

Creating a Waiter

Product or feature is covered by the [Pre-GA Offerings Terms \(/terms/service-terms#1\)](/terms/service-terms#1) of the Google Cloud Platform of Service. Pre-GA products and features may have limited support, and changes to pre-GA products and features may not be compatible with other pre-GA versions. For more information, see the [launch stage descriptions \(/docs/launch-stages#product-launch-stages\)](#).

This page explains how to create a Waiter resource. To learn more about waiters, read the [Runtime Configurator Fundamentals \(/deployment-manager/runtime-configurator\)](/deployment-manager/runtime-configurator).

A Waiter resource waits for a certain success or failure condition before returning a response. For both success and failure, you set a Cardinality condition, where the waiter waits for some number of variables to be created under a specific path prefix. After the variables have been created, the waiter returns. Your application code can then respond to its success or failure. If the current state of your variables already match either the success or failure end conditions, the waiter will return success or failure immediately.

The waiter does not care about the value of the variables under the specific prefix; it only cares that there is the desired number of variables under the desired prefix. Also, cardinality conditions are recursive, so the waiter counts variables under a specific key, including any keys that might, in turn, contain a subtree of keys.

Before you begin

- If you want to use the command-line examples in this guide, install the [`gcloud` command-line tool \(/sdk\)](/sdk).
- If you want to use the API examples in this guide, set up [API access \(/deployment-manager/docs/reference/latest\)](/deployment-manager/docs/reference/latest).
- Read [Runtime Configurator Fundamentals \(/deployment-manager/runtime-configurator\)](/deployment-manager/runtime-configurator).
- Read [Creating and Deleting RuntimeConfig Resources \(/deployment-manager/runtime-configurator/create-and-delete-runtimeconfig-resources\)](/deployment-manager/runtime-configurator/create-and-delete-runtimeconfig-resources).

- Read [Setting and Getting Data](#) (/deployment-manager/runtime-configurator/set-and-get-variables).

Creating a waiter

To create a waiter:

1. Determine your success, and optionally, failure condition for the waiter.

For example, the following sample code sets the conditions for success and failure, where the waiter returns successfully if the number of paths under `/status/success` is three and fails if the path under `/status/failure` is two:

```
{
  'name': 'projects/[PROJECT_ID]/configs/[CONFIG_NAME]/waiters/[WAITER_NAME]',
  'timeout': '360s',
  'success': {
    'cardinality': {
      'path': '/status/success',
      'number': 3
    }
  },
  'failure': {
    'cardinality': {
      'path': '/status/failure',
      'number': 2
    }
  }
}
```

Best practices for defining a waiter:

- Only one success condition and one failure condition is allowed per waiter.
- You should maintain one waiter per path.
- Failure conditions are always evaluated before success conditions.
- Do not overlap path prefixes between conditions.

2. Create the waiter.

Deployment Manager `gcloud` (#gcloud) API (#api)

To create a waiter in Deployment Manager, specify the waiter type:

```
runtimeconfig.v1beta1.waiter
```

In the properties of the waiter, provide the `name`, `location`, `timeout`, and the end conditions of the waiter:

```
- name: [NAME]
  type: runtimeconfig.v1beta1.waiter
  properties:
    parent: $(ref.[CONFIG_NAME].name)
    waiter: [WAITER_NAME]
    timeout: [TIMEOUT_SECS]
    success:
      cardinality:
        path: [SUCCESS_PATH_PREFIX]
        number: [SUCCESS_NUMBER]
```

where:

- `[NAME]` is the resource name.
- `[CONFIG_NAME]` is the Config resource for this request.
- `[WAITER_NAME]` is the name for this waiter.
- `[TIMEOUT_SECS]` is the number of seconds to wait before the waiter times out. For example, for 300 seconds, use `300s`.
- `[SUCCESS_PATH_PREFIX]` is the path prefix to watch for a success condition.
- `[SUCCESS_NUMBER]` is the number of variables that exist under this path to be considered successful.

Polling a waiter

After creating a waiter, poll the related Operations resource

(`/deployment-manager/runtime-configurator/reference/rest/v1beta1/projects.configs.operations`) to

check whether the waiter has met one of the end conditions. If the waiter has met an end condition or has timed out, the operation returns as done and returns a response based on the results of the waiter.

If a newly-created waiter's operation finishes immediately, check whether the current state of your variables all either end condition.

Use `gcloud` or the API to poll a waiter.

`gcloudAPI` (#api)

With the `gcloud` command-line tool, when you make a request to create a waiter, the tool automatically performs polling and waits for the waiter to return. The tool prints a response like the following while it is polling the waiter:

```
Waiting for waiter [WAITER_NAME] to finish...
```

If you don't want the tool to poll the waiter after creating it, supply the `--async` flag with your creation request.

Waiter responses

Successful end condition

If the waiter has met a **successful** end condition, the operation returns the Waiter resource:

```
me": "projects/[PROJECT_NAME]/configs/[CONFIG_NAME]/operations/waiters/[WAITER_NAME]
ne": true,
sponse": {
@type": "type.googleapis.com/google.cloud.runtimeconfig.v1beta1.Waiter",
name": "projects/[PROJECT_NAME]/configs/[CONFIG_NAME]/waiters/[WAITER_NAME]",
timeout": "360.000s",
failure": {
  "cardinality": {
```

```
  "path": "[SUCCESS_PATH_PREFIX]",
  "number": "[SUCCESS_NUMBER]"
}
,
success": {
  "cardinality": {
    "path": "[FAILURE_PATH_PREFIX]",
    "number": [FAILURE_NUMBER]
  }
}
,
createTime": "2016-04-12T18:02:13.316695490Z",
done": true
```

Failure condition

If the waiter meets the **failure** end condition or timed out, the operation returns an error.

Failure condition was met

```
me": "projects/[PROJECT_NAME]/configs/[CONFIG_NAME]/operations/waiters/[WAITER_NAME]",
ne": true,
ror": {
  code": 9,
  message": "Failure condition satisfied."
```

Waiter timed out

```
me": "projects/[PROJECT_NAME]/configs/[CONFIG_NAME]/operations/waiters/[WAITER_NAME]",
ne": true,
ror": {
  code": 4,
  message": "Timeout expired."
```

What's next

- Learn about [Runtime Configurator](/deployment-manager/runtime-configurator/) (/deployment-manager/runtime-configurator).
- Learn how to [set and get variables](/deployment-manager/runtime-configurator/set-and-get-variables/) (/deployment-manager/runtime-configurator/set-and-get-variables).
- Set a [Watcher a specific variable](/deployment-manager/runtime-configurator/watching-a-variable/) (/deployment-manager/runtime-configurator/watching-a-variable).
- [Create and delete RuntimeConfig Resources](/deployment-manager/runtime-configurator/create-and-delete-runtimeconfig-resources/) (/deployment-manager/runtime-configurator/create-and-delete-runtimeconfig-resources).
- Refer to the [v1beta1 reference](/deployment-manager/runtime-configurator/reference/rest/) (/deployment-manager/runtime-configurator/reference/rest).
- Refer to the [Quotas for Runtime Configurator](/deployment-manager/pricing-and-quotas#runtime_configurator/) (/deployment-manager/pricing-and-quotas#runtime_configurator).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-27 UTC.