



Red Hat 3scale API Management 2.8

スタートガイド

3scale API Management システムのスタートガイド

Red Hat 3scale API Management 2.8 スタートガイド

3scale API Management システムのスタートガイド

法律上の通知

Copyright © 2020 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, the Red Hat 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 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.

概要

本ガイドでは、3scale の操作を開始する方法について説明します。

目次

第1章 3SCALE の初期設定	3
1.1. プロダクトとバックエンド	3
1.2. 3SCALE ウィザードを使用した最初の API の設定	3
1.3. API の初期設定	4
第2章 API の公開	11
2.1. 最初の API を公開するためのパス	11
2.2. プロトタイプパスの手順	12
2.3. ベーシックパスの手順	14
2.4. アドバンストパスの手順	17
2.5. 実稼働環境への移行	18

第1章 3SCALE の初期設定

本章では、Red Hat 3scale API Management で利用することのできる機能を紹介します。顧客がアクセスする API および内部 API 両方の観点から、API を統合および管理する方法について説明します。詳細について知りたい場合には、各セクションに参考情報へのリンクも用意されています。

1.1. プロダクトとバックエンド

本セクションでは、Red Hat 3scale API Management が API と共にどのように動作するかについて説明します。3scale では、API は 2 つのグループに大別されます。

- **バックエンド/backend**: プロダクトに結び付けられる内部 API。バックエンドにより、API プロバイダーは内部 API の組織構造を自由に 3scale にマッピングすることができます。
- **プロダクト/product**: 顧客がアクセスする API。プロダクトにより、API の利用者に対して強固でシンプルなオフリングを容易に作成することができます。

1 つのプロダクトに複数のバックエンドを含めることや、1 つのバックエンドを複数のプロダクトで使用することができます。つまり、3scale で API を統合および管理するには、以下の 2 つの要素を作成する必要があります。

- 最低でも API の URL が含まれるバックエンド。オプションとして、再使用性を高めるために、バックエンドにマッピングルール、メソッド、およびメトリクスを含めることもできます。
- アプリケーションプランを定義し、APIcast を設定するプロダクト

1.2. 3SCALE ウィザードを使用した最初の API の設定

本セクションでは、プロダクトおよびバックエンドの初期操作を容易にする 3scale ウィザードについて説明します。

前提条件

- 3scale アカウントが必要です。

手順

1. ウィザードを実行します。3scale ウィザードは、2 とおりの方法で実行することができます。

- 3scale アカウントへの初回ログイン時に。
- 以下の URL アドレスの **[Your_admin_domain]** を置き換えてウィザードを実行して。
[https://\[Your-admin-domain\]-admin.3scale.net/p/admin/onboarding/wizard/intro](https://[Your-admin-domain]-admin.3scale.net/p/admin/onboarding/wizard/intro)

たとえば、管理ドメインが **testing-area** であれば、ウィザードは以下の URL から利用することができます。

<https://testing-area-admin.3scale.net/p/admin/onboarding/wizard/intro>

2. **OK, how does 3scale work?** をクリックします。
3. 紹介のアニメーションが表示されます。終了したら、**Got it! Let's add my API** をクリックします。
4. バックエンドを作成します。

- a. バックエンドの名前を指定します。たとえば、**Inventory** と指定します。
 - b. バックエンドのベース URL を指定します。たとえば、<https://echo-api.3scale.net:443>。
 - c. **Add this Backend** をクリックします。
5. プロダクトを作成します。
- a. プロダクト名を指定します。たとえば、**Petstore** と指定します。
 - b. **Add this Product** をクリックします。
6. バックエンドをプロダクトに接続するためのパスを指定し、**Add the Backend to the Product** をクリックします。
- デフォルト値のままにすることができます。デフォルトのパスは / です。
7. GET リクエストによりテストを行うには、**Send request** をクリックします。
- GET メソッドを指定することができます。
8. 上記のステップを完了すると、API Gateway へのリクエストが成功したことを示す確認メッセージが表示されます。これ以降の設定についての詳細を確認するには、**Cool, what's next?** をクリックします。

推奨される追加設定を確認したら、3scale を使用する準備ができています。**Got it! Take me to my API on 3scale** をクリックすると、プロダクトの **Overview** ページが表示されます。

1.3. API の初期設定

本セクションでは、API トラフィックを API キーで保護し、追跡し、3scale に設定した基本的な流量制御および管理機能で監視するための初期設定について、概要を説明します。3scale の操作が初めてであれば、[ウィザードを実行して](#) 最初の API を容易に設定することができます。

1.3.1. 新規プロダクトの作成

プロダクトは顧客がアクセスする API で、1つまたは複数のバックエンドが含まれています。本セクションでは、プロダクトを 3scale に追加する方法について説明します。

以下のオプションのどちらかに従って、新規プロダクトを作成することができます。

- 手動でプロダクトを定義する。
- OpenShift からプロダクトをインポートする。

本セクションでは、手動での定義について詳細に説明します。OpenShift からプロダクトをインポートする場合には、『管理ポータルガイド』の「[サービスディスカバリー](#)」を参照してください。

前提条件

- 3scale アカウントが必要です。

手順

1. Dashboard に移動します。

2. APIS セクションで **Products** タブを選択します。
3. **New Product** をクリックします。
4. 以下の情報を指定します。
 - **Name:** プロダクトの識別子
 - **System name:** 内部での処理に使用される識別子。プロダクトの **system_name** を使用して、プロキシエンドポイントおよびドメイン名が生成されます。デフォルトでは、**system_name** はラベルの一部で、ラベルのパターンは以下のどちらかになります。
 - ステージング環境用 APIcast の場合: **%{system_name}-%{tenant_name}-apicast-staging**
 - 実稼働環境用 APIcast の場合: **%{system_name}-%{tenant_name}-apicast-production**OpenShift が Zync により作成されるルートを許可するには、ラベルの文字数を最大でも 63 文字にする必要があります (パターン内のダッシュを含む)。したがって、**system_name** を指定する場合、前述したパターンの文字数の制限を考慮する必要があります。
 - **Description:** プロダクトに関するより詳しい情報を記述するオプションフィールド
5. **Create Product** をクリックします。

これらの手順を完了すると、一般に公開される API となるプロダクトが作成されます。次のステップは、[バックエンドを作成し プロダクトに追加する](#) ことです。

1.3.2. 新規バックエンドの作成

バックエンドは、プロダクトに結び付けられる内部 API です。本セクションでは、バックエンドを作成してそれをプロダクトに追加する方法について説明します。

前提条件

- 3scale アカウントが必要です。

手順

1. Dashboard に移動します。
2. APIS セクションで **Backends** タブを選択します。
3. **New Backend** をクリックします。
4. 以下の情報を指定します。
 - **Name:** バックエンドの識別子
 - **System name:** 内部での処理に使用される識別子
 - **Description:** バックエンドに関するより詳しい情報を記述するオプションフィールド
 - **Private endpoint** プライベート API のベース URL
5. **Create Backend** をクリックします。

これらの手順を完了すると、内部 API が作成されます。必要なだけのバックエンドを追加することができます。次のステップは、[このバックエンドをプロダクトに追加する](#) ことです。

1.3.3. プロダクトへのバックエンドの追加

本セクションでは、プロダクトへのバックエンドの追加について説明します。プロダクトに追加する各バックエンドについて、この手順を繰り返すことができます。

前提条件

- プロダクト。プロダクトの作成については、[「新規プロダクトの作成」](#)を参照してください。
- 1つまたは複数のバックエンド。バックエンドの作成については、[「新規バックエンドの作成」](#)を参照してください。

手順

1. Dashboard の [Your_product_name] から、**Integration > Backends** の順に移動します。
2. **Add Backend** をクリックします。
3. ドロップダウンリストから既存のバックエンドを選択します。
4. **Path** テキストボックスでルーティングパスを指定します。詳細については、[「バックエンドパス」](#)を参照してください。
5. **Add to Product** をクリックします。

これらの手順を完了すると、プロダクトにバックエンドが追加されます。プロダクトにバックエンドをさらに追加する、あるいは別のプロダクトにバックエンドを追加する場合には、再度この手順を実施することができます。

バックエンドパス

バックエンドをプロダクトに追加する場合、その特定のプロダクトに関してバックエンドが使用するパスを定義します。このパスは、バックエンドの一部ではありません。

[「プロダクトへのバックエンドの追加」](#)に記載の手順から、APIcast はステップ 4 で指定したバックエンドのパスを使用します。APIcast はリクエストされたエンドポイントパスを前方一致で照合し、指定したパスを使用してトラフィックをバックエンドにリダイレクトします。

バックエンドのパスを定義する際の留意事項は以下のとおりです。

- バックエンドの1つのパスとして、/を指定することができます。
- 1つのプロダクト内では、パスは一意でなければなりません。つまり、1つのプロダクト内では、同じパスを使用して2つのバックエンドを追加することはできません。また、1つのプロダクトに、同じバックエンドを2回追加することもできません。
- プロダクトが異なっていれば、1つのバックエンドに同じパスを指定することができます。

バックエンドパスが機能する仕組みを以下に示します。

- プロダクトに追加されたそれぞれのバックエンドは、指定したパスにマウントされます。
- トラフィックをリダイレクトする前に、プロダクトに対するリクエストの公開 URL からパスが削除されます。

例

プロダクトにバックエンドを追加する以下の構成について考えます。

- バックエンド: Inventory
- リソースパス: /list
- プロダクト: Petstore
- バックエンドパス: /supplies

この構成で使用される URL は以下のとおりです。

- 公開 URL: <public-api-domain>/supplies/list
- プライベート URL: <private-api-domain>/list

リクエストが送信された際のアクションのフローを以下に示します。

1. アプリケーションがリクエストを送信する。
2. リクエストが公開 URL <public-api-domain>/supplies/list に送信される。
3. プライベート URL <private-api-domain>/list にリダイレクトする前に、バックエンドパスが削除される。
4. リクエストが Inventory バックエンドにリダイレクトされる。

1.3.4. マッピングルールの定義

本セクションでは、バックエンドレベルでのマッピングルールの定義について説明します。マッピングルールは、バックエンドレベルおよびプロダクトレベルで定義することができます。バックエンドレベルでマッピングルールを定義することのメリットは、任意のプロダクトに追加することのできる再使用可能なバックエンドを作成できることです。(プロダクトレベルおよびバックエンドレベルの両方で) API に対するリクエストに応じてレポートするメトリクスまたはメソッドの詳細については、『API ゲートウェイの管理』の「[マッピングルール](#)」を参照してください。

前提条件

- バックエンド。バックエンドの作成については、「[新規バックエンドの作成](#)」を参照してください。

手順

1. Dashboard から **Backends** タブを選択します。
2. 設定を行うバックエンドの名前をクリックします。たとえば、**API Backend** をクリックします。
3. バックエンドについての詳細が表示されるページで、**Mapping Rules** に移動します。
4. **Add Mapping Rule** をクリックします。
5. 以下の設定を指定します。
 - **Verb**: HTTP リクエストの動詞 (GET、POST、DELETE、または PUT)

- **Pattern:** 照合するパターン。たとえば、`/hello` 等。
- **Metric or method to increment** メトリクスまたはメソッドの名前
- **Increment by:** メトリクスのカウントを増やす数。たとえば、`1` 等。
- **Last?:** このマッピングルールを最後のルールとして、他のマッピングルールの処理を停止するかどうかを定義します。
- **Position:** マッピングルール実行の順番を表す数字。マッピングルールの並べ替えに使用します。

6. Create Mapping Rule をクリックします。

これらの手順を完了すると、バックエンドにマッピングルールが追加されます。このマッピングルールは、現在そのバックエンドを使用している各プロダクトでも利用することができます。マッピングルールをプロダクトレベルでアクティブにするには、プロダクトの **Integration** ページで最新のプロキシ設定をプロモートします。

最新のプロキシ設定をプロモートすると、3scale はバックエンドに設定されたマッピングルールをプロダクトレベルでアクティブ化します。マッピングルールは、プロダクトで指定したバックエンドパスに続きます。たとえば、以下に示す設定の場合には、

- バックエンドでのマッピングルールのパターン: `/thousands`
- バックエンドをプロダクトに追加する際のパス: `/unitprice`

最終的なプロダクトレベルでのマッピングルールは、`/unitprice/thousands` となります。

1.3.5. アプリケーションプランの作成

本セクションでは、API に基本的なアプリケーションプランを作成する方法について説明します。アプリケーションプランにより、API の使用に対するルール (制限、課金、および機能) をプロダクトレベルで定義します。詳細については、『管理ポータルガイド』の「[アプリケーションプラン](#)」および「[API の定義 \(メソッドおよびメトリクス\)](#)」を参照してください。

前提条件

- プロダクト。プロダクトの作成については、「[新規プロダクトの作成](#)」を参照してください。

手順

テスト用に新たなアプリケーションプランを作成するには、以下の手順に従います。

1. `[Your_product_name]` > **Applications** > **Application Plans**の順に移動します。
2. **Create Application Plan** をクリックします。
3. **Create Application Plan** のページで、新しいプランの名前およびシステム名を入力します。
 - システム名は一意でなければなりません。
4. **Create Application Plan** をクリックします。

1.3.6. アプリケーションの作成

本セクションでは、デフォルトの **Developer** アカウントに基本的なアプリケーションを作成する手順の概要を説明します。

アプリケーションはアプリケーションプランを参照します。アプリケーションは、必ずアプリケーションプランに関連付けられます。アプリケーションは、開発者アカウント内に保管されます。3scale の Basic プランで許可されるアプリケーションは1つだけですが、Enterprise プランではアカウントごとに複数のアプリケーションが許可されます。

前提条件

- アプリケーションプラン。アプリケーションプランの作成については、「[アプリケーションプランの作成](#)」を参照してください。

手順

1. **Dashboard > Audience > Account**の順に移動します。
2. **Developer** というデフォルトアカウントをクリックします。
3. **Application** タブに移動します。
4. **Create Application** をクリックします。
5. アプリケーションプランを選択します。
6. 名前および説明を指定します。
7. **Create Application** をクリックします。

1.3.7. プロダクトを含めたバックエンドのテスト

本セクションでは、バックエンドがどのようにプロダクトに接続されるか、およびその接続の確認方法について説明します。プロダクトを含めたバックエンドのテストでは、リクエストコールを元にテストを行うために、APIcast 設定をステージング環境および実稼働環境にプロモートする必要があります。

プロダクトに対するリクエストは、パスに従って対応するバックエンドにリダイレクトされます。このパスは、バックエンドをプロダクトに追加する際に設定されます。たとえば、プロダクトに2つのバックエンドを追加している場合、それぞれのバックエンドは固有のパスを持ちます。

前提条件

- プロダクトに追加された1つまたは複数の [バックエンド](#)
- プロダクトに追加された各バックエンドの [マッピングルール](#)
- アクセスポリシーを定義するための [アプリケーションプラン](#)
- アプリケーションプランを参照する [アプリケーション](#)

手順

1. **[Your_product_name] > Integration > Configuration**の順に移動して、APIcast 設定をステージング環境にプロモートします。
2. **APIcast Configuration** セクションに、プロダクトに追加された各バックエンドのマッピングルールが表示されます。**Promote v.[n] to Staging APIcast**をクリックします。

- `v.[n]` は、プロモートされるバージョン番号を表します。
3. ステージング環境にプロモートしたら、実稼働環境にプロモートすることができます。**Staging APIcast** セクションで、**Promote v.[n] to Production APIcast** をクリックします。
 - `v.[n]` は、プロモートされるバージョン番号を表します。
 4. コマンドラインで API へのリクエストをテストするには、**Example curl for testing** で提供されるコマンドを使用します。
 - `curl` コマンドの例は、プロダクトの最初のマッピングルールに基づいています。

API へのリクエストをテストする際に、[メソッドおよびメトリクスを追加して](#) マッピングルールを変更することができます。

設定を変更したら、API への呼び出しを行う前に、必ずステージング環境および実稼働環境にプロモートするようにしてください。ステージング環境にプロモートする保留中の変更がある場合には、管理ポータルの **Integration** メニュー項目の横に感嘆符が表示されます。

1.3.8. その他の参考情報

プロダクトおよびバックエンドについての詳細は、以下のアートを参照してください。

- [APIs as a Product: Get the value out of your APIs](#)
- [APIs as a Product: Get started in no time](#)

第2章 API の公開

本章では、Red Hat 3scale API Management を使用して API を公開するための主要な手順について説明します。本章で説明する手順は、以下の状況を前提としています。

- 3scale に内部 API **API Backend** ([backend](#) を参照) が設定されている。
- 3scale に顧客がアクセスする API **Echo API** ([product](#) を参照) が設定されている。これが、デベロッパーポータルを通じて公開される API です。

本章では、API プロダクトを公開するための以下の手順を説明します。

1. プロダクトを保護する。
2. アプリケーションプランを使用してプロダクトへのアクセスポリシーを設定する。
3. デベロッパーポータルを通じて開発者を参加させる。
4. 実稼働環境に移行する。

2.1. 最初の API を公開するためのパス

3scale の操作を開始するにあたり、API プロダクトを一般に公開する 3 つのパスのいずれかを選択することができます。**API プロダクト** とは、一般に公開され顧客がアクセスする API です。

期間のガイドラインは、API プロダクトの複雑さや作業に割くことのできるリソースにより異なります。ほとんどの時間は、API プロダクトの調整およびデベロッパーポータルコンテンツの準備に費やされます。安定したプロダクトおよびドキュメントのコンテンツがすでに用意されている場合には、1 週間で実稼働環境に移行することができます。

最初の API プロダクトを公開するためのパスを以下に示します。

プロトタイプ

- 目的: 3scale と一般に公開されるシンプルな API プロダクトのインテグレーションを、エンドツーエンドで完了する。
- 対象: このパスは、3scale のエンドツーエンドの機能を大まかに理解するのに役立ちます。ベーシックの手順を実施する前に、このパスを実施する必要があります。管理ポータルの導入ウィザードを正常に完了している場合には、このパスを省略して次のパスに進むことができます。
- 所要時間: 1 時間以内

ベーシック

- 目的: API プロダクトを実稼働環境で公開するための操作手順をすべて完了する。
- 対象: API プロダクトを実稼働環境で公開したいが時間が限られている場合には、このベーシックパスにより必要な操作のほとんどがカバーされます。
- 所要時間: 1 週間以内

アドバンスド

- 目的: ベーシックパスを完了した後のオプション操作。API プロダクトの高度なコントロールおよびデベロッパーポータル複雑なカスタマイズが可能。
- 対象: 要求がより複雑である、またはベーシックパスをすでに完了している場合には、アドバンストのオプションの検討が必要な場合があります。
- 所要時間: 数週間

2.2. プロトタイプパスの手順

プロトタイプパスの手順を、個別にエンドツーエンドで実施することができます。あるいは、必要に応じてこのパスの手順の一部を抜き出して実施することもできます。それぞれのパスは互いに独立していますが、**プロトタイプ**、**ベーシック**、および **アドバンスト** パスは、順に積み上げて高度な設定を行うことが可能です。

2.2.1. API の保護

以下の状況のいずれかであれば、数分でプロトタイプの 3scale アクセス制御レイヤーを設定することができます。

- ホスト型 3scale (SaaS) において、プロダクトが一般に公開されている。
- オンプレミス型 3scale において、プロダクトが 3scale インストール環境からアクセス可能である。

ここでは、公開されたプロダクトの例として **Echo API** を使用します。このプロダクトの機能は以下のとおりです。

- これは、任意のパスを受け入れてレスポンスのボディでリクエストに関する情報 (パス、リクエストパラメーター、ヘッダー等) を返すシンプルな API です。
- Echo API には <https://echo-api.3scale.net> の URL からアクセスすることができます。
- 初めて 3scale をアクティブ化すると、既存の API ごとにプロダクトが作成されます。この初回アクティベート時には、プロダクトと API バックエンドの間に 1対1の関係が存在します。つまり、**API Backend** が含まれる **Echo API** プロダクトが作成されます。

Echo API プロダクトを保護するには、以下の手順に従います。

1. プロダクトにアクセス可能であることを確認します。たとえば、<https://echo-api.3scale.net/v1/fast/track>。
 - セキュリティーレイヤー実装後は、バックエンドホストを隠したりアクセスを制限したりすることができます。
2. **[Your_product_name] > Integration > Configuration**の順に移動します。
3. **[Your_product_name]** について、プライベートエンドポイントがデフォルトの API Backend に設定されていることを確認します。たとえば、<https://echo-api.3scale.net:443>。
4. ボタンをクリックして、ステージング環境にプロモートします。
5. デフォルトクレデンシャルとして **user_key** が含まれる cURL ステートメントをコピーして、コマンドラインから呼び出しを行います。


```
curl "https://api-2445581407825.staging.apicast.io:443/v1/fast/track?
user_key=287d64924e6120d215b1000ac07c063b"
```

異なる呼び出しを行うことができます。たとえば、同じ **user_key** を追加して、別のエンドポイントに対する呼び出しを行います。



注記

いずれかの開発者アカウントにあるアプリケーションの詳細ページから、API プロダクトキーを取得することができます。

3scale のアクセス制御レイヤーにより、認証された呼び出しだけがご自分のバックエンド API にアクセスできるようになります。

2.2.2. アプリケーションプランによる API アクセスポリシーの設定

「**API の保護**」の手順を実施すると、認証された呼び出しだけがご自分の API にアクセスできるようになります。本セクションでは、ポリシーを適用して流量制御を変更します。

3scale では、**アプリケーション** によりプロダクトにアクセスするためのクレデンシャルが定義されます。アプリケーションには必ず1つ **アプリケーションプラン** が関連付けられ、アクセスポリシーが定義されます。アプリケーションは、**開発者アカウント** 内に保管されます。3scale の Basic プランで許可されるアプリケーションは1つだけですが、より高度なプランではアカウントごとに複数のアプリケーションが許可されます。

この例では、前項で使用した **Echo API** プロダクトにポリシーを追加します。その手順を以下に示します。

1. **[Your_product_name] > Applications > Application Plans**の順に移動します。
2. **Application plans** セクションで **Basic** アプリケーションプランに進み、3scale のインストールまたはサインアップ後にサンプルデータから生成されたプランのいずれかを編集します。
3. **Metrics, Methods, Limits & Pricing Rules**セクションで、**Hits** 行の **Limits** を選択し、1時間あたり3回の新しい使用限度を作成します。
4. **[Your_product_name] > Applications > Listing**の順に移動し、サンプルアプリケーションを1つ選択します。アプリケーションが **Basic** プランに設定されていることを確認します。設定されていない場合は、アプリケーションの詳細ページで **プランを変更**します。
5. このアプリケーションのクレデンシャルを使用して、上記のサンプルコールを少なくとも3回繰り返します。

これで、3scale Basic プランのすべてのアプリケーションに対して、より制限の厳しいアクセスポリシーが正しく定義されました。

2.2.3. デベロッパーポータルを通じた開発者の参加

プロトタイプパスでは、何らかのドキュメントコンテンツを作成する必要はありません。通常は、ワークフローが要求に適合していることを確認するだけで十分です。

プロダクトが開発/テスト段階の間は、以下の手順により完全なセルフサービスワークフローを無効にすることができます。

1. 管理ポータルから **Audience** セクションに移動し、**Developer Portal** メニューの **Visit Portal** のリンクをクリックします。
2. テストサインアップを作成し、すべての手順を実施します。
3. 通常、セルフサービスはデフォルトで有効になっています。この設定を変更するには、**Audience > Accounts > Usage Rules**の順に移動し、**Account approval required** のチェックボックスを選択します。
4. テストサインアップの手順をすべて繰り返し、ユーザーがログインするには管理ポータルでアカウントを承認しなければならないことを確認します。

これで、デベロッパーポータルのワークフローをカスタマイズできるようになりました。

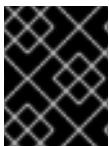
2.3. ベーシックパスの手順

ベーシックパスの手順を、個別にエンドツーエンドで実施することができます。あるいは、必要に応じてこのパスの手順の一部を抜き出して実施することもできます。それぞれのパスは互いに独立していますが、**プロトタイプ**、**ベーシック**、および **アドバンスド** パスは、順に積み上げて高度な設定を行うことが可能です。

2.3.1. API の保護

実稼働環境向けの完全な実装のためには、プロダクトの構成および 3scale とのインテグレーション方法に関して、基本方針を決定する必要があります。

プロダクトトラフィックの認証モードには、いくつかの選択肢があります。[利用可能なオプションに関するガイド](#)を確認し、設定を行ってください。



重要

認証の設定を行ったら、認証モードを切り替えないでください。この操作により、既存のクレデンシャルが無効になる可能性があるためです。

API トラフィックマネージャーレイヤーのデプロイメントにも、いくつかのオプションがあります。設定の容易さとパフォーマンスのバランスから、3scale ユーザーの間では APIcast (NGINX ベースの API ゲートウェイ) が好まれています。Hosted APIcast を使用すればすぐに運用を開始することができますが、ボリュームの制約やレイテンシー増加の問題が生じます。一方、APIcast を専用のサーバーにデプロイすれば、最高のパフォーマンスが得られ、トラフィックボリュームの制約から完全に解放されます。

Hosted APIcast

1. 管理ポータルへの初回ログイン後に、導入ウィザードの手順に従います。
2. 実稼働環境に適したバージョンが得られるまで、アクセスポリシーの調整などのプロダクト設定を繰り返します。
3. APIcast 設定を実稼働環境用ゲートウェイにプロモートします。

Self-managed APIcast

1. [OpenShift](#) サーバー上に、API ゲートウェイのテスト用システムを設定します。

2. 実稼働環境に適したバージョンが得られるまで、API 設定を繰り返します (アクセスポリシーの調整など)。
3. APIcast 設定を実稼働環境用ゲートウェイにプロモートします。
4. Self-managed APIcast の詳細な情報は、『3scale のインストール』の「[APIcast のインストール](#)」を参照してください。また、『API ゲートウェイの管理』の「[APIcast ポリシー](#)」には、API のアクセスポリシーを設定する際の概念が記載されています。

2.3.2. アプリケーションプランによる API アクセスポリシーの設定

「[API の保護](#)」の手順を実施すると、認証された呼び出しだけがご自分のプロダクトにアクセスできるようになります。本セクションでは、ポリシーを適用して流量制御を変更します。

3scale では、**アプリケーション** により API プロダクトにアクセスするためのクレデンシャルが定義されます。アプリケーションには必ず1つ **アプリケーションプラン** が関連付けられ、アクセスポリシーが定義されます。アプリケーションは、**開発者アカウント** 内に保管されます。3scale の Basic プランでは、1つのアプリケーションしか許可されません。より高度なプランでは、アカウントごとに複数のアプリケーションが許可されます。

プロトタイプ では、プロダクトの総ヒット数に基づいてアクセスを制御することしかできません。カスタムメソッドおよびメトリクスを使用して初めて、3scale の柔軟性を体感することができます。カスタムメソッドおよびメトリクスを使用することで、アプリケーションプランおよびプロダクトの解析をより高度化することができます。詳細は、[解析に関するガイド](#) を参照してください。



重要

- API の構造と 3scale のメソッドまたはメトリクスとの間のマッピングは論理的です。整合性のとれたルールを定義すれば、3scale からプロダクトの使用状況に関するレポートを受け取ることができます。詳細さのレベルを決める必要があります。一般的には、5 - 20 メソッド/メトリクスを目標とする必要があります。
- 3scale に報告される値は、増加するだけです。絶対値を設定したりカウンターを減らしたりすることはできません。
- 新たなメソッドまたはメトリクスを 3scale に追加したら、API ゲートウェイまたはコードプラグイン等のインテグレーションポイントに新しいシステム名を追加することが重要です。
- 実行時に、再デプロイせずに流量制御などに変更を加えることができます。

Echo API プロダクトのアプリケーションプランにポリシーを追加するには、以下の例に示す手順を実施します。

1. 操作対象のプロダクトを選択します。
2. **Application Plans** セクションで **Basic** を選択し、3scale へのサインアップまたはインスタンスのデプロイ後に自動的に生成されたプランのいずれかを編集します。
3. **Hits** に流量制御が設定されている場合には、その設定を削除します。
4. プランの **Hits** メトリックに **新規メソッド** を追加し、システム名を「test」に設定します。
5. test メソッドの流量制御を1時間あたり5回に設定します。
6. 2つの **新規メトリクス** を追加し、システム名を **v1** および **v2** に設定します。

- v2 メトリックの **Enabled** 列をクリックし、アクセスを無効にします。これは、流量制御をゼロに設定するのと同じ効果を持ちます。

APIcast のデプロイメント

- [Your_product_name] > Integration > Configuration の順に移動します。
- MAPPING RULES セクションを展開し、以下のマッピングを追加します。

Verb	Pattern		Metric or Method	Last?	Position	
GET	/[version]/test	1	hits	false	1	
GET	/v1	1	hits	false	2	
GET	/v2	1	hits	false	3	



注記

「/」のデフォルトのマッピングは削除されています。このデフォルトマッピングを使用すると、ヒット数が二重にカウントされます。

コードプラグインのデプロイメント

- 3scale の承認およびレポート呼び出しにカスタムメソッドおよびメトリクスの使用状況を追加するには、プラグインライブラリーの手順および例に従ってください。
- URL 構造からカスタムメソッド **test** へのマッピングを確認します。
- URL からカスタムメトリクス **v1** および **v2** へのマッピングを確認します。
- 3scale の Basic プランに関連付けられたアプリケーションのクレデンシャルを使用して、呼び出しをテストします。

- 呼び出しは許可されます。

```
curl "https://api-2445581407825.staging.apicast.io:443/v1/test?
user_key=287d64924e6120d215b1000ac07c063b"
```

5 回目以降は、呼び出しが拒否されます。これは、test メソッドに設定した制限のためです。

- 3scale の Basic プランでは **v2** は許可されないため、呼び出しは拒否されます。

```
curl "https://api-2445581407825.staging.apicast.io:443/v2/test?
user_key=287d64924e6120d215b1000ac07c063b"
```

- missing に設定されたマッピングルールがないため、呼び出しは拒否されます。

```
curl "https://api-2445581407825.staging.apicast.io:443/missing?
user_key=287d64924e6120d215b1000ac07c063b"
```

- プラグインのマッピングの実装状態によっては、これらの呼び出しは NGINX に対して許可されます。以下の呼び出しで「404 not found」のレスポンスが返されるかどうかは、お使いのアプリケーションによります。これを避けるには、マッピングを調整してください。

```
curl "https://api-2445581407825.staging.apicast.io:443/noversion/test?
user_key=287d64924e6120d215b1000ac07c063b"
```

この基本概念により、プロダクトレベルおよびバックエンドレベルの両方で、柔軟に API レイヤーを定義することができます。カスタムメソッドおよびメトリクスに何をを使うかについて、早期に決定しておく必要があります。システム名に変更を加えた場合には、ベーシックパスの「[API の保護](#)」で説明したように、必ず変更を再デプロイします。

2.3.3. デベロッパーポータルを通じた開発者の参加

デベロッパーポータルの [作成](#) およびデベロッパーポータルを使用した [API の提供](#) については、関連のドキュメントを参照してください。コンテンツを Textile または Markdown で記述することを検討してください。オプションとして考慮すべき項目を以下に示します。

- ドキュメントにインタラクティブ機能を追加して開発者のドキュメント作業を容易にするために、[ActiveDocs](#) を設定する。
- ファビコンを追加する。
- CMS で **partial** セクションの **analytics** を編集して、Google Analytics トラッカーコードを追加する。
- [サインアップのワークフロー](#)を設定する。
- [メールアドレス](#) および [メールテンプレートのコンテンツ](#) をカスタマイズする (共にホスト型 3scale (SaaS) のドキュメントを参照)。

2.4. アドバンストパスの手順

アドバンストパスの手順を、個別にエンドツーエンドで実施することができます。あるいは、必要に応じてこのパスの手順の一部を抜き出して実施することもできます。それぞれのパスは互いに独立していますが、[プロトタイプ](#)、[ベーシック](#)、および [アドバンスト](#) パスは、順に積み上げて高度な設定を行うことが可能です。

2.4.1. API の保護

プロダクトを保護するには、以下のオプションがあります。

高度な認証モード: OpenID Connect (OIDC)

[OpenID Connect とのインテグレーション](#) を通じた APIcast と Red Hat Single Sign-On (RH-SSO) の組み合わせにより、プロダクトを保護します。3scale のアプリケーションは、アイデンティティプロバイダー (IdP) (この場合は RH-SSO) と同期されます。現在、これはエンドツーエンドでサポートされるソリューションです。以下に示すメインの OAuth 2.0 フローに対応します。

- 認可コード
- リソースオーナーパスワード
- クライアントクレデンシャル
- インプリシット付与

コードプラグインのデプロイメント

プロダクトのパフォーマンスを向上させたい場合には、何らかのキャッシングライブラリーを使用して 3scale への承認呼び出しをキャッシュすることができます。

2.4.2. アプリケーションプランによる API アクセスポリシーの設定

「[API の保護](#)」に記載のオプションを使用すると、認証された呼び出しだけがご自分のプロダクトにアクセスできるようになります。本セクションでは、ポリシーを適用して流量制御を変更します。

3scale では、**アプリケーション** によりプロダクトにアクセスするためのクレデンシャルが定義されます。アプリケーションには必ず1つ **アプリケーションプラン** が関連付けられ、アクセスポリシーが定義されます。アプリケーションは、**開発者アカウント** 内に保管されます。3scale の Basic プランでは、1つのアプリケーションしか許可されません。より高度なプランでは、アカウントごとに複数のアプリケーションが許可されます。

アラートを設定して、通知を電子メールで送付したり Web コンソールに表示したりすることができます。

1. API Settings のページに移動します (`[Your_product_name]` > `Integration` > `Settings`)。
2. ページの **ALERTS** セクションに移動します。ここで、必要なアラートを流量制御値のパーセンテージで設定することができます。

3scale では、流量制御のレベルを柔軟に設定することができます。

- ソフト流量制御: 上限を超えた呼び出しでも許容される
- ハード流量制御: アプリケーションにアクセスする前に呼び出しが拒否される

コードプラグインでは、実装するタイプを決める必要があります。一方 APIcast の場合は、デフォルトではハード制御が定義されます。上限を超えた呼び出しが拒否されるのを防ぐために、Lua ファイルでこれらの制御をカスタマイズすることができます。

2.4.3. デベロッパーポータルを通じた開発者の参加

[ベーシック](#) パスの設定を完了している場合、以下の2つの項目についてデベロッパーポータルの高度な使用方法を検討することができます。

- [Liquid マークアップ](#) のタグおよびドロップにより、システムオブジェクトに直接アクセスすることができ、デベロッパーポータルページのダイナミックレンダリングが可能になります。
- 3scale のシステムページはすべてカスタマイズが可能です。ただし、HTML が複雑になるため、カスタマイズは熟達したユーザー向けです。極端に言えば、デベロッパーポータルのほとんどすべてのページをカスタマイズすることができます。しかしながら、一部の CSS を変更すれば、通常はデフォルトのページでまったく問題ありません。

2.5. 実稼働環境への移行

本セクションでは、API プロダクトを一般に公開する前の最終チェックリストについて説明します。



注記

カスタムドメインおよびメールアドレスのリードタイムは長いので、できるだけ早く要求します。

1. カスタムドメインを準備する。詳細な情報は、ホスト型 3scale (SaaS) ドキュメントの「[カスタムドメインの設定](#)」を参照してください。
2. オプションとして、発信用のカスタムメールアドレスを準備する。この手順の実施方法は、「[メールドメインの設定](#)」を参照してください。
3. **Audience > Developer Portal > Domains & Access**からデベロッパーポータルへのアクセスコードを削除する。

追加の検討事項を以下に示します。

- 課金ルールを追加して、API プロダクトから直接収益を得る。この機能は、ホスト型 3scale (SaaS) アカウントでのみ利用可能です。
- プロダクト解析 (管理ポータル内の **Analytics** セクション) からの知見を活用して、アプリケーションプランを調整する。