

# Cloud Monitoring Release Notes

API v1.4 (Aug 22, 2012)



[docs.rackspace.com/api](https://docs.rackspace.com/api)

Copyright © 2011, 2012 Rackspace All rights reserved.

RACKSPACE MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS DOCUMENT AND RESERVES THE RIGHT TO MAKE CHANGES TO SPECIFICATIONS AND PRODUCT/SERVICES DESCRIPTION AT ANY TIME WITHOUT NOTICE. RACKSPACE SERVICES OFFERINGS ARE SUBJECT TO CHANGE WITHOUT NOTICE. USERS MUST TAKE FULL RESPONSIBILITY FOR APPLICATION OF ANY SERVICES MENTIONED HEREIN. EXCEPT AS SET FORTH IN RACKSPACE GENERAL TERMS AND CONDITIONS AND/OR CLOUD TERMS OF SERVICE, RACKSPACE ASSUMES NO LIABILITY WHATSOEVER, AND DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO ITS SERVICES INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT.

Rackspace®, Rackspace logo and Fanatical Support® are registered service marks of Rackspace US, Inc. All other product names and trademarks used in this document are for identification purposes only and are property of their respective owners.



- *Retrieve overview by `entityId` or `uri`:* Include query strings `uri` or `entityId` to retrieve specific mapped Rackspace Cloud Servers (for now). You can also pull down a set of entities by referencing their id's. Keep in mind you cannot combine pulling both `uri`'s and `entityId`'s in the same request, but you can have up to 100 of them in a single request.

Overview list by `entityId` and Overview list by `uri`.

### Enhancements

- Added friendly names for the `alarm_examples` API.
- Add a "metadata" attribute to the `alarm` API.
- Allow including no conditionals in the `alarm language`, allowing you to set global filters, and rely on the check availability to determine the returned state.

### Resolved Issues

- Fix un-setting optional attributes on an update (PUT).
- Fix the `check_type` -> `alarm` mapping for delivering inconsistent and non-deterministic updates.
- Fixed how we handle zlib streams when inflating a gzip http stream.
- When we combined two services we had briefly did not insert the initial OK state for the `alarm change log`, this is fixed now.
- Fixed updating the `target_alias` after setting the `target_hostname` on the initial creation, and vice-versa.

## Internal Changes

### New Features

- Deployed a staging Graylog2 cluster on the gold config.
- Consecutive count is now working in production, it is not highlighted due to it's lack of stability across restarts.

### Enhancements

- Open many connections to the Cassandra server to up performance.
- Added 3 extra monitoring zones to staging.
- Added 99.9% percentile calculation to our internal metrics system.
- Deployed data warehouser analytics for the EBI team.
- Changed subject on Dead Letter emails so we can eventually add them to their ticket queue.



These release notes correspond to the pre "Unlimited Availability" release of Cloud Monitoring.

- **Traceroute API:** This API lets you run a Traceroute from a Monitoring Zone to a Hostname or IP address. Like all Cloud Monitoring features, the Traceroute API is fully dual stack, supporting both IPv4 and IPv6. The Traceroute API can be used to debug networking issues between the Cloud Monitoring collectors and your infrastructure.

- *Metric interpolation on the Alarm status string:* This allows you to include metrics from a Check in the status string returned by an Alarm. For example, to include the HTTP status code from a `remote.http` Check, your Alarm could look like this:

Alarm DSL Docs

- **Add payload to the remote.http check:** This allows you send a request body during a HTTP/HTTPS request.

## Enhancements

- Allow a user to specify the number of ICMP packets to send out on the `remote.ping` check.
- Updated Certificate Authority chain used to validate SSL certificates, resulting in less false positives.
- `remote.tcp` check now features the `body_match_metric` when specifying the `send_body` command.
- Add the remote IP address of the collector that alerted you to the Webhook notification payload.

- Fixed writable date and time fields that were supposed to be immutable.
- Fixed the DSL string escaping to allow you to specify special characters, example below:

---

4







# v1.2, May 10, 2012

These release notes correspond to the post-EAP release of Cloud Monitoring.

## New Features

- **Test an Existing Check**

Test execute an existing check. This allows you to execute a check like the traditional test check, but on a check that is already running. This will not actually execute the new check in the same check context, it executes in a new context.

- **Alarm Templates**

Get a list of alarm recipes with templating capabilities. This helps kick-start your thinking on all the possible ways to leverage this incredibly powerful part of the monitoring system. The link to the API documentation is here:

[Alarm Examples API Documentation](#)

- **Cascading Entity Deletion**

Delete entities and now have it cascade to all of the associated checks and alarms in the system. This allows you to de-provision monitors quickly and easily without having to manually make the individual calls yourself. This saves you time and effort.

- **Type Checking on Alarms**

Type check alarms so you receive alerts if you set up an incorrect alarm, or it isn't as expected. As a bonus the alarm also validates all the metrics you are checking are present, if they are not present, it automatically changes the alarm state to CRITICAL.

## Enhancements

- Fixed a defect in the networking partitioning logic that leads to more consistent operation.
- Added more collector infrastructure in IAD.
- Added a meta-data field on the check type.

## Resolved Issues

- Alarm states have a race condition. During replay it's possible to unintentionally store duplicates.
- Fix missing version parameter in next href.
- Fix banner match to be more consistent, be present but empty if it didn't match.
- Removed the ability to use duplicate monitoring zones.







# v1.0, December 15, 2011

These release notes correspond to the Private Beta for Rackspace Cloud Monitoring.

## Product Features

- **Multiple Monitoring Zones Perspective**

Monitor a single resource from many different monitoring zones.

- **Alarm History**

Use an API to retrieve time series data from the system. Use this to go back in time and verify which notifications went out and why.

- Remote IPv4/v6 Checks

Monitor IPv4/IPv6 services via IP addresses or using DNS resolution. Using DNS resolution, specify which type of address to resolve, A or AAAA.

- **Data Model**

We've disconnected the concept of alerting and data collection so you can create robust and rich monitors. Check many aspects of a single resource; for instance, make sure a site is responding with a 201-status code, and features the copyright as a body match on the bottom of the page. Ensure all these conditions are met before the status of an alarm is "OK."

- **Alarm Language**

Create a unique query with our language purposefully built to express thresholds and alert conditions. String compare metrics, use a regular expression operator, check the rate of change, and much more. With each of these primitives you can chain together multiple conditionals to set the state of an alarm. Our documentation highlights best practices for building robust monitors and alarms.

- **On-demand Simulation**

Run a check on-demand, to verify its status before adding it. This allows you to get a feel for a monitor before being alerted by it. On top of that, you can take the results and pass it another endpoint to simulate your alarm language criteria match! This is extremely powerful as it takes some of the complexity out of guessing and allows you to simulate responses as they happen in normal operation.

- **Audit Log**

Track all changes to an entity, check and alarm. Get the before and after of every changed object in the system. You can even add a "who" and "why" parameter to each POST/PUT request to keep a log of why the changes were made.

- **Change Alert Destination**

