Sneakers-actu

Performance Test

4IW2

Version 0.1

28/06/2022

Samy Hamed-e-saberi

Théo Sigaud

Mohamed Kajeiou

Waseem Nassuraly

1. Préambule

Ce dossier a pour but de livrer un plan précis et détailler de test de performance en utilisant un scénario fidèle à la réalité.
Avant d'établir le scénario, il est important de parler du projet/site que nous allons tester.

Dans ce test de performance, nous allons réaliser nos tests sur Sneakers-actus. Sneakers-actus sont un blog dédié aux sneakers. Ce blog est revendiqué comme « l'encyclopédie de la culture sneakers ». Le type d'utilisateurs est une population assez jeune (15-30 ans) et susceptible et naviguer pendant une courte durée.

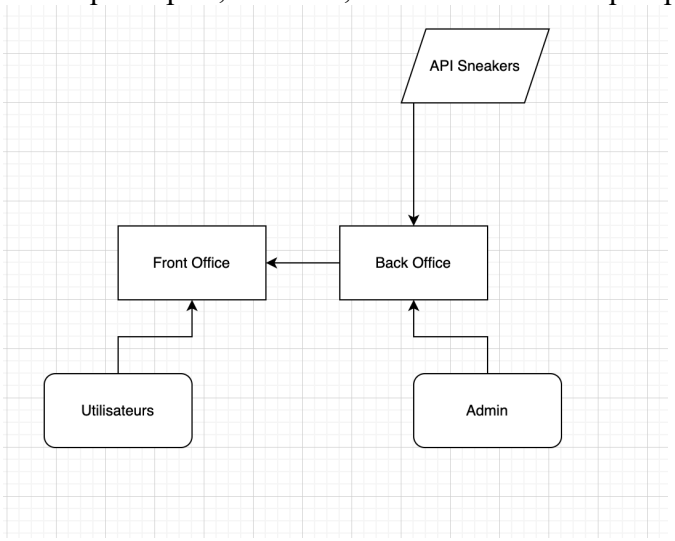
2. Architecture de l'application

Étant données que nous n'avons pas eu la possibilité d'entrer en contact avec le propriétaire du blog, nous avons utilisé un site (builtwith.com) permettant d'obtenir la liste des stacks utilisées.

Partons du postula que les informations récupérer à partir de ce site soit exact :

- Docker
- Php avec le framework Symfony
- Twig

Concernant les services utilisés, une api est utilisé afin d'obtenir toutes les informations de base concernant une sneakers tel que le prix, son nom, la date de sortie et quelques photos.



L'environnement de prod est composé d'un conteneur sous Docker.

3. Exigences du test

Business Transactions	**User Load**	**Response Time**	**Transactions per hour**
Acces login admin page	5	<1	<2
Access sneakers page	200	1	500

4. Environnement de test

L'environnement de test est hébergé sur un vps OVH.

CPU	1 vCore
Mémoire	2 Go
OS	Linux debian
Stockage	40 Go SSD
Bande passante	250 Mbit/s

Environnement de prod :

CPU	2 vCore
Mémoire	4 Go
OS	Linux debian
Stockage	80 Go SSD
Bande passante	500 Mbit/s

Comme nous pouvons le constater, l'environnement de prod possède une architecture 2 fois plus puissante que celle de l'environnement de test. Nous pouvons donc admettre que leur coefficient est de 0.5.

5. Planification des tests

La planification des tests vont nous permettre de déterminer les métriques que nous allons surveiller et les critères de réussite de nos tests.

Étant données que notre site possède un certain nombre de trafics sur une petite durée (lors de la sortie d'un pair ou autre événement telle que les soldes), le type de tests le plus approprié sera donc réalisé nos tests grâce au **Load testing**.
Les métriques que nous souhaitons surveiller sont les temps de réponse et le taux d'erreurs.

Nous pourrons déterminer que le test sera concluant et réussi si le temps de réponse sera le plus faible possible.

6. Étapes des tests

Step #	**Business Process Name : Product Ordering**
1	Home Page
2	"Les marques" Page
3	Sélection de la marque de sneakers
4	Sélection d'une catégorie de sneakers
5	Sélection d'une paire de sneakers

7. Exécution des tests

Cycle d'exécution :

Test	Scénario
Cycle 1 - #1	Load Test – 2 Heures
Cycle 1 - #2	Réitération du Load Test – 1 heures
Cycle 2 - #1	Load Test – 2 Heures
Cycle 2 - #2	Réitération du Load Test – 1 heures
Cycle 3 - #1	Load Test – 2 Heures
Cycle 23- #2	Réitération du Load Test – 1 heures

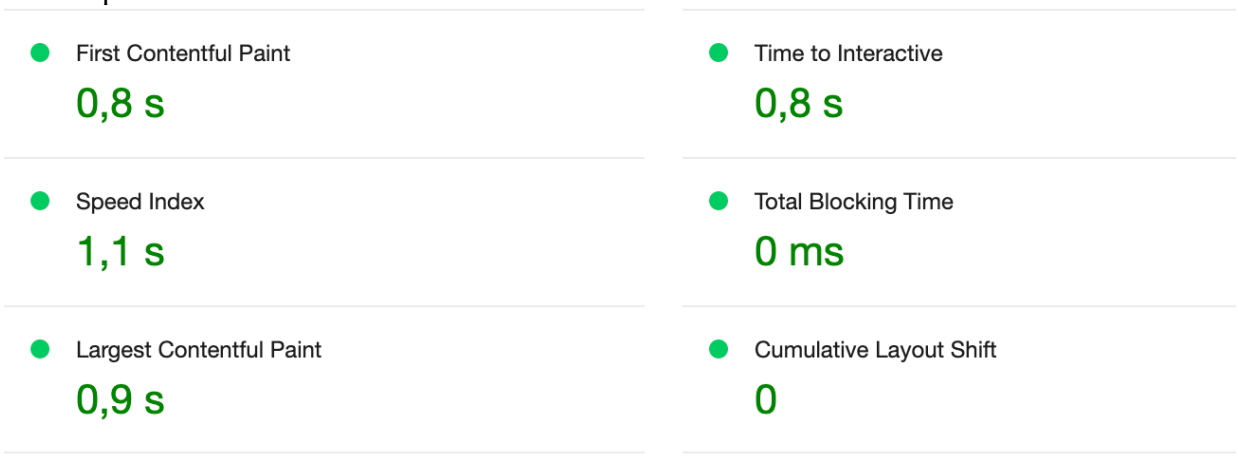
Load test – explication :

	Détail
But	Le but de ce test va être de surcharger le site dans le but de connaitre son point de rupture. Le système devra maintenir son temps de réponse pendant les différentes phases de surcharge.
Nombre de test	6 (2 par cycle)
Durée	Ramp-Up : 5VUser / 15s Steady : 200VUser – 60 minutes Ramp-Down : 20VUser/ 15s
Nom du scénario	Load test
User Load / Volume	200 VUser
Critère de réussite	• Temps de réponse moyen (Average latency)< 3 secondes • Taux d'erreur des transactions < 10% • Throughput

8. Résultats des tests

Score lighthouse :

Statistique :



Résultat Végéta :