



Noviembre de 2022
Laboratorio 7
Monitoreo y supervisión de redes

INTRODUCCIÓN

Topología

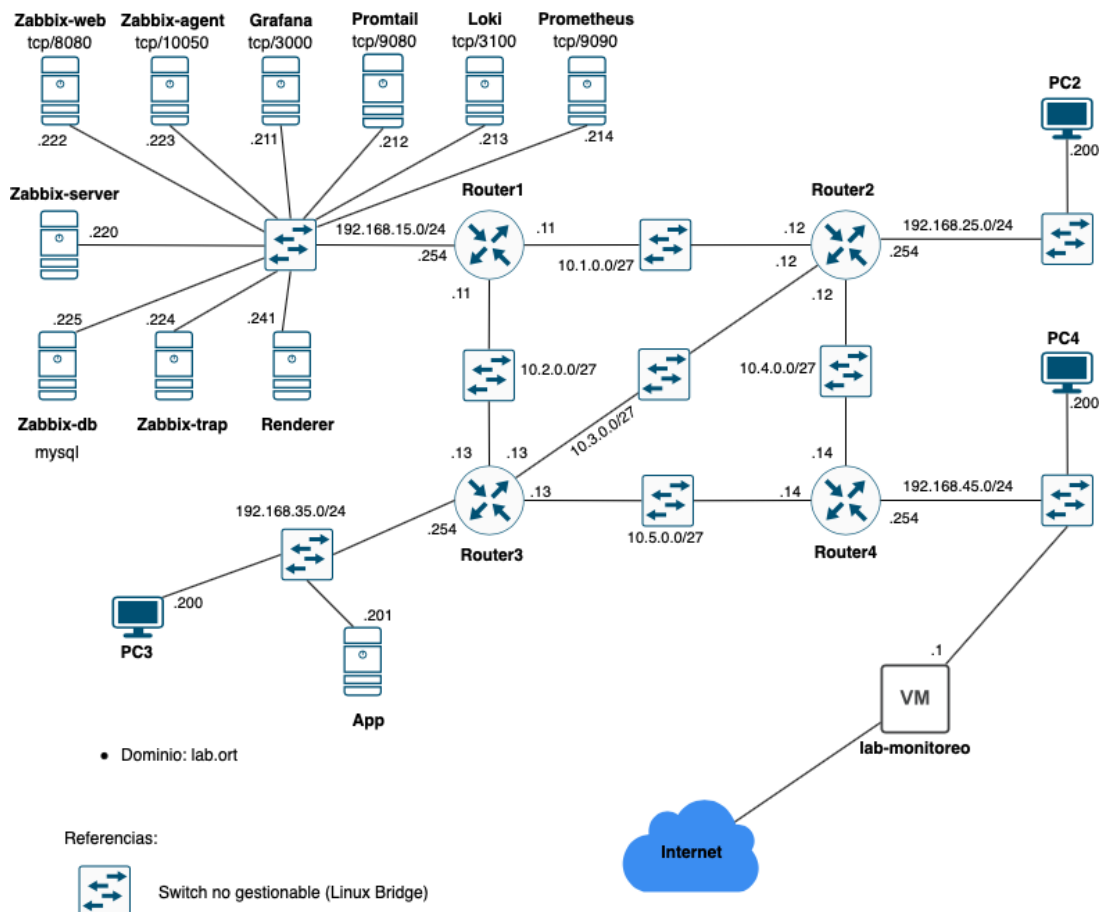


Imagen 1: Topología de red

La conexión se realiza mediante SSH a la máquina virtual lab-monitoreo. Por defecto se presenta en el puerto tcp/2222 de la interfaz de red de la computadora personal del estudiante

Usuario	estudiante
Password	estudiante
Puerto	2222

En este práctico trabajaremos con el módulo de alertas de Grafana y con el chatbot Hubot para extraer datos de Grafana

1. Conectarse por SSH a la VM lab-monitoreo con redirección X11 habilitada

2. Si ingresa con MobaXterm viene habilitado por defecto

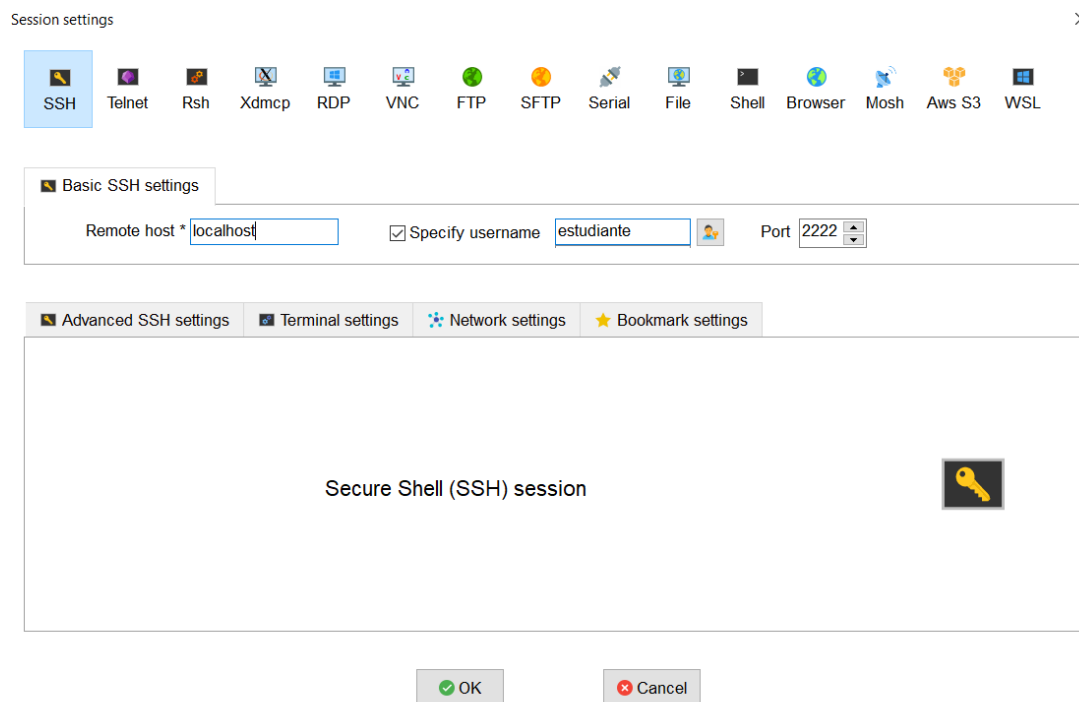


Imagen 2: Ejemplo en MobaXterm

3. Para habilitar X11 en Putty:

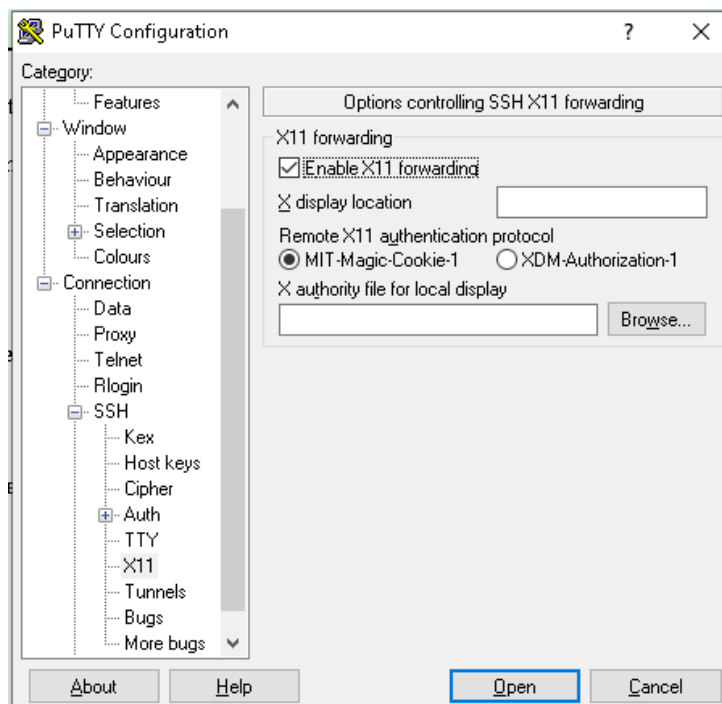


Imagen 3: Configuración Putty

4. Primero que nada prepararemos el ambiente para la primera parte de este laboratorio, para eso ejecutemos:

- *lab6*

5. Abramos un navegador en nuestra PC
6. Se ha desplegado una aplicación de prueba, en la que nos basaremos para monitorear, para acceder ingresemos a <http://localhost:2225>

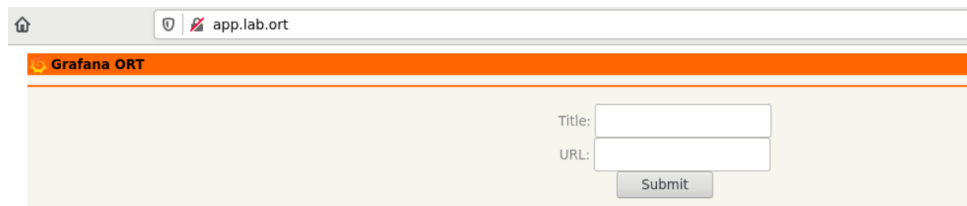


Imagen 4: App

7. En otra pestaña ingresemos a Grafana: <http://localhost:2224>
8. Credenciales de acceso:

Usuario	admin
Password	admin

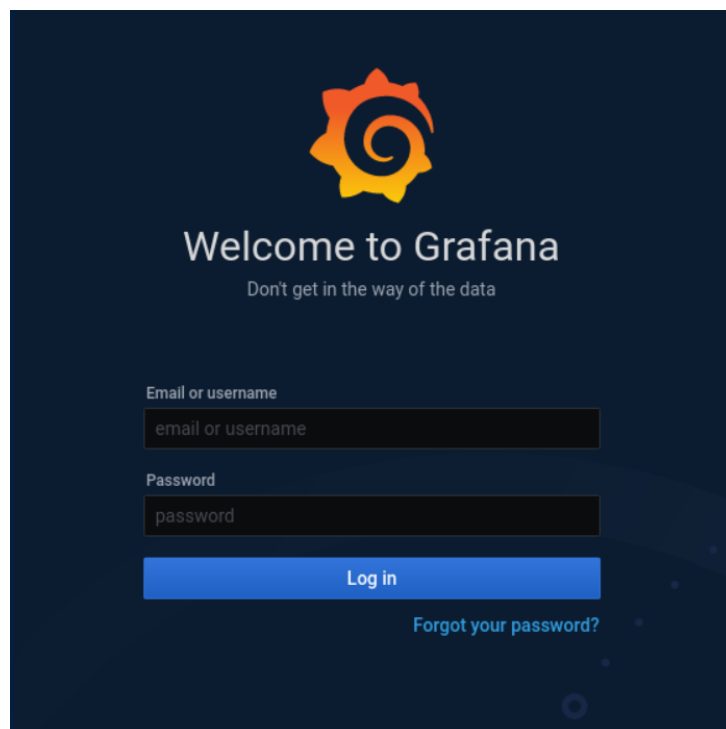


Imagen 5: Login Grafana

PRÁCTICOS

Práctico 1

Alertas Grafana

Veremos cómo definir alertas y envío de notificaciones

1. Vayamos a crear un Notification Channel

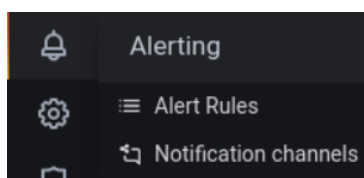


Imagen 6: Grafana

2. Add channel

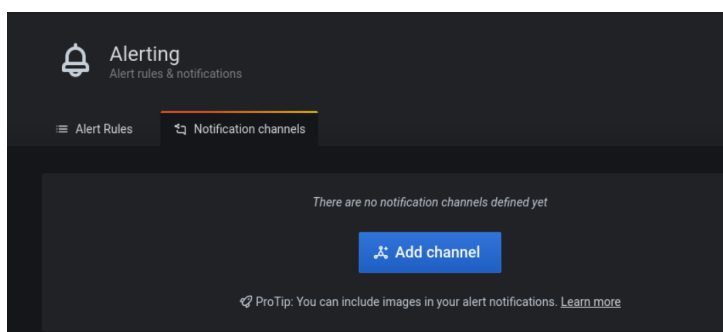


Imagen 7: Grafana

3. Utilicemos los datos del Bot de Telegram creado para las notificaciones de Zabbix (ver laboratorio 5)

Alerting
Alert rules & notifications

Alert Rules Notification channels

Edit notification channel

Name
Telegram

Type
Telegram

BOT API Token
5617944688:AAEPQxBHrZMza0QXle4ocoKNRwz1HcHe8c4

Chat ID
Integer Telegram Chat Identifier
5784403396

Notification settings

☒ Default
Use this notification for all alerts

☒ Include image
Captures an image and include it in the notification

☐ Disable Resolve Message

Imagen 8: Grafana

- Name: Telegram
 - Type: Telegram
 - BOT API Token: Token de Telegram generado en el laboratorio 5
 - Chat ID: ID de usuario de Telegram
 - Default: marcado
 - Include image: marcado
4. Probemos la integración con el botón "Test"
 5. Para crear una Alert Rule vayamos a Dashboards->Manage-> [nombre del dashboard creado en laboratorio 6]
 6. Sobre un panel seleccione editar
 7. En la pestaña Alert podrá definir las condiciones de alerta

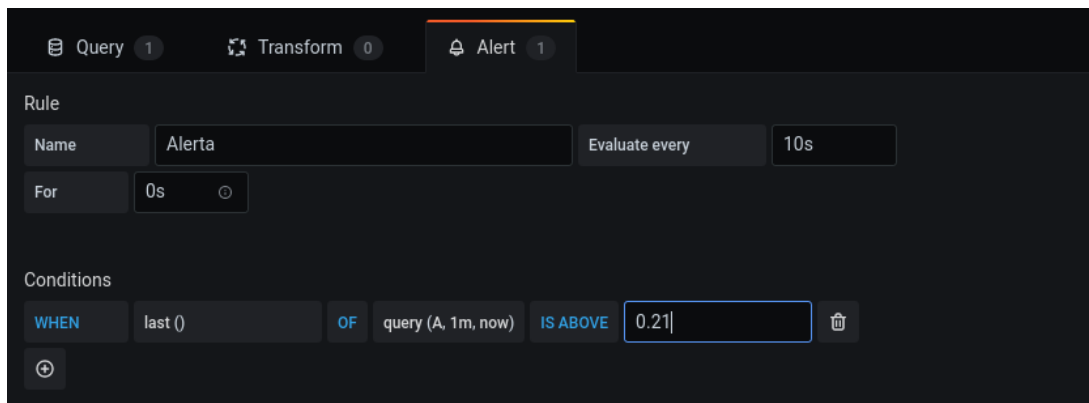


Imagen 9: Grafana Alerts

- Name: Alerta
- Evaluate every: 10s
- For: 0s
- When: last()
- OF: query(A, 1m, now)
- Is above: 0.21
- Send To: Telegram

8. Vuelva a ingresar a <http://localhost:2225> e interactúe con la aplicación hasta que sobrepase el umbral y genere alertas

Práctico 2

Configuración Hubot

Hubot es un chatbot que permite integrarse con múltiples aplicaciones. En este práctico veremos cómo integrarlo con Grafana para consumir datos, y con Telegram para poder interactuar con él

1. En la topología se agregó un contenedor donde correrá hubot (hubot.lab.ort)
2. Primero deberemos obtener una API KEY en Grafana
 - Ingresar a <http://localhost:2224>
 - Configuration->API Keys->New API Key



- [illegible]

4. Luego de realizado esto ejecutemos en lab-monitoreo:
 - *lab7*
 - Solicitará durante la ejecución la API Key recién generada para Grafana. Y enseguida el token del bot que desean usar de Telegram (para simplificar usar el mismo que para las alertas)
 - Luego de finalizada la preparación del laboratorio 7 nos encontraremos en condiciones de iniciar el bot
5. Conectarse por ssh a `hubot.lab.ort`

- `ssh hubot.lab.ort`
 - `hubot: # rc-service hubot start`
6. Luego de iniciado el bot vayamos a Telegram y comencemos a interactuar
7. Para ver los comandos válidos se puede ver la guía oficial: <https://grafana.com/tutorials/integrate-hubot/>
- `ort help`: nos indicará todos los comandos aceptados por nuestro bot

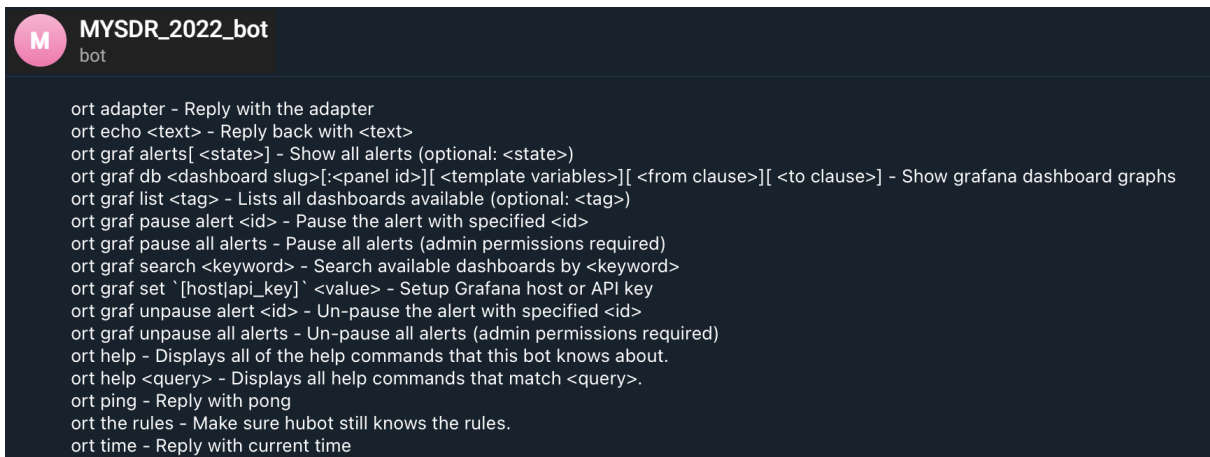


Imagen 12: BOT ort help

- `ort graf alerts`: nos mostrará las alertas que hay definidas en Grafana y el último cambio de estado
- `ort graf list`: nos mostrará la lista de dashboards que hay definidos
- `ort graf db` (nombre dashboard): nos devolverá la URL y gráfico de todos los paneles que componen el dashboard indicado