CHAPTER 8. EXPOSING 3SCALE APICAST METRICS TO PROMETHEUS

IMPORTANT

For this release of Red Hat 3scale, Prometheus installation and configuration is not supported. Optionally, you can use the community version of Prometheus to visualize metrics and alerts for APIcast-managed API services.

8.1. ABOUT PROMETHEUS

Prometheus is an open-source systems monitoring toolkit that you can use to monitor 3scale APIcast services deployed in the Red Hat OpenShift environment.

If you want to monitor your services with Prometheus, your services need to expose a Prometheus endpoint. This endpoint is an HTTP interface that exposes a list of metrics and the current value of the metrics. Prometheus periodically scraps these target-defined endpoints and writes the collected data into its database. In the Prometheus UI, you can then write queries in Prometheus Query Language (PromQL) to extract metric information. With PromQL, you can select and aggregate time series data in real time.

For example, you can use the following query to select all the values that Prometheus has recorded within the last 5 minutes for all time series that have the metric name **http_requests_total**:

```
http_requests_total[5m]
```

You can further define or filter the results of a query by specifying a *label* (a key:value pair) for the metric. For example, you can use the following query to select all the values that Prometheus has recorded within the last 5 minutes for all time series that have the metric name **http_requests_total** and a **job** label set to **integration**:

```
http_requests_total{job="integration"}[5m]
```

The result of a query can either be shown as a graph, viewed as tabular data in Prometheus's expression browser, or consumed by external systems via the Prometheus HTTP API. Prometheus provides a graphical view of the data. For a more robust graphical dashboard to view Prometheus metrics, Grafana is a popular choice.

You can also use the the PromQL language to configure alerts in the Prometheus alertmanager tool.

NOTE

Grafana is a community-supported feature. Deploying Grafana to monitor Red Hat 3scale products is not supported with Red Hat production service level agreements (SLAs).

8.2. APICAST INTEGRATION WITH PROMETHEUS

APIcast integration with Prometheus is available for the following deployment options:

- Self-managed APIcast (both with hosted or on-premises API manager)

- Built-in APIcasts in on-premises

Note that APIcast integration with Prometheus is not available in hosted API manager and hosted APIcast.

By default, Prometheus can monitor the APIcast metrics listed in Table 8.2, “Prometheus Default Metrics for 3scale APIcast”.

Optionally, if you have **cluster admin** access to the OpenShift cluster, you can extend the **total_response_time** _seconds, **upstream_response _time_seconds**, and **upstream_status** metrics to include **service_id** and **service_system_name** labels. To extend these metrics, set the **APICAST_EXTENDED_METRICS** OpenShift environment variable to **true** with this command:

```
oc set env dc/apicast APICAST_EXTENDED_METRICS=true
```

If you use the APIcast Batch Policy (described in Section 4.1.2, “3scale Batcher policy”), Prometheus can also monitor the metrics listed in Table 8.3, “Prometheus Metrics for 3scale APIcast Batch Policy” .

NOTE

If a metric has no value, Prometheus hides the metric. For example, if **nginx_error_log** has no errors to report, Prometheus does not display the **nginx_error_log** metric. The **nginx_error_log** metric is only visible if it has a value.

Additional resources

> For information about how to use Prometheus, go to: https://prometheus.io/docs/prometheus/latest/getting_started/

8.3. OPENSHIFT ENVIRONMENT VARIABLES FOR 3SCALE APICAST

To configure your 3scale APIcast service’s Prometheus instance, you can set the OpenShift environment variable described in Table 8.1, “Prometheus Environment Variables for 3scale APIcast” .

Table 8.1. Prometheus Environment Variables for 3scale APIcast

Environment Variable	Description	Default

Environment Variable	Description	Default
APICAST_EXTENDED_METRICS	A boolean value that enables additional information on Prometheus metrics. The following metrics have the service_id and service_system_name labels which provide more in-depth details about APIcast: ● **total_response_time_seconds** ● **upstream_response_time_seconds** ● **upstream_status**	**false**

Additional resources

For information on setting environment variables, see the relevant OpenShift guide:

- The OpenShift 4.x *Applications* guide (https://access.redhat.com/documentation/en-us/openshift_container_platform/4.2/html-single/applications/)
- The OpenShift 3.1.1 *Developer Guide* (https://access.redhat.com/documentation/en-us/openshift_container_platform/3.11/html/developer_guide/)

8.4. 3SCALE APICAST METRICS EXPOSED TO PROMETHEUS

After you set up Prometheus to monitor 3scale APIcast, by default it can monitor the metrics listed in in Table 8.2, “Prometheus Default Metrics for 3scale APIcast” .

The metrics listed in Table 8.3, “Prometheus Metrics for 3scale APIcast Batch Policy” are only available when you use the Section 4.1.2, “3scale Batcher policy” .

Table 8.2. Prometheus Default Metrics for 3scale APIcast

Metric	Description	Type	Labels
nginx_http_connections	Number of HTTP connections	gauge	state(accepted,active,handled,reading,total,waiting,writing)
nginx_error_log	APIcast errors	counter	**level(debug,info,notice,warn,error,crit,alert,emerg)**
openresty_shdict_capacity	Capacity of the dictionaries shared between workers	gauge	**dict**(one for every dictionary)

Metric	Description	Type	Labels
openresty_shdict_free_space	Free space of the dictionaries shared between workers	gauge	**dict**(one for every dictionary)
nginx_metric_errors_total	Number of errors of the Lua library that manages the metrics	counter	*none*
total_response_time_seconds	Time needed to sent a response to the client (in seconds) **Note**: To access the **service_id** and **service_system_name** labels, you must set the **APICAST_EXTENDED_METRICS** environment variable to **true** as described in Section 8.2, "APIcast integration with Prometheus".	histogram	**service_id**, **service_system_name**
upstream_response_time_seconds	Response times from upstream servers (in seconds) **Note**: To access the **service_id** and **service_system_name** labels, you must set the **APICAST_EXTENDED_METRICS** environment variable to **true** as described in Section 8.2, "APIcast integration with Prometheus".	histogram	**service_id**, **service_system_name**

Metric	Description	Type	Labels
upstream_status	HTTP status from upstream servers **Note**: To access the **service_id** and **service_system_name** labels, you must set the **APICAST_EXTENDED_METRICS** environment variable to **true** as described in Section 8.2, "APIcast integration with Prometheus".	counter	**status**, **service_id**, **service_system_name**
threescale_backend_calls	Authorize and report requests to the 3scale backend (Apisonator)	counter	**endpoint**(**authrep**, **auth**, **report**), **status**(**2xx**, **4xx**, **5xx**)

Table 8.3. Prometheus Metrics for 3scale APIcast Batch Policy

Metric	Description	Type	Labels
batching_policy_auths_cache_hits	Hits in the auths cache of the 3scale batching policy	counter	*none*
batching_policy_auths_cache_misses	Misses in the auths cache of the 3scale batching policy	counter	*none*