

# Monitoreo de clúster HAProxy con DataDog

Jhojan Stiven Espinosa Montes, Brayan David Maca Macancela y Juan Fernando Puche

Jhojan.espinosa@uao.edu.co, brayan.maca@uao.edu.co y juan.puche@uao.edu.co

*Universidad autónoma de occidente*

## I. INTRODUCCIÓN

Un clúster computacional es un grupo de servidores que trabajan juntos para proporcionar un servicio o aplicación. El monitoreo de un clúster computacional es esencial para garantizar su disponibilidad y rendimiento. La plataforma DataDog es una solución de monitoreo de infraestructura que ofrece una amplia gama de funciones para el monitoreo de clústers computacionales. Estas funciones incluyen la recopilación de métricas y logs de los servidores del clúster, la visualización de datos en tiempo real y la generación de alertas.

## II. PROBLEMÁTICA

El monitoreo con Datadog es necesario para garantizar la disponibilidad, el rendimiento y la escalabilidad de las aplicaciones y servicios web. Datadog ofrece una plataforma de monitoreo integral que recopila métricas y logs de los servidores, aplicaciones y servicios web. Estas métricas y logs se pueden utilizar para supervisar el rendimiento, identificar problemas y tomar medidas correctivas.

Hay varias razones por las que el monitoreo con Datadog es necesario:

**Disponibilidad:** Datadog puede detectar problemas de disponibilidad, como servidores caídos o aplicaciones que no responden. Esto puede ayudar a los equipos de operaciones a tomar medidas correctivas rápidamente para evitar que las aplicaciones y servicios web se caigan.

**Rendimiento:** Datadog puede ayudar a los equipos de operaciones a identificar problemas de rendimiento, como cuellos de botella de red o aplicaciones que utilizan demasiados recursos. Esto puede ayudar a los equipos de operaciones a mejorar el rendimiento de las aplicaciones y servicios web.

**Escalar las aplicaciones y servicios web:** Datadog puede ayudar a los equipos de operaciones a escalar las aplicaciones y servicios web de manera eficiente. Datadog puede recopilar métricas de rendimiento para identificar los cuellos de botella que pueden limitar el rendimiento de las aplicaciones y servicios web.

## III. ALTERNATIVAS DE SOLUCIÓN

Las aplicaciones de monitoreo son herramientas que se utilizan para recopilar, almacenar y analizar datos sobre el rendimiento, la disponibilidad y la salud de los sistemas informáticos y las aplicaciones. Estas aplicaciones se utilizan para ayudar a los equipos de operaciones a identificar y solucionar problemas,

mejorar el rendimiento y garantizar la disponibilidad de los sistemas y aplicaciones.

**Nagios** es una solución de monitoreo de sistemas y redes de código abierto. Es una de las soluciones de monitoreo más populares del mundo y se utiliza en empresas de todos los tamaños.

**Nagios** ofrece una amplia gama de funciones y capacidades, que incluyen:

- Monitoreo de la disponibilidad y el rendimiento de los servidores, redes, aplicaciones y servicios.
- Generación de alertas cuando se detectan problemas.
- Visualización de datos de monitoreo en tiempo real.
- Creación de informes de monitoreo personalizados.

**Zabbix** es una solución de monitoreo de sistemas y redes de código abierto. Es una solución de monitoreo muy completa y ofrece una amplia gama de funciones y capacidades, que incluyen:

- Monitoreo de la disponibilidad y el rendimiento de los servidores, redes, aplicaciones y servicios.
- Generación de alertas.
- Diferentes formas de visualización de datos de monitoreo en tiempo real.
- Creación de informes de monitoreo personalizados.
- Descubrimiento automático de dispositivos y servicios.
- Soporte para múltiples arquitecturas de TI, incluidas nubes públicas, privadas e híbridas.

En este caso se trabajó con **Datadog**, es la plataforma esencial de monitoreo y seguridad para aplicaciones en la nube. Reunimos seguimientos, métricas y registros de un extremo a otro para que sus aplicaciones, infraestructura y servicios de terceros sean completamente observables. Estas capacidades ayudan a las empresas a proteger sus sistemas, evitar tiempos de inactividad y garantizar que los clientes obtengan la mejor experiencia de usuario.

## IV. DISEÑO DE LA SOLUCIÓN

Se realizó el monitoreo de una red de máquinas virtuales con sistema operativo Linux, las cuales cada una cumple un rol dentro de esta red. Hay una máquina virtual que tiene instalado el haproxy para balancear la carga de las peticiones realizadas

por el usuario, mientras que las otras dos máquinas tienen la función de responder al cambio y mostrar la información. Siguiendo este orden de ideas, así se realizó la implementación.

#### A. Vagrant File

```
Vagrant.configure("2") do |config|
  config.vm.define :haproxy do |haproxy|
    haproxy.vm.box = "jhojanespinoza/box"
    haproxy.vm.network :private_network, ip: "192.168.50.5"
    haproxy.vm.hostname = "haproxy"
  end
  config.vm.define :cliente do |cliente|
    cliente.vm.box = "jhojanespinoza/box"
    cliente.vm.network :private_network, ip: "192.168.50.6"
    cliente.vm.hostname = "cliente"
  end
  config.vm.define :maquina do |maquina|
    maquina.vm.box = "jhojanespinoza/box"
    maquina.vm.network :private_network, ip: "192.168.50.7"
    maquina.vm.hostname = "maquina"
  end
end
```

#### B. Maquina Haproxy

Se deberá realizar los siguientes comandos:

Instalar haproxy → Sudo yum install haproxy

Archivo de configuración del haproxy → Sudo vim /etc/haproxy/haproxy.cfg

Agregar las siguientes líneas en la parte de abajo del archivo:

```
listen stats # Define a listen section called "stats"
  bind :9000 # Listen on port 9000
  mode http
  stats enable # Enable stats page
  stats hide-version # Hide HAProxy version
  stats realm Haproxy\ Statistics # Title text for popup window
  stats uri /haproxy_stats # Stats URI
  stats auth Jhojan:34264116jhojan # Authentication credentials
```

Además, realizar la siguiente configuración en ese mismo archivo:

```
#-----
# main frontend which proxys to the backends
#-----
frontend main
  bind *:80 #Puerto que recibe las peticiones http
  acl url_static path_beg -i /static /images /javascript /stylesheets
  acl url_static path_end -i .jpg .gif .png .css .js
  default_backend app
#-----
# static backend for serving up images, stylesheets and such
#-----
backend static
  balance roundrobin
  server static 127.0.0.1:4331 check
#-----
# round robin balancing between the various backends
#-----
backend app
  balance roundrobin
  server cliente 192.168.50.6:80 check #Cambiar la ip por la ip que está en uso
  server maquina 192.168.50.7:80 check ##Cambiar la ip por la ip que está en uso

Posteriormente en la maquina haproxy instalar la integración de haproxy que ofrece datadog, como datadog usa una API key única por usuario, se debe crear una cuenta, siguiendo el paso a paso de datadog, escoger CentOS/Red Hat como agent a instalar, después aparecerá un comando como el siguiente con su respectiva API KEY.

DD_API_KEY=XXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX DD_SITE="us5.datadoghq.com"
DD_APM_INSTRUMENTATION_ENABLED=host bash -c
"$(curl -L https://s3.amazonaws.com/dd-agent/scripts/install_script_agent7.sh)"
```

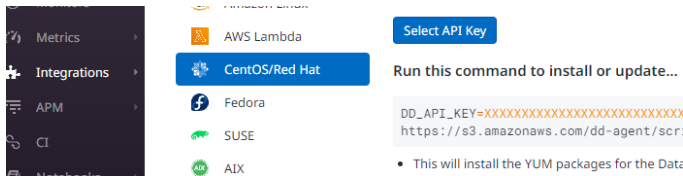


Fig. 1 Integración CentOS/RedHat

Para confirmar que los servicios estén funcionando, realizar los siguientes comandos:

Datadog → sudo systemctl restart datadog-agent

Haproxy → sudo service haproxy restart

### C. Maquinas “cliente” y “maquina”:

Realizar los siguientes comandos:

Instalar el servicio de apache/http → Sudo yum install httpd

Directorio de configuración → cd /var/www/html

Configuración de la página que visualiza el usuario → sudo vim main.html

En **main.html** se realiza la página web al gusto del desarrollador, de preferencia que cada maquina tenga un objeto diferenciador para que sea visible el cambio de página y el balanceo.

## V. PRUEBAS

Para comprobar el correcto funcionamiento del balanceo de carga y el monitoreo en tiempo real de la red de maquinas propuesto, se puede realizar lo siguiente:

- En el apartado de la infraestructura debe estar el nombre de la maquina en la que se instaló el haproxy.
- Para el balanceo, por cada recarga de página debe variar entre “cliente” y “maquina”, mientras se realiza está acción varias veces las graficas en el Datadog van a comenzar a realizar variaciones en el disco, red, etc. Ya que se está haciendo uso de los recursos asignados a esa máquina.
- Si la máquina virtual encargada del haproxy se apaga, DataDog ya no hace más marcaciones en sus gráficos de monitoreo.

## VI. DISCUSIÓN DE LAS PRUEBAS

En el apartado de infraestructura debe aparecer el nombre de la maquina en la que se instaló el agent de haproxy (ver imagen).

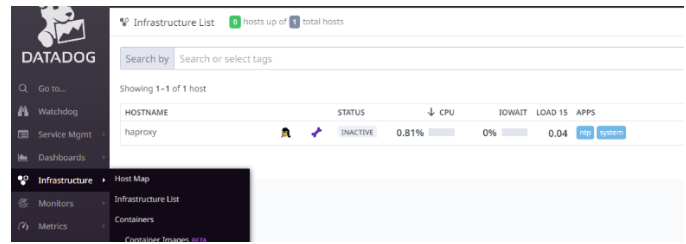


Fig. 2 Infraestructura

En el apartado de dashboard, aparecerán varios por default o también hay la opción de crear uno nuevo.

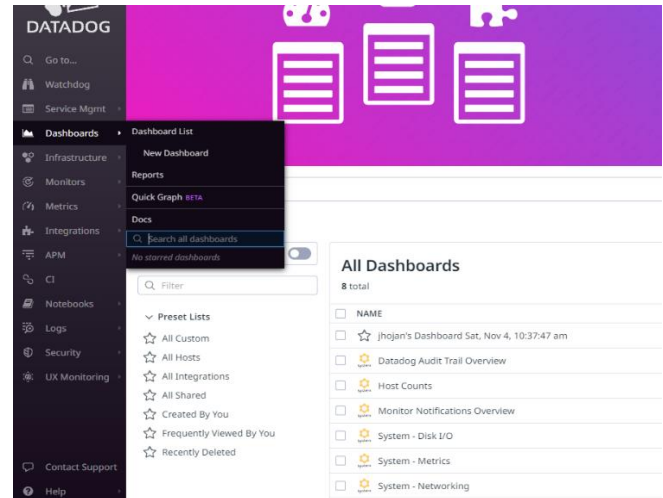


Fig. 3 Lista de dashboard

Las graficas de este dashboard nos enseñan el tráfico de la red.

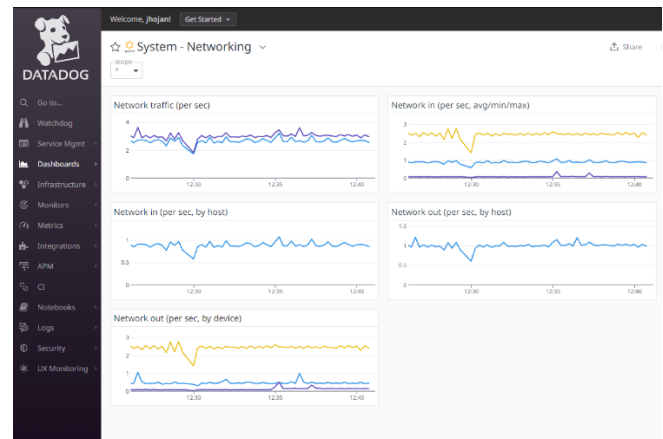


Fig. 4 Dashboard de la red

Ingresar a través del buscador a la ip de la maquina "haproxy" (192.168.50.5)



Fig. 5 IP haproxy en el buscador

De esta manera debe cargar alguna de las dos páginas web que realizaste en pasos anteriores:

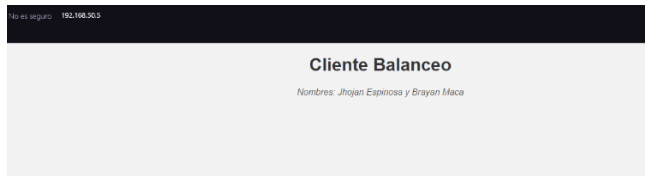


Fig. 6 Comprobación de funcionamiento

## VII. CONCLUSIÓN

Datadog es una plataforma de monitoreo integral que proporciona una vista unificada de la infraestructura, las aplicaciones y los servicios. Recopila y analiza datos de diversas fuentes, incluidos registros, métricas y trazas, para ayudar a las organizaciones a identificar y resolver problemas de rendimiento, mejorar la utilización de los recursos y garantizar la salud y disponibilidad general de sus sistemas. Por otro lado, haproxy y DataDog tienen un despliegue excelente en el sistema operativo Linux, y gracias a ello su usabilidad es cómoda.

El proyecto “Monitoreo de Clúster Haproxy con DataDog. Se requiere realizar monitoreo completo de un clúster Haproxy usando la plataforma DataDog” aporta al entendimiento total sobre el funcionamiento del monitoreo, como, por ejemplo, el consumo de disco, memoria, tráfico de red y como a partir de aquellos datos, ayuda a detectar fallos, garantizar su debida corrección y funcionamiento adecuado.

**GitHub:** <https://github.com/jhojan16/ProyectoFinalST.git>

## REFERENCIAS

[1]“Datadog”. Leadership. Accedido el 26 de octubre de 2023. [En línea]. Disponible: <https://www.datadoghq.com/about/leadership/>

[2]“Nagios Open Source”. Nagios Open Source. Accedido el 1 de noviembre de 2023. [En línea]. Disponible: <https://www.nagios.org/about/>

[3]“Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution”. Zabbix :: The Enterprise-Class Open Source Network Monitoring Solution. Accedido el 29 de octubre de 2023. [En línea]. Disponible: <https://www.zabbix.com>

[4]“Getting Started with Datadog”. Datadog Infrastructure and Application Monitoring. Accedido el 30 de octubre de 2023. [En línea]. Disponible: <https://docs.datadoghq.com/integrations/haproxy/?tab=host>

[5]“Formato IEEE para trabajos escritos (Normas IEEE) « Formato IEEE para trabajos escritos”. Portales de la Comunidad – Portales de la comunidad Universitaria. Accedido el 3 de noviembre de 2023. [En línea]. Disponible: <https://comunidad.udistrital.edu.co/formatoieee/2023/01/23/formato-ieee-para-trabajos-escritos-normas-ieee/>