» Resource: helm_release

A Release is an instance of a chart running in a Kubernetes cluster. A Chart is a Helm package. It contains all of the resource definitions necessary to run an application, tool, or service inside of a Kubernetes cluster.

`helm_release` describes the desired status of a chart in a kubernetes cluster.

» Example Usage

```
data "helm_repository" "stable" {
  name = "stable"
  url  = "https://kubernetes-charts.storage.googleapis.com"
}

resource "helm_release" "example" {
  name       = "my-redis-release"
  repository = data.helm_repository.stable.metadata[0].name
  chart      = "redis"
  version    = "6.0.1"

  values = [
    "${file("values.yaml")}"
  ]

  set {
    name  = "cluster.enabled"
    value = "true"
  }

  set {
    name  = "metrics.enabled"
    value = "true"
  }

  set_string {
    name  = "service.annotations.prometheus\\.io/port"
    value = "9127"
  }
}
```

» Example Usage - Local Chart

In case a Chart is not available from a repository, a path may be used:

```
resource "helm_release" "local" {
  name       = "my-local-chart"
  chart      = "./charts/example"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Release name.
- `chart` - (Required) Chart name to be installed. A path may be used.
- `repository` - (Optional) Repository where to locate the requested chart. If is an URL the chart is installed without installing the repository.
- `repository_key_file` - (Optional) The repositories cert key file
- `repository_cert_file` - (Optional) The repositories cert file
- `repository_ca_file` - (Optional) The Repositories CA File.
- `repository_username` - (Optional) Username for HTTP basic authentication against the repository.
- `repository_password` - (Optional) Password for HTTP basic authentication against the reposotory.
- `devel` - (Optional) Use chart development versions, too. Equivalent to version '>0.0.0-0'. If version is set, this is ignored.
- `version` - (Optional) Specify the exact chart version to install. If this is not specified, the latest version is installed.
- `namespace` - (Optional) The namespace to install the release into. Defaults to `default`
- `verify` - (Optional) Verify the package before installing it. Defaults to `false`
- `keyring` - (Optional) Location of public keys used for verification. Used only if `verify` is true. Defaults to `/.gnupg/pubring.gpg` in the location set by `home`
- `timeout` - (Optional) Time in seconds to wait for any individual kubernetes operation (like Jobs for hooks). Defaults to `300` seconds.
- `disable_webhooks` - (Optional) Prevent hooks from running. Defauts to `false`
- `reuse_values` - (Optional) When upgrading, reuse the last release's values and merge in any overrides. If 'reset_values' is specified, this is ignored. Defaults to `false`.
- `reset_values` - (Optional) When upgrading, reset the values to the ones built into the chart. Defaults to `false`.
- `force_update` - (Optional) Force resource update through delete/recreate if needed. Defaults to `false`.
- `recreate_pods` - (Optional) Perform pods restart during upgrade/rollback. Defaults to `false`.
- `cleanup_on_fail` - (Optional) Allow deletion of new resources created in

this upgrade when upgrade fails. Defaults to `false`.
- `max_history` - (Optional) Maximum number of release versions stored per release. Defaults to `0` (no limit).
- `atomic` - (Optional) If set, installation process purges chart on fail. The wait flag will be set automatically if atomic is used. Defaults to `false`.
- `skip_crds` - (Optional) If set, no CRDs will be installed. By default, CRDs are installed if not already present. Defaults to `false`.
- `render_subchart_notes` - (Optional) If set, render subchart notes along with the parent. Defaults to `true`.
- `disable_openapi_validation` - (Optional) If set, the installation process will not validate rendered templates against the Kubernetes OpenAPI Schema. Defaults to `false`.
- `wait` - (Optional) Will wait until all resources are in a ready state before marking the release as successful. It will wait for as long as `timeout`. Defaults to `true`.
- `values` - (Optional) List of values in raw yaml to pass to helm. Values will be merged, in order, as Helm does with multiple `-f` options.
- `set` - (Optional) Value block with custom values to be merged with the values yaml.
- `set_sensitive` - (Optional) Value block with custom sensitive values to be merged with the values yaml that won't be exposed in the plan's diff.
- `set_string` - (Optional) Value block with custom STRING values to be merged with the values yaml.
- `dependency_update` - (Optional) Runs helm dependency update before installing the chart. Defaults to `false`.
- `replace` - (Optional) Re-use the given name, even if that name is already used. This is unsafe in production. Defaults to `false`.
- `description` - (Optional) Set release description attribute (visible in the history).
- `postrender` - (Optional) Configure a command to run after helm renders the manifest which can alter the manifest contents.

The `set`, `set_sensitive` and `set_strings` blocks support:

- `name` - (Required) full name of the variable to be set.
- `value` - (Required) value of the variable to be set.

The `postrender` block supports a single attribute:

- `binary_path` - (Required) relative or full path to command binary.

» Attributes Reference

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

- `metadata` - Block status of the deployed release.

The `metadata` block supports:

- `chart` - The name of the chart.
- `name` - Name is the name of the release.
- `namespace` - Namespace is the kubernetes namespace of the release.
- `revision` - Version is an int32 which represents the version of the release.
- `status` - Status of the release.
- `version` - A SemVer 2 conformant version string of the chart.
- `values` - The compounded values from `values` and `set*` attributes.

» Import

A Helm Release resource can be imported using its namespace and name e.g.

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

NOTE: Since the `repository` attribute is not being persisted as metadata by helm, it will not be set to any value by default. All other provider specific attributes will be set to their default values and they can be overriden after running `apply` using the resource definition configuration.

» Data Source: helm_repository

A chart repository is a location where packaged charts can be stored and shared.

`helm_repository` describes a helm repository.

» Example Usage

```
data "helm_repository" "incubator" {
  name = "incubator"
  url  = "https://kubernetes-charts-incubator.storage.googleapis.com"
}

resource "helm_release" "my_cache" {
  name       = "my-cache"
  repository = data.helm_repository.incubator.metadata[0].name
  chart      = "redis-cache"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Chart repository name.
- `url` - (Required) Chart repository URL.
- `key_file` - (Optional) Identify HTTPS client using this SSL key file
- `cert_file` - (Optional) Identify HTTPS client using this SSL certificate file.
- `ca_file` - (Optional) Verify certificates of HTTPS-enabled servers using this CA bundle
- `username` - (Optional) Username for HTTP basic authentication.
- `password` - (Optional) Password for HTTP basic authentication.

» Attributes Reference

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

- `metadata` - Status of the deployed release.

The `metadata` block supports:

- `name` - Name of the repository read from the home.
- `url` - URL of the repository read from the home.

» Old resource helm_repository

Before 0.9.0 `helm_repository` was a resource and not a data source. The old resource is now a shim to the data source to preserve backwards compatibility. As the use of the resource is deprecated it is strongly suggested to move to the new data source as the compatibility will be removed in a future release.