

Editer un Cluster

Sommaire

Onglet : Général
Onglet : Données
Onglet : Supervision
Onglet : Valeurs par défaut pour les checks
Onglet : Droits de l'utilisateur
Onglet : Checks
 Essayer les checks
 Les checks cachés
Onglet : Notifications
Onglet : Expert

Onglet : Général

Propriété	Valeur	Venant des modèles
Nom *		
Définition		
Modèles de cluster hérités	[Par défaut [Aucun]]	
Royaume	Par défaut [All]	
Impact métier		Par défaut ***
URL externe		
Liste des URL externes		+ Créer une URL
Format fichier Cfg		
Activé		Par défaut Vrai
Pack		

- **Nom**

- Clé d'import : host_name

Cette propriété permet de définir le nom utilisé pour identifier le cluster. Il permet de référencer cet élément dans la définition d'autres clusters.

Les caractères ~!\$%^&*"'|<>?,()=/+ sont interdits dans ce champ.

— Un caractère non autorisé provoquera un compteur d'erreur

★ Ce champ ne peut pas rester vide

- **Définition**

- Clé d'import : bp_rule

Cette propriété est utilisée pour configurer en une règle logique une agrégation de plusieurs autres états (hôtes et checks).

Deux opérateurs de bases sont disponibles pour définir l'agrégation des états: & et |

Propriété	Valeur	Venant des modèles
Nom *	EOP	
Définition	non valide (1 ou 2) & (non OK (1) ou OK (2)) & (non valide ou OK (1) ou OK (2))	
Modèles de cluster hérités	[Par défaut [Aucun]]	
Royaume	Par défaut [All]	
Impact métier		Par défaut ***
URL externe		
Liste des URL externes		+ Créer une URL
Format fichier Cfg		
Activé		Par défaut Vrai
Pack		



Eléments avec des caractères spéciaux

Par exemple, il est possible :

- D'agréger par nom d'hôte :

```
"hote_1" & (!"hote_2" | "host_3")
```

- D'agréger par nom de check :

```
"hote_1", "check_1" & "host_2", "check_2"
```

- D'agréger par expression régulière sur le nom d'hôte :

```
r: /^host_/
```



Pour plus de détails

Vous trouverez l'ensemble des syntaxes possibles dans la page "[Les Clusters](#)"

- **Modèles de cluster hérités**

- *Clé d'import* : use

Les modèles de cluster qui sont attachés au cluster pour hériter de leurs propriétés et leurs données.

- L'ordre des modèles est important.
- Le premier modèle qui définit une propriété définit la valeur de la propriété pour le cluster.
- Si une propriété est définie directement sur le cluster, la valeur définie sera utilisée et pas celle du modèle.



Pour un administrateur de SI, certains modèles ne sont pas "visible" et ne seront pas disponibles dans la liste des choix.

- **Royaume**

- *Clé d'import* : realm

Cette propriété permet de définir le royaume où le cluster se trouvera.

En plaçant un cluster dans un royaume particulier, un des schedulers de ce royaume gèrera ce cluster.

- **Impact métier**

- *Clé d'import* : business_impact

Cette propriété permet de définir l'importance que l'on donne à un cluster pour le business. (valeur numérique entre 0 et 5 dans les fichiers d'imports, mais affichée dans l'interface avec des étoiles ou des dollars suivant le niveau) :

- du moins important (0 => *)
- (1 => **)
- (2 => ***)
- (3 => \$\$\$\$)
- (4 => \$\$\$\$\$)
- au plus important (5 => \$\$\$\$\$\$)



La valeur par défaut est 2 (***).

- **URL externe**

- *Clé d'import* : notes_url

Cette propriété est utilisée pour définir une URL qui peut être utilisée pour fournir plus d'information sur ce cluster. N'importe quelle URL valide peut être mise, commençant par *http://*

Cette URL sera disponible dans l'interface de visualisation dans le volet détail présentant le cluster.

Cela peut être pratique si vous désirez faire une description écrite détaillée du cluster, une procédure de reprise sur panne, ... qui sera visible pour les autres membres de l'équipe.

Les caractères < > " ' sont interdits.

  Une URL qui ne commence pas par `http://` ou `https://` provoquera un compteur d'erreur et il sera impossible d'enregistrer le check

Les caractères " et les espaces sont interdits dans ce champ .

 Un caractère non autorisé provoquera un compteur d'erreur et il sera impossible d'enregistrer le check

- **Liste des URL externes**

- Clé d'import : `notes_multi_url`

Cette propriété est utilisée pour définir plusieurs URL permettant de fournir plus d'informations sur ce cluster (une description écrite détaillée de l'hôte, une procédure de reprise sur panne, etc ...).

Ces URL seront disponibles dans l'interface de Visualisation (les listes et dans le volet détail présentant le cluster).

La définition d'une URL est composée de 4 parties :

- (1) Du nom de l'URL (pour l'affichage dans l'interface de Visualisation l'URL)

 Les caractères <> & " ' / sont interdits dans ce champ .

 Un caractère non autorisé provoquera un compteur d'erreur et il sera impossible d'enregistrer le check

- (2) De l'adresse de l'URL (doit commencer par `http://` ou `https://`)

 Les caractères " et les espaces sont interdits dans ce champ .

 Un caractère non autorisé provoquera un compteur d'erreur et il sera impossible d'enregistrer le check

- (3) D'une icône (permettant de reconnaître l'URL dans l'interface de Visualisation)
- (4) Du type d'affichage
 - Caché
 - Ouverture dans un nouvel onglet
 - Ouverture dans un popup interne à Shinken
- (5) Valeur à copier-coller dans un fichier `cfg` pour retrouver cette propriété dans l'état

Cela peut être pratique si vous désirez faire une description écrite détaillée de l'hôte du cluster, une procédure de reprise sur panne, ou toute autre information qui sera visible pour les autres membres de l'équipe.

Les caractères < > " ' sont interdits.

 Une URL qui ne commence pas par `http://` ou `https://` provoquera un compteur erreur

 Un caractère non autorisé provoquera un compteur d'erreur

Liste des URL externes

Documentation

1

http://doc.shinken-solutions.com

2



3

Nouvel onglet

Bug track

http://bug-tracker.com



Popup 80%

Format fichier Cfg

notes_multi_url Documentation~=http://doc.shinken-solutions.com~=documentation~=OPEN_IN_NEW_TAB~#~Bug track~=http://bug-tracker.com~=bug~=POPUP(80)

5

Une URL externe peut contenir le mot clé ##USER## qui sera remplacé par le nom de l'utilisateur courant dans l'interface de Visualisation



Dans cette propriété, des données (macro) de ce cluster peuvent être utilisées dans l'URL.

- La référence à ces données se fait via le principe de [LES VARIABLES \(Remplacement dynamique de contenu - Anciennement les MACROS \)](#).
- Pour intégrer une donnée dans l'URL, il suffit de rajouter \$_HOSTDATA1\$
 - HOSTUUID

Les données (macros) qui peuvent être utilisées sont les suivantes:

- Données globales
- Données accrochées au cluster présente dans l'onglet Données
- et les propriétés (données statiques) suivantes:
 - HOSTUUID
 - HOSTNAME
 - HOSTDISPLAYNAME
 - HOSTALIAS
 - HOSTADDRESS
 - HOSTREALM
 - HOSTADDRESS

- **Activé**

- Clé d'import : enabled

Permettre que cet élément soit pris en compte lors du passage de Staging en Production.

Cette propriété n'est pas héritable.

- **Pack**

- Clé d'import : pack

Permet de donner un nom commun à un ensemble d'éléments de Shinken. Il sera ainsi possible de filtrer par ce nom commun dans les listes des éléments pour retrouver l'ensemble des éléments.

Cette propriété n'est pas héritable.



Les caractères < > & ' ' / sont interdits dans ce champ.



Un caractère non autorisé provoquera un compteur d'erreur et il sera impossible d'enregistrer l'hôte

Onglet : Données

Cet onglet définit des données qui pourront être utilisées en tant que Utilisation des Données en tant que Variable, notamment à l'utilisation de la [Les commandes](#) des checks attachés à ce cluster. Consulter ces pages pour plus d'information.



Si vous donnez un nom de donnée protégée à l'une de vos données, vous ne pourrez plus modifier ce nom par la suite.

- Vous devrez le supprimer puis le recréer si vous voulez le renommer.

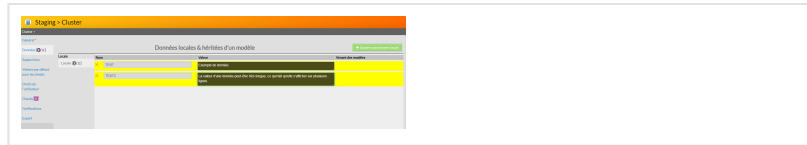
Cette modification est interdite afin d'éviter qu'une donnée protégée devienne visible, en changeant son nom.

La valeur de la donnée pouvant être longue, il est possible d'agrandir le champ de la valeur afin d'améliorer la lisibilité du champ.

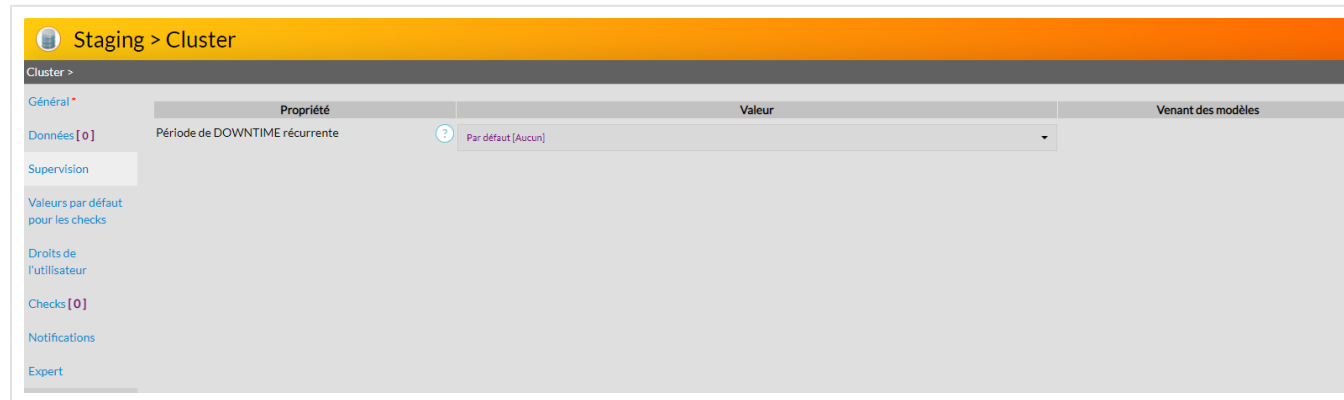
Pour agrandir le champ de la valeur, il faut maintenir le clic sur l'icône  située en bas à droite du champ et réajuster verticalement.



L'agrandissement manuel du champ n'est pas disponible pour le navigateur Internet Explorer. Mais les champs avec des valeurs longues sont automatiquement agrandis jusqu'à une certaine taille lorsque la valeur du champ dépasse une ligne.



Onglet : Supervision

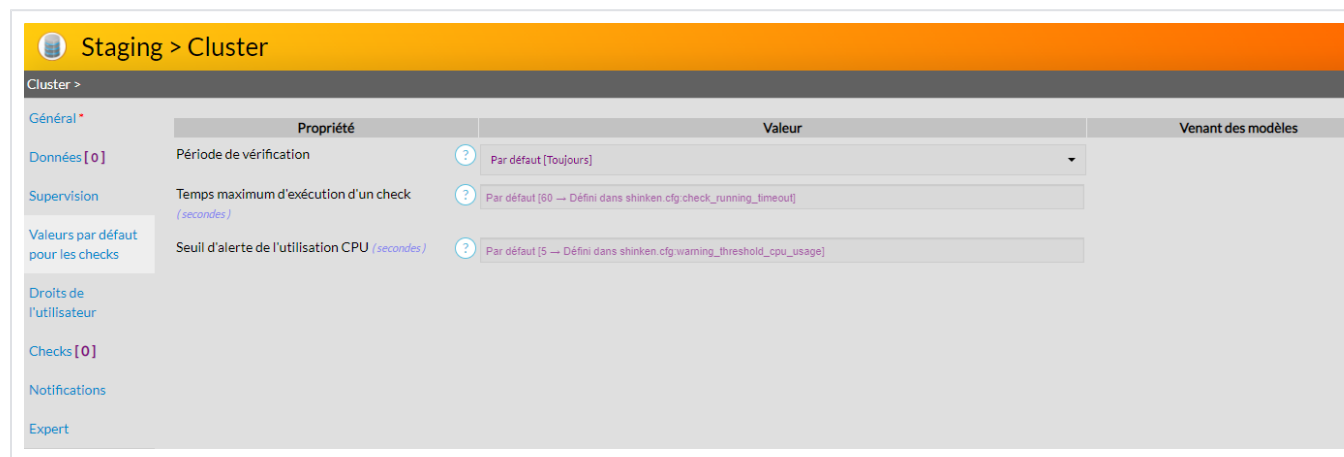


- **Période de DOWNTIME récurrente**
 - Clé d'import : `maintenance_period`

Cette propriété permet de définir une période de temps durant laquelle le cluster sera en maintenance, et en sortira dès que la période de temps sera fini.

En maintenance, les checks du cluster seront toujours exécutés (donc il n'y aura pas de manques dans les métriques ou l'historique du cluster), mais les notifications et les commandes du gestionnaire d'événements ne seront pas lancées.

Onglet : Valeurs par défaut pour les checks



- **Période de vérification**
 - Clé d'import : `check_period`

Cette propriété permet de choisir la période de supervision pendant laquelle sont faites les vérifications de l'état du cluster par Shinken.

- **Temps maximum d'exécution d'un check (secondes)**
 - Clé d'import : `check_running_timeout`

Cette propriété permet de définir la valeur en secondes du temps maximum permis pour le lancement d'une commande de supervision. Passé ce délai, la commande sera terminée. Si le check ou la commande définit également ce paramètre, leur valeur sera prioritaire par rapport à celle du cluster

Aucune valeur ou une valeur à -1, indique qu'il n'y a pas de Temps maximum défini.

Si ni l'hôte, ni le check, ni la commande ne définissent ce paramètre, alors la valeur définie globalement pour Shinken (via le fichier shinken.cfg et le paramètre check_running_timeout) sera utilisée à la place.

 La limite est fixée à 157852800 (soit 5 ans).

 Un caractère non autorisé provoquera un compteur d'erreur

- **Seuil d'alerte de l'utilisation CPU (secondes)**
 - Clé d'import : warning_threshold_cpu_usage

Nombre de secondes que peut consommer une commande avant d'apparaître en Warning dans le check 'Scheduler - Performance'

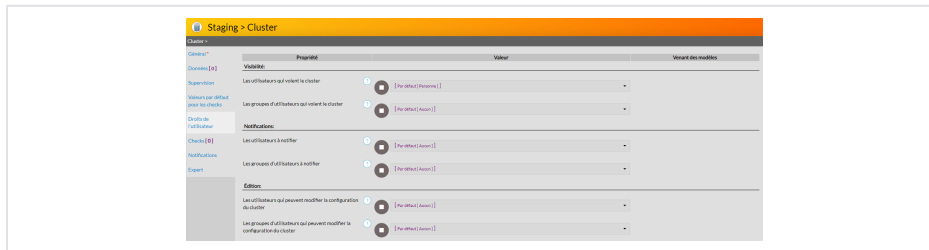
La valeur sera récupérée dans l'ordre, par la commande, par l'élément Check, par l'élément Cluster, ou par la valeur par défaut (shinken.cfg:warning_threshold_cpu_usage)

 La limite est fixée à 157852800 (soit 5 ans).

 Un caractère non autorisé provoquera un compteur d'erreur

Onglet : Droits de l'utilisateur

Il est possible pour chaque cluster de définir qui peut voir, éditer, ou encore recevoir les notifications. Le fonctionnement de ce mécanisme est expliqué dans la page Droits d'accès à un cluster. Ces propriétés gèrent l'[Héritage additif \(le +\)](#).



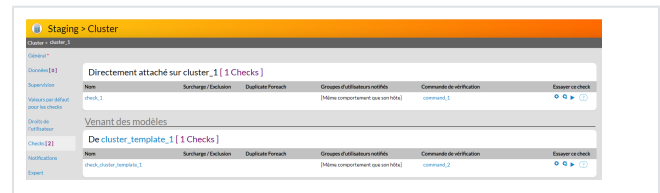
Ces 6 propriétés se synchronisent les unes avec les autres, pour éviter d'avoir des droits aberrants.

Par exemple : avoir le droit d'éditer l'élément, mais sans avoir droit de le voir.

Onglet : Checks

Cet onglet détaille la liste des checks qui seront appliqués au cluster, et leur provenance (venant de quel modèle de cluster, ou directement appliqué au cluster) :

- Tout d'abord les checks accrochés directement sur le cluster sont listés.
- Ensuite sont listés les modèles dans l'ordre de leur apparition dans la liste des modèles hérités.
 - Et pour chacun leurs checks.
 - Seront aussi listés les checks duplication par la fonctionnalité permettant de Dupliquer pour chaque valeur de la Donnée du cluster. Ici, dans l'image, un seul check utilise des Duplicate Foreach (la colonne Duplicate Foreach a une valeur pour ce check).
- Sur cet onglet vous pouvez surcharger les valeurs des checks (voir [La surcharge des propriétés pour un check](#)) et exclure les checks que vous ne voulez pas appliquer sur ce cluster (voir [L'exclusion des checks](#))



Pour chaque check, vous pourrez essayer le check depuis sa configuration actuelle afin de vérifier son résultat.

Essayer les checks

Les boutons dans la colonne [Essayer ce check] permettent d'évaluer ou d'essayer directement l'exécution d'un check, avec la résolution de ses données.

- Le premier bouton (roue crantée) permet de simuler la résolution des arguments, sans l'exécution du check. Cette évaluation vous permettra d'afficher la commande avec les données interprétées.

- Le second bouton (roue crantée + icône play) permet d'évaluer et de simuler son exécution depuis la plateforme de configuration (Synchronizer). Cette exécution n'utilisera pas vos pollers. Vous pouvez donc utiliser ce bouton pour tester votre commande sans affecter vos serveurs pollers en production. Le tableau récapitulatif présente les données récupérées, et le résultat de la commande en prenant en compte les éventuelles modulations.

check 1

(Même comportement que son hôte) command_1:shinken:testing

Ligne de commande:

```
AnsibleRunningshellcheck_dummy $ARG1$ $ARG2$
```

Evaluation:

Arguments	Valueur à évaluer	Valueur obtenue	Trouvé dans (type)
ARG1	shinken	shinken	Arguments de la commande
ARG2	testing	testing	Arguments de la commande

Ligne de commande avec les données interpolées:

```
AnsibleRunningshellcheck_dummy shinken testing
```

Exécution sur la plateforme de configuration (test) (synchronizer-master):

Statut	Résultat	Exécuté dans un Shell	Temps d'exécution
OK	check_dummy: Arguments to check_dummy must be an integer (usage: check_dummy [optional test])	faux	0.026

Modulation du résultat: Une modulation du check a été utilisée

Statut trouvé: OK (shinken) Expression régulière sur le texte du résultat: / / Statut modulé: / /

Exécution avec Modulation:

Statut	Résultat	Exécuté dans un Shell	Temps d'exécution
OK	check_dummy: Arguments to check_dummy must be an integer (usage: check_dummy [optional test])	faux	0.026

- Le troisième bouton (icône play) permet d'évaluer et de simuler son exécution directement sur les Pollers, comme lors de l'exécution normale sur votre architecture Shinken. Vous pouvez donc utiliser ce bouton pour tester votre commande sur votre environnement de production. Le tableau récapitulatif présente les données récupérées, et le résultat de la commande en prenant en compte les éventuelles modulations.

check 2

(Même comportement que son hôte) command_1:shinken:testing

Ligne de commande:

```
AnsibleRunningshellcheck_dummy $ARG1$ $ARG2$
```

Evaluation:

Arguments	Valueur à évaluer	Valueur obtenue	Trouvé dans (type)
ARG1	shinken	shinken	Arguments de la commande
ARG2	testing	testing	Arguments de la commande

Ligne de commande avec les données interpolées:

```
AnsibleRunningshellcheck_dummy shinken testing
```

Exécution sur le poller (production) (poller-master):

Statut	Résultat	Exécuté dans un Shell	Temps d'exécution
OK	check_dummy: Arguments to check_dummy must be an integer (usage: check_dummy [optional test])	faux	0.498

Modulation du résultat: Une modulation du check a été utilisée

Statut trouvé: OK (shinken) Expression régulière sur le texte du résultat: / / Statut modulé: / /

Exécution avec Modulation:

Statut	Résultat	Exécuté dans un Shell	Temps d'exécution
OK	check_dummy: Arguments to check_dummy must be an integer (usage: check_dummy [optional test])	faux	0.498

⚠ Si votre check utilise des tags de Poller, l'exécution ne peut avoir lieu que si l'un des Pollers définis dans votre architecture dispose d'un tag de Poller correspondant à celui du check que vous essayez. Si le check est dans un sous royaume, il faut que l'un des Pollers du sous-royaume ait le tag rechercher.

Veuillez consulter la page [Le Poller](#) pour plus d'informations sur les tags de poller.

- Le dernier bouton (croix) permet de cacher le bloc d'essai du check.

ⓘ Afin de tester au mieux vos checks, si une erreur survient pendant l'essai du check, celle-ci vous sera affichée à la place des résultats.

⚠ Les checks ayant une commande de supervision bp_rule ne pourront pas effectuer d'évaluation ou d'essai.

Les checks cachés

Staging > Cluster

Cluster > cluster_1

Général *

Données [2]

Supervision

Valeurs par défaut pour les checks

Droits de l'utilisateur

Checks [2]

Notifications

Expert

Directement attaché sur cluster_1 [1 Checks]

Nom	Surcharge / Exclusion	Duplicate Foreach	Groupes d'utilisateurs notifiés	Commande de vérification	Essayer ce check
check_1 MODULATION			[Même comportement que son hôte]	command_1shinken/testing	

Venant des modèles

De cluster_template_1 [1 Checks]

Nom	Surcharge / Exclusion	Duplicate Foreach	Groupes d'utilisateurs notifiés	Commande de vérification	Essayer ce check
check_cluster_template_1 MODULATION			[Même comportement que son hôte]	command_2	

De cluster_template_2 [0 Checks]

Nom	Surcharge / Exclusion	Duplicate Foreach	Groupes d'utilisateurs notifiés	Commande de vérification	Essayer ce check
check_cluster_template_1 MODULATION Caché Dupliquer					

Certains checks peuvent être affichés en grisé avec le libellé "Caché".

Cette situation se produit lorsque deux checks ayant le même nom sont attachés sur deux modèles de cluster attachés ou bien directement attachés sur le cluster.

Dans l'ordre d'attachement, le premier check sera donc visible et les autres seront cachés (visibles pour l'utilisateur, mais grisé pour qu'il comprenne que seul le premier sera pris en compte et visible dans l'interface de visualisation).

Inverser l'ordre d'héritage de ces modèles de cluster inversera également le statut (caché / actif) des checks.

Onglet : Notifications

Staging > Cluster

Cluster >

Général *

Données [0]

Supervision

Valeurs par défaut pour les checks

Droits de l'utilisateur

Checks [0]

Notifications

Expert

Propriété	Valeur	Venant des modèles
Notifications activées	☒ Vrai ☐ Faux Par défaut Vrai	
Intervalle de temps entre les notifications (minutes)	Par défaut [1440 (= 1 Journée)]	
Période de temps de notification	Par défaut [Toujours]	
Options de notification	Par défaut [d,u,r,f]	
Décalage de la première notification (minutes)	Par défaut [0]	
Escalades	Par défaut [Aucun]	

- **Notifications activées**

- Clé d'import : notifications_enabled

Cette propriété permet de définir si Shinken va envoyer des notifications aux utilisateurs listés comme cibles (Utilisateurs ou Groupes d'utilisateurs) sur ce cluster

Les notifications sont envoyées:

- Premièrement après la fin de la période de nouvelle tentative, une fois que l'état CRITICAL est confirmé.
- ensuite après chaque **intervalle de notification**, exprimé en minutes
- quand le cluster redevient OK

Les valeurs possibles sont:

- Vrai: Shinken va envoyer les notifications
- Faux: Shinken n'enverra pas les notifications

- **Intervalle de temps entre les notifications (minutes)**

- Clé d'import : notification_interval

Cette propriété permet de définir la période de temps durant laquelle les notifications sont autorisées.

En dehors de cette période, aucune notification ne sera envoyée.

Par défaut, il n'y a pas de période de temps, et donc les notifications ne seront jamais bloquées.

 La limite est fixée à 2630880 (soit 5 ans).

 Une valeur non numérique provoquera un compteur erreur

- **Période de temps de notification**

- Clé d'import : notification_period

Cette propriété permet de définir la période de temps durant laquelle les notifications sont autorisées.

En dehors de cette période, aucune notification ne sera envoyée.

Par défaut, il n'y a pas de période de temps, et donc les notifications ne seront jamais bloquées.

- **Options de notification**

- Clé d'import : notification_options

Cette propriété permet de définir quels états et contextes du cluster peuvent envoyer des notifications.

La valeur est une liste d'options séparées par des virgules:

- c: envoie une notification sur un état CRITIQUE
- w: envoie une notification sur un état AVERTISSEMENT
- u: envoie une notification sur un état INCONNU
- d: envoie une notification sur un état CRITIQUE et AVERTISSEMENT
- r: envoie une notification lors d'un rétablissement (signifie que l'état redevient OK)
- f: envoie une notification lors de l'entrée ou la sortie en FLAPPING
- s: envoie une notification lors de l'entrée ou la sortie en DOWNTIME
- n: désactive toutes les notifications

Par défaut, tous les états permettent d'envoyer une notification.

 Une option non existante provoquera un compteur erreur

- **Décalage de la première notification (minutes)**

- Clé d'import : first_notification_delay

Cette propriété permet de définir combien de minutes Shinken doit attendre avant d'envoyer la première notification.

Ce temps additionnel peut être utilisé pour limiter une avalanche de notifications ; en effet, les clusters n'ont pas de gestion HARD /SOFT et leur état est donc susceptible de changer plus fréquemment.

Ce temps additionnel peut être mis à profit par les utilisateurs pour prendre en compte le cluster depuis l'interface de visualisation avant que la notification ne soit envoyée.

Par défaut la valeur est 0, ceci signifie que la première notification sera envoyée sans attendre.

 La limite est fixée à 2630880 (soit 5 ans).

 Une valeur non numérique provoquera un compteur erreur

- **Escalades**

- Clé d'import : escalations

Cette propriété permet de lier ce cluster à une ou plusieurs définitions d'escalade.

Si, au bout d'un certain temps, le cluster n'est toujours pas revenu OK ou pas pris en compte (Contexte ACKNOWLEDGE ou DOWNTIME), la règle d'escalade sera appliquée.

Cette propriété gère l'[Héritage additif \(le +\)](#).

Onglet : Expert

Staging > Cluster

Cluster > cluster_1

Général*	Propriété	Valeur	Venant des modèles
Données [2]	Tag de Reactionner	Par défaut [non taggé]	
Supervision	Détection du FLAPPING activée	☐ Vrai ☐ Faux Par défaut Vrai	
Valeurs par défaut pour les checks	Options de la détection du FLAPPING	Par défaut [o,w,c,u]	
Droits de l'utilisateur	Sortie du Contexte FLAPPING	<input type="range"/> Par défaut 25%	
	Entrée du Contexte FLAPPING	<input type="range"/> Par défaut 50%	
Checks [2]	Gestionnaire d'événements activé	☐ Vrai ☐ Faux Par défaut Faux	
Notifications	Commande lancée par le gestionnaire d'événements	Par défaut [Aucun]	
Expert	Modulations d'impact métier	☐ [Par défaut [Aucun]]	
	Modulations de données	☐ [Par défaut [Aucun]]	
	Modulations de résultats	☐ [Par défaut [Aucun]]	

- **Tag de Reactionner**

- Clé d'import : reactionner_tag

Cette propriété permet de définir le reactionner_tag du cluster.

Toutes les notifications sur le cluster seront exécutées que par les Reactionners qui ont cette valeur dans leur paramètre reactionner_tags.

Par défaut, la valeur de **Tag de Reactionner** est *non taggé*, donc les Reactionners n'ayant aucun reactionner_tag prendront en compte les checks d'un cluster non taggé, car la valeur par défaut pour les Reactionners est aussi *non taggé*.

- **Détection du FLAPPING activée**

- Clé d'import : flap_detection_enabled

Cette propriété permet de définir si la détection du Contexte FLAPPING est active sur le cluster.

Peut être:

- Vrai: active la détection du contexte FLAPPING.
- Faux: désactive la détection du contexte FLAPPING.

- **Options de la détection du FLAPPING**

- Clé d'import : flap_detection_options

Cette propriété permet de définir quel statut du cluster sont pris en compte pour le calcul du % de FLAPPING.

C'est une combinaison de l'un ou de plusieurs valeurs:

- o = statuts OK
- w = statuts WARNING
- c = statuts CRITICAL
- u = statuts UNREACHABLE

Le séparateur est la virgule.

o est obligatoire.

Exemple: o,w,u

⊖ Une option non existante provoquera un compte erreur

- **Sortie du Contexte FLAPPING**

- Clé d'import : low_flap_threshold

Sur les 21 derniers statuts, chaque fois qu'un statut est différent du précédent (de OK à Warning par exemple), le % de FLAPPING augmente. Donc 10 changements représenteront un % de flapping de 50% et 20 représenteront 100%.

Si ce % calculé est **inférieur** au **% de sortie du Contexte FLAPPING**, alors le Contexte de l'hôte ne sera plus FLAPPING.

- **Entrée du Contexte FLAPPING**

- *Clé d'import* : high_flap_threshold

Sur les 21 derniers statuts, chaque fois qu'un statut est différent du précédent (de OK à Warning par exemple), le % de FLAPPING augmente. Donc pour 10 changements, cela représentera un % de FLAPPING de 50% et pour 20, cela représentera 100%.

Si ce % calculé est **supérieur** au % d'entrée dans le Contexte **FLAPPING**, alors le Contexte de l'hôte deviendra FLAPPING.

Il sortira de ce Contexte quand ce pourcentage calculé sera **inférieur** au % de sortie du Contexte **FLAPPING**.

- **Gestionnaire d'événements activé**

- *Clé d'import* : event_handler_enabled

Cette propriété permet de définir si Shinken va lancer une commande (définie par le paramètre **commande de gestionnaire d'évènements**) à des étapes spécifiques du statut du cluster:

- quand l'hôte passe d'un état OK à un état à problème (CRITICAL ou UNKNOWN)
- à chaque vérification additionnelle, donc un nombre **maximum de confirmation du statut**

Si la propriété est:

- Vrai: va lancer la commande quand nécessaire
- Faux: ne va pas lancer cette commande

- **Commande lancée par le gestionnaire d'événements**

- *Clé d'import* : event_handler

Cette propriété permet de définir la commande que lancera le gestionnaire d'évènements pour cet cluster.

- **Modulations d'impact métier**

- *Clé d'import* : business_impact_modulations

Cette propriété permet de définir une ou plusieurs modulations d'impact métier. Les modulations ont une période temps durant laquelle elles sont actives.

Pendant cette période, la valeur d'impact métier de l'hôte sera changée par celle de la modulation.

Cette propriété gère l'[Héritage additif \(le +\)](#).

- **Modulations de données**

- *Clé d'import* : macromodulations

Cette propriété permet de définir une ou plusieurs modulations de données. Les modulations ont une période temps durant laquelle elles sont actives.

Pendant cette période, les données de l'hôte seront changées par celle de la modulation.

Cette propriété gère l'[Héritage additif \(le +\)](#).

- **Modulations de résultats**

- *Clé d'import* : resultmodulations

Cette propriété permet de définir une ou plusieurs modulations de résultats. (maximum 4)

Les modulations de résultats redéfinissent le statut de sortie du cluster, en fonction de son statut initial, d'une période de temps, ou de sa sortie.

Cette propriété gère l'[Héritage additif \(le +\)](#).