



JMeter

Test de Performance SOSIE 2

INSTALLATION DE JMeter

<http://jmeter.apache.org/>

Aller sur [Download Releases](#)



About

- Overview
- License

Download

- Download Releases
- Release Notes

Documentation

- Get Started
- User Manual
- Best Practices
- Component Reference
- Functions Reference
- Properties Reference
- Change History
- Javadocs
- JMeter Wiki
- FAQ (Wiki)

Tutorials

- Distributed Testing
- Recording Tests
- Unit Sampler

Apache JMeter™

[Twitter](#)[github](#)

The **Apache JMeter™** application is open source software, a 100% pure Java application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

What can I do with it?

Apache JMeter may be used to test performance both on static and dynamic resources, Web dynamic applications. It can be used to simulate a heavy load on a server, group of servers, network or object to test its strength or to analyze overall performance under different load types.

Apache JMeter features include:

- Ability to load and performance test many different applications/server/protocol types:
 - Web - HTTP, HTTPS (Java, NodeJS, PHP, ASP.NET, ...)
 - SOAP / REST Webservices
 - FTP
 - Database via JDBC
 - LDAP
 - Message-oriented middleware (MOM) via JMS
 - Mail - SMTP(S), POP3(S) and IMAP(S)
 - Native commands or shell scripts
 - TCR

Puis sur Binares pour télécharger l'archive JMeter

• [FAQ \(Wiki\)](#)

Tutorials

- Distributed Testing
- Recording Tests
- Unit Sampler
- Access Log Sampler
- Extending JMeter

Community

- Issue Tracking
- Security
- Mailing Lists
- Source Repositories
- Building and Contributing
- Project info at Apache
- Contributors

Foundation

- The Apache Software Foundation (ASF)
- Get Involved in the ASF
- Sponsorship
- Thanks

KEYS

Apache JMeter 3.3 (Requires Java 8. JMeter 3.3 does not yet support Java 9.)

Binaries

[apache-jmeter-3.3.tgz md5 sha512 pgp](#)
[apache-jmeter-3.3.zip md5 sha512 pgp](#)

Source

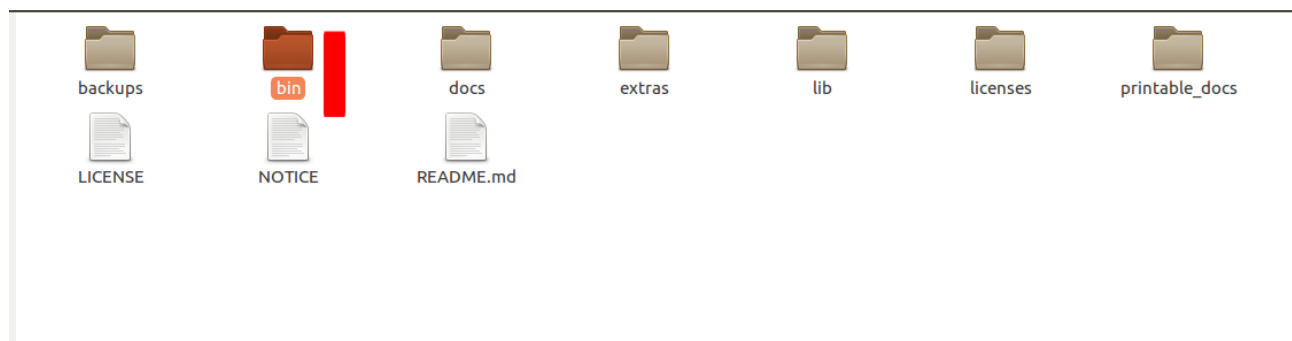
[apache-jmeter-3.3_src.tgz md5 sha512 pgp](#)
[apache-jmeter-3.3_src.zip md5 sha512 pgp](#)

Archives

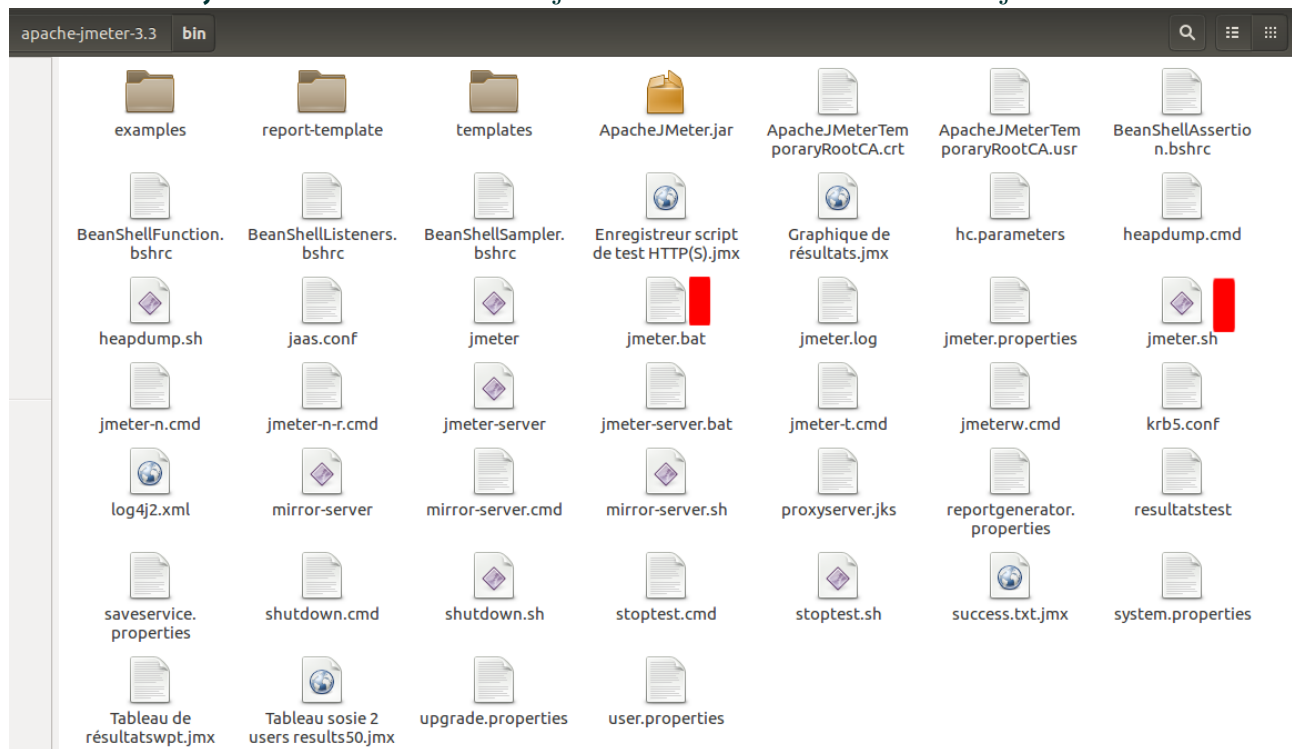
Older releases can be obtained from the archives.

Décompresser l'archive

Dans bin, vous trouverez l'exécutable permettant de lancer JMeter



Sous Linux, exécuter le fichier `jmeter.sh` et sous windows `jmeter.bat`

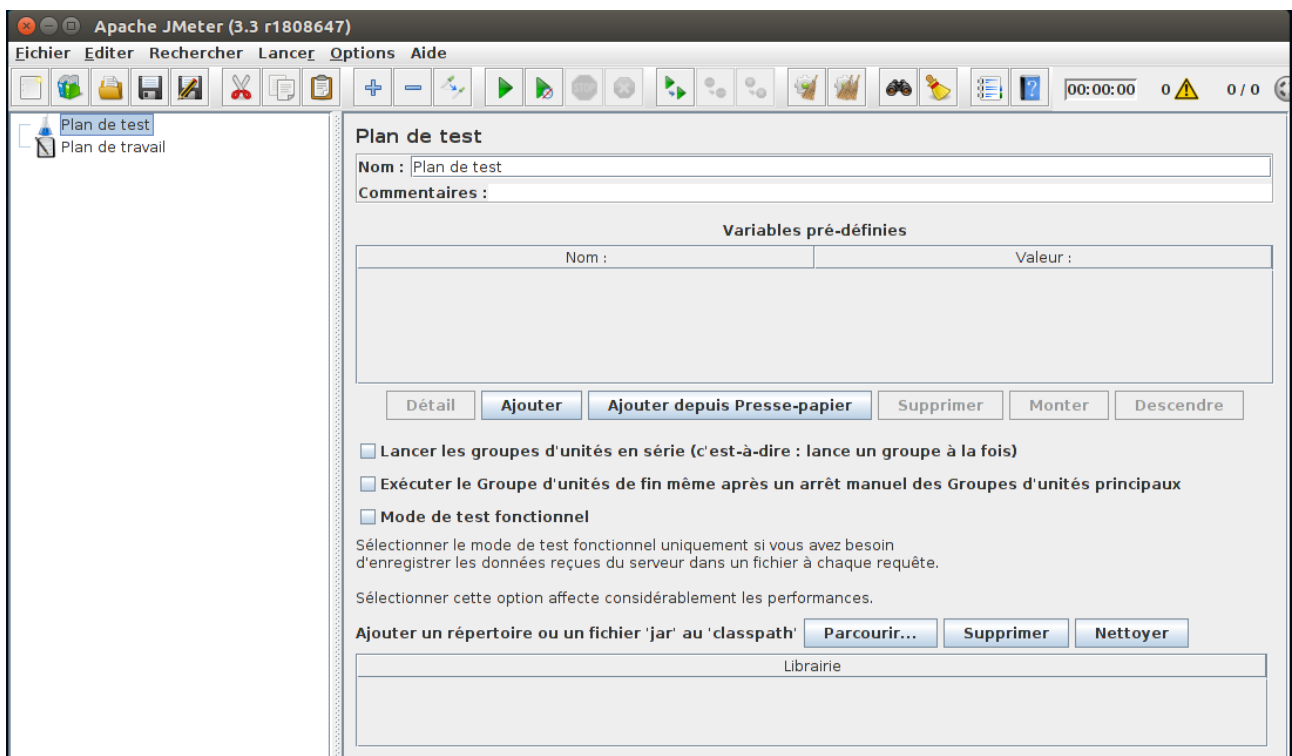


```

vaadl@vaadl-Satellite-P745:~$ cd /usr/local/apache-jmeter-3.3/bin/
vaadl@vaadl-Satellite-P745:/usr/local/apache-jmeter-3.3/bin$ ./jmeter.sh
=====
Don't use GUI mode for load testing, only for Test creation and Test debugging !
For load testing, use NON GUI Mode:
    jmeter -n -t [jmx file] -l [results file] -e -o [Path to output folder]
& adapt Java Heap to your test requirements:
    Modify HEAP="-Xms512m -Xmx512m" in the JMeter batch file
=====

```

L'IHM

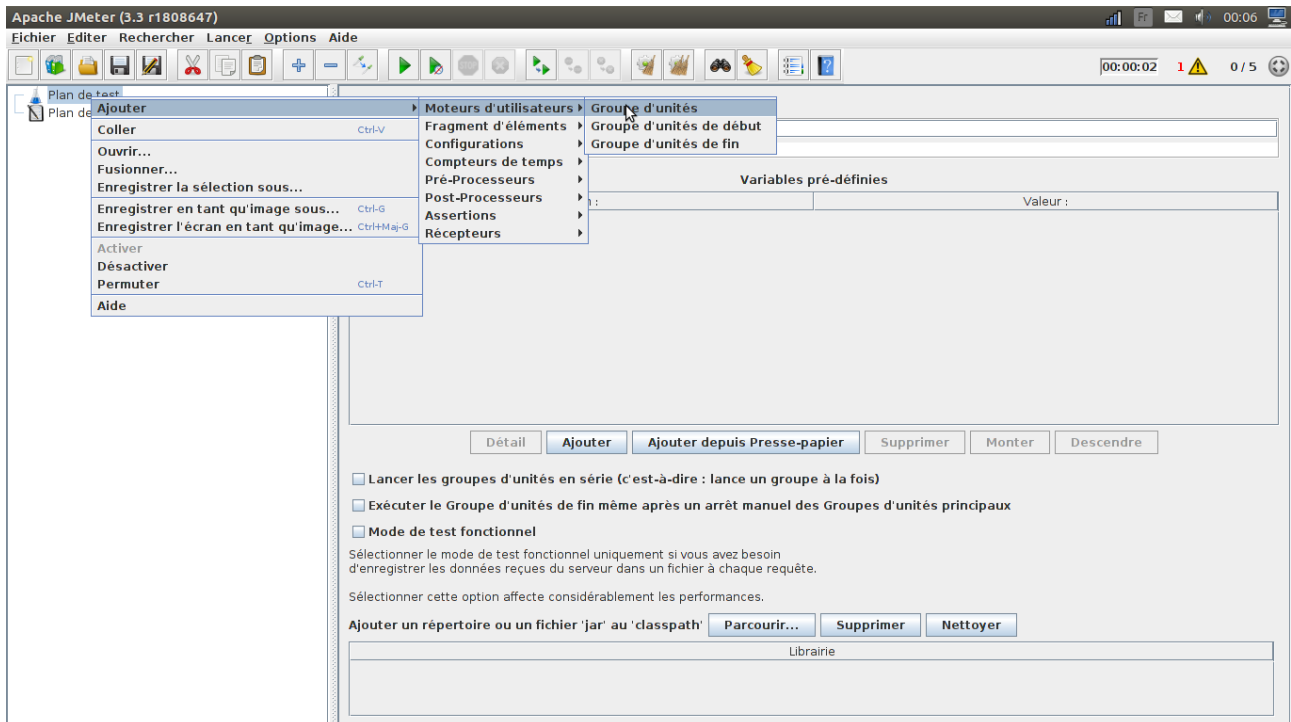


Sur l'IHM nous avons deux items: un plan de test dans lequel on stocke les différentes étapes du test et un plan de travail pouvant stocker des éléments constitutifs du test.

PLAN DE TEST SOSIE 2

Le plan de test est **une série d'étape** que JMeter va exécuter lorsqu'on va lancer le test.

Dans notre plan de test on va ajouter **un groupe d'unités** c'est-à-dire le **nombre d'utilisateurs** qui vont être simulé par le serveur lors du test.



Dans notre groupe d'unités on spécifiera **le nombre d'itérations** et ainsi qu'une **durée de montée en charge**, c'est **temps sur lequel nous allons répartir nos utilisateurs**.

On renomme le groupe d'unités «Sosie 2 users»

Eichier Editer Rechercher Lancer Options Aide

Plan de test
Groupe d'unités
Plan de travail

Groupe d'unités

Nom : Sosie 2 users

Commentaires :

Action à suivre après une erreur d'échantillon

☒ Continuer ☐ Démarrer itération suivante ☐ Arrêter l'unité ☐ Arrêter le test ☐ Arrêter le test immédiatement

Propriétés du groupe d'unités

Nombre d'unités (utilisateurs) : 10

Durée de montée en charge (en secondes) : 5

Nombre d'itérations : ☐ Infini 1

☐ Créer les unités seulement quand nécessaire

☐ Programmeur de démarrage

Configuration du programmeur

Durée (secondes) :

Délai avant démarrage (secondes) :

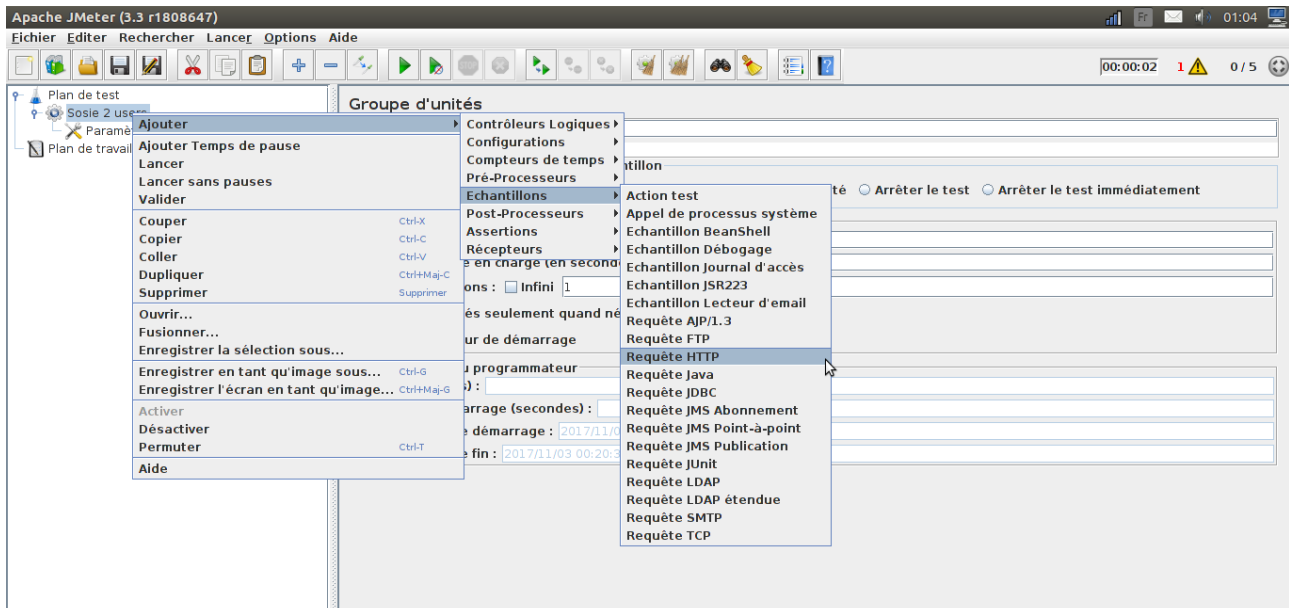
Date et heure de démarrage : 2017/11/03 00:20:36

Date et heure de fin : 2017/11/03 00:20:36

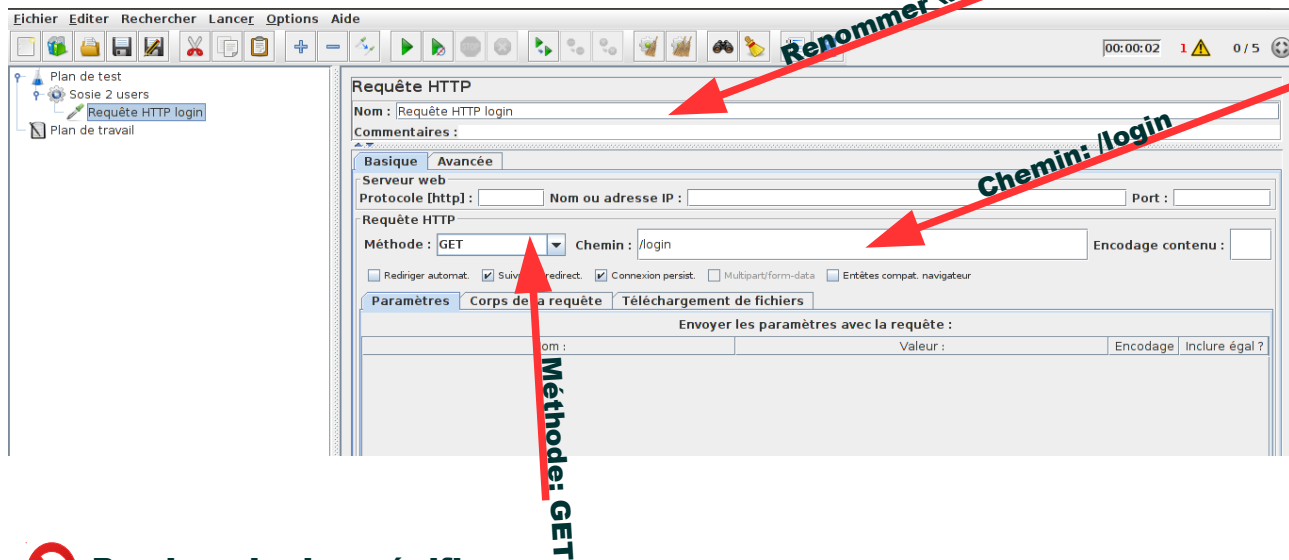
Temps : 5 s
Utilisateurs : 10
Itérations : 1

ÉCHANTILLONS DE REQUÊTES HTTP

Dans le groupe d'unités, nous ajoutons nos échantillons de requêtes http



Échantillon 1: /login

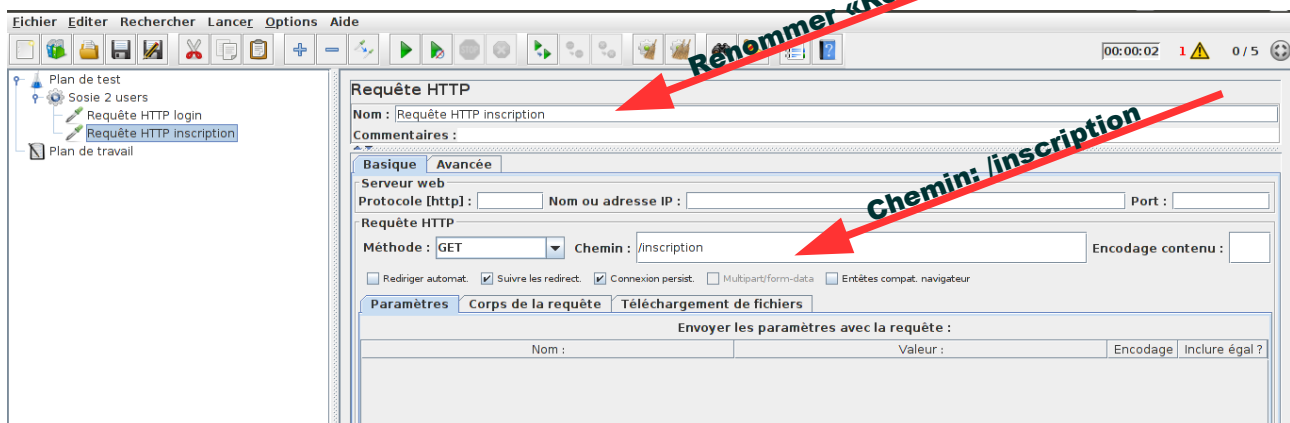


Pas besoin de spécifier:

- ➔ **Protocole**
- ➔ **Nom ou adresse IP (nom du domaine)**
- ➔ **Port**

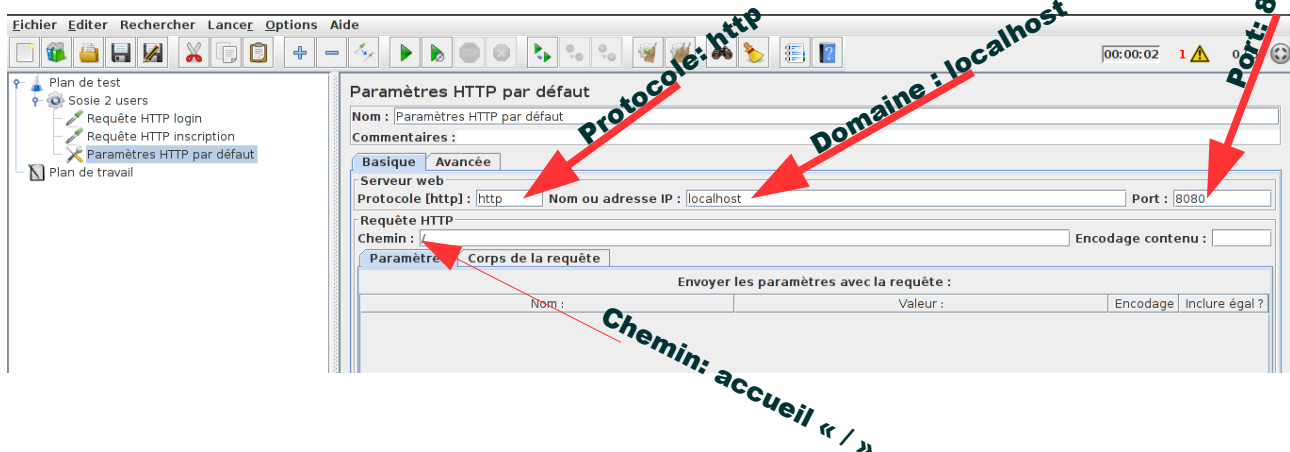
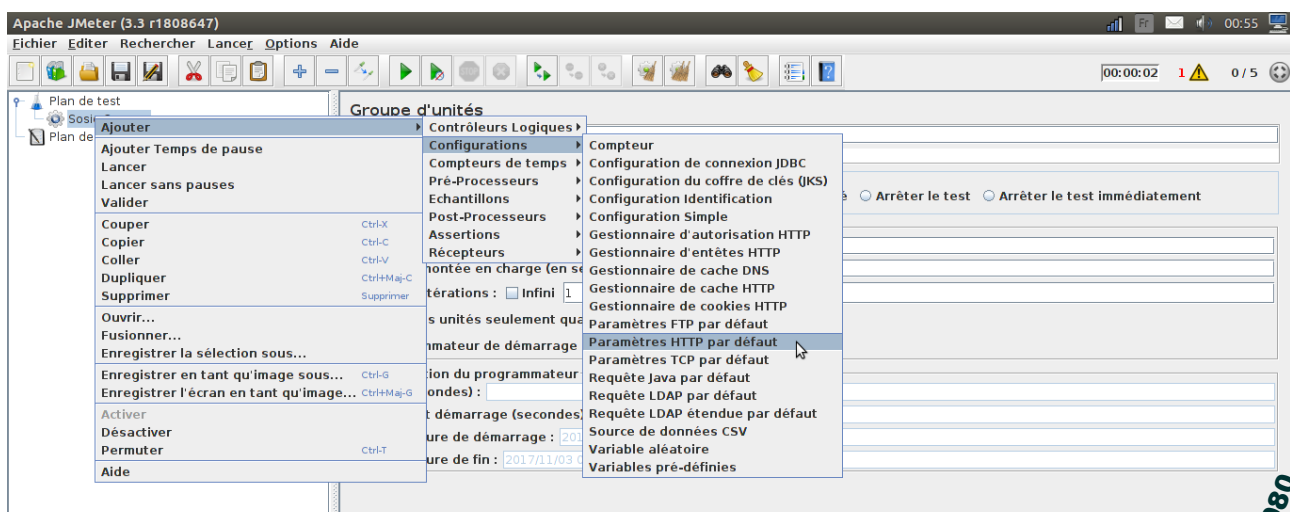
On les déterminera dans les paramètres http par défauts

Échantillon 2 : /inscription



PARAMÈTRES HTTP PAR DÉFAUT

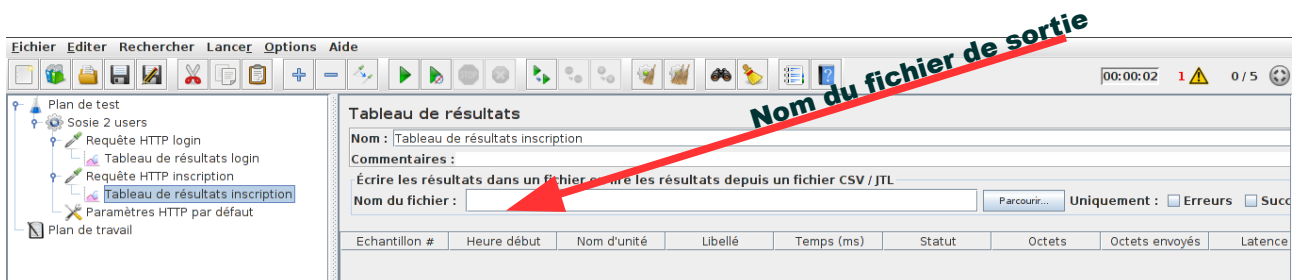
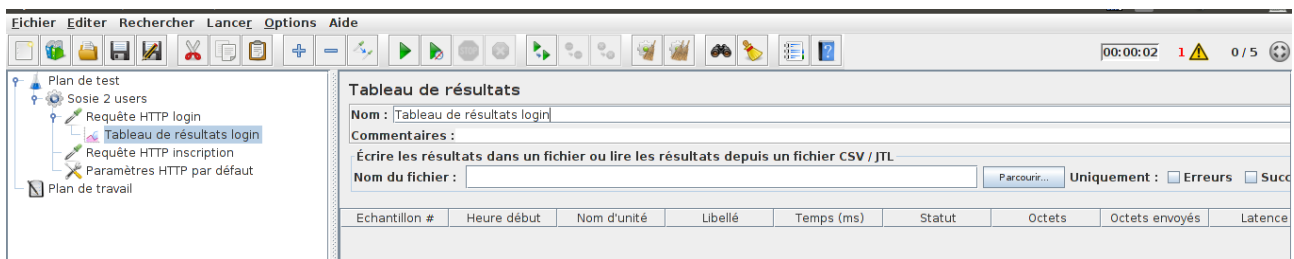
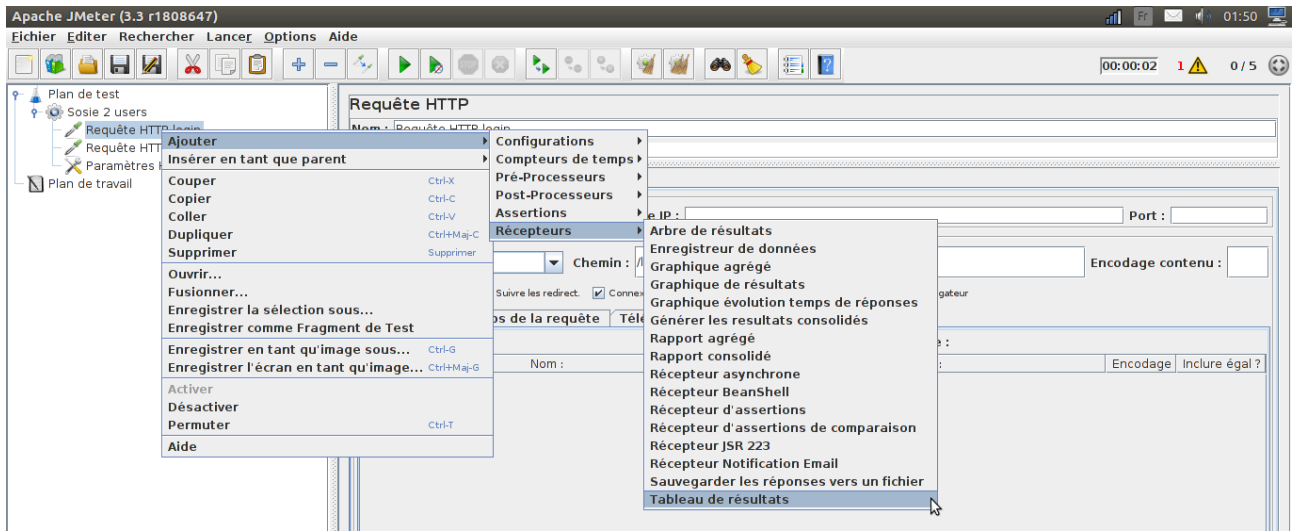
Ajout d'une configuration par défaut pour les requêtes http dans laquelle nous spécifions le nom du domaine à tester et le port, ce qui va nous éviter de les répéter dans les échantillons de requêtes.



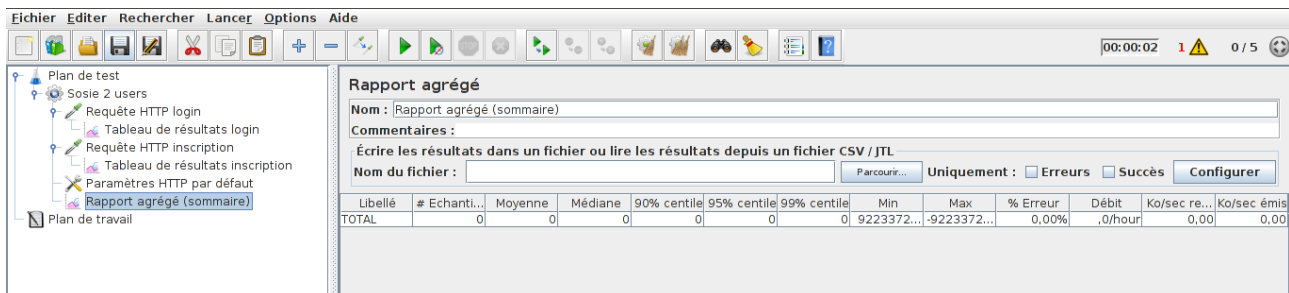
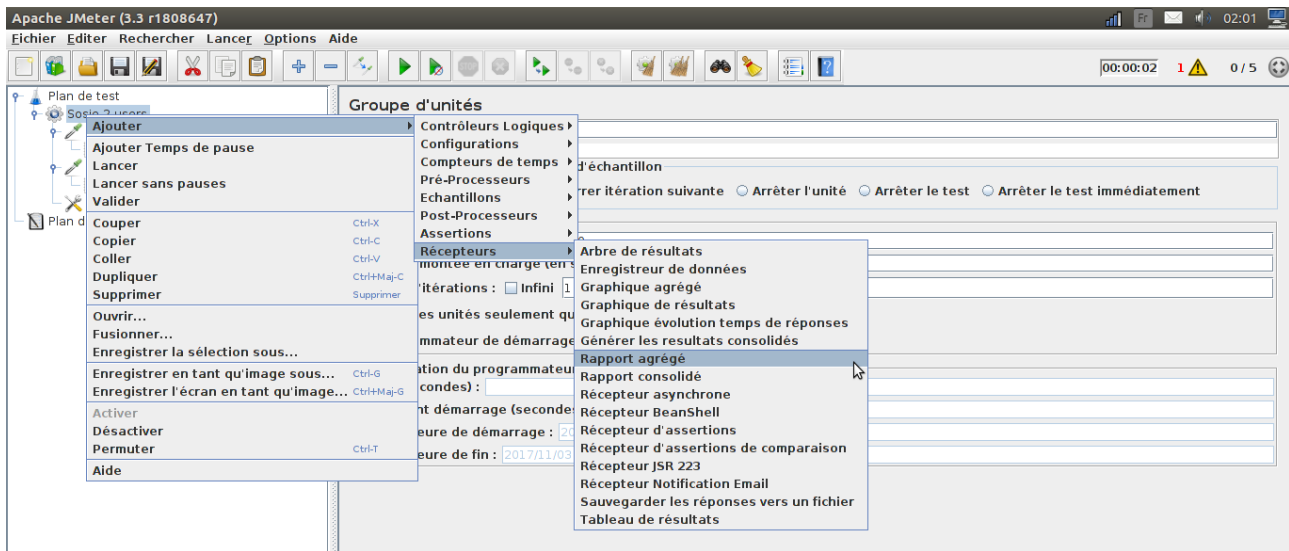
RÉCEPTEURS

Les récepteurs sont les éléments qui vont nous permettre d'observer les résultats de notre test.

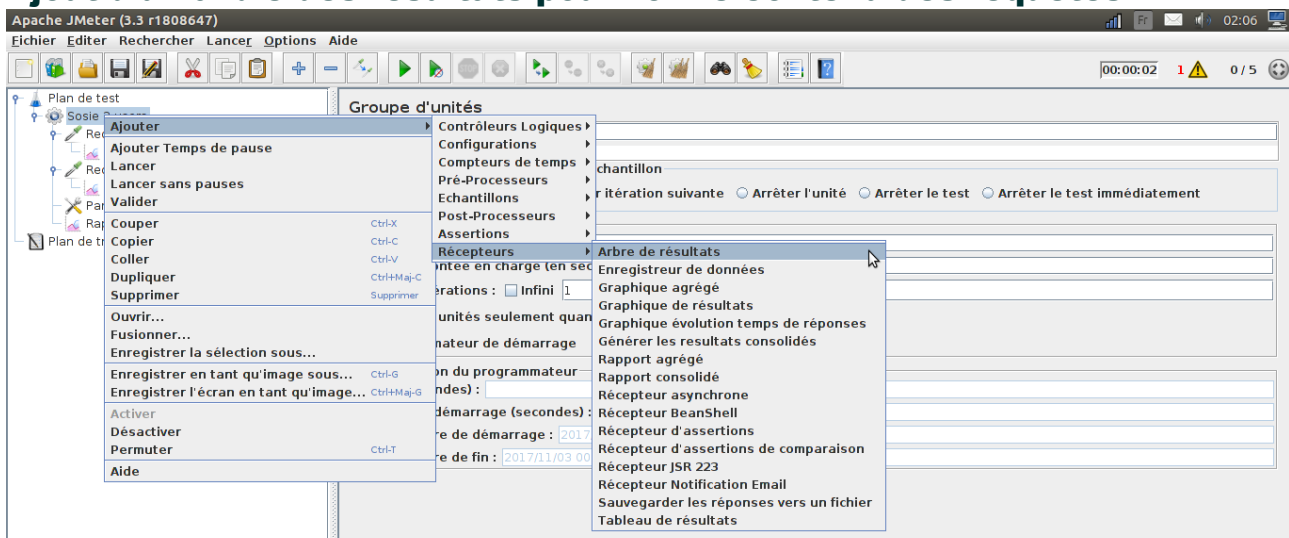
On ajoute un récepteur à chaque échantillon.

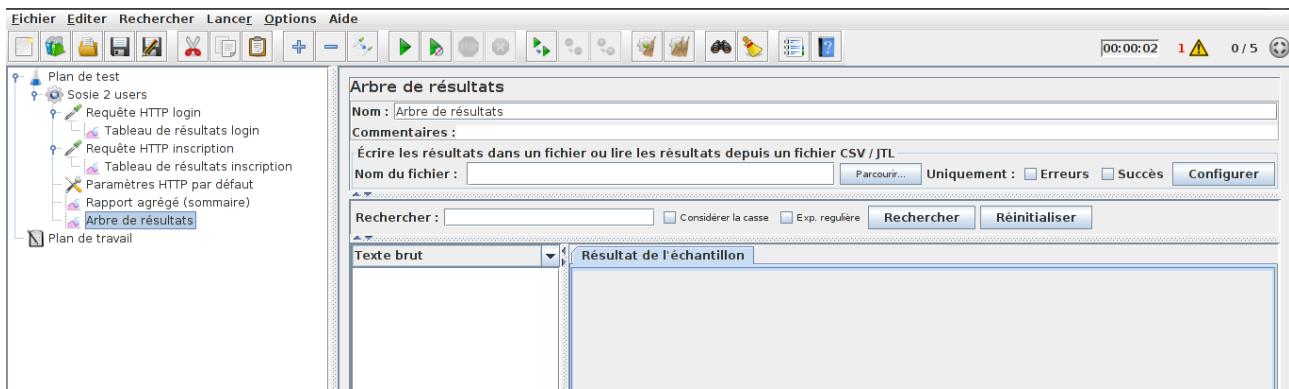


Toujours au niveau des récepteurs, Ajout d'un rapport abrégé ou rapport sommaire

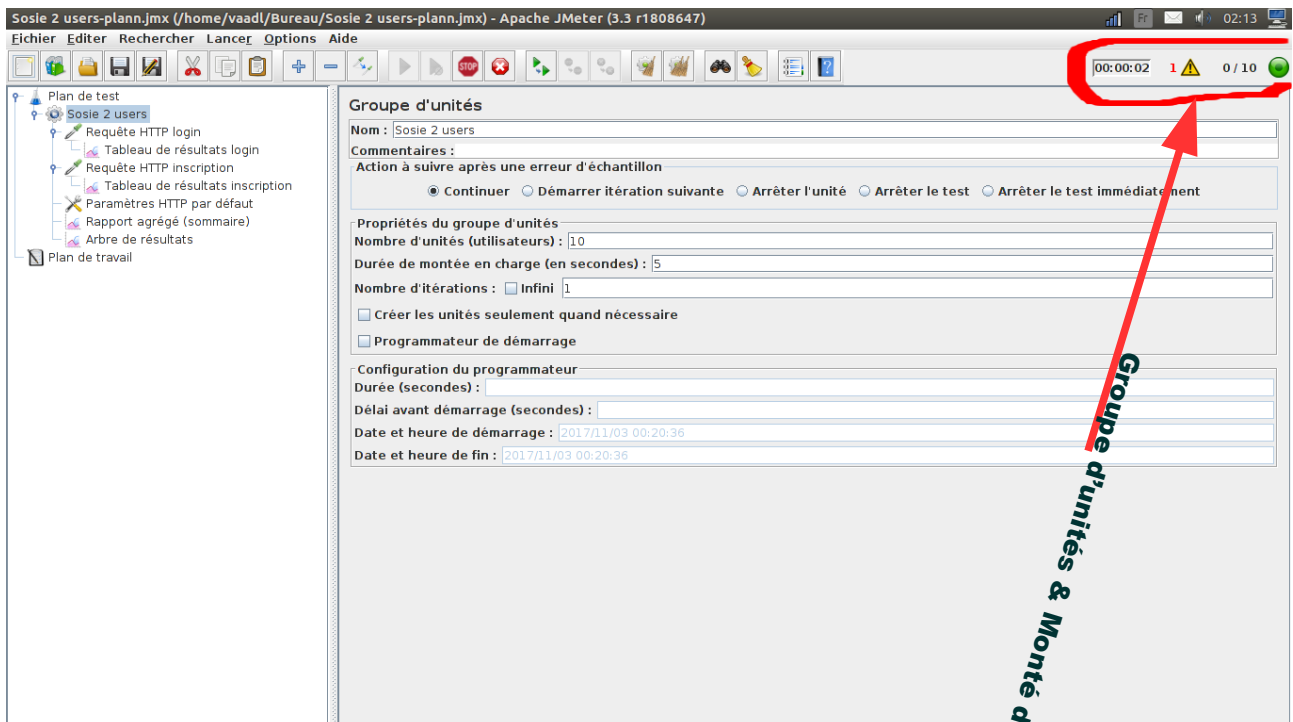
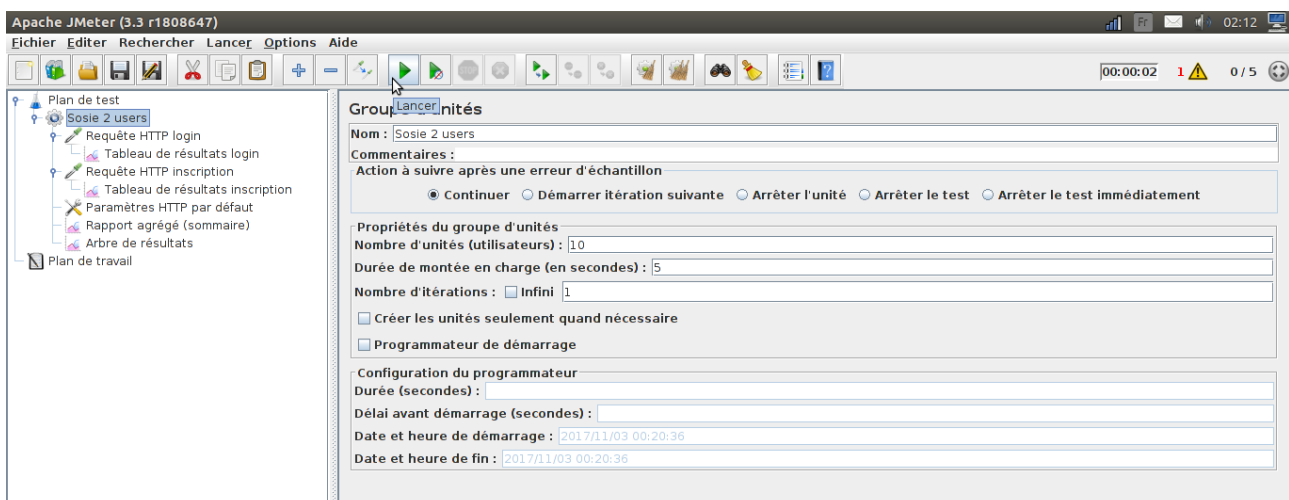


Ajout d'un arbre des résultats pour voir le contenu des requêtes





EXÉCUTION DU TEST



ANALYSE DES RÉSULTATS

Tableaux de résultats

Echier Editer Rechercher Lancer Options Aide

00:00:04 1 0 / 10

Plan de test

- Sosie 2 users
 - Requête HTTP
 - Tableau de
 - Requête HTTP
 - Tableau de**
 - Paramètres HT
 - Rapport agrégé
 - Arbre de résultat
- Plan de travail

Tableau de résultats

Nom :

Commentaires :

Écrire les résultats dans un fichier ou lire les résultats depuis un fichier CSV / JTL

Nom du fichier : Parcourir... Uniquement : ☐ Erreurs ☐ Succès Configurer

Echantillon #	Heure début	Nom d'unité	Libellé	Temps (ms)	Statut	Octets	Octets envoyés	Latence	Établ. Conn.(ms)
1	02:13:24.259	Sosie 2 users 1-1	Requête HTTP ...	23	✓	4148	129	23	0
2	02:13:24.721	Sosie 2 users 1-2	Requête HTTP ...	9	✓	4148	129	9	0
3	02:13:25.224	Sosie 2 users 1-3	Requête HTTP ...	14	✓	4148	129	14	0
4	02:13:25.727	Sosie 2 users 1-4	Requête HTTP ...	15	✓	4148	129	15	0
5	02:13:26.225	Sosie 2 users 1-5	Requête HTTP ...	13	✓	4148	129	13	0
6	02:13:26.725	Sosie 2 users 1-6	Requête HTTP ...	10	✓	4148	129	9	0
7	02:13:27.227	Sosie 2 users 1-7	Requête HTTP ...	14	✓	4148	129	13	0
8	02:13:27.727	Sosie 2 users 1-8	Requête HTTP ...	12	✓	4148	129	12	0
9	02:13:28.228	Sosie 2 users 1-9	Requête HTTP ...	11	✓	4148	129	11	0
10	02:13:28.728	Sosie 2 users 1-10	Requête HTTP ...	12	✓	4148	129	12	0

☐ Défilement automatique ? ☐ Échantillons enfants? Nombre d'échantillons 10 Dernier échantillon 12 Moyenne 13 Écart type 3

Echier Editer Rechercher Lancer Options Aide

00:00:04 1 0 / 10

Plan de test

- Sosie 2 users
 - Requête HTTP
 - Tableau de
 - Requête HTTP
 - Tableau de
 - Paramètres HT
 - Rapport agrégé
 - Arbre de résultat
- Plan de travail

Tableau de résultats

Nom :

Commentaires :

Écrire les résultats dans un fichier ou lire les résultats depuis un fichier CSV / JTL

Nom du fichier : Parcourir... Uniquement : ☐ Erreurs ☐ Succès Configurer

Echantillon #	Heure début	Nom d'unité	Libellé	Temps (ms)	Statut	Octets	Octets envoyés	Latence	Établ. Conn.(ms)
1	02:13:24.215	Sosie 2 users 1-1	Requête HTTP ...	43	✓	2907	123	43	0
2	02:13:24.715	Sosie 2 users 1-2	Requête HTTP ...	5	✓	2907	123	4	0
3	02:13:25.215	Sosie 2 users 1-3	Requête HTTP ...	6	✓	2907	123	6	0
4	02:13:25.716	Sosie 2 users 1-4	Requête HTTP ...	10	✓	2907	123	10	1
5	02:13:26.216	Sosie 2 users 1-5	Requête HTTP ...	9	✓	2907	123	8	1
6	02:13:26.717	Sosie 2 users 1-6	Requête HTTP ...	7	✓	2907	123	7	0
7	02:13:27.218	Sosie 2 users 1-7	Requête HTTP ...	8	✓	2907	123	8	1
8	02:13:27.718	Sosie 2 users 1-8	Requête HTTP ...	8	✓	2907	123	8	0
9	02:13:28.217	Sosie 2 users 1-9	Requête HTTP ...	10	✓	2907	123	9	1
10	02:13:28.718	Sosie 2 users 1-10	Requête HTTP ...	9	✓	2907	123	9	1

☐ Défilement automatique ? ☐ Échantillons enfants? Nombre d'échantillons 10 Dernier échantillon 9 Moyenne 11 Écart type 10

Nom d'unités: 10 groupes d'unités comportant chacun un thread(1-1 à 1-10). Un plan de test avec deux groupes d'unités comportant chacun deux threads utiliserait les noms suivants:

1-1

1-2

2-1

2-2

Temps (temps de réponse): différence entre le moment où la requête a été envoyée et le moment où la réponse a été entièrement reçue.

Statut: définit le résultat de l'échantillon. Si c'est égal à "OK", le statut est défini comme succès, sinon l'échantillon est marqué comme ayant échoué.

octets: octets reçus.

octets envoyés: Nombre d'octets envoyés pour l'échantillon.

Latence: différence entre le moment où la requête a été envoyée et le moment où la réponse a commencé à être reçue.

Étab. Conn: temps d'établissement de la connexion (ms). Connexion non pris en charge par tous les échantillons.

Moyenne: Moyenne des temps de réponse.

Écart type: Mesure dispersion ou variabilité de l'ensemble des temps de réponses par rapport à la moyenne. C'est l'écart type des valeurs prises par la variable considéré ici comme le temps de réponse. Le groupe d'unité présente une hétérogénéité normale.

Fichier Éditer Rechercher Lancer Options Aide

00:00:04 1 0 / 10

Plan de test

- Sosie 2 users
 - Requête HTTP lo
 - Tableau de r
 - Requête HTTP in
 - Tableau de r
 - Paramètres HT
 - Rapport agrégé**
 - Arbre de résulta
- Plan de travail

Rapport agrégé

Nom : Rapport agrégé (sommaire)

Commentaires :

Écrire les résultats dans un fichier ou lire les résultats depuis un fichier CSV / JTL

Nom du fichier : Parcourir... Uniquement : ☐ Erreurs ☐ Succès

Libellé	# Echantillo...	Moyenne	Médiane	90% centile	95% centile	99% centile	Min	Max	% Erreur	Débit	Ko/sec reçus	Ko/sec émis
Requête HT...	10	11	8	10	10	43	5	43	0,00%	2,2/sec	6,29	0,27
Requête HT...	10	13	12	15	15	23	9	23	0,00%	2,2/sec	9,04	0,28
TOTAL	20	12	10	15	15	23	5	43	0,00%	4,4/sec	15,23	0,54

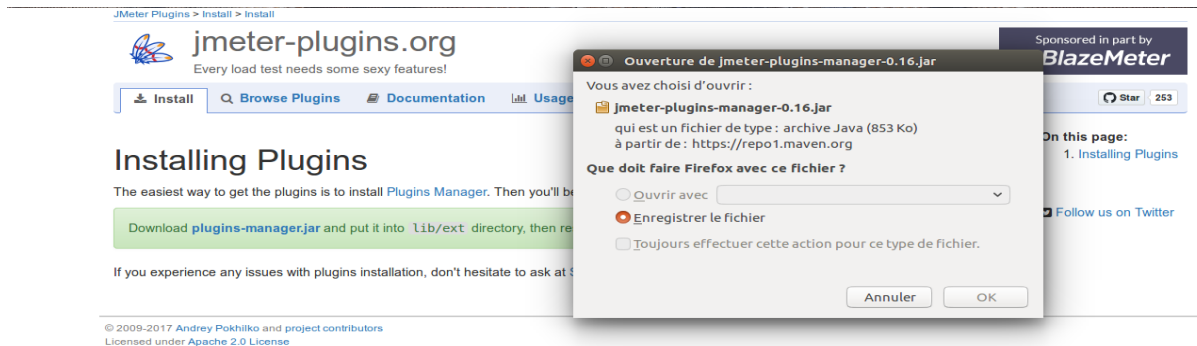
- Médiane:** Médiane des temps de réponses.
- 90% centile:** ex. en dessous de 10 ms 90% des requêtes sont exécutées.
- Min:** Temps de réponse minimal
- Max:** Temps de réponse maximal
- %Erreur:** Taux d'erreurs (ex. Exemple pourcentage des tests ayant échoué)
- Débit:** Nombre de requêtes que le serveur est capable de gérer à la seconde.
Plus ce débit sera élevé, mieux ça sera.

Utilisation des plugins JMeter

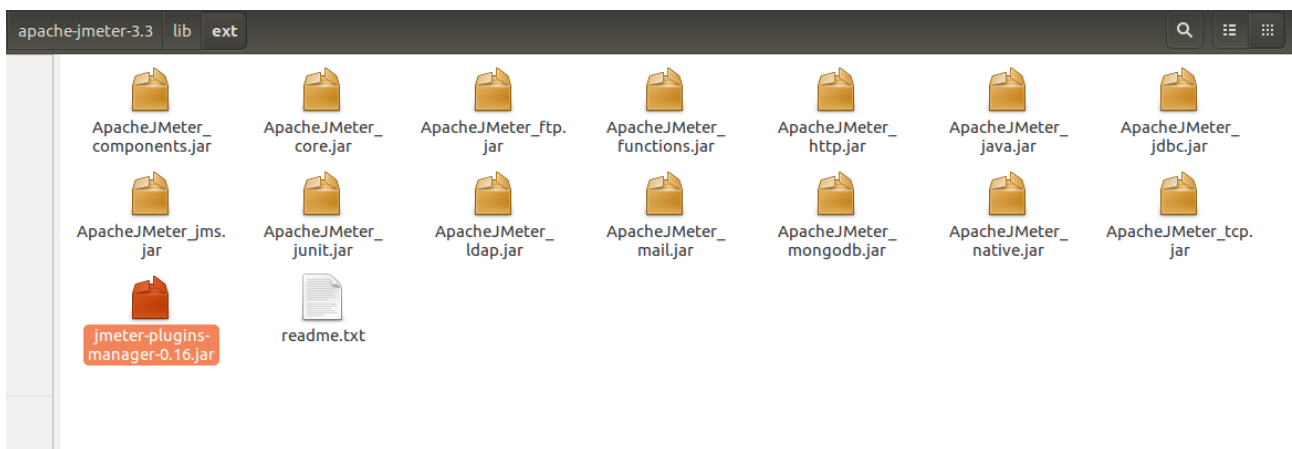
JMeter c'est aussi un ensemble de plugins permettant d'avoir plus de fonctionnalités. L'utilitaire **plugins manager** de JMeter permet d'ajouter d'autres modules en plus des modules natifs (Récepteurs, Échantillons, Graphs, Tableaux..).

JMeter plugin manager est disponible sur le lien:

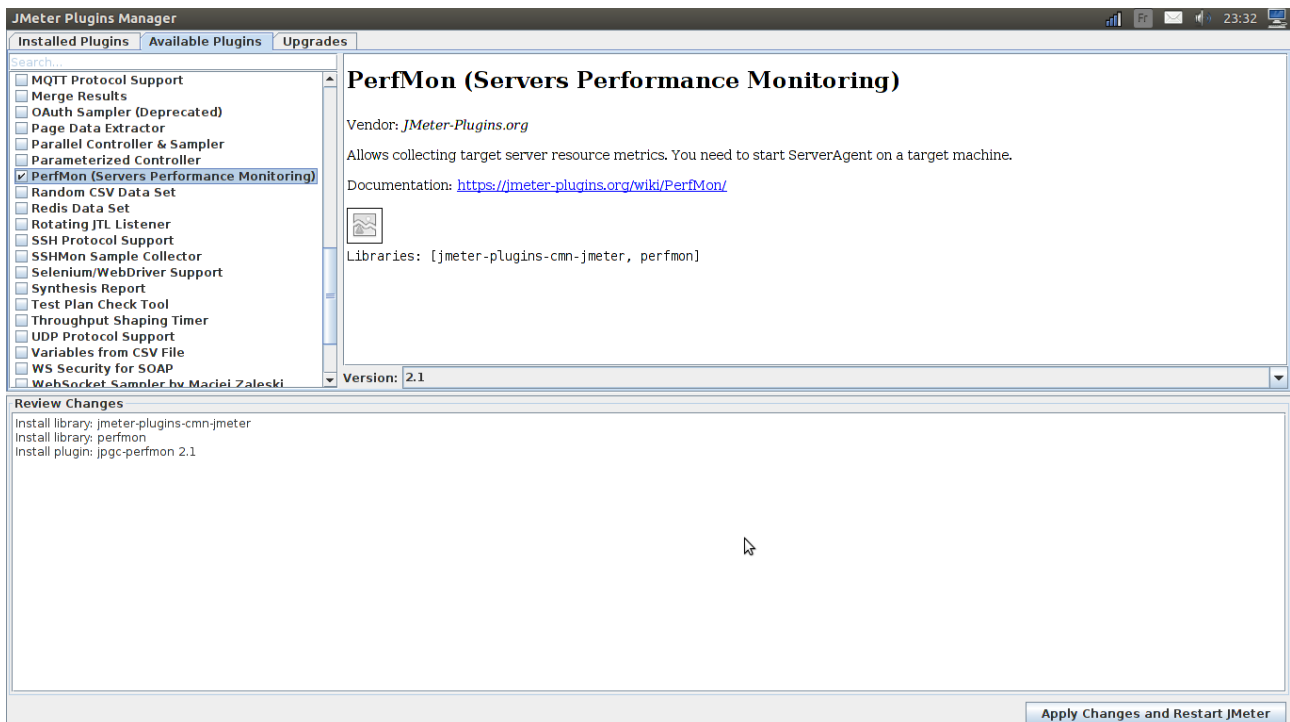
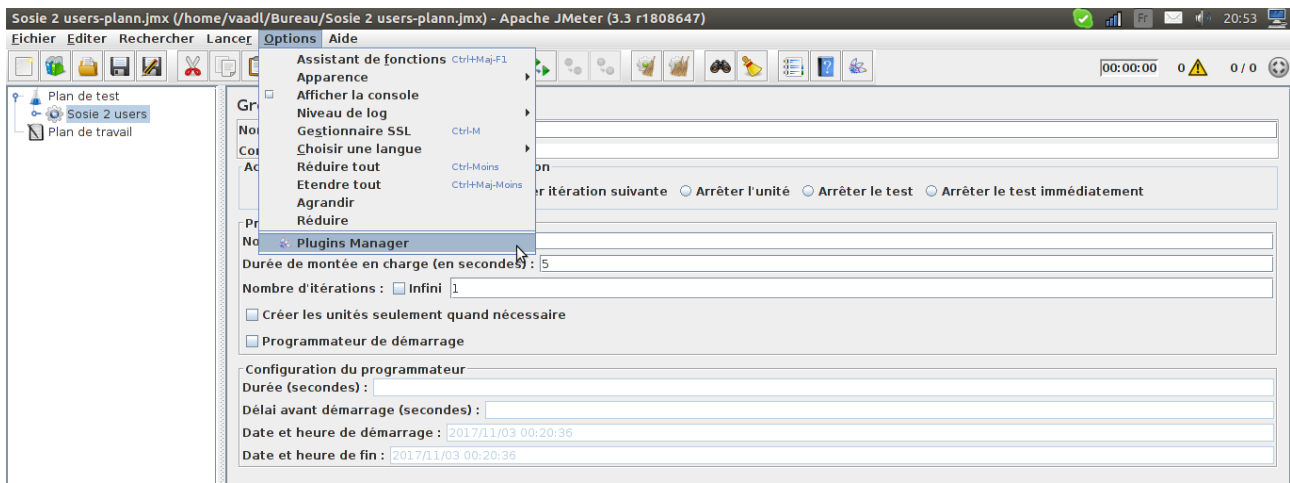
<https://jmeter-plugins.org/install/Install/>



Après le téléchargement, on le place dans le répertoire **lib/ext** et on redémarre JMeter.



Le plugin **PerfMon Metrics Collector** permet de connaître **les performances système (Mémoire, CPU, réseau...)**



Après avoir cliqué sur «**Apply Changes and Restart JMeter**» pour prendre en compte les modifications, il faudra aussi **télécharger PerfMOn ServerAgent**, extraire l'archive et le placer dans le répertoire **lib** de JMeter. ServerAgent va permettre d'automatiser les traitements.

Lien: <https://jmeter-plugins.org/wiki/PerfMonAgent/>

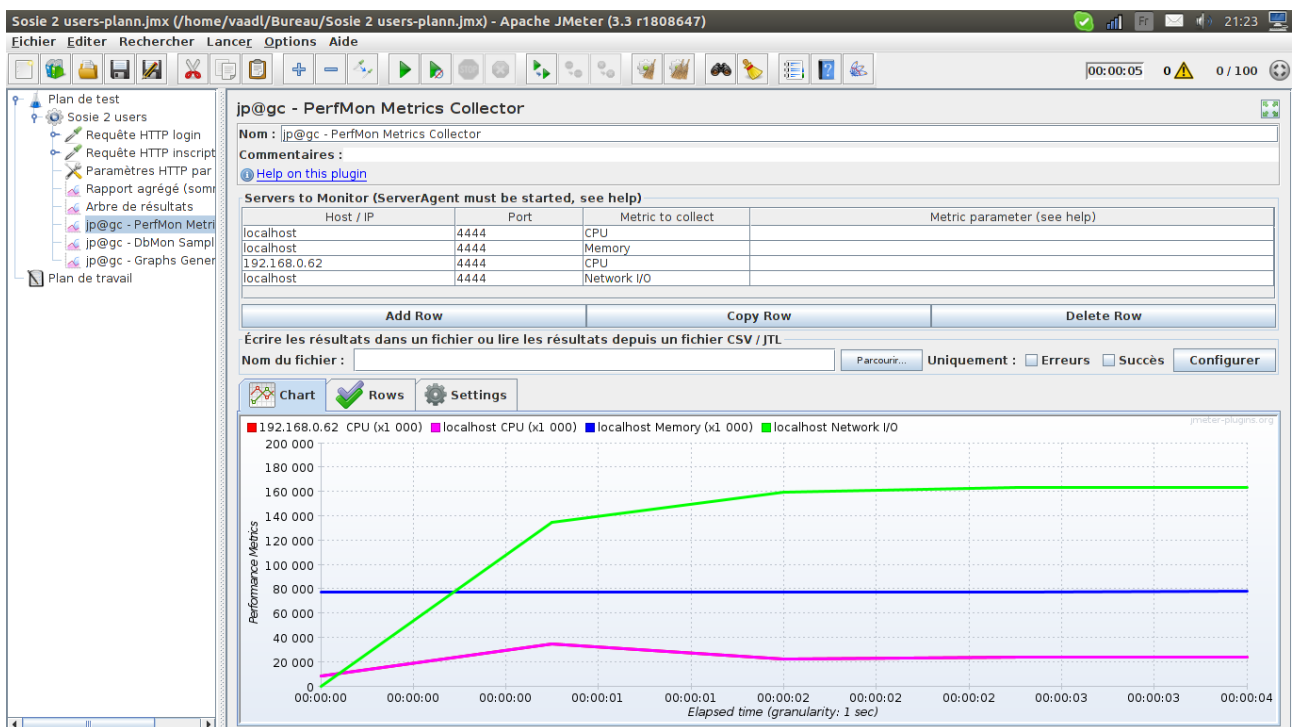


Avant de lancer les tests pour PerfMon il faudra d'abord exécuter ServerAgent:

Sur **windows** exécutez le fichier **serverAgent.bat** et sur **linux** **serverAgent.sh**

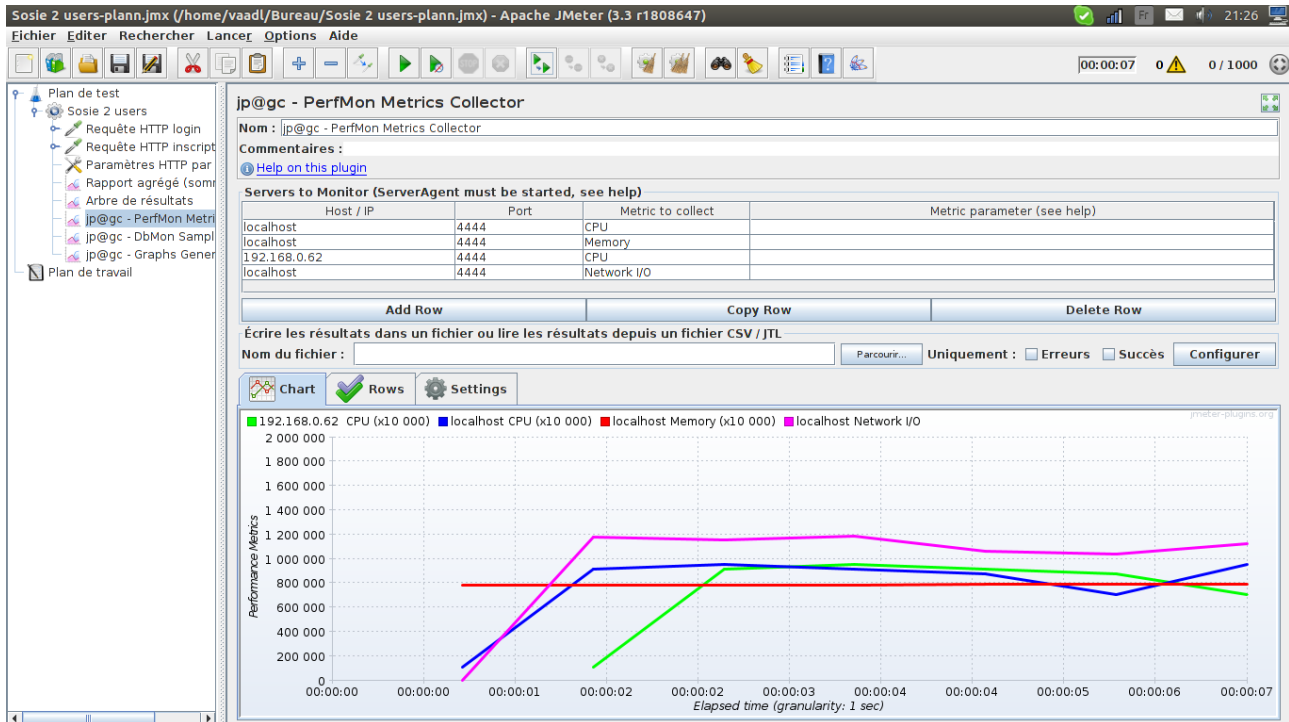
```
vaadl@vaadl-Satellite-P745: /usr/local/apache-jmeter-3.3/lib/ServerAgent-2.2.1
vaadl@vaadl-Satellite-P745:~$ cd /usr/local/apache-jmeter-3.3/lib/ServerAgent-2.2.1/
vaadl@vaadl-Satellite-P745:/usr/local/apache-jmeter-3.3/lib/ServerAgent-2.2.1$ ./startAgent.sh
```

Les métriques système avec 100 utilisateurs simultanés



**le taux d'occupation du processeur (CPU) est de loin inférieur à 80%,
idem pour la mémoire.**

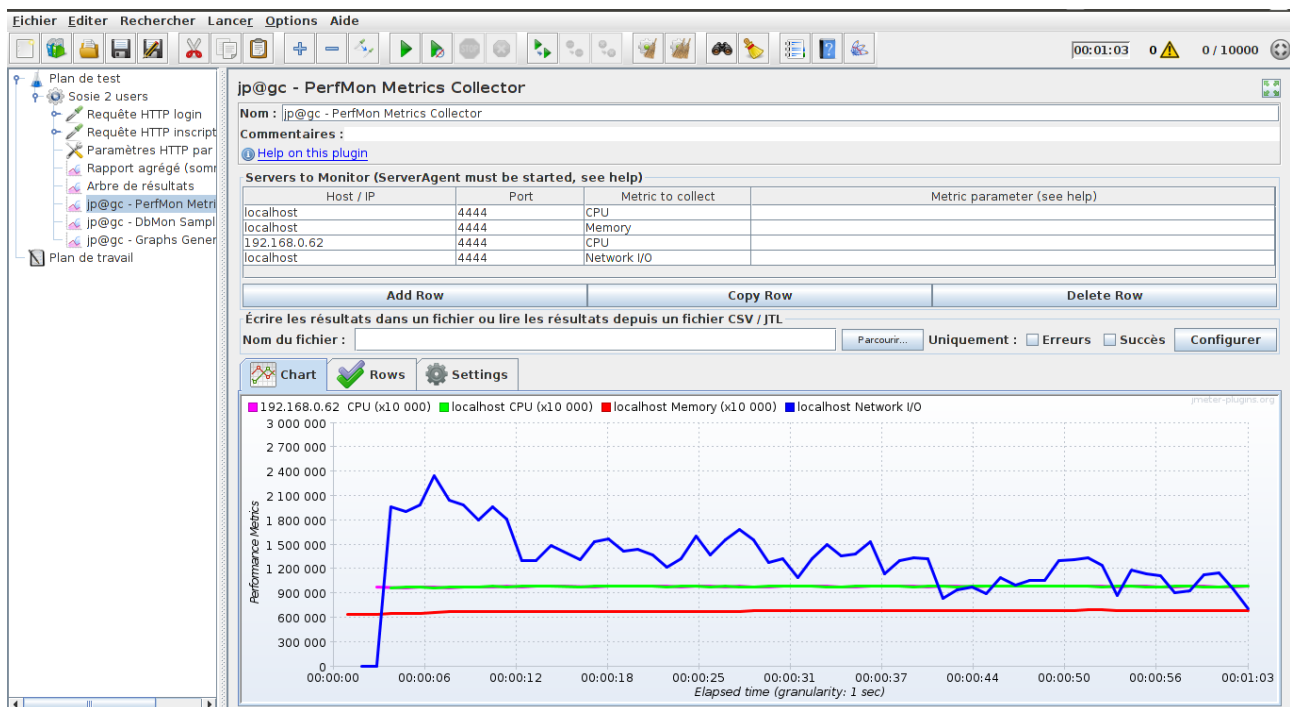
Les métriques système avec 1000 utilisateurs simultanés



Nous avons ainsi mesurer l'utilisation du processeur et de la mémoire du serveur et du réseau pendant le test en utilisant JMeter PerfMon.

On voit encore que le taux d'occupation du processeur (CPU) est de loin inférieur à 80%, idem pour la mémoire et le réseau.

Les métriques système avec 10000 utilisateurs simultanés



Là encore, le réseau est la ressource la plus consommée, il écoute l'interface eth0 pour le débit de paquets transmis.

CONCLUSION

- Le serveur SOSIE 2 est donc capable de traiter beaucoup de demandes.
- La base de données traitera donc moins rapidement les demandes avec la surcharge, ce qui risque d'en faire le point de production de plus faible cadence dans l'architecture.
- Le serveur sera obligé parfois d'être en attente.