Red Hat 3scale API Management 2.4

Access Control

Define your API, create as many plans as you need and set up limits and pricing rules.

Last Updated: 2019-05-07

Red Hat 3scale API Management 2.4 Access Control

Define your API, create as many plans as you need and set up limits and pricing rules.

Legal Notice

Copyright © 2019 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This guide documents aspects of access control with Red Hat 3scale API Management 2.4.

Table of Contents

CHAPTER 1. DEFINING YOUR API (METHODS AND METRICS)

To define your API on 3scale, navigate to **[Your_API_service] > Integration > Methods & Metrics**and add your methods and metrics.

Metrics lets you track the usage of your API in 3scale. *Hits* is the built-in metric, it exists in each API service and is used to track the hits made to your API. You can achieve finer granularity for the API usage tracking by defining **Methods** under the *Hits* metric. Reporting traffic to a method will increase counters for the method and for the *Hits* metric automatically. You can define separate methods for each endpoint of you API, or a combination of endpoint and HTTP method. See Mapping rules section to learn how to map the endpoints of your API to the methods defined here.

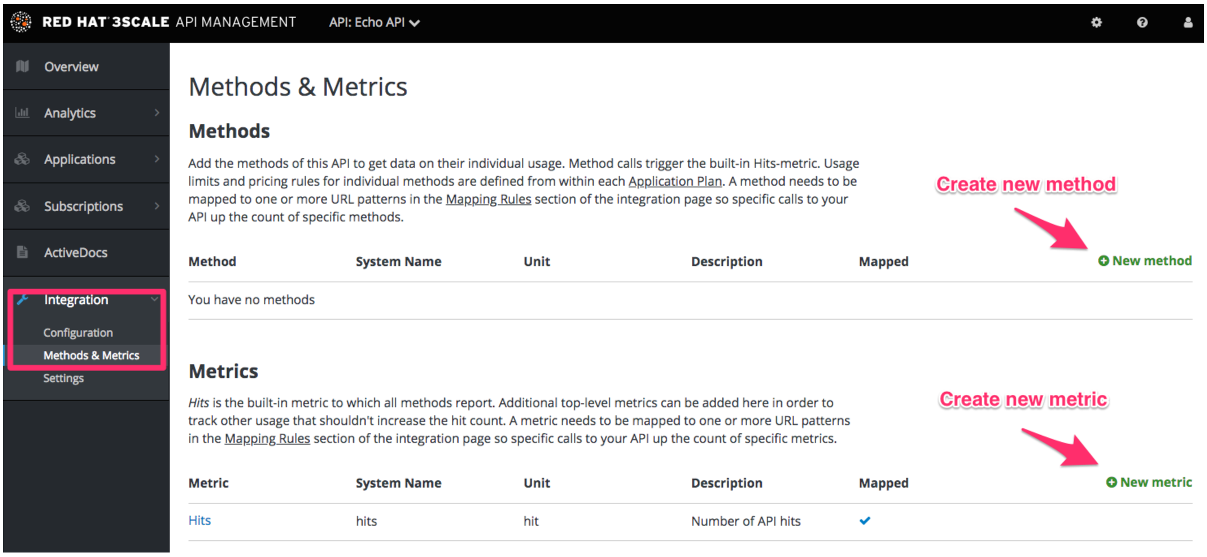
For measuring other, not hit-based usage of your API, you can define new **Metrics** and report the usage in different units. A unit can be anything meaningful: megabytes, CPU time, number of elements returned by the API etc.

Methods and metrics are also the scaffolding to package your API: each application plan lets you define different usage limits and pricing rules for each method and metric.

You will be able to see the the usage reported to metrics and methods in the Analytics section.

1.1. MANUALLY ADD METHODS AND METRICS

1. Navigate to **[Your_API_name] > Integration > Methods & Metrics**



Methods & Metrics

Methods

Add the methods of this API to get data on their individual usage. Method calls trigger the built-in Hits-metric. Usage limits and pricing rules for individual methods are defined from within each Application Plan. A method needs to be mapped to one or more URL patterns in the Mapping Rules section of the integration page so specific calls to your API up the count of specific methods.

Create new method

Method	System Name	Unit	Description	Mapped
You have no methods				

Metrics

Hits is the built-in metric to which all methods report. Additional top-level metrics can be added here in order to track other usage that shouldn't increase the hit count. A metric needs to be mapped to one or more URL patterns in the Mapping Rules section of the integration page so specific calls to your API up the count of specific metrics.

Create new metric

Metric	System Name	Unit	Description	Mapped
Hits	hits	hit	Number of API hits	✓

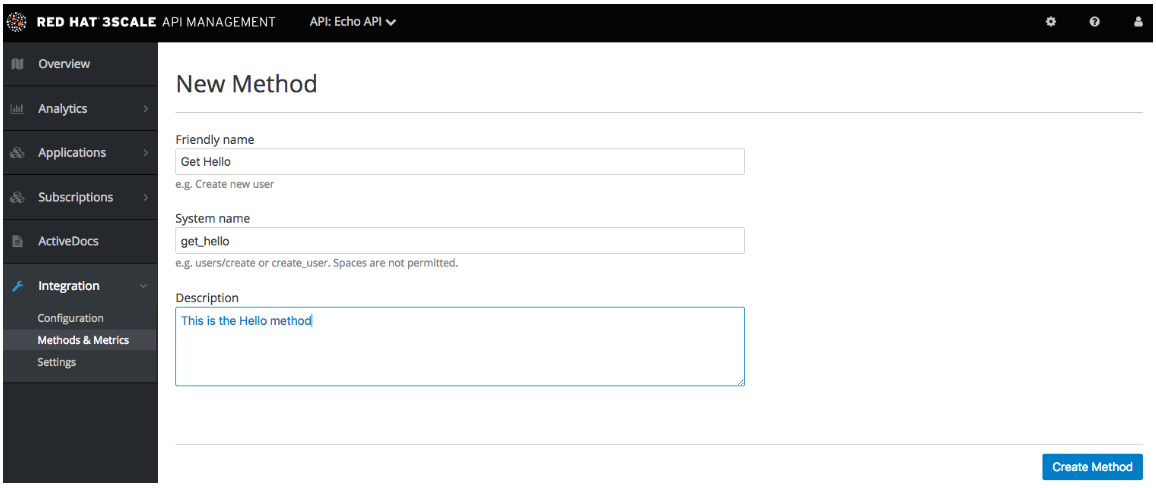
2. Click on New method.

3. Specify the parameters:

   - Friendly name is a short description of the method, it appears in different sections of the 3scale admin portal. This name must be unique for the service.

   - System name is the name of the method which will be used to report the usage through 3scale Service Management API. It also must be unique, and it should only contain alphanumeric characters, underscore _, hyphen **-** and forward slash / without spaces. Other than that, you are free to decide how the system name will look like, it can be exactly the same as the endpoint ("/status"), or for example can include the method and the path ("GET_/status").

- The Description field can be used for a more detailed description of the method, it is optional.



New Method

Friendly name
Get Hello
e.g. Create new user

System name
get_hello
e.g. users/create or create_user. Spaces are not permitted.

Description
This is the Hello method

4. Finally, click on Create Method button.

You can later change the definition of the method. Just click on the method name (in the column **Method**), update the fields and click on Update Method.

Be very careful with changing the system name of the methods and metrics or deleting them! It may break your already deployed 3scale integration, if there are mapping rules poining to the previous system name of the method.

For creating a new metric, click on New metric and provide the required parameters. When specifying the unit, use singular noun (e.g. "hit"), as it will be pluralized automatically in the Analytics charts.

These new methods and metrics will be available in all your current and future plans. You can now edit limits and pricing rules for them on each plan going to **[Your_API_service]** > **Applications** > **Application Plans** > **[plan_you_want_to_edit]**.

1.2. IMPORT YOUR METHODS AND METRICS AUTOMATICALLY

If your API has a lot of endpoints, we offer two additional ways of automatically creating your methods and metrics on 3scale:

- Importing via Swagger spec
- Importing via RAML spec

CHAPTER 2. APPLICATION PLANS

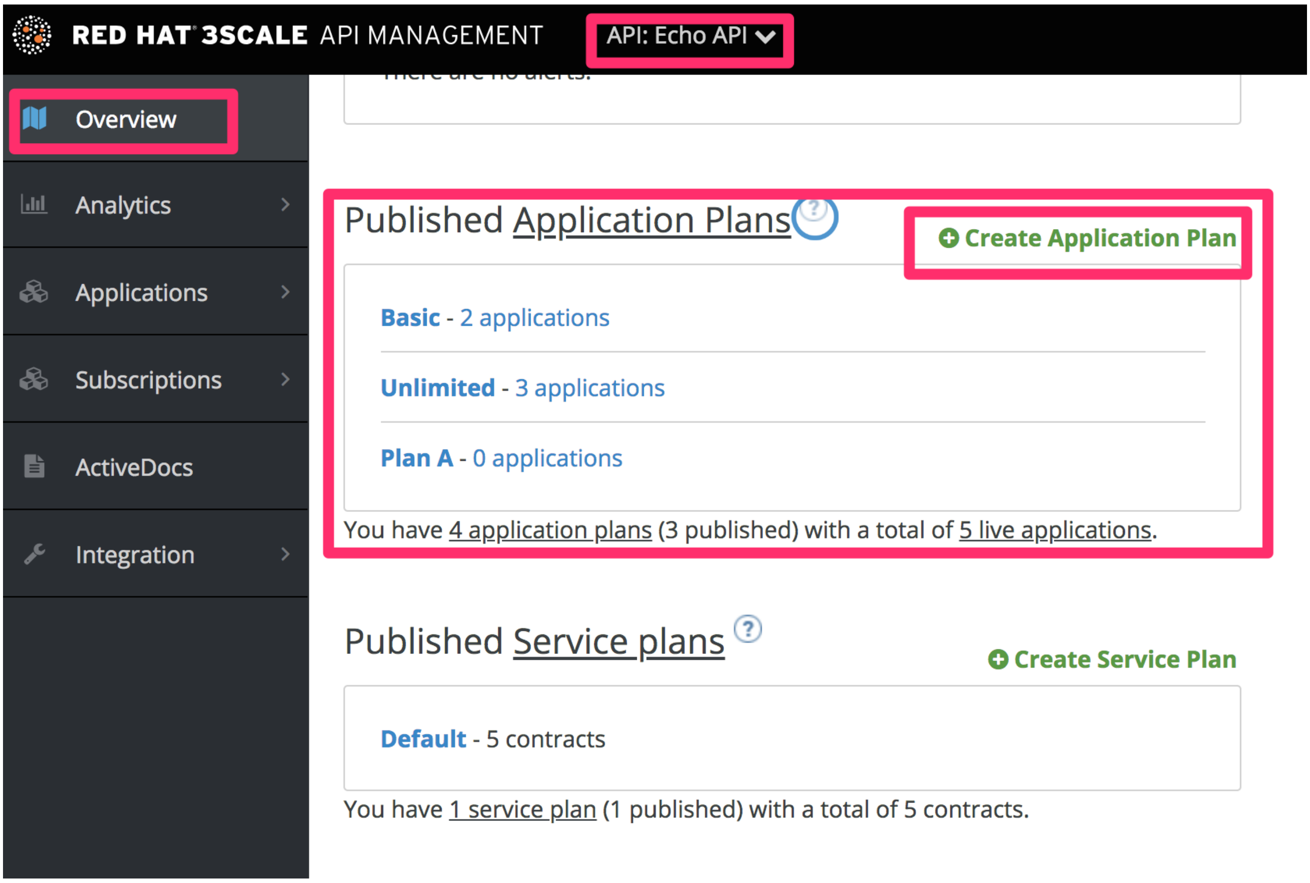
Application Plans define the different sets of access rights you might want to allow for consumers of your API. These can determine anything from rate limits, which methods or resources are accessible and which features are enabled.

2.1. HOW TO CREATE AN APPLICATION PLAN

By default, when your 3scale account is created, you are given two plans: Basic and Unlimited. You can keep and edit these or create your own. You can create as many plans as you need.

To create a new application plan, follow these steps:

1. Navigate to **[Your_API_service] > Applications > Application Plans**
2. Click 'Create Application Plan'.



Published Application Plans

Basic - 2 applications

Unlimited - 3 applications

Plan A - 0 applications

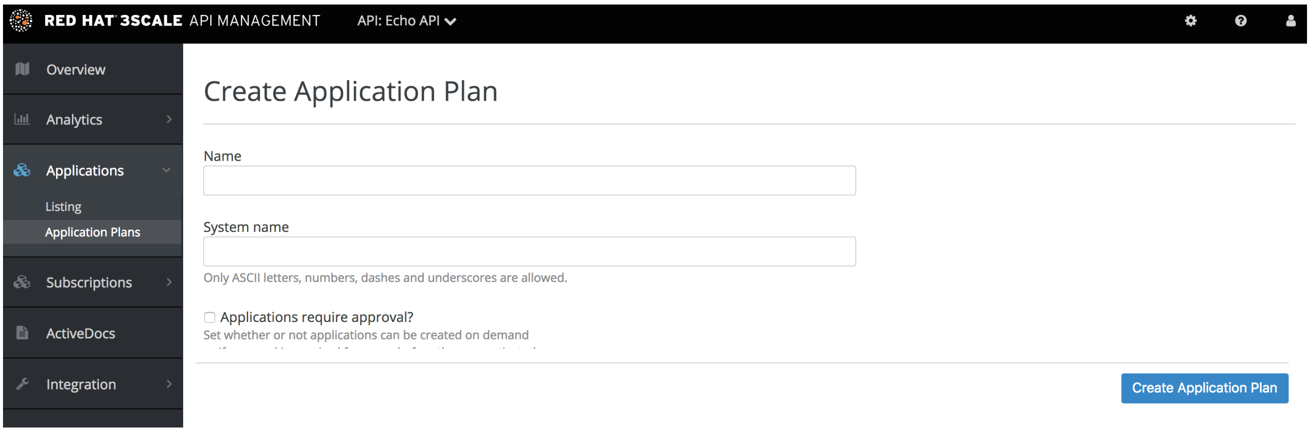
You have 4 application plans (3 published) with a total of 5 live applications.

Published Service plans

Default - 5 contracts

You have 1 service plan (1 published) with a total of 5 contracts.

In the next screen, pick a name and a system name (system names must be unique) for your new plan. If the **Applications require approval?** checkbox is selected, no applications will be able to access your API without approval.



Create Application Plan

Name

System name

Only ASCII letters, numbers, dashes and underscores are allowed.

☐ Applications require approval?

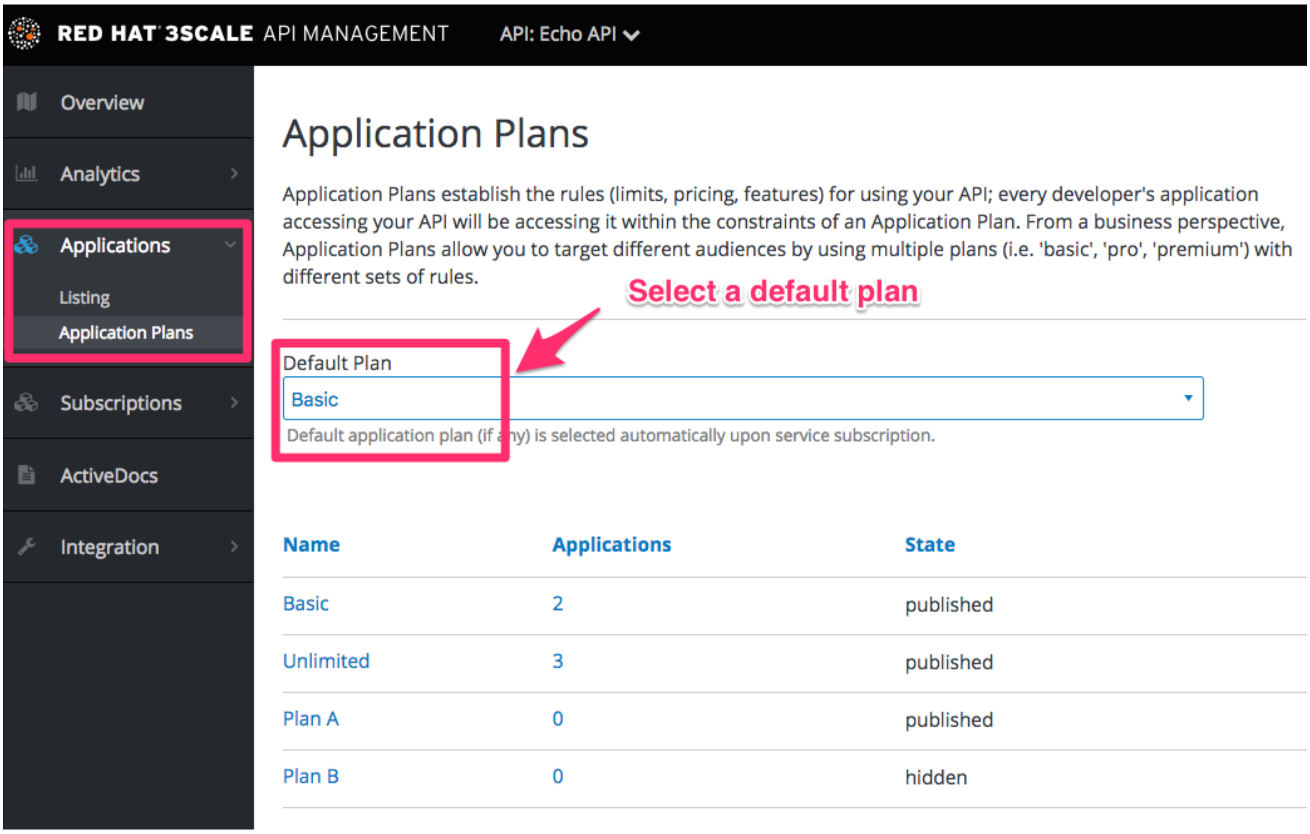
Set whether or not applications can be created on demand

Create Application Plan

Once you have created a plan, you can provision rate limits and set up paid plans.

2.2. SETTING UP A DEFAULT APPLICATION PLAN

After you have created all your plans, you can select a default plan for when people sign up to register their applications. To do so, navigate to **[Your_API_service] > Applications > Application Plans** and select the default plan:



Application Plans

Application Plans establish the rules (limits, pricing, features) for using your API; every developer's application accessing your API will be accessing it within the constraints of an Application Plan. From a business perspective, Application Plans allow you to target different audiences by using multiple plans (i.e. 'basic', 'pro', 'premium') with different sets of rules.

Default Plan: Basic

Default application plan (if any) is selected automatically upon service subscription.

Name	Applications	State
Basic	2	published
Unlimited	3	published
Plan A	0	published
Plan B	0	hidden

If you don’t indicate a default application plan, when a new user signs up to get access to your API, they won’t be created an application by default (meaning they won’t really get access to your API).

CHAPTER 3. MAPPING RULES

After defining your API creating methods and metrics, you can map your API endpoints or paths to the methods you have defined in the Definition page. To do so:

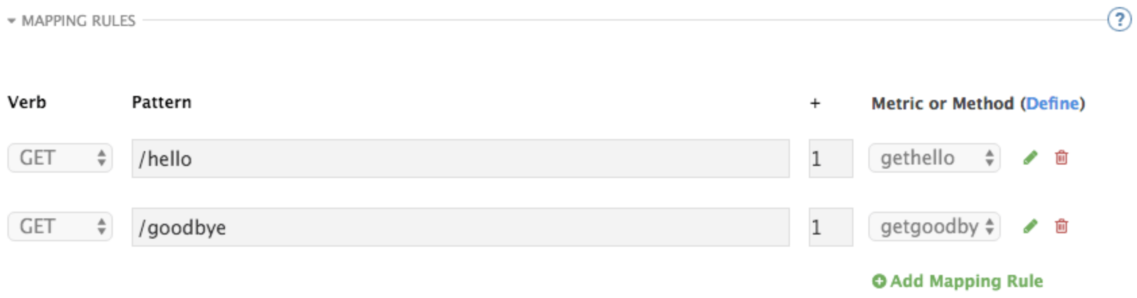
1. Navigate to **[Your_API_service] > Integration > Configuration > Edit APIcast configuration** and find the **Mapping rules** section in the page.

2. Choose the HTTP method, available on the specific endpoint path

3. Select the equivalent method to map against. Different operations (GET, PUT, POST, DELETE, etc…) on the same endpoint can be tracked separately.

The workflow to define mapping rules is as follows:

- Add new rules by clicking on **Add Mapping Rule** link. Then select an HTTP method, a pattern, a metric (or method), and its increment. When you're done, click the **Update & Test Staging Configuration** button.



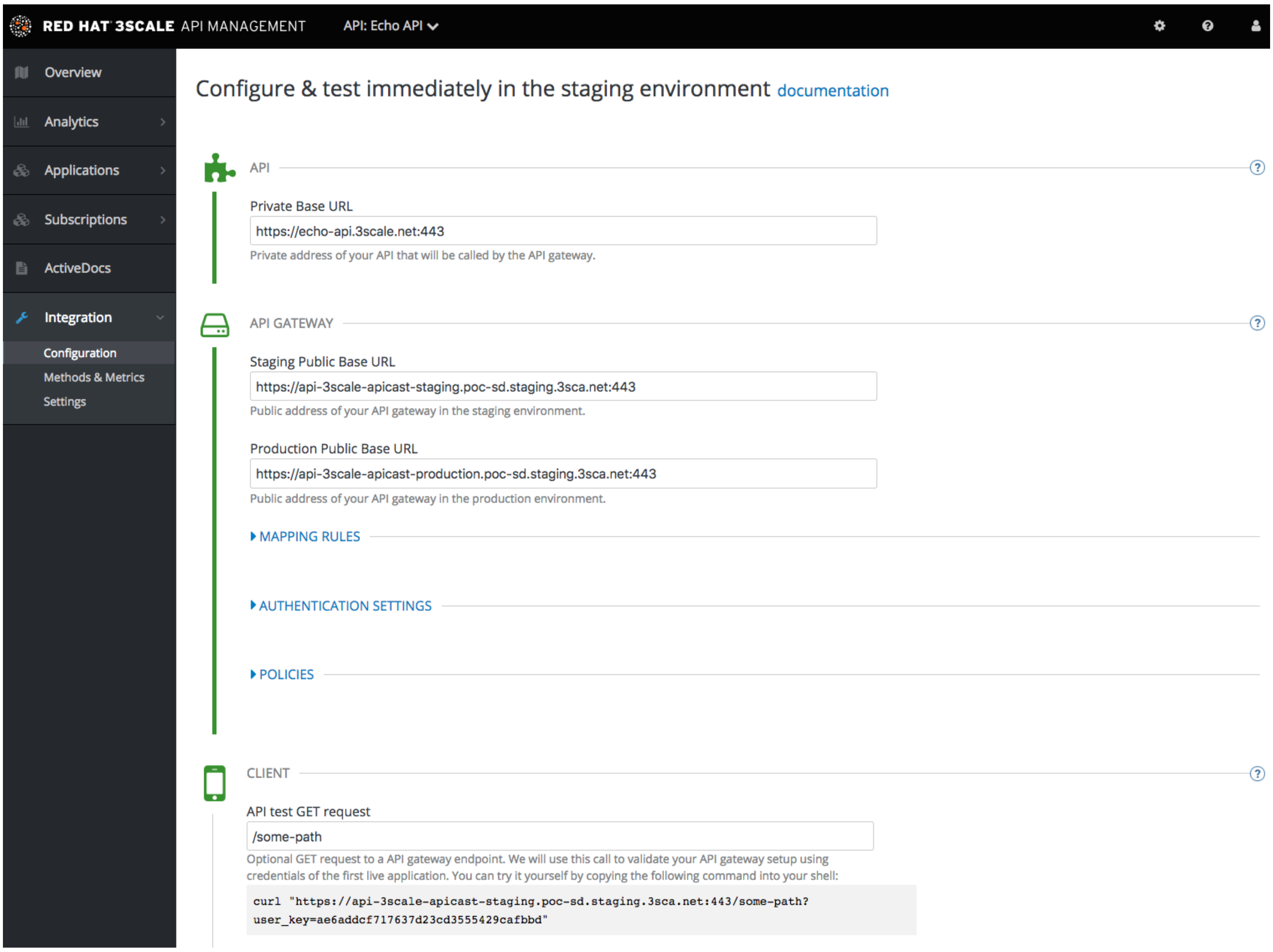
Verb	Pattern	+	Metric or Method (Define)
GET	/hello	1	gethello
GET	/goodbye	1	getgoodby

- Mapping rules will be grayed out on the next reload to prevent accidental modifications.

- To edit an existing mapping rule, you must first enable editing by clicking on the pencil icon on the right. To delete a rule, click on the red trash icon. Edits, modifications, and deletions will be saved when you hit the **Update & Test Staging Configuration**button.

Once the setup is done, you can test your integration with the **Staging APIcast Cloud Gateway** to ensure your setup will work in production.

If all the parameters and mapping rules are set correctly, you should see a green line showing a correct test integration between 3scale and your API Backend.



Configure & test immediately in the staging environment documentation

API

Private Base URL: https://echo-api.3scale.net:443

Private address of your API that will be called by the API gateway.

API GATEWAY

Staging Public Base URL: https://api-3scale-apicast-staging.poc-sd.staging.3sca.net:443

Public address of your API gateway in the staging environment.

Production Public Base URL: https://api-3scale-apicast-production.poc-sd.staging.3sca.net:443

Public address of your API gateway in the production environment.

CLIENT

API test GET request: /some-path

Optional GET request to a API gateway endpoint. We will use this call to validate your API gateway setup using credentials of the first live application. You can try it yourself by copying the following command into your shell:

```
curl "https://api-3scale-apicast-staging.poc-sd.staging.3sca.net:443/some-path?
user_key=ae6addcf717637d23cd3555429cafbbd"
```

CHAPTER 4. PROVISIONING PAID PLANS

One of the most popular ways to monetize an API is by defining subscription fees based on usage. This section focuses on how to use application plans to provision pricing tiers. It's also possible to apply pricing rules at the account and the service level – these topics are covered in advanced guides.

Below you'll learn about the pricing options for application plans and how to set up a paid plan.

4.1. STEP 1: MAKE DECISIONS ABOUT YOUR PRICING MODEL

The first decision to make is how to differentiate between the tiers in your pricing model. Will the tiers be driven by volume/usage, API functionality, access to other resources, or a combination?

- **Volume / Usage.** The most common way to differentiate between tiers is based on volume because volume usually has a strong correlation to value to the customer as well as cost to serve. You can apply a global hit count for calls on the API or a more granular measurement at the method level.

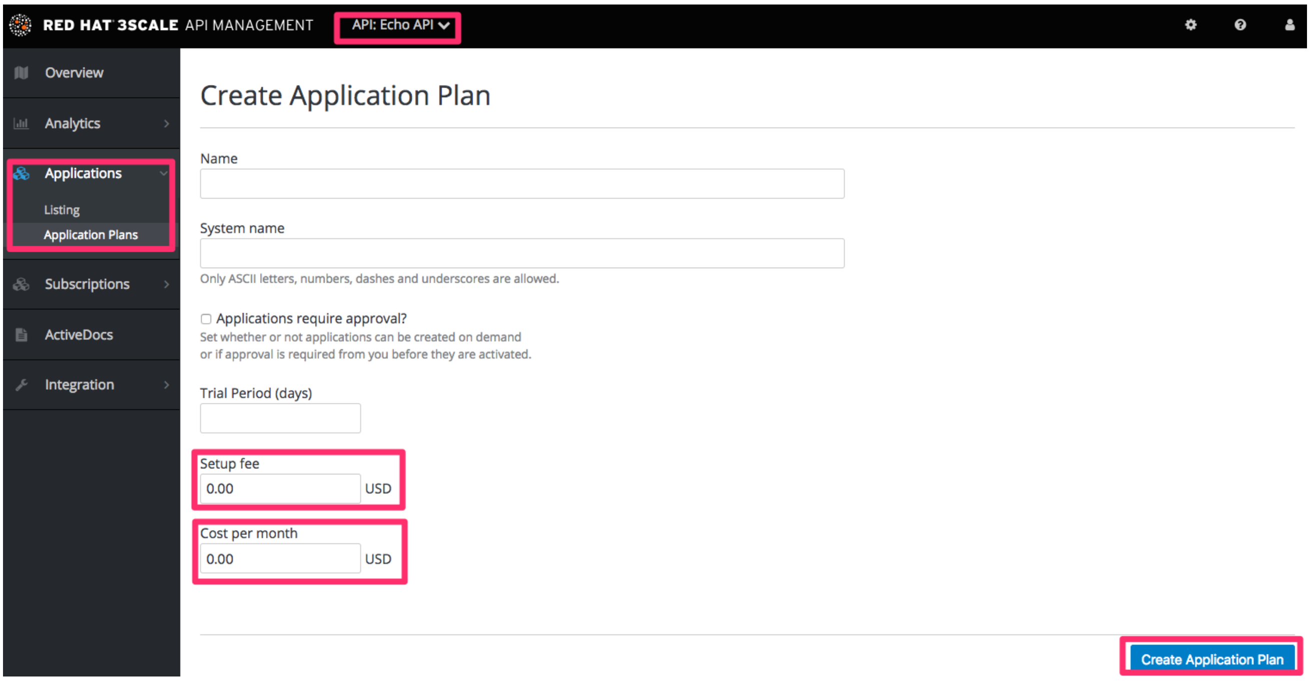
- **Functionality.** You can enable or disable access to parts of your API depending on the tier. This is a good approach to distinguish between standard and premium levels.

- **Resources.** You can also create tiers based on access to any other resources that provide value to the customer or drive costs in your infrastructure – for example, GB's of bandwidth consumed, number of users, or transaction values.

Once you have decided on your pricing drivers, you must decide whether the tiers will be based on a flat rate subscription, a variable rate, or a one-off upfront charge. All three of the pricing drivers above are compatible with the one off, or monthly flat rate subscriptions. If you decide your pricing will be based on volume of hits or resource consumption, there will of course be a variable element to your pricing.

4.2. STEP 2: SET UP AN APPLICATION PLAN WITH YOUR PRICING RULES

You can either create a new application plan or edit an existing one. When creating a new application plan, you can enter any upfront charges or flat rate subscriptions.



Create Application Plan

Name

System name

Only ASCII letters, numbers, dashes and underscores are allowed.

□ Applications require approval?
Set whether or not applications can be created on demand
or if approval is required from you before they are activated.

Trial Period (days)

Setup fee
0.00 USD

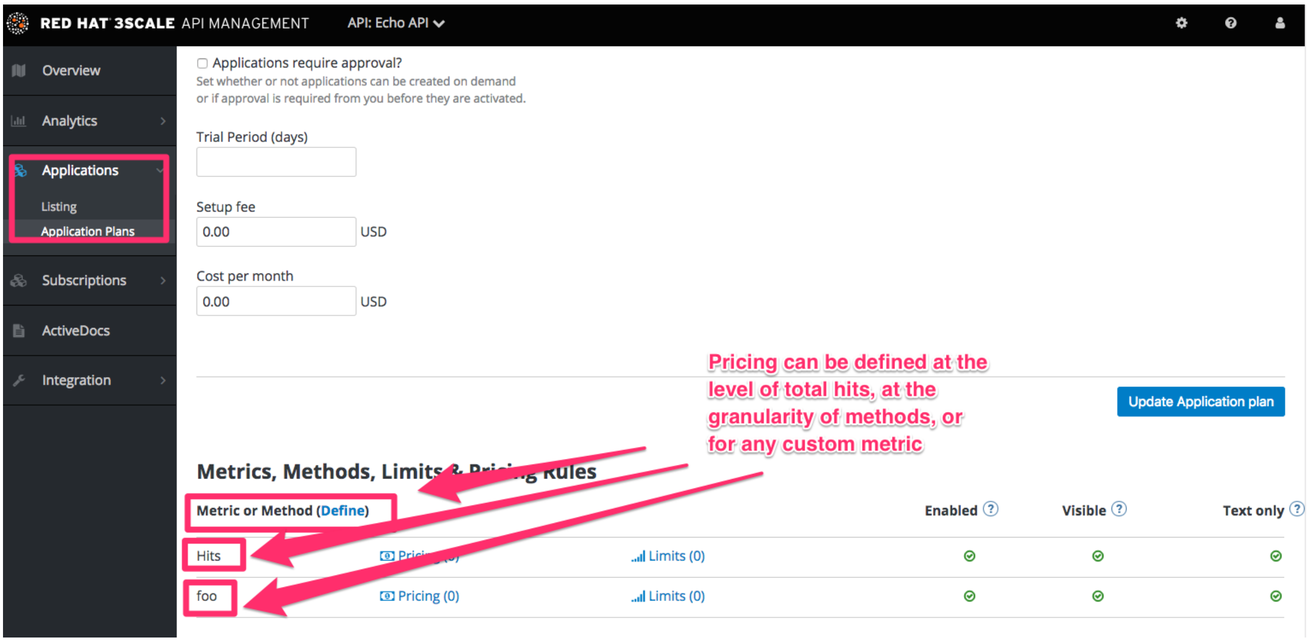
Cost per month
0.00 USD

Create Application Plan

In the edit application view, you can enter or modify the upfront charges and subscriptions.

Next, set up the pricing drivers you decided on in step 1. If some of them already exist as metrics, you can simply edit the item.

- *Volume drivers:* are applied at the level of the global hits metric, or for individual methods under hits. Multiple pricing rules can be applied to any metric. Note that the hits calculation is cumulated over a one-month billing cycle.
- *Functionality drivers* are set by enabling or disabling the metric for this plan.
- *Resource drivers* are similar to volume drivers but are applied on custom metrics.



□ Applications require approval?
Set whether or not applications can be created on demand or if approval is required from you before they are activated.

Trial Period (days)

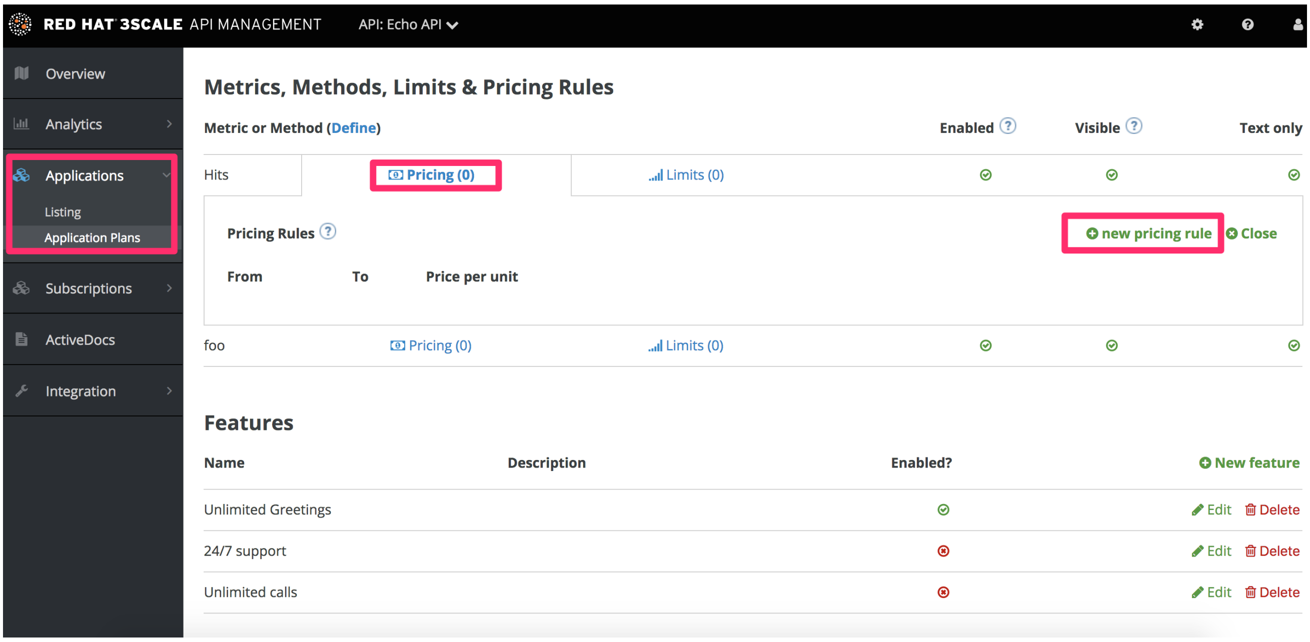
Setup fee 0.00 USD

Cost per month 0.00 USD

Pricing can be defined at the level of total hits, at the granularity of methods, or for any custom metric

Metrics, Methods, Limits & Pricing Rules

Metric or Method (Define)			Enabled	Visible	Text only
Hits	Pricing (0)	Limits (0)	✓	✓	✓
foo	Pricing (0)	Limits (0)	✓	✓	✓



Metrics, Methods, Limits & Pricing Rules

Metric or Method (Define)			Enabled	Visible	Text only
Hits	Pricing (0)	Limits (0)	✓	✓	✓
foo	Pricing (0)	Limits (0)	✓	✓	✓

Pricing Rules

From	To	Price per unit

Features

Name	Description	Enabled?	
Unlimited Greetings		✓	Edit Delete
24/7 support		✗	Edit Delete
Unlimited calls		✗	Edit Delete

Once you’re finished setting up your pricing rules, be sure to click “update application plan”.

4.3. STEP 3: CREATE FURTHER PRICING TIERS

It’s ok to define an API paid plan with a single application plan. Usually this would be the case if all your pricing rules are defined by volume or resource drivers. However if you want to offer separate plans for different segments of your developer community, you’ll need to add more application plans.

The easiest way to do this is to copy the first application plan from the application plan overview page. This way, it will be pre-populated with all the existing metrics and pricing rules. The more care you take to create a full plan the first time, the more time you will save with the plan copy feature.

4.4. STEP 4: PROVISION THE PAID PLANS

In order to provision the plans, your developers must create new applications and select one of the new paid plans. You can also do this on their behalf from the admin console. For any existing applications, it’s also possible to change from an existing plan to one of the new paid plans.

4.5. MORE INFORMATION

In conjunction with flat-rate pricing plans it’s common to differentiate between tiers using rate limits. This is explained in provision rate limits

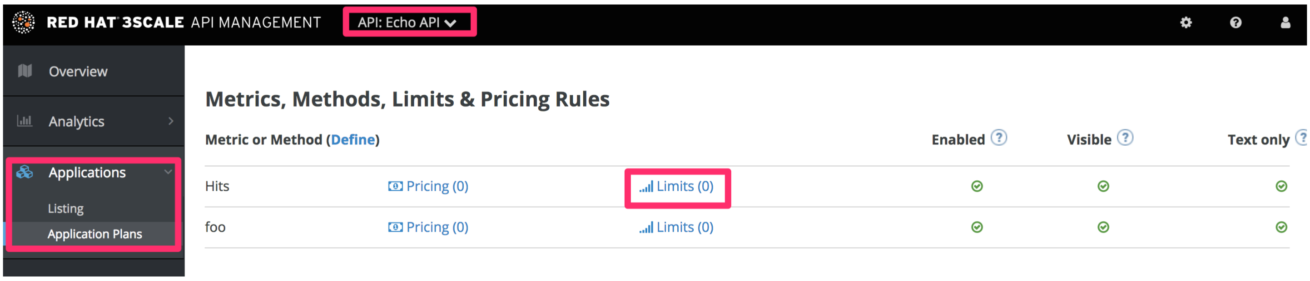
CHAPTER 5. PROVISIONING RATE LIMITS

Rate limits allow you to throttle access to your API resources. You can configure different limits for separate developer segments through the use of application plans.

Once you have rate limits in place, these limits will control the responses a developer receives when they make authorization request calls to the 3scale back end.

5.1. STEP 1: GO TO THE APPLICATION PLAN

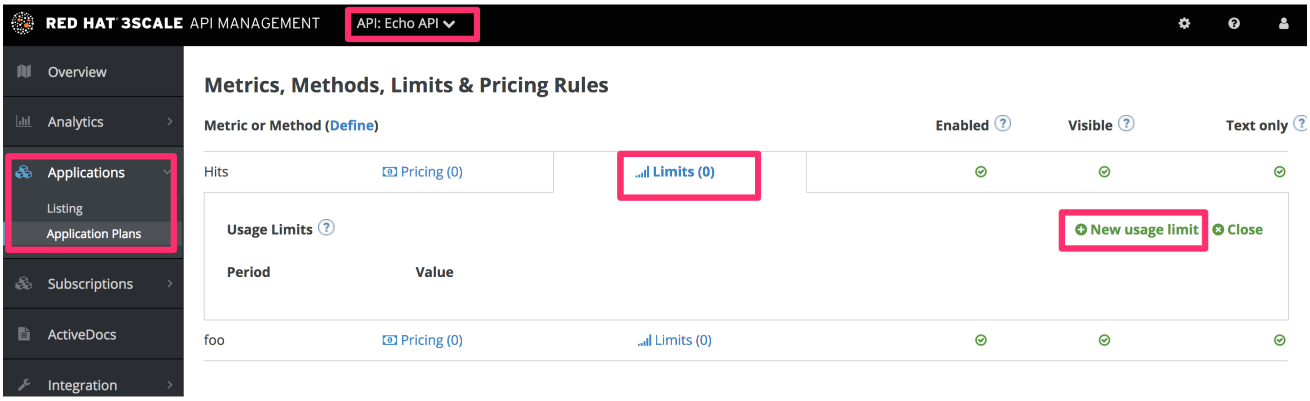
If you do not have an application plan defined yet, create one first. Otherwise, select the plan you want to set rate limits for and click *edit*.



Metrics, Methods, Limits & Pricing Rules

Metric or Method (Define)			Enabled	Visible	Text only
Hits	Pricing (0)	Limits (0)	✓	✓	✓
foo	Pricing (0)	Limits (0)	✓	✓	✓

5.2. STEP 2: SET THE RATE LIMITS



Metrics, Methods, Limits & Pricing Rules

Metric or Method (Define)			Enabled	Visible	Text only
Hits	Pricing (0)	Limits (0)	✓	✓	✓
foo	Pricing (0)	Limits (0)	✓	✓	✓

Usage Limits

Period	Value

5.3. STEP 3: UPDATE THE APPLICATION PLAN

When you are finished setting the limits you require, save your changes by clicking *Update Application plan*.

5.4. STEP 4: PUT THE NEW RATE LIMITS INTO ACTION

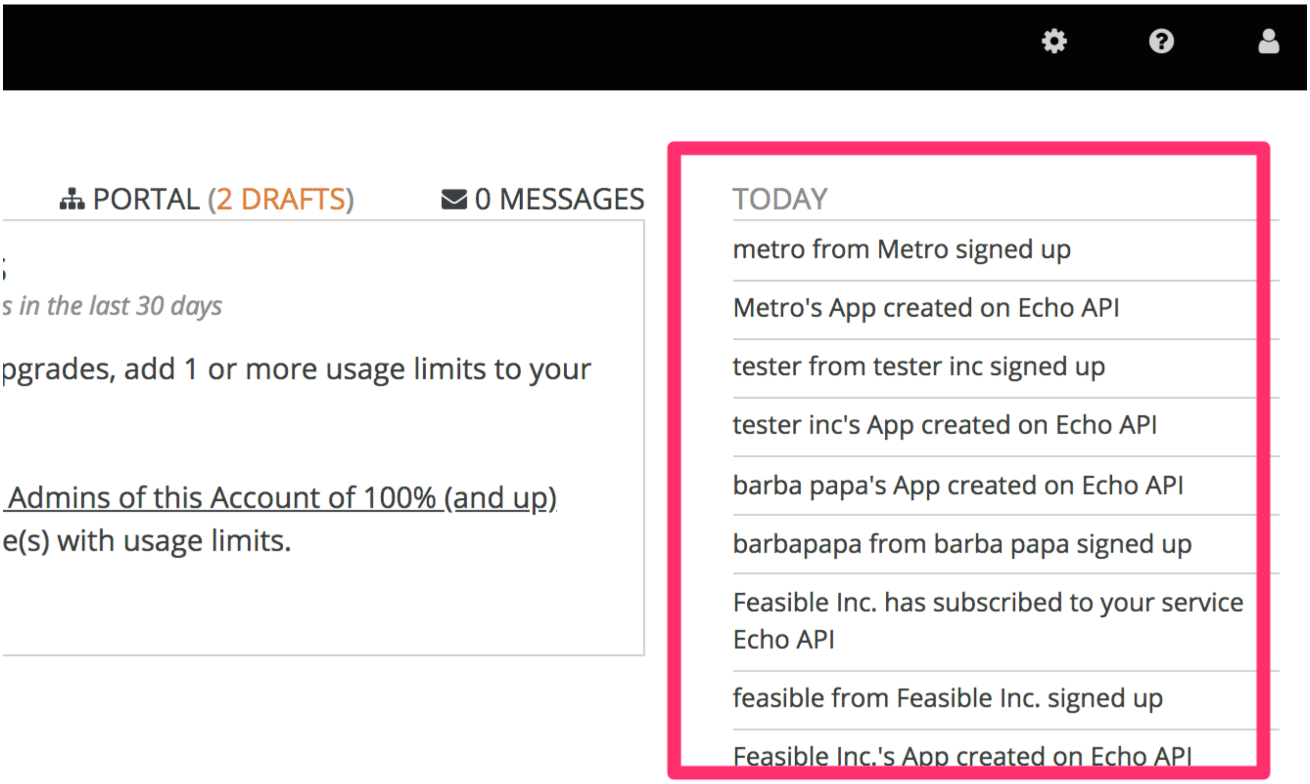
Now that you have your rate limits defined, the following will happen:

- If you have alerts configured, the new limits will be used to decide when notifications are sent.
- When you make authorization calls to the 3scale back end, the limits will be taken into account and the response will be with status code **409**. In this case the APIcast API gateway sends a response to the client with the status code **429 Too Many Requests** and the message, **Limits exceeded**.

Once your rate limits are operational, you will see the users who are reaching the limits on your dashboard, making it quick and easy to check for potential plan upgrade candidates.

NOTE

For more information about soft and hard limits, you should read 1.1.3.2. Configure your API access policies with application plans, window="_blank" in our Quickstart guide.



;
s in the last 30 days

pgrades, add 1 or more usage limits to your

Admins of this Account of 100% (and up)
e(s) with usage limits.

TODAY

metro from Metro signed up

Metro's App created on Echo API

tester from tester inc signed up

tester inc's App created on Echo API

barba papa's App created on Echo API

barbapapa from barba papa signed up

Feasible Inc. has subscribed to your service Echo API

feasible from Feasible Inc. signed up

Feasible Inc.'s App created on Echo API

5.5. MORE INFORMATION

Besides setting rate limits, you can also set variable pricing rules for the same metrics – see provision paid plans