

Commandes de notification (création et personnalisation)

Contexte

Lorsque des objets supervisés changent d'état, et qu'ils sont paramétrés pour réagir à ce changement, alors le **daemon Reactionner** permet d'exécuter une réaction.

Une réaction est une commande qui est paramétrée dans Shinken.

La réaction peut être :

- une commande de notification
- une commande du [gestionnaire d'évènements](#)

Dans cette page, **nous allons nous intéresser aux commandes de notification, et comment les personnaliser.**



Important

Avant de commencer à lire cette page, il est important de comprendre que la commande de notification peut utiliser n'importe quels scripts placés sur votre Reactionner. Donc, potentiellement, la commande peut vous alerter de manière traditionnelle, en vous envoyant un email via une commande "mail", mais pourquoi pas en envoyant un message instantané via votre serveur Jabber, ou encore en faisant clignoter une LED branchée sur votre Raspberry!

- Contexte
- Création d'une commande de notification
 - Objets à configurer
 - Corps de la commande
 - Notations globales
 - Notations liées à l'élément
 - Notations liées au contact
 - Notations de notification
- Exemple de personnalisation d'une commande de notification
 - Méthode de notification "email"
 - Commande
 - Personnalisation
- Troubleshooting
 - Tester votre commande
 - Logs
 - Configuration avancée (relais SMTP)

Création d'une commande de notification

Objets à configurer

Le principal objet à configurer est la [Commande](#) qui lancera la notification personnalisée.

Elle doit être lancée avec les bons paramètres afin de pouvoir générer une notification détaillée et lisible.

Il faudra ensuite s'assurer de l'utiliser dans une [Méthode de notification](#). Lorsqu'elle sera lancée, elle pourra accéder aux données de l'hôte ou du service à l'origine de la notification.

Corps de la commande

Les commandes sont exécutées par le [Reactionner](#) lorsque les conditions réunies par la [Logique de notification](#) sont réunies.

Elles ont accès spécifiquement à un certain nombre de [notations de remplacement dynamique de contenu](#) (MACRO).

Ces données dynamiques peuvent venir de différents endroits :

- Une donnée globale
- L'élément à l'origine de la notification.
- Le contact à notifier
- La notification en elle-même, notamment son type.

Toutes les macros disponibles sur l'élément à l'origine de la notification et le contact à notifier sont disponibles pour la notification.

Notations globales

Ces [notations de remplacement dynamique de contenu](#) permettent de paramétrer globalement le comportement des notifications.

Toutes les notations globales peuvent être utilisées indifféremment dans une commande de notifications, mais les notations suivantes leurs sont spécifiques :

Notation	Valeurs possibles	Utilisation
\$MAILURL\$	< une URL >	L'adresse de l'UI de visualisation de Shinken. (dans le but par exemple de donner au contact un lien vers l'élément).
\$SENDER\$	< une adresse émetteur >	L'adresse mail de l'envoyeur, pour une notification mail.
\$NOTIFPLUGINDIR\$	< un chemin de fichiers >	Le répertoire dans lequel se trouve la commande de notification à exécuter.

Remarque

Ces trois notations spécifiques sont mises à des valeurs par défaut que vous pouvez retrouver dans le fichier du répertoire resource.d : `/etc/shinken/resource.d/email.cfg`

Notations liées à l'élément

Les [notations de remplacement dynamique de contenu](#) de l'élément peuvent être appelées dans la notification.

Dans le cas d'une notification d'un host, les notations du host sont disponibles.

Dans le cas d'une notification d'un check, les notations du host et du check sont toutes les deux disponibles.

De la même façon que les notations globales, toutes les notations des éléments peuvent être utilisées, mais les notations suivantes sont spécifiquement utiles :

Notation	Valeurs possibles	Utilisation
\$HOSTNAME\$	< un nom d'hôte >	Le nom de l'hôte.
\$SERVICEDESC\$	< un nom de check >	Le nom du check, dans le cas d'un check.
\$HOSTADDRESS\$	< une adresse >	L'adresse du host.
\$_HOSTID\$, \$_SERVICEID\$	< un id >	L'identifiant Shinken de l'élément.
\$HOSTSTATE\$, \$SERVICESTATE\$	< un status, format texte >	Le status de l'élément : UP, DOWN, UNREACHABLE pour un host ou OK, WARNING, CRITICAL, ou UNKONWN pour un check.
\$LASTHOSTSTATEID\$, \$LASTSERVICESTATEID\$	< un status, format numérique >	Le status précédent de l'élément, par code de retour (0, 1, 2 ou 3).
\$LASTHOSTSTATECHANGE\$, \$LASTSERVICESTATECHANGE\$	< temps unix, flottant >	La date du dernier changement de status de l'élément.
\$DATE\$	< date, format dd/mm/yyyy >	La date du dernier check.
\$TIME\$	< heure, format hh:mm:ss >	L'heure du dernier check.
\$HOSTOUTPUT\$, \$SERVICEOUTPUT\$	< un output >	La sortie du check à l'origine de la notification.

\$LONGHOSTOUTPUT\$, \$LONGSERVICEOUTPUT\$	< un output >	La sortie longue du check à l'origine de la notification.
\$_HOSTMADONNEE1\$	< un texte >	La donnée MADONNEE1 de l'hôte
\$_SERVICEMADONNEE2\$	< un texte >	La donnée MADONNEE2 du check

Notations liées au contact

Les notations du contact pouvant être appelées dans la notification.

Notations	Valeurs possibles	Utilisation
\$CONTACTPAGER\$	< un numéro de téléphone >	Le numéro de téléphone du contact, pour une notification téléphonique.
\$CONTACTEMAIL\$	< une adresse destinataire >	L'adresse mail du destinataire, pour une notification mail.
\$CONTACTADDRESS1\$, [...] \$CONTACTADDRESS6\$	< les adresses >	Adresses supplémentaires du contact pour d'autres types de notifications
\$_CONTACTMADONNEE1\$	< un texte >	La donnée MADONNEE1 du contact
\$_CONTACTMADONNEE2\$	< un texte >	La donnée MADONNEE2 de contact

Notations de notification

Des notations spéciales permettent d'avoir des données concernant la notification en elle-même et la raison pour laquelle elle a été envoyée.

Notations	Valeurs possibles	Utilisation
\$NOTIFICATIONTYPE\$	< un type de notification >	Le type de notification à envoyer. Cela correspond au type d'événement qui a été constaté sur l'élément.
\$ACKAUTHOR\$	< un nom de contact >	Pour le cas d'un acknowledge, la personne à l'origine de l'acknowledge.
\$ACKDATA\$	< un message texte >	Pour le cas d'un acknowledge, le message de prise en compte.

Pour les types de notifications, la liste des valeurs possibles est la suivante :

Type	Événement source
PROBLEM	Le status de l'élément est non-OK.
RECOVERY	L'élément avait un problème, mais est de nouveau dans un status OK.
ACKNOWLEDGEMENT	Un utilisateur a envoyé par l'interface web un accusé de réception par rapport à un problème survenu à un hôte ou à un check.
FLAPPINGSTART, FLAPPINGSTOP	L'élément est entré ou sorti d'un contexte de FLAPPING.
FLAPPINGDISABLED	La détection a été désactivée pendant la durée du FLAPPING.
DOWNTIMESTART, DOWNTIMESTOP	L'élément est entré ou sorti d'une période programmée d'indisponibilité.
DOWNTIMECANCELLED	La période programmée d'indisponibilité de l'élément a été annulée en cours.

Exemple de personnalisation d'une commande de notification

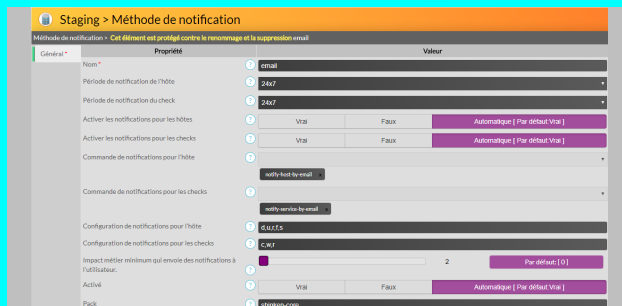
Nous allons ici vous présenter un exemple de personnalisation de la commande de notification par défaut de Shinken.

Le but ici est de modifier la commande afin que la notification par email utilise un serveur SMTP de votre choix. En effet, par défaut, la commande utilise comme relais SMTP le processus Postfix du serveur qui héberge le daemon Réactionner (serveur **localhost**).

Méthode de notification "email"

La méthode de notification incluse par défaut dans Shinken est "email". Cette méthode est d'ailleurs protégée contre le renommage et la suppression.

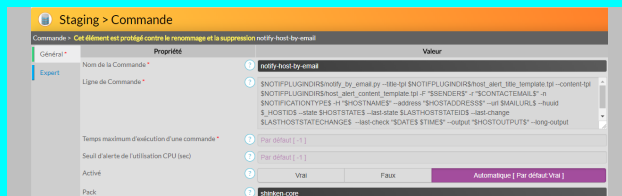
Cette méthode de notification utilise les commandes **notify-host-by-email** et **notify-service-by-email**.



Commande

Voici la ligne de commande utilisée par l'objet commande **notify-host-by-email** :

```
$NOTIFPLUGINDIR$/notify_by_email.py --title-tpl  
$NOTIFPLUGINDIR$/host_alert_title_template.tpl --  
content-tpl $NOTIFPLUGINDIR$/  
host_alert_content_template.tpl -F "$SENDER$" -r  
"$CONTACTEMAIL$" -n $NOTIFICATIONTYPE$ -H  
"$HOSTNAME$" --address "$HOSTADDRESS$" --url  
$MAILURL$ --huuid $_HOSTID$ --state $HOSTSTATE$ --  
last-state $LASTHOSTSTATEID$ --last-change  
$LASTHOSTSTATECHANGE$ --last-check "$DATE$ $TIME$"  
--output "$HOSTOUTPUT$" --long-output  
"$LONGHOSTOUTPUT$" --ack-author "$ACKAUTHOR$" --ack-  
data "$ACKDATA$"
```



Cette commande est localisée dans le répertoire par défaut **/var/lib/shinken/libexec/notifications/** (**\$NOTIFPLUGINDIR\$/**).

Le script utilisé est un script Python qui prend un certain nombre d'arguments. Comme vous pouvez le constater, la ligne de commande fait référence à de nombreuses **notations de remplacement dynamique de contenu** afin d'envoyer les valeurs représentatives à l'instant T de la notification liée à l'hôte supervisé.

Personnalisation

Ce script Python **notify_by_email.py** peut utiliser un argument **--SMTP** ou **-S** qui permet alors de passer en paramètre l'IP ou l'adresse d'un serveur SMTP pour l'envoi de l'email.

Admettons par exemple que votre serveur SMTP est le serveur 192.168.1.200, il suffit donc de rajouter l'information **--SMTP 192.168.1.200** dans votre ligne de commande :

```
$NOTIFPLUGINDIR$/notify_by_email.py --title-tpl $NOTIFPLUGINDIR$/host_alert_title_template.tpl --content-tpl  
$NOTIFPLUGINDIR$/host_alert_content_template.tpl -F "$SENDER$" -r "$CONTACTEMAIL$" -n $NOTIFICATIONTYPE$ -H  
"$HOSTNAME$" --address "$HOSTADDRESS$" --url $MAILURL$ --huuid $_HOSTID$ --state $HOSTSTATE$ --last-state  
$LASTHOSTSTATEID$ --last-change $LASTHOSTSTATECHANGE$ --last-check "$DATE$ $TIME$" --output "$HOSTOUTPUT$" --  
long-output "$LONGHOSTOUTPUT$" --ack-author "$ACKAUTHOR$" --ack-data "$ACKDATA$" --SMTP 192.168.1.200
```



Remarque

Au lieu de rajouter l'adresse en dur dans la ligne de commande, pour pourriez également utiliser une notation de remplacement dynamique de contenu, référant à une donnée de l'hôte par exemple, comme : **\$_HOSTIPSMTPSERVEURS**
La donnée **IPSMTPSERVEUR** devra alors être présente sur l'objet hôte, directement, ou via un modèle par exemple.

Troubleshooting

Tester votre commande

Si vous souhaitez envoyer une commande depuis votre Réactionner afin de vérifier le bon envoi de l'email, loguez vous en shinken via :

```
su - shinken
```

puis envoyer votre commande, par exemple :

```
/var/lib/shinken/libexec/notifications/notify_by_email.py --title-tpl /var/lib/shinken/libexec/notifications/service_alert_title_template.tpl --content-tpl /var/lib/shinken/libexec/notifications/service_alert_content_template.tpl -F "serveur-reactionner@shinken-solutions.com" -r "d.labardin@shinken-solutions.com" -n PROBLEME -H "mon-serveur" --address "172.16.0.10" --url http://172.16.0.10:7767 --huuid 7b0513f631a011e889e9080027da5b5c --check "Memory" --state CRITICAL --last-state 0 --last-change 1525338011.76 --last-check "03-05-2018 11:00:56" --output "Critical : memory consumption is too high 66%" --long-output "" --ack-author "" --ack-data ""
```

Vous devriez alors recevoir :

```
DATE,426:INFO: Mail sent successfully
```

Et vous devriez recevoir l'email dans votre boîte email.

Logs

Si vous avez des difficultés, veuillez vérifier les logs :

- du Réactionner dans **/var/log/shinken/**
- de votre serveur mail linux (postfix/smtp) dans **/var/log/maillog**

Configuration avancée (relais SMTP)

Si votre serveur SMTP principal est un serveur connu et généralisé dans votre architecture, vous pouvez aussi modifier votre configuration afin de relayer n'importe quel email vers ce serveur :

- Editer le fichier **/etc/postfix/main.cf**
- Modifier la ligne avec l'ip de votre serveur SMTP (ici 192.168.1.240) : **relayhost = 192.168.1.240**
- Redémarrer postfix
 - service postfix restart
- test d'envoi de mail :
 - ls | mailx -s test d.labardin@shinkendom.local

