PagerDuty Provider

PagerDuty (https://www.pagerduty.com/) is an alarm aggregation and dispatching service for system administrators and support teams. It collects alerts from your monitoring tools, gives you an overall view of all of your monitoring alarms, and alerts an on duty engineer if there's a problem.

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

Example Usage

```
# Configure the PagerDuty provider
provider "pagerduty" {
  token = "${var.pagerduty_token}"
}

# Create a PagerDuty team
resource "pagerduty_team" "engineering" {
  name        = "Engineering"
  description = "All engineering"
}

# Create a PagerDuty user
resource "pagerduty_user" "earline" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.engineering.id}"]
}
```

Argument Reference

The following arguments are supported:

- `token` - (Required) The v2 authorization token. It can also be sourced from the PAGERDUTY_TOKEN environment variable. See API Documentation (https://v2.developer.pagerduty.com/docs/authentication) for more information.
- `skip_credentials_validation` - (Optional) Skip validation of the token against the PagerDuty API.

pagerduty_escalation_policy

Use this data source to get information about a specific escalation policy (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Escalation_Policies/get_escalation_policies) that you can use for other PagerDuty resources.

Example Usage

```
data "pagerduty_escalation_policy" "test" {
  name = "Engineering Escalation Policy"
}

resource "pagerduty_service" "test" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${data.pagerduty_escalation_policy.test.id}"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use to find an escalation policy in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found escalation policy.
- `name` - The short name of the found escalation policy.

pagerduty_extension_schema

Use this data source to get information about a specific extension (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Extension_Schemas/get_extension_schemas) vendor that you can use for a service (e.g: Slack, Generic Webhook, ServiceNow).

Example Usage

```
data "pagerduty_extension_schema" "webhook" {
  name = "Generic V2 Webhook"
}

resource "pagerduty_user" "example" {
  name  = "Howard James"
  email = "howard.james@example.domain"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}

resource "pagerduty_service" "example" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${pagerduty_escalation_policy.example.id}"
}


resource "pagerduty_extension" "slack"{
  name = "My Web App Extension"
  endpoint_url = "https://generic_webhook_url/XXXXXX/BBBBBB"
  extension_schema = "${data.pagerduty_extension_schema.webhook.id}"
  extension_objects    = ["${pagerduty_service.example.id}"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The extension name to use to find an extension vendor in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found extension vendor.
- `name` - The short name of the found extension vendor.
- `type` - The generic service type for this extension vendor.

pagerduty_schedule

Use this data source to get information about a specific schedule (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Schedules/get_schedules) that you can use for other PagerDuty resources.

Example Usage

```
data "pagerduty_schedule" "test" {
  name = "Daily Engineering Rotation"
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "schedule"
      id   = "${data.pagerduty_schedule.test.id}"
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name to use to find a schedule in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found schedule.
- `name` - The short name of the found schedule.

pagerduty_service

Use this data source to get information about a specific service (https://api-reference.pagerduty.com/#!/Services/get_services).

Example Usage

```
data "pagerduty_service" "example" {
  name = "My Service"
}

data "pagerduty_vendor" "datadog" {
  name = "Datadog"
}

resource "pagerduty_service_integration" "example" {
  name    = "Datadog Integration"
  vendor  = data.pagerduty_vendor.datadog.id
  service = data.pagerduty_service.example.id
  type    = "generic_events_api_inbound_integration"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The service name to use to find a service in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found service.
- `name` - The short name of the found service.

pagerduty_team

Use this data source to get information about a specific team (https://v1.developer.pagerduty.com/documentation/rest/teams/list) that you can use for other PagerDuty resources.

Example Usage

```
data "pagerduty_user" "me" {
  email = "me@example.com"
}

data "pagerduty_team" "devops" {
  name = "devops"
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "DevOps Escalation Policy"
  num_loops = 2

  teams = ["${data.pagerduty_team.devops.id}"]

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${data.pagerduty_user.me.id}"
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the team to find in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found team.
- `name` - The name of the found team.
- `description` - A description of the found team.

pagerduty_user

Use this data source to get information about a specific user (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Users/get_users) that you can use for other PagerDuty resources.

Example Usage

```
data "pagerduty_user" "me" {
  email = "me@example.com"
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${data.pagerduty_user.me.id}"
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `email` - (Required) The email to use to find a user in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found user.
- `name` - The short name of the found user.

pagerduty_vendor

Use this data source to get information about a specific vendor (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Vendors/get_vendors) that you can use for a service integration (e.g Amazon Cloudwatch, Splunk, Datadog).

Example Usage

```
data "pagerduty_vendor" "datadog" {
  name = "Datadog"
}

resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}

resource "pagerduty_service" "example" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${pagerduty_escalation_policy.example.id}"
}

resource "pagerduty_service_integration" "example" {
  name    = "Datadog Integration"
  vendor  = "${data.pagerduty_vendor.datadog.id}"
  service = "${pagerduty_service.example.id}"
  type    = "generic_events_api_inbound_integration"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The vendor name to use to find a vendor in the PagerDuty API.

Attributes Reference

- `id` - The ID of the found vendor.
- `name` - The short name of the found vendor.
- `type` - The generic service type for this vendor.

pagerduty_addon

With add-ons (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Add-ons/get_addons), third-party developers can write their own add-ons to PagerDuty's UI. Given a configuration containing a src parameter, that URL will be embedded in an iframe on a page that's available to users from a drop-down menu.

Example Usage

```
resource "pagerduty_addon" "example" {
  name = "Internal Status Page"
  src  = "https://intranet.example.com/status"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the add-on.
- `src` - (Required) The source URL to display in a frame in the PagerDuty UI. `HTTPS` is required.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the add-on.

Import

Add-ons can be imported using the `id`, e.g.

```
$ terraform import pagerduty_addon.example P3DH5M6
```

pagerduty_escalation_policy

An escalation policy (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Escalation_Policies/get_escalation_policies) determines what user or schedule will be notified first, second, and so on when an incident is triggered. Escalation policies are used by one or more services.

Example Usage

```
resource "pagerduty_team" "example" {
  name        = "Engineering"
  description = "All engineering"
}

resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "example" {
  name      = "Engineering Escalation Policy"
  num_loops = 2
  teams     = ["${pagerduty_team.example.id}"]

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the escalation policy.
- `teams` - (Optional) Teams associated with the policy. Account must have the `teams` ability to use this parameter.
- `description` - (Optional) A human-friendly description of the escalation policy. If not set, a placeholder of "Managed by Terraform" will be set.
- `num_loops` - (Optional) The number of times the escalation policy will repeat after reaching the end of its escalation.
- `rule` - (Required) An Escalation rule block. Escalation rules documented below.

Escalation rules (`rule`) supports the following:

- `escalation_delay_in_minutes` - (Required) The number of minutes before an unacknowledged incident escalates away from this rule.
- `targets` - (Required) A target block. Target blocks documented below.

Targets (`target`) supports the following:

- `type` - (Optional) Can be `user` , `schedule` , `user_reference` or `schedule_reference` . Defaults to `user_reference`
- `id` - (Required) A target ID

Attributes Reference

The following attributes are exported:

- `id` - The ID of the escalation policy.

Import

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

```
$ terraform import pagerduty_escalation_policy.main PLBP09X
```

pagerduty_event_rule

An event rule (https://v2.developer.pagerduty.com/docs/global-event-rules-api) determines what happens to an event that is sent to PagerDuty by monitoring tools and other integrations.

Example Usage

```
resource "pagerduty_event_rule" "second" {
    action_json = jsonencode([
        [
            "route",
            "P5DTL0K"
        ],
        [
            "severity",
            "warning"
        ],
        [
            "annotate",
            "2 Managed by terraform"
        ],
        [
            "priority",
            "PL451DT"
        ]
    ])
    condition_json = jsonencode([
        "and",
        ["contains",["path","payload","source"],"website"],
        ["contains",["path","headers","from","0","address"],"homer"]
    ])
    advanced_condition_json = jsonencode([
        [
            "scheduled-weekly",
            1565392127032,
            3600000,
            "America/Los_Angeles",
            [
                1,
                2,
                3,
                5,
                7
            ]
        ]
    ])
}
resource "pagerduty_event_rule" "third" {
    action_json = jsonencode([
        [
            "route",
            "P5DTL0K"
        ],
        [
            "severity"
```

```
            "severity",
            "warning"
        ],
        [
            "annotate",
            "3 Managed by terraform"
        ],
        [
            "priority",
            "PL451DT"
        ]
    ])
    condition_json = jsonencode([
        "and",
        ["contains",["path","payload","source"],"website"],
        ["contains",["path","headers","from","0","address"],"homer"]
    ])
    depends_on = [pagerduty_event_rule.two]
}
```

Argument Reference

The following arguments are supported:

- `action_json` - (Required) A list of one or more actions for each rule. Each action within the list is itself a list.
- `condition_json` - (Required) Contains a list of conditions. The first field in the list is `and` or `or`, followed by a list of operators and values.
- `advanced_condition_json` - (Optional) Contains a list of specific conditions including `active-between`, `scheduled-weekly`, and `frequency-over`. The first element in the list is the label for the condition, followed by a list of values for the specific condition. For more details on these conditions see Advanced Condition (https://v2.developer.pagerduty.com/docs/global-event-rules-api#section-advanced-condition) in the PagerDuty API documentation.
- `catch_all` - (Optional) A boolean that indicates whether the rule is a catch all for the account.
- `depends_on` - (Optional) A Terraform meta-parameter (https://www.terraform.io/docs/configuration-0-11/resources.html#depends_on) that ensures that the `event_rule` specified is created before the current rule. This is important because Event Rules in PagerDuty are executed in order. `depends_on` ensures that the rules are created in the order specified.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the event rule.

Import

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

```
$ terraform import pagerduty_event_rule.main 19acac92-027a-4ea0-b06c-bbf516519601
```

pagerduty_extension

An extension (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Extensions/post_extensions) can be associated with a service.

Example Usage

```
data "pagerduty_extension_schema" "webhook" {
  name = "Generic V2 Webhook"
}

resource "pagerduty_user" "example" {
  name  = "Howard James"
  email = "howard.james@example.domain"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}

resource "pagerduty_service" "example" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${pagerduty_escalation_policy.example.id}"
}


resource "pagerduty_extension" "slack"{
  name = "My Web App Extension"
  endpoint_url = "https://generic_webhook_url/XXXXXX/BBBBBB"
  extension_schema = "${data.pagerduty_extension_schema.webhook.id}"
  extension_objects    = ["${pagerduty_service.example.id}"]

  config = <<EOF
{
    "restrict": "any",
    "notify_types": {
            "resolve": false,
            "acknowledge": false,
            "assignments": false
    },
    "access_token": "XXX"
}
EOF

}
```

Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the service extension.
- `endpoint_url` - (Optional) The url of the extension.
- `extension_schema` - (Required) This is the schema for this extension.
- `extension_objects` - (Required) This is the objects for which the extension applies (An array of service ids).
- `config` - (Optional) The configuration of the service extension as string containing plain JSON-encoded data.

  Note: You can use the `pagerduty_extension_schema` data source to locate the appropriate extension vendor ID.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the extension.
- `html_url` - URL at which the entity is uniquely displayed in the Web app

Import

Extensions can be imported using the id.e.g.

```
$ terraform import pagerduty_extension.main PLBP09X
```

pagerduty_maintenance_window

A maintenance window (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Maintenance_Windows/get_maintenance_windows) is used to temporarily disable one or more services for a set period of time. No incidents will be triggered and no notifications will be received while a service is disabled by a maintenance window.

Maintenance windows are specified to start at a certain time and end after they have begun. Once started, a maintenance window cannot be deleted; it can only be ended immediately to re-enable the service.

Example Usage

```
resource "pagerduty_maintenance_window" "example" {
  start_time  = "2015-11-09T20:00:00-05:00"
  end_time    = "2015-11-09T22:00:00-05:00"
  services    = ["${pagerduty_service.example.id}"]
}
```

Argument Reference

The following arguments are supported:

- `start_time` - (Required) The maintenance window's start time. This is when the services will stop creating incidents. If this date is in the past, it will be updated to be the current time.
- `end_time` - (Required) The maintenance window's end time. This is when the services will start creating incidents again. This date must be in the future and after the `start_time`.
- `services` - (Required) A list of service IDs to include in the maintenance window.
- `description` - (Optional) A description for the maintenance window.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the maintenance window.

Import

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

```
$ terraform import pagerduty_maintenance_window.main PLBP09X
```

pagerduty_schedule

A schedule (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Schedules/get_schedules) determines the time periods that users are on call. Only on-call users are eligible to receive notifications from incidents.

Example Usage

```
resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_schedule" "foo" {
  name      = "Daily Engineering Rotation"
  time_zone = "America/New_York"

  layer {
    name                          = "Night Shift"
    start                         = "2015-11-06T20:00:00-05:00"
    rotation_virtual_start        = "2015-11-06T20:00:00-05:00"
    rotation_turn_length_seconds = 86400
    users                         = ["${pagerduty_user.foo.id}"]

    restriction {
      type              = "daily_restriction"
      start_time_of_day = "08:00:00"
      duration_seconds  = 32400
    }
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the schedule.
- `time_zone` - (Required) The time zone of the schedule (e.g Europe/Berlin).
- `description` - (Optional) The description of the schedule
- `layer` - (Required) A schedule layer block. Schedule layers documented below.
- `overflow` - (Optional) Any on-call schedule entries that pass the date range bounds will be truncated at the bounds, unless the parameter `overflow` is passed. For instance, if your schedule is a rotation that changes daily at midnight UTC, and your date range is from `2011-06-01T10:00:00Z` to `2011-06-01T14:00:00Z` : If you don't pass the overflow=true parameter, you will get one schedule entry returned with a start of `2011-06-01T10:00:00Z` and end of `2011-06-01T14:00:00Z` . If you do pass the `overflow` parameter, you will get one schedule entry returned with a start of `2011-06-01T00:00:00Z` and end of `2011-06-02T00:00:00Z` .

Schedule layers (`layer`) supports the following:

- `name` - (Optional) The name of the schedule layer.
- `start` - (Required) The start time of the schedule layer. This value will not be read back from the PagerDuty API because the API will always return a new `start` time, which represents the last updated time of the schedule layer.
- `end` - (Optional) The end time of the schedule layer. If not specified, the layer does not end.
- `rotation_virtual_start` - (Required) The effective start time of the schedule layer. This can be before the start time of the schedule.
- `rotation_turn_length_seconds` - (Required) The duration of each on-call shift in `seconds` .
- `users` - (Required) The ordered list of users on this layer. The position of the user on the list determines their order in the layer.
- `restriction` - (Optional) A schedule layer restriction block. Restriction blocks documented below.

Restriction blocks (`restriction`) supports the following:

- `type` - (Required) Can be `daily_restriction` or `weekly_restriction`
- `start_time_of_day` - (Required) The start time in `HH:mm:ss` format.
- `duration_seconds` - (Required) The duration of the restriction in `seconds` .
- `start_day_of_week` - (Required for `weekly_restriction`) Number of the day when restriction starts. From 1 to 7 where 1 is Monday and 7 is Sunday.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the schedule

Import

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

```
$ terraform import pagerduty_schedule.main PLBP09X
```

pagerduty_service

A service (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Services/get_services) represents something you monitor (like a web service, email service, or database service). It is a container for related incidents that associates them with escalation policies.

Example Usage

```
resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}

resource "pagerduty_service" "example" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${pagerduty_escalation_policy.example.id}"
  alert_creation          = "create_incidents"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the service.
- `description` - (Optional) A human-friendly description of the service. If not set, a placeholder of "Managed by Terraform" will be set.
- `auto_resolve_timeout` - (Optional) Time in seconds that an incident is automatically resolved if left open for that long. Disabled if set to the `"null"` string.
- `acknowledgement_timeout` - (Optional) Time in seconds that an incident changes to the Triggered State after being Acknowledged. Disabled if set to the `"null"` string.

- `escalation_policy` - (Required) The escalation policy used by this service.
- `alert_creation` - (Optional) Must be one of two values. PagerDuty receives events from your monitoring systems and can then create incidents in different ways. Value "create_incidents" is default: events will create an incident that cannot be merged. Value "create_alerts_and_incidents" is the alternative: events will create an alert and then add it to a new incident, these incidents can be merged.
- `alert_grouping` - (Optional) Defines how alerts on this service will be automatically grouped into incidents. Note that the alert grouping features are available only on certain plans. If not set, each alert will create a separate incident; If value is set to `time` : All alerts within a specified duration will be grouped into the same incident. This duration is set in the `alert_grouping_timeout` setting (described below). Available on Standard, Enterprise, and Event Intelligence plans; If value is set to "intelligent" - Alerts will be intelligently grouped based on a machine learning model that looks at the alert summary, timing, and the history of grouped alerts. Available on Enterprise and Event Intelligence plan.
- `alert_grouping_timeout` - (Optional) The duration in minutes within which to automatically group incoming alerts. This setting applies only when `alert_grouping` is set to `time` . To continue grouping alerts until the incident is resolved, set this value to `0` .

You may specify one optional `incident_urgency_rule` block configuring what urgencies to use. Your PagerDuty account must have the `urgencies` ability to assign an incident urgency rule. The block contains the following arguments:

- `type` - The type of incident urgency: `constant` or `use_support_hours` (when depending on specific support hours; see `support_hours`).
- `urgency` - The urgency: `low` Notify responders (does not escalate), `high` (follows escalation rules) or `severity_based` Set's the urgency of the incident based on the severity set by the triggering monitoring tool.
- `during_support_hours` - (Optional) Incidents' urgency during support hours.
- `outside_support_hours` - (Optional) Incidents' urgency outside of support hours.

When using `type = "use_support_hours"` in `incident_urgency_rule` you must specify exactly one (otherwise optional) `support_hours` block. Your PagerDuty account must have the `service_support_hours` ability to assign support hours. The block contains the following arguments:

- `type` - The type of support hours. Can be `fixed_time_per_day` .
- `time_zone` - The time zone for the support hours.
- `days_of_week` - Array of days of week as integers. `1` to `7` , `1` being Monday and `7` being Sunday.
- `start_time` - The support hours' starting time of day.
- `end_time` - The support hours' ending time of day.

When using `type = "use_support_hours"` in `incident_urgency_rule` you must specify at least one (otherwise optional) `scheduled_actions` block. The block contains the following arguments:

- `type` - The type of scheduled action. Currently, this must be set to `urgency_change` .
- `to_urgency` - The urgency to change to: `low` (does not escalate), or `high` (follows escalation rules).
- `at` - A block representing when the scheduled action will occur.

The `at` block contains the following arguments: * `type` - The type of time specification. Currently, this must be set to `named_time` . * `name` - Designates either the start or the end of the scheduled action. Can be `support_hours_start` or `support_hours_end` .

Below is an example for a `pagerduty_service` resource with `incident_urgency_rules` with `type = "use_support_hours"`, `support_hours` and a default `scheduled_action` as well.

```
resource "pagerduty_service" "foo" {
  name                    = "bar"
  description             = "bar bar bar"
  auto_resolve_timeout    = 3600
  acknowledgement_timeout = 3600
  escalation_policy       = "${pagerduty_escalation_policy.foo.id}"

  incident_urgency_rule {
    type = "use_support_hours"

    during_support_hours {
      type    = "constant"
      urgency = "high"
    }

    outside_support_hours {
      type    = "constant"
      urgency = "low"
    }
  }

  support_hours {
    type         = "fixed_time_per_day"
    time_zone    = "America/Lima"
    start_time   = "09:00:00"
    end_time     = "17:00:00"
    days_of_week = [1, 2, 3, 4, 5]
  }

  scheduled_actions {
    type       = "urgency_change"
    to_urgency = "high"

    at {
      type = "named_time"
      name = "support_hours_start"
    }
  }
}
```

Attributes Reference

The following attributes are exported:

- `id` - The ID of the service.
- `last_incident_timestamp` - Last incident timestamp of the service
- `created_at` - Creation timestamp of the service
- `status` - The status of the service

- `html_url` - URL at which the entity is uniquely displayed in the Web app

Import

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

```
$ terraform import pagerduty_service.main PLBP09X
```

pagerduty_service_integration

A service integration (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Services/post_services_id_integrations) is an integration that belongs to a service.

Example Usage

```
resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_escalation_policy" "foo" {
  name      = "Engineering Escalation Policy"
  num_loops = 2

  rule {
    escalation_delay_in_minutes = 10

    target {
      type = "user"
      id   = "${pagerduty_user.example.id}"
    }
  }
}

resource "pagerduty_service" "example" {
  name                    = "My Web App"
  auto_resolve_timeout    = 14400
  acknowledgement_timeout = 600
  escalation_policy       = "${pagerduty_escalation_policy.example.id}"
}

resource "pagerduty_service_integration" "example" {
  name    = "Generic API Service Integration"
  type    = "generic_events_api_inbound_integration"
  service = "${pagerduty_service.example.id}"
}

data "pagerduty_vendor" "datadog" {
  name = "Datadog"
}

resource "pagerduty_service_integration" "datadog" {
  name    = "${data.pagerduty_vendor.datadog.name}"
  service = "${pagerduty_service.example.id}"
  vendor  = "${data.pagerduty_vendor.datadog.id}"
}

data "pagerduty_vendor" "cloudwatch" {
  name = "Cloudwatch"
}

resource "pagerduty_service_integration" "cloudwatch" {
  name    = "${data.pagerduty_vendor.cloudwatch.name}"
  service = "${pagerduty_service.example.id}"
  vendor  = "${data.pagerduty_vendor.cloudwatch.id}"
}
```

Argument Reference

The following arguments are supported:

- `service` - (Required) The ID of the service the integration should belong to.
- `name` - (Optional) The name of the service integration.
- `type` - (Optional) The service type. Can be: `aws_cloudwatch_inbound_integration`, `cloudkick_inbound_integration`, `event_transformer_api_inbound_integration`, `events_api_v2_inbound_integration` (requires service `alert_creation` to be `create_alerts_and_incidents`), `generic_email_inbound_integration`, `generic_events_api_inbound_integration`, `keynote_inbound_integration`, `nagios_inbound_integration`, `pingdom_inbound_integration` or `sql_monitor_inbound_integration`.

  Note: This is meant for **generic** service integrations. To integrate with a **vendor** (e.g Datadog or Amazon Cloudwatch) use the `vendor` field instead.
- `vendor` - (Optional) The ID of the vendor the integration should integrate with (e.g Datadog or Amazon Cloudwatch).
- `integration_key` - (Optional) This is the unique key used to route events to this integration when received via the PagerDuty Events API.
- `integration_email` - (Optional) This is the unique fully-qualified email address used for routing emails to this integration for processing.

  Note: You can use the `pagerduty_vendor` data source to locate the appropriate vendor ID.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the service integration.
- `integration_key` - This is the unique key used to route events to this integration when received via the PagerDuty Events API.
- `integration_email` - This is the unique fully-qualified email address used for routing emails to this integration for processing.
- `html_url` - URL at which the entity is uniquely displayed in the Web app

To configure an event, please use the `integration_key` in the following interpolation:

```
https://events.pagerduty.com/integration/${pagerduty_service_integration.slack.integration_key}/enqueue
```

Import

Services can be imported using their related `service` id and service integration `id` separated by a dot, e.g.

```
$ terraform import pagerduty_service_integration.main PLSSSSS.PLIIIII
```

pagerduty_team

A team (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Teams/get_teams) is a collection of users and escalation policies that represent a group of people within an organization.

The account must have the `teams` ability to use the following resource.

Example Usage

```
resource "pagerduty_team" "example" {
  name        = "Engineering"
  description = "All engineering"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the group.
- `description` - (Optional) A human-friendly description of the team. If not set, a placeholder of "Managed by Terraform" will be set.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the team.
- `html_url` - URL at which the entity is uniquely displayed in the Web app

Import

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

```
$ terraform import pagerduty_team.main PLBP09X
```

pagerduty_team_membership

A team membership (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Teams/put_teams_id_users_user_id) manages memberships within a team.

Example Usage

```
resource "pagerduty_user" "foo" {
  name = "foo"
  email = "foo@bar.com"
}

resource "pagerduty_team" "foo" {
  name        = "foo"
  description = "foo"
}

resource "pagerduty_team_membership" "foo" {
  user_id = "${pagerduty_user.foo.id}"
  team_id = "${pagerduty_team.foo.id}"
}
```

Argument Reference

The following arguments are supported:

- `user_id` - (Required) The ID of the user to add to the team.
- `team_id` - (Required) The ID of the team in which the user will belong.

Attributes Reference

The following attributes are exported:

- `user_id` - The ID of the user belonging to the team.
- `team_id` - The team ID the user belongs to.

Import

Team memberships can be imported using the `user_id` and `team_id`, e.g.

```
$ terraform import pagerduty_team_membership.main PLBP09X:PLB09Z
```

pagerduty_user_contact_method

A contact method (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Users/get_users_id_contact_methods) is a contact method for a PagerDuty user (email, phone or SMS).

Example Usage

```
resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}

resource "pagerduty_user_contact_method" "email" {
  user_id = "${pagerduty_user.example.id}"
  type    = "email_contact_method"
  address = "foo@bar.com"
  label   = "Work"
}

resource "pagerduty_user_contact_method" "phone" {
  user_id      = "${pagerduty_user.example.id}"
  type         = "phone_contact_method"
  country_code = "+1"
  address      = "2025550199"
  label        = "Work"
}

resource "pagerduty_user_contact_method" "sms" {
  user_id      = "${pagerduty_user.example.id}"
  type         = "sms_contact_method"
  country_code = "+1"
  address      = "2025550199"
  label        = "Work"
}
```

Argument Reference

The following arguments are supported:

- `user_id` - (Required) The ID of the user.
- `type` - (Required) The contact method type. May be (`email_contact_method` , `phone_contact_method` , `sms_contact_method` , `push_notification_contact_method`).
- `send_short_email` - (Optional) Send an abbreviated email message instead of the standard email output.
- `country_code` - (Optional) The 1-to-3 digit country calling code. Required when using `phone_contact_method` or `sms_contact_method` .
- `label` - (Required) The label (e.g., "Work", "Mobile", etc.).

- `address` - (Required) The "address" to deliver to: `email`, `phone number`, etc., depending on the type.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the contact method.
- `blacklisted` - If true, this phone has been blacklisted by PagerDuty and no messages will be sent to it.
- `enabled` - If true, this phone is capable of receiving SMS messages.

Import

Contact methods can be imported using the `user_id` and the `id`, e.g.

```
$ terraform import pagerduty_user_contact_method.main PLBP09X:PLBP09X
```

pagerduty_user

A user (https://v2.developer.pagerduty.com/v2/page/api-reference#!/Users/get_users) is a member of a PagerDuty account that have the ability to interact with incidents and other data on the account.

Example Usage

```
resource "pagerduty_team" "example" {
  name        = "Engineering"
  description = "All engineering"
}

resource "pagerduty_user" "example" {
  name  = "Earline Greenholt"
  email = "125.greenholt.earline@graham.name"
  teams = ["${pagerduty_team.example.id}"]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the user.
- `email` - (Required) The user's email address.
- `color` - (Optional) The schedule color for the user. Valid options are purple, red, green, blue, teal, orange, brown, turquoise, dark-slate-blue, cayenne, orange-red, dark-orchid, dark-slate-grey, lime, dark-magenta, lime-green, midnight-blue, deep-pink, dark-green, dark-orange, dark-cyan, darkolive-green, dark-slate-gray, grey20, firebrick, maroon, crimson, dark-red, dark-goldenrod, chocolate, medium-violet-red, sea-green, olivedrab, forest-green, dark-olive-green, blue-violet, royal-blue, indigo, slate-blue, saddle-brown, or steel-blue.
- `role` - (Optional) The user role. Account must have the `read_only_users` ability to set a user as a `read_only_user`. Can be `admin`, `limited_user`, `owner`, `read_only_user`, `team_responder` or `user`
- `job_title` - (Optional) The user's title.
- `teams` - (Optional, **DEPRECATED**) A list of teams the user should belong to. Please use `pagerduty_team_membership` instead.
- `description` - (Optional) A human-friendly description of the user. If not set, a placeholder of "Managed by Terraform" will be set.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the user.

- `avatar_url` - The URL of the user's avatar.
- `time_zone` - The timezone of the user
- `html_url` - URL at which the entity is uniquely displayed in the Web app
- `invitation_sent` - If true, the user has an outstanding invitation.

Import

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

```
$ terraform import pagerduty_user.main PLBP09X
```