» runscope_bucket

Use this data source to get information about a specific bucket that you can use with other runscope resources.

» Example Usage

```
data "runscope_bucket" "website" {
    key = "t2f4bkvnggct"
}

resource "runscope_environment" "environment" {
    bucket_id = "${runscope_bucket.website.id}"
    name      = "test-environment"
}
```

» Argument Reference

The following arguments are supported:

- `key` - (Required) The unique key of the bucket.

» Attributes Reference

The following attributes are exported:

- `id` - The unique key of the found bucket.
- `team_uuid` - The team unique identifier that owns the bucket.
- `name` - Type name of the bucket.

» runscope_buckets

Use this data source to get information about matching buckets that you can use with other runscope resources.

» Example Usage

```
data "runscope_buckets" "buckets" {
    filter = [
        {
            name = "name"
```

```
            values = ["test-bucket"]
        }
    ]
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) Filter to reduce the list of buckets returned.

Variables (`filter`) supports the following:

- `name` - The name of the field to filter on, currently either: `key`, `name`.
- `values` - The list of values to match against

» Attributes Reference

The following attributes are exported:

- `keys` - A list of the keys of matching buckets.

» runscope_integration

Use this data source to get information about a specific integration that you can with other runscope resources.

» Example Usage

```
data "runscope_integration" "pagerduty" {
  team_uuid = "d26553c0-3537-40a8-9d3c-64b0453262a9"
  type = "pagerduty"
}

resource "runscope_environment" "environment" {
  bucket_id    = "${runscope_bucket.bucket.id}"
  name         = "test-environment"

  integrations = [
    {
      id                 = "${data.runscope_integration.pagerduty.id}"
      integration_type = "pagerduty"
    }
```

```
    ]
}
```

» Argument Reference

The following arguments are supported:

- `team_uuid` - (Required) Your team unique identifier.
- `type` - (Required) Type of integration to lookup i.e. pagerduty

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier of the found integration.

» runscope_integration

Use this data source to list all of your integrations that you can use with other runscope resources.

» Example Usage

```
data "runscope_integrations" "slack" {
    team_uuid = "d26553c0-3537-40a8-9d3c-64b0453262a9"
    filter = {
        name = "type"
        values = ["slack"]
    }
}
```

» Argument Reference

The following arguments are supported:

- `filter` - (Optional) Filter to reduce the list of integrations returned.

Variables (`filter`) supports the following:

- `name` - The name of the field to filter on, currently either: `id`, `type` or `description`.
- `values` - The list of values to match against

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier of the found integration.

» runscope_bucket

A bucket resource. Buckets are a simple way to organize your requests and tests.

» Example Usage

```
# Add a bucket to your runscope account
resource "runscope_bucket" "main" {
  name      = "a-bucket"
  team_uuid = "870ed937-bc6e-4d8b-a9a5-d7f9f2412fa3"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (String, Required) The name of this bucket.
- `team_uuid` - (String, Required) Unique identifier for the team this bucket is being created for.

» Attributes Reference

The following attributes are exported:

- `name` - The name of this bucket.
- `team_uuid` - Unique identifier for the team this bucket belongs to.

» Import

Buckets can be imported using the bucket `key`, e.g.

```
$ terraform import runscope_bucket.example t2f4bkvnggcx
```

» runscope_environment

An environment resource. An environment is is a group of configuration settings (initial variables, locations, notifications, integrations, etc.) used when running a test. Every test has at least one environment, but you can create additional environments as needed. For common settings (base URLs, API keys) that you'd like to use across all tests within a bucket, use a Shared Environment.

» Creating a shared environment

> Note: to create a shared environment you do not include a `test_id`

```
resource "runscope_environment" "environment" {
  bucket_id    = "${runscope_bucket.bucket.id}"
  name         = "shared-environment"

  integrations = [
    "${data.runscope_integration.pagerduty.id}"
  ]

  initial_variables {
    var1 = "true",
    var2 = "value2"
  }
}

data "runscope_integration" "pagerduty" {
  team_uuid = "%s"
  type = "pagerduty"
}
```

» Creating a test environment

> Note: to create an environment specific to a test include the associated `test_id`

```
resource "runscope_environment" "environment" {
  bucket_id    = "${runscope_bucket.bucket.id}"
  test_id      = "${runscope_test.api.id}
  name         = "test-environment"

  integrations = [
    "${data.runscope_integration.pagerduty.id}"
  ]
```

```
  initial_variables {
    var1 = "true",
    var2 = "value2"
  }
}

data "runscope_integration" "pagerduty" {
  team_uuid = "194204f3-19a3-4ef7-a492-b14a277025da"
  type = "pagerduty"
}

# Add a test to a bucket
resource "runscope_test" "api" {
  name         = "api-test"
  description  = "checks the api is up and running"
  bucket_id    = "${runscope_bucket.main}"
}

# Create a bucket
resource "runscope_bucket" "main" {
  name         = "terraform-ftw"
  team_uuid    = "870ed937-bc6e-4d8b-a9a5-d7f9f2412fa3"
}
```

» Argument Reference

The following arguments are supported:

- `bucket_id` - (Required) The id of the bucket to associate this environment with.
- `test_id` - (Optional) The id of the test to associate this environment with. If given, creates a test specific environment, otherwise creates a shared environment.
- `name` - (Required) The name of environment.
- `script` - (Optional) The script to to run to setup the environment
- `preserve_cookies` - (Optional) If this is set to true, tests using this enviornment will manage cookies between steps.
- `initial_variables` - (Optional) Map of keys and values being used for variables when the test begins.
- `integrations` - (Optional) A list of integration ids to enable for test runs using this environment.
- `regions` - (Optional) A list of Runscope regions to execute test runs in when using this environment.
- `remote_agents` - (Optional) A list of Remote Agents to execute test runs in when using this environment. Remote Agents documented below.

6

Remote Agents (`remote_agents`) supports the following:

- `name` - (Required) The name of the remote agent
- `uuid` - (Required) The uuid of the remote agent

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the environment.

» runscope_test

A test resource. Tests are made up of a collection of test steps and an environment.

» Example Usage

```
# Add a test to a bucket
resource "runscope_test" "api" {
  name        = "api-test"
  description = "checks the api is up and running"
  bucket_id   = "${runscope_bucket.main}"
}

# Create a bucket
resource "runscope_bucket" "main" {
  name        = "terraform-ftw"
  team_uuid   = "870ed937-bc6e-4d8b-a9a5-d7f9f2412fa3"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (String, Required) The name of this test.
- `description` - (String, Optional) Human-readable description of the new test. is being created for.

» Attributes Reference

The following attributes are exported:

- `id` - The unique identifier for the test.
- `name` - The name of this test.
- `description` - Human-readable description of the new test.

» runscope_schedule

A schedule resource. Tests can be scheduled to run on frequencies up to every minute. One ore more schedules can be configured per test with each schedule using a unique Test-specific or Shared Environment.

» Creating a schedule

```
resource "runscope_schedule" "daily" {
  bucket_id      = "${runscope_bucket.bucket.id}"
  test_id        = "${runscope_test.test.id}"
  interval       = "1d"
  note           = "This is a daily schedule"
  environment_id = "${runscope_environment.environment.id}"
}

resource "runscope_test" "test" {
  bucket_id   = "${runscope_bucket.bucket.id}"
  name        = "runscope test"
  description = "This is a test test..."
}

resource "runscope_bucket" "bucket" {
  name      = "terraform-provider-test"
  team_uuid = "d038db69-b5a9-45af-80d8-3be47c37e309"
}

resource "runscope_environment" "environment" {
  bucket_id = "${runscope_bucket.bucket.id}"
  name      = "test-environment"

  initial_variables {
    var1 = "true",
    var2 = "value2"
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket_id` - (Required) The id of the bucket to associate this schedule with.
- `test_id` - (Required) The id of the test to associate this schedule with.
- `environment_id` - (Required) The id of the environment to use when running the test. If given, creates a test specific schedule, otherwise creates a shared schedule.
- `interval` - (Required) The schedule's interval, must be one of:
  - 1m — every minute
  - 5m — every 5 minutes
  - 15m — every 15 minutes
  - 30m — every 30 minutes
  - 1h — every hour
  - 6h — every 6 hours
  - 1d — every day.
- `note` - (Optional) A human-friendly description for the schedule.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the schedule.

» runscope_step

A step resource. API tests are comprised of a series of steps, most often HTTP requests. In addition to requests, you can also add additional types of steps to your tests like pauses and conditions.

» Creating a step

```
resource "runscope_step" "main_page" {
  bucket_id       = "${runscope_bucket.bucket.id}"
  test_id         = "${runscope_test.test.id}"
  step_type       = "request"
  url             = "http://example.com"
  method          = "GET"
  variables       = [
    {
      name        = "httpStatus"
```

```
      source   = "response_status"
  },
  {
      name      = "httpContentEncoding"
      source   = "response_header"
      property = "Content-Encoding"
  },
]
assertions    = [
  {
      source     = "response_status"
      comparison = "equal_number"
      value      = "200"
  },
  {
      source     = "response_json"
      comparison = "equal"
      value      = "c5baeb4a-2379-478a-9cda-1b671de77cf9",
      property   = "data.id"
  },
],
auth = {
      username  = "myUsername"
      auth_type = "basic"
      password  = "myPassword"
},
before_scripts  = [<<EOF
      var endVar = new Date();
      var startVar = new Date();
      alert('this is a multi-line before script')
  EOF
],
scripts         = [<<EOF
      var endVar = new Date();
      var startVar = new Date();
      alert('this is a multi-line after script')
  EOF
],
headers         = [
  {
      header = "Accept-Encoding",
      value  = "application/json"
  },
  {
      header = "Accept-Encoding",
      value  = "application/xml"
```

10

```
    },
    {
        header = "Authorization",
        value  = "Bearer bb74fe7b-b9f2-48bd-9445-bdc60e1edc6a",
    }
  ]
}

resource "runscope_test" "test" {
  bucket_id   = "${runscope_bucket.bucket.id}"
  name        = "runscope test"
  description = "This is a test test..."
}

resource "runscope_bucket" "bucket" {
  name      = "terraform-provider-test"
  team_uuid = "dfb75aac-eeb3-4451-8675-3a37ab421e4f"
}
```

» Argument Reference

The following arguments are supported:

- `bucket_id` - (Required) The id of the bucket to associate this step with.
- `test_id` - (Required) The id of the test to associate this step with.
- `step_type` - (Required) The type of step.
  - request
  - pause
  - condition
  - ghost
  - subtest

» Request steps

When creating a `request` type of step the additional arguments also apply:

- `method` - (Required) The HTTP method for this request step.
- `variables` - (Optional) A list of variables to extract out of the HTTP response from this request. Variables documented below.
- `assertions` - (Optional) A list of assertions to apply to the HTTP response from this request. Assertions documented below.
- `headers` - (Optional) A list of headers to apply to the request. Headers documented below.
- `body` - (Optional) A string to use as the body of the request.
- `auth` - (Optional) The credentials used to authenticate the request

- `before_script` - (Optional) Runs a script before the request is made
- `script` - (Optional) Runs a script after the request is made

Variables (`variables`) supports the following:

- `name` - (Required) Name of the variable to define.
- `property` - (Required) The name of the source property. i.e. header name or json path
- `source` - (Required) The variable source, for list of allowed values see: https://www.runscope.com/docs/api/steps#assertions

Assertions (`assertions`) supports the following:

- `source` - (Required) The assertion source, for list of allowed values see: https://www.runscope.com/docs/api/steps#assertions
- `property` - (Optional) The name of the source property. i.e. header name or json path
- `comparison` - (Required) The assertion comparison to make i.e. `equals`, for list of allowed values see: https://www.runscope.com/docs/api/steps# assertions
- `value` - (Optional) The value the `comparison` will use

Example Assertions

Status Code == 200

```
"assertions": [
    {
        "source": "response_status",
        "comparison": "equal_number",
        "value": 200
    }
]
```

JSON element 'address' contains the text "avenue"

```
"assertions": [
    {
        "source": "response_json",
        "property": "address",
        "comparison": "contains",
        "value": "avenue"
    }
]
```

Response Time is faster than 1 second.

```
"assertions": [
    {
        "source": "response_time",
        "comparison": "is_less_than",
```

```
        "value": 1000
    }
]
```

The `headers` list supports the following:

- `header` - (Required) The name of the header
- `value` - (Required) The name header value

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the step.