



Académie régionale Casablanca-Settat
Direction Provinciale hay hassani
Lycée qualifiant AL KENDI

Projet de fin d'étude
En vue de l'obtention du titre
Technicien Supérieur
Filière : Intelligence Artificielle

Sous le thème :

CASA SHOP

Révolutionner le commerce de détail en ligne grâce à l'IA

PROJET FIN D'ÉTUDE

Soutenu le 01 juillet 2024 par :

Ahmed Yassine METKOUL
Youness ATIF

Encadré par

Pr Madame LAMIAA EL JADIRI

Devant les membres du jury :

Pr .Lamia ELJADIRI
Pr .Hassna HAFIDI
Pr .Loubna AFLAH

Année Universitaire
2023-2024

Table des matières

Remerciements	4
Dédicaces	5
Introduction	6
Chapitre 1	7
1. Django c'est quoi?	7
2. Qu'est-ce que Flask ?	10
3. Qu'est-ce que FastAPI ?	12
Microservices	15
Chapitre 2	20
Web scraping avec Python	20
Qu'est-ce qu'un RNN ?	24
Qu'est-ce qu'une LSTM ?	24

Chapitre 3	_____	28
Qu'est-ce que la recommandation produit ?	_____	28
Étapes pour Construire un Système de Recommandation de Produits	_____	28
Chapitre 4	_____	32
Qu'est-ce que Docker et comment l'utiliser ?	_____	34
Qu'est-ce que GitHub Actions ?	_____	36
Kubernetes, c'est quoi ?	_____	41
Qu'est ce qui se cache derrière Kubernetes ?	_____	42
Conclusion	_____	44

Remerciements

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de l'Ecole ALKENDY et les intervenants professionnels responsables de la formation BTS, pour avoir assuré la partie théorique et pratique de celle-ci. Nous tenons à remercier en deuxième lieu, Monsieur Monsieur Safouane Zouine Chef de projet R&D ingénieur et chef d'entreprise CISCM, qui a accepté nous accueillir comme stagiaire au sein de son entreprise et Ayoub Lazraq Directeur financier et cofondateur de CISCM, pour nous avoir donné l'opportunité de passer notre stage de fin d'étude dans cette entreprise.

Au terme de notre Projet de fin d'étude nous tenons à exprimer nos sincères remerciement et gratitude à notre encadrante MADAME LAMIAA EL JADIRI, pour leur encouragement, leur soutien morale et leur conseil et aide qui nous a octroyé tout au long de la période de notre stage.

Nous adressons enfin nos vifs remerciements aux membres du jury d'avoir bien voulu examiner et juger notre travail.

Nous remercions enfin toute personne ayant contribué de près ou de loin à l'élaboration de ce projet.

Dédicaces

Je dédie ce travail à ... 

Celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère ...

Mon père, qui a été mon ombre durant toutes les années des études, et qui a veillé tout au long de ma vie à m'encourager, à m'aider et à me protéger.

Que dieu les garde et les protège.

Mes frères et Ma soeur, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour eux.

Tous mes professeurs, qui ont contribué à ma formation intellectuelle depuis mon premier pas dans l'environnement éducatif.

Mes ami(e)s, un remerciement particulier et sincère pour tous leurs efforts fournis. Ils ont toujours été présents.

Que ce travail soit un témoignage de ma gratitude et mon profond respect.

Ahmed Yassine

Youness AIT F

Introduction

Dans un paysage numérique en constante évolution, l'intégration de technologies avancées est devenue cruciale pour les entreprises cherchant à se démarquer. Casa Shop incarne cette transformation en tant que plateforme de commerce électronique innovante, basée sur une architecture de microservices. Ce rapport explore en détail le développement et l'implémentation de Casa Shop, mettant en lumière l'utilisation de modèles de Deep Learning pour le traitement du langage naturel (NLP) afin d'améliorer les fonctionnalités de filtrage et de recommandation de produits.

À travers une série de composants soigneusement conçus, Casa Shop offre une gestion efficace des opérations de commerce électronique, de la gestion des produits à l'authentification des utilisateurs, en passant par l'application des politiques de plateforme. Chaque application, qu'il s'agisse de Django pour la gestion générale ou de Flask et FastAPI pour des fonctionnalités spécialisées comme la recherche de produits et la classification basée sur les politiques, contribue à créer une expérience utilisateur fluide et sécurisée.

Ce rapport détaillera également les processus de déploiement des serveurs pour chaque composant, mettant en lumière les bonnes pratiques et les étapes nécessaires pour assurer un fonctionnement harmonieux de l'ensemble du système. Enfin, nous explorerons les avantages et les défis rencontrés lors de l'intégration de ces technologies avancées.

Casa Shop représente une illustration concrète de l'impact positif de l'architecture microservices et des modèles de Deep Learning dans le secteur du commerce électronique, offrant ainsi une vision claire de la manière dont les entreprises peuvent adopter et tirer parti de ces innovations pour stimuler leur croissance et leur compétitivité sur le marché mondial.

1

Chapitre 1

Django Microservices

django

1. Django c'est quoi?

Django est un framework python open-source consacré au développement web 2.0 . Les concepteurs de Django lui ont attribué le slogan suivant: « Le framework web pour les perfectionnistes sous pression ». Il est donc clairement orienté pour les développeurs ayant comme besoin de produire un projet solide rapidement et sans surprise ... c'est à dire à tous les développeurs !

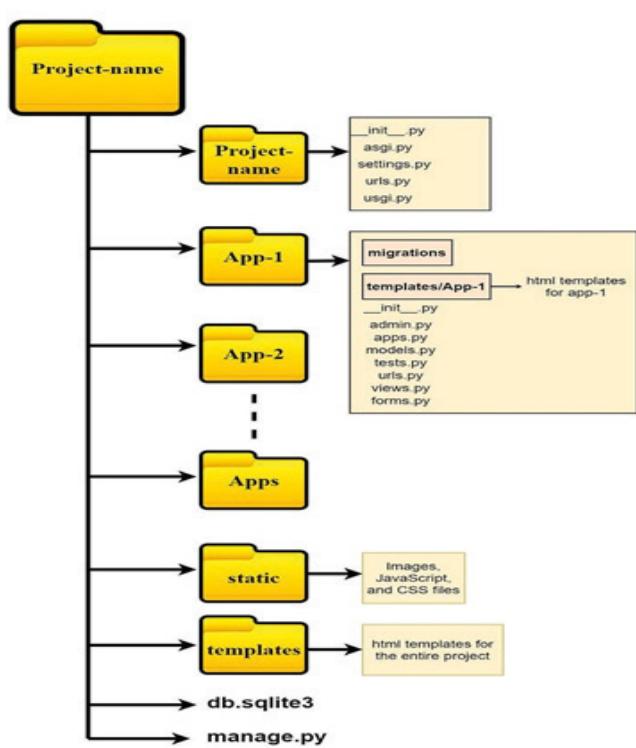
Comme il est toujours compliqué de partir de rien, Django vous propose une base de projet solide. Django est donc une belle boîte à outils qui aide et oriente le développeur dans la construction de ses projets

Pour la petite histoire Django a vu le jour en 2003 et a été publié sous licence BSD en juillet 2005.

1.1 Pourquoi travailler avec le framework Django?

Si vous êtes fan de python c'est déjà une bonne raison de s'intéresser à Django . Beaucoup de développeurs sont beaucoup plus productifs sur python que sur n'importe quel autre language. La technologie derrière ce framework peut être une bonne raison, mais Django est une raison à part entière. Tout est pensé de base. Inutile de réinventer la roue à chaque projet web: les bases sont déjà présentes. Il vous fournira les outils nécessaires à sécuriser votre application, à gérer la structure de vos modèles avec un [ORM](#), etc. Vous n'avez plus qu'à vous consacrer au code métier, à vous concentrer sur l'essence même de votre projet et rien d'autre.

1.2 La structure Django



Django s'inspire du modèle MVC (disons plutot MVT), c'est-à-dire que la structure du framework sépare les données ([models](#)) qui sont séparées des traitements ([controller](#)) qui sont eux-mêmes séparés de la vue ([view / template](#)). On vous oblige à bien coder, une structure doit être respectée et cela ne peut être que profitable au travail collaboratif ou simplement la cohérence / communication entre différents projets.

Le [moteur de template](#) de base est le plus simple que j'ai rencontré, efficace souple et facile à prendre en main. Un routeur permet de rediriger les actions en fonctions des URL et une [API](#) permet de fournir des informations sur votre projet sans passer par la case SQL. D'ailleurs l'[ORM](#) inclu vous éloignera de toute requête SQL.

Chaque projet Django vous propose de lancer son propre serveur web et d'y faire à peu près tout ce que l'on veut dans un environnement de test.

Un des concepts des plus intelligent de Django est de proposer un [espace admin](#) tout fait dans ses [contrib](#) . Une fois que vous avez créé vos modèles (la structure de votre projet) vous avez accès à une interface web [CRUD](#) en quelques minutes c'est vraiment impressionnant.un mécanisme de programmation. C'est d'ailleurs très intéressant cette homogé-

néité de structure pour des échanges de données entre plusieurs projets Django vous avez accès à une interface web CRUD en quelques minutes c'est vraiment impressionnant.

Vous pouvez évidemment personnaliser cet espace admin ou créer vos propres templates et vos propres [formulaires](#). Django vous proposera des outils pour sécuriser les données ou afficher les erreurs si besoin

1.3 Idéal pour un projet collaboratif

Un nouveau développeur qui intègre un projet Django est opérationnel immédiatement. D'une part par l'installation et la configuration de son environnement de développement mais également par le fait que tous les projets Django ont la même structure, celui-ci ne perdra jamais de temps à trouver une fonction ou comprendre

1.4 Les projets annexes



Django est au centre de beaucoup d'attentions. Etant très populaire auprès des développeurs web, des projets ambitieux comme [Django Rest Framework](#)(ou DRF) sont naturellement apparu et proposent de transformer vos projets en réelles machines de guerre!

Django propose également une grande variété de projets annexes, qui sont des extensions optionnelles développées par la communauté qui permettent d'ajouter de nouvelles fonctionnalités au framework. Ces projets annexes couvrent une grande variété de domaines, allant des formulaires aux traductions en passant par la gestion des fichiers multimédias. Ils sont faciles à utiliser et peuvent être intégrés à votre projet Django en quelques lignes de code seulement. En utilisant ces projets annexes, vous pouvez rapidement et facilement ajouter de nouvelles fonctionnalités à votre application sans avoir à réinventer la roue. Ils représentent donc une ressource précieuse pour tous ceux qui souhaitent étendre les possibilités de Django



2. Qu'est-ce que Flask ?

Flask est **un micro framework open source écrit en Python**. On parle de micro framework web parce que Flask ne propose par défaut que peu de fonctionnalités essentielles au développement web, comme la gestion des requêtes HTTP, le serveur web ou la gestion des cookies. L'objectif est de **conserver un noyau simple et léger, et surtout extensible**. Il est donc possible d'ajouter des extensions supplémentaires en fonction des besoins de l'application web. Par exemple, une base de données, un système d'authentification, la gestion des téléchargements ou une validation de formulaire.

À l'origine, Flask est né d'un défi lancé par **Armin Ronacher** qui souhaitait réaliser un framework web ultra complet contenu dans un seul fichier Python. Le moins que l'on puisse dire, c'est que le pari a réussi. Et s'il est possible que la web application tienne dans un seul fichier Python, ce n'est pas forcément nécessaire. L'idée est de pouvoir créer à la fois des apps en seulement 7 lignes de code ou beaucoup plus selon la complexité du projet.

Dans tous les cas, Flask offre aux développeurs un cadre de travail minimal pour qu'ils puissent **créer des applications web de manière rapide, simple et efficace**.

2.1 Quels sont les avantages de Flask ?

Flask figure parmi les frameworks les plus populaires de la communauté Python. Et pour cause, il a été élu « Framework web le plus populaire » par le **Python Developers Survey** en 2018. En outre, il est utilisé par certains grands sites, comme Pinterest ou LinkedIn.

Alors pourquoi un tel succès ? Voici quelques explications :

La légèreté

Flask est un **framework** léger qui ne prend pas beaucoup de place. Comme les fonctionnalités sont limitées, les développeurs ne sélectionnent que celles dont ils ont vraiment besoin.

Les extensions

ce n'est pas parce que Flask est micro qu'il est limité. En effet, le framework propose une multitude de fonctionnalités, ce qui permet aux développeurs de concevoir une grande quantité d'applications.

La personnalisation des web apps

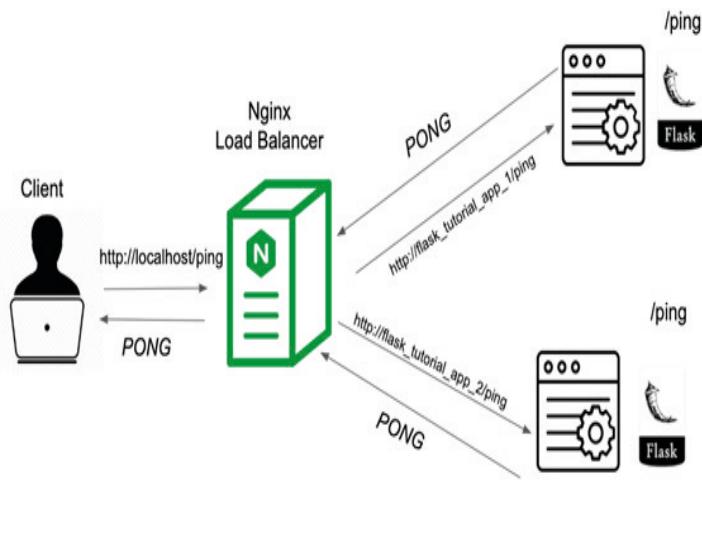
avec ses quelques fonctionnalités par défaut, Flask ne prend que peu de décisions. C'est au développeur de choisir les extensions et d'apporter les modifications nécessaires. Cela permet alors de créer une application parfaitement sur-mesure.

Même s'il s'agit d'un framework léger, Flask offre toutes les fonctionnalités indispensables au développement web. Ce faisant, les développeurs disposent d'une base de travail solide.

FastAPI

3. Qu'est-ce que FastAPI ?

FastAPI est un framework web haute performance, open source, permettant de **créer des APIs avec Python à partir de la version 3.6**. Lancé en 2018, il fut créé par Sebastián Ramírez qui n'était pas satisfait par les frameworks existants comme Flask et DRF. C'est la raison pour laquelle il a créé son propre framework à l'aide d'outils comme **Starlette** et **Pydantic**.



Aujourd'hui, de nombreuses entreprises de renommée mondiale comme **Uber**, **Netflix** et **Microsoft** utilisent FastAPI pour créer leurs applications.

Son premier point fort est un haut niveau de performances, comparables avec NodeJS et Go grâce à Starlette et pydantic. Ce framework est également **très rapide à coder**, offrant une accélération du développement.

Le nombre de bugs et de possibilités d'erreurs humaines est fortement réduit. En outre, FastAPI est très intuitif à l'utilisation, avec des fonctionnalités de complétion et de débogage. Conçu

pour être simple à apprendre et à utiliser, cet outil réduit le temps de lecture de documentation. La duplication de code est également minimisée. En outre, FastAPI délivre

un code prêt pour la production avec une documentation interactive automatique. Enfin, il est basé sur les standards ouverts OpenAI et JSON Schema

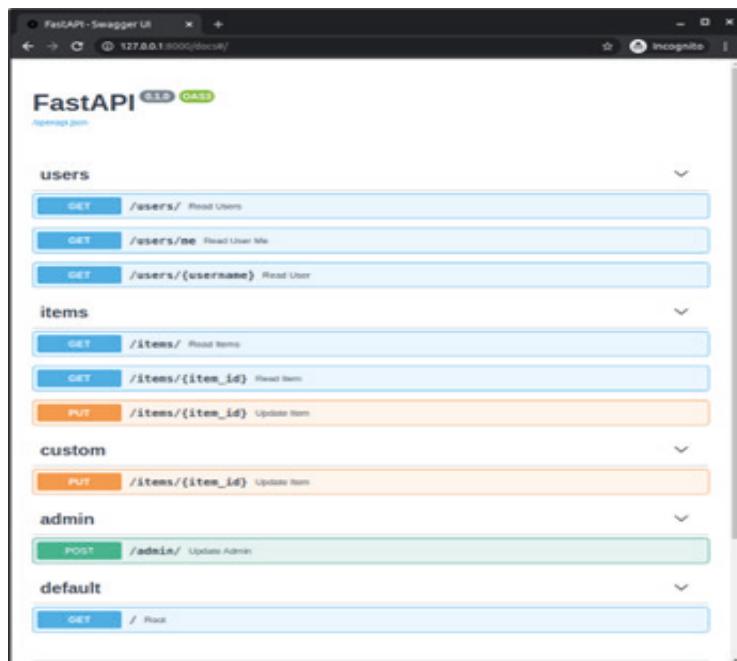
Ce framework est conçu pour optimiser l'expérience du développeur, afin qu'il puisse écrire **un code simple pour construire des APIs prêtes pour la production** en usant des meilleures pratiques

3.1 À quoi sert FastAPI ?

L'un des principaux cas d'usage de FastAPI est la **création de endpoints API**. Cette tâche peut être accomplie très facilement en utilisant le **standard OpenAPI** et notamment l'interface utilisateur interactive Swagger UI.

Toutefois, FastAPI ne se limite pas aux APIs. Il est possible de l'utiliser pour n'importe quel cas d'usage d'un framework web, comme la livraison d'anciennes pages web avec le moteur de template **Jinja2** ou pour les applications basées sur **WebSockets**.

On utilise principalement FastAPI pour les applications de **Data Science** et de e-commerce. Il permet aux développeurs d'utiliser l'**interface REST API** et d'utiliser de nombreuses fonctions pour les implémenter dans les applications.



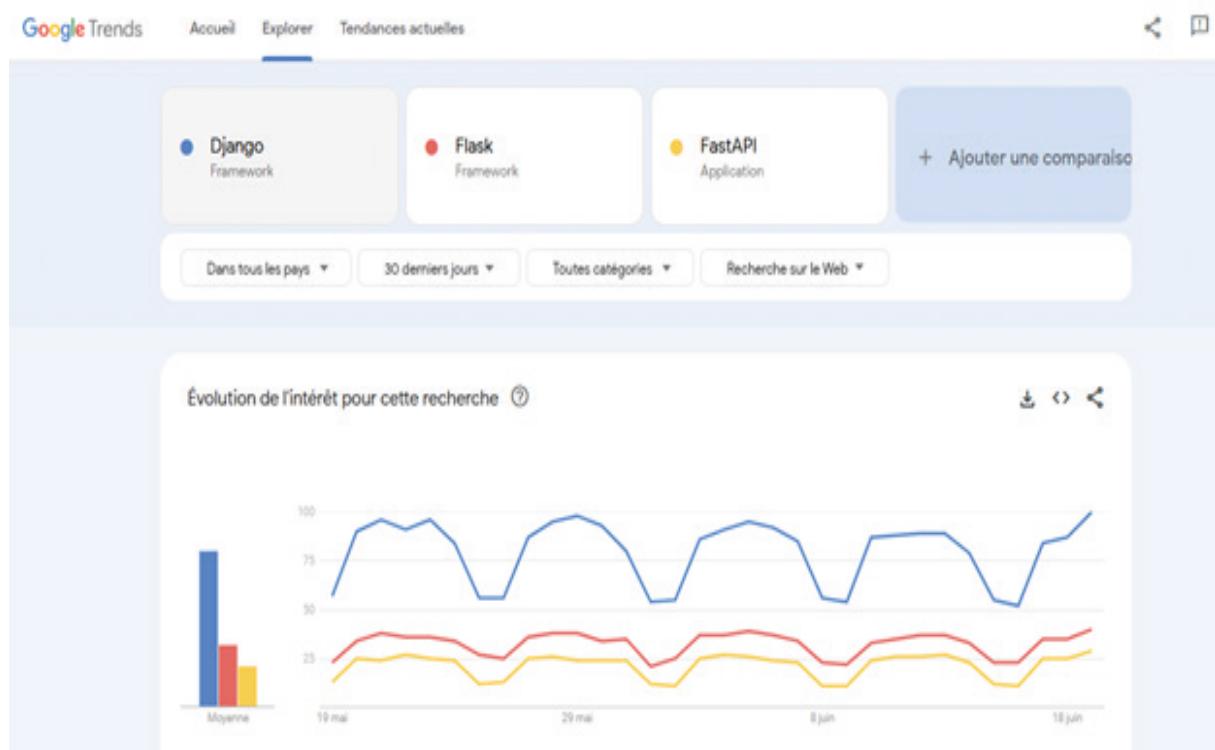
3.2 FastAPI vs Flask vs Django

Les trois principaux frameworks en Python sont **Django**, **Flask** et **FastAPI**. Chacun présente ses avantages et ses inconvénients en fonction des cas d'usage. Découvrez leurs avantages et inconvénients respectifs.

Fast API est un framework web Python hautement performant, open-source, utilisé pour développer des APIs Web avec **les types hints de Python 3.6 ou supérieur**. Il permet de valider les types de données même au sein de requêtes JSON. Il est conçu à partir de standards comme JSON Schema, OAuth 2.0 ou OpenAPI. En outre, il facilite **la création d'API GraphQL** avec une bibliothèque Python appelée graphene-python.

En revanche, son point faible est que **ce framework est relativement nouveau**. La communauté est donc plutôt réduite par rapport aux autres frameworks, et il existe peu de matériel éducatif hormis la documentation officielle.

On utilise notamment **FastAPI pour les cas d'usage où la vitesse est une priorité**. Par exemple, Netflix l'utilise pour la gestion de crise en interne. Ce framework est aussi parfaitement adapté pour le déploiement de modèles de Machine Learning puisque les modèles fonctionnent mieux en production lorsqu'ils sont accompagnés d'une **REST API** et déployés en tant que microservices.

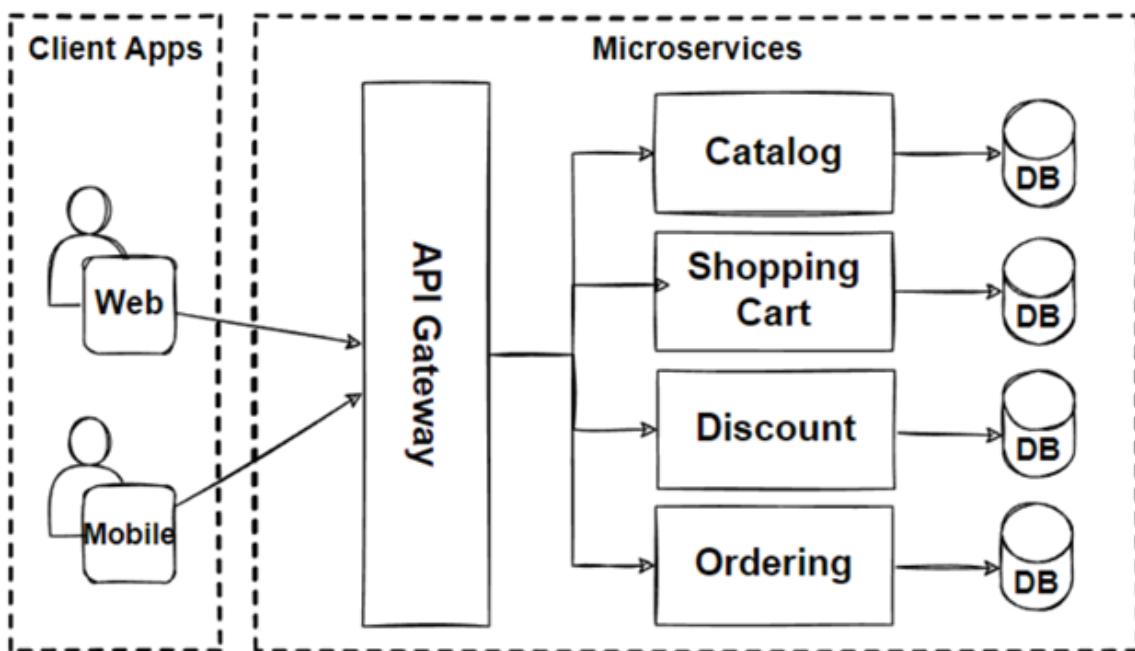


Microservices

définition

Les microservices désignent à la fois une architecture et une approche de développement logiciel qui consiste à décomposer les applications en éléments les plus simples, indépendants les uns des autres. Contrairement à une approche monolithique classique, selon laquelle tous les composants forment une entité indissociable, les microservices fonctionnent en synergie pour accomplir les mêmes tâches, tout en étant séparés. Chacun de ces composants ou processus est un microservice. Granulaire et léger, ce type de développement logiciel permet d'utiliser un processus similaire dans plusieurs applications. Il s'agit d'un élément essentiel pour optimiser le développement des applications en vue de l'adoption d'un modèle cloud-native.

Mais quel est l'intérêt d'une infrastructure basée sur des microservices ? L'objectif, qui consiste tout simplement à proposer des logiciels de qualité en un temps record, devient atteignable grâce aux microservices. Pour autant, d'autres éléments entrent également en ligne de compte. La décomposition des applications en microservices ne suffit pas. Il faut aussi gérer ces microservices, les orchestrer et traiter les données qui sont générées et modifiées par microservices.



Quels sont les avantages des microservices ?

Par rapport aux applications monolithiques, les microservices sont beaucoup plus faciles à créer, tester, déployer et mettre à jour. Chez Red Hat, nous pensons qu'ils permettent à votre entreprise de réagir plus rapidement aux nouvelles demandes et d'éviter un processus de développement interminable sur plusieurs années. Aujourd'hui, les différentes tâches de dévelop-

pement peuvent être réalisées simultanément et de façon agile pour apporter immédiatement de la valeur aux clients.

Découvrez les principes de base des microservices ainsi que leurs avantages et inconvénients, et suivez notre [formation à la demande](#) pour apprendre à créer une architecture basée sur des microservices.

Quelle est la différence entre les microservices et les API ?

Les microservices et les API sont deux approches de la conception logicielle modulaire. La programmation modulaire vise à concevoir des composants logiciels plus petits interagissant les uns avec les autres pour exécuter des fonctions complexes. C'est plus efficace que de concevoir un logiciel en tant que grande base de code pour toutes les fonctions. Les microservices sont une approche architecturale composant le logiciel en petits services indépendants très spécialisés. Chaque microservice résout un seul problème ou exécute une tâche spécifique.

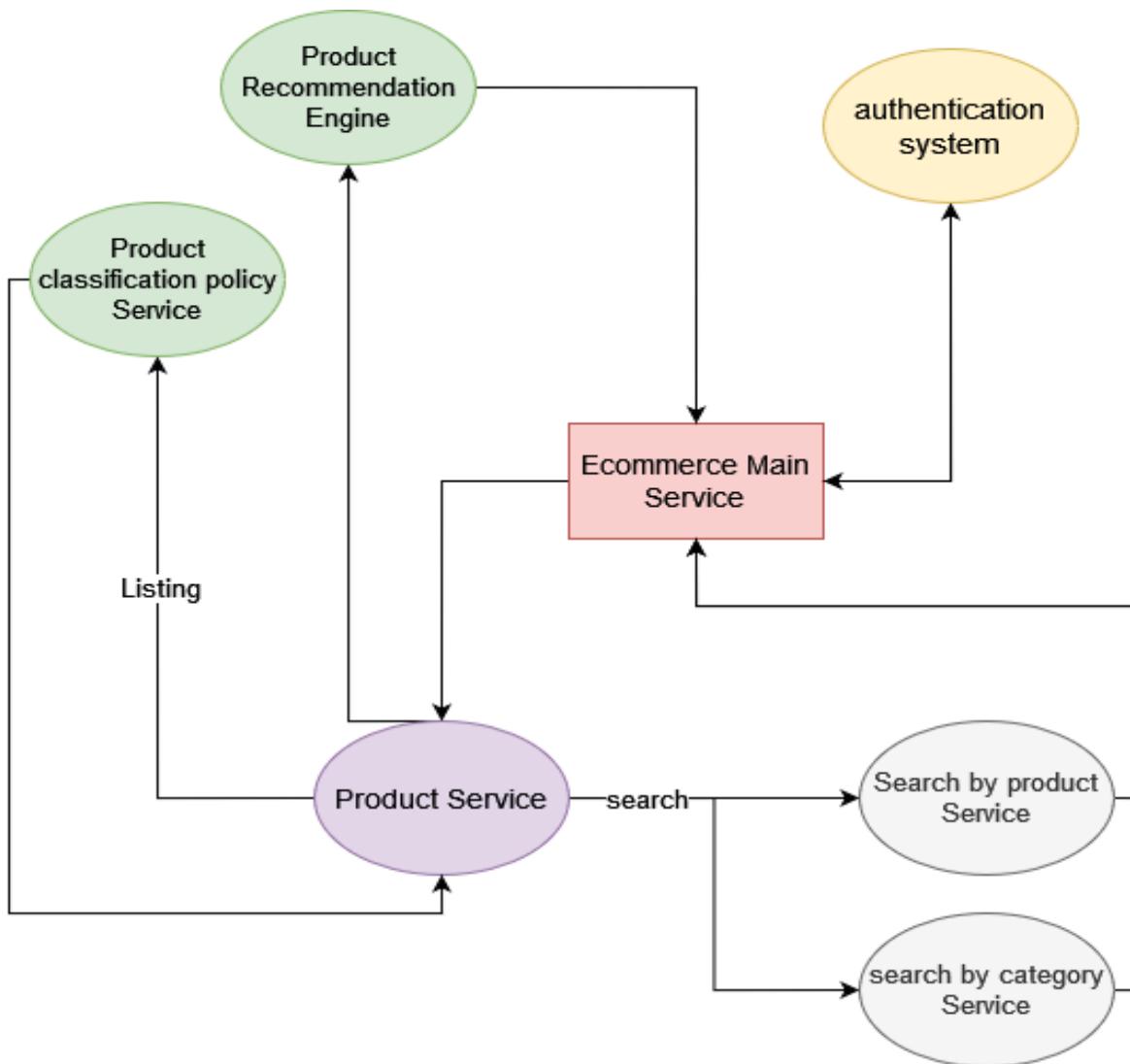
En revanche, une API est un contrat de communication entre deux composants logiciels. Les API spécifient les données dont le composant logiciel a besoin pour exécuter une fonction, ainsi que la réponse attendue. Les microservices communiquent entre eux à l'aide d'API. Toutefois, les développeurs utilisent également des API tierces pour intégrer des fonctionnalités préexistantes dans leur code.

Les API RESTful d'aujourd'hui font partie intégrante du développement d'applications Web, mobiles et back-end, comme la plomberie qui connecte le vaste écosystème numérique. Ils offrent un moyen efficace, évolutif et puissant permettant à différents composants logiciels d'interagir, d'échanger des données et de s'appuyer les uns sur les autres. L'adoption des API RESTful ne se limite pas au développement de logiciels traditionnels. Le monde des plates-formes sans code telles AppMaster a intégré des API RESTful pour garantir que les applications créées par les clients sont efficaces, adaptables, robustes et capables de communiquer avec d'autres systèmes, quels que soient leurs langages de programmation ou leur architecture. Avec la progression de la technologie, l'importance des API RESTful ne fera que croître. À mesure que la demande d'interconnectivité et de flexibilité fonctionnelle dans les applications augmente, le recours aux API RESTful en tant qu'intermédiaire efficace pour la transmission de données se renforce encore davantage.

Ainsi, comprendre et exploiter les avantages des API REST est primordial pour le développement d'applications modernes. En conclusion, qu'il s'agisse de rationaliser la communication entre les systèmes internes, d'intégrer des services tiers ou de libérer le potentiel des architectures de microservices , les API RESTful constituent la solution puissante et évolutive pour relever les défis de développement d'applications modernes. Guidées par les principes de simplicité, d'apartheid et d'évolutivité, les API RESTful ont transformé la façon dont les applications sont développées, atteignant de nouveaux sommets en termes d'efficience et d'efficacité. En intégrant des API RESTful, des plateformes comme AppMaster offrent à leurs utilisateurs la possibilité de créer des applications plus réactives, adap-

tables et efficaces, démocratisant ainsi le processus de développement d'applications. Que vous soyez un développeur chevronné, un développeur citoyen ou une entreprise cherchant à étendre sa présence numérique, l'utilisation des API RESTful, en particulier au sein des plateformes no-code, ne manquera pas d'amplifier votre succès.

Notre architectures de microservices



Cette image montre un système permettant d'acheter des choses en ligne. Il comporte différentes parties qui travaillent ensemble pour que tout se réalise.

Examinons chaque partie et comment elles fonctionnent ensemble.

1. Système d'authentification

- Ce service gère l'authentification des utilisateurs. Les utilisateurs doivent être authentifiés avant de pouvoir interagir avec le système principal de commerce électronique.
- Il est connecté au **Service principal de commerce électronique** (Ecommerce Main Service).

2. Service principal de commerce électronique

- C'est le cœur du système e-commerce. Il orchestre les interactions entre les différents services.
- Il interagit avec le **Système d'authentification**, le **Moteur de recommandation de produits** (Product Recommendation Engine), le **Service de produits** (Product Service), et les services de recherche.

3. Service de produits

- Ce service gère les produits disponibles sur la plateforme. Il est responsable de l'ajout, de la mise à jour et de la suppression des produits.
- Il reçoit des listings de produits du **Service de classification des produits** (Product Classification Policy Service) et peut effectuer des recherches de produits via les **Services de recherche**.
- Il interagit également avec le **Service principal de commerce électronique**.

4. Service de classification des produits

- Ce service s'occupe de la classification et de la catégorisation des produits selon des politiques définies.
- Il envoie les listings de produits au **Service de produits** et au **Moteur de recommandation de produits**.

5. Moteur de recommandation de produits:

- Ce service fournit des recommandations de produits aux utilisateurs, basées sur leurs comportements, préférences et autres critères.
- Il est alimenté par les données du **Service de classification des produits** et interagit avec le **Service principal de commerce électronique**.

6. Services de recherche

- Recherche par produit** (Search by product Service) : Ce service permet aux utilisateurs de rechercher des produits spécifiques.
 - Recherche par catégorie** (Search by category Service) : Ce service permet aux utilisateurs de rechercher des produits par catégorie.
 - Les deux services de recherche sont alimentés par le **Service de produits**.
- Les flèches montrent comment les différentes parties du système de commerce électronique sont intégrées et interagissent entre elles.

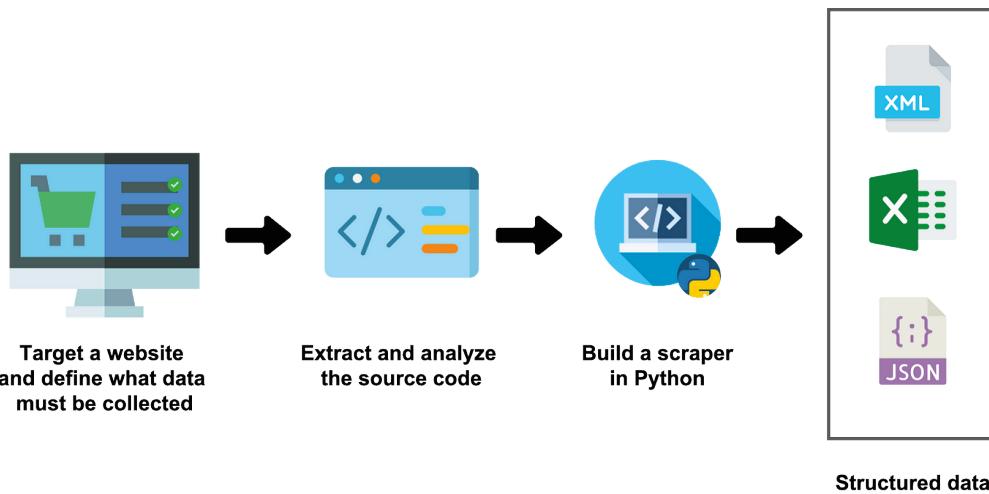
nique communiquent entre elles.

Chaque partie a son propre travail et communique clairement avec les autres. Cela aide le système à se développer et à continuer de bien fonctionner.

Chapitre 2

Data Scraping & Preprocessing for Product Classifier

Web scraping avec Python



Le World Wide Web est composé de milliards de documents reliés entre eux et appelés communément « sites Internet ». Le code source de ces sites Internet est écrit en langage Hypertext Markup Language (HTML). Ce code source HTML est un mélange d'informations lisibles par l'homme et de codes lisibles par les machines, que l'on appelle balises. Le navigateur web – par ex. Chrome, Firefox, Safari ou Edge – traite le code source, interprète les balises et met les informations qu'elles contiennent à disposition de l'utilisateur. Des logiciels spécifiques sont utilisés afin d'extraire uniquement du code source les informations intéressantes pour l'être humain. Ces programmes – connus sous le nom de « web

scrapers », « robots d'indexation », « spiders » ou simplement « bots » – parcourrent le code source des sites Internet à la recherche de schémas et extraient les informations contenues à ces endroits. Les informations obtenues lors du web scraping sont rassemblées, combinées, analysées ou enregistrées pour une utilisation ultérieure.

Domaines d'application du web scraping

Le web scraping peut être utilisé à des fins diverses. En dehors de l'indexation par les moteurs de recherche, il est notamment utilisé pour :

- ◎ créer des bases de données de contact,
- ◎ surveiller et comparer les prix des offres en ligne,
- ◎ rassembler des données de différentes sources en ligne,
- ◎ assurer un suivi de la présence et de la réputation en ligne,
- ◎ collecter des données financières, météorologiques et autres,
- ◎ surveiller les modifications apportées aux contenus web,
- ◎ collecter des données pour la recherche,
- ◎ miner des données.

Outils de web scraping Python

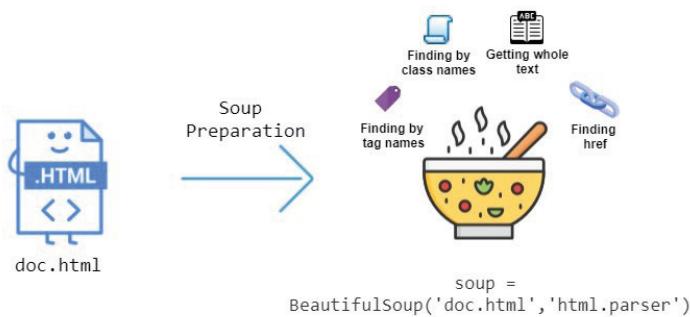


Dans l'écosystème Python, il existe plusieurs outils établis permettant de réaliser un projet de web scraping :

- ◎ Scrapy
- ◎ Selenium
- ◎ BeautifulSoup

Nous vous présentons maintenant les avantages et les inconvénients de ces trois technologies.

Web scraping avec BeautifulSoup



BeautifulSoup est le plus ancien des outils de web scraping Python présentés ici. À l'instar de Scrapy, il s'agit d'un analyseur syntaxique HTML. Par conséquent, un web scraping avec BeautifulSoup suit le schéma suivant :

URL → requête HTTP → HTML → BeautifulSoup

Contrairement à Scrapy, le développement d'un scraper avec BeautifulSoup ne requiert pas de programmation orientée objet. À la place, le scraper y est écrit comme un script simple. BeautifulSoup est ainsi le moyen le plus simple d'obtenir des informations ciblées dans cette « soupe de balises »

Extraction de Données sur les Produits Non Conformes et Conformes à partir d'Alibaba et Amazon

Le premier script

```
1 import requests
2 from bs4 import BeautifulSoup
3
4
5 file = open("No_products_datasets_v2.txt", 'a',encoding="utf-8")
6
7
8 for i in range(200):
9
10     #drugs product
11     URL_drugs = f"https://www.alibaba.com/trade/search?spm=a2700.galleryofferlist.0.0.6
12     page = requests.get(URL_drugs)
13
14     #all product in page
15     soup_drugs = BeautifulSoup(page.content, 'lxml')
16     products_drugs = soup_drugs.find('div',{"class":"organic-list"})
17     products = products_drugs.find_all('h2',{"class":"search-card-e-title"})
18
19     for prod in products:
20         result = f'{prod.text.strip().replace(",",".")},No\n'
21         print(result)
22         file.write(result)
23
24
25     #weapons product
26
27     URL_weapons = f"https://www.alibaba.com/trade/search?spm=a2700.galleryofferlist.0.0.6
28     page = requests.get(URL_weapons)
```

Le premier script

Destiné à l'extraction de données sur les produits non conformes, fonctionne en ciblant des catégories spécifiques de produits généralement interdits ou restreints sur la plateforme Alibaba. Utilisant les bibliothèques Python `requests` et `BeautifulSoup`, le script récupère et analyse le contenu des pages web de résultats de recherche pour des mots-clés tels que «drogues», «armes» et «objets sexuels». Pour chaque catégorie, il parcourt plusieurs pages de résultats et extrait les noms des produits affichés, qui sont ensuite nettoyés et formatés pour éviter les erreurs de caractères spéciaux. Les noms de ces produits sont ensuite enregistrés dans un fichier texte nommé `No_products_datasets_v2.txt`, avec une étiquette «No» pour indiquer qu'il s'agit de produits non conformes. Ce processus permet de constituer une base de données des produits interdits, utile pour des analyses ultérieures ou le développement d'algorithmes de filtrage automatique.

Le second script

Se concentre sur l'extraction de données sur les produits conformes, en utilisant des fichiers HTML enregistrés de la plateforme Amazon. Ce script emploie également la bibliothèque `BeautifulSoup` pour analyser le contenu HTML des fichiers. Il parcourt ces fichiers pour extraire les noms des produits à partir d'éléments HTML spécifiques identifiés par leurs classes CSS. Le script traite chaque fichier HTML un par un, identifiant les produits qui respectent les politiques de la plateforme. Les noms des produits conformes sont ensuite stockés dans un fichier texte nommé `Yes_products_datasets.txt`, accompagnés d'une étiquette «Yes» pour indiquer leur conformité. Ce processus génère un ensemble de données de produits autorisés, qui peut être utilisé comme référence pour comparer avec les produits non conformes ou pour entraîner des modèles d'apprentissage automatique à reconnaître les produits conformes. Ensemble, ces deux scripts fournissent des jeux de données complets et distincts pour l'analyse de la conformité des produits sur les plateformes de commerce électronique.

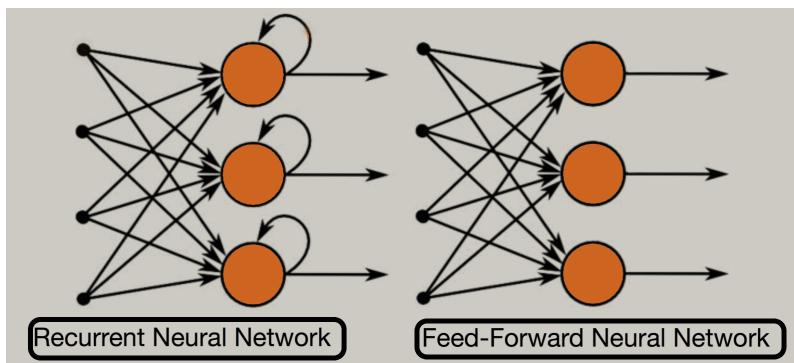
```

product-filter-policy-model-with-nlp / amazon products / scrap amazon products.py □
Code Blame 26 lines (18 loc) + 661 Bytes Code 55% faster with GitHub Copilot
ynstf init

1  from bs4 import BeautifulSoup
2  import os
3
4  cmd='dir'
5  res = os.popen(cmd).read()
6
7  page_number = len(res.split(".htm"))-1
8
9  file = open("Yes_products_datasets.txt", 'a',encoding="utf-8")
10
11 for n in range(1,page_number+1):
12
13     #file = input('enter your amazon page : ')
14     with open(f"Amazon{n}.htm","r", encoding="utf-8") as html:
15         soup = BeautifulSoup(html, 'lxml')
16
17         products = soup.find_all('div',{"class":"DealContent-module__truncate_swbxETx42ZPStTc9jwySwI"})
18
19         for prod in products:
20             result = f'{prod.text.strip().replace(",",".")},Yes\n'
21             #print(result)
22             print(n)
23             file.write(result)
24
25     #close the file
26     file.close()

```

Qu'est-ce qu'un RNN ?



Les réseaux neuronaux récurrents RNN sont un type de réseau neuronal conçu pour traiter des données séquentielles. Ils peuvent analyser des données ayant une dimension temporelle, telles que des séries chronologiques, des discours et des textes. Les RNN y parviennent en utilisant un état caché transmis d'un pas de temps à l'autre. L'état caché est mis à jour à chaque étape en fonction de l'entrée et de l'état caché précédent. Les RNN sont capables de capturer les dépendances à court terme dans les données séquentielles, mais ils ont du mal à capturer les dépendances à long terme.

Qu'est-ce qu'une LSTM ?

Les LSTM (Long Short-Term Memory) sont un type de réseau neuronal récurrent RNN (Recurrent Neural Network) capable de retenir des dépendances à long terme dans des données séquentielles. Les LSTM sont capables de traiter et d'analyser des données séquentielles, telles que des séries temporelles, des textes et des discours. Ils utilisent une cellule de mémoire et des portes pour contrôler le flux d'informations, ce qui leur permet de retenir ou d'éliminer sélectivement des informations en fonction des besoins et d'éviter ainsi le problème de la disparition du gradient qui affecte les RNN traditionnels. Les LSTM sont largement utilisées dans diverses applications telles que le traitement du langage naturel, la reconnaissance vocale et la prévision des séries temporelles.

Structure des LSTM

Un réseau LSTM (Long Short-Term Memory) est un type de réseau neuronal récurrent RNN capable de traiter des données séquentielles. La structure d'un réseau LSTM consiste en une série de cellules LSTM, dont chacune possède un ensemble de portes (portes d'entrée, de sortie et d'oubli) qui contrôlent le flux d'informations entrant et sortant de la cellule. Les portes sont utilisées pour oublier ou conserver de manière sélective les informations des pas de temps précédents, ce qui permet à la LSTM de maintenir des dépendances à long terme dans les données d'entrée.

La cellule LSTM possède également une cellule mémoire qui stocke les informations des pas de temps précédents et les utilise pour influencer la sortie de la cellule au pas de temps actuel. La sortie de chaque cellule LSTM est transmise à la cellule suivante du réseau, ce qui permet à la LSTM de traiter et d'analyser des données séquentielles sur plusieurs pas de temps.



Prédiction de la Conformité aux Politiques des Produits en Utilisant un Modèle LSTM : Étapes et Résultats

Cette étude vise à développer un modèle prédictif de la conformité aux politiques des produits en utilisant les réseaux de neurones Long Short-Term Memory (LSTM), un type de réseau de neurones récurrent (RNN) réputé pour sa capacité à apprendre à partir de données séquentielles. Les étapes suivantes ont été suivies pour mener à bien cette recherche :

1. Prétraitement des données

- Fusion de plusieurs ensembles de données pour créer un dataset complet contenant les noms de produits et leurs statuts de conformité.
- Suppression des doublons pour assurer l'unicité des données.
- Transformation des indicateurs catégoriels de conformité en valeurs numériques binaires pour faciliter l'analyse.

```
In [10]:  
data_no_duplicates = data.drop_duplicates(subset='Name of Product', keep='first')  
# Print the length of the new DataFrame to confirm the removal of duplicates  
print("Length after removing duplicates:", len(data_no_duplicates))
```

Length after removing duplicates: 6311

```
In [11]:  
data = data_no_duplicates  
print('the data with No in Is Accepted Policy', len(data[data["Is Accepted Policy"]=="No"]))  
print('the data with Yes in Is Accepted Policy', len(data[data["Is Accepted Policy"]=="Yes"]))
```

the data with No in Is Accepted Policy 5012
the data with Yes in Is Accepted Policy 1299

2. Prétraitement du texte

- Tokenisation et lemmatisation des données textuelles en utilisant le Stemmer de Porter, ce qui permet de réduire les mots à leur forme de base et d'améliorer la compréhension sémantique par le modèle.

select the fixed-length input sequences

```
In [15]:  
from tensorflow.keras.preprocessing.sequence import pad_sequences  
  
MAX_SEQUENCE_LENGTH = 15  
  
X = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)  
X
```

```
Out[15]:  
array([[ 0,  0,  0, ...,  5,  8, 18],  
      [ 0,  0,  0, ..., 114,  5,  8],  
      [ 0,  0,  0, ...,  0,  5,  1],  
      ...  
      [ 0, 19, 244, ..., 153,  6,  1],  
      [ 0,  0,  0, ..., 74,  1, 22],  
      [ 0,  0,  0, ..., 11, 201, 55]], dtype=int32)
```

3. Division des données

- Séparation des données en ensembles d'entraînement et de test pour permettre l'évaluation de la performance du modèle.

split data to train & test

```
In [17]:  
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
  
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train, test_size=0.33, random_state=42)
```

4. Entraînement du modèle LSTM

- Utilisation de l'ensemble d'entraînement pour apprendre les dépendances temporelles et les motifs présents dans la séquence de tokens des noms de produits.

train model

```
In [20]:  
batch_size = 32  
epochs = 5  
model.fit(X_train, y_train, validation_data=(X_test, y_test), batch_size=batch_size, verbose=1, epochs=epochs)  
  
Epoch 1/5  
106/106 [=====] - 8s 45ms/step - loss: 0.3228 - accuracy: 0.8672 - val_loss: 0.1591 - val_accuracy: 0.9359  
Epoch 2/5  
106/106 [=====] - 4s 40ms/step - loss: 0.1376 - accuracy: 0.9438 - val_loss: 0.1482 - val_accuracy: 0.9438  
Epoch 3/5  
106/106 [=====] - 5s 43ms/step - loss: 0.1205 - accuracy: 0.9542 - val_loss: 0.1414 - val_accuracy: 0.9422  
Epoch 4/5  
106/106 [=====] - 5s 43ms/step - loss: 0.1111 - accuracy: 0.9565 - val_loss: 0.1471 - val_accuracy: 0.9414  
Epoch 5/5  
106/106 [=====] - 4s 42ms/step - loss: 0.1012 - accuracy: 0.9598 - val_loss: 0.1478 - val_accuracy: 0.9446  
  
Out[20]:  
<keras.callbacks.History at 0x7a9e131487f0>
```

5. Évaluation des performances

- Utilisation de métriques clés telles que la précision, le rappel et l'exactitude pour évaluer les performances du modèle.

classification report

```
In [27]: from sklearn.metrics import classification_report

#calculate and print classification report
print("Classification Report:")
print(classification_report(y_test_binary, y_pred_binary))
```

Classification Report:				
	precision	recall	f1-score	support
0	0.98	0.95	0.96	999
1	0.83	0.92	0.87	264
accuracy			0.94	1263
macro avg	0.91	0.94	0.92	1263
weighted avg	0.95	0.94	0.95	1263

Les résultats montrent que le modèle LSTM prédit efficacement la conformité aux politiques, offrant ainsi un outil précieux pour les entreprises afin de vérifier automatiquement les noms de produits par rapport aux politiques de conformité. Cela permet de garantir le respect des normes réglementaires et de minimiser le risque de non-conformité. Les travaux futurs pourraient explorer l'amélioration du modèle en intégrant des fonctionnalités supplémentaires telles que les descriptions de produits et les avis des utilisateurs, ce qui pourrait potentiellement améliorer encore la précision des prédictions.

The screenshot shows a web-based product submission form for 'CASA Shop'. At the top, there are navigation links for 'Browse Categories', 'Home', 'Shop', 'Product', 'Pages', 'Blog', 'Elements', and a 'Logout' button. On the right side, there are icons for 'Compare', 'Wishlist', and 'Cart'. A red arrow points to an orange banner at the top of the page with the text: 'This product does not comply with our policy. [Check the policy for details.](#)' Below this, another red arrow points to a redacted name field in a 'New Product' form. The 'New Product' form has fields for 'Name:' (with a red arrow pointing to it) and 'Short description:' (with a red arrow pointing to it). The 'Short description:' field contains the word 'test'.

Chapitre 3

Systèmes de recommandation de produit

Les systèmes de recommandation de produit sont souvent les premières applications d'intelligence artificielle que souhaitent mettre en place les e-commerçants sur leur site web. C'est un choix logique qui aura un impact visible très rapidement sur les ventes. Voici quelques informations pour expliciter l'intérêt de l'intelligence artificielle sur la recommandation des produits..

Qu'est-ce que la recommandation produit ?

La recommandation produit existe sur la majorité des sites e-commerce mais aussi sur des outils marketing ou d'aide à la vente. Dans le cas d'un e-commerce elle est visible en bas d'une fiche produit ("vous aimerez aussi..."), lors de l'ajout d'un produit au panier ("les clients qui ont acheté ce produit ont également acheté..."), ou à d'autres emplacements du site. On trouve également très fréquemment des recommandations de produits dans les newsletters.

Sans algorithmes d'intelligence artificielle, les produits recommandés sont soit définis par le e-commerçant ou l'administrateur des ventes, soit définis par une règle statistique basique (historique des ventes). Avec de l'intelligence artificielle, il devient possible d'automatiser ces suggestions et de les personnaliser à chaque client, en fonction des pages visitées, de son historique d'achat, du contenu de son panier, de son profil... et ainsi de proposer intelligemment les bons produits au bon client, au moment opportun. Cette personnalisation accrue est une stratégie très efficace pour développer les ventes, que ce soit dans un e-commerce ou dans le CRM d'un commercial.

Le modèle de recommandation décrit est basé sur le filtrage collaboratif item-item. Contrairement au filtrage collaboratif utilisateur-utilisateur qui se concentre sur les simi-

litudes entre les utilisateurs en fonction de leurs préférences, le filtrage collaboratif item-item se concentre sur les similitudes entre les produits eux-mêmes. Il utilise l'historique d'achats des clients pour calculer à quel point les produits sont similaires en termes de la fréquence à laquelle ils sont achetés ensemble par les mêmes clients. À partir de cette analyse des interactions produit-produit, le modèle génère des recommandations en identifiant d'autres produits que les clients pourraient apprécier sur la base des produits qu'ils ont déjà achetés ou envisagés. Ainsi, il exploite les associations entre les produits pour prédire les préférences des clients et fournir des recommandations personnalisées.

Le type de système de recommandation décrit dans les étapes fournies est principalement un système de recommandation par *filtrage collaboratif*, plus spécifiquement basé sur le **filtrage collaboratif item-item**. Voici pourquoi :

Filtrage Collaboratif

Le filtrage collaboratif est une approche populaire dans les systèmes de recommandation qui utilise le comportement des utilisateurs ou des articles pour prédire leurs intérêts. Il ne repose pas sur des caractéristiques explicites ou des attributs des articles, mais plutôt sur les modèles de comportement des utilisateurs ou des associations d'articles.

Filtrage Collaboratif Item-Item

Dans le filtrage collaboratif item-item :

- Calcul de Similarité : Il calcule la similarité entre les articles en fonction de la fréquence à laquelle ils sont achetés ensemble par les utilisateurs. Cela est capturé dans la «matrice pondérée d'interactions entre produits» et par la suite dans la «matrice de probabilités» où la probabilité représente la probabilité de co-achat entre les articles.
- Génération de Recommandations : Lorsqu'un utilisateur ajoute des articles à son panier, le système identifie d'autres articles qui ont été fréquemment achetés avec ceux du panier.

Cela se fait en récupérant et en triant les articles en fonction de leurs probabilités dans la matrice de probabilités.

Caractéristiques

- *Approche Basée sur la Mémoire* : Elle utilise l'ensemble de l'historique des interactions des utilisateurs (historique des achats) directement pour générer des recommandations sans apprendre explicitement un modèle.
- Interaction Utilisateur-Item : Bien que cela ne soit pas détaillé explicitement dans les étapes, le filtrage collaboratif considère généralement les interactions utilisateur-item (comportement d'achat) pour déduire les préférences des utilisateurs.

Types de Systèmes de Recommandation

Dans la taxonomie des systèmes de recommandation, cette approche relève de la catégorie du *filtrage collaboratif*, qui se subdivise en :

- Filtrage Collaboratif Utilisateur-Utilisateur : Se concentre sur les similarités entre les utilisateurs en fonction de leurs préférences.
- Filtrage Collaboratif Item-Item : Se concentre sur les similarités entre les articles en fonction des interactions des utilisateurs.

Conclusion

Ainsi, en se basant sur les étapes fournies (exploitant les probabilités de co-achat de produits dérivées de l'historique des achats des clients), le système de recommandation décrit est principalement un système de recommandation par *filtrage collaboratif item-item*. Il recommande des articles aux utilisateurs en fonction des similarités dans les patterns de co-achat des articles, ce qui en fait une approche puissante et efficace pour de nombreuses tâches de recommandation.

Étapes pour Construire un Système de Recommandation de Produits

1. Importer les Bibliothèques et Charger les Données

- Importer les bibliothèques nécessaires telles que pandas, numpy, networkx, matplotlib, ainsi que toute bibliothèque spécifique pour la détection de communautés.
- Charger le jeu de données contenant l'historique des achats des clients.

2. Nettoyage et Préparation des Données

- Traiter les valeurs manquantes, notamment celles de CustomerID.
- Convertir les colonnes nécessaires aux types de données appropriés.
- Regrouper les données par CustomerID et StockCode, en agrégeant les quantités si nécessaire.

3. Créer la Matrice de Notations (Ratings Matrix) :*

- Pivoter les données nettoyées pour créer une matrice de notations où les lignes représentent les clients, les colonnes représentent les produits, et les valeurs représentent les quantités d'achat ou des indicateurs binaires.

4. Binariser la Matrice de Notations

- Convertir la matrice de notations en une forme binaire si nécessaire, indiquant si un client a acheté un produit ou non.

5. Créer la Matrice Pondérée d'Interactions entre Produits

- Initialiser une matrice pour capturer les interactions entre produits basées sur les achats des clients.
- Remplir cette matrice en comptant combien de fois des paires de produits sont achetées ensemble.

6. Générer la Matrice de Probabilités

- Transformer la matrice pondérée d'interactions entre produits en une matrice de probabilités.
- Normaliser les valeurs dans la matrice pour représenter les probabilités de co-achats de produits.

7. Détection de Communautés

- Utiliser des algorithmes de détection de communautés (par exemple, méthode de Louvain) pour identifier des groupes de produits (communautés) fréquemment achetés ensemble.

8. Logique de Recommandation

- Pour un panier donné d'articles ajoutés par un client :
- Récupérer les probabilités correspondantes depuis la matrice de probabilités.
- Trier et sélectionner les meilleurs produits avec les probabilités les plus élevées.
- Recommander ces produits au client comme articles supplémentaires qu'ils pourraient envisager d'acheter.

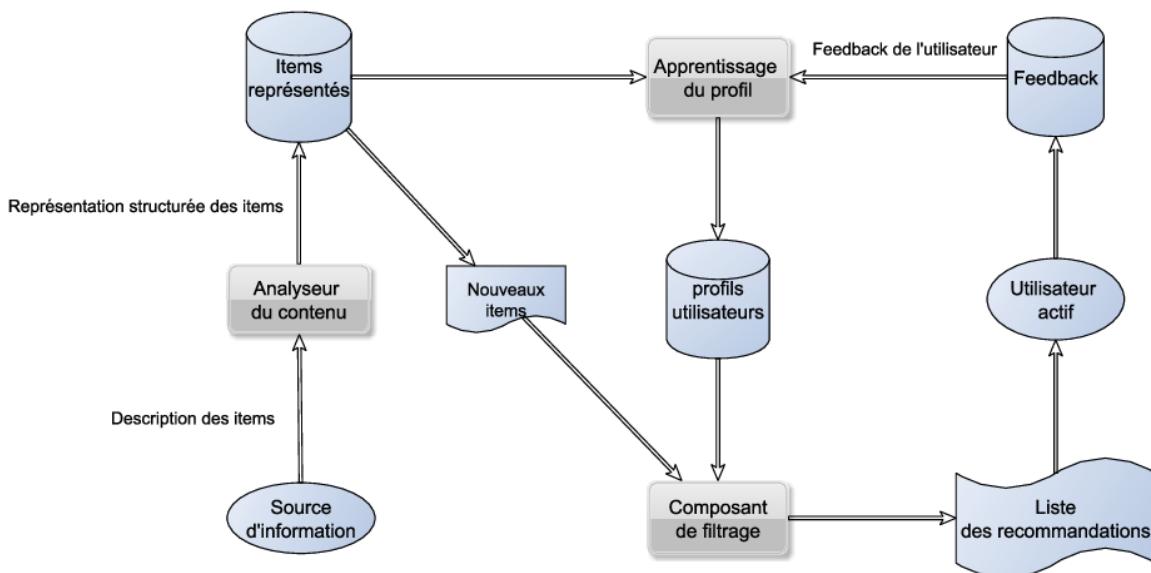
9. Exportation et Visualisation

- Exporter les données ou matrices pertinentes (par exemple, la matrice de probabilités) pour une utilisation dans une interface utilisateur ou une analyse ultérieure.
- Visualiser les communautés de produits ou les recommandations à l'aide d'outils tels que NetworkX ou matplotlib si nécessaire.

10. Déploiement et Test

- Implémenter le système de recommandation dans une interface utilisateur ou une application.
- Tester et valider les performances du système avec des interactions réelles ou simulées des utilisateurs.

Ces étapes décrivent le flux de travail typique pour construire un système de recommandation de produits basé sur les données d'achat des clients. Chaque étape peut impliquer une exploration de données supplémentaire, un réglage des paramètres ou une personnalisation en fonction des exigences commerciales spécifiques et des caractéristiques des données..



Chapitre 4

CI/CD Deployment (GitHub Actions, Docker Hub, Kubernetes)

L'intégration continue et le déploiement continu réunis s'appellent le pipeline CI/CD. Le pipeline CI/CD est une combinaison d'outils, de processus et de pratiques qui fonctionnent ensemble pour intégrer, tester et déployer en continu les changements de code en production.

Il vise à raccourcir les cycles de développement, à augmenter la fréquence des releases et à améliorer la qualité du code. Certaines des meilleures pratiques pour mettre en œuvre le pipeline CI/CD dans un environnement DevOps sont :

- Automatiser le processus de conception
- Utiliser les bons outils (Système de gestion de contrôle de source/ Serveur d'intégration continue)
- Tester tous les niveaux du code (y compris les tests unitaires, d'intégration et fonctionnels)
- Définir et appliquer les pratiques de revue de code
- Utiliser les technologies de virtualisation et de conteneurisation
- Surveiller et mesurez en permanence le pipeline CI/CD
- Intégrer la sécurité
- Avoir et utiliser un document d'orientation CI/CD Pipeline
- Améliorer et adapter en continu le pipeline CI/CD

Afin d'optimiser l'utilisation du pipeline CI/CD, il doit être intégré à d'autres composants du flux de travail DevOps et doit s'intégrer dans l'environnement DevOps global.

Un pipeline CI/CD qui fonctionne bien est essentiel pour un projet DevOps réussi et donc une gestion de projet DevOps. Le pipeline CI/CD permet à l'équipe DevOps de fournir des logiciels plus rapidement, plus fiables et avec moins de risques globaux.

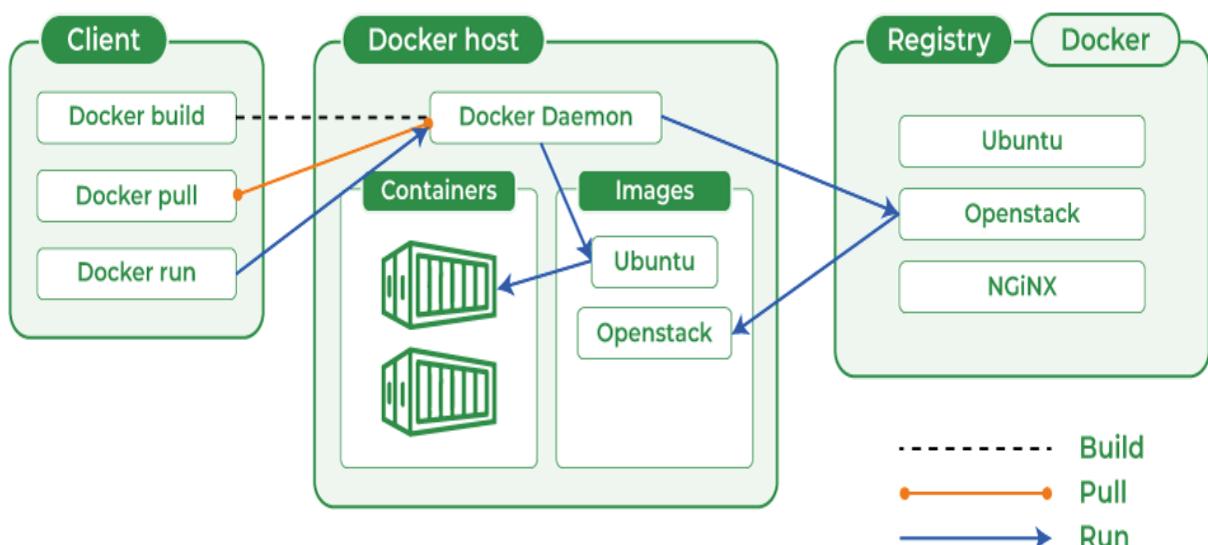


Qu'est-ce que c'est et comment l'utiliser ?

Docker est comme une boîte magique dans laquelle vous pouvez placer tous vos jouets (ou programmes) au même endroit afin qu'ils puissent bien jouer ensemble. Vous pouvez facilement déplacer cette boîte et la partager avec vos amis. Pour utiliser Docker, il vous suffit d'apprendre à mettre vos jouets (ou programmes) dans la boîte et à les sortir lorsque vous souhaitez jouer avec eux.

Docker est comme une boîte magique qui permet aux programmes informatiques de fonctionner de manière fluide et rapide. C'est comme avoir une petite maison spéciale dans laquelle chaque programme peut vivre, afin qu'ils ne se gênent pas mutuellement. Cela permet aux programmes de fonctionner mieux et plus rapidement, en particulier lorsqu'ils doivent augmenter ou diminuer leur taille.

Docker est un outil qui aide les utilisateurs à gérer et à utiliser facilement des conteneurs pour leurs applications. Il a été introduit en 2013 et est devenu très populaire car il est convivial et bénéficie du soutien d'un grand groupe de personnes qui travaillent ensemble pour l'améliorer. Avec Docker, vous pouvez créer, déployer et contrôler des applications dans des conteneurs sur différents systèmes d'exploitation comme Linux, Windows et macOS. La partie principale de Docker est le Docker Engine, qui est le programme principal que vous installez sur votre ordinateur pour créer, exécuter et gérer des conteneurs. Il existe deux versions de Docker Engine : l'une est gratuite et ouverte à tous, et l'autre est payante et pos-



sède des fonctionnalités supplémentaires telles que la surveillance des groupes de conteneurs et la recherche et la résolution des problèmes de sécurité.

Docker est comme une boîte à outils magique pour créer et exécuter des programmes sur votre ordinateur. Le démon Docker est comme le patron qui gère tout, comme les images, les boîtes et le stockage. Le client Docker est comme une baguette magique que vous pouvez utiliser pour dire au patron quoi faire. Le Dockerfile est comme une recette spéciale qui indique au patron comment créer un nouveau programme. Et les images Docker sont comme des boîtes spéciales contenant tout ce dont un programme a besoin pour fonctionner parfaitement.

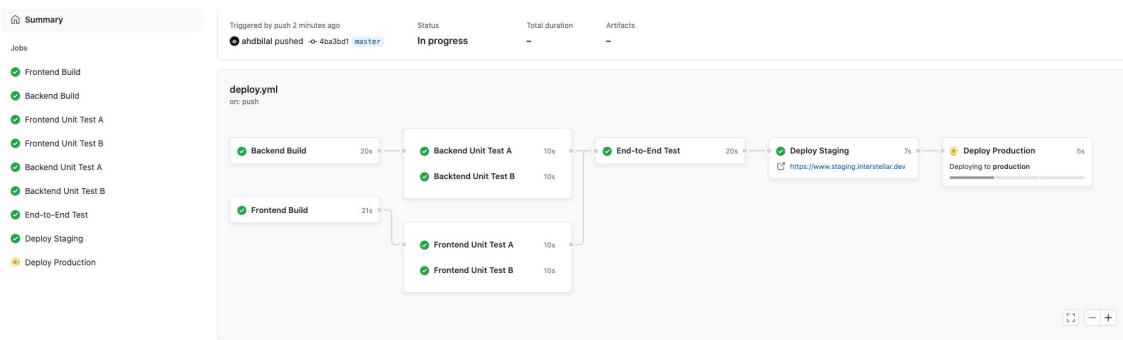
Dans GitHub Actions, Docker contribue au bon fonctionnement de tout en garantissant la cohérence du processus de création et de déploiement. Il permet de séparer les différentes parties du processus et facilite la réutilisation et la gestion des environnements de construction. Docker permet également que tout fonctionne de la même manière à chaque fois afin que les déploiements soient stables et fiables. Et en vérifiant les problèmes de sécurité avec les images Docker, nous pouvons nous assurer que tout reste en sécurité.

CiHub Actions – Construire, tester et déployer sans effort (CI/CD)



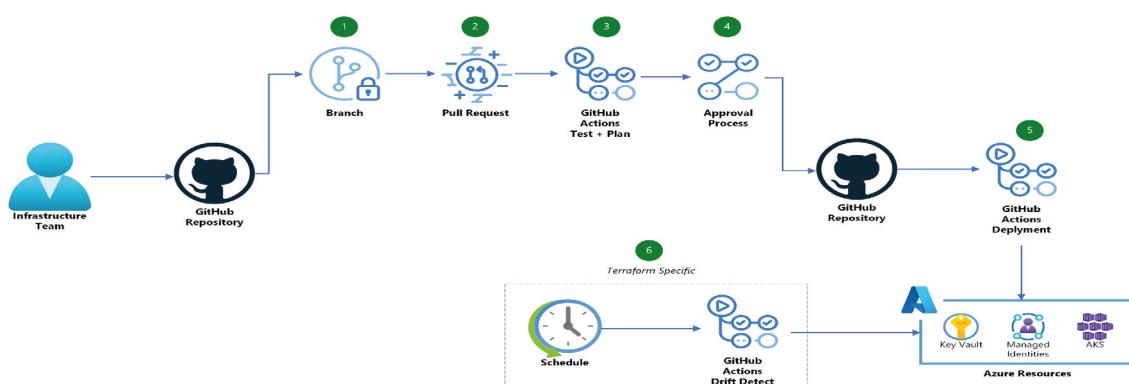
GitHub Actions

GitHub Actions est devenu un acteur très important dans la communauté Open Source et de manière générale dans le monde du développement. Depuis mon intégration à l'équipe Phalcon, j'utilise au maximum GitHub Actions pour construire mes projets, les tester et les déployer de manière automatique. C'est le concept du CI/CD (intégration continue / livraison continue)



Qu'est-ce que CiHub Actions ?

C'est un onglet / une partie accessible sur un projet géré / versionné par GitHub. permet de créer tout un workflow / processus. Vous pouvez en créer autant que vous le souhaitez par



projet. Ces processus se déclenchent lorsqu'un événement définit ce produit. Par exemple : lancer un processus de validation lorsqu'un push (envoi de code) est effectué.

Je suis sûr que vous commencez déjà à percevoir les avantages. En fonction du résultat du processus, vous pouvez bloquer ou non la réception d'une modification de code. Ce qui est très utile lorsque vous travaillez en équipe ou en communauté dans le cadre d'un projet Open Source.

Quels sont les systèmes d'exploitation gérés par GitHub Actions ?

Il s'agit d'un autre avantage majeur de GitHub. Vous avez la possibilité de compiler, tester et déployer votre code sur un ou plusieurs systèmes d'exploitation. GitHub Actions vous permet de lancer des machines Linux / Windows / MacOS pour faire de l'intégration en continu de manière automatique et sans coût supplémentaire de licence.

Vous pouvez aussi utiliser des images Docker pré-construites afin de vous faciliter encore plus la vie.

De plus, vous avez accès à un système de logs en temps réel, et détaillé pour débugger vos processus. GitHub propose aussi toute une marketplace de Workflows pré-développés en fonction du langage de votre projet, des outils tiers que vous utilisez...

L'utilisation des actions GitHub pour déployer des images Docker sur Docker Hub est un moyen automatisé puissant de gérer votre flux de travail CI/CD. Voici un aperçu détaillé de ce processus : Imaginons que Docker et GitHub Actions soient comme des outils spéciaux qui nous aident à faire les choses mieux et plus rapidement lorsque nous jouons avec nos jouets ou construisons des objets. Tout comme nous utilisons un marteau pour construire une maison, nous pouvons utiliser Docker et GitHub Actions pour que nos projets fonctionnent de manière fluide et simple. C'est comme avoir des pouvoirs magiques pour rendre nos jouets encore plus cool !

Docker est comme une boîte spéciale qui facilite l'utilisation des applications, tandis que GitHub Actions est comme une aide qui permet aux développeurs de partager et de mettre à jour leurs applications plus rapidement et plus facilement. Des choses que vous devez avoir ou faire avant de pouvoir faire autre chose.

1. **Dépôt GitHub** : le référentiel qui contient votre fichier Docker et tous les fichiers nécessaires à la création de votre image Docker.
2. **Compte Docker Hub** : le compte sur Docker Hub où votre référentiel Docker est créé.
3. **Secret GitHub** : pour des raisons de sécurité, vos informations d'identification Docker Hub (nom d'utilisateur et mot de passe) doivent être stockées en tant que secret dans votre référentiel GitHub

Configurer la façon dont GitHub effectue automatiquement les tâches.

Le fichier YAML indique à l'ordinateur quoi faire lorsqu'une nouvelle version est publiée sur GitHub. Il répertorie toutes les choses que l'ordinateur doit faire à chaque fois que cela se produit.

1. **Workflow Trigger** : Le workflow est déclenché par l'événement de publication de version (« version : publiée »).
2. **Exécution du travail** : Le travail `push_to_registry` s'exécute sur la dernière version d'Ubuntu.
3. **Autorisations** configure les autorisations requises, y compris l'écriture de packages et d'attestations.
4. **Extraire le référentiel** Le référentiel est cloné à l'aide de `actions/checkout@v4`.
5. **Connectez-vous à Docker Hub** : utilisez `docker/login-action@v2` pour vous connecter à Docker Hub à l'aide des secrets GitHub (`DOCKER_USERNAME` et `DOCKER_PASSWORD`).
6. **Extraction de métadonnées** utilise `docker/metadata-action@v3` pour extraire des métadonnées telles que des balises et des étiquettes.
7. **Créer et publier une image Docker** : utilise `docker/build-push-action@v2` pour créer une image Docker à partir d'un fichier Docker, puis la transfère vers Docker Hub. Les balises et les étiquettes sont appliquées à l'image.
8. **Générer la provenance des artefacts** : utilise `actions/attest-build-provenance@v1` pour générer une preuve d'artefact, augmentant ainsi la sécurité de la chaîne d'approvisionnement.

Voici quelques avantages de l'utilisation de cette automatisation.

- **Efficacité** : l'automatisation de la création et du déploiement d'images Docker permet de gagner du temps et de réduire les erreurs humaines.
- **Sécurité** : les informations sensibles sont stockées sous forme secrète, de sorte que les informations d'identification ne sont jamais exposées.
- **Cohérence** : le même processus est déclenché pour chaque build publiée sur GitHub, de sorte que les images Docker sont toujours créées et publiées de manière cohérente.
- **Traçabilité** : la génération de preuves d'artefacts fournit des informations vérifiables sur comment et où une image a été créée, offrant ainsi une couche de sécurité supplémentaire.

Configurer GitHub Secrets, c'est comme conserver un code secret pour vos informations

importantes afin que vous seul puissiez y accéder.

1. Accédez aux paramètres de votre référentiel GitHub.
2. Accédez à Secrets et variables > Actions.
3. Cliquez sur Nouveau secret de référentiel pour ajouter des secrets.
4. Ajoutez DOCKER_USERNAME et DOCKER_PASSWORD à vos informations d'identification Docker Hub.

En termes simples, la conclusion est comme la conclusion d'une histoire ou d'un film. C'est là que tout s'assemble et que nous apprenons les dernières pensées ou idées.

En utilisant GitHub Actions pour placer automatiquement vos images Docker sur Docker Hub, vous facilitez la mise à jour et le déploiement de vos applications par votre équipe. Cela permet de garder vos applications en sécurité et prêtes à être utilisées chaque fois que vous souhaitez publier une nouvelle version. Cela permet à votre équipe de développer de travailler plus rapidement et garantit que vos déploiements sont solides et sécurisés.

Exemple d'utilisation des actions github dans notre projet :

```
name: Django CI

on:
  workflow_call:
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest
    strategy:
      max-parallel: 4
      matrix:
        python-version: [3.7]

    steps:
    - uses: actions/checkout@v3
    - name: Set up Python ${{ matrix.python-version }}
      uses: actions/setup-python@v3
      with:
        python-version: ${{ matrix.python-version }}
    - name: Install Dependencies
      run: |
        python -m pip install --upgrade pip
        pip install -r requirements.txt
    - name: Run Tests
      run: |
        python manage.py test
```

Exemple d'utilisation github actions avec docker hub dans notre projet :

```
name: Build and Push Docker image to Docker Hub

on:
  workflow_call:
  workflow_dispatch:
  push:
    branches: [main]
  pull_request:
    branches: [main]

jobs:
  test_django_job:
    uses: ./github/workflows/django.yml
  push_to_registry:
    name: Push Docker image to Docker Hub
    runs-on: ubuntu-latest
    needs: [test_django_job]
    steps:
      - name: Check out the repo
        uses: actions/checkout@v3

      - name: Login to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

      - name: Build and push Docker image
        uses: docker/build-push-action@v4
        with:
          context: .
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/Casa:latest
```



kubernetes

c'est quoi et comment ça marche ?

Kubernetes, c'est quoi ?

Kubernetes ou K8S, est une solution d'orchestration de conteneurs. Plus précisément, c'est un logiciel d'orchestration qui permet avec des scripts de type YAML, de déployer une application, de la mettre à l'échelle automatiquement, et d'avoir plusieurs possibilités de déploiements dans des environnements différents.

Sur le web, nous retrouvons souvent Kubernetes sous l'acronyme K8S.

Pourquoi utiliser Kubernetes ?

Dans le monde du cloud, le DevOPS est de plus en plus utilisé pour automatiser les déploiements d'applications.

Le but premier de Kubernetes est de pouvoir déployer en quelques clics une infrastructure complète d'une application, en parallèle ou en remplacement d'un déploiement précédent.

De plus, Kubernetes permet de gérer la mise à l'échelle de cette infrastructure et donc, d'optimiser le rendu pour l'utilisateur final à la visite de l'application.

Quand vous templatez en YAML un script Kubernetes, vous pouvez lui dire

en fonction de la performance du Node ou du Pod en question : le scaler horizontalement ou sur un autre serveur.

Qu'est ce qui se cache derrière Kubernetes ?

Il faut savoir que Kubernetes est un logiciel d'orchestration et non un logiciel permettant le lancement de conteneurs ! En d'autres termes, sans un Docker ou un autre logiciel de ce type permettant de créer des conteneurs, Kubernetes ne sert à rien !

Docker, c'est quoi ?

Pour rappel, Docker est un logiciel de virtualisation permettant le lancement de conteneurs. On utilise Docker pour éviter d'avoir une adhérence système. En d'autres termes, en utilisant Docker, vous évitez que votre application soit dépendante des mises à jour et des problèmes du système (ex: Debian). Votre application est alors conteneurisée et donc, isolée de l'OS.

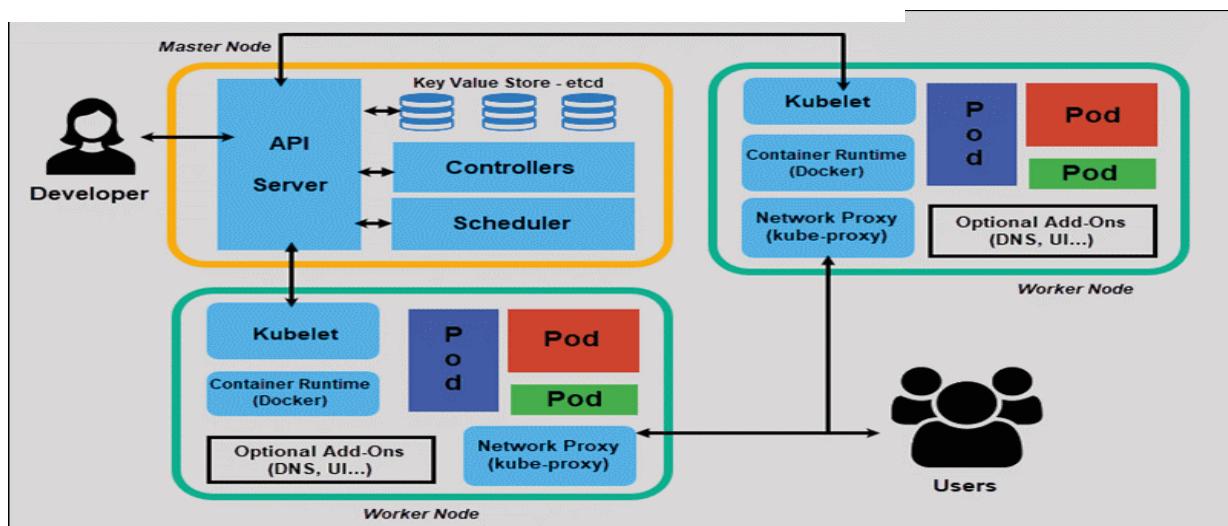
La nouvelle voie du DevOPS : faire des conteneurs pour éviter une adhérence système de l'application en déployant une application et ses librairies dans un container (une enveloppe) qui va être complètement abstrait de l'OS de base.

On peut tout aussi bien mettre Docker sur un Windows qu'une Debian, Ubuntu... du moment qu'il y a Docker, les containers seront toujours les mêmes !

Certes, un container est isolé de l'OS, mais il est aussi isolé du réseau ! De base, un container sur Docker n'a aucune adhérence système, c'est à dire qu'il ne va pas reprendre l'IP du système. Il aura une IP interne qu'on pourra exposer sur l'extérieur, ce qui sécurise un peu l'infrastructure.

Kubernetes, comment ça marche ?

Voici le schéma général de l'architecture de Kubernetes que vous retrouverez dans la plupart des projets. Il est composé de deux grandes parties : le master et le worker (ou node).



Nos configuration k8s dans notre project

..		
configmap.yaml	Casa shop	2 months ago
deployment.yaml	Casa shop	2 months ago
ingress.yaml	Casa shop	2 months ago
service.yaml	Casa shop	2 months ago

description

Ce fichier YAML définit une ConfigMap Kubernetes appelée `Casa-configmap`, qui stocke des configurations essentielles sous forme de paires clé-valeur. Il s'agit d'une ressource de type ConfigMap (indiquée par `kind: ConfigMap`), versionnée selon l'API `v1`. Le `meta-data` fournit des informations de base, incluant le nom unique `Casa-configmap` dans

son namespace. La section `data` contient les adresses de services critiques de l'application, chacune mappée à un port spécifique : `AUTH_SVC_ADDRESS` pointe vers `auth` sur le port `7777`, `POLICY_SVC_ADDRESS` vers `policy` sur le port `9999`, `PRODUCT_SVC_ADDRESS` vers `product` sur le port `5555`, `SEARCH_SVC_ADDRESS` vers `search` sur le port `5550`, et `CATEGORY_SVC_ADDRESS` vers `category` sur le port `5500`.

Ces configurations permettent aux pods de récupérer dynamiquement les adresses des services sans nécessiter de modifications du code de l'application. Lors de l'utilisation, cette ConfigMap peut être montée dans un pod, exposant ses valeurs comme variables d'environnement. Par exemple, dans un pod Kubernetes, ces variables peuvent être référencées via `valueFrom` et `configMapKeyRef`, permettant ainsi à l'application de communiquer avec les services configurés via des noms d'hôtes et des ports spécifiques, sans que ces détails soient codés en dur dans l'application.

Ce fichier YAML configure un déploiement Kubernetes pour l'application «Casa», comprenant un Deployment, un Service et un Ingress. Le Deployment, nommé `Casa-deployment`, déploie un pod unique utilisant l'image Docker `Casa`. Les configurations environnementales sont importées d'une ConfigMap nommée `Casa-configmap`, et le conteneur expose le port 8000. Le Service, `Casa-service`, de type LoadBalancer, expose le port 80 et redirige le trafic vers le port 8000 du conteneur. Enfin, l'Ingress, `Casa-ingress`, définit des règles pour acheminer le trafic HTTP de `www.Casa.com` vers le Service `Casa-service`, avec des annotations pour configurer le comportement du proxy NGINX, comme les temps d'attente et la taille maximale du corps de la requête. Ce fichier facilite le déploiement, la gestion et l'exposition de l'application «Casa» dans un cluster Kubernetes.

Conclusion

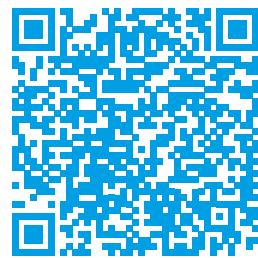
Conclusion

En résumé, Casa Shop illustre l'intégration réussie de l'architecture de microservices et des technologies avancées de Deep Learning dans le domaine du commerce électronique. Ce projet a démontré comment une approche modulaire permet une gestion efficace des opérations, tout en intégrant des fonctionnalités innovantes telles que le traitement du langage naturel pour optimiser le filtrage et les recommandations de produits.

L'implémentation de Django, Flask, et FastAPI pour différents aspects de la plateforme a permis de répondre de manière ciblée aux besoins de gestion des produits, d'authentification des utilisateurs, et de recherche avancée de produits par catégorie. Chaque composant a été conçu pour fonctionner de manière autonome tout en contribuant à une expérience utilisateur fluide et sécurisée.

Ce projet a également souligné l'importance cruciale de l'innovation technologique dans un environnement concurrentiel. Casa Shop, avec son infrastructure robuste et ses fonctionnalités avancées, est positionné pour continuer à évoluer et à prospérer dans un paysage numérique dynamique. Il représente un modèle inspirant pour les entreprises cherchant à adopter des technologies de pointe pour une croissance stratégique et une adaptation rapide aux exigences du marché.

En conclusion, Casa Shop incarne l'excellence dans l'application des technologies modernes au service du commerce électronique, ouvrant la voie à de nouvelles possibilités et à des standards élevés pour l'industrie.



Année universitaire : 2023-2024