» heroku_addon

Provides a Heroku Add-On resource. These can be attach services to a Heroku app.

» Example Usage

```
# Create a new Heroku app
resource "heroku_app" "default" {
  name = "test-app"
}

# Create a database, and configure the app to use it
resource "heroku_addon" "database" {
  app  = "${heroku_app.default.name}"
  plan = "heroku-postgresql:hobby-basic"
}

# Add a web-hook addon for the app
resource "heroku_addon" "webhook" {
  app  = "${heroku_app.default.name}"
  plan = "deployhooks:http"

  config {
    url = "http://google.com"
  }
}
```

» Argument Reference

The following arguments are supported:

- `app` - (Required) The Heroku app to add to.
- `plan` - (Required) The addon to add.
- `config` - (Optional) Optional plan configuration.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the add-on
- `name` - The add-on name
- `plan` - The plan name
- `provider_id` - The ID of the plan provider

- `config_vars` - The Configuration variables of the add-on

» Import

Addons can be imported using the Addon `id`, e.g.

```
$ terraform import heroku_addon.foobar 12345678
```

» heroku_addon_attachment

Attaches a Heroku Addon Resource to an additional Heroku App.

» Example Usage

```
resource "heroku_addon_attachment" "database" {
  app_id  = "${heroku_app.default.id}"
  addon_id = "${heroku_addon.database.id}"
}
```

» Argument Reference

The following arguments are supported:

- `app_id` - (Required) The ID of the Heroku App to attach to.
- `addon_id` - (Required) The ID of the existing Heroku Addon to attach.
- `name` - (Optional) A friendly name for the Heroku Addon Attachment.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID of the add-on attachment

» Import

Addons can be imported using the unique Addon Attachment `id`, e.g.

```
$ terraform import heroku_addon_attachment.foobar 01234567-89ab-cdef-0123-456789abcdef
```

» heroku_app

Provides a Heroku App resource. This can be used to create and manage applications on Heroku.

» Example Usage

```
# Create a new Heroku app
resource "heroku_app" "default" {
  name   = "my-cool-app"
  region = "us"

  config_vars {
    FOOBAR = "baz"
  }

  buildpacks = [
    "heroku/go"
  ]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the application. In Heroku, this is also the unique ID, so it must be unique and have a minimum of 3 characters.
- `region` - (Required) The region that the app should be deployed in.
- `stack` - (Optional) The application stack is what platform to run the application in.
- `buildpacks` - (Optional) Buildpack names or URLs for the application. Buildpacks configured externally won't be altered if this is not present.
- `config_vars` - (Optional) Configuration variables for the application. The config variables in this map are not the final set of configuration variables, but rather variables you want present. That is, other configuration variables set externally won't be removed by Terraform if they aren't present in this list.
- `space` - (Optional) The name of a private space to create the app in.
- `organization` - (Optional) A block that can be specified once to define organization settings for this app. The fields for this block are documented below.

The `organization` block supports:

- `name` (string) - The name of the organization.

- `locked` (boolean)
- `personal` (boolean)

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the app. This is also the name of the application.
- `name` - The name of the application. In Heroku, this is also the unique ID.
- `stack` - The application stack is what platform to run the application in.
- `space` - The private space the app should run in.
- `region` - The region that the app should be deployed in.
- `git_url` - The Git URL for the application. This is used for deploying new versions of the app.
- `web_url` - The web (HTTP) URL that the application can be accessed at by default.
- `heroku_hostname` - A hostname for the Heroku application, suitable for pointing DNS records.
- `all_config_vars` - A map of all of the configuration variables that exist for the app, containing both those set by Terraform and those set externally.

» Import

Apps can be imported using the App `id`, e.g.

```
$ terraform import heroku_app.foobar MyApp
```

» heroku_app_feature

Provides a Heroku App Feature resource. This can be used to create and manage App Features on Heroku.

» Example Usage

```
resource "heroku_app_feature" "log_runtime_metrics" {
  app = "test-app"
  name = "log-runtime-metrics"
}
```

» Argument Reference

The following arguments are supported:

- `app` - (Required) The Heroku app to link to.
- `name` - (Required) The name of the App Feature to manage.
- `enabled` - (Optional) Whether to enable or disable the App Feature. The default value is true.

» heroku_cert

Provides a Heroku SSL certificate resource. It allows to set a given certificate for a Heroku app.

» Example Usage

```
# Create a new Heroku app
resource "heroku_app" "default" {
  name = "test-app"
}

# Add-on SSL to application
resource "heroku_addon" "ssl" {
  app  = "${heroku_app.default.name}"
  plan = "ssl"
}

# Establish certificate for a given application
resource "heroku_cert" "ssl_certificate" {
  app               = "${heroku_app.default.name}"
  certificate_chain = "${file("server.crt")}"
  private_key       = "${file("server.key")}"
  depends_on        = "heroku_addon.ssl"
}
```

» Argument Reference

The following arguments are supported:

- `app` - (Required) The Heroku app to add to.
- `certificate_chain` - (Required) The certificate chain to add
- `private_key` - (Required) The private key for a given certificate chain

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the add-on
- `cname` - The CNAME for the SSL endpoint
- `name` - The name of the SSL certificate

» Importing

When importing a Heroku cert resource, the ID must be built using the app name colon the unique ID from the Heroku API. For an app named `production-api` with a certificate ID of `b85d9224-310b-409b-891e-c903f5a40568`, you would import it as: `$ terraform import heroku_cert.production_api production-api:b85d9224-310b-409b-891e-c903f5a40568`.

» heroku_domain

Provides a Heroku App resource. This can be used to create and manage applications on Heroku.

» Example Usage

```
# Create a new Heroku app
resource "heroku_app" "default" {
  name = "test-app"
}

# Associate a custom domain
resource "heroku_domain" "default" {
  app      = "${heroku_app.default.name}"
  hostname = "terraform.example.com"
}
```

» Argument Reference

The following arguments are supported:

- `hostname` - (Required) The hostname to serve requests from.
- `app` - (Required) The Heroku app to link to.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the of the domain record.
- `hostname` - The hostname traffic will be served as.
- `cname` - The CNAME traffic should route to.

» Importing

When importing a Heroku domain resource, the ID must be built using the app name colon the unique ID from the Heroku API. For an app named `production-api` with a domain ID of `b85d9224-310b-409b-891e-c903f5a40568`, you would import it as: `$ terraform import heroku_domain.production_api production-api:b85d9224-310b-409b-891e-c903f5a40568`.

» heroku_drain

Provides a Heroku Drain resource. This can be used to create and manage Log Drains on Heroku.

» Example Usage

```
resource "heroku_drain" "default" {
  app = "test-app"
  url = "syslog://terraform.example.com:1234"
}
```

» Argument Reference

The following arguments are supported:

- `url` - (Required) The URL for Heroku to drain your logs to.
- `app` - (Required) The Heroku app to link to.

» Attributes Reference

The following attributes are exported:

- `token` - The unique token for your created drain.

» Importing

When importing a Heroku drain resource, the ID must be built using the app name colon the unique ID from the Heroku API. For an app named `production-api` with a drain ID of `b85d9224-310b-409b-891e-c903f5a40568`, you would import it as: `$ terraform import heroku_drain.production_api production-api:b85d9224-310b-409b-891e-c903f5a40568`

» heroku_pipeline

Provides a Heroku Pipeline resource.

A pipeline is a group of Heroku apps that share the same codebase. Once a pipeline is created, and apps are added to different stages using `heroku_pipeline_coupling`, you can promote app slugs to the next stage.

» Example Usage

```
# Create Heroku apps for staging and production
resource "heroku_app" "staging" {
  name = "test-app-staging"
}

resource "heroku_app" "production" {
  name = "test-app-production"
}

# Create a Heroku pipeline
resource "heroku_pipeline" "test-app" {
  name = "test-app"
}

# Couple apps to different pipeline stages
resource "heroku_pipeline_coupling" "staging" {
  app      = "${heroku_app.staging.name}"
  pipeline = "${heroku_pipeline.test-app.id}"
  stage    = "staging"
}

resource "heroku_pipeline_coupling" "production" {
  app      = "${heroku_app.production.name}"
  pipeline = "${heroku_pipeline.test-app.id}"
  stage    = "production"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the pipeline.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of the pipeline.
- `name` - The name of the pipeline.

» Import

Pipelines can be imported using the Pipeline `id`, e.g.

```
$ terraform import heroku_pipeline.foobar 12345678
```

» heroku_pipeline_coupling

Provides a Heroku Pipeline Coupling resource.

A pipeline is a group of Heroku apps that share the same codebase. Once a pipeline is created using `heroku_pipeline`, and apps are added to different stages using `heroku_pipeline_coupling`, you can promote app slugs to the downstream stages.

» Example Usage

```
# Create Heroku apps for staging and production
resource "heroku_app" "staging" {
  name = "test-app-staging"
}

resource "heroku_app" "production" {
  name = "test-app-production"
}

# Create a Heroku pipeline
resource "heroku_pipeline" "test-app" {
  name = "test-app"
}
```

```
# Couple apps to different pipeline stages
resource "heroku_pipeline_coupling" "staging" {
  app      = "${heroku_app.staging.name}"
  pipeline = "${heroku_pipeline.test-app.id}"
  stage    = "staging"
}

resource "heroku_pipeline_coupling" "production" {
  app      = "${heroku_app.production.name}"
  pipeline = "${heroku_pipeline.test-app.id}"
  stage    = "production"
}
```

» Argument Reference

The following arguments are supported:

- `app` - (Required) The name of the app for this coupling.
- `pipeline` - (Required) The ID of the pipeline to add this app to.
- `stage` - (Required) The stage to couple this app to. Must be one of `review`, `development`, `staging`, or `production`.

» Attributes Reference

The following attributes are exported:

- `id` - The UUID of this pipeline coupling.
- `app` - The name of the application.
- `app_id` - The ID of the application.
- `pipeline` - The UUID of the pipeline.
- `stage` - The stage for this coupling.

» Import

Pipeline couplings can be imported using the Pipeline coupling `id`, e.g.

```
$ terraform import heroku_pipeline_coupling.foobar 12345678
```

» heroku_space

Provides a Heroku Space resource for running apps in isolated, highly available, secure app execution environments.

» Example Usage

```
// Create a new Heroku space
resource "heroku_space" "default" {
  name = "test-space"
  organization = "my-company"
  region = "virginia"
}

// Create a new Heroku app in test-space
resource "heroku_app" "default" {
  name = "test-app"
  space = "${heroku_space.default.name}"
  organization = {
    name = "my-company"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the space.
- `organization` - (Required) The name of the organization which will own the space.
- `region` - (Optional) The region that the space should be created in.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the space.
- `name` - The space's name.
- `organization` - The space's organization.
- `region` - The space's region.

» Import

Spaces can be imported using the space `id`, e.g.

```
$ terraform import heroku_space.foobar MySpace
```