

## » **chef\_\_data\_\_bag**

A data bag is a collection of configuration objects that are stored as JSON in Chef Server and can be retrieved and used in Chef recipes.

This resource creates the data bag itself. Inside each data bag is a collection of items which can be created using the `chef_data_bag_item` resource.

### » **Example Usage**

```
resource "chef_data_bag" "example" {  
  name = "example-data-bag"  
}
```

### » **Argument Reference**

The following arguments are supported:

- **name** - (Required) The unique name to assign to the data bag. This is the name that other server clients will use to find and retrieve data from the data bag.

### » **Attributes Reference**

The following attributes are exported:

- **api\_url** - The URL representing this data bag in the Chef server API.

## » **chef\_data\_bag\_item**

A data bag is a collection of configuration objects that are stored as JSON in Chef Server and can be retrieved and used in Chef recipes.

This resource creates objects within an existing data bag. To create the data bag itself, use the `chef_data_bag` resource.

### » **Example Usage**

```
resource "chef_data_bag_item" "example" {  
  data_bag_name = "example-data-bag"  
  
  content_json = <<EOT
```

```
{
  "id": "example-item",
  "any_arbitrary_data": true
}
EOT
}
```

## » Argument Reference

The following arguments are supported:

- **data\_bag\_name** - (Required) The name of the data bag into which this item will be placed.
- **content\_json** - (Required) A string containing a JSON object that will be the content of the item. Must at minimum contain a property called "id" that is unique within the data bag, which will become the identifier of the created item.

## » Attributes Reference

The following attributes are exported:

- **id** - The value of the "id" property in the `content_json` JSON object, which can be used by clients to retrieve this item's content.

## » chef\_environment

An environment is a container for Chef nodes that share a set of attribute values and may have a set of version constraints for which cookbook versions may be used on its nodes.

## » Example Usage

```
resource "chef_environment" "example" {
  name = "example-environment"
}
```

## » Argument Reference

The following arguments are supported:

- **name** - (Required) The unique name to assign to the environment. This name will be used when nodes are created within the environment.
- **description** - (Optional) A human-friendly description of the environment. If not set, a placeholder of "Managed by Terraform" will be set.
- **default\_attributes\_json** - (Optional) String containing a JSON-serialized object containing the default attributes for the environment.
- **override\_attributes\_json** - (Optional) String containing a JSON-serialized object containing the override attributes for the environment.
- **cookbook\_constraints** - (Optional) Mapping of cookbook names to cookbook version constraints that should apply for this environment.

## » Attributes Reference

This resource exports no further attributes.

## » chef\_node

A node is a computer whose configuration is managed by Chef.

Although this resource allows a node to be registered, it does not actually configure the computer in question to interact with Chef. In most cases it is better to use the **chef** provisioner to configure the Chef client on a computer and have it register itself with the Chef server.

## » Example Usage

```
resource "chef_node" "example" {
  name          = "example-environment"
  environment_name = "${chef_environment.example.name}"
  run_list      = ["recipe[example]", "role[app_server]"]
}
```

## » Argument Reference

The following arguments are supported:

- **name** - (Required) The unique name to assign to the node.
- **automatic\_attributes\_json** - (Optional) String containing a JSON-serialized object containing the automatic attributes for the node.
- **normal\_attributes\_json** - (Optional) String containing a JSON-serialized object containing the normal attributes for the node.

- `default_attributes_json` - (Optional) String containing a JSON-serialized object containing the default attributes for the node.
- `override_attributes_json` - (Optional) String containing a JSON-serialized object containing the override attributes for the node.
- `run_list` - (Optional) List of strings to set as the run list for the node.

## » Attributes Reference

This resource exports no further attributes.

## » `chef_role`

A role is a set of standard configuration that can apply across multiple nodes that perform the same function.

## » Example Usage

```
resource "chef_role" "example" {
  name      = "example-role"
  run_list = ["recipe[example]"]
}
```

## » Argument Reference

The following arguments are supported:

- `name` - (Required) The unique name to assign to the role.
- `description` - (Optional) A human-friendly description of the role. If not set, a placeholder of "Managed by Terraform" will be set.
- `default_attributes_json` - (Optional) String containing a JSON-serialized object containing the default attributes for the role.
- `override_attributes_json` - (Optional) String containing a JSON-serialized object containing the override attributes for the role.
- `run_list` - (Optional) List of strings to set as the run list for any nodes that belong to this role.

## » Attributes Reference

This resource exports no further attributes.