

INFORME PRÁCTICA 5

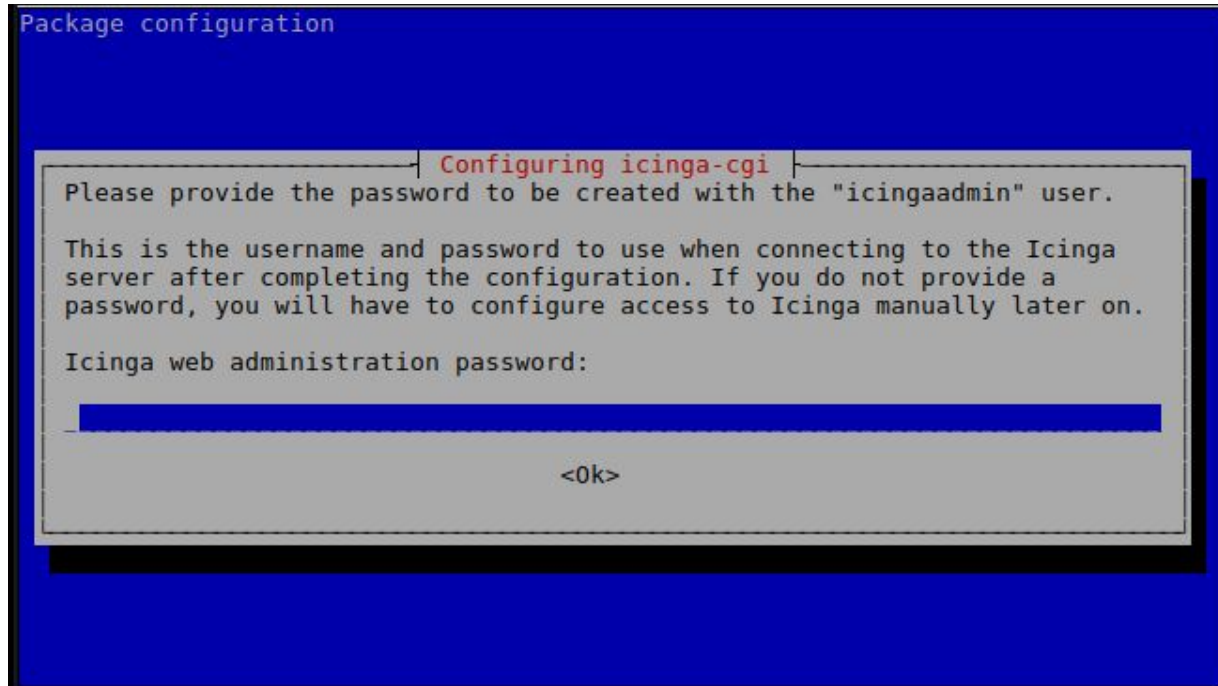
Julio César Velásquez Cárdenas 1397896

David Sánchez González 1401641

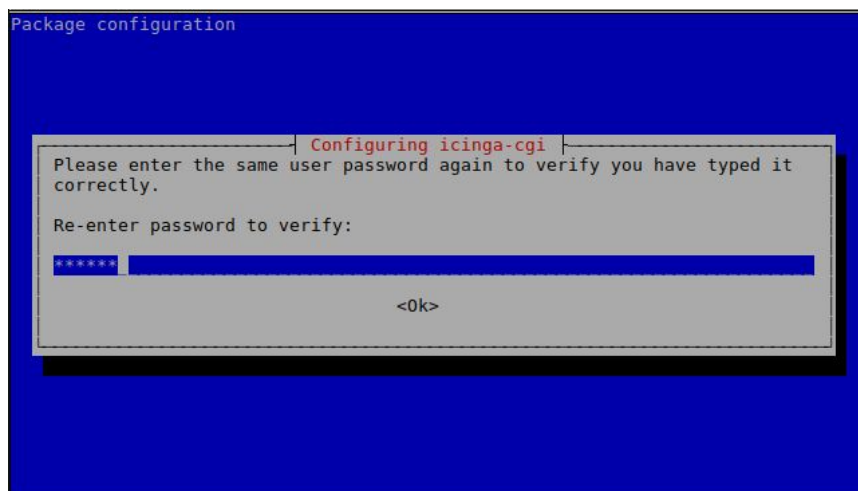
Asignatura: Gestió i Administració de Xarxes

Ejercicio 1

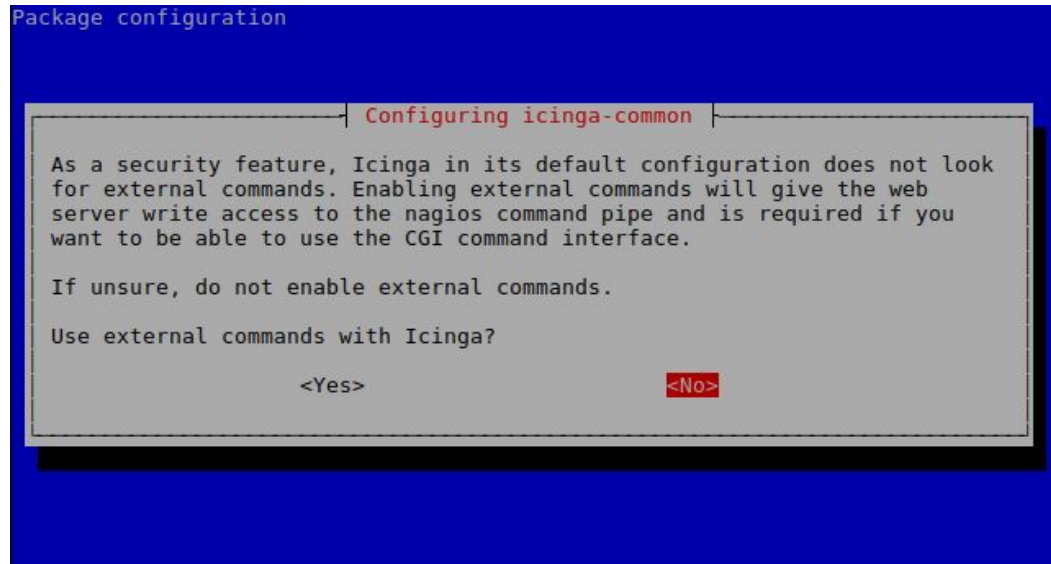
Antes de instalar Icinga, hemos de usar el comando *apt update* para actualizar todos los paquetes del repositorio Debian. Después de esto, usaremos el comando *apt install icinga* para instalar dicho servicio. En un momento dado durante la instalación nos aparecerá el siguiente mensaje que nos indica que seleccionemos una contraseña para el usuario icingaadmin.



En nuestro caso usaremos la contraseña icinga. En el siguiente mensaje, tendremos que volver a escribir la misma contraseña para confirmar.



Cerca del final de la instalación nos aparecerá este mensaje que nos pregunta si queremos activar una de las configuraciones de Icinga. Al no estar del todo seguros, no activaremos estos comandos externos.



Al acabar la instalación, añadiremos los siguientes hosts al archivo `/etc/hosts` de la máquina A.

```
10.10.10.53    icingagw.gax.org
20.20.20.44    icinga.gax.org
```

Accedemos al servicio Icinga conectándonos a un navegador a la dirección icinga.gax.org/icinga/. La página que nos aparecerá es la siguiente.

The screenshot displays the Icinga web interface in a browser window. The address bar shows icinga.gax.org/icinga/. The interface features a top status bar with various monitoring metrics: 1 UP, 0 DOWN, 0 UNREACHABLE, 0 PENDING, 0 TOTAL; 2 OK, 0 WARNING, 1 CRITICAL, 0 UNKNOWN, 3 PENDING, 4 TOTAL. The left sidebar contains a navigation menu with sections: General (Home, Docs, Search), Status (Tactical Overview, Host Detail, Service Detail, Hostgroup Overview, Hostgroup Summary, Servicegroup Overview, Servicegroup Summary, Status Map), Problems (Service Problems, Unhandled Services, Host Problems, Unhandled Hosts, All Unhandled Problems, All Problems, Network Outages), System (Comments, Downtime, Process Info, Performance Info, Scheduling Queue), and Reporting (Trends, Availability, Alert Histogram, Alert History, Alert Summary).

The main content area is titled "Tactical Monitoring Overview" and includes the following sections:

- Network Outages:** 0 Outages
- Hosts:** 0 Down, 0 Unreachable, 1 Up, 0 Pending
- Services:** 1 Critical, 0 Warning, 0 Unknown, 2 Ok, 3 Pending
- Unhandled Problems:** 1 Active
- Host Checks:**

Active	Passive
Enabled 1 Enabled	Enabled No Passive Checks
- Service Checks:**

Active	Passive
Enabled 6 Enabled	Enabled No Passive Checks
- Monitoring Features:**

Flap Detection	Notifications	Event Handlers
Enabled All Services Enabled	Enabled All Services Enabled	Enabled All Services Enabled
Enabled No Services Flapping	Enabled All Hosts Enabled	Enabled All Hosts Enabled
Enabled All Hosts Enabled		

The "Network Health" section on the right shows "Host Health" as green and "Service Health" as red.

Como queremos monitorizar no tan solo A sino también B y C, hemos de añadirlos al archivo *localhost_icinga.cfg* que se encuentra en */etc/icinga/objects*. El archivo ha de quedar de la siguiente manera.

```
# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in icinga-common_services.cfg may also apply.
#

define host{
    use                generic-host           ; Name of host template to use
    host_name          localhost
    alias              localhost
    address            127.0.0.1
}

define host{
    use                generic-host
    host_name          slaveB
    alias              slaveB
    address            20.20.25.57
}

define host{
    use                generic-host
    host_name          slaveC
    alias              slaveC
    address            20.20.20.200
}
```

Haciendo esto correctamente, la página donde se nos muestra los hosts que estamos monitorizando en Icinga debería quedar de la siguiente manera.

The screenshot shows the Icinga Classic UI interface. At the top, there's a navigation bar with the Icinga logo and various status indicators. The main content area is divided into several sections:

- General:** Includes links to Home, Docs, and a search bar.
- Status:** A sidebar menu with options like Tactical Overview, Host Detail, Service Detail, Hostgroup Overview, Servicegroup Overview, and Status Map.
- Problems:** A sidebar menu with options like Service Problems, Unhandled Services, Host Problems, Unhandled Hosts, All Unhandled Problems, All Problems, and Network Outages.
- Current Network Status:** A section showing the overall network status, including the number of hosts and services in different states (UP, DOWN, UNREACHABLE, PENDING, CRITICAL, UNKNOWN).
- Host Status Details For All Hosts:** A table showing the status of individual hosts.

The **Host Status Details For All Hosts** table is as follows:

Host	Status	Last Check	Duration	Attempt	Status Information
localhost	UP	2020-12-12 15:33:19	1d 23h 48m 39s	1/10	PING OK - Packet loss = 0%, RTA = 0.09 ms
slaveB	UP	2020-12-12 15:29:49	0d 0h 9m 38s	1/10	PING OK - Packet loss = 0%, RTA = 1.44 ms
slaveC	UP	2020-12-12 15:31:29	1d 23h 4m 41s	1/10	PING OK - Packet loss = 0%, RTA = 0.93 ms

Para monitorizar la conexión SSH, Apache y hacer un PING a estas máquinas debemos hacer lo siguiente. Primero hemos de crear un nuevo hostgroup en el archivo `hostgroups_icinga.cfg` y añadir las máquinas B y C a los grupos ya creados.

```
# A list of your web servers
define hostgroup {
    hostgroup_name  http-servers
        alias      HTTP servers
        members     localhost,slaveB,slaveC
    }

# A list of your ssh-accessible servers
define hostgroup {
    hostgroup_name  ssh-servers
        alias      SSH servers
        members     localhost,slaveB,slaveC
    }

define hostgroup {
    hostgroup_name  ping-servers
        alias      PING servers
        members     localhost,slaveB,slaveC
    }
```

Después hemos de crear el servicio PING en `services_icinga.cfg`.

```
define service{
    hostgroup_name      ping-servers
    service_description  PING
    check_command        check_ping!100.0,20%!500.0,60%
    use                 generic-service
    notification_interval 0 ; set > 0 if you wantt be renotified
}
```


Reiniciamos Icinga y ahora deberían salir estos servicios en la página principal.

Set Filters

Service Status Details For All Hosts

Page 1 of 1 Results: 50

Host	Service	Status	Last Check	Duration	Attempt	Status Information	
localhost	Current Load	OK	2020-12-13 21:22:47	3d 5h 41m 52s	1/4	OK - load average: 0.07, 0.06, 0.09	<input type="checkbox"/>
	Current Users	OK	2020-12-13 21:25:17	3d 5h 41m 2s	1/4	USERS OK - 1 users currently logged in	<input type="checkbox"/>
	Disk Space	CRITICAL	2020-12-13 21:23:37	3d 5h 40m 12s	4/4	DISK CRITICAL - /run/user/0/gvfs is not accessible: Permission denied	<input type="checkbox"/>
	HTTP	OK	2020-12-13 21:23:08	3d 5h 39m 22s	1/4	HTTP OK: HTTP/1.1 200 OK - 10969 bytes in 0.002 second response time	<input type="checkbox"/>
	PING	OK	2020-12-13 21:23:29	0d 0h 14m 11s	1/4	PING OK - Packet loss = 0%, RTA = 0.07 ms	<input type="checkbox"/>
	SSH	CRITICAL	2020-12-13 21:24:38	0d 2h 28m 46s	4/4	connect to address 127.0.0.1 and port 22: Connection refused	<input type="checkbox"/>
	Total Processes	OK	2020-12-13 21:26:57	3d 5h 37m 42s	1/4	PROCS OK: 129 processes	<input type="checkbox"/>
slaveB	HTTP	OK	2020-12-13 21:27:30	0d 0h 25m 10s	1/4	HTTP OK: HTTP/1.1 200 OK - 10967 bytes in 0.004 second response time	<input type="checkbox"/>
	PING	OK	2020-12-13 21:22:43	0d 0h 4m 57s	1/4	PING OK - Packet loss = 0%, RTA = 0.77 ms	<input type="checkbox"/>
	SSH	OK	2020-12-13 21:24:00	0d 0h 23m 40s	1/4	SSH OK - OpenSSH 7.4p1 Debian-10+deb9u7 (protocol 2.0)	<input type="checkbox"/>
slaveC	HTTP	OK	2020-12-13 21:25:30	0d 0h 22m 10s	1/4	HTTP OK: HTTP/1.1 200 OK - 10975 bytes in 0.003 second response time	<input type="checkbox"/>
	PING	OK	2020-12-13 21:23:52	0d 0h 3m 48s	1/4	PING OK - Packet loss = 0%, RTA = 0.79 ms	<input type="checkbox"/>
	SSH	OK	2020-12-13 21:26:08	0d 0h 21m 32s	1/4	SSH OK - OpenSSH 7.4p1 Debian-10+deb9u7 (protocol 2.0)	<input type="checkbox"/>

Para la monitorización de B usando un agente SNMP primero hemos de instalar Nagios NRPE sobre A usando el comando `apt-get --no-install-recommends install nagios-nrpe-plugin` y sobre B usando el comando `apt-get install nagios-nrpe-server`. Después de la instalación hemos de configurar el archivo `nrpe.cfg` en la máquina B para que pueda aceptar las órdenes del servidor en A.

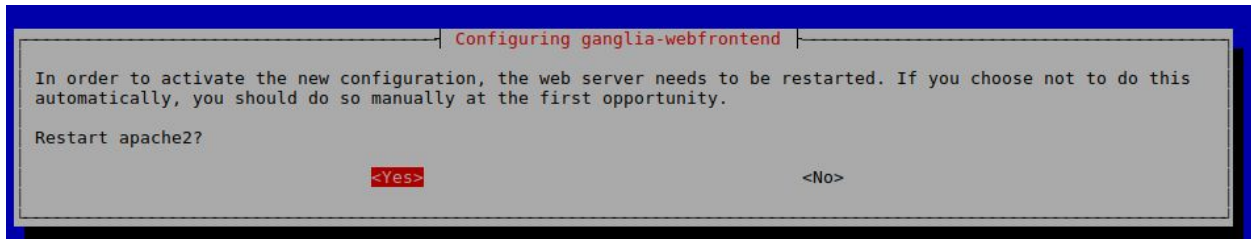
```
# ALLOWED HOST ADDRESSES
# This is an optional comma-delimited list of IP address or hostnames
# that are allowed to talk to the NRPE daemon. Network addresses with a bit mask
# (i.e. 192.168.1.0/24) are also supported. Hostname wildcards are not currently
# supported.
#
# Note: The daemon only does rudimentary checking of the client's IP
# address. I would highly recommend adding entries in your /etc/hosts.allow
# file to allow only the specified host to connect to the port
# you are running this daemon on.
#
# NOTE: This option is ignored if NRPE is running under either inetd or xinetd
allowed_hosts=127.0.0.1,20.20.20.44
```

En este punto se debería probar la conexión de la máquina A y B usando el comando pero como se puede observar, por alguna razón, la conexión es rechazada aun cuando la configuración y el reinicio del servicio han sido hechos.

```
root@master-1-8:~# /usr/lib/nagios/plugins/check_nrpe -H 20.20.25.57  
CHECK_NRPE: Error - Could not connect to 20.20.25.57: Connection reset by peer
```


Ejercicio 2

Instalamos Ganglia en la máquina A usando el comando `apt-get install ganglia-monitor gmetad ganglia-webfrontend`. Durante la instalación nos aparecerán unos mensajes diciéndonos que tenemos que reiniciar el servicio Apache. Les decimos que sí.



Hemos de configurar unos archivos antes de empezar. Primero modificaremos el archivo `gmond.conf` de la siguiente manera.

```
/* If a cluster attribute is specified, then all gmond hosts are wrapped inside
 * of a <CLUSTER> tag. If you do not specify a cluster tag, then all <HOSTS> will
 * NOT be wrapped inside of a <CLUSTER> tag. */
cluster {
    name = "my_cluster"
    owner = "unspecified"
    latlong = "unspecified"
    url = "unspecified"
```

```
/* Feel free to specify as many udp_send_channels as you like. Gmond
   used to only support having a single channel */
udp_send_channel {
    #mcast_join = 239.2.11.71
    host = 10.10.10.53
    port = 8649
    ttl = 1
}

/* You can specify as many udp_rcv_channels as you like as well. */
udp_rcv_channel {
    #mcast_join = 239.2.11.71
    port = 8649
    #bind = 239.2.11.71
}
```

Y el archivo `gmetad.conf` de la siguiente manera.

```
data_source "my_cluster" 10.10.10.53
```

```
# If you want any host which connects to the gmetad XML to receive  
# data, then set this value to "on"  
# default: off  
all_trusted on
```

Ejercicio 3

a) Ganglia

Antes de comenzar, crearemos una nueva MV en la que instalaremos el Docker Engine y el Docker Compose.

A continuación clonamos el repositorio de ganglia a nuestra máquina, lo que nos creará una carpeta con todos los ficheros del repositorio, llamada *docker-ganglia*.

```
root@extral-1-8:~# ls -l -h docker-ganglia/
total 20K
-rw-r--r-- 1 root root 209 Dec 10 16:08 docker-compose.yml
-rw-r--r-- 1 root root 492 Dec 10 16:08 Dockerfile
drwxr-xr-x 3 root root 4.0K Dec 10 16:08 files
-rw-r--r-- 1 root root 1.1K Dec 10 16:08 LICENSE
-rw-r--r-- 1 root root 2.4K Dec 10 16:08 README.md
```

Como se puede ver en ella se encuentra el Dockerfile que necesitaremos para hacer un *build* y crear la imagen. Pero lo debemos modificar, ya que está pensado para una versión más antigua de Ubuntu, lo cambiaremos para que utilice la última versión del mismo.

```
FROM ubuntu
MAINTAINER Kurt Huwig

RUN apt-get update \
    && DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        ganglia-monitor \
        ganglia-webfrontend \
        gmetad \
        supervisor \
    && rm -rf /var/lib/apt/lists/* /var/cache/apt/*

RUN ln -s /etc/ganglia-webfrontend/apache.conf /etc/apache2/conf-available/ganglia.conf \
    && a2enconf ganglia

COPY files/ /

VOLUME ["/var/lib/ganglia"]

CMD ["/entrypoint.sh"]

EXPOSE 80 8649 8649/udp
```

Una vez hecho esto, ya podemos ir a la carpeta y ejecutar *docker build --tag=ganglia .*, con lo que creará la imagen de *Ganglia*.

```

root@extral-1-8:~/docker-ganglia# docker build --tag=ganglia .
Sending build context to Docker daemon 108.5kB
Step 1/8 : FROM ubuntu
--> f643c72bc252
Step 2/8 : MAINTAINER Kurt Huwig
--> Using cache
--> 9e1bcabcfe0f
Step 3/8 : RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends ganglia-
monitor ganglia-webfrontend gmetad supervisor && rm -rf /var/lib/apt/lists/* /var/cache/apt/*
--> Using cache
--> 6328e32500ae
Step 4/8 : RUN ln -s /etc/ganglia-webfrontend/apache.conf /etc/apache2/conf-available/ganglia.conf && a2enconf ganglia
--> Using cache
--> 78e98a2d8487
Step 5/8 : COPY files/ /
--> dc7452371846
Step 6/8 : VOLUME ["/var/lib/ganglia"]
--> Running in bdb606564ac9
Removing intermediate container bdb606564ac9
--> 0b816b1fa485
Step 7/8 : CMD ["/entrypoint.sh"]
--> Running in 19b803407ae3
Removing intermediate container 19b803407ae3
--> 65cd884580c2
Step 8/8 : EXPOSE 80 8649 8649/udp
--> Running in 4232813a7e56
Removing intermediate container 4232813a7e56
--> 66329c97964e
Successfully built 66329c97964e
Successfully tagged ganglia:latest
root@extral-1-8:~/docker-ganglia# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ganglia	latest	66329c97964e	20 seconds ago	263MB
<none>	<none>	8cc4bd8a40bb	21 hours ago	263MB
ubuntu	latest	f643c72bc252	2 weeks ago	72.9MB

Para tener un almacenamiento persistente, debemos crear un par de carpetas que serán utilizadas por el contenedor para almacenar los datos que necesite. Estas carpetas son *etc* y *data*, que serán creadas dentro del directorio *docker-ganglia*.

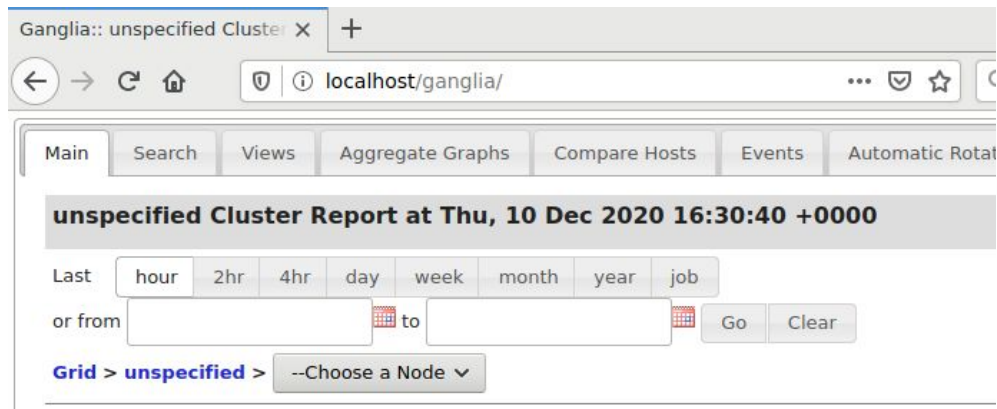
A continuación ejecutaremos el contenedor tal como se indica en la página del repositorio.

```

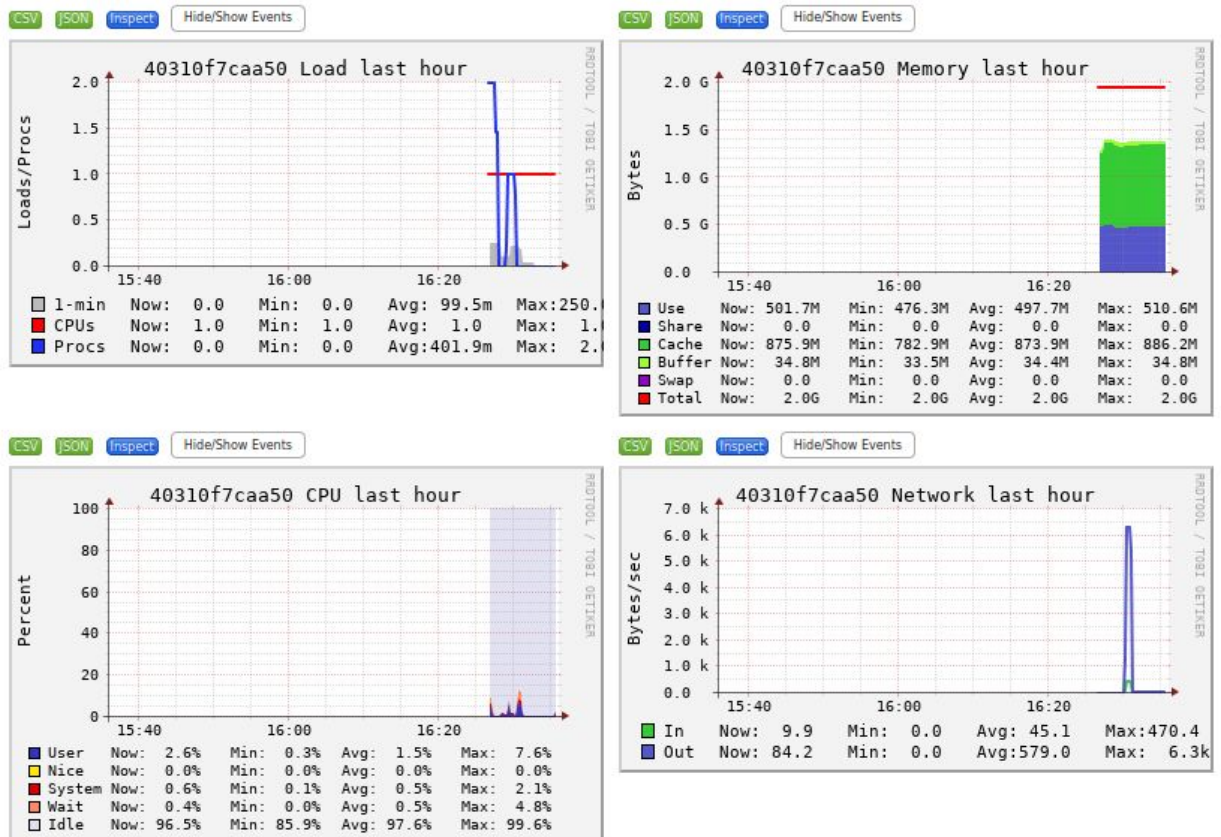
root@extral-1-8:~/docker-ganglia# docker run \
> -d \
> -v etc:/etc/ganglia \
> -v data:/var/lib/ganglia \
> -p 127.0.0.1:80:80 \
> -p 8649:8649 \
> -p 8649:8649/udp \
> ganglia
40310f7caa5077deb0262a5157e3f920c35b995c8ef94770ed77a8a995c39068

```

Si accedemos a *localhost/ganglia* en el navegador, nos encontraremos con la página de inicio de *Ganglia*, donde podremos ver diferentes opciones.



Si seleccionamos un nodo, en nuestro caso el nodo `40310f7caa50`, podremos ver diferentes estadísticas de uso de los principales componentes de una máquina, como son el uso de memoria, CPU y red.



b) cAdvisor

Ejecutamos el comando que iniciaría el contenedor de cAdvisor, especificando antes una variable de entorno que indique la última versión, en el momento de escribir esto, es la versión 0.38.0.

```
root@extral-1-8:~# VERSION=v0.38.0 # use the latest release version from https://github.com/google/cadvisor/releases
root@extral-1-8:~# sudo docker run \
> --volume=:/rootfs:ro \
> --volume=/var/run:/var/run:ro \
> --volume=/sys:/sys:ro \
> --volume=/var/lib/docker:/var/lib/docker:ro \
> --volume=/dev/disk:/dev/disk:ro \
> --publish=8080:8080 \
> --detach=true \
> --name=cadvisor \
> --privileged \
> --device=/dev/kmsg \
> gcr.io/cadvisor/cadvisor:$VERSION
```

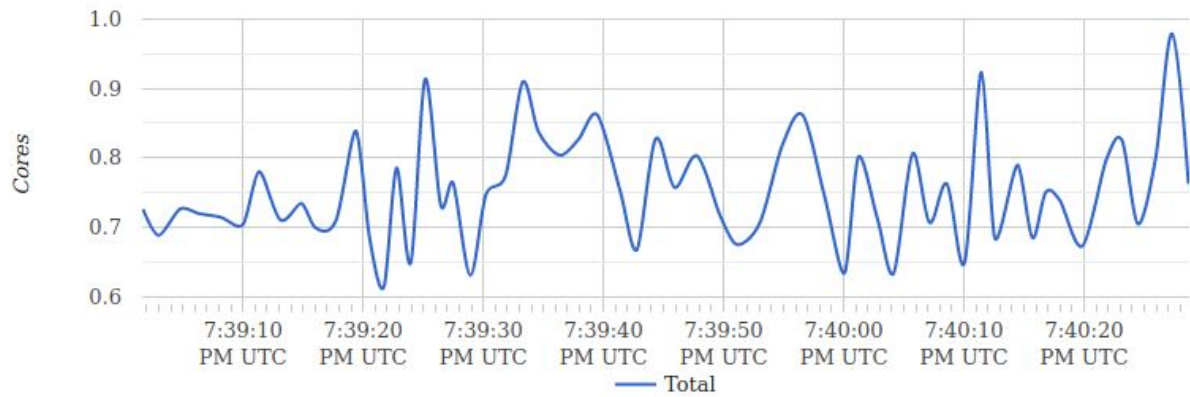
Una vez se ejecute, comenzará a descargar e instalar lo necesario para ejecutar el contenedor. Que será accesible a través del navegador en *localhost:8080*.



Como se puede ver a continuación, nos encontramos con elementos iguales a Ganglia, como información sobre la CPU y la memoria.

Memory
Reservation unlimited
Limit 1.96 GB
Swap Limit 590.00 MB

Overview



c) Ventajas y desventajas de cada uno

Tenemos dos diferencias notables entre ambas herramientas, en las que sale favorecida cAdvisor sobre Ganglia, y es el hecho de que cAdvisor permite la monitorización en tiempo real, mientras que Ganglia también lo hace en tiempo real, salvo que hay que recargar la página a cada momento, por lo que no se puede considerar a tiempo real.

Otra de las diferencias es la posibilidad con cAdvisor de monitorizar otros contenedores de Docker.



Lo que es de mucha utilidad si quieres monitorizar un fragmento de una máquina en lugar de su totalidad.

Ejercicio 4

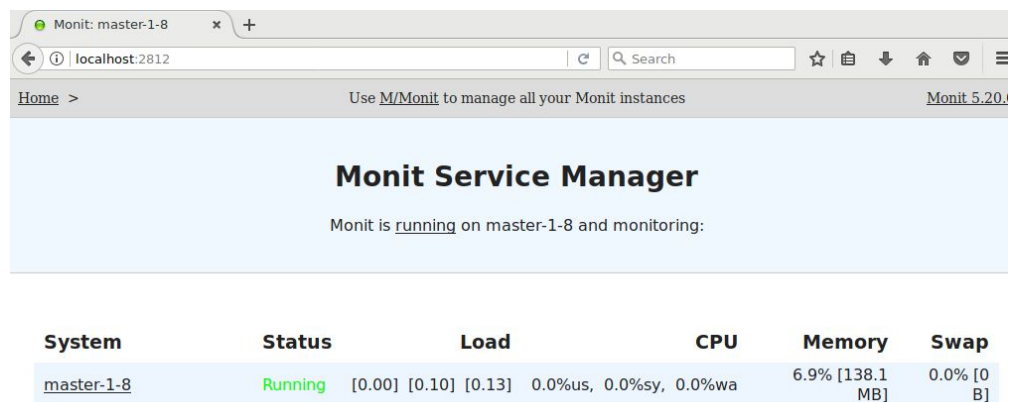
Comenzamos instalando Monit, que se encuentra en los repositorios de Debian, una vez hecho esto, iniciamos y activamos el servicio.

```
root@master-1-8:~# systemctl enable monit
monit.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable monit
root@master-1-8:~# systemctl start monit
```

Modificamos el fichero de configuración de Monit (*/etc/monit/monitrc*), para modificar las líneas necesarias para activar el servicio web de monitorización.

```
set httpd port 2812 and
  use address localhost # only accept connection from localhost
  allow localhost      # allow localhost to connect to the server and
  allow admin:monit    # require user 'admin' with password 'monit'
```

Con lo que una vez reiniciado el servicio, podemos acceder a la interfaz web.



System	Status	Load	CPU	Memory	Swap
master-1-8	Running	[0.00] [0.10] [0.13]	0.0%us, 0.0%sy, 0.0%wa	6.9% [138.1 MB]	0.0% [0 B]

Dónde, igual que en los otros servicios, podremos acceder a un resumen de las estadísticas de uso de la máquina o máquinas que se están monitorizando.

System status

Parameter	Value
Name	master-1-8
Status	Running
Monitoring status	Monitored
Monitoring mode	active
On reboot	start
Load average	[0.03] [0.03] [0.07]
Cpu	0.4%us 0.1%sy 0.0%wa
Memory usage	331.2 MB [16.5%]
Swap usage	0 B [0.0%]
Uptime	22m
Boot time	Sun, 13 Dec 2020 17:08:50
Data collected	Sun, 13 Dec 2020 17:30:16

Para activar la monitorización de Apache y SSH podemos ver que existe una carpeta con archivos de configuración por defecto, donde se encuentran ambos servicios.

```
root@master-1-8:~# ls -l -h /etc/monit/conf-available/
total 60K
-rw-r--r-- 1 root root 481 Oct 9 2019 acpid
-rw-r--r-- 1 root root 640 Oct 9 2019 apache2
-rw-r--r-- 1 root root 455 Oct 9 2019 at
-rw-r--r-- 1 root root 691 Oct 9 2019 cron
-rw-r--r-- 1 root root 602 Oct 9 2019 mdadm
-rw-r--r-- 1 root root 669 Oct 9 2019 memcached
-rw-r--r-- 1 root root 703 Oct 9 2019 mysql
-rw-r--r-- 1 root root 521 Oct 9 2019 nginx
-rw-r--r-- 1 root root 471 Oct 9 2019 openntpd
-rw-r--r-- 1 root root 950 Oct 9 2019 openssh-server
-rw-r--r-- 1 root root 885 Oct 9 2019 pdns-recursor
-rw-r--r-- 1 root root 1.4K Oct 9 2019 postfix
-rw-r--r-- 1 root root 869 Oct 9 2019 rsyslog
-rw-r--r-- 1 root root 501 Oct 9 2019 smartmontools
-rw-r--r-- 1 root root 306 Oct 9 2019 snmpd
```

Para activar la monitorización de ambos servicios será suficiente con copiar ambos archivos de `/etc/monit/conf-available` a `/etc/monit/conf-enabled` o establecer un enlace simbólico y reiniciar el servicio.

```
root@master-1-8:~# ln -s /etc/monit/conf-available/apache2 /etc/monit/conf-enabled/
root@master-1-8:~# ln -s /etc/monit/conf-available/openssh-server /etc/monit/conf-enabled/
root@master-1-8:~# systemctl restart monit
```

Una vez hecho esto, nos aseguramos que se haya hecho el enlace correctamente y comprobamos que podemos ver el estado de los servicios en la interfaz web de Monit.

```
root@master-1-8:~# ls -l -h /etc/monit/conf-enabled/
total 0
lrwxrwxrwx 1 root root 33 Dec 13 18:56 apache2 -> /etc/monit/conf-available/apache2
lrwxrwxrwx 1 root root 40 Dec 13 18:57 openssh-server -> /etc/monit/conf-available/openssh-server
```

Como se puede comprobar en la interfaz, ya monitoriza tanto SSH como Apache, aunque por alguna razón, se nos queda en *pending* con SSH.

System	Status	Load	CPU	Memory	Swap
master-1-8	Running	[0.00] [0.00] [0.00]	8.1%us, 1.5%sy, 0.0%wa	15.8% [316.5 MB]	0.0% [0 B]

Process	Status	Uptime	CPU Total	Memory Total
sshd	Initializing - start pending	-	-	-
apache	Running	0m	0.0%	4.7% [94.0 MB]