

Taller de Proyecto II

Ingeniería en Computación

Informe de avance

Monitoreo de Raspberry Pi con Prometheus y Grafana PS.4-2

Arreche Cristian - 01515/4

Blasco Federico - 01678/4

26 de octubre de 2020

Facultad de Informática
Universidad Nacional de La Plata



1. Proyecto

El proyecto a desarrollar se basa en la recolección y análisis de métricas de múltiples Raspberry Pi (RPi). El enfoque principal del proyecto es la configuración y uso de las herramientas de software Grafana y Prometheus.

El objetivo principal del proyecto es poder conectar varias RPi y realizar un monitoreo en tiempo real, donde se pueda obtener información de uso de hardware y software de cada una de ellas. Esta información debe ser representada de manera agradable mediante la utilización de gráficos y paneles, donde también se podrán configurar alertas y/o notificaciones de las métricas en base a eventos definidos.

Algunas de las métricas que pueden ser monitoreadas son:

- Uso de la CPU.
- Métricas del sistema (Número de CPUs, memoria total).
- Métricas del SO.

Objetivos originales:

- Instalar y configurar Grafana y Prometheus en varias RPi
- Visualizar la información de la base de datos con Grafana
- Configurar un panel de observación útil y auto explicativo
- Configurar alertas/notificaciones para detectar eventos y/o potenciales problemas con las RPi

Objetivos modificados por evaluaciones de la cátedra:

- Las RPi serán simuladas mediante el software de virtualización VirtualBox, logrando una buena aproximación del hardware real.

Objetivos modificados por otras decisiones:

- Configurar un contenedor de Docker para los servicios de monitoreo y análisis.
- Instalar Grafana y Prometheus en un servidor aparte para no perder performance de las RPi
- Instalar y configurar Node-exporter en cada una de las RPi para enviar los datos al servidor

2. Materiales y presupuesto

Se trata de un proyecto casi exclusivamente de software, donde las herramientas de software a utilizar son gratuitas. Para el desarrollo del proyecto no es necesario ningún elemento de hardware adicional provisto por la cátedra. De la misma manera, tampoco es necesario la suscripción de servicios en la nube.

De este modo, no se necesita ningún tipo de presupuesto, el desarrollo del proyecto se puede llevar a cabo de una forma totalmente gratuita. El único hardware necesario son las computadoras de ambos integrantes del grupo que desarrollan el proyecto.

Entonces, el hardware a utilizar será:

- Una PC (PC1) con el software VirtualBox donde se simularán las RPi.
- Una PC (PC2) usada como servidor con los servicios de Prometheus y Grafana.

Se decidió hacerlo de esta manera no solo por el costo que conlleva comprar varias Raspberry Pi, sino también por la facilidad para compartir los avances y pruebas entre ambos integrantes del grupo. De esta manera, más allá de no contar con el hardware, se puede lograr una buena aproximación de cómo debería funcionar en el hardware real.

La simulación del hardware (RPi) se hará utilizando el software VirtualBox, creando varias máquinas virtuales con Raspberry OS (Raspbian) instalado. Cada una de las máquinas virtuales tendrá su puerto 22 asociado a un puerto de la máquina host, esto nos permitirá tener acceso remoto a las máquinas virtuales a través del protocolo SSH.

3. Esquema gráfico del proyecto

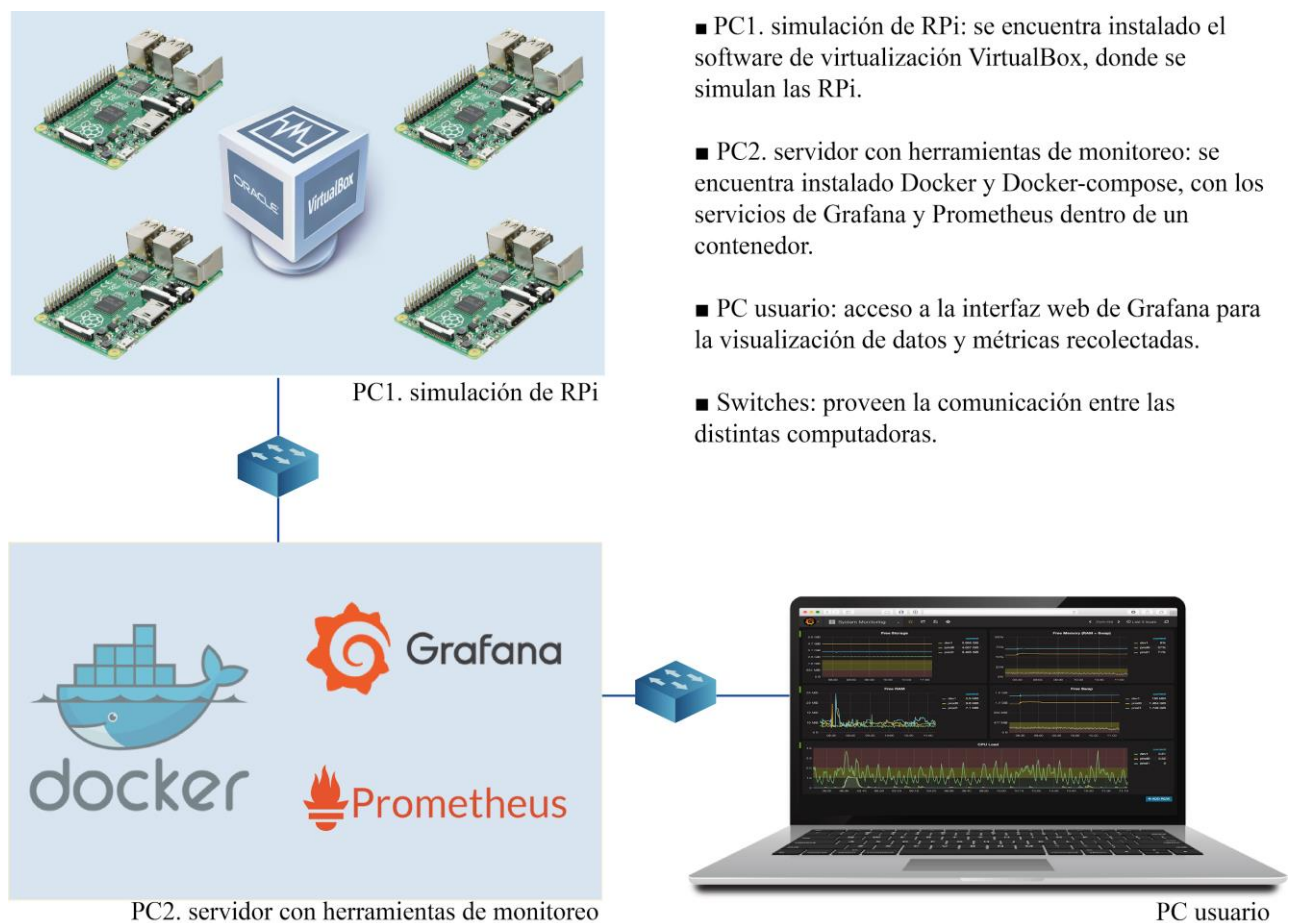


Figura 1. Esquema gráfico del proyecto

El grado de avance del proyecto hasta el día de la fecha es muy bueno. Recorriendo algunos de los puntos de los objetivos, se llevan realizadas las siguientes tareas:

- Simular varias RPi mediante el software de virtualización VirtualBox, logrando una buena aproximación del hardware real: En la PC1 ya se encuentra instalado el software VirtualBox con las instancias correspondientes para simular varias RPi. Cada una de las máquinas virtuales cuenta con un servicio de SSH para poder conectarse de manera remota.

- Configurar contenedor de Docker para instalar las herramientas de monitoreo y análisis: En la PC2 ya se encuentra instalado Docker y Docker-compose, a fin de poder levantar los servicios de Grafana y Prometheus en un contenedor.
- Instalar Grafana y Prometheus en un servidor aparte para no perder performance de las RPi: En la PC2, ya se encuentran funcionando tanto los servicios de Grafana (localhost:3000) como los de Prometheus (localhost:9090). (Ver inciso 5, figuras 2 y 3)

De esta manera, nos encontramos con todas las herramientas de software instaladas correctamente, siendo esto lo que generalmente más tiempo demanda, dejándonos el camino libre para proceder a trabajar con los objetivos restantes.

Para ver los avances de manera cronológica, y más capturas de los ítems mencionados anteriormente, visitar la Wiki de trabajo en:

<https://github.com/tpll/2020-ps-4.2-prometheus-grafana/wiki/Bit%C3%A1cora>

4. Problemas y soluciones

Conexión por SSH

Uno de los problemas con el que nos encontramos fue a la hora de establecer la comunicación mediante el protocolo SSH. Hubo que investigar de qué manera abrir puertos en el software VirtualBox, y luego de tener los puertos de las máquinas virtuales asociados a puertos de la máquina host, tuvimos que abrir los puertos del router para que se pueda acceder a las terminales SSH desde fuera de la red. Ya que, sin esto, no habría manera de ingresar a las máquinas virtuales ambos integrantes del grupo. (Ver inciso 5, figuras 4, 5, 6 y 7).

Creación de contenedores Docker

Otro problema menor que tuvimos fue cuando tratamos de crear los contenedores de Docker, con Docker-compose, ya que no se podían descargar las imágenes de Grafana ni Prometheus. Luego de investigar un poco y tratar de hacer un `docker pull prom/prometheus` descubrimos que el problema era que había que tener una cuenta en Docker hub para poder descargar las imágenes. Luego de crear la cuenta y usar el comando `docker login` no tuvimos más problemas.

Creación del .yaml

Al buscar en internet un archivo .yaml a fin de levantar los servicios de Grafana y Prometheus, casi todos tenían incluidos varios servicios más, por lo que el .yaml deberá ser modificado y “pasarle en limpio” más adelante para que solo tenga los servicios necesarios. El archivo .yaml utilizado actualmente es el siguiente:

```
version: "2"
services:
  influxdb:
    image: influxdb
    ports:
      - "8086:8086"
    restart: always

  mysql:
    image: mysql
    ports:
      - "3306:3306"
    environment:
      MYSQL_ROOT_PASSWORD: root

  grafana:
    image: grafana/grafana
    ports:
      - "3000:3000"
    links:
      - influxdb
      - mysql
      - prometheus
    restart: always

  telegraf:
    image: telegraf
    links:
      - influxdb
    volumes:
      - ./telegraf.conf:/etc/telegraf/telegraf.conf
    restart: always

  prometheus:
    image: prom/prometheus
    ports:
      - "9090:9090"
    restart: always
```

5. Documentación en formato gráfico y video

El video con la demostración de las tareas realizadas hasta el momento está disponible para descargar en formato .mp4 en:

[https://github.com/tpll/2020-ps-4.2-prometheus-grafana/blob/master/2.Informe%20de%20avance/Video informe avance.mp4](https://github.com/tpll/2020-ps-4.2-prometheus-grafana/blob/master/2.Informe%20de%20avance/Video%20informe%20avance.mp4)

A continuación, se encuentran disponibles algunas capturas de pantalla de diferentes tareas realizadas:

- Inciso 3: Grado de avance del proyecto

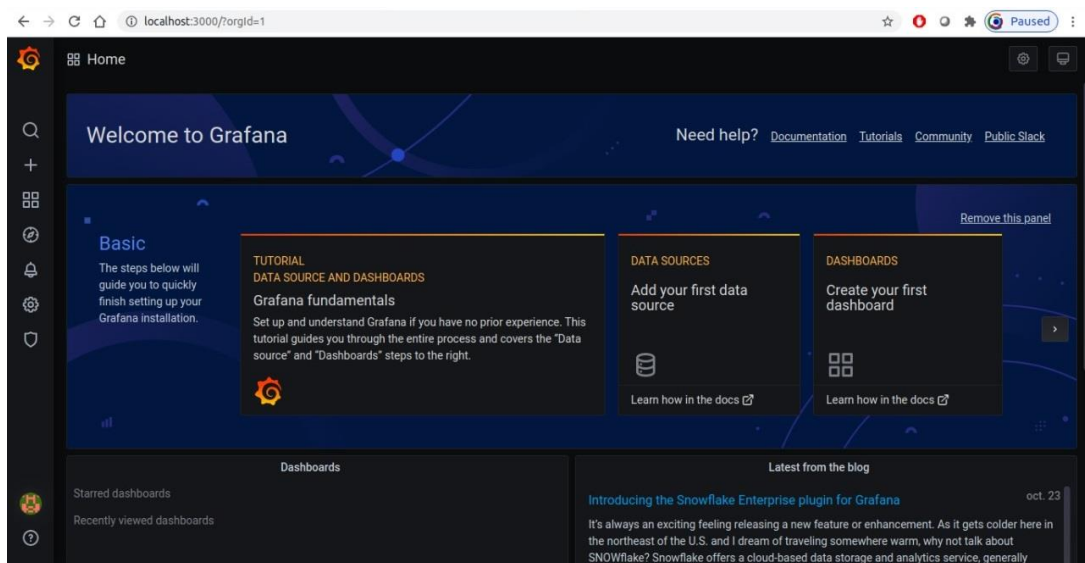


Figura 2. Grafana corriendo en el puerto 3000

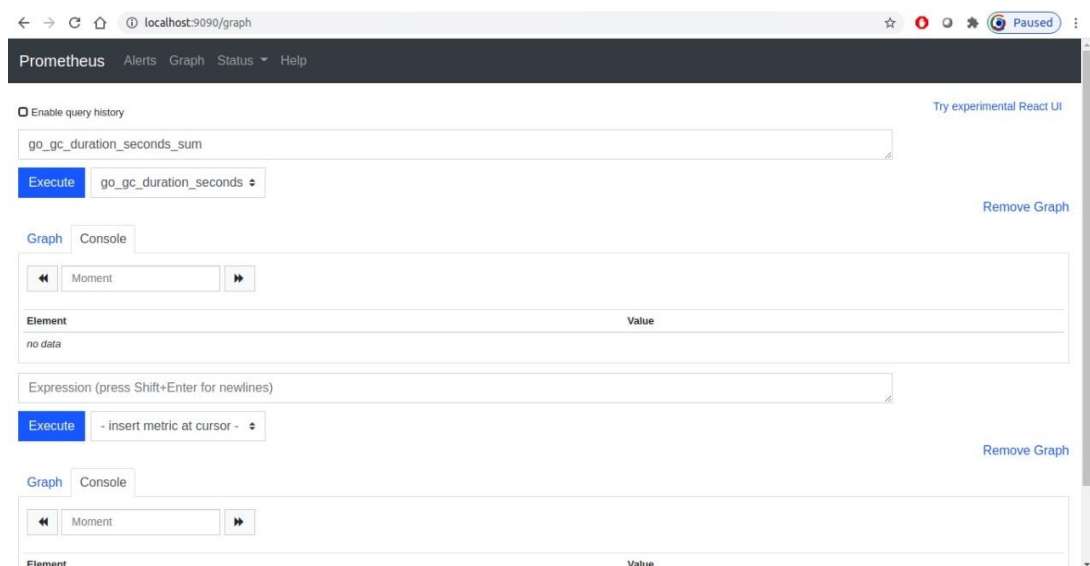


Figura 3. Prometheus corriendo en el puerto 9090

- Inciso 4: Conexión por SSH

SSH Raspi1	2222	2222		TCP/UDP	2222	2222	192.168.0.6	wanbridge	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SSH Raspi2	2223	2223		TCP/UDP	2223	2223	192.168.0.6	wanbridge	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figura 4. Puertos abiertos en el router

```

pi@raspberrypi: ~/Desktop
pi@raspberrypi:~$ ls
Desktop  Downloads  HolaSoyRaspi1  Music  Public  Videos
Documents  hola  MagPi  Pictures  Templates
pi@raspberrypi:~$ cd Desktop/
pi@raspberrypi:~/Desktop$ ls
Taller2
pi@raspberrypi:~/Desktop$ mkdir soyCrisDesdeChascomus1
pi@raspberrypi:~/Desktop$ ls
soyCrisDesdeChascomus1  Taller2
pi@raspberrypi:~/Desktop$

```

Figura 5. Conexión por SSH a la máquina virtual 1

```

pi@raspberrypi: ~/Desktop
pi@raspberrypi:~$ ls
Desktop  Downloads  MagPi  Pictures  Templates
Documents  HolaSoyRaspi2  Music  Public  Videos
pi@raspberrypi:~$ cd Desktop/
pi@raspberrypi:~/Desktop$ ls
pi@raspberrypi:~/Desktop$ mkdir soyCrisDesdeChascomus2
pi@raspberrypi:~/Desktop$ ls
soyCrisDesdeChascomus2
pi@raspberrypi:~/Desktop$

```

Figura 6. Conexión por SSH a la máquina virtual 2

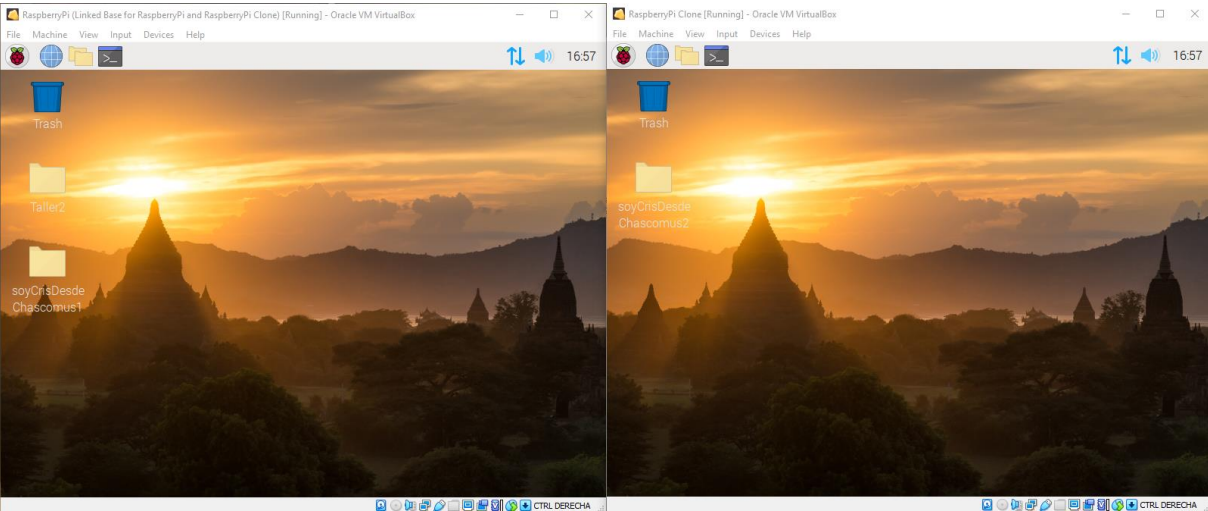


Figura 7. Máquinas virtuales con las carpetas creadas