

\$KEY\$ Frontend Responses by SSH (pour le modèle HAproxy-by-SSH__frontend)

Sommaire

- Contexte
- Paramétrage
 - Données utilisées provenant du modèle
 - Données communes pour les checks des modèles
 - Données spécifiques pour ce check
 - Les données DFE (Duplicate Foreach)
 - Données utilisées provenant du check
 - Données globales
 - Propriétés de l'hôte
- Résultat
 - Exemple
 - Interprétation des données
 - Statut
 - Résultat
 - Résultat long
- Métriques
 - Interprétation des métriques
 - Exemple
- Erreurs et pré-requis
 - Erreurs de connexion (communes à tous les checks)
 - UNKNOWN – SSH authentication failed.
 - UNKNOWN – SSH private key could not be loaded
 - UNKNOWN HAProxy statistics query failed

Contexte

Le check **\$KEY\$ Frontend Responses by SSH** permet de superviser les réponses HTTP retournées par un frontend HAProxy.

Il permet de suivre précisément la répartition des codes de réponses générés par celui-ci, afin d'identifier rapidement des comportements anormaux ou une dégradation du service exposé.



En fonction du trafic et des applications publiées, un frontend peut retourner différents types de réponses HTTP. Une augmentation inhabituelle de certains codes peut révéler :

- Une erreur applicative côté backend
- Un problème de routage ou de configuration HAProxy
- Une indisponibilité partielle du service
- Une surcharge de l'infrastructure
- Une attaque ou un comportement anormal des clients

Ce check permet donc :

- De surveiller l'évolution des réponses HTTP du frontend
- De détecter rapidement une hausse des erreurs 4xx ou 5xx
- D'identifier des anomalies de fonctionnement applicatif

Pour résumer, ce check est un indicateur clé pour assurer la bonne qualité des réponses délivrées aux utilisateurs.

Statut	Nom de check	Résultat	Résultat Long														
	[main] Frontend Responses by SSH	 Collected response counters for frontend main .	<p>HAProxy Frontend Responses: main :</p> <table border="1"><thead><tr><th>Frontend</th><th>2xx Success</th><th>3xx Redirect</th><th>4xx Client Err</th><th>5xx Server Err</th><th>Other</th><th>Frontend State</th></tr></thead><tbody><tr><td>main</td><td>1454242</td><td>0</td><td>1019</td><td>0</td><td>0</td><td>OPEN</td></tr></tbody></table> <p>Response classes are grouped as success (2xx), redirect (3xx), client errors (4xx), server errors (5xx), and other (1xx/other). Frontend state still drives the plugin exit status.</p>	Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State	main	1454242	0	1019	0	0	OPEN
Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State											
main	1454242	0	1019	0	0	OPEN											

Paramétrage

Le check utilise la ligne de commande suivante :

```
$GLOBAL_HAPROXY-BY-SSH_SHINKEN_PLUGINDIR$/check_haproxy_ssh.py
-H "$HOSTADDRESS$"
-u "$_HOSTHAPROXY-BY-SSH__SSH-USER$"
-k "$_HOSTHAPROXY-BY-SSH__SSH-KEY$"
-x "$_HOSTHAPROXY-BY-SSH__SSH-PASSPHRASE$"
-p "$_HOSTHAPROXY-BY-SSH__SSH-PORT$"
--mode frontend_responses
-d "$_HOSTHAPROXY-BY-SSH__EXCLUDE-PAUSED-DRAIN$"
-n "$ARG1$"
```

Données utilisées provenant du modèle

Données communes pour les checks des modèles

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation de Shinken	Description
HAPROXY-BY-SSH__SSH-KEY	l'Hôte (Onglet Données)	--	\$HAPROXY-BY-SSH__SSH-KEY\$	/var/lib/shinken/ssh/id_rsa	Chemin vers la clé SSH privé de l'utilisateur shinken , sur le serveur hébergeant le Poller qui exécutera le check. <ul style="list-style-type: none"> Cette clé doit être présente dans les clés autorisées du compte utilisateur utilisé pour se connecter sur le serveur linux supervisé (voir la donnée <code>SSH_USER</code> ci dessous).
HAPROXY-BY-SSH__SSH-PASSPHRASE	l'Hôte (Onglet Données)	--	\$HAPROXY-BY-SSH__SSH-PASSPHRASE\$	\$SSH_KEY_PASSPHRASE\$	Phrase secrète utilisée pour déchiffrer la clé privée de l'utilisateur (<i>si celle-ci est protégée par une passphrase</i>). La clé privée déchiffré est ensuite utilisée pour authentifier l'utilisateur.
HAPROXY-BY-SSH__SSH-PORT	l'Hôte (Onglet Données)	--	\$HAPROXY-BY-SSH__SSH-PORT\$	22	Port de connexion SSH.
HAPROXY-BY-SSH__SSH-USER	l'Hôte (Onglet Données)	--	\$HAPROXY-BY-SSH__SSH-USER\$	shinken-user	Nom de l'utilisateur pour se connecter sur le serveur supervisé.

Données spécifiques pour ce check

Pas de données spécifiques pour ce check

Les données DFE (Duplicate Foreach)

Nom	Modifiable sur	Unités	Défaut	Valeur par défaut à l'installation de Shinken	Description
HAPROXY-BY-SSH__FRONTENDS	l'Hôte (Onglet Données)	---	frontend1\$(frontend1)\$	frontend1\$(frontend1)\$	Nom du ou des frontends (<i>séparés par une virgule</i>), exemple "main\$(main)\$,app\$(app)\$". Check(s) impacté(s) : <ul style="list-style-type: none"> [\$KEY\$] Frontend Status by SSH (pour le modèle <code>Haproxy-by-SSH__frontend</code>). [\$KEY\$] Frontend traffic by SSH (pour le modèle <code>Haproxy-by-SSH__frontend</code>). [\$KEY\$] Frontend Response by SSH (pour le modèle <code>Haproxy-by-SSH__frontend</code>).

Données utilisées provenant du check

Pas de données spécifiques pour ce check

Données globales


Nom	Modifiable sur	Unité	Défaut	Valeur par défaut à l'installation	Description
USERPLUGINDIR	Non modifiable <i>(Sauf Admin Shinken)</i>	--	/var/lib/shinken-user/libexec	/var/lib/shinken-user/libexec	Chemin absolu contenant les sondes installées par Shinken.
GLOBAL_HAPROXY-BY-SSH_SHINKEN_VENDOR	Non modifiable <i>(Sauf Admin Shinken)</i>	--	shinken-additional-packs	shinken-additional-packs	Dossier fourni par Shinken.
GLOBAL_HAPROXY-BY-SSH_SHINKEN_PACKNAME	Non modifiable <i>(Sauf Admin Shinken)</i>		Haproxy-by-SSH__shinken	Haproxy-by-SSH__shinken	Dossier contenant les sondes.
GLOBAL_HAPROXY-BY-SSH_SHINKEN_PLUGINSDIR	Non modifiable <i>(Sauf Admin Shinken)</i>	--	\$USERPLUGINDIR\$ /\$GLOBAL_HAPROXY-BY-SSH_SHINKEN_VENDOR\$ /\$GLOBAL_HAPROXY-BY-SSH_SHINKEN_PACKNAME\$	/var/lib/shinken-user/libexec/shinken-additional-packs/Haproxy-by-SSH__shinken	Chemin absolu du dossier contenant les sondes du pack H Aproxoy-by-SSH__shinken (<i>non modifiable</i>).

Propriétés de l'hôte

Nom	Modifiable sur	Unité	Défaut	Valeur par défaut	Description
HOSTADDRESS	l'Hôte <i>(Onglet Général)</i>	--	Nom de l'hôte	Nom de l'hôte	Adresse de l'hôte

Résultat

Exemple

Statut	Nom de check	Résultat	Résultat Long														
	[main] Frontend Responses by SSH	OK Collected response counters for frontend main .	<p>HAProxy Frontend Responses: main :</p> <table border="1"> <thead> <tr> <th>Frontend</th> <th>2xx Success</th> <th>3xx Redirect</th> <th>4xx Client Err</th> <th>5xx Server Err</th> <th>Other</th> <th>Frontend State</th> </tr> </thead> <tbody> <tr> <td>main</td> <td>1454242</td> <td>0</td> <td>1019</td> <td>0</td> <td>0</td> <td>OPEN</td> </tr> </tbody> </table> <p>Response classes are grouped as success (2xx), redirect (3xx), client errors (4xx), server errors (5xx), and other (1xx/other). Frontend state still drives the plugin exit status.</p>	Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State	main	1454242	0	1019	0	0	OPEN
Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State											
main	1454242	0	1019	0	0	OPEN											

Interprétation des données

Statut

Il peut prendre 4 états **OK** / **CRITIQUE** / **ATTENTION** / **INCONNU** .

- Le statut va dépendre du retour de sonde.
- Voici un tableau récapitulatif du statut attendu suivant le retour de sonde :

Situation	Statut	Exemple

<ul style="list-style-type: none"> Le frontend sélectionné est dans l'état "STOP". 	CRITIQUE	<table border="1"> <thead> <tr> <th>Statut</th> <th>Nom de check</th> <th>Résultat</th> <th>Résultat Long</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">❌</td> <td>[main] Frontend Responses by SSH</td> <td style="color: red;">CRITICAL</td> <td>Collected response counters for frontend main, which is in a critical state.</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="7">HAProxy Frontend Responses: main :</th> </tr> <tr> <th>Frontend</th> <th>2xx Success</th> <th>3xx Redirect</th> <th>4xx Client Err</th> <th>5xx Server Err</th> <th>Other</th> <th>Frontend State</th> </tr> </thead> <tbody> <tr> <td>main</td> <td>1454242</td> <td>0</td> <td style="background-color: #ffcc00;">1019</td> <td>0</td> <td>0</td> <td style="background-color: #ff0000; color: white; text-align: center;">STOP</td> </tr> </tbody> </table> <p><small>Response classes are grouped as success (2xx), redirect (3xx), client errors (4xx), server errors (5xx), and other (1xx/other). Frontend state still drives the plugin exit status.</small></p>	Statut	Nom de check	Résultat	Résultat Long	❌	[main] Frontend Responses by SSH	CRITICAL	Collected response counters for frontend main, which is in a critical state.	HAProxy Frontend Responses: main :							Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State	main	1454242	0	1019	0	0	STOP
Statut	Nom de check	Résultat	Résultat Long																												
❌	[main] Frontend Responses by SSH	CRITICAL	Collected response counters for frontend main, which is in a critical state.																												
HAProxy Frontend Responses: main :																															
Frontend	2xx Success	3xx Redirect	4xx Client Err	5xx Server Err	Other	Frontend State																									
main	1454242	0	1019	0	0	STOP																									

Résultat

Le résultat court affiche, en une ligne, un résumé des status détectés. Ce retour compact permet de pouvoir visualiser l'information même avec la taille des lignes réduites dans l'interface de visualisation.

Résultat long

Le résultat long affiche un tableau regroupant l'ensemble des informations sur les codes détectés.

Métriques

Interprétation des métriques

Nom de la métrique	Unité	Description	Seuil d'avertissement	Seuil critique
hrsp_2xx_(frontend)	--	Retourne le nombre de codes de type 200 détectés sur le HAProxy.		
hrsp_3xx_(frontend)	--	Retourne le nombre de codes de type 300 détectés sur le HAProxy.		
hrsp_4xx_(frontend)	--	Retourne le nombre de codes de type 400 détectés sur le HAProxy.		
hrsp_5xx_(frontend)	--	Retourne le nombre de codes de type 500 détectés sur le HAProxy.		
hrsp_other_(frontend)	--	Retourne le nombre de codes inconnus détectés sur le HAProxy.		

Exemple

Métrique	Valeur	Seuil d'avertissement	Seuil critique
hrsp_2xx_main	1454242		
hrsp_3xx_main	0		
hrsp_4xx_main	1019		
hrsp_5xx_main	0		
hrsp_other_main	0		

Erreurs et pré-requis

Erreurs de connexion (communes à tous les checks)

UNKNOWN – SSH authentication failed.

L'authentification par clé SSH n'as pas pu s'effectuer.

Statut	Nom de check	Résultat	Résultat Long
?	Global Status by SSH	UNKNOWN	SSH authentication failed.


Résolution :

Possibles raisons :

- L'utilisateur utilisé n'existe pas
- La paire utilisateur / clef publique n'est pas autorisée à se connecter sur la machine supervisée.
- La clé publique n'a pas été déposée dans le "authorized_keys" côté serveur.

UNKNOWN – SSH private key could not be loaded

La clé SSH à utiliser sur le poller de Shinken n'as pas été trouvée.


Statut	Nom de check	Résultat	Résultat Long
	Global Status by SSH	UNKNOWN	SSH private key could not be loaded.

Résolution :

Vérifier sur les pollers si la clé privé indiqué dans la donnée "\$_HOSTHAPROXY-BY-SSH__SSH-KEY\$" est bien présente et accessible par l'utilisateur shinken.

UNKNOWN HAProxy statistics query failed

L'utilisateur utilisé sur le serveur HAProxy n'appartient pas au groupe HAProxy ou le socket HAProxy n'est pas lancé.

Statut	Nom de check	Résultat	Résultat Long
	Global Status by SSH	UNKNOWN	HAProxy statistics query failed.

Résolution :

La commande ci-dessous permet de lister les utilisateurs du groupe haproxy :

```
cat /etc/group | grep haproxy
```

Pour ajouter un utilisateur au groupe haproxy :

```
usermod -a -G haproxy user-service-shinken
```

Pour vérifier si le socket existe bien :

```
ls -l /var/run/haproxy.sock
```