» opsgenie_user

Manages existing User within Opsgenie.

» Example Usage

```
data "opsgenie_user" "test" {
  username = "user@domain.com"
}
```

» Argument Reference

The following arguments are supported:

- `username` - (Required) The email address associated with this user. Opsgenie defines that this must not be longer than 100 characters.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie User.
- `full_name` - The Full Name of the User.
- `role` - The Role assigned to the User. Either a built-in such as 'Owner', 'Admin' or 'User' - or the name of a custom role.
- `locale` - Location information for the user. Please look at Supported Locale Ids for available locales.
- `timezone` - Timezone information of the user. Please look at Supported Timezone Ids for available timezones.

» opsgenie_team

Manages existing Team within Opsgenie.

» Example Usage

```
data "opsgenie_team" "sre-team" {
  name = "sre-team"
}
```

» Argument Reference

The following arguments are supported:

- `name` - The name associated with this team. Opsgenie defines that this must not be longer than 100 characters.

The following attributes are exported:

- `id` - The ID of the Opsgenie Team.
- `member` - A Member block as documented below.
- `description` - A description for this team.

» opsgenie_heartbeat

Manages existing heartbeat within Opsgenie.

» Example Usage

```
data "opsgenie_heartbeat" "test" {
  name = "genieheartbeat-existing"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the heartbeat

» Attributes Reference

The following attributes are exported:

- `description` - An optional description of the heartbeat
- `interval_unit` - Interval specified as minutes, hours or days.
- `interval` - Specifies how often a heartbeat message should be expected.
- `enabled` - Enable/disable heartbeat monitoring.
- `owner_team_id` - Owner team of the heartbeat.

- `alert_message` - Specifies the alert message for heartbeat expiration alert. If this is not provided, default alert message is "HeartbeatName is expired".
- `alert_priority` - Specifies the alert priority for heartbeat expiration alert. If this is not provided, default priority is P3.
- `alert_tags` - Specifies the alert tags for heartbeat expiration alert.

» opsgenie_escalation

Manages an Escalation within Opsgenie.

» Example Usage

```
data "opsgenie_escalation" "test" {
  name = "existing-escalation"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the escalation.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Escalation.
- `rules` - Escalation rules
- `description` - Escalation Description
- `repeat` - Escalation repeat preferences
- `owner_team_id` - If owner team exist the id of the team is exported

» opsgenie_schedule

Manages a Schedule within Opsgenie.

» Example Usage

```
data "opsgenie_schedule" "test" {
  name = "sre-team schedule"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the schedule.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule.
- `rules` - A Member block as documented below.
- `description` - Timezone of schedule. Please look at Supported Timezone Ids for available timezones - Defaults to "America/New_York".
- `timezone` - The description of schedule.
- `enabled` - Enable/disable state of schedule
- `owner_team_id` - Owner team id of the schedule.

» opsgenie_user

Manages a User within Opsgenie.

» Example Usage

```
resource "opsgenie_user" "test" {
  username  = "user@domain.com"
  full_name = "Test User"
  role      = "User"
  locale    = "en_US"
  timezone  = "America/New_York"
}
```

» Argument Reference

The following arguments are supported:

- `username` - (Required) The email address associated with this user. Opsgenie defines that this must not be longer than 100 characters and must contain lowercase characters only.
- `full_name` - (Required) The Full Name of the User.
- `role` - (Required) The Role assigned to the User. Either a built-in such as 'Admin' or 'User' - or the name of a custom role.
- `locale` - (Optional) Location information for the user. Please look at Supported Locale Ids for available locales.
- `timezone` - (Optional) Timezone information of the user. Please look at Supported Timezone Ids for available timezones.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie User.

» Import

Users can be imported using the `id`, e.g.

```
$ terraform import opsgenie_user.user da4faf16-5546-41e4-8330-4d0002b74048s
```

» opsgenie_user_contact

Manages a User Contact.

» Example Usage

```
resource "opsgenie_user_contact" "sms" {
  username = "${opsgenie_user.exampleuser.username}"
  to       = "39-123"
  method   = "sms"
}

resource "opsgenie_user_contact" "email" {
  username = "${opsgenie_user.exampleuser.username}"
```

```
  to       = "fahri@opsgenie.com"
  method   = "email"
}

resource "opsgenie_user_contact" "voice" {
  username = "${opsgenie_user.exampleuser.username}"
  to       = "39-123"
  method   = "voice"
}
```

» Argument Reference

The following arguments are supported:

- `username` - (Required) The username for contact.(reference)
- `to` - (Required) to field is the address of given method.
- `method` - (Required) This parameter is the contact method of user and should be one of email, sms or voice. Please note that adding mobile is not supported from API.
- `enabled` - (Optional) Enable contact of the user in OpsGenie. Default value is true.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Contact.

» Import

Users can be imported using the `id`, e.g.

```
$ terraform import opsgenie_user_contact.testcontact username/contactId
```

For this example: - Username = `genie@awesometeam.com` - Contact Id = `2d1a78d0-c13e-47d3-af0a-8b6d0cc2b7b1`

```
$ terraform import opsgenie_user_contact.testcontact genie@awesometeam.com/2d1a78d0-c13e-47c
```

» opsgenie_team

Manages a Team within Opsgenie.

» Example Usage

```
resource "opsgenie_user" "first" {
  username  = "user@domain.com"
  full_name = "name "
  role      = "User"
}

resource "opsgenie_user" "second" {
  username  = "test@domain.com"
  full_name = "name "
  role      = "User"
}

resource "opsgenie_team" "test" {
  name        = "example"
  description = "This team deals with all the things"

  member {
    id   = "${opsgenie_user.first.id}"
    role = "admin"
  }

  member {
    id   = "${opsgenie_user.second.id}"
    role = "user"
  }
}

resource "opsgenie_team" "self-service" {
  name           = "Self Service"
  description    = "Membership in this team is managed via OpsGenie web UI only"
  ignore_members = true
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name associated with this team. Opsgenie defines that this must not be longer than 100 characters.
- `description` - (Optional) A description for this team.
- `ignore_members` - (Optional) Set to true to ignore any configured member blocks and any team member added/updated/removed via OpsGenie web

UI. Use this option e.g. to maintain membership via web UI only and use it only for new teams. Changing the value for existing teams might lead to strange behaviour. Defaults to false.

- `member` - (Optional) A Member block as documented below.

`member` supports the following:

- `id` - (Required) The UUID for the member to add to this Team.
- `role` - (Optional) The role for the user within the Team - can be either 'admin' or 'user', defaults to 'user' if not set.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Team.

» Import

Teams can be imported using the `id`, e.g.

```
$ terraform import opsgenie_team.team1 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_team_routing_rule

Manages a Team Routing Rule within Opsgenie.

» Example Usage

```
resource "opsgenie_schedule" "test" {
  name = "genieschedule"
  description = "schedule test"
  timezone = "Europe/Rome"
  enabled = false
}

resource "opsgenie_team" "test" {
  name        = "example team"
  description = "This team deals with all the things"

}

resource "opsgenie_team_routing_rule" "test" {
```

```
  name     = "routing rule example"
  team_id  = "${opsgenie_team.test.id}"
  order    = 0
  timezone = "America/Los_Angeles"
  criteria {
    type = "match-any-condition"
    conditions {
      field          = "message"
      operation      = "contains"
      expected_value = "expected1"
      not            = false

    }
  }
  time_restriction {
    type = "weekday-and-time-of-day"
    restrictions {
      start_day  = "monday"
      start_hour = 8
      start_min  = 0
      end_day    = "tuesday"
      end_hour   = 18
      end_min    = 30
    }
  }
  notify {
    name = "${opsgenie_schedule.test.name}"
    type = "schedule"
  }

}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) Name of the team routing rule
- `order` - (Optional) The order of the team routing rule within the rules. order value is actually the index of the team routing rule whose minimum value is 0 and whose maximum value is n-1 (number of team routing rules is n)
- `timezone` - (Optional) Timezone of team routing rule. If timezone field is not given, account timezone is used as default.You can refer to Supported Locale IDs for available timezones

- `criteria` - (Optional) You can refer Criteria for detailed information about criteria and its fields
- `timeRestriction` - (Optional) You can refer Time Restriction for detailed information about time restriction and its fields.
- `notify` - (Optional) Target entity of schedule, escalation, or the reserved word none which will be notified in routing rule. The possible values for notify type : schedule, escalation, none

`notify` supports the following:

- `type` - (Required)
- `name` - (Optional)
- `id` - (Optional)

`criteria` supports the following:

- `type` - (Required) Type of the operation will be applied on conditions. Should be one of match-all, match-any-condition or match-all-conditions.
- `conditions` - (Optional) List of conditions will be checked before applying team routing rule.

`conditions` supports the following:

- `field` - (Required) Specifies which alert field will be used in condition. Possible values are message, alias, description, source, entity, tags, actions, extra-properties, recipients, teams or priority.
- `key` - (Optional) If field is set as extra-properties, key could be used for key-value pair.
- `not` - (Optional) Indicates behaviour of the given operation. Default value is false.
- `operation` - (true) It is the operation that will be executed for the given field and key. Possible operations are matches, contains, starts-with, ends-with, equals, contains-key, contains-value, greater-than, less-than, is-empty and equals-ignore-whitespace.
- `expectedValue` - (Optional) User defined value that will be compared with alert field according to the operation. Default value is empty string.
- `order` - (Optional) Order of the condition in conditions list.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Team Routing Rule.

» Import

Team Routing Rules can be imported using the `id`, e.g.

```
$ terraform import opsgenie_team_routing_rule.ruletest teamId/routingRuleId
```

For this example: - Team Id = `c827c472-31f2-497b-9ec6-8ec855d7d94c` - Routing Rule Id = `2d1a78d0-c13e-47d3-af0a-8b6d0cc2b7b1`

```
$ terraform import opsgenie_team_routing_rule.ruletest c827c472-31f2-497b-9ec6-8ec855d7d94c/
```

» opsgenie_api_integration

Manages an API Integration within Opsgenie.

» Example Usage

```
resource "opsgenie_api_integration" "example-api-integration" {
  name = "api-based-int"
  type = "API"

  responders {
    type = "user"
    id   = "${opsgenie_user.user.id}"
  }

  responders {
    type = "user"
    id   = "${opsgenie_user.fahri.id}"
  }
}

resource "opsgenie_api_integration" "example-api-integration" {
  name = "api-based-int-2"
  type = "Prometheus"

  responders {
    type = "user"
    id   = "${opsgenie_user.user.id}"
  }

  enabled                         = false
  allow_write_access              = false
  ignore_responders_from_payload = true
  suppress_notifications          = true
```

```
  owner_team_id                   = "${opsgenie_team.team.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the integration. Name must be unique for each integration.
- `type` - (Optional) Type of the integration (API, Marid, Prometheus, etc). The full list of options can be found here.
- `allow_write_access` - (Optional) This parameter is for configuring the write access of integration. If write access is restricted, the integration will not be authorized to write within any domain. Defaults to true.
- `enabled` - (Optional) This parameter is for specifying whether the integration will be enabled or not. Defaults to true
- `ignore_responders_from_payload` - (Optional) If enabled, the integration will ignore recipients sent in request payloads. Defaults to false.
- `suppress_notifications` - (Optional) If enabled, notifications that come from alerts will be suppressed. Defaults to false.
- `owner_team_id` - (Optional) Owner team id of the integration.
- `responders` - (Optional) User, schedule, teams or escalation names to calculate which users will receive the notifications of the alert.

`responders` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie API Integration.
- `api_key` - (Computed) API key of the created integration

» Import

API Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_team.team1 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_email_integration

Manages an Email Integration within Opsgenie.

» Example Usage

```
resource "opsgenie_email_integration" "test" {
  name           = "genieintegration-name"
  email_username = "fahri"
}

resource "opsgenie_email_integration" "test" {
  name = "genieintegration-%s"

  responders {
    type = "user"
    id   = "${opsgenie_user.test.id}"
  }

  responders {
    type = "schedule"
    id   = "${opsgenie_schedule.test.id}"
  }

  responders {
    type = "escalation"
    id   = "${opsgenie_escalation.test.id}"
  }

  responders {
    type = "team"
    id   = "${opsgenie_team.test2.id}"
  }

  email_username                = "test"
  enabled                       = true
  ignore_responders_from_payload = true
  suppress_notifications        = true
}


resource "opsgenie_email_integration" "test" {
  name = "genieintegration-%s"
```

```
  responders {
    type = "user"
    id   = "${opsgenie_user.test.id}"
  }

  email_username                = "test"
  enabled                       = true
  ignore_responders_from_payload = true
  suppress_notifications        = true
  owner_team_id                 = "${opsgenie_team_genies.id}"

}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the integration. Name must be unique for each integration.
- `email_username` - (Required) The username part of the email address. It must be unique for each integration.
- `enabled` - (Optional) A Member block as documented below.
- `ignore_responders_from_payload` - (Optional) If enabled, the integration will ignore recipients sent in request payloads. Defaults to false.
- `suppress_notifications` - (Optional) If enabled, notifications that come from alerts will be suppressed. Defaults to false.
- `owner_team_id` - (Optional) Owner team id of the integration.
- `responder` - (Optional) User, schedule, teams or escalation names to calculate which users will receive the notifications of the alert.

`responder` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Email based Integration.

» Import

Email Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_email_integration.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_heartbeat

Manages heartbeat within Opsgenie.

» Example Usage

```
resource "opsgenie_heartbeat" "test" {
    name           = "genieheartbeat-%s"
    description    = "test opsgenie heartbeat terraform"
    interval_unit  = "minutes"
    interval       = 10
    enabled        = false
    alert_message  = "Test"
    alert_priority = "P3"
    alert_tags     = ["test","fahri"]
    owner_team_id  = "${opsgenie_team.test.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the heartbeat
- `description` - (Optional) An optional description of the heartbeat
- `interval_unit` - (Required) Interval specified as minutes, hours or days.
- `interval` - (Required) Specifies how often a heartbeat message should be expected.
- `enabled` - (True) Enable/disable heartbeat monitoring.
- `owner_team_id` - (Optional) Owner team of the heartbeat.
- `alert_message` - (Optional) Specifies the alert message for heartbeat expiration alert. If this is not provided, default alert message is "Heartbeat-Name is expired".
- `alert_priority` - (Optional) Specifies the alert priority for heartbeat expiration alert. If this is not provided, default priority is P3.

- `alert_tags` - (Optional) Specifies the alert tags for heartbeat expiration alert.

» Attributes Reference

Only the arguments listed above are exposed as attributes.

» Import

Heartbeat Integrations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_heartbeat.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_escalation

Manages an Escalation within Opsgenie.

» Example Usage

```
resource "opsgenie_escalation" "test" {
  name = "genieescalation-%s"

  rules {
    condition   = "if-not-acked"
    notify_type = "default"
    delay       = 1

    recipient {
      type = "user"
      id   = "${opsgenie_user.test.id}"
        }
    }
}

resource "opsgenie_escalation" "test" {
  name          = "genieescalation-%s"
  description   = "test"
  owner_team_id = "${opsgenie_team.test.id}"

  rules {
    condition   = "if-not-acked"
    notify_type = "default"
```

```
    delay       = 1

    recipient {
      type = "user"
      id   = "${opsgenie_user.test.id}"
    }

    recipient {
      type = "team"
      id   = "${opsgenie_team.test.id}"
    }

    recipient {
      type = "schedule"
      id   = "${opsgenie_schedule.test.id}"
    }
  }

  repeat  {
    wait_interval          = 10
    count                  = 1
    reset_recipient_states = true
    close_alert_after_all  = false
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the escalation.
- `rules` - (Required) List of the escalation rules.
- `description` - (Optional) Description of the escalation.
- `owner_team_id` - (Optional) Owner team id of the escalation.
- `repeat` - (Optional) Repeat preferences of the escalation including repeat interval, count, reverting acknowledge and seen states back and closing an alert automatically as soon as repeats are completed

`rules` supports the following:

- `condition` - (Required) The condition for notifying the recipient of escalation rule that is based on the alert state. Possible values are: if-not-acked and if-not-closed. If not given, if-not-acked is used.

- `notify_type` - (Required) Recipient calculation logic for schedules. Possible values are: `default: on call users next: next users in rotation previous: previous users on rotation users: users of the team admins: admins of the team all: all members of the team`
- `recipient` - (Required) Object of schedule, team, or users which will be notified in escalation. The possible values for participants are: user, schedule, team.
- `delay` - (Required) Time delay of the escalation rule. This parameter takes an object that consists timeAmount field that takes time amount in minutes.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Escalation.

» Import

Escalations can be imported using the `id`, e.g.

```
$ terraform import opsgenie_escalation.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_schedule

Manages a Schedule within Opsgenie.

» Example Usage

```
resource "opsgenie_schedule" "test" {
  name        = "genieschedule-%s"
  description = "schedule test"
  timezone    = "Europe/Rome"
  enabled     = false
}

resource "opsgenie_schedule" "test" {
  name          = "genieschedule-%s"
  description   = "schedule test"
  timezone      = "Europe/Rome"
```

```
  enabled       = false
  owner_team_id = "${opsgenie_team.test.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the schedule.
- `rules` - (Required) A Member block as documented below.
- `description` - (Optional) The description of schedule.
- `timezone` - (Optional) Timezone of schedule. Please look at Supported Timezone Ids for available timezones - Defaults to "America/New_York".
- `enabled` - (Optional) Enable/disable state of schedule
- `owner_team_id` - (Optional) Owner team id of the schedule.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule.

» Import

Schedule can be imported using the `id`, e.g.

```
$ terraform import opsgenie_schedule.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_schedule_rotation

Manages a Schedule Rotation within Opsgenie.

» Example Usage

```
resource "opsgenie_schedule_rotation" "test" {
  schedule_id = "${opsgenie_schedule.test.id}"
  name        = "test"
  start_date  = "2019-06-18T17:45:00Z"
  end_date    ="2019-06-20T17:45:00Z"
  type        ="hourly"
```

```
  length      = 6

  participant {
    type = "user"
    id   = "${opsgenie_user.test.id}"
  }

  time_restriction {
    type = "time-of-day"

    restriction {
      start_hour = 1
      start_min  = 1
      end_hour   = 10
      end_min    = 1
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `schedule_id` - (Required) Identifier of the schedule.
- `name` - (Optional) Name of rotation.
- `start_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00). Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically
- `end_date` - (Optional) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00). Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically
- `type` - (Required) Type of rotation. May be one of daily, weekly and hourly.
- `length` - (Required) Length of the rotation with default value 1.
- `participant` - (Required) List of escalations, teams, users or the reserved word none which will be used in schedule. Each of them can be used multiple times and will be rotated in the order they given. "user,escalation,team,none"
- `time_restriction` - (Required)

`participant` supports the following:

- `type` - (Required) The responder type.
- `id` - (Required) The id of the responder.

`time_restriction` supports the following:

- `type` - (Required) This parameter should be set time-of-day
- `restriction` - (Required) It is a restriction object which is described below. In this case startDay/endDay fields are not supported.

  `restriction` supports the following:

  - `start_hour` - (Required) Value of the hour that frame will start
  - `start_min` - (Required) Value of the minute that frame will start. Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically.
  - `end_hour` - (Required) Value of the hour that frame will end.
  - `end_min` - (Required) Value of the minute that frame will end. Minutes may take 0 or 30 as value. Otherwise they will be converted to nearest 0 or 30 automatically.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Schedule Rotation

» Import

Schedule Rotations can be imported using the `id` and `schedule_id`, e.g.

- `terraform import opsgenie_schedule_rotation.test schedule_id/id`

For this example: - Schedule Id = `c827c472-31f2-497b-9ec6-8ec855d7d94c` - Rotation Id = `2d1a78d0-c13e-47d3-af0a-8b6d0cc2b7b1`

```
$ terraform import opsgenie_schedule_rotation.test c827c472-31f2-497b-9ec6-8ec855d7d94c/2d1a
```

» opsgenie_maintenance

Manages a Maintenance within Opsgenie.

» Example Usage

```
resource "opsgenie_maintenance" "test" {
  description = "geniemaintenance-%s"

  time {
    type       = "schedule"
    start_date = "2019-06-20T17:45:00Z"
    end_date   = "2019-06-20T17:50:00Z"
  }

  rules{}
}

resource "opsgenie_maintenance" "test" {
  description = "geniemaintenance-%s"

  time {
    type       = "schedule"
    start_date = "2019-06-20T17:45:00Z"
    end_date   = "2019-06-%dT17:50:00Z"
  }

  rules {
    state = "enabled"

    entity {
      id   = "${opsgenie_email_integration.test.id}"
      type = "integration"
    }
  }
}
```

» Argument Reference

The following arguments are supported:

- `time` - (Required) Time configuration of maintenance. It takes a time object which has type, startDate and endDate fields
- `rules` - (Required) Rules of maintenance, which takes a list of rule objects and defines the maintenance rules over integrations and policies.
- `description` - (Optional) Description for the maintenance.

`times` supports the following:

- `type` - (Required) This parameter defines when the maintenance will be active. It can take one of for-5-minutes, for-30-minutes, for-1-hour, indefinitely or schedule.
- `start_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00).
- `end_date` - (Required) This parameter takes a date format as (yyyy-MM-dd'T'HH:mm:ssZ) (e.g. 2019-06-11T08:00:00+02:00).

`rules` supports the following:

- `entity` - (Required) This field represents the entity that maintenance will be applied. Entity field takes two mandatory fields as id and type.
  - `id` - (Required) The id of the entity that maintenance will be applied.
  - `type` - (Required) The type of the entity that maintenance will be applied. It can be either integration or policy.
- `state` - (Required) State of rule that will be defined in maintenance and can take either enabled or disabled for policy type rules. This field has to be disabled for integration type entity rules.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Maintenance Policy.

» Import

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

```
$ terraform import opsgenie_maintenance.test 812be1a1-32c8-4666-a7fb-03ecc385106c
```

» opsgenie_notification_policy

Manages a Notification Policy within Opsgenie.

» Example Usage

```
resource "opsgenie_team" "test" {
  name        = "example team"
  description = "This team deals with all the things"

}
```

```
resource "opsgenie_notification_policy" "test" {
  name               = "example policy"
  team_id            = opsgenie_team.test.id
  policy_description = "This policy has a delay action"
  delay_action {
    delay_option = "next-time"
    until_minute = 1
    until_hour   = 9
  }
  filter {}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the notification policy
- `team_id` - (Required) Id of team that this policy belons to.
- `enabled` - (Optional) If policy should be enabled. Default: true
- `policy_description` - (Optional) Description of the policy. This can be max 512 characters.
- `filter` - (Required) A notification filter which will be applied. This filter can be empty: filter {} - this means 'match-all'. This is a block, structure is documented below.
- `time_restriction` - (Optional) Time restrictions specified in this field must be met for this policy to work. This is a block, structure is documented below.
- `auto_close_action` - (Optional) Auto Restart Action of the policy. This is a block, structure is documented below.
- `auto_restart_action` - (Optional) Auto Restart Action of the policy. This is a block, structure is documented below.
- `de_duplication_action` - (Optional) Deduplication Action of the policy. This is a block, structure is documented below.
- `delay_action` - (Optional) Delay notifications. This is a block, structure is documented below.
- `suppress` - (Optional) Suppress value of the policy. Values are: true, false. Default: false

The `filter` block supports:

- `type` (Optional) - A filter type, supported types are: "match-all", "match-any-condition", "match-all-conditions". Default: "match-all"

- `conditions` (Optional) Conditions applied to filter. This is a block, structure is documented below.

The `conditions` block supports:

- `field` - (Required) Specifies which alert field will be used in condition. Possible values are "message", "alias", "description", "source", "entity", "tags", "actions", "details", "extra-properties", "recipients", "teams", "priority"
- `operation` - (Required) It is the operation that will be executed for the given field and key. Possible operations are "matches", "contains", "starts-with", "ends-with", "equals", "contains-key", "contains-value", "greater-than", "less-than", "is-empty", "equals-ignore-whitespace".
- `key` - (Optional) If `field` is set as extra-properties, key could be used for key-value pair
- `not` - (Optional) Indicates behaviour of the given operation. Default: false
- `expected_value` - (Optional) User defined value that will be compared with alert field according to the operation. Default: empty string
- `order` - (Optional) Order of the condition in conditions list

The `time_restriction` block supports:

- `type` - (Required) Defines if restriction should apply daily on given hours or on certain days and hours. Possible values are: "time-of-day", "weekday-and-time-of-day"
- `restrictions` - (Optional) List of days and hours definitions for field type = "weekday-and-time-of-day". This is a block, structure is documented below.
- `restriction` - (Optional) A definition of hourly definition applied daily, this has to be used with combination: type = "time-of-day". This is a block, structure is documented below.

The `restrictions` block supports:

- `start_day` - (Required) Starting day of restriction (eg. "monday")
- `end_day` - (Required) Ending day of restriction (eg. "wednesday)
- `start_hour` - (Required) Starting hour of restriction on defined `start_day`
- `end_hour` - (Required) Ending hour of restriction on defined `end_day`
- `start_minute` - (Required) Staring minute of restriction on defined `start_hour`
- `end_minute` - (Required) Ending minute of restriction on defined `end_hour`

The `restriction` block supports:

- `start_hour` - (Required) Starting hour of restriction.
- `end_hour` - (Required) Ending hour of restriction.
- `start_minute` - (Required) Staring minute of restriction on defined `start_hour`
- `end_minute` - (Required) Ending minute of restriction on defined `end_hour`

The `auto_close_action` block supports:

- `duration` - (Required) Duration of this action. This is a block, structure is documented below.

The `auto_restart_action` block supports:

- `duration` - (Required) Duration of this action. This is a block, structure is documented below.
- `max_repeat_count` - (Required) How many times to repeat. This is a integer attribute.

The `de_duplication_action` block supports:

- `duration` - (Required) Duration of this action. This is a block, structure is documented below.
- `de_duplication_action_type` - (Required) Deduplication type. Possible values are: "value-based", "frequency-based"
- `count` - (Required) - Count

The `delay_action` block supports:

- `delay_option` - (Required) Defines until what day to delay or for what duration. Possible values are: "for-duration", "next-time", "next-weekday", "next-monday", "next-tuesday", "next-wednesday", "next-thursday", "next-friday", "next-saturday", "next-sunday"
- `duration` - (Optional) Duration of this action. If `delay_option` = "for-duration" this has to be set. This is a block, structure is documented below.
- `until_hour` - (Optional) Until what hour notifications will be delayed. If `delay_option` is set to antyhing else then "for-duration" this has to be set.
- `until_minute` - (Optional) Until what minute on `until_hour` notifications will be delayed. If `delay_option` is set to antyhing else then "for-duration" this has to be set.

The `duration` block supports:

- `time_unit` - (Optional) Valid time units are: ”minutes”, ”hours”, ”days”. Default: minutes
- `time_amount` - (Required) A amount of time in `time_units`. This is a integer attribute.

» Attributes Reference

The following attributes are exported:

- `id` - The ID of the Opsgenie Notification Policy.

» Import

Notification policies can be imported using the `team id` and `id`, e.g.

```
$ terraform import opsgenie_notification_policy.test teamId/Id
```

For this example: - Team Id = `c827c472-31f2-497b-9ec6-8ec855d7d94c` - Notification Policy Id = `2d1a78d0-c13e-47d3-af0a-8b6d0cc2b7b1`

```
$ terraform import opsgenie_notification_policy.test c827c472-31f2-497b-9ec6-8ec855d7d94c/2c
```