



# Supervision d'environnement virtuel



URL: <https://pandorafms.com/manual!/current/>  
Permanent link:  
[https://pandorafms.com/manual!/current/fr/documentation/pandorafms/monitoring/05\\_virtual\\_environment\\_monitoring](https://pandorafms.com/manual!/current/fr/documentation/pandorafms/monitoring/05_virtual_environment_monitoring)  
24/06/10 14:36





# Supervision d'environnement virtuel

Pour surveiller [Amazon EC2®](#), [DB2®](#), [SAP®](#) and [VMware®](#) les éléments suivants sont utilisés [Discovery PFMS®](#).

## RHEV

Red Hat® Enterprise Virtualization (RHEV) est une des technologies les plus utilisées par les entreprises qui ont comme base le système d'exploitation Red Hat dans son Data Center. Pandora FMS offre la possibilité de superviser les architectures virtuelles basées sur RHEV par le biais du plugin RHEV Monitoring Plugin, qui permet contrôler de manière simple tous les aspects liés à l'architecture virtuelle RHEV.

### Architecture à superviser

- Data Centers.
- Host Clusters.
- Storage Domains.
- Hosts.
- Virtual Machines.

Pour ça, Pandora FMS utilise l'API officielle fournie par le système de virtualisation RHEV.

### Supervision avec RHEV Monitoring Plugin

Pour la supervision du système d'exploitation installé dans les machines virtuelles, il est recommandé d'utiliser un Agent de Pandora FMS au lieu de l'API RHEV.

La supervision d'environnements virtuels RHEV se base dans ces composants:

1. Un *plugin* d'Agent qui fait les tâches de auto-découverte et collecte de données. Le *plugin* d'Agent est chargé d'envoyer les informations vers Pandora FMS.
2. Un *script* de reconnaissance qui met à jour de différents valeurs pour les entités découvertes, ce script est nécessaire pour le correcte fonctionnement des extensions du *plugin*.
3. Extension RHEV Viewer et RHEV Manager. Ils sont des extensions qui ajoutent une valeur supplémentaire permettant de visualiser l'infrastructure supervisée et opérer en allumant/étendant les machines virtuelles, tout ça depuis la console web de Pandora FMS.

Pour pouvoir utiliser le *script* de reconnaissance il est nécessaire d'avoir activé le [Discovery server](#). Afin que

certaines variables de l'API reflètent la valeur réelle de la machine virtuelle associée, il est nécessaire d'installer l'agent RHEV ; vous pourrez trouver tout regardant la documentation de sa version RHEV.

### Fonctionnement interne du plugin

Le plugin de surveillance RHEV extrait les informations via l'API web servie par l'environnement de virtualisation RHEV.

Si vous avez seulement besoin de l'information de supervision, vous avez seulement à configurer le *plugin* de l'agent qui fera cette tâche. La configuration du *plugin* permet de choisir quels éléments seront superviser et la configuration de ses modules. Une fois créés les XML, le *plugin* d'agent envoie les fichiers, soit grâce à **Tentacle** ou en les copiant vers un répertoire local, selon la méthode de transfert choisie.

Si en outre vous allez utiliser les extensions RHEV Viewer et RHEV Manager vous aurez besoin d'utiliser le script de reconnaissance. Le script de reconnaissance est chargé de mettre à jour quelques variables de chacun des agents détectés dans Pandora FMS selon les valeurs configurés dans RHEV. Ces variables sont nécessaires pour visualiser les entités correctement dans l'extension RHEV Viewer et gérer correctement les machines virtuelles avec l'extension RHEV Manager.

### Conditions préalables à l'installation du plugin RHEV

- curl.
- perl-XML-Simple.
- **Agent Logiciel PFMS** et `tentacle_client` de Pandora FMS.

### Red Hat

Vous pouvez installer les dépendances dans les systèmes fondés sur Red Hat® avec la commande:

```
yum install perl-XML-Simple curl
```

### SLES

Dans les systèmes fondés sur SUSE, vous pouvez installer les dépendances avec la commande:

```
zypper install perl-XML-Simple curl
```

## Debian/Ubuntu

Dans les systèmes fondés sur Debian/Ubuntu, vous pouvez installer les dépendances avec la commande:

```
apt-get install libxml-simple-perl curl
```

## Téléchargement du certificat RHEV

Avant d'utiliser le plugin, il sera nécessaire de télécharger le certificat qui permet la connexion par moyen de HTTPS à l'API de RHEV. Pour ça, exécutez la commande suivante:

```
curl -o rhevm.cer http://[RHEVM-HOST]:8080/ca.crt
```

Où [rhevm-host] est le nom du serveur auquel l'API de RHEV fournit du service. Un exemple pourrait être:

```
curl -o rhevm.cer http://rhevm.server:8080/ca.crt
```

Après le certificat est téléchargé, vous pouvez vérifier que la connexion à l'API se fait de manière correcte par le biais de l'utilisation des connecteurs de ligne \:

```
curl -X GET \  
      -H "Accept: application/xml" \  
      -u [USER:PASS] \  
      --cacert [CERT] https://[RHEVM-HOST]:8443/api
```

Avec les valeurs suivantes :

- USER: *user @domain* pour se connecter à l'API.
- PASS: mot de passe de l'utilisateur avec lequel vous vous connecterez à l'API.
- CERT: le chemin d'accès au certificat téléchargé à l'étape précédente.
- RHEVM-HOST: adresse de l'hôte qui fournit du service à l'API.

Exemple avec des données concrètes de la commande:

```
curl -X GET \  
      -H "Accept: application/xml" \  
      -u [user@testdomain:12345] \  
      --cacert /home/user/ca.crt https://rhevm.server:8443/api
```

Si l'exécution de la commande est positive, il retournera une sortie en format XML avec l'information général sur l'API de RHEV.

## Considerations préalables sur la configuration RHEV

Dans l'environnement de virtualisation RHEV il se peut que quelques entités aient le même nom. Cela suppose un problème, puisque dans Pandora FMS ces entités se convertissent en des agents qui ne permettent pas la duplication des noms. En outre il y a aussi des problèmes lors de la correction avec des patches le résultat retourné par l'API en format XML, vous montrant un erreur pareille à celui-ci :

```
Warning: <data_center> element has non-unique value in 'name' key attribute:
Default at ./plugin-rhev.pl line 199
```

Afin de résoudre le problème, ce dont vous avez besoin est de suivre une nomenclature de noms pour les entités dans l'environnement de virtualisation RHEV dans lequel les noms ne se répètent pas.

## Installation du plugin d'agent

Afin d'installer le plugin d'agent vous n'avez qu'à copier le *script* `rhev-plugin.pl` et le fichier de configuration `rhev-plugin.conf` dans un répertoire de la machine où l'agent Pandora FMS qui exécutera le plugin est installé. Le *plugin* peut être exécuté dans un agent installé dans les mêmes machines que le serveur Pandora FMS ou dans une autre machine.

Pour exécuter le plugin, ajoutez le fichier de configuration de l'agent (par défaut `/etc/pandora/pandora_agent.conf`) la ligne suivante:

```
module_plugin /root/rhev-plugin.pl /root/rhev-plugin.conf
```

En ajoutant cette ligne, le plugin de l'agent fera ses fonctions dans chaque exécution.

## En supervisant l'architecture virtuelle RHEV

Afin de voir le résultat de l'exécution du plugin d'agent, allez vers [Operation](#) → [Monitoring](#) → [Views](#) → [Agent Detail](#). Lorsque vous cliquez sur le nom de un agent, vous pourrez voir les modules de supervision créés par le plugin, outre d'autres données relatives.

Le plugin crée un agent dans Pandora FMS pour chacune des entités détectées lors de la découverte de l'architecture RHEV. Pour chaque type d'entité, une série de modules déterminés sont automatiquement créés, surveillant les informations importantes de chacun d'entre eux.

Si l'agent sélectionné correspond à un hôte et non à une machine virtuelle, les modules de surveillance seront différents.

Le plugin RHEV surveille également les événements qui se produisent au sein de l'architecture virtuelle. Le plugin crée un module pour chaque événement surveillé au sein de chaque entité concernée. Les données des modules créés à partir des événements sont les suivantes : heure de l'événement, description de l'événement.

En plus des agents et des modules liés à l'architecture RHEV elle-même, un module appelé, par défaut, RHEV Plugin, est généré dans l'agent qui exécute le plugin.

## Modules d'agent de l'architecture virtuelle RHEV

### Data Center

- *Status*: État du Data Center.

### Storage Domain

- *Available Space* : Espace disponible dans le Storage Domain.
- *Committed Space* : Espace compromis dans le Storage Domain.
- *Used Space* : Espace utilisé dans le Storage Domain.
- *Percent Free Space* : Pourcentage d'espace libre dans le Storage Domain.

### Network

- *Status* : État du réseau virtuel.
- *STP Status* : État de la fonctionnalité Spanning Tree Protocol.

### Cluster

- *Overcommit Percent* : Pourcentage de surallocation du *cluster*.
- *Transparent HugePages* : État de la fonctionnalité Transparent HugePages.
- *High threshold* : Limite supérieure dans les politiques de planification.
- *Low threshold* : Limite inférieure des politiques de planification.
- *Threshold duration* : Durée des limites dans les politiques de planification.

### Host

- *Status* : État de l'hôte.
- *Buffers size* : Taille des *buffers*.
- *Cache size* : Taille du *cache*.
- *Cached swap* : Quantité de mémoire Swap *caching* (en octets).
- *Free memory* : Quantité de mémoire libre (en octets).
- *Percent free memory* : Pourcentage de mémoire libre.
- *Swap cached percent* : Pourcentage de mémoire Swap *caching*.
- *Swap free* : Quantité de mémoire Swap libre (en octets).
- *Swap free percent* : Pourcentage de mémoire Swap libre.

- *Total Memory* : Quantité de mémoire de l'hôte (en octets).
- *Total Swap* : Quantité totale de mémoire Swap (en octets).
- *Used memory* : Quantité totale de mémoire utilisée (en octets).
- *Used Swap* : Quantité totale de mémoire Swap utilisée (en octets).
- *Nic [x] TX* et *Nic [x] RX* : Ratio de transfert et réception de l'interface du réseau [x] (en octets/seconde). Un est généré par chaque interface réseau détectée.
- *Nic [x] error TX* et *Nic [x] error RX* : Nombre d'erreurs de transmission et réception de l'interface réseau [x]. Un est généré par chaque interface réseau détectée.
- *User CPU* : Pourcentage d'UCT utilisé par l'utilisateur.
- *System CPU* : Pourcentage d'UCT utilisée par le système.
- *CPU Idle* : Pourcentage d'UCT sans utiliser.
- *CPU Load* : Charge moyenne d'UCT des derniers 5 minutes.
- *KSM CPU* : Pourcentage d'UCT utilisé par KSM.
- *Active VM* : Nombre de machines virtuelles actives sur l'hôte.
- *Migrating VM* : Nombre de machines virtuelles dans la migration dans l'hôte.
- *Total VM* : Nombre totale de machines virtuelles dans l'hôte.
- *Fence Status* : État du de l'hôte.

## Virtual Machine

- *Status* : État de la machine virtuelle.
- *Disk [x] read* : Taux de lecture du disque x (octets/seconde). Un est généré par chaque disque (stockage) détecté.
- *Disk [x] write* : Taux d'écriture du disque x (octets/seconde). Un est généré par chaque disque détecté.
- *Disk [x] size* : Taille du disque x (en octets). Un est généré par chaque disque détecté.
- *Disk [x] status* : État du disque x. Un est généré par chaque disque détecté.
- *Nic [x] TX* et GitLab PFMS 12929 .: Ratio de transfert et réception pour l'interface réseau [x] (en octets/segundo). Un est généré par chaque disque détecté.
- *Nic [x] error TX* et *Nic [x] error RX* : Nombre d'erreurs transmission et réception pour l'interface réseau [x]. Un est généré par chaque disque détecté.
- *Installed memory* : Quantité de mémoire installée (en octets).
- *Percent free memory* : Pourcentage de mémoire libre.
- *Used memory* : Quantité de mémoire utilisée (en octets).
- *Stateless* : État de la fonctionnalité Stateless.
- *HA Status* : État de la fonctionnalité du HA.
- *Total CPU* : Pourcentage totale de l'UCT utilisé par la machine virtuel.
- *Hypervisor CPU* : Pourcentage de l'UCT de l'Hypervisor utilisé par la même machine virtuelle.
- *Guest CPU* : Pourcentage de l'UCT de l'hôte qui utilise la machine virtuelle.

## Événements

- *Event [x]* : Description de l'événement x qui se passe dans le système. Un sera créé par chaque événement détecté dans les agents affectés.

## Gestion et visualisation de l'architecture RHEV



## Tâches de reconnaissance

Il existe la possibilité de créer des tâches personnalisés de reconnaissance grâce au [Discovery server](#).

## Installation d'extensions RHEV View et RHEV Manager

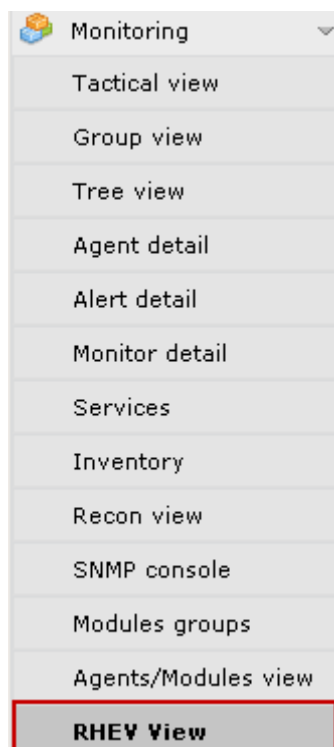
Pour installer les extensions juste copiez le contenu du dossier extensions, que vous trouverez lorsque vous décompressez le plugin, dans le dossier correspondante extensions de la Console Pandora FMS. La commande à exécuter est la suivante:

```
cp -R extensions/* <pandora_console_dir>/enterprise/extensions/
```

À partir de ce moment les extensions de supervision RHEV seront disponibles.

## Utilisation de l'extension RHEV View

Afin d'utiliser l'extension RHEV View juste cliquez sur l'option RHEV View dans le submenu Monitoring.



L'extension vous montrera une carte comme la suivante, avec tous les composants de l'architecture RHEV découverts par le plugin.

## Utilisation de l'extension RHEV Manager

L'extension RHEV Manager est disponible dans la vue d'opération des agents Pandora FMS qui se correspondent avec les machines virtuelles dans l'architecture de virtualisation RHEV.

Cette extension utilise la commande curl, donc il sera nécessaire qu'il soit installé et accessible pour le serveur web qui supporte la Console Pandora FMS.

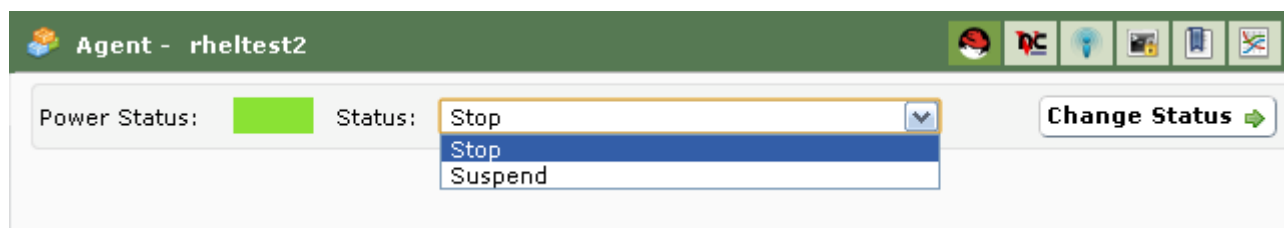
Pour accéder à l'extension, cliquez sur le bouton avec le logo de Red Hat que vous trouverez ensemble avec les autres cils de l'agent.



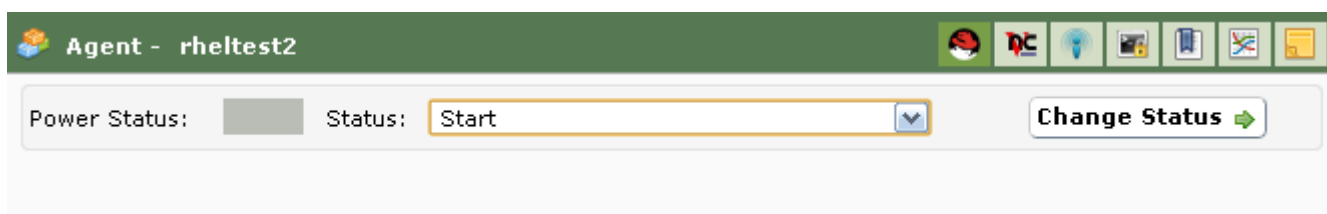
L'extension permet de gérer des machines virtuelles (allumer, éteindre et suspendre) sans besoin d'ouvrir la console de gestion RHEV. Dans l'extension l'état actuel de la machine virtuelle est montré avec un code de couleurs:

- Vert = Allumé.
- Orange = Suspendu.
- Gris = Arrêté.

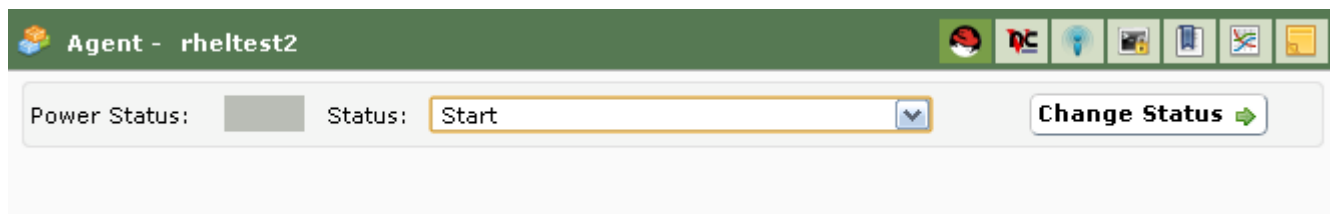
Avec une combinaison des états disponibles et des états dans lesquels la machine virtuelle peut être amenée en appuyant sur le bouton Change Status.



Si vous choisissez l'état Stop pour arrêter la machine virtuelle, l'extension connectera avec l'API RHEV et enverra l'ordre. Le résultat sera le changement d'état dans la machine virtuelle et les options du combo :



Le changement de quelques état n'est pas automatique, comme par exemple de l'état Stop vers Start. Dans ce cas, l'extension montrera l'état de la machine virtuelle selon le changement dans l'architecture de virtualisation. Exemple :



## Configuration du plugin de l'agent

La configuration du plugin de l'agent se fait par le biais d'un fichier de configuration dont le nom par défaut est `rhev-plugin.conf`.

Par défaut, le plugin de l'agent choisit toutes les entités et rée tous les modules correspondants avec des valeurs prédéfinis pour le nom et la descripton. Tous ces aspects, ainsi que les variables générales du plugin, peuvent être configurés à travers le fichier de configuration.

### Fichier de configuration

Le fichier de configuration a deux sections très bien définies: les variables globales et la configuration de supervision.

La section de variables globales commence avec le jeton Configuration et contient les informations de configuration du plugin. Les paramètres autorisés dans cette section:

- *module\_name*: Nom du module de l'agent avec l'état de l'exécution du plugin.
- *server*: Nom de l'hôte qui fournit du service de l'API de RHEV.
- *user*: Utilisateur sous format *user @domain* pour se connecter à l'API.
- *pass*: Mot de passe pour se connecter à l'API.
- *cert*: Chemin du certificat pour se connecter à l'API.
- *temporal*: Répertoire temporel.
- *logfile*: Fichier journal.
- *transfer\_mode*: Mode de transfert. Vous pouvez prendre les valeurs : `local` ou `tentacle`.
- *tentacle\_ip*: Adresse IP du serveur Tentacle auquel envoyer les informations. Normalement il se trouvera dans la même machine que celle du serveur Pandora FMS. Cette option s'utilise seulement si `transfer_mode` a la valeur `tentacle`.
- *tentacle\_port*: Port du serveur Tentacle. Cette option seulement s'utilise si `transfer_mode` a la valeur `tentacle`.
- *tentacle\_opts*: Options d'envoi de données pour Tentacle. Cette option seulement s'utilise si `transfer_mode` a la valeur `tentacle`.

La section de configuration de la supervision se divise en quelques sous-sections. La première sous-section a comme jeton Reject et liste les entités de l'environnement de virtualisation qui seront écartés de la supervision. Afin d'écartier une entité il sera nécessaire d'entrer son nom dans

cette liste. Par exemple:

```
#Dismissed entities
Reject
mv1
mv_Windows10
mv_WebServer1
...
```

Il est possible d'écarter toutes les entités du même type, par exemple toutes les hôtes, toutes les machines virtuelles, etc. Les jetons pour chaque entité sont: all\_dc (Data Center), all\_host (Hosts), all\_network (Networks), all\_storage (Storage Domain), all\_cluster (Cluster), all\_vm (machines virtuelles). Exemple d'utilisation de ces jetons:

```
#Dismissed entities
Reject
all_dc
all_host
all_network
all_storage
all_cluster
all_vm
```

La deuxième section a comme jeton Rename et sert à changer les noms des entités supervisées à travers le plugin. Cette fonctionnalité est très utile si vous voulez combiner la supervision d'agent logiciels avec les données extraits de l'API de Pandora FMS. La configuration de cette section se fait en écrivant au début le nom ancienne et après le nouveau nom séparé par un espace; par exemple:

```
#Rename entities
Rename
mv_WebServer1 WebServer1
mv_Windows10 Windows10 Test
...
```

Les sous-sections suivantes correspondent à la configuration de supervision pour chaque entité. Chaque entité a son propre jeton, qui sont les suivants: DataCenter, StorageDomain, Network, Cluster, Host et VM. Pour chacune des entités, il est possible de définir les modules qui seront désactivés ou définir des valeurs personnalisés pour le nom, la description et les plages de maximums et minimums pour les états Warning et Critical. Un exemple serait le suivant:

```
#VM Modules
VM
status disabled
errors_total_tx name = Errors TX Net [%s]; desc = Total network TX errors;
limits = 60 70 71 100
memory_used name = Memory in use; desc = Memory used by the virtual machine;
limits = 256 1024 1025 2048
```

...

Chaque ligne de configuration des modules de supervision se correspond à deux options disponibles:

- `<module> disabled`: Le module NE se pas créé.
- `<module> nom = <nom>; desc = <description>; limits = <min_warning> <max_warning> <min_critical> <max_critical>` Le module sera créé avec le nom et la description fournis et en outre les seuils pour les maximums et minimums des valeurs `Warning` et `Critical`.

Il est très important d'avoir sur compte l'estructure des lignes de fichier de configuration et surtout de voir que le caractère `;` est à côté du nom et la description u module. Ces deux lignes NE SONT PAS ÉQUIVALENTES (voyez les espaces avant le caractère `;`):

```
errors_total_tx name = Net TX Errors [%s]; desc = Total Network TX Errors;
limits = 60 70 71 100 #Correct
errors_total_tx name = Net TX Errors [%s] ; desc = Total Network TX Errors ;
limits = 60 70 71 100 #Incorrect
```

Les modules sont référencés par son nom court, un nom équivalent plus facile à écrire dans la ligne de commande. La table de correspondance entre les noms courts et étendus se trouve dans la section suivante.

Exemple de configuration pour les machines virtuelles, section VM:

Pour la supervision de machines virtuelles une série de modules activés ou pas définis dans la section VM du fichier de configuration ont été définis. Particulièrement: le module `status` a été désactivé et pour les modules `errors_total_tx` et `memory_used` des valeurs personnalisables ont été définis. Les autres modules qui n'apparaîtront pas dans la liste seront créés avec les valeurs par défaut. Avec cette configuration, le module `memory_used` prendra les valeurs suivantes:

- *Nom* : Mémoire en utilisation.
- *Description* : Mémoire utilisée par la machine virtuelle.
- *Min Warning* : 256.
- *Max Warning* : 1024.
- *Min Critical* : 1025.
- *Max Critical* : 2048.

Les modules se génèrent dynamiquement; par exemple, deux relatifs ais disques ou interfaces dont l'un de chaque élément détecté est créé. Ils ont une syntaxe spéciale pour le nom du module, qui est la suivante:

```
errors_total_tx name = Net TX Errors [%s]; desc = Total Network TX Errors;
limits = 60 70 71 100
```

Dans ces cas, comme le nom a une partie dynamique, ce qui permet d'utiliser la macro `%s` qui sera remplacé par le plugin par la partie variable du nom du module.

Par exemple, le module `errors_total_tx` a un nom par défaut:

```
Nic [nic1] errors TX
```

Il changera son nom par :

```
Errors TX Net [nic1]
```

`nic1` étant la partie dynamique du nom du module.

Tous les erreurs relatifs au fichier de configuration sont présentés dans le log défini dans le fichier de configuration et en outre sont envoyés en tant que module asynchrone vers Pandora FMS qui sera reflété en tant que module dans l'agent qui exécute le plugin.

Outre les sections spécifiques à chaque élément de l'architecture, le fichier de configuration comporte une section commune pour les événements. Cette section est définie par le jeton `EventCodes` et énumère les codes d'événements à surveiller; par exemple:

```
EventCodes
30
920
980
509
956
```

Si vous ne définissez pas cette section, la supervision d'événements ne se fera pas.

### **Diviser la charge de la supervision entre différents agents logiciels**

Par le biais du fichier de configuration du plugin de l'agent, il est possible de diviser la charge de supervision de l'infrastructure de virtualisation RHEV.

Pour ça les entités à superviser entre les différents agents seront divisés entre les différents agents. Supposons qu'il a l'architecture suivante:

```
DC1
|
|- Cluster 1.1
   |- c1.1mv1
   |- c1.1mv2
   |- c1.1mv3
```

```
| - Cluster 1.2
  | - c1.2mv1
  | - c1.2mv2
  | - c1.2mv3

DC2
|
| - Cluster 2.1
  | - c2.1mv1
  | - c2.1mv2
  | - c2.1mv3

| - Cluster 2.2
  | - c2.2mv1
  | - c2.2mv2
  | - c2.2mv3
```

Une façon de répartir la charge serait d'attribuer un centre de données à chacun des agents logiciels ; pour ce faire, nous utiliserions la fonctionnalité permettant d'écarter les entités à surveiller (*jeton Reject*).

Le premier agent logiciel supervise le Datacenter DC1 et écarte les entités DC2.

```
Reject
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
c2.2mv2
c2.2mv3
```

Le deuxième agent logiciel supervise le Datacenter DC2 et écarte les entités de DC1.

```
Reject
DC1
Cluster 1.1
Cluster 1.2
c1.1mv1
c1.1mv2
c1.1mv3
c1.2mv1
c1.2mv2
c1.2mv3
```

Nous pourrions aussi diviser la charge partant des *clusters*, par exemple. Pour chaque *cluster* des deux Datacenters un agent des quatre premiers sera alloué.

Agent logiciel 1, superviser Cluster 1.1 et écarte les autres entités.

```
Reject
DC1
Cluster 1.2
c1.2mv1
c1.2mv2
c1.2mv3
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
c2.2mv2
c2.2mv3
```

Agent logiciel 2, superviser Cluster 1.2 et il écarte les autres entités.

```
Reject
DC1
Cluster 1.1
c1.1mv1
c1.1mv2
c1.1mv3
DC2
Cluster 2.1
Cluster 2.2
c2.1mv1
c2.1mv2
c2.1mv3
c2.2mv1
c2.2mv2
c2.2mv3
```

Agent logiciel 3, superviser Cluster 2.1 et écarte les autres entités.

```
Reject
DC1
Cluster 1.1
Cluster 1.2
c1.1mv1
c1.1mv2
c1.1mv3
c1.2mv1
c1.2mv2
c1.2mv3
DC2
Cluster 2.2
c2.2mv1
```



```
c2.2mv2  
c2.2mv3
```

Agent logiciel 4, superviser Cluster 2.2 et il écarte les autres entités.

```
Reject  
DC1  
Cluster 1.1  
Cluster 1.2  
c1.1mv1  
c1.1mv2  
c1.1mv3  
c1.2mv1  
c1.2mv2  
c1.2mv3  
DC2  
Cluster 2.1  
c2.1mv1  
c2.1mv2  
c2.1mv3
```

La configuration des entités écartées est complètement flexible et la charge peut se diviser en allouant quelques entités à chaque agent logiciel.

## Nutanix

### Fonctionnement du plugin

Le plugin Nutanix® est un programme écrit dans Perl, qui se connectera à l'API REST de Nutanix PRISM®, en récupérant les métriques nécessaires pour surveiller les éléments suivants:

- Clusters Nutanix®.
- Dispositifs de stockage
- Conteneurs.
- Machines Virtuelles.
- *Hosts*.
- Etats des processus de réplication.

### Pré-requis du plugin

Pour pouvoir récupérer l'information de l'API REST, il nous faudra:

- L'adresse IP/ FQDN du portail.
- Un "utilisateur" avec des autorisations de lecture sur l'API.
- Le *mot de passe* de l'utilisateur.

En ce qui concerne la communication des résultats de la surveillance à votre Pandora FMS, il faudra :

- Le *mode de transfert de l'information*, soit local ou via Tentacle.
- En local, l'adresse du répertoire où les fichiers XML avec les résultats doivent être remis, tels que les autorisations d'écriture dans ledit répertoire.
- Via Tentacle, il faudra pouvoir connecter contre l'adresse IP ou FQDN du serveur Pandora FMS, le port utilisé par l'installation Tentacle, la localisation du client de Tentacle, comme n'importe qu'elle option extraordinaire qu'il ait défini.

## Installation du plugin

Téléchargez les archives requises par le plugin de la [bibliothèque de modules](#). Transférez les archives à l'équipement à distance d'où vous souhaitez réaliser la surveillance de votre infrastructure Nutanix et extrayez les archives du plugin :

```
tar xvzf pandora_nutanix.tar.gz
```

## Configuration du plugin Nutanix

Les champs suivants sont déclarés:

### *Nutanix API configuration*

- `nx_fqdn`: adresse du serveur principal Prism.
- `nx_port`: port dans lequel l'API REST est publié (par défaut 9440).
- `nx_user`: utilisateur avec des avantages de lecture sur l'API REST.
- `nx_pass`: mot de passe de l'utilisateur.
- `use_https`: Utilisation https (1) ou non (0)
- `nx_rest_version`: Version de l'API Rest (par défaut 'v1')

### *Nutanix agent configuration*

- `agent_interval`: Intervalle des agents générés par le plugin (par défaut 300).
- `agent_group`: groupe auquel appartiendront les agents générés (si 'autocreate\_group' est commenté dans la configuration de votre PandoraServer), par défaut Nutanix.
- `module_interval`: intervalle des modules des agents générés (facteur de multiplication, par défaut 1).
- `module_tags`: Étiquettes associées aux nouveaux modules des agents générés.
- `module_group`: groupe auquel appartiendront les nouveaux modules.

### *Configuration de la communication vers le serveur de Pandora FMS*

- `mode`: Mode de transfert de données, "local" ou "tentacle".
- `tentacle_ip`: Adresse IP du serveur Pandora FMS, qui ne s'applique qu'en mode Tentacle.
- `tentacle_port`: Port sur lequel le service Tentacle écoute.
- `tentacle_opts`: Toute option extra que vous ayez configurée sur votre service Tentacle.
- `tentacle_client`: Chemin d'accès complet à votre client Tentacle.

- temp: Répertoire de travail temporaire.
- local\_folder: Chemin d'accès de transfert des données pour le mode "local".

### Filtres

- cluster\_monitoring: Activer(1) ou non (0) la surveillance de clusters.
- storage\_monitoring: Activer (1) ou non (0) la surveillance de dispositifs de stockage.
- container\_monitoring: Activer (1) ou non (0) la surveillance des conteneurs de stockage.
- vm\_monitoring: Activer (1) ou non (0) la surveillance de machines virtuelles.
- host\_monitoring: Activer (1) ou non (0) la surveillance de serveurs de machines virtuelles (nœuds Nutanix).
- pd\_monitoring: Activer (1) ou non (0) la surveillance de domaines de protection.

### Personnalisations

- cluster\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type cluster.
- storage\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type dispositif de stockage.  
;host\_agent\_header
- host\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type serveur de machines virtuelles (nœuds Nutanix).
- container\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type conteneurs de stockage.
- vm\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type machine virtuelle.
- pd\_agent\_header: En-tête pour le nom de l'agent de dispositifs de type domaine de protection.

### Règles de génération des modules

- vm\_stat: Règle pour l'ajout de modules pour la surveillance de machines virtuelles, par défaut `hypervisor_cpu_usage_ppm|hypervisor_memory_usage_ppm|. *avg.*`. Ceci indique les modules extraordinaires qui se généreront quand le nom de la métrique coïncidera avec les expressions régulières indiquées dans ce champs. Ajoutez la valeur `.*` pour surveiller toutes les métriques disponibles.
- host\_stat: Règle pour l'ajout de modules pour la surveillances de machines virtuelles (noeuds Nutanix), par défaut `hypervisor_cpu_usage_ppm|hypervisor_memory_usage_ppm|. *avg.*`. Ceci indique les modules extraordinaires qui se généreront quand le nom de la métrique coïncidera avec les expressions régulières indiquées dans ce champs. Ajoutez la valeur `.*` pour surveiller toutes les métriques disponibles.
- pd\_stat: Règle pour l'ajout de modules pour la surveillance de domaines de protection, par défaut `replication_transmitted_bandwidth_kBps|replication_total_transmitted_bytes`. Ceci indique les modules extraordinaires qui se généreront quand le nom de la métrique coïncidera avec les expressions régulières indiquées dans ce champs. Ajoutez la valeur `.*` pour surveiller toutes les métriques disponibles.

### Renommé des entités

- `RENAME aaa TO bbb`: Règle pour renommer des entités. Vous pouvez définir autant de directives que d'élément à renommer.

### Exclusion des entités

- `REJECT aaa`: Règle pour l'exclusion de surveillance des entités. Vous pouvez définir autant de directives que d'éléments à exclure.

## Exécution du plugin Nutanix

Il est recommandé d'exécuter le plugin à distance, depuis un équipement avec un accès aussi bien à Pandora FMS server qu'à votre infrastructure Nutanix® à surveiller.

Exécution manuelle:

```
./pandora_nutanix-linux-x64 pandora_nutanix.conf
```

Vous pouvez automatiser l'exécution du plugin dans le cron du système en ajoutant la ligne suivante à /etc/crontab.

```
/5 * * * * root /path/to/plugin/pandora_nutanix-linux-x64  
/path/to/plugin/pandora_nutanix.conf
```

## XenServer

### Fonctionnement du plugin XenServer

Le plugin Pandora FMS pour la surveillance des environnements Xen est écrit sur Python. Il utilise XenAPI pour récupérer toute l'information nécessaire. Il permet de surveiller les types d'éléments suivants:

- Systèmes virtualisés dans Xen.
- Ressources de stockage.
- Propre serveur Xen 6.5 et 7.2 ((host)).

### Pré-requis du plugin

Il est indispensable que le système qui exécute le plugin dispose des pré-requis suivants:

- Python installé
- Bibliothèques Python installées: XenAPI et xmltodict.
- Accès à l'API de votre XenServer (web, il active le trafic depuis l'équipement qui exécute le plugin au port 443 ou 80 du XenServer).
- Il est recommandé que les machines virtuelles aient Xen Server Tools d'installée puisque, dans le cas contraire, l'information disponible est assez limitée.

### Installation du plugin

Téléchargez copie du plugin de Pandora FMS pour XenServer depuis la [bibliothèque de modules](#). Vous pouvez le déployer dans l'équipement que vous préférez (MS Windows® ou GNU/Linux®), en extrayant le contenu de l'archive dans un répertoire stable d'où vous pouvez l'exécuter, et qui

utilise déjà correctement l'agent Pandora FMS ou le cron système.

## Configuration du plugin

Configuration disponible pour le plugin Pandora FMS pour Xen:

### *Bloc de configuration [CONF]*

- `xen_server_ip`: Adresse IP/FQDN du serveur Xen.
- `user`: Utilisateur avec autorisations de consultation sur l'API de Xen.
- `password`: Mot de passe de l'utilisateur.
- `temporal`: Adresse de travail temporaire.

### *Bloc de configuration [PANDORA]*

- `tentacle_client`: Emplacement binaire du client de Tentacle
- `tentacle_ip`: Adresse IP sur lequel le service Tentacle est écouté.
- `tentacle_port`: Port sur lequel le service Tentacle est écouté.
- `logfile`: Chemin d'accès complet au fichier de log.
- `interval`: Intervalle des agents générés.
- `group`: Groupe assigné aux agents générés.

### *Bloc de configuration [TUNNING]*

- `time_adjustment`: Paramètre qui permet l'ajustement des différences de temps possibles entre l'équipement qui exécute le plugin et le serveur Xen. (Par défaut = 10, mesuré en secondes).
- `scan_vm_ip`: Paramètre qui permet de définir si le plugin essaiera d'obtenir les IP adresses des VM du serveur Xen. Seules les IP adresses de ces VMs avec celles XenTools installées peuvent être utilisées. On peut les activer (`scan_vm_ip = true`) ou les désactiver (`scan_vm_ip = false`). Si on ne le précise pas, ce sera considéré comme activé.

### *Bloc de configuration [RENAME]*

- `xen_element_name = pandora_agent_name`: Dans ce bloc, il est possible de définir autant d'entrées que souhaitées avec ce format. Cela permet de changer les noms des éléments du Xen Server en d'autres distincts, afin d'être utilisés commues des noms d'agents dans Pandora FMS. Les VMS, les SRs et le XenServer lui même, peuvent être renommés.

## Exécution du plugin

Vous pouvez programmer l'exécution du plugin depuis n'importe quel agent de Pandora FMS, en ajoutant à la configuration de ce dernier:

```
module_plugin python "<path>\xen-plugin.py" "<path>\xen-plugin.conf"
```

Pour le programmer grâce au cron du système, vous pouvez ajouter la ligne suivante à `/etc/crontab`:

```
/5 * * * * root python "<path>\xen-plugin.py" "<path>\xen-plugin.conf">
/dev/null 2>&1
```

Si vous exécutez le plugin manuellement, la sortie doit être semblable à la suivante:

```
python "<path>\xen-plugin.py" "<path>\xen-plugin.conf"
<module>
<name><![CDATA[XenServer Plugin]]></name>
<type><![CDATA[async_string]]></type>
<description><![CDATA[Result of XenServer Plugin execution]]></description>
<data><![CDATA[OK]]></data>
</module>
```

## OpenNebula

### Fonctionnement du plugin OpenNebula

Le plugin Pandora FMS, pour la surveillance d'environnements OpenNebula, est écrit sur Perl. Il s'exécute localement dans le serveur OpenNebula et récupérera toute l'information nécessaire en utilisant les propres commandes de gestion de OpenNebula. Il permet la surveillance des types d'éléments suivants :

- Clusters
- Hosts
- Machines virtuelles
- Ressources de stockage

### Pré-requis du plugin

Il est indispensable que le système qui exécute le plugin dispose des pré-requis suivants :

- Perl disponible sur l'équipement
- Utilisateur avec des droits pour exécuter les commandes suivantes :
  - onehost
  - onecluster
  - onedatastore

Le bon fonctionnement du plugin a fait ses preuves sur des systèmes OpenNebula 5.X.X

### Installation du plugin

Téléchargez copie du plugin de Pandora FMS pour OpenNebula de la [bibliothèque des modules](#). Vous devez extraire le contenu de l'archive dans un répertoire stable, d'où vous pourrez

l'exécuter, qui utilise déjà correctement l'agent de Pandora FMS ou le cron du système.

```
unzip pandora_OpenNebula.zip
```

## Configuration du plugin OpenNebula

### Configuration de la communication vers le serveur Pandora FMS

- mode: Mode de transfert de données, "local" ou "tentacle".
- tentacle\_ip: Adresse IP du serveur Pandora FMS, qui ne s'applique qu'en mode tentacle.
- tentacle\_port: Port dans lequel le service Tentacle est écouté.
- tentacle\_opts: Toute option supplémentaire qui soit configurée dans votre service Tentacle.
- tentacle\_client: Chemin d'accès complet au client Tentacle.
- temp: Adresse de travail temporaire.
- local\_folder: Chemin d'accès de l'emplacement des données pour le mode de transfert "local".

### Configuration de l'agent

- agent\_interval: Intervalle de l'agent, par défaut 300 secondes.
- agent\_group: Groupe de l'agent, par défaut OpenNebula.

### Personnalisation des modules

- MODULE\_GROUP: Groupe des modules, par défaut OpenNebula.
- MODULE\_INTERVAL: Intervalle des modules (multiplicateur), par défaut 1.
- MODULE\_TAGS: Étiquettes pour les modules.

### Personnalisation des noms

- cluster\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type cluster.
- host\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type serveur de machines virtuelles.
- storage\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type dispositif de stockage.
- vm\_agent\_header: En-tête pour le nom de l'agent des dispositifs de type machine virtuelle.

### Filtres

- cluster\_monitoring: Activer (1) ou non (0) la surveillance clusters.
- host\_monitoring: Activer (1) ou non (0) la surveillance de serveurs de machines virtuelles.
- storage\_monitoring: Activer (1) ou non (0) la surveillance de dispositifs de stockage.
- vm\_monitoring: Activer (1) ou non (0) la surveillance de machines virtuelles.

### Renommer des entités

RENAME aaa TO bbb: Règle pour renommer des entités : vous pouvez définir autant de directives

que d'éléments qui nécessitent d'être renommés.

## Exclusion des entités

REJECT aaa: Règle pour l'exclusion de surveillance des entités : vous pouvez définir autant de directives que d'éléments à exclure.

## Exécution du plugin

Pour le programmer grâce au cron du système, vous pouvez ajouter la ligne suivante à `/etc/crontab`:

```
/5 * * * * root "<path>/pandora_opennebula" "<path>/pandora_opennebula.conf">
/dev/null 2>&1
```

Si vous l'exécutez manuellement, la sortie doit ressembler à la suivante :

```
[root@valhalla ~]# ./pandora_opennebula pandora_opennebula.conf
[root@valhalla ~]# echo $?
0
```

## IBM HMC

Ce **plugin** permet de surveiller des équipements de virtualisation IBM AIX grâce à la console d'administration de hardware HMC. Ce plugin collectera des informations de toutes les partitions logiques créées dans un environnement AIX administré par un système HMC, en créant un agent pour chaque serveur administré, chaque partition logique et chaque serveur IO virtuel.

Pour récupérer l'information via SSH, le plugin peut utiliser trois modes de travail :

1. Basé sur expect en utilisant le *script* `ssh_launcher.sh`
2. Basé sur la bibliothèque `Net::SSH::Perl`
3. Basé sur la bibliothèque `Net::SSH::Expect`

Pour compléter l'information capturée, des requêtes sur l'API REST seront réalisées par défaut sur:

```
https://fqdn:12443/rest/api/{root_element}
```

## Pré-requis

Les paramètres nécessaires pour la surveillance, que doit fournir le domaine qui nécessite les



services de surveillance, sont :

- Nom d'utilisateur pour s'authentifier dans le système HMC (lecture seule)
- L'utilisateur doit avoir l'autorisation pour pouvoir se connecter à l'API REST et pour se connecter dans la shell du HMC et exécuter les commandes suivantes (minimum requis) :
  - lssyscfg
  - lshwres
- Mot de passe dudit utilisateur
- Emplacement (FQDN/IP) du HMC (p.e. myhmc.mydomain)
- URL de base de l'API REST du HMC p.e. :

```
https://myhmc.mydomain:12443
```

## Modules générés par le plugin

Les paramètres qui surveille le plugin sont (regroupés par type d'élément) :

- Current logical partitions
- Max logical partitions
- Max memory available
- Max memory installed
- Proc pool DefaultPool current proc units
- Proc pool DefaultPool max proc units
- Proc pool DevelopmentPool current proc units
- Proc pool DevelopmentPool max proc units
- Proc pool ProductionPool current proc units
- Proc pool ProductionPool max proc units
- Proc pool TestPool current proc units
- Proc pool TestPool max proc units
- Proc pool VIOPool current proc units
- Proc pool VIOPool max proc units
- Processor pools configured
- Processor units available
- Processor units installed
- State
- UUID
- Virtual proc units max

LPAR :

- Auto start : Configuration de partitions logiques auto démarrage.
- LPAR type : Type de partition logique.
- LPAR UUID : Utilisé pour vérifier les HMC API.
- Max memory : Memoire maximale.
- Max memory : Memoire disponible.
- Processor units available : Unités de traitement disponibles.
- Processor units current : Unités de traitement installées
- RMC IP address : Adresse IP RMC.
- RMC state : État du RMC dans LPAR
- State : État de la partition logique.

- Virtual proc units : Unités de traitement virtuel alloués à ce LPAR.

#### Virtual IO :

- Auto start : Configuration de partitions logiques auto-démarrage.
- LPAR type : Type de partition logique.
- LPAR UUID : Utilisé pour vérifier les HMC API.
- Max memory : Mémoire maximale.
- Max memory current : Mémoire disponible.
- Processor units available: Unités de traitement disponibles
- Processor units current : Unités de traitement instaladas
- RMC IP address : Adresse IP RMC.
- RMC state RMC : État du RMC en LPAR.
- State : État de la partition logique.
- Virtual proc units : Unités de traitement virtuel alloués à ce LPAR.

## Configuration du plugin IBM HMC

### Configuration de communication IBM HMC vers le serveur de Pandora FMS

- mode: Mode de transfert de données, "local" ou "tentacle".
- tentacle\_ip: Adresse IP du serveur Pandora FMS, qui ne s'applique qu'en mode tentacle.
- tentacle\_port: Port sur lequel le service Tentacle est écouté.
- tentacle\_opts: Toute option supplémentaire qui ait été configurée sur le service Tentacle.
- tentacle\_client: Chemin d'accès complet au client Tentacle.
- temp: Répertoire de travail temporaire.
- local\_folder: Chemin d'accès de dépôt des données en mode transfert "local".

### Configuration de l'accès à HMC

- hmc\_host: IP ou FQDN de l'HMC.
- hmc\_user: Utilisateur avec autorisation de lecture.
- hmc\_pass: Mot de passe.
- as\_agent\_plugin: La sortie du plugin sera renvoyée en format XML pour des exécutions programmées avec l'agent de Pandora FMS (as\_agent\_plugin = 1). Ou sortie standard (as\_agent\_plugin = 0) pour des exécutions programmées avec le cron système ou réalisées comme plugin de serveur.

### Configuration de l'agent HMC

- agent\_name: Optionnel, indiquer un nom pour l'agent père, par défaut "hostname".
- agent\_interval: Intervalle de l'agent, par défaut 300 secondes.
- agent\_group: Groupe de l'agent, par défaut IBM.

### Personnalisation des modules IBM HMC

- module\_group: Groupe des modules, par défaut IBM.
- module\_interval: Intervalle des modules (multiplicateur), par défaut 1.
- module\_tags: Étiquettes pour les modules.

## Renommer des entités

Pour renommer des entités, un renommage pour chaque bloc est utilisé :

```
rename
MyLPAR_NAME TO my new name
MyLPAR_NAME2 TO my second new name
rename_end
```

## Exécution du plugin IBM HMC

Le plugin de Pandora FMS pour la surveillance des systèmes IBM AIX grâce à HMC se déploie de la façon suivante:

En configurant le paramètre `as_agent_plugin` à 1 (exécution en tant que plugin d'agent) :

```
module_plugin /usr/bin/perl pandora_hmc.pl pandora_hmc.conf
```

En configurant le paramètre `as_agent_plugin` à 0 (exécution en tant que plugin de serveur) :

```
# /etc/crontab
*/5 * * * * root /usr/bin/perl /root/hmc/pandora_hmc.pl
/root/vmware/pandora_hmc.conf
```

## HPVM

### Fonctionnement du plugin HPVM

Ce plugin permet de surveiller des équipements de virtualisation HPVM. Il se lance comme un plugin d'agent, en générant en parallèle, un agent en plus pour chaque équipement virtualisé et hébergé dans le système surveillé.

Pour collecter l'information, des commandes locales sont utilisées.

### Pré-requis du plugin HPVM

1. Déployer un agent de Pandora FMS dans l'équipement que vous souhaitez surveiller
2. Disposer d'un utilisateur avec des autorisations pour exécuter le plugin
3. Cet utilisateur doit disposer des autorisations pour exécuter la commande `hvvmstatus` pour pouvoir interpréter la sortie :
  1. `hvvmstatus`
  2. `hvvmstatus -X`
  3. `hvvmstatus -r -X`

## Installation du plugin HPVM

Téléchargez votre copie du plugin de Pandora FMS pour HPVM HP Virtualization Manager monitoring de la [bibliothèque de modules](#). Vous pouvez programmer l'exécution en utilisant les collectes et l'agent de Pandora FMS, déployer ou bien extraire le contenu de l'archive dans un répertoire stable à partir duquel vous pourrez l'exécuter grâce au cron de votre système.

```
unzip pandora_HPVM.zip
```

## Configuration du plugin HPVM

### Configuration de communication HPVM vers le serveur de Pandora FMS

- mode: Mode de transfert de données, "local" ou "Tentacle".
- tentacle\_ip: Adresse IP du serveur Pandora FMS, qui ne s'applique qu'en mode tentacle.
- tentacle\_port: Port sur lequel le service Tentacle est écouté.
- tentacle\_opts: Toute option supplémentaire qui ait été configurée dans le service Tentacle.
- tentacle\_client: Chemin d'accès complet au client Tentacle.
- temp: Répertoire de travail temporaire.
- local\_folder: Chemin d'accès de dépôt des données en mode "local".

### Configuration de l'agent HPVM

- agent\_name: facultatif, indiquer un nom pour l'agent père, par défaut hostname.
- agent\_interval: intervalle de l'agent, par défaut 300.
- agent\_group: groupe auquel les agents appartiendront, par défaut HPVM.

### Personnalisation des modules HPVM

- module\_group: Groupes des modules.
- module\_interval: intervalle des modules (multiplicateur), par défaut 1.
- module\_tags: Étiquettes pour les modules.

## Exécution du plugin HPVM

En exécutant le plugin depuis l'agent de Pandora FMS, la ligne suivante apparaîtra dans le fichier de configuration de l'agent:

```
module_plugin /usr/bin/perl pandora_hpvm.pl pandora_hpvm.conf
```

Pour un test manuel, configurez le plugin en suivant les étapes décrites. Vous pouvez le lancer comme ci-après:

```
perl pandora_hpvm.pl pandora_hpvm.conf
```



---

[Retour à l'index de documentation du Pandora FMS](#)