

New Relic Provider

New Relic (<https://newrelic.com/>) offers a performance management solution enabling developers to diagnose and fix application performance problems in real time.

Use the navigation to the left to read about the available resources.

Example Usage

```
# Configure the New Relic provider
provider "newrelic" {
  api_key = "${var.newrelic_api_key}"
}

# Create an alert policy
resource "newrelic_alert_policy" "alert" {
  name = "Alert"
}

# Add a condition
resource "newrelic_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.alert.id}"

  name      = "foo"
  type      = "apm_app_metric"
  entities  = ["12345"]
  metric    = "apdex"
  runbook_url = "https://docs.example.com/my-runbook"

  # You can look this up in New Relic

  term {
    duration      = 5
    operator      = "below"
    priority      = "critical"
    threshold     = "0.75"
    time_function = "all"
  }
}

# Add a notification channel
resource "newrelic_alert_channel" "email" {
  name = "email"
  type = "email"

  configuration = {
    recipients          = "paul@example.com"
    include_json_attachment = "1"
  }
}

# Link the channel to the policy
resource "newrelic_alert_policy_channel" "alert_email" {
  policy_id = "${newrelic_alert_policy.alert.id}"
  channel_id = "${newrelic_alert_channel.email.id}"
}
```

Argument Reference

The following arguments are supported:

- `api_key` - (Required) Your New Relic API key. Can also use `NEWRELIC_API_KEY` environment variable.

newrelic_alert_channel

Use this data source to get information about an specific alert channel in New Relic which already exists (e.g newrelic user).

Example Usage

```
data "newrelic_alert_channel" "foo" {
  name = "foo@example.com"
}

resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_alert_policy_channel" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"
  channel_id = "${newrelic_alert_channel.foo.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the alert channel in New Relic.

Attributes Reference

- `id` - The ID of the alert channel.
- `type` - Alert channel type, either: `campfire`, `email`, `hipchat`, `opsgenie`, `pagerduty`, `slack`, `victorops`, or `webhook`..
- `policy_ids` - A list of policy IDs associated with the alert channel.

newrelic_application

Use this data source to get information about a specific application in New Relic.

Example Usage

```
data "newrelic_application" "app" {
  name = "my-app"
}

resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "foo"
  type          = "apm_app_metric"
  entities      = ["${data.newrelic_application.app.id}"]
  metric        = "apdex"
  runbook_url   = "https://www.example.com"

  term {
    duration      = 5
    operator       = "below"
    priority       = "critical"
    threshold      = "0.75"
    time_function = "all"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the application in New Relic.

Attributes Reference

- `id` - The ID of the application.
- `instance_ids` - A list of instance IDs associated with the application.
- `host_ids` - A list of host IDs associated with the application.

newrelic_key_transaction

Use this data source to get information about a specific key transaction in New Relic.

Example Usage

```
data "newrelic_key_transaction" "txn" {
  name = "txn"
}

resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "foo"
  type          = "apm_kt_metric"
  entities      = ["${data.newrelic_key_transaction.txn.id}"]
  metric        = "error_percentage"
  runbook_url   = "https://www.example.com"

  term {
    duration      = 5
    operator      = "below"
    priority      = "critical"
    threshold     = "0.75"
    time_function = "all"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the application in New Relic.

Attributes Reference

- `id` - The ID of the application.

newrelic_synthetics_monitor

Use this data source to get information about a specific synthetics monitor in New Relic. This can then be used to set up a synthetics alert condition.

Example Usage

```
data "newrelic_synthetics_monitor" "bar" {
  name = "bar"
}

resource "newrelic_synthetics_alert_condition" "baz" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "baz"
  monitor_id    = "${data.newrelic_synthetics_monitor.bar.id}"
  runbook_url   = "https://www.example.com"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the synthetics monitor in New Relic.

Attributes Reference

- `id` - The ID of the synthetics monitor.

newrelic_alert_channel

Example Usage

```
resource "newrelic_alert_channel" "foo" {
  name = "foo"
  type = "email"

  configuration = {
    recipients          = "foo@example.com"
    include_json_attachment = "1"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the channel.
- `type` - (Required) The type of channel. One of: `campfire`, `email`, `hipchat`, `opsgenie`, `pagerduty`, `slack`, `victorops`, or `webhook`.
- `configuration` - (Required) A map of key / value pairs with channel type specific values.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the channel.

Import

Alert channels can be imported using the `id`, e.g.

```
$ terraform import newrelic_alert_channel.main 12345
```

newrelic_alert_condition

Example Usage

```
data "newrelic_application" "app" {
  name = "my-app"
}

resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "foo"
  type          = "apm_app_metric"
  entities      = ["${data.newrelic_application.app.id}"]
  metric        = "apdex"
  runbook_url   = "https://www.example.com"

  term {
    duration      = 5
    operator      = "below"
    priority      = "critical"
    threshold     = "0.75"
    time_function = "all"
  }
}
```

Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The ID of the policy where this condition should be used.
- `name` - (Required) The title of the condition
- `type` - (Required) The type of condition. One of: `apm_app_metric`, `apm_jvm_metric`, `apm_kt_metric`, `servers_metric`, `browser_metric`, `mobile_metric`
- `entities` - (Required) The instance IDS associated with this condition.
- `metric` - (Required) The metric field accepts parameters based on the `type` set.
- `gc_metric` - (Optional) A valid Garbage Collection metric e.g. `GC/G1 Young Generation`. This is required if you are using `apm_jvm_metric` with `gc_cpu_time` condition type.
- `violation_close_timer` - (Optional) Automatically close instance-based violations, including JVM health metric violations, after the number of hours specified. Must be: 1, 2, 4, 8, 12 or 24.
- `runbook_url` - (Optional) Runbook URL to display in notifications.
- `condition_scope` - (Optional) instance or application. This is required if you are using the JVM plugin in New Relic.

- `term` - (Required) A list of terms for this condition. See Terms below for details.
- `user_defined_metric` - (Optional) A custom metric to be evaluated.
- `user_defined_value_function` - (Optional) One of: `average`, `min`, `max`, `total`, or `sample_size`.

Terms

The term mapping supports the following arguments:

- `duration` - (Required) In minutes, must be: 5, 10, 15, 30, 60, or 120.
- `operator` - (Optional) `above`, `below`, or `equal`. Defaults to `equal`.
- `priority` - (Optional) `critical` or `warning`. Defaults to `critical`.
- `threshold` - (Required) Must be 0 or greater.
- `time_function` - (Required) `all` or `any`.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the alert condition.

Import

Alert conditions can be imported using the `id`, e.g.

```
$ terraform import newrelic_alert_condition.main 12345
```

newrelic_alert_policy

Example Usage

```
resource "newrelic_alert_policy" "foo" {  
  name = "foo"  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the policy.
- `incident_preference` - (Optional) The rollup strategy for the policy. Options include: `PER_POLICY`, `PER_CONDITION`, or `PER_CONDITION_AND_TARGET`. The default is `PER_POLICY`.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the policy.
- `created_at` - The time the policy was created.
- `updated_at` - The time the policy was last updated.

Import

Alert policies can be imported using the `id`, e.g.

```
$ terraform import newrelic_alert_policy.main 12345
```

newrelic_alert_policy_channel

Example Usage

```
resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_alert_channel" "foo" {
  name = "foo"
  type = "email"

  configuration = {
    recipients          = "foo@example.com"
    include_json_attachment = "1"
  }
}

resource "newrelic_alert_policy_channel" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"
  channel_id = "${newrelic_alert_channel.foo.id}"
}
```

Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The ID of the policy.
- `channel_id` - (Required) The ID of the channel.

newrelic_dashboard

Example Usage

```
resource "newrelic_dashboard" "exampledash" {
  title = "New Relic Terraform Example"

  widget {
    title      = "Average Transaction Duration"
    row        = 1
    column     = 1
    width      = 2
    visualization = "faceted_line_chart"
    nrql       = "SELECT AVERAGE(duration) from Transaction FACET appName TIMESERIES auto"
  }

  widget {
    title      = "Page Views"
    row        = 1
    column     = 3
    visualization = "billboard"
    nrql       = "SELECT count(*) FROM PageView SINCE 1 week ago"
  }
}
```

Argument Reference

The following arguments are supported:

- `title` - (Required) The title of the dashboard.
- `icon` - (Optional) The icon for the dashboard. Defaults to `bar-chart`.
- `visibility` - (Optional) Who can see the dashboard in an account. Must be `owner` or `all`. Defaults to `all`.
- `widget` - (Optional) A widget that describes a visualization. See [Widgets](#) below for details.
- `editable` - (Optional) Who can edit the dashboard in an account. Must be `read_only`, `editable_by_owner`, `editable_by_all`, or `all`. Defaults to `editable_by_all`.

Widgets

The widget mapping supports the following arguments:

- `title` - (Required) A title for the widget.
- `visualization` - (Required) How the widget visualizes data.
- `row` - (Required) Row position of widget from top left, starting at 1.
- `column` - (Required) Column position of widget from top left, starting at 1.

- `width` - (Optional) Width of the widget. Defaults to 1.
- `height` - (Optional) Height of the widget. Defaults to 1.
- `notes` - (Optional) Description of the widget.
- `nrql` - (Optional) Valid NRQL query string. See Writing NRQL Queries (<https://docs.newrelic.com/docs/insights/nrql-new-relic-query-language/using-nrql/introduction-nrql>) for help.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the dashboard.

newrelic_infra_alert_condition

Example Usage

```
resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_infra_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name      = "High disk usage"
  type      = "infra_metric"
  event     = "StorageSample"
  select    = "diskUsedPercent"
  comparison = "above"
  where     = "(`hostname` LIKE '%frontend%')"

  critical {
    duration      = 25
    value         = 90
    time_function = "all"
  }
}
```

Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The ID of the alert policy where this condition should be used.
- `name` - (Required) The Infrastructure alert condition's name.
- `enabled` - (Optional) Set whether to enable the alert condition. Defaults to `true`.
- `type` - (Required) The type of Infrastructure alert condition: "infra_process_running", "infra_metric", or "infra_host_not_reporting".
- `event` - (Required) The metric event; for example, system metrics, process metrics, storage metrics, or network metrics.
- `select` - (Required) The attribute name to identify the type of metric condition; for example, "network", "process", "system", or "storage".
- `comparison` - (Required) The operator used to evaluate the threshold value; "above", "below", "equal".
- `critical` - (Required) Identifies the critical threshold parameters for triggering an alert notification. See [Thresholds](#) below for details.
- `warning` - (Optional) Identifies the warning threshold parameters. See [Thresholds](#) below for details.
- `where` - (Optional) Infrastructure host filter for the alert condition.
- `process_where` - (Optional) Any filters applied to processes; for example: "commandName = 'java'".

- `integration_provider` - (Optional) For alerts on integrations, use this instead of event.

Thresholds

The critical and warning threshold mapping supports the following arguments:

- `duration` - (Required) Identifies the number of minutes the threshold must be passed or met for the alert to trigger. Threshold durations must be between 1 and 60 minutes (inclusive).
- `value` - (Optional) Threshold value, computed against the comparison operator. Supported by "infra_metric" and "infra_process_running" alert condition types.
- `time_function` - (Optional) Indicates if the condition needs to be sustained or to just break the threshold once; all or any. Supported by the "infra_metric" alert condition type.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Infrastructure alert condition.

newrelic_nrql_alert_condition

Example Usage

```
resource "newrelic_alert_policy" "foo" {
  name = "foo"
}

resource "newrelic_nrql_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "foo"
  runbook_url   = "https://www.example.com"
  enabled       = true

  term {
    duration      = 5
    operator       = "below"
    priority       = "critical"
    threshold      = "1"
    time_function = "all"
  }

  nrql {
    query         = "SELECT count(*) FROM SyntheticCheck WHERE monitorId = '<monitorId>'"
    since_value   = "3"
  }

  value_function = "single_value"
}
```

Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The ID of the policy where this condition should be used.
- `name` - (Required) The title of the condition
- `runbook_url` - (Optional) Runbook URL to display in notifications.
- `enabled` - (Optional) Set whether to enable the alert condition. Defaults to `true`.
- `term` - (Required) A list of terms for this condition. See Terms below for details.
- `nrql` - (Required) A NRQL query. See NRQL below for details.
- `value_function` - (Optional) Possible values are `single_value`, `sum`.

Terms

The term mapping supports the following arguments:

- `duration` - (Required) In minutes, must be: 1, 2, 3, 4, 5, 10, 15, 30, 60, or 120.
- `operator` - (Optional) above, below, or equal. Defaults to equal.
- `priority` - (Optional) critical or warning. Defaults to critical.
- `threshold` - (Required) Must be 0 or greater.
- `time_function` - (Required) all or any.

NRQL

The `nrql` attribute supports the following arguments: `* query` - (Required) The NRQL query to execute for the condition `* since_value` - (Required) The value to be used in the `SINCE <X> MINUTES AGO` clause for the NRQL query. Must be: 1, 2, 3, 4, or 5.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the NRQL alert condition.

Import

Alert conditions can be imported using the `id`, e.g.

```
$ terraform import newrelic_nrql_alert_condition.main 12345
```

newrelic_synthetics_alert_condition

Example Usage

```
data "newrelic_synthetics_monitor" "foo" {
  name = "foo"
}

resource "newrelic_synthetics_alert_condition" "foo" {
  policy_id = "${newrelic_alert_policy.foo.id}"

  name          = "foo"
  monitor_id    = "${data.newrelic_synthetics_monitor.foo.id}"
  runbook_url   = "https://www.example.com"
}
```

Argument Reference

The following arguments are supported:

- `policy_id` - (Required) The ID of the policy where this condition should be used.
- `name` - (Required) The title of this condition.
- `monitor_id` - (Required) The ID of the Synthetics monitor to be referenced in the alert condition.
- `runbook_url` - (Optional) Runbook URL to display in notifications.
- `enabled` - (Optional) Set whether to enable the alert condition. Defaults to `true`.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the Synthetics alert condition.