently; - "Forbid": forbids concurrent runs, skipping next run if previous run hasn't finished yet; - "Replace": cancels currently running job and replaces it with a new one
- `failed_jobs_history_limit` - (Optional) The number of failed finished jobs to retain. This is a pointer to distinguish between explicit zero and not specified. Defaults to 1.
- `job_template` - (Required) Specifies the job that will be created when executing a CronJob.
- `schedule` - (Required) The schedule in Cron format, see https://en.wikipedia.org/wiki/Cron.
- `starting_deadline_seconds` - (Optional) Deadline in seconds for starting the job if it misses scheduled time for any reason. Missed jobs executions will be counted as failed ones.
- `successful_jobs_history_limit` - (Optional) The number of successful finished jobs to retain. This is a pointer to distinguish between explicit zero and not specified. Defaults to 3.
- `suspend` - (Optional) This flag tells the controller to suspend subsequent executions, it does not apply to already started executions. Defaults to false.

» `job_template`

» **Arguments**

- `metadata` - (Required) Standard object's metadata of the jobs created from this template. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata
- `spec` - (Required) Specification of the desired behavior of the job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

» `spec`

» **Arguments**

- `active_deadline_seconds` - (Optional) Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
- `backoff_limit` - (Optional) Specifies the number of retries before marking this job failed. Defaults to 6
- `completions` - (Optional) Specifies the desired number of successfully finished pods the job should be run with. Setting to nil means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
- `manual_selector` - (Optional) Controls generation of pod labels and pod selectors. Leave `manualSelector` unset unless you are certain what you are doing. When false or unset, the system pick labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see `manualSelector=true` in jobs that were created with the old `extensions/v1beta1` API. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector
- `parallelism` - (Optional) Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when `((.spec.completions - .status.successful) < .spec.parallelism)`, i.e. when the work left to do is less than max parallelism. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
- `selector` - (Optional) A label query over pods that should match the pod count. Normally, the system sets this field for you. More

info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
- `template` - (Optional) Describes the pod that will be created when executing a job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

» `selector`

» Arguments

- `match_expressions` - (Optional) A list of label selector requirements. The requirements are ANDed.
- `match_labels` - (Optional) A map of `{key,value}` pairs. A single `{key,value}` in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

» `template`

» Arguments

These arguments are the same as the for the `spec` block of a Pod.

Please see the Pod resource for reference.

» kubernetes_daemonset

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

» Example Usage

```
resource "kubernetes_daemonset" "example" {
  metadata {
    name      = "terraform-example"
    namespace = "something"
    labels = {
      test = "MyExampleApp"
    }
  }
```

```
spec {
  selector {
    match_labels = {
      test = "MyExampleApp"
    }
  }

  template {
    metadata {
      labels = {
        test = "MyExampleApp"
      }
    }

    spec {
      container {
        image = "nginx:1.7.8"
        name  = "example"

        resources {
          limits {
            cpu    = "0.5"
            memory = "512Mi"
          }
          requests {
            cpu    = "250m"
            memory = "50Mi"
          }
        }

        liveness_probe {
          http_get {
            path = "/nginx_status"
            port = 80

            http_header {
              name  = "X-Custom-Header"
              value = "Awesome"
            }
          }

          initial_delay_seconds = 3
          period_seconds        = 3
        }

      }
```

```
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard daemonset's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the specification of the desired behavior of the daemonset. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the deployment that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the deployment. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). Must match `selector`**. For more info see Kubernetes reference

- `name` - (Optional) Name of the deployment, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the deployment must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this deployment that can be used by clients to determine when deployment has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this deployment.
- `uid` - The unique in time and space value for this deployment. For more info see Kubernetes reference

» `spec`

» Arguments

- `min_ready_seconds` - (Optional) Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
- `revision_history_limit` - (Optional) The number of old ReplicaSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.
- `strategy` - (Optional) The update strategy to use to replace existing pods with new ones.
- `selector` - (Optional) A label query over pods that should match the Replicas count. Label keys and values that must match in order to be controlled by this deployment. **Must match labels (`metadata.0.labels`)**. For more info see Kubernetes reference
- `template` - (Required) Describes the pod that will be created per Node. This takes precedence over a TemplateRef. For more info see Kubernetes reference

» `strategy`

» Arguments

- `type` - Type of daemon set update. Can be 'RollingUpdate' or 'OnDelete'. Default is 'RollingUpdate'.
- `rolling_update` - Rolling update config params. Present only if type = 'RollingUpdate'.

» `rolling_update`

» Arguments

- `max_unavailable` - The maximum number of DaemonSet pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of total number of DaemonSet pods at the start of the update (ex: 10%). Absolute number is calculated from percentage by rounding up. This cannot be 0. Default value is 1. Example: when this is set to 30%, at most 30% of the total number of nodes that should be running the daemon pod (i.e. status.desiredNumberScheduled) can have their pods stopped for an update at any given time. The update starts by stopping at most 30% of those DaemonSet pods and then brings up new DaemonSet pods in their place. Once the new pods are available, it then proceeds onto other DaemonSet pods, thus ensuring that at least 70% of original number of DaemonSet pods are available at all times during the update.

» `template`

» Arguments

- `metadata` - (Required) Standard object's metadata. For more info see https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata
- `spec` - (Required) Specification of the desired behavior of the pod. For more info see https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

» template `spec`

» Arguments

- `affinity` - (Optional) A group of affinity scheduling rules. If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
- `active_deadline_seconds` - (Optional) Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
- `automount_service_account_token` - (Optional) Indicates whether a service account token should be automatically mounted.
- `container` - (Optional) List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated. For more info see Kubernetes reference

- `init_container` - (Optional) List of init containers belonging to the pod. Init containers always run to completion and each must complete successfully before the next is started. For more info see Kubernetes reference/
- `dns_policy` - (Optional) Set DNS policy for containers within the pod. Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'. Optional: Defaults to 'ClusterFirst', see Kubernetes reference.
- `dns_config` - (Optional) Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy. Defaults to empty. See `dns_config` block definition below.
- `host_alias` - (Optional) List of hosts and IPs that will be injected into the pod's hosts file if specified. Optional: Defaults to empty. See `host_alias` block definition below.
- `host_ipc` - (Optional) Use the host's ipc namespace. Optional: Defaults to false.
- `host_network` - (Optional) Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified.
- `host_pid` - (Optional) Use the host's pid namespace.
- `hostname` - (Optional) Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.
- `image_pull_secrets` - (Optional) ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. For example, in the case of docker, only DockerConfig type secrets are honored. For more info see Kubernetes reference
- `node_name` - (Optional) NodeName is a request to schedule this pod onto a specific node. If it is non-empty, the scheduler simply schedules this pod onto that node, assuming that it fits resource requirements.
- `node_selector` - (Optional) NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. For more info see Kubernetes reference.
- `restart_policy` - (Optional) Restart policy for all containers within the pod. One of Always, OnFailure, Never. For more info see Kubernetes reference.
- `security_context` - (Optional) SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty
- `service_account_name` - (Optional) ServiceAccountName is the name of the ServiceAccount to use to run this pod. For more info see http://releases.k8s.io/HEAD/docs/design/service_accounts.md.

- `share_process_namespace` - (Optional) Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set.
- `subdomain` - (Optional) If specified, the fully qualified Pod hostname will be "...svc.". If not specified, the pod will not have a domainname at all..
- `termination_grace_period_seconds` - (Optional) Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process.
- `toleration` - (Optional) Optional pod node tolerations. For more info see Kubernetes reference
- `volume` - (Optional) List of volumes that can be mounted by containers belonging to the pod. For more info see Kubernetes reference

» `affinity`

» Arguments

- `node_affinity` - (Optional) Node affinity scheduling rules for the pod. For more info see Kubernetes reference
- `pod_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference
- `pod_anti_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference

» `node_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution`

» Arguments

- `node_selector_term` - (Required) A list of node selector terms. The terms are ORed.

» `node_selector_term`

» Arguments

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» Arguments

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `preferred_during_scheduling_ignored_during_execution`

» Arguments

- `preference` - (Required) A node selector term, associated with the corresponding weight.
- `weight` - (Required) Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

» `preference`

» Arguments

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» Arguments

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `pod_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `pod_anti_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the anti-affinity requirements specified by this field are not met

at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.

- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution` (pod_affinity_term)

» Arguments

- `label_selector` - (Optional) A label query over a set of resources, in this case pods.
- `namespaces` - (Optional) Specifies which namespaces the `label_selector` applies to (matches against). Null or empty list means "this pod's namespace"
- `topology_key` - (Optional) This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the `label_selector` in the specified namespaces, where co-located is defined as running on a node whose value of the label with key `topology_key` matches that of any node on which any of the selected pods is running. Empty `topology_key` is not allowed.

» `preferred_during_scheduling_ignored_during_execution`

» Arguments

- `pod_affinity_term` - (Required) A pod affinity term, associated with the corresponding weight.
- `weight` - (Required) Weight associated with matching the corresponding `pod_affinity_term`, in the range 1-100.

» `container`

» Arguments

- `args` - (Optional) Arguments to the entrypoint. The docker image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference

- `command` - (Optional) Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference
- `env` - (Optional) Block of string name and value pairs to set in the container's environment. May be declared multiple times. Cannot be updated.
- `env_from` - (Optional) List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
- `image` - (Optional) Docker image name. For more info see Kubernetes reference
- `image_pull_policy` - (Optional) Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. For more info see Kubernetes reference
- `lifecycle` - (Optional) Actions that the management system should take in response to container lifecycle events
- `liveness_probe` - (Optional) Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. For more info see Kubernetes reference
- `name` - (Required) Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
- `port` - (Optional) List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.
- `readiness_probe` - (Optional) Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. For more info see Kubernetes reference
- `resources` - (Optional) Compute Resources required by this container. Cannot be updated. For more info see Kubernetes reference
- `security_context` - (Optional) Security options the pod should run with. For more info see http://releases.k8s.io/HEAD/docs/design/security_context.md
- `stdin` - (Optional) Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the

container will always result in EOF.

- `stdin_once` - (Optional) Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF.
- `termination_message_path` - (Optional) Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Defaults to /dev/termination-log. Cannot be updated.
- `tty` - (Optional) Whether this container should allocate a TTY for itself
- `volume_mount` - (Optional) Pod volumes to mount into the container's filesystem. Cannot be updated.
- `working_dir` - (Optional) Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

» `aws_elastic_block_store`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
- `read_only` - (Optional) Whether to set the read-only property in VolumeMounts to "true". If omitted, the default is "false". For more info see Kubernetes reference
- `volume_id` - (Required) Unique ID of the persistent disk resource in AWS (Amazon EBS volume). For more info see Kubernetes reference

» `azure_disk`

» Arguments

- `caching_mode` - (Required) Host Caching mode: None, Read Only, Read Write.

- `data_disk_uri` - (Required) The URI the data disk in the blob storage
- `disk_name` - (Required) The Name of the data disk in the blob storage
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).

» `azure_file`

» **Arguments**

- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `secret_name` - (Required) The name of secret that contains Azure Storage Account Name and Key
- `share_name` - (Required) Share Name

» `capabilities`

» **Arguments**

- `add` - (Optional) Added capabilities
- `drop` - (Optional) Removed capabilities

» `ceph_fs`

» **Arguments**

- `monitors` - (Required) Monitors is a collection of Ceph monitors For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `path` - (Optional) Used as the mounted root, rather than the full Ceph tree, default is /
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to `false` (read/write). For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_file` - (Optional) The path to key ring for User, default is /etc/ceph/user.secret For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_ref` - (Optional) Reference to the authentication secret for User, default is empty. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

- `user` - (Optional) User is the rados user name, default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

» `cinder`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write). For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `volume_id` - (Required) Volume ID used to identify the volume in Cinder. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

» `config_map`

» Arguments

- `default_mode` - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `config_map_ref`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the ConfigMap must be defined

» config_map_key_ref

» Arguments

- key - (Optional) The key to select.
- name - (Optional) Name of the referent. For more info see Kubernetes reference

» dns_config

» Arguments

- nameservers - (Optional) A list of DNS name server IP addresses specified as strings. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed. Optional: Defaults to empty.
- option - (Optional) A list of DNS resolver options specified as blocks with name/value pairs. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy. Optional: Defaults to empty.
- searches - (Optional) A list of DNS search domains for host-name lookup specified as strings. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed. Optional: Defaults to empty.

The option block supports the following:

- name - (Required) Name of the option.
- value - (Optional) Value of the option. Optional: Defaults to empty.

» downward_api

» Arguments

- default_mode - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- items - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.

» `empty_dir`

» **Arguments**

- `medium` - (Optional) What type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. For more info see Kubernetes reference

» `env`

» **Arguments**

- `name` - (Required) Name of the environment variable. Must be a C_IDENTIFIER
- `value` - (Optional) Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
- `value_from` - (Optional) Source for the environment variable's value

» `env_from`

» **Arguments**

- `config_map_ref` - (Optional) The ConfigMap to select from
- `prefix` - (Optional) An optional identifer to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER..
- `secret_ref` - (Optional) The Secret to select from

» `exec`

» **Arguments**

- `command` - (Optional) Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

» `fc`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `lun` - (Required) FC target lun number
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `target_ww_ns` - (Required) FC target worldwide names (WWNs)

» `field_ref`

» **Arguments**

- `api_version` - (Optional) Version of the schema the FieldPath is written in terms of, defaults to "v1".
- `field_path` - (Optional) Path of the field to select in the specified API version

» `flex_volume`

» **Arguments**

- `driver` - (Required) Driver is the name of the driver to use for this volume.
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
- `options` - (Optional) Extra command options if any.
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false (read/write).
- `secret_ref` - (Optional) Reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

» `flocker`

» **Arguments**

- `dataset_name` - (Optional) Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
- `dataset_uuid` - (Optional) UUID of the dataset. This is unique identifier of a Flocker dataset

» `gce_persistent_disk`

» **Arguments**

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as ”1”. Similarly, the volume partition for /dev/sda is ”0” (or you can leave the property empty). For more info see Kubernetes reference
- `pd_name` - (Required) Unique name of the PD resource in GCE. Used to identify the disk in GCE. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false. For more info see Kubernetes reference

» `git_repo`

» **Arguments**

- `directory` - (Optional) Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
- `repository` - (Optional) Repository URL
- `revision` - (Optional) Commit hash for the specified revision.

» `glusterfs`

» **Arguments**

- `endpoints_name` - (Required) The endpoint name that details Glusterfs topology. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `path` - (Required) The Glusterfs volume path. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `read_only` - (Optional) Whether to force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

» `host_alias`

» Arguments

- `hostnames` - (Required) Hostnames for the IP address.
- `ip` - (Required) IP address of the host file entry.

» `host_path`

» Arguments

- `path` - (Optional) Path of the directory on the host. For more info see Kubernetes reference
- `type` - (Optional) Type for HostPath volume. Defaults to "". For more info see Kubernetes reference

» `http_get`

» Arguments

- `host` - (Optional) Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
- `http_header` - (Optional) Scheme to use for connecting to the host.
- `path` - (Optional) Path to access on the HTTP server.
- `port` - (Optional) Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
- `scheme` - (Optional) Scheme to use for connecting to the host.

» `http_header`

» Arguments

- `name` - (Optional) The header field name
- `value` - (Optional) The header field value

» `image_pull_secrets`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference

» `iscsi`

» **Arguments**

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see Kubernetes reference
- `iqn` - (Required) Target iSCSI Qualified Name.
- `iscsi_interface` - (Optional) iSCSI interface name that uses an iSCSI transport. Defaults to 'default' (tcp).
- `lun` - (Optional) iSCSI target lun number.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false.
- `target_portal` - (Required) iSCSI target portal. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

» `items`

» **Arguments**

- `key` - (Optional) The key to project.
- `mode` - (Optional) Optional: mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `path` - (Optional) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `lifecycle`

» **Arguments**

- `post_start` - (Optional) post_start is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. For more info see Kubernetes reference
- `pre_stop` - (Optional) pre_stop is called immediately before a container is terminated. The container is terminated after the handler completes. The reason for termination is passed to the handler. Regardless of the outcome of the handler, the container is eventually terminated. Other management of the container blocks until the hook completes. For more info see Kubernetes reference

» `limits`

» **Arguments**

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `liveness_probe`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `nfs`

» **Arguments**

- `path` - (Required) Path that is exported by the NFS server. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the NFS export to be mounted with read-only permissions. Defaults to false. For more info see Kubernetes reference
- `server` - (Required) Server is the hostname or IP address of the NFS server. For more info see Kubernetes reference

» `persistent_volume_claim`

» **Arguments**

- `claim_name` - (Optional) ClaimName is the name of a PersistentVolumeClaim in the same
- `read_only` - (Optional) Will force the ReadOnly setting in VolumeMounts.

» `photon_persistent_disk`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `pd_id` - (Required) ID that identifies Photon Controller persistent disk

» `port`

» **Arguments**

- `container_port` - (Required) Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
- `host_ip` - (Optional) What host IP to bind the external port to.
- `host_port` - (Optional) Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
- `name` - (Optional) If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services
- `protocol` - (Optional) Protocol for port. Must be UDP or TCP. Defaults to "TCP".

» `post_start`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `pre_stop`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `quobyte`

» **Arguments**

- `group` - (Optional) Group to map volume access to Default is no group
- `read_only` - (Optional) Whether to force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
- `registry` - (Required) Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
- `user` - (Optional) User to map volume access to Defaults to serivceaccount user
- `volume` - (Required) Volume is a string that references an already created Quobyte volume by name.

» `rbd`

» **Arguments**

- `ceph_monitors` - (Required) A collection of Ceph monitors. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `keyring` - (Optional) Keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rados_user` - (Optional) The rados user name. Default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_image` - (Required) The rados image name. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_pool` - (Optional) The rados pool name. Default is rbd. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `secret_ref` - (Optional) Name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

» `readiness_probe`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `resources`

» **Arguments**

- `limits` - (Optional) Describes the maximum amount of compute resources allowed. For more info see Kubernetes reference/
- `requests` - (Optional) Describes the minimum amount of compute resources required.

» `requests`

» **Arguments**

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `resource_field_ref`

» **Arguments**

- `container_name` - (Optional) The name of the container
- `resource` - (Required) Resource to select

» `se_linux_options`

» **Arguments**

- `level` - (Optional) Level is SELinux level label that applies to the container.
- `role` - (Optional) Role is a SELinux role label that applies to the container.
- `type` - (Optional) Type is a SELinux type label that applies to the container.
- `user` - (Optional) User is a SELinux user label that applies to the container.

» `secret`

» Arguments

- `default_mode` - (Optional) Mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) List of Secret Items to project into the volume. See `items` block definition below. If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked `optional`. Paths must be relative and may not contain the '..' path or start with '..'.
- `optional` - (Optional) Specify whether the Secret or it's keys must be defined.
- `secret_name` - (Optional) Name of the secret in the pod's namespace to use. For more info see Kubernetes reference

The `items` block supports the following:

- `key` - (Required) The key to project.
- `mode` - (Optional) Mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used.
- `path` - (Required) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `secret_ref`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the Secret must be defined

» `secret_key_ref`

» **Arguments**

- `key` - (Optional) The key of the secret to select from. Must be a valid secret key.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `secret_ref`

» **Arguments**

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» **container** `security_context`

» **Arguments**

- `allow_privilege_escalation` - (Optional) AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN
- `capabilities` - (Optional) The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
- `privileged` - (Optional) Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.
- `read_only_root_filesystem` - (Optional) Whether this container has a read-only root filesystem. Default is false.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `se_linux_options` - (Optional) The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random

SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

» `capabilities`

» Arguments

- `add` - (Optional) A list of added capabilities.
- `drop` - (Optional) A list of removed capabilities.

» pod `security_context`

» Arguments

- `fs_group` - (Optional) A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `se_linux_options` - (Optional) The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `supplemental_groups` - (Optional) A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.

» `tcp_socket`

» Arguments

- `port` - (Required) Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

» `value_from`

» Arguments

- `config_map_key_ref` - (Optional) Selects a key of a ConfigMap.
- `field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..
- `resource_field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..
- `secret_key_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..

» `toleration`

» Arguments

- `effect` - (Optional) Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
- `key` - (Optional) Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
- `operator` - (Optional) Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
- `toleration_seconds` - (Optional) TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
- `value` - (Optional) Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

» `volume`

» **Arguments**

- `aws_elastic_block_store` - (Optional) Represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. For more info see Kubernetes reference
- `azure_disk` - (Optional) Represents an Azure Data Disk mount on the host and bind mount to the pod.
- `azure_file` - (Optional) Represents an Azure File Service mount on the host and bind mount to the pod.
- `ceph_fs` - (Optional) Represents a Ceph FS mount on the host that shares a pod's lifetime
- `cinder` - (Optional) Represents a cinder volume attached and mounted on kubelets host machine. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `config_map` - (Optional) ConfigMap represents a configMap that should populate this volume
- `downward_api` - (Optional) DownwardAPI represents downward API about the pod that should populate this volume
- `empty_dir` - (Optional) EmptyDir represents a temporary directory that shares a pod's lifetime. For more info see Kubernetes reference
- `fc` - (Optional) Represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
- `flex_volume` - (Optional) Represents a generic volume resource that is provisioned/attached using an exec based plugin. This is an alpha feature and may change in future.
- `flocker` - (Optional) Represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running
- `gce_persistent_disk` - (Optional) Represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin. For more info see Kubernetes reference
- `git_repo` - (Optional) GitRepo represents a git repository at a particular revision.
- `glusterfs` - (Optional) Represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md
- `host_path` - (Optional) Represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. For more info see Kubernetes reference
- `iscsi` - (Optional) Represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.

- `name` - (Optional) Volume's name. Must be a DNS_LABEL and unique within the pod. For more info see Kubernetes reference
- `nfs` - (Optional) Represents an NFS mount on the host. Provisioned by an admin. For more info see Kubernetes reference
- `persistent_volume_claim` - (Optional) The specification of a persistent volume.
- `photon_persistent_disk` - (Optional) Represents a PhotonController persistent disk attached and mounted on kubelets host machine
- `quobyte` - (Optional) Quobyte represents a Quobyte mount on the host that shares a pod's lifetime
- `rbd` - (Optional) Represents a Rados Block Device mount on the host that shares a pod's lifetime. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md
- `secret` - (Optional) Secret represents a secret that should populate this volume. For more info see Kubernetes reference
- `vsphere_volume` - (Optional) Represents a vSphere volume attached and mounted on kubelets host machine

» `volume_mount`

» Arguments

- `mount_path` - (Required) Path within the container at which the volume should be mounted. Must not contain ':'.
- `name` - (Required) This must match the Name of a Volume.
- `read_only` - (Optional) Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
- `sub_path` - (Optional) Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

» `vsphere_volume`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `volume_path` - (Required) Path that identifies vSphere volume vmdk

» Timeouts

The following Timeout configuration options are available for the `kubernetes_daemonset` resource:

- `create` - (Default `10 minutes`) Used for creating new controller

- `update` - (Default `10 minutes`) Used for updating a controller
- `delete` - (Default `10 minutes`) Used for destroying a controller

» Import

DaemonSet can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_daemonset.example default/terraform-example
```

» kubernetes_deployment

A Deployment ensures that a specified number of pod "replicas" are running at any one time. In other words, a Deployment makes sure that a pod or homogeneous set of pods are always up and available. If there are too many pods, it will kill some. If there are too few, the Deployment will start more.

» Example Usage

```
resource "kubernetes_deployment" "example" {
  metadata {
    name = "terraform-example"
    labels = {
      test = "MyExampleApp"
    }
  }

  spec {
    replicas = 3

    selector {
      match_labels = {
        test = "MyExampleApp"
      }
    }

    template {
      metadata {
        labels = {
          test = "MyExampleApp"
        }
      }

      spec {
```

```
        container {
          image = "nginx:1.7.8"
          name  = "example"

          resources {
            limits {
              cpu    = "0.5"
              memory = "512Mi"
            }
            requests {
              cpu    = "250m"
              memory = "50Mi"
            }
          }

          liveness_probe {
            http_get {
              path = "/nginx_status"
              port = 80

              http_header {
                name  = "X-Custom-Header"
                value = "Awesome"
              }
            }

            initial_delay_seconds = 3
            period_seconds        = 3
          }
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard deployment's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the specification of the desired behavior of the deployment. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the deployment that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the deployment. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). Must match `selector`**. For more info see Kubernetes reference
- `name` - (Optional) Name of the deployment, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the deployment must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this deployment that can be used by clients to determine when deployment has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this deployment.
- `uid` - The unique in time and space value for this deployment. For more info see Kubernetes reference

» `spec`

» **Arguments**

- `min_ready_seconds` - (Optional) Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
- `paused` - (Optional) Indicates that the deployment is paused.
- `progress_deadline_seconds` - (Optional) The maximum time in seconds for a deployment to make progress before it is considered to be failed. The deployment controller will continue to process failed deployments and a condition with a ProgressDeadlineExceeded reason will be surfaced in the deployment status. Note that progress will not be estimated during the time a deployment is paused. Defaults to 600s.
- `replicas` - (Optional) The number of desired replicas. Defaults to 1. For more info see Kubernetes reference
- `revision_history_limit` - (Optional) The number of old ReplicaSets to retain to allow rollback. This is a pointer to distinguish between explicit zero and not specified. Defaults to 10.
- `strategy` - (Optional) The deployment strategy to use to replace existing pods with new ones.
- `selector` - (Optional) A label query over pods that should match the Replicas count. Label keys and values that must match in order to be controlled by this deployment. **Must match labels (`metadata.0.labels`)**. For more info see Kubernetes reference
- `template` - (Required) Describes the pod that will be created if insufficient replicas are detected. This takes precedence over a TemplateRef. For more info see Kubernetes reference

» `strategy`

» **Arguments**

- `type` - Type of deployment. Can be 'Recreate' or 'RollingUpdate'. Default is RollingUpdate.
- `rolling_update` - Rolling update config params. Present only if type = RollingUpdate.

» `rolling_update`

» **Arguments**

- `max_surge` - The maximum number of pods that can be scheduled above the desired number of pods. Value can be an absolute number (ex: 5) or

a percentage of desired pods (ex: 10%). This can not be 0 if MaxUnavailable is 0. Absolute number is calculated from percentage by rounding up. Defaults to 25%. Example: when this is set to 30%, the new RC can be scaled up immediately when the rolling update starts, such that the total number of old and new pods do not exceed 130% of desired pods. Once old pods have been killed, new RC can be scaled up further, ensuring that total number of pods running at any time during the update is atmost 130% of desired pods.

- `max_unavailable` - The maximum number of pods that can be unavailable during the update. Value can be an absolute number (ex: 5) or a percentage of desired pods (ex: 10%). Absolute number is calculated from percentage by rounding down. This can not be 0 if MaxSurge is 0. Defaults to 25%. Example: when this is set to 30%, the old RC can be scaled down to 70% of desired pods immediately when the rolling update starts. Once new pods are ready, old RC can be scaled down further, followed by scaling up the new RC, ensuring that the total number of pods available at all times during the update is at least 70% of desired pods.

» `template`

» Arguments

- `metadata` - (Required) Standard object's metadata. For more info see https://git.k8s.io/community/contributors/devel/api-conventions.md#metadata
- `spec` - (Required) Specification of the desired behavior of the pod. For more info see https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

» template `spec`

» Arguments

- `affinity` - (Optional) A group of affinity scheduling rules. If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
- `active_deadline_seconds` - (Optional) Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
- `automount_service_account_token` - (Optional) Indicates whether a service account token should be automatically mounted.
- `container` - (Optional) List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one

container in a Pod. Cannot be updated. For more info see Kubernetes reference

- `init_container` - (Optional) List of init containers belonging to the pod. Init containers always run to completion and each must complete successfully before the next is started. For more info see Kubernetes reference/
- `dns_policy` - (Optional) Set DNS policy for containers within the pod. Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'. Optional: Defaults to 'ClusterFirst', see Kubernetes reference.
- `dns_config` - (Optional) Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy. Defaults to empty. See `dns_config` block definition below.
- `host_alias` - (Optional) List of hosts and IPs that will be injected into the pod's hosts file if specified. Optional: Defaults to empty. See `host_alias` block definition below.
- `host_ipc` - (Optional) Use the host's ipc namespace. Optional: Defaults to false.
- `host_network` - (Optional) Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified.
- `host_pid` - (Optional) Use the host's pid namespace.
- `hostname` - (Optional) Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.
- `image_pull_secrets` - (Optional) ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. For example, in the case of docker, only DockerConfig type secrets are honored. For more info see Kubernetes reference
- `node_name` - (Optional) NodeName is a request to schedule this pod onto a specific node. If it is non-empty, the scheduler simply schedules this pod onto that node, assuming that it fits resource requirements.
- `node_selector` - (Optional) NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. For more info see Kubernetes reference.
- `restart_policy` - (Optional) Restart policy for all containers within the pod. One of Always, OnFailure, Never. For more info see Kubernetes reference.
- `security_context` - (Optional) SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty
- `service_account_name` - (Optional) ServiceAccountName is the name

of the ServiceAccount to use to run this pod. For more info see http://releases.k8s.io/HEAD/docs/design/service_accounts.md.

- `share_process_namespace` - (Optional) Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set.
- `subdomain` - (Optional) If specified, the fully qualified Pod hostname will be "...svc.". If not specified, the pod will not have a domainname at all..
- `termination_grace_period_seconds` - (Optional) Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates delete immediately. If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process.
- `toleration` - (Optional) Optional pod node tolerations. For more info see Kubernetes reference
- `volume` - (Optional) List of volumes that can be mounted by containers belonging to the pod. For more info see Kubernetes reference

» `affinity`

» Arguments

- `node_affinity` - (Optional) Node affinity scheduling rules for the pod. For more info see Kubernetes reference
- `pod_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference
- `pod_anti_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference

» `node_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point

during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.

- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution`

» **Arguments**

- `node_selector_term` - (Required) A list of node selector terms. The terms are ORed.

» `node_selector_term`

» **Arguments**

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» **Arguments**

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `preferred_during_scheduling_ignored_during_execution`

» **Arguments**

- `preference` - (Required) A node selector term, associated with the corresponding weight.

- `weight` - (Required) Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

» `preference`

» Arguments

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» Arguments

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `pod_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `pod_anti_affinity`

» **Arguments**

- `required_during_scheduling_ignored_during_execution` - (Optional) If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution` **(pod_affinity_term)**

» **Arguments**

- `label_selector` - (Optional) A label query over a set of resources, in this case pods.
- `namespaces` - (Optional) Specifies which namespaces the `label_selector` applies to (matches against). Null or empty list means "this pod's namespace"
- `topology_key` - (Optional) This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the `label_selector` in the specified namespaces, where co-located is defined as running on a node whose value of the label with key `topology_key` matches that of any node on which any of the selected pods is running. Empty `topology_key` is not allowed.

» `preferred_during_scheduling_ignored_during_execution`

» **Arguments**

- `pod_affinity_term` - (Required) A pod affinity term, associated with the corresponding weight.
- `weight` - (Required) Weight associated with matching the corresponding `pod_affinity_term`, in the range 1-100.

» `container`

» **Arguments**

- `args` - (Optional) Arguments to the entrypoint. The docker image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference
- `command` - (Optional) Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference
- `env` - (Optional) Block of string name and value pairs to set in the container's environment. May be declared multiple times. Cannot be updated.
- `env_from` - (Optional) List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
- `image` - (Optional) Docker image name. For more info see Kubernetes reference
- `image_pull_policy` - (Optional) Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. For more info see Kubernetes reference
- `lifecycle` - (Optional) Actions that the management system should take in response to container lifecycle events
- `liveness_probe` - (Optional) Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. For more info see Kubernetes reference
- `name` - (Required) Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
- `port` - (Optional) List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.
- `readiness_probe` - (Optional) Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. For more info see Kubernetes reference

- `resources` - (Optional) Compute Resources required by this container. Cannot be updated. For more info see Kubernetes reference
- `security_context` - (Optional) Security options the pod should run with. For more info see http://releases.k8s.io/HEAD/docs/design/security_context.md
- `stdin` - (Optional) Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF.
- `stdin_once` - (Optional) Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF.
- `termination_message_path` - (Optional) Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Defaults to /dev/termination-log. Cannot be updated.
- `tty` - (Optional) Whether this container should allocate a TTY for itself
- `volume_mount` - (Optional) Pod volumes to mount into the container's filesystem. Cannot be updated.
- `working_dir` - (Optional) Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

» `aws_elastic_block_store`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
- `read_only` - (Optional) Whether to set the read-only property in VolumeMounts to "true". If omitted, the default is "false". For more info see Kubernetes reference
- `volume_id` - (Required) Unique ID of the persistent disk resource in AWS (Amazon EBS volume). For more info see Kubernetes reference

» `azure_disk`

» **Arguments**

- `caching_mode` - (Required) Host Caching mode: None, Read Only, Read Write.
- `data_disk_uri` - (Required) The URI the data disk in the blob storage
- `disk_name` - (Required) The Name of the data disk in the blob storage
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).

» `azure_file`

» **Arguments**

- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `secret_name` - (Required) The name of secret that contains Azure Storage Account Name and Key
- `share_name` - (Required) Share Name

» `capabilities`

» **Arguments**

- `add` - (Optional) Added capabilities
- `drop` - (Optional) Removed capabilities

» `ceph_fs`

» **Arguments**

- `monitors` - (Required) Monitors is a collection of Ceph monitors For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `path` - (Optional) Used as the mounted root, rather than the full Ceph tree, default is /
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to `false` (read/write). For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

- `secret_file` - (Optional) The path to key ring for User, default is /etc/ceph/user.secret For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_ref` - (Optional) Reference to the authentication secret for User, default is empty. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `user` - (Optional) User is the rados user name, default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

» `cinder`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write). For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `volume_id` - (Required) Volume ID used to identify the volume in Cinder. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

» `config_map`

» Arguments

- `default_mode` - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `config_map_ref`

» **Arguments**

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the ConfigMap must be defined

» `config_map_key_ref`

» **Arguments**

- `key` - (Optional) The key to select.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `dns_config`

» **Arguments**

- `nameservers` - (Optional) A list of DNS name server IP addresses specified as strings. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed. Optional: Defaults to empty.
- `option` - (Optional) A list of DNS resolver options specified as blocks with `name`/`value` pairs. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy. Optional: Defaults to empty.
- `searches` - (Optional) A list of DNS search domains for host-name lookup specified as strings. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed. Optional: Defaults to empty.

The `option` block supports the following:

- `name` - (Required) Name of the option.
- `value` - (Optional) Value of the option. Optional: Defaults to empty.

» `downward_api`

» **Arguments**

- `default_mode` - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict

with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.

» `empty_dir`

» **Arguments**

- `medium` - (Optional) What type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. For more info see Kubernetes reference

» `env`

» **Arguments**

- `name` - (Required) Name of the environment variable. Must be a C_IDENTIFIER
- `value` - (Optional) Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
- `value_from` - (Optional) Source for the environment variable's value

» `env_from`

» **Arguments**

- `config_map_ref` - (Optional) The ConfigMap to select from
- `prefix` - (Optional) An optional identifer to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER..
- `secret_ref` - (Optional) The Secret to select from

» `exec`

» **Arguments**

- `command` - (Optional) Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

» `fc`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `lun` - (Required) FC target lun number
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `target_ww_ns` - (Required) FC target worldwide names (WWNs)

» `field_ref`

» **Arguments**

- `api_version` - (Optional) Version of the schema the FieldPath is written in terms of, defaults to "v1".
- `field_path` - (Optional) Path of the field to select in the specified API version

» `flex_volume`

» **Arguments**

- `driver` - (Required) Driver is the name of the driver to use for this volume.
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
- `options` - (Optional) Extra command options if any.
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false (read/write).
- `secret_ref` - (Optional) Reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if

no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

» `flocker`

» Arguments

- `dataset_name` - (Optional) Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
- `dataset_uuid` - (Optional) UUID of the dataset. This is unique identifier of a Flocker dataset

» `gce_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). For more info see Kubernetes reference
- `pd_name` - (Required) Unique name of the PD resource in GCE. Used to identify the disk in GCE. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false. For more info see Kubernetes reference

» `git_repo`

» Arguments

- `directory` - (Optional) Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
- `repository` - (Optional) Repository URL
- `revision` - (Optional) Commit hash for the specified revision.

» `glusterfs`

» Arguments

- `endpoints_name` - (Required) The endpoint name that details Glusterfs topology. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `path` - (Required) The Glusterfs volume path. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `read_only` - (Optional) Whether to force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

» `host_alias`

» Arguments

- `hostnames` - (Required) Hostnames for the IP address.
- `ip` - (Required) IP address of the host file entry.

» `host_path`

» Arguments

- `path` - (Optional) Path of the directory on the host. For more info see Kubernetes reference
- `type` - (Optional) Type for HostPath volume. Defaults to "". For more info see Kubernetes reference

» `http_get`

» Arguments

- `host` - (Optional) Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
- `http_header` - (Optional) Scheme to use for connecting to the host.
- `path` - (Optional) Path to access on the HTTP server.
- `port` - (Optional) Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
- `scheme` - (Optional) Scheme to use for connecting to the host.

» `http_header`

» Arguments

- `name` - (Optional) The header field name

- `value` - (Optional) The header field value

» `image_pull_secrets`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference

» `iscsi`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `iqn` - (Required) Target iSCSI Qualified Name.
- `iscsi_interface` - (Optional) iSCSI interface name that uses an iSCSI transport. Defaults to 'default' (tcp).
- `lun` - (Optional) iSCSI target lun number.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false.
- `target_portal` - (Required) iSCSI target portal. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

» `items`

» Arguments

- `key` - (Optional) The key to project.
- `mode` - (Optional) Optional: mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `path` - (Optional) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `lifecycle`

» Arguments

- `post_start` - (Optional) post_start is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. For more info see Kubernetes reference
- `pre_stop` - (Optional) pre_stop is called immediately before a container is terminated. The container is terminated after the handler completes. The reason for termination is passed to the handler. Regardless of the outcome of the handler, the container is eventually terminated. Other management of the container blocks until the hook completes. For more info see Kubernetes reference

» `limits`

» Arguments

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `liveness_probe`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `nfs`

» Arguments

- `path` - (Required) Path that is exported by the NFS server. For more info see Kubernetes reference

- `read_only` - (Optional) Whether to force the NFS export to be mounted with read-only permissions. Defaults to false. For more info see Kubernetes reference
- `server` - (Required) Server is the hostname or IP address of the NFS server. For more info see Kubernetes reference

» `persistent_volume_claim`

» Arguments

- `claim_name` - (Optional) ClaimName is the name of a PersistentVolumeClaim in the same
- `read_only` - (Optional) Will force the ReadOnly setting in VolumeMounts.

» `photon_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `pd_id` - (Required) ID that identifies Photon Controller persistent disk

» `port`

» Arguments

- `container_port` - (Required) Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
- `host_ip` - (Optional) What host IP to bind the external port to.
- `host_port` - (Optional) Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
- `name` - (Optional) If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services
- `protocol` - (Optional) Protocol for port. Must be UDP or TCP. Defaults to "TCP".

» `post_start`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `pre_stop`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `quobyte`

» **Arguments**

- `group` - (Optional) Group to map volume access to Default is no group
- `read_only` - (Optional) Whether to force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
- `registry` - (Required) Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
- `user` - (Optional) User to map volume access to Defaults to serivceaccount user
- `volume` - (Required) Volume is a string that references an already created Quobyte volume by name.

» `rbd`

» **Arguments**

- `ceph_monitors` - (Required) A collection of Ceph monitors. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see Kubernetes reference
- `keyring` - (Optional) Keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

- `rados_user` - (Optional) The rados user name. Default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_image` - (Required) The rados image name. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_pool` - (Optional) The rados pool name. Default is rbd. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `secret_ref` - (Optional) Name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

» `readiness_probe`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `resources`

» Arguments

- `limits` - (Optional) Describes the maximum amount of compute resources allowed. For more info see Kubernetes reference/
- `requests` - (Optional) Describes the minimum amount of compute resources required.

» `requests`

» **Arguments**

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `resource_field_ref`

» **Arguments**

- `container_name` - (Optional) The name of the container
- `resource` - (Required) Resource to select

» `se_linux_options`

» **Arguments**

- `level` - (Optional) Level is SELinux level label that applies to the container.
- `role` - (Optional) Role is a SELinux role label that applies to the container.
- `type` - (Optional) Type is a SELinux type label that applies to the container.
- `user` - (Optional) User is a SELinux user label that applies to the container.

» `secret`

» **Arguments**

- `default_mode` - (Optional) Mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) List of Secret Items to project into the volume. See `items` block definition below. If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked `optional`. Paths must be relative and may not contain the '..' path or start with '..'.
- `optional` - (Optional) Specify whether the Secret or it's keys must be defined.

- `secret_name` - (Optional) Name of the secret in the pod's namespace to use. For more info see Kubernetes reference

The `items` block supports the following:

- `key` - (Required) The key to project.
- `mode` - (Optional) Mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used.
- `path` - (Required) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `secret_ref`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the Secret must be defined

» `secret_key_ref`

» Arguments

- `key` - (Optional) The key of the secret to select from. Must be a valid secret key.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `secret_ref`

» Arguments

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» container `security_context`

» Arguments

- `allow_privilege_escalation` - (Optional) AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN

- `capabilities` - (Optional) The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
- `privileged` - (Optional) Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.
- `read_only_root_filesystem` - (Optional) Whether this container has a read-only root filesystem. Default is false.
- `run_as_group` - (Optional) The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `se_linux_options` - (Optional) The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

» `capabilities`

» Arguments

- `add` - (Optional) A list of added capabilities.
- `drop` - (Optional) A list of removed capabilities.

» pod `security_context`

» Arguments

- `fs_group` - (Optional) A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd

with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume.

- `run_as_group` - (Optional) The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `se_linux_options` - (Optional) The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `supplemental_groups` - (Optional) A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.

» `tcp_socket`

» Arguments

- `port` - (Required) Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

» `value_from`

» Arguments

- `config_map_key_ref` - (Optional) Selects a key of a ConfigMap.
- `field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..
- `resource_field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, meta-

data.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..
- `secret_key_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..

» `toleration`

» Arguments

- `effect` - (Optional) Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
- `key` - (Optional) Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
- `operator` - (Optional) Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
- `toleration_seconds` - (Optional) TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
- `value` - (Optional) Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

» `volume`

» Arguments

- `aws_elastic_block_store` - (Optional) Represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. For more info see Kubernetes reference
- `azure_disk` - (Optional) Represents an Azure Data Disk mount on the host and bind mount to the pod.
- `azure_file` - (Optional) Represents an Azure File Service mount on the host and bind mount to the pod.
- `ceph_fs` - (Optional) Represents a Ceph FS mount on the host that shares a pod's lifetime
- `cinder` - (Optional) Represents a cinder volume attached and mounted on kubelets host machine. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

- `config_map` - (Optional) ConfigMap represents a configMap that should populate this volume
- `downward_api` - (Optional) DownwardAPI represents downward API about the pod that should populate this volume
- `empty_dir` - (Optional) EmptyDir represents a temporary directory that shares a pod's lifetime. For more info see Kubernetes reference
- `fc` - (Optional) Represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
- `flex_volume` - (Optional) Represents a generic volume resource that is provisioned/attached using an exec based plugin. This is an alpha feature and may change in future.
- `flocker` - (Optional) Represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running
- `gce_persistent_disk` - (Optional) Represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin. For more info see Kubernetes reference
- `git_repo` - (Optional) GitRepo represents a git repository at a particular revision.
- `glusterfs` - (Optional) Represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md
- `host_path` - (Optional) Represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. For more info see Kubernetes reference
- `iscsi` - (Optional) Represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.
- `name` - (Optional) Volume's name. Must be a DNS_LABEL and unique within the pod. For more info see Kubernetes reference
- `nfs` - (Optional) Represents an NFS mount on the host. Provisioned by an admin. For more info see Kubernetes reference
- `persistent_volume_claim` - (Optional) The specification of a persistent volume.
- `photon_persistent_disk` - (Optional) Represents a PhotonController persistent disk attached and mounted on kubelets host machine
- `quobyte` - (Optional) Quobyte represents a Quobyte mount on the host that shares a pod's lifetime
- `rbd` - (Optional) Represents a Rados Block Device mount on the host that shares a pod's lifetime. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md
- `secret` - (Optional) Secret represents a secret that should populate this volume. For more info see Kubernetes reference
- `vsphere_volume` - (Optional) Represents a vSphere volume attached and

mounted on kubelets host machine

» `volume_mount`

» Arguments

- `mount_path` - (Required) Path within the container at which the volume should be mounted. Must not contain ':'.
- `name` - (Required) This must match the Name of a Volume.
- `read_only` - (Optional) Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
- `sub_path` - (Optional) Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

» `vsphere_volume`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `volume_path` - (Required) Path that identifies vSphere volume vmdk

» Timeouts

The following Timeout configuration options are available for the `kubernetes_deployment` resource:

- `create` - (Default `10 minutes`) Used for creating new controller
- `update` - (Default `10 minutes`) Used for updating a controller
- `delete` - (Default `10 minutes`) Used for destroying a controller

» Import

Deployment can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_deployment.example default/terraform-example
```

» kubernetes_endpoints

An Endpoints resource is an abstraction, linked to a Service, which defines the list of endpoints that actually implement the service.

» Example Usage

```
resource "kubernetes_endpoints" "example" {
  metadata {
    name = "terraform-example"
  }

  subset {
    address {
      ip = "10.0.0.4"
    }

    address {
      ip = "10.0.0.5"
    }

    port {
      name     = "http"
      port     = 80
      protocol = "TCP"
    }

    port {
      name     = "https"
      port     = 443
      protocol = "TCP"
    }
  }

  subset {
    address {
      ip = "10.0.1.4"
    }

    address {
      ip = "10.0.1.5"
    }

    port {
      name     = "http"
      port     = 80
      protocol = "TCP"
    }

    port {
```

```
      name     = "https"
      port     = 443
      protocol = "TCP"
    }
  }
}

resource "kubernetes_service" "example" {
  metadata {
    name = "${kubernetes_endpoints.example.metadata.0.name}"
  }

  spec {
    port {
      port        = 8080
      target_port = 80
    }

    port {
      port        = 8443
      target_port = 443
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard endpoints' metadata. For more info see Kubernetes reference
- `subset` - (Optional) Set of addresses and ports that comprise a service. Can be repeated multiple times.

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the endpoints resource that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and conse-**

quently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the endpoints resource. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the endpoints resource, must be unique. Cannot be updated. This name should correspond with an accompanying Service resource. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the endpoints resource must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this endpoints resource that can be used by clients to determine when endpoints resource has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this endpoints resource.
- `uid` - The unique in time and space value for this endpoints resource. For more info see Kubernetes reference

» `subset`

» Arguments

- `address` - (Optional) An IP address block which offers the related ports and is ready to accept traffic. These endpoints should be considered safe for load balancers and clients to utilize. Can be repeated multiple times.

- `not_ready_address` - (Optional) A IP address block which offers the related ports but is not currently marked as ready because it have not yet finished starting, have recently failed a readiness check, or have recently failed a liveness check. Can be repeated multiple times.
- `port` - (Optional) A port number block available on the related IP addresses. Can be repeated multiple times.

» `address`

» Attributes

- `ip` - The IP of this endpoint. May not be loopback (127.0.0.0/8), link-local (169.254.0.0/16), or link-local multicast ((224.0.0.0/24).
- `hostname` - (Optional) The Hostname of this endpoint.
- `node_name` - (Optional) Node hosting this endpoint. This can be used to determine endpoints local to a node.

» `not_ready_address`

» Attributes

- `ip` - The IP of this endpoint. May not be loopback (127.0.0.0/8), link-local (169.254.0.0/16), or link-local multicast ((224.0.0.0/24).
- `hostname` - (Optional) The Hostname of this endpoint.
- `node_name` - (Optional) Node hosting this endpoint. This can be used to determine endpoints local to a node.

» `port`

» Arguments

- `name` - (Optional) The name of this port within the endpoint. All ports within the endpoint must have unique names. Optional if only one port is defined on this endpoint.
- `port` - (Required) The port that will be utilized by this endpoint.
- `protocol` - (Optional) The IP protocol for this port. Supports `TCP` and `UDP`. Default is `TCP`.

» Import

An Endpoints resource can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_endpoints.example default/terraform-name
```

» kubernetes_horizontal_pod_autoscaler

Horizontal Pod Autoscaler automatically scales the number of pods in a replication controller, deployment or replica set based on observed CPU utilization.

» Example Usage

```
resource "kubernetes_horizontal_pod_autoscaler" "example" {
  metadata {
    name = "terraform-example"
  }
  spec {
    max_replicas = 10
    min_replicas = 8
    scale_target_ref {
      kind = "ReplicationController"
      name = "MyApp"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard horizontal pod autoscaler's metadata. For more info see Kubernetes reference
- `spec` - (Required) Behaviour of the autoscaler. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the horizontal pod autoscaler that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource**

attributes and manage them as expected (while still avoiding the perpetual diff problem). For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the horizontal pod autoscaler. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the horizontal pod autoscaler, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the horizontal pod autoscaler must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this horizontal pod autoscaler that can be used by clients to determine when horizontal pod autoscaler has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this horizontal pod autoscaler.
- `uid` - The unique in time and space value for this horizontal pod autoscaler. For more info see Kubernetes reference

» `spec`

» Arguments

- `max_replicas` - (Required) Upper limit for the number of pods that can be set by the autoscaler.
- `min_replicas` - (Optional) Lower limit for the number of pods that can be set by the autoscaler, defaults to `1`.
- `scale_target_ref` - (Required) Reference to scaled resource. e.g. Replication Controller

- `target_cpu_utilization_percentage` - (Optional) Target average CPU utilization (represented as a percentage of requested CPU) over all the pods. If not specified the default autoscaling policy will be used.

» `scale_target_ref`

» Arguments

- `api_version` - (Optional) API version of the referent
- `kind` - (Required) Kind of the referent. e.g. `ReplicationController`. For more info see https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#types-kinds
- `name` - (Required) Name of the referent. For more info see Kubernetes reference

» Import

Horizontal Pod Autoscaler can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_horizontal_pod_autoscaler.example default/terraform-example
```

» kubernetes_ingress

Ingress is a collection of rules that allow inbound connections to reach the endpoints defined by a backend. An Ingress can be configured to give services externally-reachable urls, load balance traffic, terminate SSL, offer name based virtual hosting etc.

» Example Usage

```
resource "kubernetes_ingress" "example_ingress" {
  metadata {
    name = "example-ingress"
  }

  spec {
    backend {
      service_name = "MyApp1"
      service_port = 8080
    }

    rule {
      http {
```

```
        path {
          backend {
            service_name = "MyApp1"
            service_port = 8080
          }

          path = "/app1/*"
        }

        path {
          backend {
            service_name = "MyApp2"
            service_port = 8080
          }

          path = "/app2/*"
        }
      }
    }

    tls {
      secret_name = "tls-secret"
    }
  }
}

resource "kubernetes_pod" "example" {
  metadata {
    name = "terraform-example"
    labels = {
      app = "MyApp1"
    }
  }

  spec {
    container {
      image = "nginx:1.7.9"
      name  = "example"

      port {
        container_port = 8080
      }
    }
  }
}
```

```
resource "kubernetes_pod" "example2" {
  metadata {
    name = "terraform-example2"
    labels = {
      app = "MyApp2"
    }
  }

  spec {
    container {
      image = "nginx:1.7.9"
      name  = "example"

      port {
        container_port = 8080
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard ingress's metadata. More info: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#metadata
- `spec` - (Required) Spec defines the behavior of a ingress. https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#spec-and-status

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the ingress that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes**

and manage them as expected (while still avoiding the perpetual diff problem). More info: http://kubernetes.io/docs/user-guide/annotations

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. Read more: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#idempotency
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/labels
- `name` - (Optional) Name of the service, must be unique. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names
- `namespace` - (Optional) Namespace defines the space within which name of the service must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service that can be used by clients to determine when service has changed. Read more: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#concurrency-control-and-consistency
- `self_link` - A URL representing this service.
- `uid` - The unique in time and space value for this service. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

» `spec`

» Arguments

- `backend` - (Optional) Backend defines the referenced service endpoint to which the traffic will be forwarded. See `backend` block attributes below.

- `rule` - (Optional) A list of host rules used to configure the Ingress. If unspecified, or no rule matches, all traffic is sent to the default backend. See `rule` block attributes below.
- `tls` - (Optional) TLS configuration. Currently the Ingress only supports a single TLS port, 443. If multiple members of this list specify different hosts, they will be multiplexed on the same port according to the hostname specified through the SNI TLS extension, if the ingress controller fulfilling the ingress supports SNI. See `tls` block attributes below.

» `backend`

» Arguments

- `service_name` - (Optional) Specifies the name of the referenced service.
- `service_port` - (Optional) Specifies the port of the referenced service.

» `rule`

» Arguments

- `host` - (Optional) Host is the fully qualified domain name of a network host, as defined by RFC 3986. Note the following deviations from the \"host\" part of the URI as defined in the RFC: 1. IPs are not allowed. Currently an IngressRuleValue can only apply to the IP in the Spec of the parent Ingress. 2. The : delimiter is not respected because ports are not allowed. Currently the port of an Ingress is implicitly :80 for http and :443 for https. Both these may change in the future. Incoming requests are matched against the host before the IngressRuleValue. If the host is unspecified, the Ingress routes all traffic based on the specified IngressRuleValue.
- `http` - (Required) http is a list of http selectors pointing to backends. In the example: http:///? -> backend where where parts of the url correspond to RFC 3986, this resource will be used to match against everything after the last '/' and before the first '?' or '#'. See `http` block attributes below.

» `http`

- `path` - (Required) Path array of path regex associated with a backend. Incoming urls matching the path are forwarded to the backend, see below for `path` block structure.

» `path`

- `path` - (Required) A string or an extended POSIX regular expression as defined by IEEE Std 1003.1, (i.e this follows the egrep/unix syntax, not the perl syntax) matched against the path of an incoming request. Currently it can contain characters disallowed from the conventional \"path\" part of a URL as defined by RFC 3986. Paths must begin with a '/'. If unspecified, the path defaults to a catch all sending traffic to the backend.
- `backend` - (Required) Backend defines the referenced service endpoint to which the traffic will be forwarded to.

» `tls`

» Arguments

- `hosts` - (Optional) Hosts are a list of hosts included in the TLS certificate. The values in this list must match the name/s used in the tlsSecret. Defaults to the wildcard host setting for the loadbalancer controller fulfilling this Ingress, if left unspecified.
- `secret_name` - (Optional) SecretName is the name of the secret used to terminate SSL traffic on 443. Field is left optional to allow SSL routing based on SNI hostname alone. If the SNI host in a listener conflicts with the \"Host\" header field used by an IngressRule, the SNI host is used for termination and value of the Host header is used for routing.

» Attributes

- `load_balancer_ingress` - A list containing ingress points for the load-balancer

» `load_balancer_ingress`

» Attributes

- `ip` - IP which is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)
- `hostname` - Hostname which is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)

» Import

Ingress can be imported using its namespace and name:

```
terraform import kubernetes_ingress.<TERRAFORM_RESOURCE_NAME> <KUBE_NAMESPACE>/<KUBE_INGRESS
```

e.g. `$ terraform import kubernetes_ingress.example default/terraform-name`

» kubernetes_job

A Job creates one or more Pods and ensures that a specified number of them successfully terminate. As pods successfully complete, the Job tracks the successful completions. When a specified number of successful completions is reached, the task (ie, Job) is complete. Deleting a Job will clean up the Pods it created.

A simple case is to create one Job object in order to reliably run one Pod to completion. The Job object will start a new Pod if the first Pod fails or is deleted (for example due to a node hardware failure or a node reboot.

You can also use a Job to run multiple Pods in parallel.

» Example Usage

```
resource "kubernetes_job" "demo" {
  metadata {
    name = "demo"
  }
  spec {
    template {
      metadata {}
      spec {
        container {
          name    = "pi"
          image   = "perl"
          command = ["perl", "-Mbignum=bpi", "-wle", "print bpi(2000)"]
        }
        restart_policy = "Never"
      }
    }
    backoff_limit = 4
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard resource's metadata. More info: https://github.com/kubernetes/community/blob/master/contributors/devel/sig-architecture/api-conventions.md#spec-and-status
- `spec` - (Required) Specification of the desired behavior of a job. More info: https://git.k8s.io/community/contributors/devel/api-conventions.md#spec-and-status

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the resource that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/annotations
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. Read more: https://github.com/kubernetes/community/blob/master/contributors/devel/api-conventions.md#idempotency
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/labels
- `name` - (Optional) Name of the service, must be unique. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names
- `namespace` - (Optional) Namespace defines the space within which name of the service must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service that can be used by clients to determine when service has changed. Read more: https://github.com/kubernetes/community/blob/

master/contributors/devel/api-conventions.md#concurrency-control-and-consistency
- `self_link` - A URL representing this service.
- `uid` - The unique in time and space value for this service. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

» `spec`

» Arguments

- `active_deadline_seconds` - (Optional) Specifies the duration in seconds relative to the startTime that the job may be active before the system tries to terminate it; value must be positive integer.
- `backoff_limit` - (Optional) Specifies the number of retries before marking this job failed. Defaults to 6
- `completions` - (Optional) Specifies the desired number of successfully finished pods the job should be run with. Setting to nil means that the success of any pod signals the success of all pods, and allows parallelism to have any positive value. Setting to 1 means that parallelism is limited to 1 and the success of that pod signals the success of the job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
- `manual_selector` - (Optional) Controls generation of pod labels and pod selectors. Leave `manualSelector` unset unless you are certain what you are doing. When false or unset, the system pick labels unique to this job and appends those labels to the pod template. When true, the user is responsible for picking unique labels and specifying the selector. Failure to pick a unique label may cause this and other jobs to not function correctly. However, You may see `manualSelector=true` in jobs that were created with the old `extensions/v1beta1` API. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/#specifying-your-own-pod-selector
- `parallelism` - (Optional) Specifies the maximum desired number of pods the job should run at any given time. The actual number of pods running in steady state will be less than this number when `((.spec.completions - .status.successful) < .spec.parallelism)`, i.e. when the work left to do is less than max parallelism. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/
- `selector` - (Optional) A label query over pods that should match the pod count. Normally, the system sets this field for you. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/#label-selectors
- `template` - (Optional) Describes the pod that will be created when executing a job. More info: https://kubernetes.io/docs/concepts/workloads/controllers/jobs-run-to-completion/

» `selector`

» Arguments

- `match_expressions` - (Optional) A list of label selector requirements. The requirements are ANDed.
- `match_labels` - (Optional) A map of `{key,value}` pairs. A single `{key,value}` in the matchLabels map is equivalent to an element of matchExpressions, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

» `template`

» Arguments

These arguments are the same as the for the `spec` block of a Pod.

Please see the Pod resource for reference.

» kubernetes_limit_range

Limit Range sets resource usage limits (e.g. memory, cpu, storage) for supported kinds of resources in a namespace.

Read more in the official docs.

» Example Usage

```
resource "kubernetes_limit_range" "example" {
  metadata {
    name = "terraform-example"
  }
  spec {
    limit {
      type = "Pod"
      max = {
        cpu    = "200m"
        memory = "1024M"
      }
    }
    limit {
      type = "PersistentVolumeClaim"
      min = {
        storage = "24M"
```

```
      }
    }
    limit {
      type = "Container"
      default = {
        cpu    = "50m"
        memory = "24M"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard limit range's metadata. For more info see Kubernetes reference
- `spec` - (Optional) Spec defines the limits enforced. For more info see Kubernetes reference

» Nested Blocks

» `spec`

» Arguments

- `limit` - (Optional) The list of limits that are enforced.

» `limit`

» Arguments

- `default` - (Optional) Default resource requirement limit value by resource name if resource limit is omitted.
- `default_request` - (Optional) The default resource requirement request value by resource name if resource request is omitted.
- `max` - (Optional) Max usage constraints on this kind by resource name.
- `max_limit_request_ratio` - (Optional) The named resource must have a request and limit that are both non-zero where limit divided by request is less than or equal to the enumerated value; this represents the max burst for the named resource.
- `min` - (Optional) Min usage constraints on this kind by resource name.
- `type` - (Optional) Type of resource that this limit applies to. e.g. `Pod`, `Container` or `PersistentVolumeClaim`

» `metadata`

» **Arguments**

- `annotations` - (Optional) An unstructured key value map stored with the limit range that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the limit range. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the limit range, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the limit range must be unique.

» **Attributes**

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this limit range that can be used by clients to determine when limit range has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this limit range.
- `uid` - The unique in time and space value for this limit range. For more info see Kubernetes reference

» Import

Limit Range can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_limit_range.example default/terraform-example
```

» kubernetes_namespace

Kubernetes supports multiple virtual clusters backed by the same physical cluster. These virtual clusters are called namespaces. Read more about namespaces at Kubernetes reference/

» Example Usage

```
resource "kubernetes_namespace" "example" {
  metadata {
    annotations = {
      name = "example-annotation"
    }

    labels = {
      mylabel = "label-value"
    }

    name = "terraform-example-namespace"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard namespace's metadata.

» Timeouts

`kubernetes_namespace` provides the following Timeouts configuration options:

- `delete` - Default `5 minutes`

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the namespace that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. Read more about name idempotency.
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) namespaces. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the namespace, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this namespace that can be used by clients to determine when namespaces have changed. Read more about concurrency control and consistency.
- `self_link` - A URL representing this namespace.
- `uid` - The unique in time and space value for this namespace. For more info see Kubernetes reference

» Import

Namespaces can be imported using their name, e.g.

```
$ terraform import kubernetes_namespace.n terraform-example-namespace
```

» kubernetes_network_policy

Kubernetes supports network policies to specificy of how groups of pods are allowed to communicate with each other and other network endpoints. NetworkPolicy resources use labels to select pods and define rules which specify what traffic is allowed to the selected pods. Read more about network policies at https://kubernetes.io/docs/concepts/services-networking/network-policies/

» Example Usage

```
resource "kubernetes_network_policy" "example" {
  metadata {
    name      = "terraform-example-network-policy"
    namespace = "default"
  }

  spec {
    pod_selector {
      match_expressions {
        key      = "name"
        operator = "In"
        values   = ["webfront", "api"]
      }
    }

    ingress {
      ports {
        port     = "http"
        protocol = "TCP"
      }
      ports {
        port     = "8125"
        protocol = "UDP"
      }

      from {
        namespace_selector {
          match_labels = {
```

```
            name = "default"
          }
        }
      }

      from {
        ip_block {
          cidr = "10.0.0.0/8"
          except = [
            "10.0.0.0/24",
            "10.0.1.0/24",
          ]
        }
      }
    }

    egress {} # single empty rule to allow all egress traffic

    policy_types = ["Ingress", "Egress"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard network policy's metadata.

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the network policy that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/annotations

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. Read more about name idempotency.
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) network policies. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** More info: http://kubernetes.io/docs/user-guide/labels
- `name` - (Optional) Name of the network policy, must be unique. Cannot be updated. More info: http://kubernetes.io/docs/user-guide/identifiers#names

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this network policy that can be used by clients to determine when network policies have changed. Read more about concurrency control and consistency.
- `self_link` - A URL representing this network policy.
- `uid` - The unique in time and space value for this network policy. More info: http://kubernetes.io/docs/user-guide/identifiers#uids

» `spec`

» Arguments

- `egress` - (Optional) List of egress rules to be applied to the selected pods. Outgoing traffic is allowed if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic matches at least one egress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy limits all outgoing traffic (and serves solely to ensure that the pods it selects are isolated by default). This field is beta-level in 1.8
- `ingress` - (Optional) List of ingress rules to be applied to the selected pods. Traffic is allowed to a pod if there are no NetworkPolicies selecting the pod (and cluster policy otherwise allows the traffic), OR if the traffic

source is the pod's local node, OR if the traffic matches at least one ingress rule across all of the NetworkPolicy objects whose podSelector matches the pod. If this field is empty then this NetworkPolicy does not allow any traffic (and serves solely to ensure that the pods it selects are isolated by default).

- `pod_selector` - (Required) Selects the pods to which this NetworkPolicy object applies. The array of ingress rules is applied to any pods selected by this field. Multiple network policies can select the same set of pods. In this case, the ingress rules for each are combined additively. This field is NOT optional and follows standard label selector semantics. An empty podSelector matches all pods in this namespace.
- `policy_types` (Required) List of rule types that the NetworkPolicy relates to. Valid options are `Ingress`, `Egress`, or `Ingress,Egress`. This field is beta-level in 1.8 **Note**: the native Kubernetes API allows not to specify the `policy_types` property with the following description: > If this field is not specified, it will default based on the existence of Ingress or Egress rules; policies that contain an Egress section are assumed to affect Egress, and all policies (whether or not they contain an Ingress section) are assumed to affect Ingress. If you want to write an egress-only policy, you must explicitly specify policyTypes ["Egress"]. Likewise, if you want to write a policy that specifies that no egress is allowed, you must specify a policyTypes value that include "Egress" (since such a policy would not include an Egress section and would otherwise default to just ["Ingress"]).

Leaving the `policy_types` property optional here would have prevented an `egress` rule added to a Network Policy initially created without any `egress` rule nor `policy_types` from working as expected. Indeed, the PolicyTypes would have stuck to Ingress server side as the default value is only computed server side on resource creation, not on updates.

» `ingress`

» Arguments

- `from` - (Optional) List of sources which should be able to access the pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all sources (traffic not restricted by source). If this field is present and contains at least on item, this rule allows traffic only if the traffic matches at least one item in the from list.
- `ports` - (Optional) List of ports which should be made accessible on the pods selected for this rule. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least

one port in the list.

» `egress`

» **Arguments**

- `to` - (Optional) List of destinations for outgoing traffic of pods selected for this rule. Items in this list are combined using a logical OR operation. If this field is empty or missing, this rule matches all destinations (traffic not restricted by destination). If this field is present and contains at least one item, this rule allows traffic only if the traffic matches at least one item in the to list.
- `ports` - (Optional) List of destination ports for outgoing traffic. Each item in this list is combined using a logical OR. If this field is empty or missing, this rule matches all ports (traffic not restricted by port). If this field is present and contains at least one item, then this rule allows traffic only if the traffic matches at least one port in the list.

» `from`

» **Arguments**

- `namespace_selector` - (Optional) Selects Namespaces using cluster scoped-labels. This matches all pods in all namespaces selected by this label selector. This field follows standard label selector semantics. If present but empty, this selector selects all namespaces.
- `pod_selector` - (Optional) This is a label selector which selects Pods in this namespace. This field follows standard label selector semantics. If present but empty, this selector selects all pods in this namespace.

» `ports`

» **Arguments**

- `port` - (Optional) The port on the given protocol. This can either be a numerical or named port on a pod. If this field is not provided, this matches all port names and numbers.
- `protocol` - (Optional) The protocol (TCP or UDP) which traffic must match. If not specified, this field defaults to TCP.

» `to`

» **Arguments**

- `ip_block` - (Optional) IPBlock defines policy on a particular IPBlock

- `namespace_selector` - (Optional) Selects Namespaces using cluster scoped-labels. This matches all pods in all namespaces selected by this label selector. This field follows standard label selector semantics. If present but empty, this selector selects all namespaces.
- `pod_selector` - (Optional) This is a label selector which selects Pods in this namespace. This field follows standard label selector semantics. If present but empty, this selector selects all pods in this namespace.

» `ip_block`

» **Arguments**

- `cidr` - (Optional) CIDR is a string representing the IP Block Valid examples are "192.168.1.1/24"
- `except` - (Optional) Except is a slice of CIDRs that should not be included within an IP Block. Valid examples are "192.168.1.1/24". Except values will be rejected if they are outside the CIDR range.

» `namespace_selector`

» **Arguments**

- `match_expressions` - (Optional) A list of label selector requirements. The requirements are ANDed.
- `match_labels` - (Optional) A map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of `match_expressions`, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

» `pod_selector`

» **Arguments**

- `match_expressions` - (Optional) A list of label selector requirements. The requirements are ANDed.
- `match_labels` - (Optional) A map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of `match_expressions`, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

» `match_expressions`

» **Arguments**

- `key` - (Optional) The label key that the selector applies to.

- `operator` - (Optional) A key's relationship to a set of values. Valid operators ard `In`, `NotIn`, `Exists` and `DoesNotExist`.
- `values` - (Optional) An array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

» Import

Network policies can be imported using their identifier consisting of `<namespace-name>/<network-policy-name>`, e.g.:

```
$ terraform import kubernetes_network_policy.example default/terraform-example-network-polic
```

» kubernetes_persistent_volume

The resource provides a piece of networked storage in the cluster provisioned by an administrator. It is a resource in the cluster just like a node is a cluster resource. Persistent Volumes have a lifecycle independent of any individual pod that uses the PV.

For more info see Kubernetes reference/

» Example Usage

```
resource "kubernetes_persistent_volume" "example" {
  metadata {
    name = "terraform-example"
  }
  spec {
    capacity = {
      storage = "2Gi"
    }
    access_modes = ["ReadWriteMany"]
    persistent_volume_source {
      vsphere_volume {
        volume_path = "/absolute/path"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard persistent volume's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec of the persistent volume owned by the cluster. See below.

» Nested Blocks

» `spec`

» Arguments

- `access_modes` - (Required) Contains all ways the volume can be mounted. For more info see Kubernetes reference
- `capacity` - (Required) A description of the persistent volume's resources and capacity. For more info see Kubernetes reference
- `node_affinity` - (Optional) NodeAffinity defines constraints that limit what nodes this volume can be accessed from. This field influences the scheduling of pods that use this volume.
- `persistent_volume_reclaim_policy` - (Optional) What happens to a persistent volume when released from its claim. Valid options are Retain (default) and Recycle. Recycling must be supported by the volume plugin underlying this persistent volume. For more info see Kubernetes reference
- `persistent_volume_source` - (Required) The specification of a persistent volume.
- `storage_class_name` - (Optional) The name of the persistent volume's storage class. For more info see Kubernetes reference

» `node_affinity`

» Arguments

- `required` - (Optional) Required specifies hard node constraints that must be met.

» `required`

» Arguments

- `node_selector_term` - (Required) A list of node selector terms. The terms are ORed.

» node_selector_term

» Arguments

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» match_expressions and match_fields

» Arguments

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are `In`, `NotIn`, `Exists`, `DoesNotExist`. `Gt`, and `Lt`.
- `values` - (Optional) An array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. If the operator is `Gt` or `Lt`, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

» persistent_volume_source

» Arguments

- `aws_elastic_block_store` - (Optional) Represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. For more info see Kubernetes reference
- `azure_disk` - (Optional) Represents an Azure Data Disk mount on the host and bind mount to the pod.
- `azure_file` - (Optional) Represents an Azure File Service mount on the host and bind mount to the pod.
- `ceph_fs` - (Optional) Represents a Ceph FS mount on the host that shares a pod's lifetime
- `cinder` - (Optional) Represents a cinder volume attached and mounted on kubelets host machine. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `fc` - (Optional) Represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
- `flex_volume` - (Optional) Represents a generic volume resource that is provisioned/attached using an exec based plugin. This is an alpha feature and may change in future.
- `flocker` - (Optional) Represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running

- `gce_persistent_disk` - (Optional) Represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin. For more info see Kubernetes reference
- `glusterfs` - (Optional) Represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md
- `host_path` - (Optional) Represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. For more info see Kubernetes reference
- `iscsi` - (Optional) Represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.
- `local` - (Optional) Represents a local storage volume on the host. Provisioned by an admin. For more info see Kubernetes reference
- `nfs` - (Optional) Represents an NFS mount on the host. Provisioned by an admin. For more info see Kubernetes reference
- `photon_persistent_disk` - (Optional) Represents a PhotonController persistent disk attached and mounted on kubelets host machine
- `quobyte` - (Optional) Quobyte represents a Quobyte mount on the host that shares a pod's lifetime
- `rbd` - (Optional) Represents a Rados Block Device mount on the host that shares a pod's lifetime. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md
- `vsphere_volume` - (Optional) Represents a vSphere volume attached and mounted on kubelets host machine

» `aws_elastic_block_store`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
- `read_only` - (Optional) Whether to set the read-only property in VolumeMounts to "true". If omitted, the default is "false". For more info see Kubernetes reference
- `volume_id` - (Required) Unique ID of the persistent disk resource in AWS (Amazon EBS volume). For more info see Kubernetes reference

» `azure_disk`

» **Arguments**

- `caching_mode` - (Required) Host Caching mode: None, Read Only, Read Write.
- `data_disk_uri` - (Required) The URI the data disk in the blob storage
- `disk_name` - (Required) The Name of the data disk in the blob storage
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).

» `azure_file`

» **Arguments**

- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `secret_name` - (Required) The name of secret that contains Azure Storage Account Name and Key
- `share_name` - (Required) Share Name

» `ceph_fs`

» **Arguments**

- `monitors` - (Required) Monitors is a collection of Ceph monitors For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `path` - (Optional) Used as the mounted root, rather than the full Ceph tree, default is /
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to `false` (read/write). For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_file` - (Optional) The path to key ring for User, default is /etc/ceph/user.secret For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_ref` - (Optional) Reference to the authentication secret for User, default is empty. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `user` - (Optional) User is the rados user name, default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

» `cinder`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write). For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `volume_id` - (Required) Volume ID used to identify the volume in Cinder. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

» `fc`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified.
- `lun` - (Required) FC target lun number
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `target_ww_ns` - (Required) FC target worldwide names (WWNs)

» `flex_volume`

» **Arguments**

- `driver` - (Required) Driver is the name of the driver to use for this volume.
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. The default filesystem depends on FlexVolume script.
- `options` - (Optional) Extra command options if any.
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false (read/write).
- `secret_ref` - (Optional) Reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

» `flocker`

» Arguments

- `dataset_name` - (Optional) Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
- `dataset_uuid` - (Optional) UUID of the dataset. This is unique identifier of a Flocker dataset

» `gce_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). For more info see Kubernetes reference
- `pd_name` - (Required) Unique name of the PD resource in GCE. Used to identify the disk in GCE. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false. For more info see Kubernetes reference

» `glusterfs`

» Arguments

- `endpoints_name` - (Required) The endpoint name that details Glusterfs topology. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `path` - (Required) The Glusterfs volume path. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `read_only` - (Optional) Whether to force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

» `host_path`

» Arguments

- `path` - (Optional) Path of the directory on the host. For more info see Kubernetes reference
- `type` - (Optional) Type for HostPath volume. Defaults to "". For more info see Kubernetes reference

» `iscsi`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `iqn` - (Required) Target iSCSI Qualified Name.
- `iscsi_interface` - (Optional) iSCSI interface name that uses an iSCSI transport. Defaults to 'default' (tcp).
- `lun` - (Optional) iSCSI target lun number.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false.
- `target_portal` - (Required) iSCSI target portal. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

» `local`

» Arguments

- `path` - (Optional) Path of the directory on the host. For more info see Kubernetes reference

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the persistent volume that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference

- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the persistent volume. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the persistent volume, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this persistent volume that can be used by clients to determine when persistent volume has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this persistent volume.
- `uid` - The unique in time and space value for this persistent volume. For more info see Kubernetes reference

» `nfs`

» Arguments

- `path` - (Required) Path that is exported by the NFS server. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the NFS export to be mounted with read-only permissions. Defaults to false. For more info see Kubernetes reference
- `server` - (Required) Server is the hostname or IP address of the NFS server. For more info see Kubernetes reference

» `photon_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified.
- `pd_id` - (Required) ID that identifies Photon Controller persistent disk

» `quobyte`

» **Arguments**

- `group` - (Optional) Group to map volume access to Default is no group
- `read_only` - (Optional) Whether to force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
- `registry` - (Required) Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
- `user` - (Optional) User to map volume access to Defaults to serivceaccount user
- `volume` - (Required) Volume is a string that references an already created Quobyte volume by name.

» `rbd`

» **Arguments**

- `ceph_monitors` - (Required) A collection of Ceph monitors. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `keyring` - (Optional) Keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rados_user` - (Optional) The rados user name. Default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_image` - (Required) The rados image name. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_pool` - (Optional) The rados pool name. Default is rbd. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `secret_ref` - (Optional) Name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

» secret_ref

» Arguments

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» vsphere_volume

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified.
- `volume_path` - (Required) Path that identifies vSphere volume vmdk

» Import

Persistent Volume can be imported using its name, e.g.

```
$ terraform import kubernetes_persistent_volume.example terraform-example
```

» kubernetes_persistent_volume_claim

This resource allows the user to request for and claim to a persistent volume.

» Example Usage

```
resource "kubernetes_persistent_volume_claim" "example" {
  metadata {
    name = "exampleclaimname"
  }
  spec {
    access_modes = ["ReadWriteMany"]
    resources {
      requests = {
        storage = "5Gi"
      }
    }
    volume_name = "${kubernetes_persistent_volume.example.metadata.0.name}"
  }
}
```

```
resource "kubernetes_persistent_volume" "example" {
  metadata {
    name = "examplevolumename"
  }
  spec {
    capacity = {
      storage = "10Gi"
    }
    access_modes = ["ReadWriteMany"]
    persistent_volume_source {
      gce_persistent_disk {
        pd_name = "test-123"
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard persistent volume claim's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the desired characteristics of a volume requested by a pod author. For more info see Kubernetes reference
- `wait_until_bound` - (Optional) Whether to wait for the claim to reach `Bound` state (to find volume in which to claim the space)

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the persistent volume claim that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the persistent volume claim. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the persistent volume claim, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the persistent volume claim must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this persistent volume claim that can be used by clients to determine when persistent volume claim has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this persistent volume claim.
- `uid` - The unique in time and space value for this persistent volume claim. For more info see Kubernetes reference

» `spec`

» Arguments

- `access_modes` - (Required) A set of the desired access modes the volume should have. For more info see Kubernetes reference
- `resources` - (Required) A list of the minimum resources the volume should have. For more info see Kubernetes reference
- `selector` - (Optional) A label query over volumes to consider for binding.
- `volume_name` - (Optional) The binding reference to the PersistentVolume backing this claim.
- `storage_class_name` - (Optional) Name of the storage class requested by the claim

» `match_expressions`

» Arguments

- `key` - (Optional) The label key that the selector applies to.
- `operator` - (Optional) A key's relationship to a set of values. Valid operators ard `In`, `NotIn`, `Exists` and `DoesNotExist`.
- `values` - (Optional) An array of string values. If the operator is `In` or `NotIn`, the values array must be non-empty. If the operator is `Exists` or `DoesNotExist`, the values array must be empty. This array is replaced during a strategic merge patch.

» `resources`

» Arguments

- `limits` - (Optional) Map describing the maximum amount of compute resources allowed. For more info see Kubernetes reference/
- `requests` - (Optional) Map describing the minimum amount of compute resources required. If this is omitted for a container, it defaults to `limits` if that is explicitly specified, otherwise to an implementation-defined value. For more info see Kubernetes reference/

» `selector`

» Arguments

- `match_expressions` - (Optional) A list of label selector requirements. The requirements are ANDed.
- `match_labels` - (Optional) A map of {key,value} pairs. A single {key,value} in the matchLabels map is equivalent to an element of `match_expressions`, whose key field is "key", the operator is "In", and the values array contains only "value". The requirements are ANDed.

» Import

Persistent Volume Claim can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_persistent_volume_claim.example default/example-name
```

» kubernetes_pod

A pod is a group of one or more containers, the shared storage for those containers, and options about how to run the containers. Pods are always co-located

and co-scheduled, and run in a shared context.

Read more at Kubernetes reference/

» Example Usage

```
resource "kubernetes_pod" "test" {
  metadata {
    name = "terraform-example"
  }

  spec {
    container {
      image = "nginx:1.7.9"
      name  = "example"

      env {
        name  = "environment"
        value = "test"
      }

      liveness_probe {
        http_get {
          path = "/nginx_status"
          port = 80

          http_header {
            name  = "X-Custom-Header"
            value = "Awesome"
          }
        }

        initial_delay_seconds = 3
        period_seconds        = 3
      }
    }

    dns_config {
      nameservers = ["1.1.1.1", "8.8.8.8", "9.9.9.9"]
      searches    = ["example.com"]

      option {
        name  = "ndots"
        value = 1
      }
```

```
      option {
        name = "use-vc"
      }
    }

    dns_policy = "None"
  }
}
```

terraform version of the pods/pod-with-node-affinity.yaml example.

```
resource "kubernetes_pod" "with_node_affinity" {
  metadata {
    name = "with-node-affinity"
  }

  spec {
    affinity {
      node_affinity {
        required_during_scheduling_ignored_during_execution {
          node_selector_term {
            match_expressions {
              key      = "kubernetes.io/e2e-az-name"
              operator = "In"
              values   = ["e2e-az1", "e2e-az2"]
            }
          }
        }

        preferred_during_scheduling_ignored_during_execution {
          weight = 1

          preference {
            match_expressions {
              key      = "another-node-label-key"
              operator = "In"
              values   = ["another-node-label-value"]
            }
          }
        }
      }
    }

    container {
      name  = "with-node-affinity"
      image = "k8s.gcr.io/pause:2.0"
```

```
    }
  }
}
```

terraform version of the pods/pod-with-pod-affinity.yaml example.

```
resource "kubernetes_pod" "with_pod_affinity" {
  metadata {
    name = "with-pod-affinity"
  }

  spec {
    affinity {
      pod_affinity {
        required_during_scheduling_ignored_during_execution {
          label_selector {
            match_expressions {
              key      = "security"
              operator = "In"
              values   = ["S1"]
            }
          }

          topology_key = "failure-domain.beta.kubernetes.io/zone"
        }
      }

      pod_anti_affinity {
        preferred_during_scheduling_ignored_during_execution {
          weight = 100

          pod_affinity_term {
            label_selector {
              match_expressions {
                key      = "security"
                operator = "In"
                values   = ["S2"]
              }
            }

            topology_key = "failure-domain.beta.kubernetes.io/zone"
          }
        }
      }
    }

    container {
```

```
      name  = "with-pod-affinity"
      image = "k8s.gcr.io/pause:2.0"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard pod's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec of the pod owned by the cluster

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the pod that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the pod. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference

- `name` - (Optional) Name of the pod, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the pod must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this pod that can be used by clients to determine when pod has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this pod.
- `uid` - The unique in time and space value for this pod. For more info see Kubernetes reference

» `spec`

» Arguments

- `affinity` - (Optional) A group of affinity scheduling rules. If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
- `active_deadline_seconds` - (Optional) Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
- `automount_service_account_token` - (Optional) Indicates whether a service account token should be automatically mounted. Defaults to false for Pods.
- `container` - (Optional) List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated. For more info see Kubernetes reference
- `init_container` - (Optional) List of init containers belonging to the pod. Init containers always run to completion and each must complete successfully before the next is started. For more info see Kubernetes reference/
- `dns_policy` - (Optional) Set DNS policy for containers within the pod. Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'. Optional: Defaults to 'ClusterFirst', see Kubernetes reference.

- `dns_config` - (Optional) Specifies the DNS parameters of a pod. Parameters specified here will be merged to the generated DNS configuration based on DNSPolicy. Defaults to empty. See `dns_config` block definition below.
- `host_alias` - (Optional) List of hosts and IPs that will be injected into the pod's hosts file if specified. Optional: Defaults to empty. See `host_alias` block definition below.
- `host_ipc` - (Optional) Use the host's ipc namespace. Optional: Defaults to false.
- `host_network` - (Optional) Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified.
- `host_pid` - (Optional) Use the host's pid namespace.
- `hostname` - (Optional) Specifies the hostname of the Pod If not specified, the pod's hostname will be set to a system-defined value.
- `image_pull_secrets` - (Optional) ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. For example, in the case of docker, only DockerConfig type secrets are honored. For more info see Kubernetes reference
- `node_name` - (Optional) NodeName is a request to schedule this pod onto a specific node. If it is non-empty, the scheduler simply schedules this pod onto that node, assuming that it fits resource requirements.
- `node_selector` - (Optional) NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node. For more info see Kubernetes reference.
- `restart_policy` - (Optional) Restart policy for all containers within the pod. One of Always, OnFailure, Never. For more info see Kubernetes reference.
- `security_context` - (Optional) SecurityContext holds pod-level security attributes and common container settings. Optional: Defaults to empty
- `service_account_name` - (Optional) ServiceAccountName is the name of the ServiceAccount to use to run this pod. For more info see http://releases.k8s.io/HEAD/docs/design/service_accounts.md.
- `share_process_namespace` - (Optional) Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set.
- `subdomain` - (Optional) If specified, the fully qualified Pod hostname will be "...svc.". If not specified, the pod will not have a domainname at all..
- `termination_grace_period_seconds` - (Optional) Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates

delete immediately. If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process.

- `toleration` - (Optional) Optional pod node tolerations. For more info see Kubernetes reference
- `volume` - (Optional) List of volumes that can be mounted by containers belonging to the pod. For more info see Kubernetes reference

» `affinity`

» Arguments

- `node_affinity` - (Optional) Node affinity scheduling rules for the pod. For more info see Kubernetes reference
- `pod_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference
- `pod_anti_affinity` - (Optional) Inter-pod topological affinity. rules that specify that certain pods should be placed in the same topological domain (e.g. same node, same rack, same zone, same power domain, etc.) For more info see Kubernetes reference

» `node_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to an update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution`

» Arguments

- `node_selector_term` - (Required) A list of node selector terms. The terms are ORed.

» `node_selector_term`

» **Arguments**

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» **Arguments**

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `preferred_during_scheduling_ignored_during_execution`

» **Arguments**

- `preference` - (Required) A node selector term, associated with the corresponding weight.
- `weight` - (Required) Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

» `preference`

» **Arguments**

- `match_expressions` - (Optional) A list of node selector requirements by node's labels.
- `match_fields` - (Optional) A list of node selector requirements by node's fields.

» `match_expressions` / `match_fields`

» Arguments

- `key` - (Required) The label key that the selector applies to.
- `operator` - (Required) Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt.
- `values` - (Optional) An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer.

» `pod_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `pod_anti_affinity`

» Arguments

- `required_during_scheduling_ignored_during_execution` - (Optional) If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node.
- `preferred_during_scheduling_ignored_during_execution` - (Optional) The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions.

» `required_during_scheduling_ignored_during_execution` (pod_affinity_term)

» Arguments

- `label_selector` - (Optional) A label query over a set of resources, in this case pods.
- `namespaces` - (Optional) Specifies which namespaces the `label_selector` applies to (matches against). Null or empty list means "this pod's namespace"
- `topology_key` - (Optional) This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the `label_selector` in the specified namespaces, where co-located is defined as running on a node whose value of the label with key `topology_key` matches that of any node on which any of the selected pods is running. Empty `topology_key` is not allowed.

» `preferred_during_scheduling_ignored_during_execution`

» Arguments

- `pod_affinity_term` - (Required) A pod affinity term, associated with the corresponding weight.
- `weight` - (Required) Weight associated with matching the corresponding `pod_affinity_term`, in the range 1-100.

» `container`

» Arguments

- `args` - (Optional) Arguments to the entrypoint. The docker image's CMD is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference
- `command` - (Optional) Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided. Variable references $(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. For more info see Kubernetes reference

- `env` - (Optional) Block of string name and value pairs to set in the container's environment. May be declared multiple times. Cannot be updated.
- `env_from` - (Optional) List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
- `image` - (Optional) Docker image name. For more info see Kubernetes reference
- `image_pull_policy` - (Optional) Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. For more info see Kubernetes reference
- `lifecycle` - (Optional) Actions that the management system should take in response to container lifecycle events
- `liveness_probe` - (Optional) Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated. For more info see Kubernetes reference
- `name` - (Required) Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.
- `port` - (Optional) List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Cannot be updated.
- `readiness_probe` - (Optional) Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated. For more info see Kubernetes reference
- `resources` - (Optional) Compute Resources required by this container. Cannot be updated. For more info see Kubernetes reference
- `security_context` - (Optional) Security options the pod should run with. For more info see http://releases.k8s.io/HEAD/docs/design/security_context.md
- `stdin` - (Optional) Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF.
- `stdin_once` - (Optional) Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container

processes that reads from stdin will never receive an EOF.
- `termination_message_path` - (Optional) Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Defaults to /dev/termination-log. Cannot be updated.
- `tty` - (Optional) Whether this container should allocate a TTY for itself
- `volume_mount` - (Optional) Pod volumes to mount into the container's filesystem. Cannot be updated.
- `working_dir` - (Optional) Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

» `aws_elastic_block_store`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
- `read_only` - (Optional) Whether to set the read-only property in VolumeMounts to "true". If omitted, the default is "false". For more info see Kubernetes reference
- `volume_id` - (Required) Unique ID of the persistent disk resource in AWS (Amazon EBS volume). For more info see Kubernetes reference

» `azure_disk`

» Arguments

- `caching_mode` - (Required) Host Caching mode: None, Read Only, Read Write.
- `data_disk_uri` - (Required) The URI the data disk in the blob storage
- `disk_name` - (Required) The Name of the data disk in the blob storage
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).

» `azure_file`

» **Arguments**

- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `secret_name` - (Required) The name of secret that contains Azure Storage Account Name and Key
- `share_name` - (Required) Share Name

» `capabilities`

» **Arguments**

- `add` - (Optional) Added capabilities
- `drop` - (Optional) Removed capabilities

» `ceph_fs`

» **Arguments**

- `monitors` - (Required) Monitors is a collection of Ceph monitors For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `path` - (Optional) Used as the mounted root, rather than the full Ceph tree, default is /
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to `false` (read/write). For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_file` - (Optional) The path to key ring for User, default is /etc/ceph/user.secret For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `secret_ref` - (Optional) Reference to the authentication secret for User, default is empty. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it
- `user` - (Optional) User is the rados user name, default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/cephfs/README.md#how-to-use-it

» `cinder`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write). For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `volume_id` - (Required) Volume ID used to identify the volume in Cinder. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md

» `config_map`

» Arguments

- `default_mode` - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `config_map_ref`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the ConfigMap must be defined

» `config_map_key_ref`

» Arguments

- `key` - (Optional) The key to select.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `dns_config`

» **Arguments**

- `nameservers` - (Optional) A list of DNS name server IP addresses specified as strings. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed. Optional: Defaults to empty.
- `option` - (Optional) A list of DNS resolver options specified as blocks with `name`/`value` pairs. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy. Optional: Defaults to empty.
- `searches` - (Optional) A list of DNS search domains for host-name lookup specified as strings. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed. Optional: Defaults to empty.

The `option` block supports the following:

- `name` - (Required) Name of the option.
- `value` - (Optional) Value of the option. Optional: Defaults to empty.

» `downward_api`

» **Arguments**

- `default_mode` - (Optional) Optional: mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) If unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error. Paths must be relative and may not contain the '..' path or start with '..'.

» `empty_dir`

» **Arguments**

- `medium` - (Optional) What type of storage medium should back this directory. The default is "" which means to use the node's default medium.

Must be an empty string (default) or Memory. For more info see Kubernetes reference

» `env`

» **Arguments**

- `name` - (Required) Name of the environment variable. Must be a C_IDENTIFIER
- `value` - (Optional) Variable references $(VAR_NAME) are expanded using the previous defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. The $(VAR_NAME) syntax can be escaped with a double $$, ie: $$(VAR_NAME). Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
- `value_from` - (Optional) Source for the environment variable's value

» `env_from`

» **Arguments**

- `config_map_ref` - (Optional) The ConfigMap to select from
- `prefix` - (Optional) An optional identifer to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER..
- `secret_ref` - (Optional) The Secret to select from

» `exec`

» **Arguments**

- `command` - (Optional) Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

» `fc`

» **Arguments**

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.

- `lun` - (Required) FC target lun number
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false (read/write).
- `target_ww_ns` - (Required) FC target worldwide names (WWNs)

» `field_ref`

» Arguments

- `api_version` - (Optional) Version of the schema the FieldPath is written in terms of, defaults to "v1".
- `field_path` - (Optional) Path of the field to select in the specified API version

» `flex_volume`

» Arguments

- `driver` - (Required) Driver is the name of the driver to use for this volume.
- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
- `options` - (Optional) Extra command options if any.
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false (read/write).
- `secret_ref` - (Optional) Reference to the secret object containing sensitive information to pass to the plugin scripts. This may be empty if no secret object is specified. If the secret object contains more than one secret, all secrets are passed to the plugin scripts.

» `flocker`

» Arguments

- `dataset_name` - (Optional) Name of the dataset stored as metadata -> name on the dataset for Flocker should be considered as deprecated
- `dataset_uuid` - (Optional) UUID of the dataset. This is unique identifier of a Flocker dataset

» `gce_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `partition` - (Optional) The partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). For more info see Kubernetes reference
- `pd_name` - (Required) Unique name of the PD resource in GCE. Used to identify the disk in GCE. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the ReadOnly setting in VolumeMounts. Defaults to false. For more info see Kubernetes reference

» `git_repo`

» Arguments

- `directory` - (Optional) Target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
- `repository` - (Optional) Repository URL
- `revision` - (Optional) Commit hash for the specified revision.

» `glusterfs`

» Arguments

- `endpoints_name` - (Required) The endpoint name that details Glusterfs topology. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `path` - (Required) The Glusterfs volume path. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod
- `read_only` - (Optional) Whether to force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md#create-a-pod

» `host_alias`

» Arguments

- `hostnames` - (Required) Hostnames for the IP address.

- `ip` - (Required) IP address of the host file entry.

» `host_path`

» **Arguments**

- `path` - (Optional) Path of the directory on the host. For more info see Kubernetes reference
- `type` - (Optional) Type for HostPath volume. Defaults to "". For more info see Kubernetes reference

» `http_get`

» **Arguments**

- `host` - (Optional) Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
- `http_header` - (Optional) Scheme to use for connecting to the host.
- `path` - (Optional) Path to access on the HTTP server.
- `port` - (Optional) Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
- `scheme` - (Optional) Scheme to use for connecting to the host.

» `http_header`

» **Arguments**

- `name` - (Optional) The header field name
- `value` - (Optional) The header field value

» `image_pull_secrets`

» **Arguments**

- `name` - (Required) Name of the referent. For more info see Kubernetes reference

» `iscsi`

» **Arguments**

- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: ”ext4”, ”xfs”, ”ntfs”. Implicitly inferred to be ”ext4” if unspecified. For more info see Kubernetes reference
- `iqn` - (Required) Target iSCSI Qualified Name.
- `iscsi_interface` - (Optional) iSCSI interface name that uses an iSCSI transport. Defaults to 'default' (tcp).
- `lun` - (Optional) iSCSI target lun number.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false.
- `target_portal` - (Required) iSCSI target portal. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

» `items`

» Arguments

- `key` - (Optional) The key to project.
- `mode` - (Optional) Optional: mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `path` - (Optional) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `lifecycle`

» Arguments

- `post_start` - (Optional) post_start is called immediately after a container is created. If the handler fails, the container is terminated and restarted according to its restart policy. Other management of the container blocks until the hook completes. For more info see Kubernetes reference
- `pre_stop` - (Optional) pre_stop is called immediately before a container is terminated. The container is terminated after the handler completes. The reason for termination is passed to the handler. Regardless of the outcome of the handler, the container is eventually terminated. Other management of the container blocks until the hook completes. For more info see Kubernetes reference

» `limits`

» Arguments

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `liveness_probe`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `nfs`

» Arguments

- `path` - (Required) Path that is exported by the NFS server. For more info see Kubernetes reference
- `read_only` - (Optional) Whether to force the NFS export to be mounted with read-only permissions. Defaults to false. For more info see Kubernetes reference
- `server` - (Required) Server is the hostname or IP address of the NFS server. For more info see Kubernetes reference

» `persistent_volume_claim`

» Arguments

- `claim_name` - (Optional) ClaimName is the name of a PersistentVolumeClaim in the same
- `read_only` - (Optional) Will force the ReadOnly setting in VolumeMounts.

» `photon_persistent_disk`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
- `pd_id` - (Required) ID that identifies Photon Controller persistent disk

» `port`

» Arguments

- `container_port` - (Required) Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
- `host_ip` - (Optional) What host IP to bind the external port to.
- `host_port` - (Optional) Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
- `name` - (Optional) If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services
- `protocol` - (Optional) Protocol for port. Must be UDP or TCP. Defaults to "TCP".

» `post_start`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `pre_stop`

» Arguments

- `exec` - (Optional) exec specifies the action to take.
- `http_get` - (Optional) Specifies the http request to perform.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported

» `quobyte`

» **Arguments**

- `group` - (Optional) Group to map volume access to Default is no group
- `read_only` - (Optional) Whether to force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
- `registry` - (Required) Registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
- `user` - (Optional) User to map volume access to Defaults to serivceaccount user
- `volume` - (Required) Volume is a string that references an already created Quobyte volume by name.

» `rbd`

» **Arguments**

- `ceph_monitors` - (Required) A collection of Ceph monitors. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `fs_type` - (Optional) Filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. For more info see Kubernetes reference
- `keyring` - (Optional) Keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rados_user` - (Optional) The rados user name. Default is admin. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_image` - (Required) The rados image name. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `rbd_pool` - (Optional) The rados pool name. Default is rbd. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it.
- `read_only` - (Optional) Whether to force the read-only setting in VolumeMounts. Defaults to false. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it
- `secret_ref` - (Optional) Name of the authentication secret for RBDUser. If provided overrides keyring. Default is nil. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md#how-to-use-it

» `readiness_probe`

» **Arguments**

- `exec` - (Optional) exec specifies the action to take.
- `failure_threshold` - (Optional) Minimum consecutive failures for the probe to be considered failed after having succeeded.
- `http_get` - (Optional) Specifies the http request to perform.
- `initial_delay_seconds` - (Optional) Number of seconds after the container has started before liveness probes are initiated. For more info see Kubernetes reference
- `period_seconds` - (Optional) How often (in seconds) to perform the probe
- `success_threshold` - (Optional) Minimum consecutive successes for the probe to be considered successful after having failed.
- `tcp_socket` - (Optional) TCPSocket specifies an action involving a TCP port. TCP hooks not yet supported
- `timeout_seconds` - (Optional) Number of seconds after which the probe times out. For more info see Kubernetes reference

» `resources`

» **Arguments**

- `limits` - (Optional) Describes the maximum amount of compute resources allowed. For more info see Kubernetes reference/
- `requests` - (Optional) Describes the minimum amount of compute resources required.

» `requests`

» **Arguments**

- `cpu` - (Optional) CPU
- `memory` - (Optional) Memory

» `resource_field_ref`

» **Arguments**

- `container_name` - (Optional) The name of the container
- `resource` - (Required) Resource to select

» `se_linux_options`

» **Arguments**

- `level` - (Optional) Level is SELinux level label that applies to the container.
- `role` - (Optional) Role is a SELinux role label that applies to the container.
- `type` - (Optional) Type is a SELinux type label that applies to the container.
- `user` - (Optional) User is a SELinux user label that applies to the container.

» `secret`

» Arguments

- `default_mode` - (Optional) Mode bits to use on created files by default. Must be a value between 0 and 0777. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
- `items` - (Optional) List of Secret Items to project into the volume. See `items` block definition below. If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked `optional`. Paths must be relative and may not contain the '..' path or start with '..'.
- `optional` - (Optional) Specify whether the Secret or it's keys must be defined.
- `secret_name` - (Optional) Name of the secret in the pod's namespace to use. For more info see Kubernetes reference

The `items` block supports the following:

- `key` - (Required) The key to project.
- `mode` - (Optional) Mode bits to use on this file, must be a value between 0 and 0777. If not specified, the volume defaultMode will be used.
- `path` - (Required) The relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

» `secret_ref`

» Arguments

- `name` - (Required) Name of the referent. For more info see Kubernetes reference
- `optional` - (Optional) Specify whether the Secret must be defined

» `secret_key_ref`

» **Arguments**

- `key` - (Optional) The key of the secret to select from. Must be a valid secret key.
- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `secret_ref`

» **Arguments**

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» **container `security_context`**

» **Arguments**

- `allow_privilege_escalation` - (Optional) AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN
- `capabilities` - (Optional) The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.
- `privileged` - (Optional) Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.
- `read_only_root_filesystem` - (Optional) Whether this container has a read-only root filesystem. Default is false.
- `run_as_group` - (Optional) The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and

PodSecurityContext, the value specified in SecurityContext takes precedence.

- `se_linux_options` - (Optional) The SELinux context to be applied to the container. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

» `capabilities`

» Arguments

- `add` - (Optional) A list of added capabilities.
- `drop` - (Optional) A list of removed capabilities.

» pod `security_context`

» Arguments

- `fs_group` - (Optional) A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume.
- `run_as_group` - (Optional) The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `run_as_non_root` - (Optional) Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
- `run_as_user` - (Optional) The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `se_linux_options` - (Optional) The SELinux context to be applied to all containers. If unspecified, the container runtime will allocate a random SELinux context for each container. May also be set in SecurityContext.

If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container.
- `supplemental_groups` - (Optional) A list of groups applied to the first process run in each container, in addition to the container's primary GID. If unspecified, no groups will be added to any container.

» `tcp_socket`

» Arguments

- `port` - (Required) Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

» `toleration`

» Arguments

- `effect` - (Optional) Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.
- `key` - (Optional) Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.
- `operator` - (Optional) Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.
- `toleration_seconds` - (Optional) TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
- `value` - (Optional) Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

» `value_from`

» Arguments

- `config_map_key_ref` - (Optional) Selects a key of a ConfigMap.
- `field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..

- `resource_field_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..
- `secret_key_ref` - (Optional) Selects a field of the pod: supports metadata.name, metadata.namespace, metadata.labels, metadata.annotations, spec.nodeName, spec.serviceAccountName, status.podIP..

» `volume`

» Arguments

- `aws_elastic_block_store` - (Optional) Represents an AWS Disk resource that is attached to a kubelet's host machine and then exposed to the pod. For more info see Kubernetes reference
- `azure_disk` - (Optional) Represents an Azure Data Disk mount on the host and bind mount to the pod.
- `azure_file` - (Optional) Represents an Azure File Service mount on the host and bind mount to the pod.
- `ceph_fs` - (Optional) Represents a Ceph FS mount on the host that shares a pod's lifetime
- `cinder` - (Optional) Represents a cinder volume attached and mounted on kubelets host machine. For more info see http://releases.k8s.io/HEAD/examples/mysql-cinder-pd/README.md
- `config_map` - (Optional) ConfigMap represents a configMap that should populate this volume
- `downward_api` - (Optional) DownwardAPI represents downward API about the pod that should populate this volume
- `empty_dir` - (Optional) EmptyDir represents a temporary directory that shares a pod's lifetime. For more info see Kubernetes reference
- `fc` - (Optional) Represents a Fibre Channel resource that is attached to a kubelet's host machine and then exposed to the pod.
- `flex_volume` - (Optional) Represents a generic volume resource that is provisioned/attached using an exec based plugin. This is an alpha feature and may change in future.
- `flocker` - (Optional) Represents a Flocker volume attached to a kubelet's host machine and exposed to the pod for its usage. This depends on the Flocker control service being running
- `gce_persistent_disk` - (Optional) Represents a GCE Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin. For more info see Kubernetes reference
- `git_repo` - (Optional) GitRepo represents a git repository at a particular revision.
- `glusterfs` - (Optional) Represents a Glusterfs volume that is attached to a host and exposed to the pod. Provisioned by an admin. For more info

see http://releases.k8s.io/HEAD/examples/volumes/glusterfs/README.md

- `host_path` - (Optional) Represents a directory on the host. Provisioned by a developer or tester. This is useful for single-node development and testing only! On-host storage is not supported in any way and WILL NOT WORK in a multi-node cluster. For more info see Kubernetes reference
- `iscsi` - (Optional) Represents an ISCSI Disk resource that is attached to a kubelet's host machine and then exposed to the pod. Provisioned by an admin.
- `name` - (Optional) Volume's name. Must be a DNS_LABEL and unique within the pod. For more info see Kubernetes reference
- `nfs` - (Optional) Represents an NFS mount on the host. Provisioned by an admin. For more info see Kubernetes reference
- `persistent_volume_claim` - (Optional) The specification of a persistent volume.
- `photon_persistent_disk` - (Optional) Represents a PhotonController persistent disk attached and mounted on kubelets host machine
- `quobyte` - (Optional) Quobyte represents a Quobyte mount on the host that shares a pod's lifetime
- `rbd` - (Optional) Represents a Rados Block Device mount on the host that shares a pod's lifetime. For more info see http://releases.k8s.io/HEAD/examples/volumes/rbd/README.md
- `secret` - (Optional) Secret represents a secret that should populate this volume. For more info see Kubernetes reference
- `vsphere_volume` - (Optional) Represents a vSphere volume attached and mounted on kubelets host machine

» `volume_mount`

» Arguments

- `mount_path` - (Required) Path within the container at which the volume should be mounted. Must not contain ':'.
- `name` - (Required) This must match the Name of a Volume.
- `read_only` - (Optional) Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.
- `sub_path` - (Optional) Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).

» `vsphere_volume`

» Arguments

- `fs_type` - (Optional) Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs".

Implicitly inferred to be ”ext4” if unspecified.
- `volume_path` - (Required) Path that identifies vSphere volume vmdk

» Timeouts

The following Timeout configuration options are available for the `kubernetes_pod` resource:

- `create` - (Default `5 minutes`) Used for Creating Pods.
- `delete` - (Default `5 minutes`) Used for Destroying Pods.

» Import

Pod can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_pod.example default/terraform-example
```

» kubernetes_replication_controller

A Replication Controller ensures that a specified number of pod “replicas” are running at any one time. In other words, a Replication Controller makes sure that a pod or homogeneous set of pods are always up and available. If there are too many pods, it will kill some. If there are too few, the Replication Controller will start more.

WARNING: In many cases it is recommended to create a Deployment instead of a Replication Controller.

» Example Usage

```
resource "kubernetes_replication_controller" "example" {
  metadata {
    name = "terraform-example"
    labels = {
      test = "MyExampleApp"
    }
  }

  spec {
    selector = {
      test = "MyExampleApp"
    }
    template {
```

```
      metadata {
        labels = {
          test = "MyExampleApp"
        }
        annotations = {
          "key1" = "value1"
        }
      }

      spec {
        container {
          image = "nginx:1.7.8"
          name  = "example"

          liveness_probe {
            http_get {
              path = "/nginx_status"
              port = 8080

              http_header {
                name  = "X-Custom-Header"
                value = "Awesome"
              }
            }

            initial_delay_seconds = 3
            period_seconds        = 3
          }

          resources {
            limits {
              cpu    = "0.5"
              memory = "512Mi"
            }
            requests {
              cpu    = "250m"
              memory = "50Mi"
            }
          }
        }
      }
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard replication controller's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the specification of the desired behavior of the replication controller. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the replication controller that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the replication controller. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the replication controller, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the replication controller must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this replication controller that can be used by clients to determine when replication controller has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this replication controller.
- `uid` - The unique in time and space value for this replication controller. For more info see Kubernetes reference

» `spec`

» Arguments

- `min_ready_seconds` - (Optional) Minimum number of seconds for which a newly created pod should be ready without any of its container crashing, for it to be considered available. Defaults to 0 (pod will be considered available as soon as it is ready)
- `replicas` - (Optional) The number of desired replicas. Defaults to 1. For more info see Kubernetes reference
- `selector` - (Required) A label query over pods that should match the Replicas count. Label keys and values that must match in order to be controlled by this replication controller. **Should match labels (`metadata.0.labels`)**. For more info see Kubernetes reference
- `template` - (Required) Template is the object that describes the pod that will be created if insufficient replicas are detected. This takes precedence over a TemplateRef. For more info see Kubernetes reference

» Nested Blocks

» `spec.template`

» Arguments

- `metadata` - (Optional) Standard object's metadata. More info: Kubernetes reference. While required by the kubernetes API, this field is marked as optional to allow the usage of the deprecated pod spec fields that were mistakenly placed directly under the `template` block.
- `spec` - (Optional) Specification of the desired behavior of the pod. More info: Kubernetes reference

NOTE: all the fields from the `spec.template.spec` block are also accepted at the `spec.template` level but that usage is deprecated. All existing configurations should be updated to only use the new fields under `spec.template.spec`. Mixing the usage of deprecated fields with new fields is not supported.

» Nested Blocks

» `spec.template.metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the replication controller that may be used to store arbitrary metadata. For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the pods managed by this replication controller . **Should match `selector`**. For more info see Kubernetes reference
- `name` - (Optional) Name of the replication controller, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the replication controller must be unique.

» Nested Blocks

» `spec.template.spec`

» Arguments

These arguments are the same as the for the `spec` block of a Pod.

Please see the Pod resource for reference.

» Timeouts

The following Timeout configuration options are available:

- `create` - (Default `10 minutes`) Used for creating new controller
- `update` - (Default `10 minutes`) Used for updating a controller
- `delete` - (Default `10 minutes`) Used for destroying a controller

» Import

Replication Controller can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_replication_controller.example default/terraform-example
```

NOTE: Imported `kubernetes_replication_controller` resource will only have their fields from the `spec.template.spec` block in the state. Deprecated fields at the `spec.template` level are not updated during import. Configurations using the deprecated fields should be updated to only use the new fields under `spec.template.spec`.

» kubernetes_resource_quota

A resource quota provides constraints that limit aggregate resource consumption per namespace. It can limit the quantity of objects that can be created in a namespace by type, as well as the total amount of compute resources that may be consumed by resources in that project.

» Example Usage

```
resource "kubernetes_resource_quota" "example" {
  metadata {
    name = "terraform-example"
  }
  spec {
    hard = {
      pods = 10
    }
    scopes = ["BestEffort"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard resource quota's metadata. For more info see Kubernetes reference
- `spec` - (Optional) Spec defines the desired quota. Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the resource quota that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the resource quota. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the resource quota, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the resource quota must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this resource quota that can be used by clients to determine when resource quota has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this resource quota.
- `uid` - The unique in time and space value for this resource quota. For more info see Kubernetes reference

» `spec`

» Arguments

- `hard` - (Optional) The set of desired hard limits for each named resource. For more info see http://releases.k8s.io/HEAD/docs/design/admission_control_resource_quota.md#admissioncontrol-plugin-resourcequota

- `scopes` - (Optional) A collection of filters that must match each object tracked by a quota. If not specified, the quota matches all objects.

» Import

Resource Quota can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_resource_quota.example default/terraform-example
```

» kubernetes_role

A role contains rules that represent a set of permissions. Permissions are purely additive (there are no “deny” rules).

» Example Usage

```
resource "kubernetes_role" "example" {
  metadata {
    name = "terraform-example"
    labels = {
      test = "MyRole"
    }
  }

  rule {
    api_groups     = [""]
    resources      = ["pods"]
    resource_names = ["foo"]
    verbs          = ["get", "list", "watch"]
  }
  rule {
    api_groups = ["apps"]
    resources  = ["deployments"]
    verbs      = ["get", "list"]
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard role's metadata. For more info see Kubernetes reference

- `rule` - (Required) List of rules that define the set of permissions for this role. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the role that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the role. **Must match `selector`. By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the role, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the role must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this role that can be used by clients to determine when role has changed. For more info see Kubernetes reference

- `self_link` - A URL representing this role.
- `uid` - The unique in time and space value for this role. For more info see Kubernetes reference

» `rule`

» Arguments

- `api_groups` - (Required) List of APIGroups that contains the resources.
- `resources` - (Required) List of resources that the rule applies to.
- `resource_names` - (Optional) White list of names that the rule applies to.
- `verbs` - (Required) List of Verbs that apply to ALL the ResourceKinds and AttributeRestrictions contained in this rule.

» Import

Role can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_role.example default/terraform-example
```

» kubernetes_role_binding

A RoleBinding may be used to grant permission at the namespace level

» Example Usage

```
resource "kubernetes_role_binding" "example" {
  metadata {
    name      = "terraform-example"
    namespace = "default"
  }
  role_ref {
    api_group = "rbac.authorization.k8s.io"
    kind      = "Role"
    name      = "admin"
  }
  subject {
    kind      = "User"
    name      = "admin"
    api_group = "rbac.authorization.k8s.io"
  }
  subject {
    kind      = "ServiceAccount"
```

```
    name      = "default"
    namespace = "kube-system"
  }
  subject {
    kind      = "Group"
    name      = "system:masters"
    api_group = "rbac.authorization.k8s.io"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard kubernetes metadata. For more info see Kubernetes reference
- `role_ref` - (Required) The Role to bind Subjects to. For more info see Kubernetes reference
- `subject` - (Required) The Users, Groups, or ServiceAccounts to grand permissions to. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the role binding that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the role binding. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated**

by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem). For more info see Kubernetes reference

- `name` - (Optional) Name of the role binding, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the role binding must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this object that can be used by clients to determine when the object has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this role binding.
- `uid` - The unique in time and space value for this role binding. For more info see Kubernetes reference

» `role_ref`

» Arguments

- `name` - (Required) The name of this Role to bind Subjects to.
- `kind` - (Required) The type of binding to use. This value must be present and defaults to `Role`
- `api_group` - (Optional) The API group to drive authorization decisions. This value must be and defaults to `rbac.authorization.k8s.io`

» `subject`

» Arguments

- `name` - (Required) The name of this Role to bind Subjects to.
- `namespace` - (Optional) Namespace defines the namespace of the ServiceAccount to bind to. This value only applies to kind `ServiceAccount`
- `kind` - (Required) The type of binding to use. This value must be `ServiceAccount`, `User` or `Group`
- `api_group` - (Optional) The API group to drive authorization decisions. This value only applies to kind `User` and `Group`. It must be `rbac.authorization.k8s.io`

» Import

RoleBinding can be imported using the name, e.g.

```
$ terraform import kubernetes_role_binding.example default/terraform-name
```

» kubernetes_secret

The resource provides mechanisms to inject containers with sensitive information, such as passwords, while keeping containers agnostic of Kubernetes. Secrets can be used to store sensitive information either as individual properties or coarse-grained entries like entire files or JSON blobs. The resource will by default create a secret which is available to any pod in the specified (or default) namespace.

Read more about security properties and risks involved with using Kubernetes secrets: Kubernetes reference

Note: All arguments including the secret data will be stored in the raw state as plain-text. Read more about sensitive data in state.

» Example Usage

```
resource "kubernetes_secret" "example" {
  metadata {
    name = "basic-auth"
  }

  data = {
    username = "admin"
    password = "P4ssw0rd"
  }

  type = "kubernetes.io/basic-auth"
}
```

» Example Usage (Docker config)

```
resource "kubernetes_secret" "example" {
  metadata {
    name = "docker-cfg"
  }

  data = {
```

```
    ".dockerconfigjson" = "${file("${path.module}/.docker/config.json")}"
  }

  type = "kubernetes.io/dockerconfigjson"
}
```

» Example Usage (Service account token)

```
resource "kubernetes_secret" "example" {
  metadata {
    annotations = {
      "kubernetes.io/service-account.name" = "my-service-account"
    }
  }

  type = "kubernetes.io/service-account-token"
}
```

» Argument Reference

The following arguments are supported:

- `data` - (Optional) A map of the secret data.
- `metadata` - (Required) Standard secret's metadata. For more info see Kubernetes reference
- `type` - (Optional) The secret type. Defaults to `Opaque`. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the secret that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the secret. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the secret, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the secret must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this secret that can be used by clients to determine when secret has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this secret.
- `uid` - The unique in time and space value for this secret. For more info see Kubernetes reference

» Import

Secret can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_secret.example default/my-secret
```

» kubernetes_service

A Service is an abstraction which defines a logical set of pods and a policy by which to access them - sometimes called a micro-service.

» Example Usage

```
resource "kubernetes_service" "example" {
  metadata {
    name = "terraform-example"
  }
  spec {
    selector = {
      app = "${kubernetes_pod.example.metadata.0.labels.app}"
    }
    session_affinity = "ClientIP"
    port {
      port        = 8080
      target_port = 80
    }

    type = "LoadBalancer"
  }
}

resource "kubernetes_pod" "example" {
  metadata {
    name = "terraform-example"
    labels = {
      app = "MyApp"
    }
  }

  spec {
    container {
      image = "nginx:1.7.9"
      name  = "example"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard service's metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the behavior of a service. Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the service that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the service, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the service must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service that can be used by clients to determine when service has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this service.
- `uid` - The unique in time and space value for this service. For more info see Kubernetes reference

» `spec`

» Arguments

- `cluster_ip` - (Optional) The IP address of the service. It is usually assigned randomly by the master. If an address is specified manually and is not in use by others, it will be allocated to the service; otherwise, creation of the service will fail. `None` can be specified for headless services when proxying is not required. Ignored if type is `ExternalName`. For more info see Kubernetes reference
- `external_ips` - (Optional) A list of IP addresses for which nodes in the cluster will also accept traffic for this service. These IPs are not managed by Kubernetes. The user is responsible for ensuring that traffic arrives at a node with this IP. A common example is external load-balancers that are not part of the Kubernetes system.
- `external_name` - (Optional) The external reference that kubedns or equivalent will return as a CNAME record for this service. No proxying will be involved. Must be a valid DNS name and requires `type` to be `ExternalName`.
- `external_traffic_policy` - (Optional) Denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints. `Local` preserves the client source IP and avoids a second hop for LoadBalancer and Nodeport type services, but risks potentially imbalanced traffic spreading. `Cluster` obscures the client source IP and may cause a second hop to another node, but should have good overall load-spreading. More info: https://kubernetes.io/docs/tutorials/services/source-ip/
- `load_balancer_ip` - (Optional) Only applies to `type = LoadBalancer`. LoadBalancer will get created with the IP specified in this field. This feature depends on whether the underlying cloud-provider supports specifying this field when a load balancer is created. This field will be ignored if the cloud-provider does not support the feature.
- `load_balancer_source_ranges` - (Optional) If specified and supported by the platform, this will restrict traffic through the cloud-provider load-balancer will be restricted to the specified client IPs. This field will be ignored if the cloud-provider does not support the feature. For more info see Kubernetes reference
- `port` - (Required) The list of ports that are exposed by this service. For more info see Kubernetes reference
- `publish_not_ready_addresses` - (Optional) When set to true, indicates that DNS implementations must publish the `notReadyAddresses` of subsets for the Endpoints associated with the Service. The default value is `false`. The primary use case for setting this field is to use a StatefulSet's Headless Service to propagate `SRV` records for its Pods without respect to their readiness for purpose of peer discovery.
- `selector` - (Optional) Route service traffic to pods with label keys and values matching this selector. Only applies to types `ClusterIP`, `NodePort`,

and `LoadBalancer`. For more info see Kubernetes reference
- `session_affinity` - (Optional) Used to maintain session affinity. Supports `ClientIP` and `None`. Defaults to `None`. For more info see Kubernetes reference
- `type` - (Optional) Determines how the service is exposed. Defaults to `ClusterIP`. Valid options are `ExternalName`, `ClusterIP`, `NodePort`, and `LoadBalancer`. `ExternalName` maps to the specified `external_name`. For more info see Kubernetes reference

» `port`

» Arguments

- `name` - (Optional) The name of this port within the service. All ports within the service must have unique names. Optional if only one ServicePort is defined on this service.
- `node_port` - (Optional) The port on each node on which this service is exposed when `type` is `NodePort` or `LoadBalancer`. Usually assigned by the system. If specified, it will be allocated to the service if unused or else creation of the service will fail. Default is to auto-allocate a port if the `type` of this service requires one. For more info see Kubernetes reference
- `port` - (Required) The port that will be exposed by this service.
- `protocol` - (Optional) The IP protocol for this port. Supports `TCP` and `UDP`. Default is `TCP`.
- `target_port` - (Optional) Number or name of the port to access on the pods targeted by the service. Number must be in the range 1 to 65535. This field is ignored for services with `cluster_ip = "None"`. For more info see Kubernetes reference

» Attributes

- `load_balancer_ingress` - A list containing ingress points for the load-balancer (only valid if `type = "LoadBalancer"`)

» `load_balancer_ingress`

» Attributes

- `ip` - IP which is set for load-balancer ingress points that are IP based (typically GCE or OpenStack load-balancers)
- `hostname` - Hostname which is set for load-balancer ingress points that are DNS based (typically AWS load-balancers)

» Import

Service can be imported using its namespace and name, e.g.

```
$ terraform import kubernetes_service.example default/terraform-name
```

» kubernetes_service_account

A service account provides an identity for processes that run in a Pod.

Read more at Kubernetes reference/

» Example Usage

```
resource "kubernetes_service_account" "example" {
  metadata {
    name = "terraform-example"
  }
  secret {
    name = "${kubernetes_secret.example.metadata.0.name}"
  }
}

resource "kubernetes_secret" "example" {
  metadata {
    name = "terraform-example"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard service account's metadata. For more info see Kubernetes reference
- `image_pull_secret` - (Optional) A list of references to secrets in the same namespace to use for pulling any images in pods that reference this Service Account. For more info see Kubernetes reference
- `secret` - (Optional) A list of secrets allowed to be used by pods running using this Service Account. For more info see Kubernetes reference
- `automount_service_account_token` - (Optional) Boolean, `true` to enable automatic mounting of the service account token

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the service account that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the service account. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the service account, must be unique. Cannot be updated. For more info see Kubernetes reference
- `namespace` - (Optional) Namespace defines the space within which name of the service account must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this service account that can be used by clients to determine when service account has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this service account.
- `uid` - The unique in time and space value for this service account. For more info see Kubernetes reference

» `image_pull_secret`

» Arguments

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» `secret`

» Arguments

- `name` - (Optional) Name of the referent. For more info see Kubernetes reference

» Attributes Reference

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

- `default_secret_name` - Name of the default secret, containing service account token, created & managed by the service.

» Import

Service account can be imported using the namespace and name, e.g.

```
$ terraform import kubernetes_service_account.example default/terraform-example
```

» kubernetes_stateful_set

Manages the deployment and scaling of a set of Pods , and provides guarantees about the ordering and uniqueness of these Pods.

Like a Deployment , a StatefulSet manages Pods that are based on an identical container spec. Unlike a Deployment, a StatefulSet maintains a sticky identity for each of their Pods. These pods are created from the same spec, but are not interchangeable: each has a persistent identifier that it maintains across any rescheduling.

A StatefulSet operates under the same pattern as any other Controller. You define your desired state in a StatefulSet object, and the StatefulSet controller makes any necessary updates to get there from the current state.

» Example Usage

```
resource "kubernetes_stateful_set" "prometheus" {
  metadata {
    annotations = {
      SomeAnnotation = "foobar"
    }

    labels = {
      k8s-app                            = "prometheus"
      "kubernetes.io/cluster-service"    = "true"
      "addonmanager.kubernetes.io/mode" = "Reconcile"
      version                            = "v2.2.1"
    }

    name = "prometheus"
  }

  spec {
    pod_management_policy  = "Parallel"
    replicas               = 1
    revision_history_limit = 5

    selector {
      match_labels = {
        k8s-app = "prometheus"
      }
    }

    service_name = "prometheus"

    template {
      metadata {
        labels = {
          k8s-app = "prometheus"
        }

        annotations = {}
      }

      spec {
        service_account_name = "prometheus"

        init_container {
          name              = "init-chown-data"
```

```
  image             = "busybox:latest"
  image_pull_policy = "IfNotPresent"
  command           = ["chown", "-R", "65534:65534", "/data"]

  volume_mount {
    name       = "prometheus-data"
    mount_path = "/data"
    sub_path   = ""
  }
}

container {
  name              = "prometheus-server-configmap-reload"
  image             = "jimmidyson/configmap-reload:v0.1"
  image_pull_policy = "IfNotPresent"

  args = [
    "--volume-dir=/etc/config",
    "--webhook-url=http://localhost:9090/-/reload",
  ]

  volume_mount {
    name       = "config-volume"
    mount_path = "/etc/config"
    read_only  = true
  }

  resources {
    limits {
      cpu    = "10m"
      memory = "10Mi"
    }

    requests {
      cpu    = "10m"
      memory = "10Mi"
    }
  }
}

container {
  name              = "prometheus-server"
  image             = "prom/prometheus:v2.2.1"
  image_pull_policy = "IfNotPresent"

  args = [
```

```
  "--config.file=/etc/config/prometheus.yml",
  "--storage.tsdb.path=/data",
  "--web.console.libraries=/etc/prometheus/console_libraries",
  "--web.console.templates=/etc/prometheus/consoles",
  "--web.enable-lifecycle",
]

port {
  container_port = 9090
}

resources {
  limits {
    cpu    = "200m"
    memory = "1000Mi"
  }

  requests {
    cpu    = "200m"
    memory = "1000Mi"
  }
}

volume_mount {
  name       = "config-volume"
  mount_path = "/etc/config"
}

volume_mount {
  name       = "prometheus-data"
  mount_path = "/data"
  sub_path   = ""
}

readiness_probe {
  http_get {
    path = "/-/ready"
    port = 9090
  }

  initial_delay_seconds = 30
  timeout_seconds       = 30
}

liveness_probe {
  http_get {
```

```
          path   = "/-/healthy"
          port   = 9090
          scheme = "HTTPS"
        }

        initial_delay_seconds = 30
        timeout_seconds       = 30
      }
    }

    termination_grace_period_seconds = 300

    volume {
      name = "config-volume"

      config_map {
        name = "prometheus-config"
      }
    }
  }
}

update_strategy {
  type = "RollingUpdate"

  rolling_update {
    partition = 1
  }
}

volume_claim_template {
  metadata {
    name = "prometheus-data"
  }

  spec {
    access_modes       = ["ReadWriteOnce"]
    storage_class_name = "standard"

    resources {
      requests = {
        storage = "16Gi"
      }
    }
  }
}
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard Kubernetes object metadata. For more info see Kubernetes reference
- `spec` - (Required) Spec defines the specification of the desired behavior of the stateful set. For more info see Kubernetes reference

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the stateful set that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the stateful set. **Must match `selector`. By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the stateful set, must be unique. Cannot be updated. For more info see Kubernetes reference

- `namespace` - (Optional) Namespace defines the space within which name of the stateful set must be unique.

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this stateful set that can be used by clients to determine when stateful set has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this stateful set.
- `uid` - The unique in time and space value for this stateful set. For more info see Kubernetes reference

» `spec`

» Arguments

- `pod_management_policy` - (Optional) podManagementPolicy controls how pods are created during initial scale up, when replacing pods on nodes, or when scaling down. The default policy is `OrderedReady`, where pods are created in increasing order (pod-0, then pod-1, etc) and the controller will wait until each pod is ready before continuing. When scaling down, the pods are removed in the opposite order. The alternative policy is `Parallel` which will create pods in parallel to match the desired scale without waiting, and on scale down will delete all pods at once. *Changing this forces a new resource to be created.*

- `replicas` - (Optional) The desired number of replicas of the given Template. These are replicas in the sense that they are instantiations of the same Template, but individual replicas also have a consistent identity. If unspecified, defaults to 1.

- `revision_history_limit` - (Optional) The maximum number of revisions that will be maintained in the StatefulSet's revision history. The revision history consists of all revisions not represented by a currently applied StatefulSetSpec version. The default value is 10. *Changing this forces a new resource to be created.*

- `selector` - (Required) A label query over pods that should match the replica count. **It must match the pod template's labels.** *Changing this forces a new resource to be created.* More info: Kubernetes reference

- `service_name` - (Required) The name of the service that governs this StatefulSet. This service must exist before the StatefulSet, and is responsible for the network identity of the set. Pods get DNS/hostnames that follow the pattern: pod-specific-string.serviceName.default.svc.cluster.local

where "pod-specific-string" is managed by the StatefulSet controller. *Changing this forces a new resource to be created.*

- `template` - (Required) The object that describes the pod that will be created if insufficient replicas are detected. Each pod stamped out by the StatefulSet will fulfill this Template, but have a unique identity from the rest of the StatefulSet.
- `update_strategy` - (Optional) Indicates the StatefulSet update strategy that will be employed to update Pods in the StatefulSet when a revision is made to Template.
- `volume_claim_template` - (Optional) A list of volume claims that pods are allowed to reference. A claim in this list takes precedence over any volumes in the template, with the same name. *Changing this forces a new resource to be created.*

» Nested Blocks

» `spec.template`

» Arguments

- `metadata` - (Required) Standard object's metadata. More info: Kubernetes reference
- `spec` - (Optional) Specification of the desired behavior of the pod. More info: Kubernetes reference

» Nested Blocks

» `spec.template.spec`

» Arguments

These arguments are the same as the for the `spec` block of a Pod.

Please see the Pod resource for reference.

» Nested Blocks

» `spec.update_strategy`

» Arguments

- `type` - (Optional) Indicates the type of the StatefulSetUpdateStrategy. There are two valid update strategies, RollingUpdate and OnDelete. Default is `RollingUpdate`.
- `rolling_update` - (Optional) The RollingUpdate update strategy will update all Pods in a StatefulSet, in reverse ordinal order, while respecting the StatefulSet guarantees.

» `spec.update_strategy.rolling_update`

» Arguments

- `partition` - (Optional) Indicates the ordinal at which the StatefulSet should be partitioned. You can perform a phased roll out (e.g. a linear, geometric, or exponential roll out) using a partitioned rolling update in a similar manner to how you rolled out a canary. To perform a phased roll out, set the partition to the ordinal at which you want the controller to pause the update. By setting the partition to 0, you allow the StatefulSet controller to continue the update process. Default value is `0`.

» Nested Blocks

» `spec.volume_claim_template`

One or more `volume_claim_template` blocks can be specified.

» Arguments

Each takes the same attibutes as a `kubernetes_persistent_volume_claim` resource.

Please see its documentation for reference.

» kubernetes_storage_class

Storage class is the foundation of dynamic provisioning, allowing cluster administrators to define abstractions for the underlying storage platform.

Read more at https://kubernetes.io/blog/2017/03/dynamic-provisioning-and-storage-classes-kubernetes/

» Example Usage

```
resource "kubernetes_storage_class" "example" {
  metadata {
    name = "terraform-example"
  }
  storage_provisioner = "kubernetes.io/gce-pd"
  reclaim_policy      = "Retain"
  parameters = {
    type = "pd-standard"
  }
}
```

» Argument Reference

The following arguments are supported:

- `metadata` - (Required) Standard storage class's metadata. For more info see Kubernetes reference
- `parameters` - (Optional) The parameters for the provisioner that should create volumes of this storage class. Read more about available parameters.
- `storage_provisioner` - (Required) Indicates the type of the provisioner
- `reclaim_policy` - (Optional) Indicates the reclaim policy to use. If no reclaimPolicy is specified when a StorageClass object is created, it will default to Delete.
- `allow_volume_expansion` - (Optional) Indicates whether the storage class allow volume expand, default true

» Nested Blocks

» `metadata`

» Arguments

- `annotations` - (Optional) An unstructured key value map stored with the storage class that may be used to store arbitrary metadata. **By default, the provider ignores any annotations whose key names end with *kubernetes.io*. This is necessary because such annotations can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such annotations in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference

- `generate_name` - (Optional) Prefix, used by the server, to generate a unique name ONLY IF the `name` field has not been provided. This value will also be combined with a unique suffix. For more info see Kubernetes reference
- `labels` - (Optional) Map of string keys and values that can be used to organize and categorize (scope and select) the storage class. May match selectors of replication controllers and services. **By default, the provider ignores any labels whose key names end with *kubernetes.io*. This is necessary because such labels can be mutated by server-side components and consequently cause a perpetual diff in the Terraform plan output. If you explicitly specify any such labels in the configuration template then Terraform will consider these as normal resource attributes and manage them as expected (while still avoiding the perpetual diff problem).** For more info see Kubernetes reference
- `name` - (Optional) Name of the storage class, must be unique. Cannot be updated. For more info see Kubernetes reference

» Attributes

- `generation` - A sequence number representing a specific generation of the desired state.
- `resource_version` - An opaque value that represents the internal version of this storage class that can be used by clients to determine when storage class has changed. For more info see Kubernetes reference
- `self_link` - A URL representing this storage class.
- `uid` - The unique in time and space value for this storage class. For more info see Kubernetes reference

» Import

kubernetes_storage_class can be imported using its name, e.g.

```
$ terraform import kubernetes_storage_class.example terraform-example
```