Projet E5FI Architecture et Micro services

Artonys, 5 micros services Docker dédiés à la vente de matériel de BTP en ligne.

1^{er} micro-service : Gestion de Produit :

Conteneur 1 : db_gestion_produit, langage MySQL, script création BDD, tables catégories et produit pour gérer le catalogue de produit.

Conteneur 2 : api produit, API Rest en Python Flask (test pour récuperer un produit en json :

http://localhost:9006/api/gestion_produit/getProductByName?name=Fix

Web: http://localhost:850/ProductResearch.php

2^e micro-service : Authentification:

Conteneur 1: auth_db, langage MySQL, script création BDD, table client (pw et login)
Conteneur 2: api_auth, API Rest en Python Flask, test de connexion pour l'user goya:
http://localhost:9007/api/authentification/authentificate?login=goya&password=azerty

Web: http://localhost:850/sign_in.php

3^e micro-service : Gestion_client:

Conteneur 1 : db_gestion_client, langage MySQL, script création BDD, table client (infos pour enregistrer le client)

Conteneur 2 : api_client , API Rest Python Flask, test d'affichage de tous les clients :

http://localhost:9008/api/gestion_client/getAllClient

Web: http://localhost:850/create account.php

4^e micro-service : Historique_commande

Conteneur 1 : db_historique_commande, langage MySQL, script création BDD, table commandes Conteneur 2 : api_historique_commande , API Rest Python Flask, test d'affichage de tous les clients : http://localhost:9009/api/historiqueCommande/getOrderInfo?clientId=1&orderId=3 (Bad request error, non investiguée faute de temps, mais semble minime)

Web: http://localhost:850/historiqueCommande.php (PHP incomplet faute de temps)

5^e micro-service : Paiement

 ${\tt Conteneur~1:db_paiement, langage~MySQL, script~cr\'{e}ation~BDD, table~creditcard}$

Conteneur 2 : api_paiement, API Rest Python Flask, test d'affichage de tous les clients :

http://localhost:9010/api/paiement/getCardFromClientId?clientId=1

Web: http://localhost:850/historiqueCommande.php (PHP incomplet faute de temps)

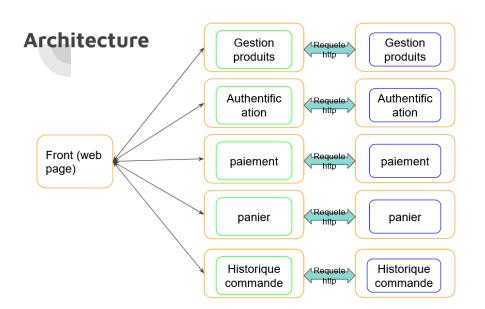
Front:

Conteneur 1 : php avec image: bitnami/php-fpm:latest, pour executer les fichiers php transmis par le conteneur apache

Conteneur 2 : serveur apache avec image: bitnami/apache:latest

Page d'accueil: http://127.0.0.1:850/index.php





Conteneur Docker

Base de données

API