

Visualisation UI

Introduction

Ce pack contient quatre modèles d'hôte permettant de superviser des bases MongoDB sur les points suivants :

- Connexions à la base Mongo (durée et pourcentage de connexions utilisées par rapport aux connexions disponibles)
- Durée moyenne de l'écriture sur disque
- Durée de la dernière écriture sur disque
- Proportion d'échec lors de la recherche dans l'index
- Durée d'attente des verrous
- Statistiques sur les opérations sur les documents (suppressions, insertions, mises à jours, requêtes)
- Etat des replicaset
- Délai de réplication

Ce pack peut se connecter à une base Mongo de deux façons différentes :

- soit directement au serveur Mongo
- soit par l'intermédiaire d'un tunnel SSH, afin d'améliorer la sécurité en configurant le serveur Mongo pour n'écouter que sur l'interface réseau locale

Configuration de la connexion SSH

Avec la configuration par défaut, le pack Mongo se connecte en direct sur le serveur MongoDB.

Lorsque le mode "SSH" est activé (voir la section **Dans la configuration de l'hôte**), il essaie de se connecter au serveur Mongo par l'intermédiaire de SSH avec l'utilisateur "shinken" et la clé privée se trouvant dans `~shinken/.ssh/id_rsa`.

Notez que si vous avez déjà effectué cette configuration pour le [Pack Linux](#), vous pouvez réutiliser la même pour le pack MongoDB.

Dans la configuration de l'hôte

Dans l'onglet **Général**, ajoutez l'un des modèles d'hôte du pack MongoDB à la liste des modèles hérités.

Dans l'onglet "Données", dans la section consacrée au modèle d'hôte MongoDB hérité :

- Spécifiez la valeur "ssh" pour la donnée "`MONGO_CONNECTION_METHOD`"
- Assurez-vous que les données "`MONGO_SSH_KEY`" et "`MONGO_SSH_USER`" font référence à la clé SSH privée définie dans la section **Côté Shinken** et à l'utilisateur créé dans la section **Côté client**.

Côté client (serveur supervisé)

- Créez un utilisateur local "shinken" avec un dossier "home" et un mot de passe

```
adduser -m -r shinken
passwd shinken
```

Côté Shinken (serveur Poller)

Connectez-vous sur le serveur en tant qu'utilisateur "shinken" et copiez la clé SSH publique vers le serveur à superviser :

```
[root@shinken-poller ~]# su - shinken
[shinken@shinken-poller ~]# ssh-copy-id shinken@remote_host
The authenticity of host '192.168.1.19 (192.168.1.19)' can't be established.
RSA key fingerprint is 00:ff:ee:dd:cc:bb:aa:d6:d3:79:1d:f6:93:47:80:27.
Are you sure you want to continue connecting (yes/no)? yes
shinken@remote_host's password: XXXXXXXXXXXX
Now try logging into the machine, with "ssh '192.168.1.19'", and check in:
  .ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.

[root@shinken-poller ~]# ssh shinken@remote_host -i ~shinken/.ssh/id_rsa
```

Cette copie de clé permet d'autoriser le Poller à se connecter sur la machine à superviser. Sur une architecture Shinken contenant plusieurs Pollers, il faudra effectuer cette opération sur chaque Poller pour lui permettre de se connecter en SSH sur le serveur à superviser.

Comment utiliser le pack MongoDB

Modèles d'hôtes fournis par le pack

Quatre modèles d'hôtes sont disponibles en fonction de votre configuration MongoDB :

- MongoDB
- MongoDB-no-replication si votre architecture Mongo n'utilise pas la réplication)

Ces deux modèles sont prévus pour être utilisés sur une version de Mongo antérieure à la version 3

- MongoDB3 et MongoDB3-no-replication sont prévus pour être utilisés avec les versions de Mongo ultérieures à la version 3.

Ces quatre modèles d'hôtes fournissent des checks identiques et se configurent de la même manière, mais seuls les checks compatibles avec la configuration choisie sont présents

Via l'interface de configuration

Dans l'interface de configuration, créez et [éditez un hôte](#), et ajoutez le modèle d'hôte choisi dans la liste des modèles d'hôtes utilisés

Via un fichier de configuration

Dans un fichier de configuration .cfg de votre choix, créer un hôte et définir la propriété "use" pour lui ajouter le nom du pack MongoDB choisi.

Ce fichier cfg doit ensuite être importé dans Shinken Entreprise via une source (plus d'informations sont disponibles dans la page de documentation sur la [Syntaxe des fichiers d'imports](#)).

Résumé des checks

	Nom du check	Description	Seuil Warning par défaut	Seuil Critique par défaut	Disponible dans les modèles
1	Mongod b-connection	Vérifie le temps d'établissement d'une connection (en secondes)	MONGO_CONNECTION_WARN >2	MONGO_CONNECTION_CRIT > 4	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb-no-replication • mongodb3-no-replication
2	Mongod b-flush-time	Vérifie le temps moyen d'écriture des données sur disque (en ms)	MONGO_FLUSHING_WARN > 100	MONGO_FLUSHING_CRIT > 200	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb-no-replication • mongodb3-no-replication
3	Mongod b-index-miss-ratio	Vérifie le pourcentage d'échec lors de recherches dans les index. Si ce pourcentage est trop haut, cela peut signifier qu'il faut ajouter d'autres index.	MONGO_INDEX_MISS_RATIO_WARN > 0.005	MONGO_INDEX_MISS_RATIO_CRIT > 0.01	<ul style="list-style-type: none"> • mongodb • mongodb-no-replication
4	Mongod b-last-flush	Vérifie le dernier temps d'écriture des données sur disque (en ms) ; si ce temps devient trop élevé cela pourrait signifier qu'il faut des disques plus rapides ou qu'il est temps de répartir les données entre plus de serveurs (sharding)	MONGO_LAST_FLUSH_TIME_WARN > 200	MONGO_LAST_FLUSH_TIME_CRIT > 400	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb-no-replication • mongodb3-no-replication
5	Mongod b-lock-time	Vérifie le pourcentage du temps passé à attendre la libération de verrous. Un pourcentage trop important signifie la plupart du temps que votre serveur est surchargé.	MONGO_LOCK_WARN > 5	MONGO_LOCK_CRIT > 10	<ul style="list-style-type: none"> • mongodb • mongodb-no-replication

6	Mongod b-open- connecti ons	Vérifie le pourcentage de connexions utilisées dans le serveur Mongo.	MONGO_CONN ECTIONS_WAR N > 70	MONGO_CON NECTIONS_CR IT > 80	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb -no- replication • mongodb 3-no- replication
7	Mongod b- queries- stat- delete	Vérifie le nombre de suppressions par seconde	MONGO_DELE TE_PER_SEC_ WARN_LEVEL > 10000	MONGO_DELE TE_PER_SEC_ CRIT_LEVEL > 20000	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb -no- replication • mongodb 3-no- replication
8	Mongod b- queries- stat- insert	Vérifie le nombre d'insertions par seconde	MONGO_INSE RT_PER_SEC_ WARN_LEVEL > 10000	MONGO_INSE RT_PER_SEC_ CRIT_LEVEL > 20000	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb -no- replication • mongodb 3-no- replication
9	Mongod b- queries- stat- query	Vérifie le nombre de requêtes par seconde	MONGO_QUE RY_PER_SEC_W ARN_LEVEL > 10000	MONGO_QUE RY_PER_SEC_ CRIT_LEVEL > 20000	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb -no- replication • mongodb 3-no- replication
10	Mongod b- queries- stat- update	Vérifie le nombre de mises à jour par seconde	MONGO_UPDA TE_PER_SEC_ WARN_LEVEL > 10000	MONGO_UPDA TE_PER_SEC_ CRIT_LEVEL > 20000	<ul style="list-style-type: none"> • mongodb • mongodb3 • mongodb -no- replication • mongodb 3-no- replication
11	Mongod b- replicaset	Vérifie le statut des noeuds dans un replicaset. Selon le statut renvoyé le check est en WARNING (statuts 0, 3, 5) CRITIQUE (statuts 4,6,8) ou OK (statuts 1, 2, 7)			<ul style="list-style-type: none"> • mongodb • mongodb3
12	Mongo- replicatio n-lag	Délai de réplication entre les serveurs Mongo (en secondes). Le check utilise la métrique "optime" renvoyée par rs.status(). Cette valeur peut être supérieure au délai réel car les requêtes ne sont effectuées qu'à intervalles de plusieurs secondes. Ainsi, une valeur inférieure à 10 peut ne correspondre à aucun délai réel, mais au délai entre deux requêtes	MONGO_REPLI CATION_LAG_ WARN > 15	MONGO_REPL ICATION_LAG > 30	<ul style="list-style-type: none"> • mongodb • mongodb3

Personnaliser les seuils de Warning et Critique

Le pack MongoDB définit des seuils par défaut sur les checks qu'il utilise et que vous pouvez modifier.

Changer les seuils pour un seul hôte

Les modèles d'hôte livrés dans Shinken contiennent souvent des variables permettant de changer les seuils et options des checks.

Pour changer ces seuils sur un seul hôte en particulier, la manière la plus simple est de changer des variables via les données présentes sur l'hôte directement:

- Dans l'interface de Configuration, éditer l'hôte et aller dans l'onglet Données
Par exemple, changer la donnée MONGO_CONNECT_CRIT à 10 aura pour effet de changer le seuil Critique du check "Mongodb-connection" à 10
- Via un fichier de configuration.
Pour changer le seuil Critique du check "Mongodb-queries-stat-delete", éditer l'hôte et changer la donnée _MONGO_DELETE_PER_SEC_CRIT_LEVEL

Changer les seuils pour tous les hôtes qui utilisent l'un des modèles du pack MongoDB

Pour changer les seuils sur tous les hôtes qui utilisent l'un des modèles d'hôte du pack MongoDB, on pourrait être tenté de modifier directement le modèle d'hôte et changer les données personnalisées de cet hôte.

Mais, dans la prochaine mise à jour de Shinken Entreprise, ces modèles d'hôtes/checks peuvent être modifiés, ce qui occasionne un changement de comportement et va probablement causer des problèmes dans votre configuration Shinken.

L'alternative conseillée est de cloner les éléments utilisés dans le pack MongoDB et de les renommer. On peut ensuite modifier sans aucun risque liés aux mises à jour de Shinken Entreprise.

Ce pack MongoDB cloné peut ensuite être modifié et personnalisé selon les besoins.

Par exemple, le changement des seuils s'effectue de la même manière en changeant les données comme décrit dans la section précédente mais dans l'un des modèles d'hôte du pack MongoDB au lieu de faire cette modification directement dans les hôtes.