



PANDORAFMS
E N T E R P R I S E

Kubernetes
Enterprise plugin

© Ártica Soluciones Tecnológicas 2005-2019

ÍNDICE

Introducción	3
Matriz de compatibilidad	3
Prerrequisitos	3
Asignación de permisos para el API	4
Instalación de metrics-server	5
Configuración	6
Configuraciones de acceso al API	6
Configuraciones de los agentes generados	8
Configuraciones de transferencia de los ficheros XML	8
Configuraciones adicionales	9
Ejecución manual	11
Módulos generados por el plugin	11

INTRODUCCIÓN

Este plugin permite obtener datos de un entorno de Kubernetes, generando agentes para cada uno de sus elementos y monitorizando estadísticas.

La información es obtenida vía web a través del API de Kubernetes, por lo que no es necesario instalar ningún software adicional para su funcionamiento, aunque para la obtención de los datos de uso de CPU y memoria de los contenedores y los nodos será necesario instalar el add-on “metrics-server” en el entorno de Kubernetes.

MATRIZ DE COMPATIBILIDAD

Desarrollado contra:

- Kubernetes v1.14.3.
- Metrics-server 1.8+.

PRERREQUISITOS

Se requiere conexión con el servicio Tentacle asociado a su servidor de Pandora FMS.

Se necesitarán las credenciales de acceso o un token de autenticación al API de Kubernetes de un usuario con permisos suficientes para consultar el API.

Opcionalmente, si se quieren obtener los datos de uso de CPU y memoria de contenedores y nodos se debe instalar el add-on “metrics-server” en el entorno de Kubernetes.

ASIGNACIÓN DE PERMISOS PARA EL API

A continuación se describen los pasos a seguir para crear un usuario con los permisos suficientes para obtener los datos de la monitorización.

1. Creación de un *"Cluster role"* de lectura llamado *"api-read-only"*: Se creará un rol que conceda los permisos *"get"*, *"list"* y *"watch"* de todos los recursos de Kubernetes.

```
cat <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  labels:
    name: api-read-only
rules:
- apiGroups:
  - '*'
  resources:
  - '*'
  verbs:
  - get
  - list
  - watch
- nonResourceURLs:
  - '*'
  verbs:
  - get
  - list
  - watch
EOF
```

2. Creación de un *"Cluster role binding"* llamado *"bind-api-read-only"*: Se vinculará el rol creado anteriormente a una *"service account"* ya existente.

```
kubectl create clusterrolebinding bind-api-read-only \
--clusterrole=api-read-only \
--serviceaccount=namespace:user
```

INSTALACIÓN DE METRICS-SERVER

Para instalar el addon de *metrics-server*, será necesario descargarlo en el entorno de Kubernetes. Se puede obtener de su proyecto de github:

<https://github.com/kubernetes-incubator/metrics-server>

Una vez descargado solo se tiene que desplegar mediante el comando *"kubectl"*:

```
kubectl apply -f metrics-server/deploy/1.8+/
```

Se puede verificar que el despliegue se haya realizado con éxito si al lanzar el siguiente comando el estado de su API figura como *"true"*:

```
$ kubectl get apiservices | grep metrics-server  
v1beta1.metrics.k8s.io      kube-system/metrics-server  True    1m
```

Si el despliegue tiene éxito, el API de *"metrics-server"* debería quedar accesible:

```
$ kubectl get --raw "/apis/metrics.k8s.io/v1beta1/"  
{  
  "kind": "APIResourceList",  
  "apiVersion": "v1",  
  "groupVersion": "metrics.k8s.io/v1beta1",  
  "resources": [  
    {  
      "name": "nodes",  
      "singularName": "",  
      "namespaced": false,  
      "kind": "NodeMetrics",  
      "verbs": ["get", "list"]  
    },  
    {  
      "name": "pods",  
      "singularName": "",  
      "namespaced": true,  
      "kind": "PodMetrics",  
      "verbs": ["get", "list"]  
    }  
  ]  
}
```

Además, pasados algunos segundos, se deberían poder obtener datos de uso de CPU y memoria de contenedores y nodos:

```
$ kubectl top node  
NAME      CPU(cores)  CPU%  MEMORY(bytes)  MEMORY%  
kube      93m         4%    968Mi          56%
```

```
$ kubectl top pod  
NAME      CPU(cores)  MEMORY(bytes)  
pod1      0m          3Mi  
pod2      0m          3Mi  
pod3      0m          2Mi  
pod4      0m          3Mi  
pod5      0m          1Mi
```

CONFIGURACIÓN

El fichero de configuración del plugin "*pandora_kubernetes*" se divide en bloques (todos los parámetros de configuración se indicarán sin comillas aunque tengan espacios en blanco):

Configuraciones de acceso al API

```
## API connection parameters
api_ip=192.168.80.145
api_port=8443
ssl=1

## HTTP bearer authentication parameters
auth_method=bearer
auth_token=auth-token

## HTTP basic authentication parameters
#auth_method=basic
#auth_user=user
#auth_pass=pass
```

api_ip

Dirección IP al servicio del API de Kubernetes.

port

Puerto a la escucha del servicio del API de Kubernetes.

ssl

Indica si la conexión al API es por HTTPS o no (1 → HTTPS, 0 → HTTP).

auth_method

Mecanismo de autenticación a utilizar. Los valores posibles son "*bearer*" y "*basic*".

auth_token

Si *"auth_method=bearer"*, indica el token de autenticación del usuario con acceso al API. Este token se puede obtener siguiendo estos pasos:

1. Obtención del *"secret name"* del usuario:

```
$ kubectl get serviceaccounts user -o yaml
apiVersion: v1
kind: ServiceAccount
metadata:
  creationTimestamp: "2019-06-19T10:11:48Z"
  name: user
  namespace: namespace
  resourceVersion: "327"
  selfLink: /api/v1/namespaces/namespace/serviceaccounts/user
  uid: a60f3652-927a-11e9-b423-ae1361a56794
secrets:
- name: secret_name
```

2. Obtención del *"token"* del *"secret name"*:

```
$ kubectl describe secret secret_name
Name:      secret-name
Namespace: namespace
Labels:    <none>
Annotations: kubernetes.io/service-account.name: default
             kubernetes.io/service-account.uid: a60f3652-927a-11e9-b423-ae1361a56794

Type: kubernetes.io/service-account-token

Data
====
token:      auth-token
ca.crt:     1066 bytes
namespace:  7 bytes
```

auth_user

Si *"auth_method=basic"*, indica el usuario con acceso al API.

auth_pass

Si *"auth_method=basic"*, indica la contraseña del usuario con acceso al API.

Configuraciones de los agentes generados

```
## Agents parameters
interval=300
group=Servers
#prefix=KUBE-
```

interval

Intervalo de monitorización en segundos de los agentes generados. Permitirá definir cuándo pasarán los módulos a estado desconocido (por defecto, 2 veces el intervalo definido sin recibir datos).

group

Grupo al que se asignarán los agentes generados. Este grupo se tendrá en cuenta solo si el parámetro *"autocreate_group"* del servidor de Pandora no se ha definido correctamente.

prefix

Permite incluir una cadena de texto delante de los nombres de los agentes generados con el objetivo de identificar la ejecución de la que surgen (en el caso de tener varias ejecuciones del plugin configuradas).

Configuraciones de transferencia de los ficheros XML

```
## XML local transfer parameters
tmp=/tmp
transfer_mode=local
local_folder=/var/spool/pandora/data_in

## XML remote transfer parameters
#tmp=/tmp
#transfer_mode=tentacle
#tentacle_ip=127.0.0.1
#tentacle_port=41121
```

tmp

Directorio temporal en el que se generarán los ficheros XML de los agentes generados antes de su transferencia al servidor de Pandora FMS.

transfer_mode

Método de transferencia de ficheros que se utilizará. Si no se establece como *"tentacle"* se considerará que el método de transferencia es *"local"* (copiando los ficheros XML del directorio temporal a uno definido).

local_folder

Directorio al que se copiarán los ficheros XML si el método de transferencia no se establece como *"tentacle"*.

tentacle_ip

Dirección IP al que se enviarán los ficheros XML si el método de transferencia se establece como *"tentacle"*.

tentacle_port

Puerto al que conectarse al servidor de Tentacle indicado en el parámetro *"tentacle_ip"*.

tentacle_opts

Opciones adicionales para la transferencia de ficheros al servidor de Tentacle indicado.

Configuraciones adicionales

```
## Disable monitoring
get_healthz=1
get_namespaces=1
get_services=1
get_components=1
get_pods=1
get_nodes=1
get_deployments=1
get_metrics=1

## Additional Kubernetes metrics
# Example:
# Apiserver request latencies
bucket=apiserver_request_latencies_bucket{component="apiserver",group="scheduling.k8s.io",resource="priorityclasses",scope="cluster",subresource="",verb="WATCH",version="v1",le="125000"}
```

Extra parameters
#debug=0

get_healthz

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/healthz"*. Su valor por defecto si no se especifica es "1" (habilitado).

get_namespaces

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/api/v1/namespaces"*. Su valor por defecto si no se especifica es "1" (habilitado).

get_services

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/api/v1/services"*. Su valor por defecto si no se especifica es "1" (habilitado).

get_components

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/api/v1/componentstatuses"*. Su valor por defecto si no se especifica es "1" (habilitado).

get_pods

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/api/v1/pods"* y del API *"/apis/metrics.k8s.io/v1beta1/pods"* (si se ha instalado el add-on *"metrics-server"*). Su valor por defecto si no se especifica es "1" (habilitado).

get_nodes

Si se establece a "0" se dejarán de obtener los datos de monitorización del API *"/api/v1/nodes"* y del API *"/apis/metrics.k8s.io/v1beta1/nodes"* (si se ha instalado el add-on *"metrics-server"*). Su valor por defecto si no se especifica es "1" (habilitado).

get_deployments

Si se establece a "0" se dejarán de obtener los datos de monitorización del API `"/apis/extensions/v1beta1/deployments"`. Su valor por defecto si no se especifica es "1" (habilitado).

discard_agents

Lista de agentes (separados por comas) que se quieran descartar de la monitorización del plugin. Deben ser los nombres de los agentes sin el prefijo indicado.

get_metrics

Si se establece a "0" se dejarán de obtener los datos de monitorización del API `"/metrics"`. Su valor por defecto si no se especifica es "1" (habilitado).

Estas métricas se deben especificar en el fichero de configuración con el formato:

nombre del modulo=metrica

Por ejemplo, si se indica en el fichero de configuración:

Apiserver request latencies

```
bucket=apiserver_request_latencies_bucket{component="apiserver",group="scheduling.k8s.io",resource="priorityclasses",scope="cluster",subresource="",verb="WATCH",version="v1",le="125000"}
```

Se generará un módulo llamado:

Apiserver request latencies bucket

con el valor de la métrica:

```
apiserver_request_latencies_bucket{component="apiserver",group="scheduling.k8s.io",resource="priorityclasses",scope="cluster",subresource="",verb="WATCH",version="v1",le="125000"}
```

debug

Si se establece a "1" se mostrará información detallada de las acciones que se realicen durante la ejecución del plugin. Su valor por defecto si no se especifica es "0" (deshabilitado).

EJECUCIÓN MANUAL

Para ejecutar el plugin configure el archivo de configuración "*pandora_kubernetes.conf*" según las instrucciones precedentes.

Ejecución del plugin:

```
./pandora_kubernetes.64 pandora_kubernetes.conf
```

MÓDULOS GENERADOS POR EL PLUGIN

La ejecución de este plugin generará los siguientes agentes y módulos:

- **Un agente llamado "Kubernetes"**: Tendrá los datos de monitorización que no sean referente a contenedores y nodos. Contendrá los módulos:
 - **API status**: Indica si el API es accesible por el plugin.
 - **Healthz**: Indica si el API es accesible por el plugin.
 - **Healthz ping**: Indica si el API es accesible por el plugin.
 - **Healthz log**: Indica si el API es accesible por el plugin.
 - **Healthz etcd**: Indica si el API es accesible por el plugin.
 - **Healthz poststarthook crd informer synced**: Indica si el API es accesible por el plugin.
 - **Healthz poststarthook generic apiserver start informers**: Indica si el API es accesible por el plugin.
 - **Healthz poststarthook start apiextensions controllers**: Indica si el API es accesible por el plugin.
 - **Healthz poststarthook start apiextensions informers**: Indica si el API es accesible por el plugin.
 - **Namespaces**: Cantidad de "*namespaces*" en el entorno.
 - **Services**: Cantidad de "*services*" en el entorno.
 - **Deployments**: Cantidad de "*deployments*" en el entorno.
 - **Components**: Cantidad de "*components*" en el entorno.
 - **Component <component>**: Generará un módulo por cada "*component*" indicando su estado (healthy).
 - **<metric module name>**: Generará un módulo por cada métrica indicada en el fichero de configuración (con los nombres indicados).
- **Un agente para cada nodo**: Su agente padre será el de "*Kubernetes*". Contendrán los módulos:
 - **Pods**: Cantidad de "*pods*" en el nodo.
 - **Pods (%)**: Porcentaje de ocupación de "*pods*" en el nodo.
 - **CPU (cores)**: Uso de CPU del nodo.
 - **CPU (%)**: Porcentaje de uso de CPU del nodo.
 - **Memory (bytes)**: Uso de memoria del nodo.
 - **Memory (%)**: Porcentaje de uso de memoria del nodo.
 - **<condition>**: Generará un módulo por cada "*condition*" en el nodo indicando su estado.

- **Un agente para cada "pod":** Su agente padre será el del nodo en el que se encuentre. Contendrán los módulos:
 - **Pod status:** Indica el estado del "pod". Estados posibles:
 - 0 → Failed
 - 1 → Running
 - 2 → Succeeded
 - 3 → Pending
 - 4 → Unknown
 - **Containers:** Cantidad de contenedores en el "pod".
 - **Container <container> CPU (cores):** Generará un módulo por cada "container" en el "pod" indicando el uso de CPU en el nodo.
 - **Container <container> CPU (%):** Generará un módulo por cada "container" en el "pod" indicando el porcentaje de uso de CPU en el nodo.
 - **Container <container> memory (bytes):** Generará un módulo por cada "container" en el "pod" indicando el uso de memoria en el nodo.
 - **Container <container> memory (%):** Generará un módulo por cada "container" en el "pod" indicando el porcentaje de uso de CPU en el nodo.
 - **<condition>:** Generará un módulo por cada "condition" en el "pod" indicando su estado.

- **Un agente para cada despliegue:** Su agente padre será el de "Kubernetes". Contendrán los módulos:
 - **Replicas:** Número total de "pods" objetivo del despliegue no terminados (Sus etiquetas coinciden con el selector).
 - **Updated replicas:** Número total de "pods" objetivo del despliegue no terminados que tienen la plantilla de especificaciones deseada.
 - **Ready replicas:** Número total de "pods" objetivo del despliegue preparados.
 - **Available replicas:** Número total de "pods" objetivo del despliegue disponibles (preparados desde el al menos "minReadySeconds").
 - **Unavailable replicas:** Número total de "pods" objetivo del despliegue no disponibles. Es el número total de "pods" que todavía son necesarios para que el despliegue tenga disponible una capacidad del 100%. Pueden ser "pods" que están iniciados pero aún no están disponibles o "pods" que aún no han sido creados.
 - **Available:** Indica si el despliegue está disponible.
 - **Progressing:** Indica si el despliegue está lanzando un nuevo "replica set".