



DOCUMENT INFORMATIF

WSO2: Utilisation et fonctionnalités



MEHDI BENNIS

À PROPOS DE WS02

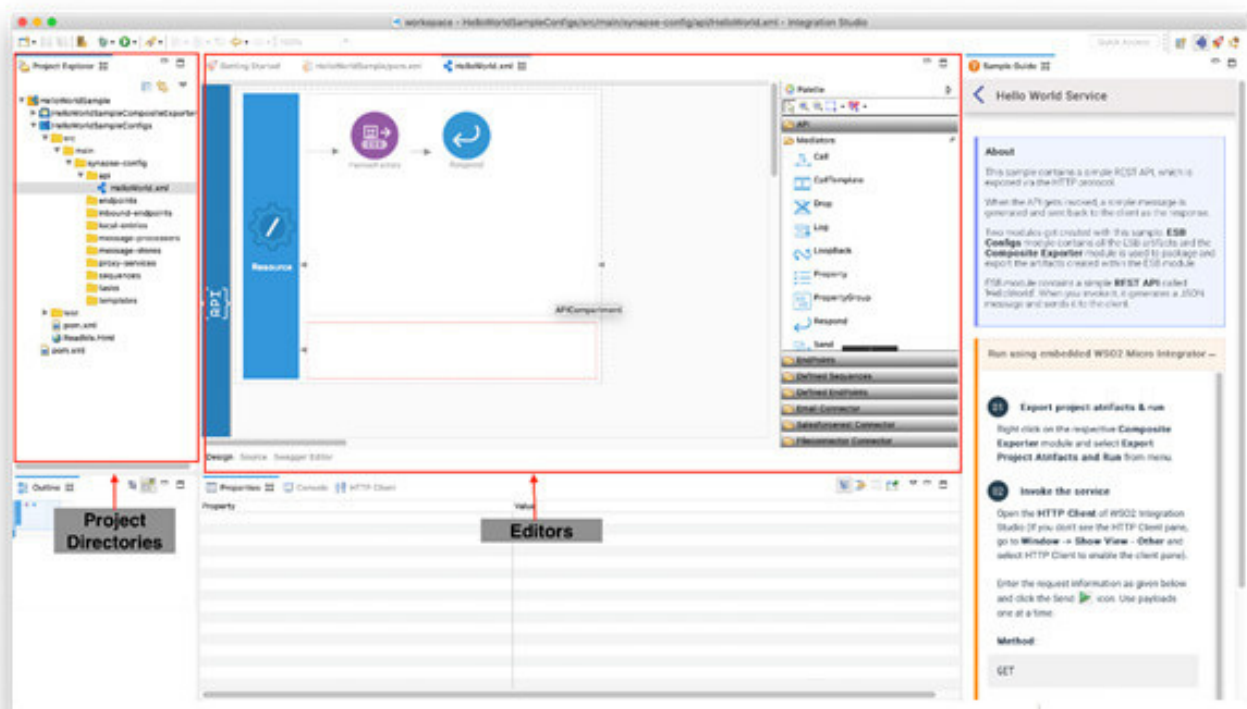
WSO2 est un fournisseur de technologie open source fondé en 2005. Il offre une plate-forme d'entreprise pour l'intégration d'interfaces de programmation d'applications, d'applications et de services Web localement et sur Internet.

TYPES DE PRODUITS DE WSO2

WSO2 Enterprise Integrator (EI) ou Micro Integrator (MI) est une plateforme d'intégration open source centrée sur les APIs. Il est l'outil API indispensable aux DSI qui ont une infrastructure de services Web sous protocoles.

Cette plateforme supporte les architectures de type micro-services ainsi que ESB. Cette solution est indépendante du Hardware, que vous soyez On Premise dans votre Datacenter, en cloud public, services Cloud ou Hybrid.

PRÉSENTATION ET CONCEPT DE WSO2 EI



L'objectif de WSO2 est de proposer une solution de Data Services étant au centre de votre SI, pouvant servir de proxy avec du message routing en fonction de différents paramètres configurables, de modification de message ainsi que de la transformation de format.

Cette solution vous permet de déployer vos développements d'APIs / de processus sous forme de package sans devoir passer par une solution annexe.

WSO2 fournit un Marketplace de connecteur vous simplifiant les interactions entre vos différentes solutions du SI, allant du simple fichier en passant par l'authentification LDAP jusqu'au service cloud comme AWS, Amazon web services, Google Cloud, Salesforce ou encore Service Now.

Au sein du Designer Studio, WSO2 fournit un environnement de développement basé sur Eclipse. Celui-ci vous offre plusieurs manières de développer vos web services / processus :

Via la version Low-code qui permet de glisser et déposer les composants dans des séquences ou des templates réutilisables par d'autres développements. Ainsi le développeur n'a pas besoin de connaître de manière pointue un langage de programmation pour l'implémentation de la solution.

Via un Mediator Script permettant d'exécuter différents langages de programmation tels que le javascript, le Ruby ou encore du Groovy

Via la création de Mediator en Java pour répondre à une spécification demandant plus de liberté de développement

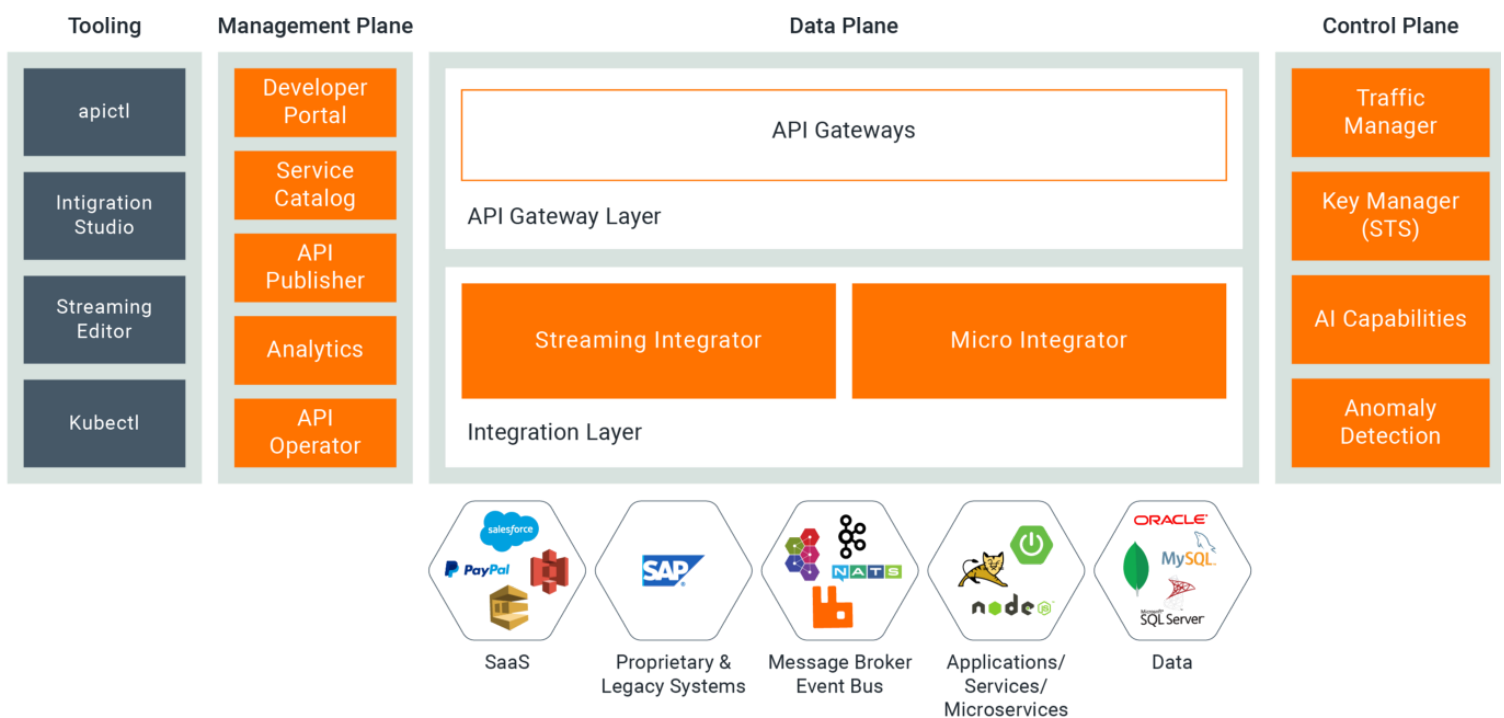
FONCTIONNALITÉS PRINCIPALES DE WSO2 EI



WSO2 propose un grand nombre de fonctionnalités allant de la génération d'APIs à la mise en place de tâches planifiées :

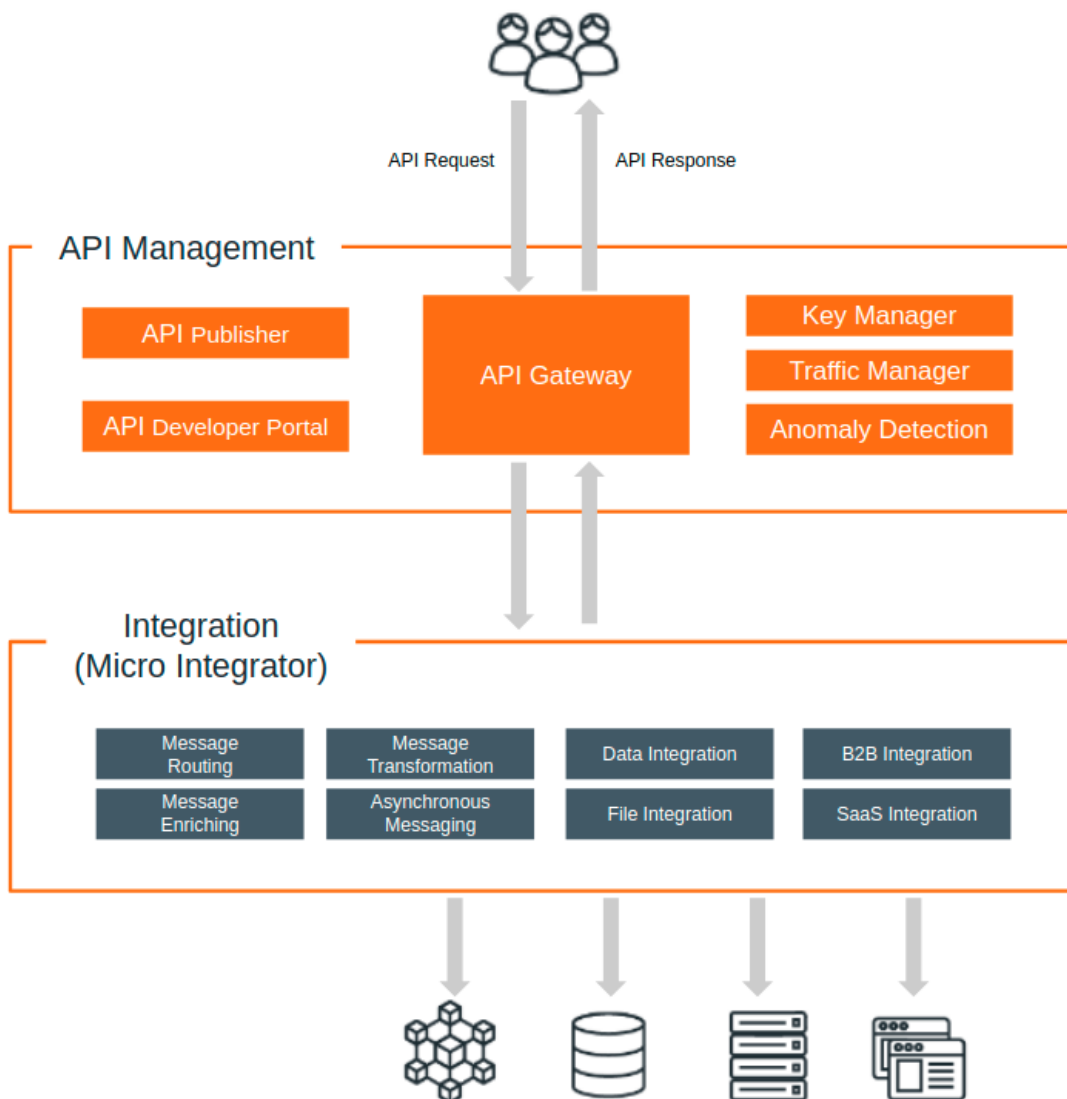
- *Création :*
 - *API (SOAP / REST) : Création d'une API à partir de zéro*
 - *Un Message Routing, comme un proxy basé sur des informations fournies dans le header ou le body d'une requête*
 - *Un Message Transforming (Réécriture de message, passage d'un format XML vers JSON)*
 - *Consommer des Message Queue comme RabbitMQ, JMS*
 - *Mise en place de templates réutilisables*
- *Intégration des données :*
 - *Mise en place facile d'un CRUD avec les connecteurs vers un grand nombre de bases de données (PostgreSQL, MySQL, MongoDB, Oracle...)*
 - *Exposition des données au format XML ou JSON*
 - *Utilisation des services OData*
- *Outils / Plateforme :*
 - *Un Designer Studio basé sur Eclipse*
 - *Déploiement / Hébergement*
 - *Une interface de management / monitoring des APIs*
 - *Un catalogue de composants*
 - *Registry, pour stocker des éléments de configuration*
 - *Un key vault pour stocker vos données sensibles dans un espace sécurisé et mutualisé entre vos APIs*
- *Installation :*
 - *A partir de l'installateur selon votre système d'exploitation*
 - *A partir des binaires*
 - *A partir des images docker fournies par WSO2 soit sur leur Registry soit sur Docker Hub avec la possibilité de mettre en place un kubernetes et gérer facilement la scalabilité et le load-balancing*

Architecture de WSO2



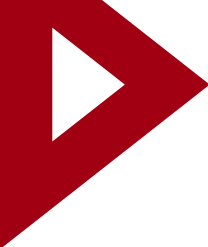
ABSORPTION DE WSO2 ENTERPRISE INTEGRATOR

Le premier changement majeur concerne le périmètre du produit. WSO2 souhaite développer sous la marque WSO2 API Manager l'ensemble de sa stack technologique d'intégration. Exit, donc, WSO2 Enterprise Integrator. WSO2 Micro Integrator (ESB) et WSO2 Streaming Integrator (DS/CEP) seront maintenant distribués à travers le produit WSO2 API Manager.



CATALOGUE DE SERVICES

Pour faciliter la gouvernance de l'ensemble, WSO2 ajoute une fonctionnalité de catalogue de service. A chaque fois que vous déployez un flux, sur WSO2 Micro Integrator ou WSO2 Streaming Integrator, celui-ci pourra être automatiquement déclaré au sein du catalogue de service. Et à partir du catalogue vous pourrez d'un simple clique exposer celui-ci à travers une API sur l'API Gateway.



WSO2 API MANAGER <small>PUBLISHER</small>						Search	ADMIN
Service Catalog							
Service	Service URL	Schema Type	Version	Created Time	Number of Usages		
Pizzashack-Endpoint	http://localhost/pizzashack	Open API V3	v1	a year ago	1	Create API	
New-Endpoint	http://localhost/new	AsyncAPI	v1	a year ago	1	Create API	
Rows per page: 10 1-2 of 2							

ARCHITECTURE DE L'API MANAGER

Profils

Les profils de lancement du produit évoluent. Exit key-manager, publisher, devportal et traffic manager.

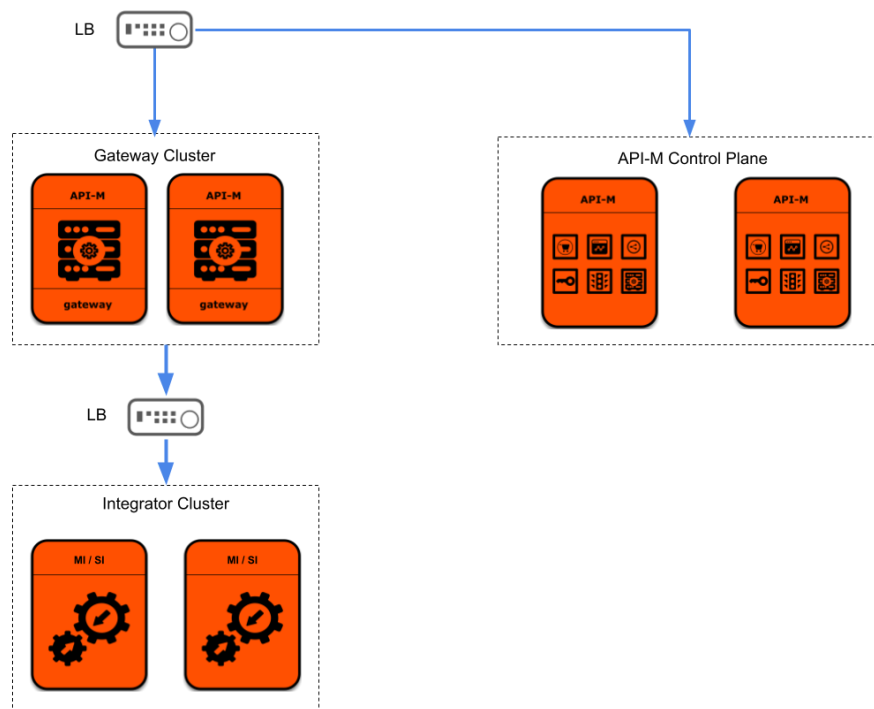
Il n'y aura maintenant plus que 4 profils possibles :

default: tous les composants

gateway-worker: uniquement la gateway

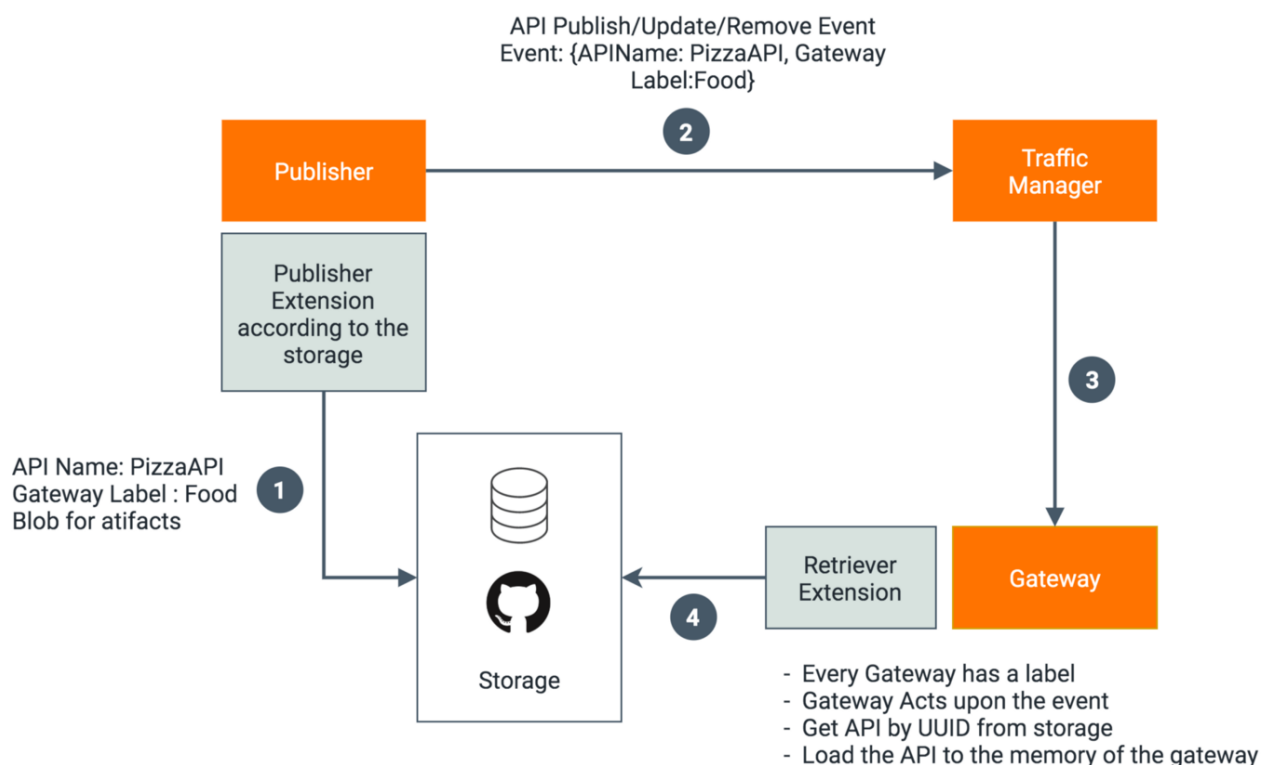
controle-plane: tous les composants sauf la gateway (traffic-manager désactivable par configuration)

traffic-manager: traffic manager seul



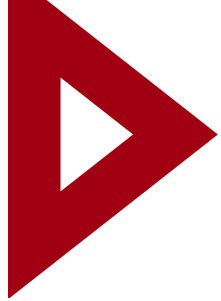
Déploiement des APIs par évènement

Introduit en 3.2.0, la publication des APIs est maintenant réalisée par défaut sur un modèle évènementiel. Plus concrètement, à chaque fois que vous publiez une API, un évènement est envoyé dans une file du broker de message embarqué. Les différentes gateway, abonnées à cette file, liront l'évènement et iront charger la nouvelle configuration d'API au sein de la base de données



Le premier intérêt de ce modèle est de pouvoir se passer d'un NFS pour synchroniser vos différentes gateway. Et les équipes WSO2, ayant pensé à tout, ils ont ajouté une étape de synchronisation des configurations avec la base de donnée au démarrage de chaque instance.

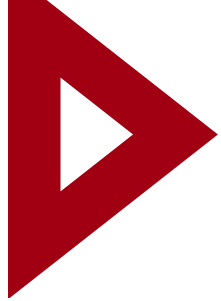
Le deuxième intérêt c'est que WSO2 en a profité, pour n'utiliser que des tokens JWT depuis la 3.2.0, et pour remonter la gestion du contrôle d'accès aux apis directement au sein de la gateway. Résultat, en dehors de requêtes d'introspection des tokens, il n'y a plus de requêtes entre gateway et key manager ce qui permet de faire sauter le ratio de 2 gateway pour un key manager qui existait depuis le début du produit. D'où la révision de la liste des profils réalisées par l'éditeur



Suppression du proxy Oauth2 au sein du composant gateway

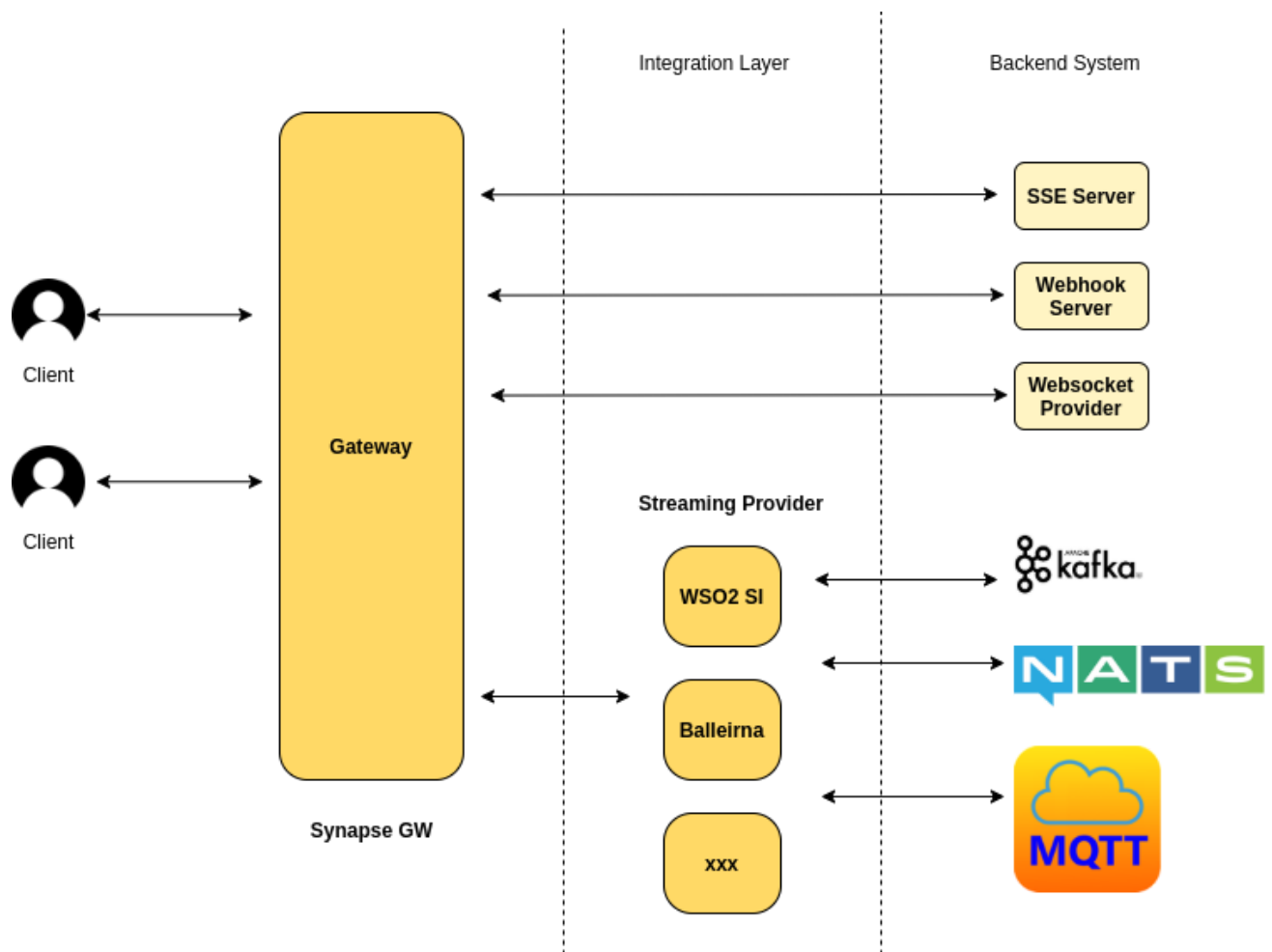
Depuis toujours nous avons l'habitude de requêter l'endpoint Oauth2, pour générer nos tokens, à travers la Gateway. La gateway, en réalité, ne faisait que passe plat vers le Key Manager. Avec la possibilité d'utiliser plusieurs Key Manager externes, depuis la 3.2.0, ce fonctionnement devenait étrange. WSO2 a donc fait le choix de supprimer ce proxy. Il vous faudra dorénavant directement interroger le Key Manager.

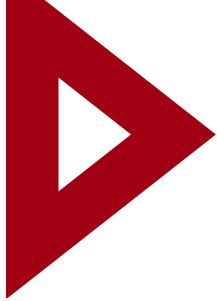
Until 3.2.0	In 4.0.0
<code>https://<gateway-host>/token</code>	<code>https://<keymanager-host>/oauth2/token</code>
<code>https://<gateway-host>/revoke</code>	<code>https://<keymanager-host>/oauth2/revoke</code>
<code>https://<gateway-host>/authorize</code>	<code>https://<keymanager-host>/oauth2/authorize</code>
<code>https://<gateway-host>/userInfo</code>	<code>https://<keymanager-host>/oauth2/userInfo</code>



APIs événementielles Standards de streaming

WSO2 API Manager étend ses capacités d'exposition en intégrant le support des APIs de streaming.





Websocket

Il gérait déjà le Websocket vous me direz?! Oui, mais de manière très limitée, il ne faisait que du proxy. Maintenant vous allez pouvoir gérer les différents canaux, les mapper aux backends et gérer leurs droits de manière individuelle.

Type ▾

Topic Name

Enter topic name

+

×

Enter topic name

/*

^

SUB

/*

🗑️

▾

PUB

/*

🗑️

▾

/notifications

^

SUB

/notifications

🗑️

▾

PUB

/notifications

🗑️

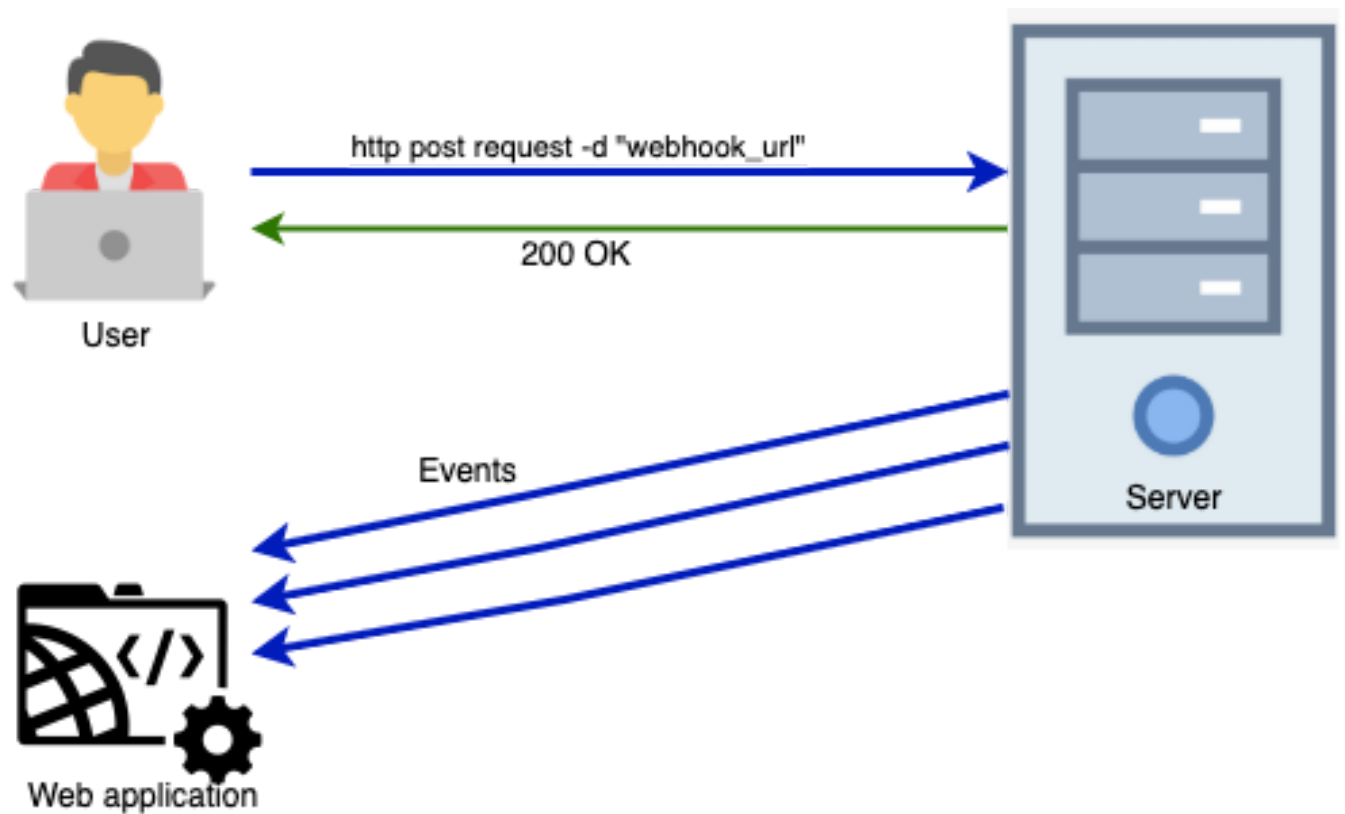
▾

Save

Reset

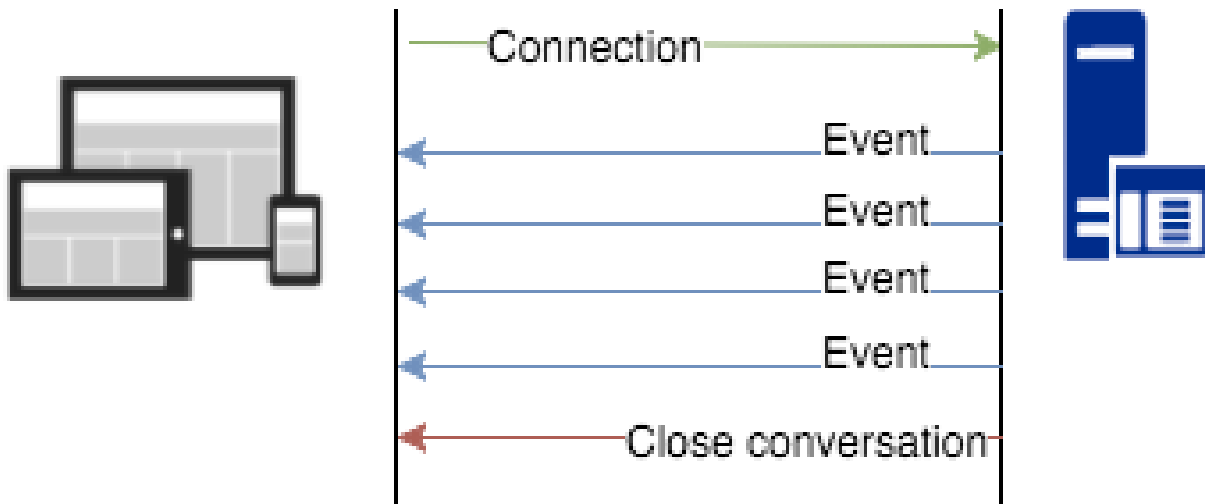
WebSub/WebHook

Le standard WebSub (anciennement PubSubHubbub) permettra à une application tierce d'enregistrer un endpoint pour être notifié d'évènements via des requêtes réalisées en POST par l'API Manager.



Server-Sent Events (SSE)

Le standard SSE permettra à une application tierce de recevoir des événements à travers une connexion HTTP déjà existante. WSO2 API Manager sera en capacité de renvoyer, sur demande, les messages non reçus par le client après une erreur de transport.



Standard de définition AsyncAPI

AsyncAPI, ou A2S pour les intimes c'est un nouveau standard de définition d'interface pour les APIs Asynchrones (créé en 2017 par la société Hitch). Sa spécification est fortement inspirée, et en partie compatible, avec la spécification OpenAPI.

Ce standard vise à remplir un manque de standardisation et de définition des messages échangés criant actuellement dans ce domaine. Contrairement à OpenAPI qui décrit uniquement des interfaces HTTP, AsyncAPI reflète l'hétérogénéité du domaine en permettant la description multi-protocolaire (MQTT, Kafka, Websockets, etc.)

WSO2 API Manager intègre maintenant nativement ce standard pour définir les points d'API événementielles avec bien sûr tout son outillage.

The screenshot displays the WSO2 API Manager interface. The top navigation bar includes the WSO2 logo, 'API MANAGER', a search bar, and user controls. The left sidebar contains a menu with sections: Overview, Develop, API Configurations, Deploy, and Publish. The 'Develop' section is expanded, and 'AsyncAPI Definition' is highlighted. The main content area shows the 'StreetlightsAPI:1.0.0' definition. The 'AsyncAPI Definition' tab is active, displaying the following YAML content:

```

1  asyncapi: 2.0.0
2  info:
3    title: Streetlights API
4    version: 1.0.0
5    description: "The Smartlighting Streetlights API allows you to remotely manage the city lights.\n\n## Check out its awesome features:\n\n* Turn a specific
6    streetlight on/off \U0001F303\n* Dim a specific streetlight \U0001F60E\n* Receive real-time information about environmental lighting conditions \U0001F4C8\n"
7  license:
8    name: Apache 2.0
9    url: 'https://www.apache.org/licenses/LICENSE-2.0'
10 servers:
11   production:
12     url: 'test.mosquitto.org:{port}'
13     protocol: mqtt
14     description: Test broker
15     variables:
16       port:
17         description: Secure connection (TLS) is available through port 8883.
18         default: '1883'
19         enum:
20           - '1883'
21           - '8883'
22   security:
23     - apiKey: []
24     - supportedOAuthFlows:
25       - 'streetlights:on'
26       - 'streetlights:off'
27       - 'streetlights:dim'
28     - openIdConnectWellKnown: []
29 defaultContentType: application/json
30 channels:
31   'smartlighting/streetlights/1/0/event/{streetlightId}/lighting/measured':
32     description: The topic on which measured values may be produced and consumed.
33     parameters:
34       streetlightId:
35         $ref: '#/components/parameters/streetlightId'

```

Gestion des révisions d'API

WSO2 API Manager gère depuis ses débuts le versionnement des APIs. Quand on parle de version d'une API on se réfère à une évolution de son contrat d'interface. Mais qu'en est-il quand on souhaite uniquement faire évoluer ses métadonnées ou sa documentation par exemple? Jusqu'à présent on écrasait les précédentes données à chaque édition et celles-ci étaient disponibles immédiatement.

Pour répondre à ce biais WSO2 vient d'ajouter la notion de révision sur une version d'API. Chaque fois que vous le jugez nécessaire vous pouvez enregistrer l'état actuel de l'API. Vous pouvez déployer explicitement une révision donnée de cette API et si elle ne vous plait pas vous pouvez restaurer la révision précédente. (WSO2 a limité à 5 révisions par API cette fonctionnalité)

The screenshot displays the WSO2 API Manager interface for an API named 'NewAPI1 :1.0' in a 'PUBLISHED' state. The interface includes a top navigation bar with various actions like 'Go To', 'View in Dev Portal', 'Create New Version', 'Download API', and 'Delete'. Below this, the 'Deployments' section shows a timeline of revisions. The first revision, 'Revision 1', is highlighted with a blue circle and has 'Restore' and 'Delete' options. A red box highlights the 'Create a new revision' button above the timeline. The 'API Gateways' section at the bottom shows a table with one gateway named 'Production and Sandbox' of type 'hybrid', with endpoints 'http://localhost:8280' and 'https://localhost:8243'. It includes a 'Deployed Revision' dropdown set to 'Select Revision', a 'Deploy' button, and a toggle for 'Display in Developer Portal' which is currently turned on.

Name	Type	Endpoints	Deployed Revision	Display in Developer Portal
Production and Sandbox	hybrid	http://localhost:8280 https://localhost:8243	Select Revision	<input checked="" type="checkbox"/>

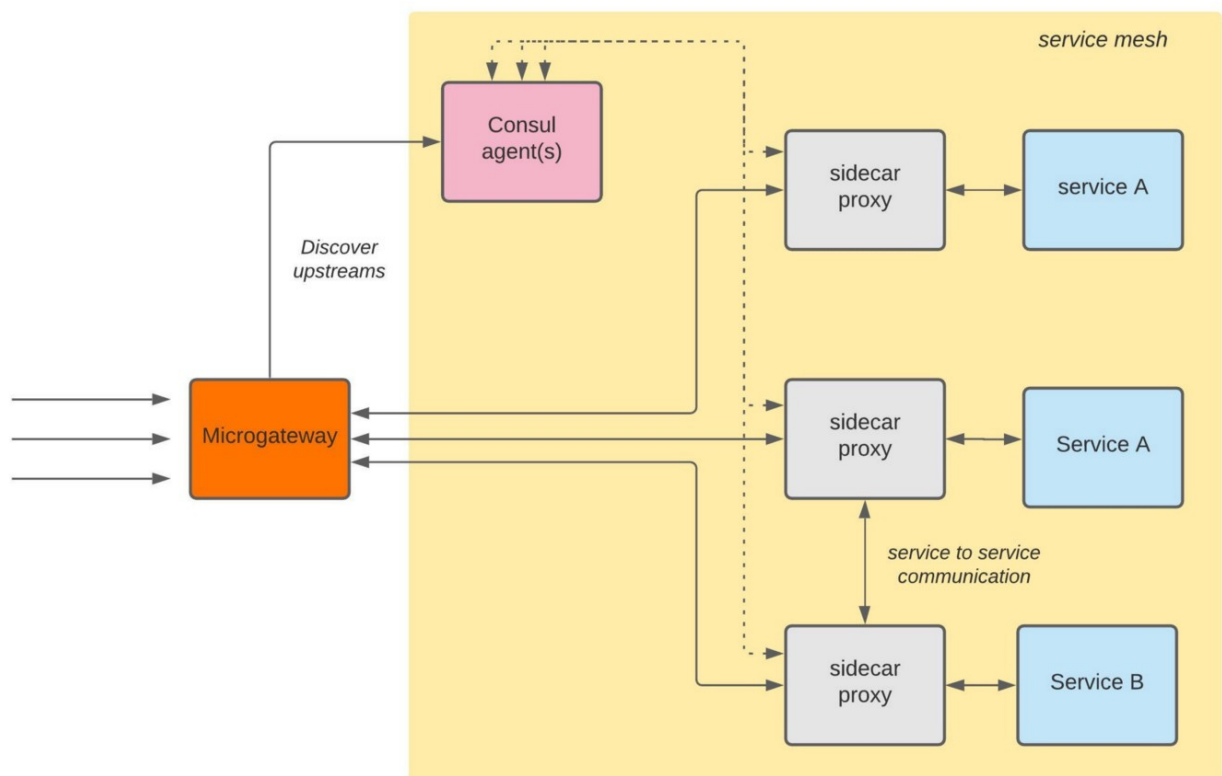
Microgateway, Choreo Connect

Les microgateway change de base technologique, elles ne seront plus basées sur le langage Ballerina mais sur le proxy Envoy.

Elles pourront être utilisées en mode autonome, comme actuellement.

Elles pourront en plus être pilotées à distance par l'outil en ligne de commande API Controller ou le Publisher de l'API Manager pour déployer à chaud des APIs.

Il sera également possible de lui demander de réaliser de la découverte de service auprès de Consul.

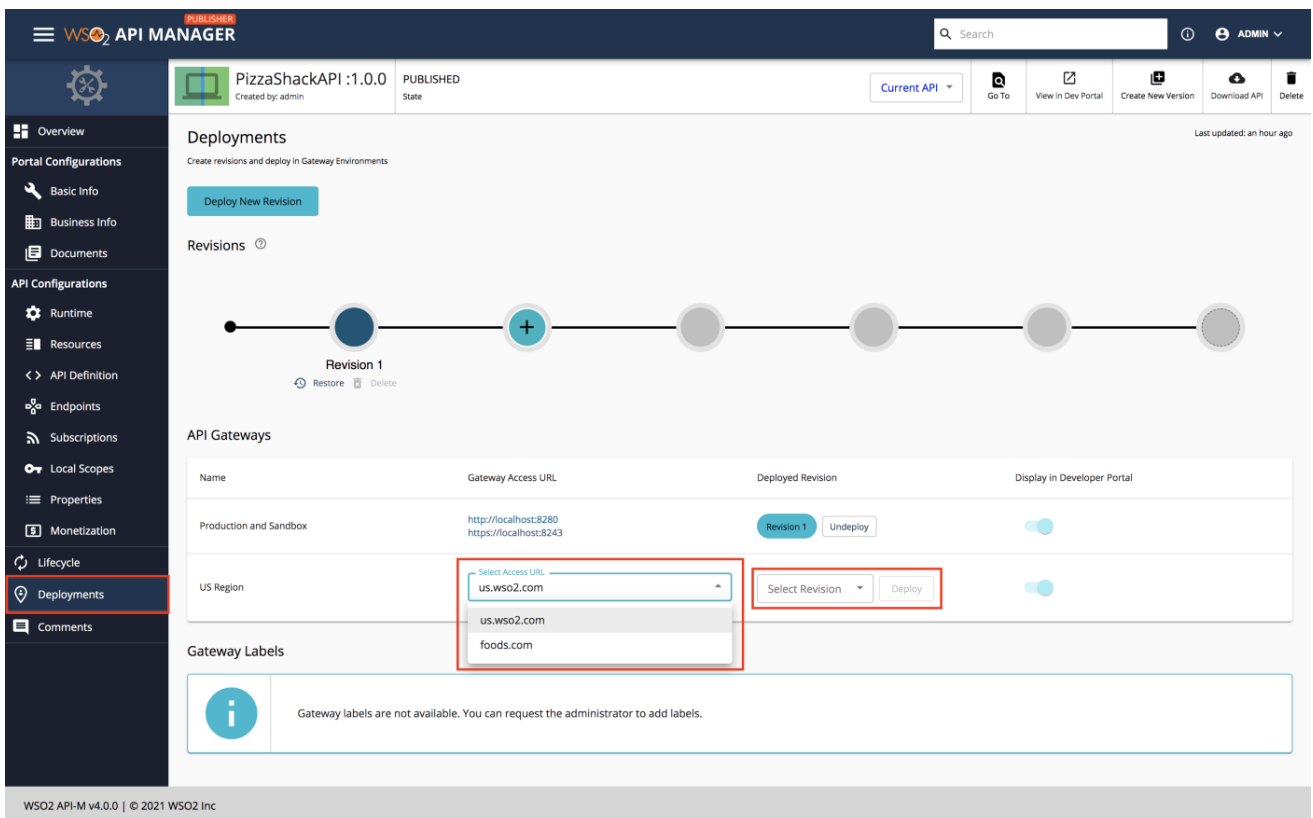


Gestion de noms de domaine virtuelle

WSO2 API Manager permettait jusqu'à présent de créer des grappes de gateway, ou Gateway environnement, qui exposaient des APIs (de production, de sandbox ou hybride) sur un même nom de domaine.

WSO2 vient d'ajouter un nouveau niveau d'abstraction pour permettre d'ajouter des vHosts.

C'est-à-dire qu'une même gateway peut exposer des APIs avec plusieurs base URL différentes (nom de domaine + path). Ainsi vous pourrez exposer à travers une même gateway 2 APIs : la première sur un domaine vhost1.com et la deuxième sur le domaine vhost2.com.



WSO2 API MANAGER

PizzaShackAPI :1.0.0 PUBLISHED State

Deployments

Create revisions and deploy in Gateway Environments

Deploy New Revision

Revisions

Revision 1

API Gateways

Name	Gateway Access URL	Deployed Revision	Display in Developer Portal
Production and Sandbox	http://localhost:8280 https://localhost:8243	Revision 1 Undeploy	<input checked="" type="checkbox"/>
US Region	Select Access URL us.wso2.com us.wso2.com foods.com	Select Revision Deploy	<input checked="" type="checkbox"/>

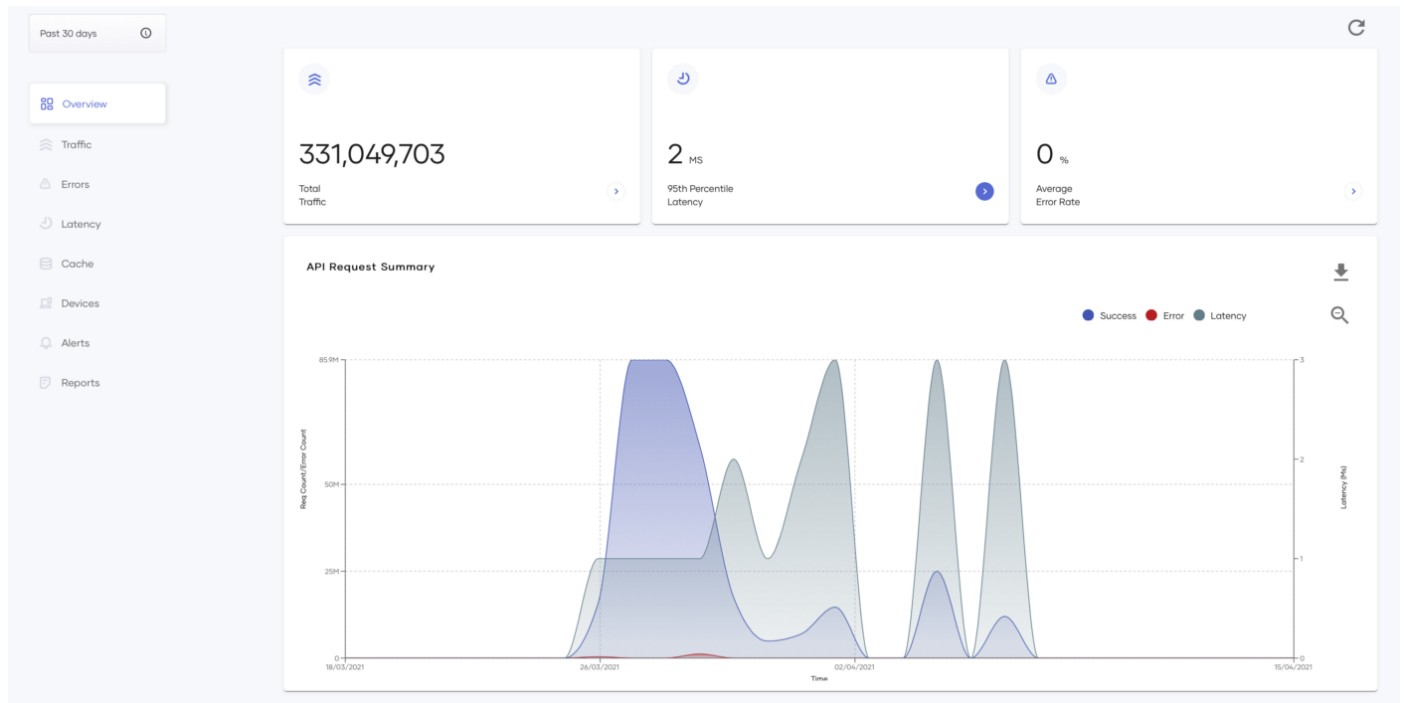
Gateway Labels

Gateway labels are not available. You can request the administrator to add labels.

WSO2 API-M v4.0.0 | © 2021 WSO2 Inc

API Analytics dans le cloud

La partie analytics du produit fait peau neuve et ... migre sur le cloud! De nombreux utilisateurs avaient du mal à gérer la volumétrie d'évènement, et WSO2 souhaitait pouvoir mieux gérer l'évolution de ce composant.



C'est sur la nouvelle plateforme Choreo, de l'éditeur, qu'elle est hébergée. Celle-ci est redondée, régionalisée et garantit la confidentialité des données transmises.

Connecteurs vers de nouveaux Key Manager externes

Depuis l'apparition de la fonctionnalité de Key Manager externe, dans la version 3.2.0, WSO2 travaille d'arrache pieds pour fournir des connecteurs avec les serveurs d'autorisation les plus standards sur le marché. A sa sortie WSO2 API Manager intégrait des connecteurs pour WSO2 Identity Server, Keycloak et Okta. Il a ajouté progressivement depuis des connecteurs pour Forgerock, Ping Federate et Auth0.

Key Managers externes supportées



Génération de Collection Postman

Beaucoup de développeurs utilisent Postman dans leurs tâches quotidiennes. Pour faciliter leur travail WSO2 génère maintenant automatiquement un fichier de définition de collection Postman téléchargeable pour les API RESTful et GraphQL.

The screenshot displays the WSO2 API Manager Developer Portal interface. The top navigation bar includes the 'API MANAGER' logo, 'APIs' and 'Applications' tabs, a search bar, and an 'ADMIN' user profile. The left sidebar contains links for 'Comments', 'Documentation', and 'SDKs'. The main content area is titled 'Security' and contains the following sections:

- Security Type:** Radio buttons for 'OAuth' (selected), 'API Key', and 'Basic'.
- Applications:** A dropdown menu showing 'DefaultApplication'.
- Key Type:** Radio buttons for 'Production' (selected) and 'Sandbox'.
- Authorization:** A text field labeled 'Authorization: Bearer' with a 'GET TEST KEY' button.
- Gateway:** A dropdown menu showing 'Production and Sandbox'.

At the bottom right of the main content area, there are two buttons: 'POSTMAN COLLECTION' (highlighted with a red box) and 'SWAGGER (/SWAGGER.JSON)'. The bottom status bar shows 'Servers'.