



Red Hat 3Scale 2-saas

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 2-saas スタートガイド

3scale API Management システムのスタートガイド

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Getting_Started.ent file |.

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

多様性を受け入れるオープンソースの強化	3
第1章 3SCALE の基礎	4
1.1. 3SCALE API プロダクトおよびバックエンド	4
1.2. 3SCALE ウィザードを使用した最初の API の設定	4
1.3. テストコールを実施するための初期 API 設定	5
1.4. プロダクトに追加するバックエンドの作成	5
1.5. API コールをテストするための新規プロダクトの作成	6
1.6. 既存プロダクトへのバックエンドの追加	7
1.7. 特定プロダクトのバックエンドパス	7
1.8. マッピングルールの定義	8
1.9. プロダクト用 3SCALE アプリケーションプランの作成	9
1.10. API コールをテストするためのデフォルトアカウントへのアプリケーションの作成	10
1.11. バックエンドのインテグレーションをテストするためのプロダクトへのリクエスト送信	11
1.12. 関連情報	12
第2章 API の公開	13
2.1. 最初の API を公開するためのパス	13
2.2. プロトタイプパスの手順	14
2.2.1. API の保護	14
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	15
2.2.3. デベロッパーポータルを通じた開発者の参加	15
2.3. ベーシックパスの手順	16
2.3.1. API の保護	16
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	17
2.3.3. デベロッパーポータルを通じた開発者の参加	19
2.4. アドバンスドパスの手順	19
2.4.1. API の保護	19
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	20
2.4.3. デベロッパーポータルを通じた開発者の参加	20
2.5. 実稼働環境への移行	20

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 3SCALE の基礎

Red Hat 3scale の機能を初めて使用するユーザーに対して、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を説明します。

1.1. 3SCALE API プロダクトおよびバックエンド

Red Hat 3scale では、API は 2 つのグループに大別されます。

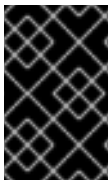
- **バックエンド**: プロダクトにバンドルされている内部 API。バックエンドにより、API プロバイダーは内部 API の組織構造を自由に 3scale にマッピングすることができます。
- **プロダクト**: 顧客向けの API。プロダクトにより、API の利用者に対して強固でシンプルなオファリングを容易に作成することができます。

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。再使用性を高めるために、オプションとして、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設定することができます。
- 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

3scale ウィザードを使用すると、プロダクトおよびバックエンドに関する初期操作を容易に行うことができます。



重要

3scale アカウントに初めてログインする場合にのみ、3scale ウィザードを実行します。ウィザードを実行してさらに 3scale コンポーネントを作成すると、既存のコンポーネントが上書きされます。

前提条件

- 3scale アカウントが必要である。

手順

1. ウィザードを実行します。3scale ウィザードは、2 通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実行する。
 - 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する。
`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。例: **Inventory**
 - b. バックエンドのベース URL を指定します。例: **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例: **Petstore**
 - 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. テストコールを実施するための初期 API 設定

初期設定により、API トラフィックが API キーで保護され、3scale に設定した基本的な流量制御および管理機能で追跡および監視されるようになります。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。

1.4. プロダクトに追加する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。

前提条件

- 3scale アカウントが必要である。

手順

1. Dashboard に移動します。
2. APIs セクションで、**Backends** カードの **Create Backend** をクリックします。
3. 以下の情報を指定します。
 - **名前**: バックエンド識別子。

- **システム名**:内部での処理に使用される ID。
- **説明**:バックエンドに関する詳細情報を記述する任意のフィールド。
- **プライベートベース URL**:プライベート API のベース URL エンドポイント。

4. **Create Backend** をクリックします。

プロダクトに追加するバックエンドの作成の手順を実施すると、内部 API が作成されます。必要なだけのバックエンドを作成することができます。

次のステップ

- [このバックエンドのプロダクトへの追加](#)

1.5. API コールをテストするための新規プロダクトの作成

3scale API プロバイダーとして、プロダクトを作成し、これらのパブリック API を使用して API コールをテストします。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。

前提条件

- 3scale アカウントが必要である。

手順

1. Dashboard に移動します。APIs セクションで、**Products** カードの **Create Product** をクリックします。
2. 以下の情報を指定します。
 - a. **名前**: プロダクト識別子
 - b. **システム名**:内部での処理に使用される ID。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**
 - 自動生成される URL ラベルが 63 文字を超えると、システムはラベルを **<truncated-label>-<unique-hash>** のように短縮します。
 - **<truncated-label>** は、元の URL の最初の 54 文字または 55 文字です。
 - **<unique-hash>** は、元のラベルから計算された一意の SHA-1 ハッシュの最初の 7 文字です。
たとえば、省略される前の URL が以下であった場合、

https://my-very-long-system-name-also-very-long-tenant-name-apicast-staging.3scale.net

省略後の URL は以下ようになります。

https://my-very-long-system-name-also-very-long-tenant-name-api-72588d2.3scale.net

c. **説明:** プロダクトに関する詳細情報を記述する任意のフィールド。

3. **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、[バックエンドの作成](#) と [プロダクトへの追加](#) です。

1.6. 既存プロダクトへのバックエンドの追加

この手順を完了すると、プロダクトにバックエンドを追加することができます。手順を繰り返して、必要なだけのバックエンドを追加します。

前提条件

- プロダクト。プロダクトの作成については、[API コールをテストするための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、[プロダクトに追加するバックエンドの作成](#) を参照してください。

手順

1. **[Your_product_name] > 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、以下のアクションの1つを実行します。
 - 既存のバックエンドを選択します。
 - **View all** をクリックして、一覧全体を表示します。
 - 新規バックエンドを作成する必要がある場合は、**Create new Backend** をクリックして詳細を入力します。
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。詳細は、[特定プロダクトのバックエンドパス](#) 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1つのバックエンドが追加されます。この手順を繰り返して、1つまたは複数のプロダクトにさらにバックエンドを追加することができます。

1.7. 特定プロダクトのバックエンドパス

バックエンドをプロダクトに追加する場合、バックエンドが特定のプロダクトと通信するのに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。

APIcast は、[既存プロダクトへのバックエンドの追加](#) に記載する手順のステップ 4 で指定したバックエンドのパスを使用する API ゲートウェイです。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設定について考えます。

- バックエンド:インベントリー
- リソースパス: **/list**
- プロダクト:Petstore
- バックエンドパス: **/supplies**

この設定で使用される 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。

1.8. マッピングルールの定義

API へのアクセスをトラッキングおよび制限するために、マッピングルールによりエンドポイントへの呼び出しを定義したメソッドおよびメトリクスに関連付けます。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、複数のプロダクトにバックエンドを追加できることです。(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基づいて使用状況に関する情報を収集するメトリクスまたはメソッドについての詳細は、[3scale API の使用状況を把握するために APIcast がマッピングルールをどのように適用するか](#) 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、[プロダクトに追加するバックエンドの作成](#) を参照してください。

手順

1. Dashboard で、マッピングルールを定義する **Backend** をクリックします。
2. ナビゲーションパネルで **Mapping Rules** をクリックします。
3. **Create mapping rule** をクリックします。
4. 以下の設定を指定します。
 - **動詞**:HTTP リクエストの動詞 (**GET**、**POST**、**DELETE**、または **PUT**)。
 - **パターン**:照合するパターン。例: **/hello** など。
 - **インクリメントするメトリックまたはメソッド**:メトリクスまたはメソッド名。
 - **次によるインクリメント**:メトリクスのカウントを増やす数。例: **1** など。
 - **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。
 - **ポジション**: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。
5. **Create mapping rule** をクリックします。

次のステップ

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 (**[Your_API_backend]** > **Mapping Rules**)。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、**[Your_product_name]** > **Integration** > **Configuration** の順に移動し、**Products** タブで最新の設定をプロモートします。

例

設定をプロモートすると、3scale はバックエンドの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、以下のような設定と仮定すると、

- 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**

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。

1.9. プロダクト用 3SCALE アプリケーションプランの作成

3scale のアプリケーションプランにより、API プロダクトを使用する上でのルール (上限など)、課金、および機能を定義します。詳細は、[アプリケーションプラン](#) および [使用状況の詳細を把握するためのメソッドの指定およびメトリクスの追加](#) 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、[API コールをテストするための新規プロダクトの作成](#) を参照してください。

手順

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 3scale インストール環境内で一意でなければなりません。
4. Create Application Plan をクリックします。

1.10. API コールをテストするためのデフォルトアカウントへのアプリケーションの作成

3scale ユーザーとして、デフォルトの **Developer** アカウントにアプリケーションを作成します。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。この呼び出しの結果、3scale は API プロダクトを呼び出すのに必要なユーザーキーを提供します。

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 のベーシックプランでは、1つのアプリケーションしか許可されません。エンタープライズプランでは、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

- 3scale API プロダクト。作成するには、[API コールをテストするための新規プロダクトの作成](#) を参照してください。
- 3scale の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、[プロダクト用 3scale 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. Audience > Accounts > Listingの順に移動します。
2. デフォルトの **Developer** アカウントをクリックします。
3. Application タブをクリックします。
4. Create application をクリックします。
5. Create application ダイアログで、アプリケーションの製品を選択します。
6. アプリケーションプランを選択します。
7. アプリケーション名を指定します。
8. Description フィールドに、このアプリケーションに関する情報を入力します。
9. Create application をクリックします。

Dashboard > Audience > Accounts > Applications > Listingの順に移動し、新しいアプリケーションを確認することができます。

1.11. バックエンドのインテグレーションをテストするためのプロダクトへのリクエスト送信

3scale API プロバイダーは、プロダクトにコマンドラインリクエストを送信して、プロダクトに追加した最初のマッピングルールに基づき、バックエンドのインテグレーションをテストすることができます。

テストリクエストを送信するには、テストするバックエンドが含まれる APIcast 設定をプロモートする必要があります。実際の APIcast 設定は、プロダクトに追加したバックエンド、ならびに対応するマッピングルール、アプリケーション、およびアプリケーションプランで設定されます。

3scale は、リクエストの呼び出しで指定したパスに従って、リクエストをプロダクトのバックエンドに送付します。バックエンドをプロダクトに追加する際に、プロダクトのバックエンドごとに **バックエンドパス** を設定します。言い換えると、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. 新しい APIcast 設定をステージング環境にプロモートします。
 - a. **[Your_product_name] > Integration > Configuration**の順に移動します。
 - b. **APIcast Configuration** セクションで、**Promote v.[n] to Staging APIcast** をクリックします。
 - **v.[n]** は、プロモート先のバージョン番号を表します。
 - プロモートする変更がなければ、**Nothing to promote** のテキストと共にボタンがグレーアウト表示されます。
2. **Staging APIcast** セクションで、**Promote v.[n] to Production APIcast** をクリックして APIcast 設定を実稼働環境にプロモートします。
 - **v.[n]** は、プロモート先のバージョン番号を表します。
 - プロモートする変更がなければ、**Nothing to promote** のテキストと共にボタンがグレーアウト表示されます。
3. API プロダクトへのリクエストをテストするには、**Example curl for testing** に記載のコマンドをコピーして、ターミナルで実行します。
 - **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。
 - コマンドを実行すると、テスト中のバックエンドからの結果が含まれる HTML レスポンスが返されるはずですが。

- レスポンスが返されない場合は、プロダクトからキャッチオールマッピングルールを削除し、新しい APIcast 設定をステージング、続いて実稼働環境にプロモートし、**curl** のサンプルコマンドを実行します。

次のステップ

上限や課金ルールなどのメトリクスおよびメソッドを変更すると、レスポンスが異なることを確認することができます。プロダクトのすべてのアプリケーションプランに関して、プロダクトへのリクエストをテストする際には、メソッドおよびメトリクスを変更します。詳細は、[使用状況の詳細を把握するためのメソッドの指定およびメトリクスの追加](#) を参照してください。

プロダクトの設定を変更したら、API への呼び出しを行う前に、更新した設定を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必要があります。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 **Integration** メニュー項目の横に感嘆符が表示されます。

1.12. 関連情報

- [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 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 **API Backend** という名前の内部 API ([backend](#) を参照) が設定されている。
- 3scale に **Echo API** という名前の顧客がアクセスする 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 時間未満。

Basic

- 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 週間未満。

詳細

- 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、ヘッダーなど) 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、以下が表示されます。**API Backend** が含まれるプロダクトの **Echo API**

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。(例: <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。

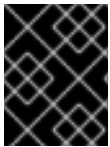
2.3. ベーシックパスの手順

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、プロダクトの設定および 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. この operator を使用して Self-managed APIcast ゲートウェイソリューションを [OpenShift](#) にデプロイします。

2. 実稼働環境に適したバージョンが得られるまで、アクセスポリシーの調整などの API 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。
4. Self-managed APIcast の詳細な情報は、[APIcast のインストール](#) を参照してください。また、API Gateway#apicast-policies[APIcast policies] では、API アクセスポリシーを設定する際の概念を説明します。

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など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。

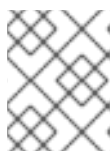
APIcast のデプロイメント

- [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。

Product: Echo API

Mapping Rules

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、[メールアドレスの設定](#) 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。