EBENEZER IT · GUÍAS PASO A PASO

GUÍA 12 · EPISODIOS 27, 28 Y 29

Vigilancia: que el servidor te avise a ti

Uptime Kuma comprueba cada minuto que tus servicios responden y te avisa si alguno se cae. Prometheus y Grafana guardan y dibujan cómo va el servidor y cada contenedor. Y un chequeo diario te escribe solo cuando algo necesita atención.

~3 horas
la primera vez, en tres sesiones

Requisito
Guías 01, 02, 03 y 06

Ubuntu 24.04 LTS
Docker · Prometheus · Grafana

www.ebenezer.org.es · Alejandro Quevedo · Versión pública 1.0 · noviembre 2026

Índice

Antes de empezar

0 Qué vas a construir y valores de esta guía

Parte A · Uptime Kuma: ¿responde?

Paso 01 Qué vigila cada herramienta

Paso 02 La carpeta, la red y las versiones

Paso 03 Uptime Kuma

Paso 04 Publicarlo solo para la VPN

Paso 05 Los monitores

Paso 06 Los avisos

Paso 07 Vigilar desde fuera

Parte B · Prometheus y Grafana: ¿cómo va?

Paso 08 Los medidores: node-exporter y cAdvisor

Paso 09 Prometheus

Paso 10 Grafana

Paso 11 Los paneles

Paso 12 Comprobar que mide el servidor

Parte C · El chequeo diario

Paso 13 El script de chequeo

Paso 14 Que te escriba solo si hay algo

Paso 15 Comprobación final e instantánea

Anexos

A Problemas frecuentes

B El docker-compose.yml completo

C Chuleta de PromQL

Cómo usar esta guía

Tienes aquí **todos los pasos y comandos**, en orden. La explicación de cada paso (el porqué de cada línea) está en el **vídeo del episodio**.

Para copiar los comandos sin errores, usa la versión web: **https://www.ebenezer.org.es/guia-12-vigilancia.html**. Al copiar desde un PDF se pueden romper los saltos de línea y las comillas.

ANTES DE EMPEZAR

Qué vas a construir

Tres herramientas que responden a tres preguntas distintas. Todas corren en Docker, ninguna publica puertos a Internet y los dos paneles web solo se abren por la VPN.

Pregunta	Herramienta	Cómo te enteras
¿Mis servicios **responden**?	**Uptime Kuma**	Un aviso al momento si algo deja de responder, y otro cuando vuelve
¿**Cómo va** el servidor? CPU, memoria, disco, red, cada contenedor	**Prometheus** (guarda) + **Grafana** (dibuja), con node-exporter y cAdvisor midiendo	Lo miras cuando quieres, con 30 días de historia
¿Hay algo que **necesite atención**? Disco casi lleno, un certificado a punto de caducar, un servicio parado	**El chequeo diario** (un script)	Un correo por la mañana, solo si hay algo

Valores de esta guía

Valor	Qué es
TU_DOMINIO	Tu dominio
TU_BUZON	Tu buzón (Guía 08), donde llegarán los avisos
TU_GATEWAY	La dirección del servidor en la red `monitoring` (paso 2), por ejemplo `172.19.0.1`
TU_SUBRED	La subred de `monitoring` (paso 2), por ejemplo `172.19.0.0/16`

Recordatorio

- Haz una instantánea en IONOS antes de empezar.
- Necesitas la VPN conectada para abrir los paneles (Guía 06).
- Bloques **verdes** en el servidor, **azules** en tu PC, **morados** son contenido de archivos.

PARTE A

Uptime Kuma: ¿responde?

Montamos la base común (carpeta, red y versiones) y Uptime Kuma, que comprueba cada minuto que tus servicios responden y te avisa si no. Corresponde al **episodio 27**.

Paso 1 Episodio 27

Qué vigila cada herramienta

Por qué: cada herramienta mira una cosa distinta, y ninguna sustituye a las otras.

Si pasa esto...	Te enteras por...
La web deja de responder o da error 502	Uptime Kuma, en un minuto
Postfix se para y no entra correo	Uptime Kuma (monitor del puerto 25)
El disco se va llenando poco a poco	El chequeo diario (aviso al 80 %) y Grafana (la gráfica)
Un certificado no se renovó	Uptime Kuma (caducidad del certificado) y el chequeo diario
Un contenedor gasta cada vez más memoria	Grafana (memoria por contenedor)
Se cae el servidor entero	**Nada de lo anterior**: todo corre en él. Paso 7

Comprueba antes de seguir

Sabes qué te avisará de cada problema y por qué hace falta vigilar también desde fuera.

Paso 2 Episodio 27

La carpeta, la red y las versiones

Por qué: todo irá en una carpeta, `/opt/monitoring`, con un solo `docker-compose.yml` y una red de Docker propia, `monitoring`, para que los contenedores se hablen entre ellos sin salir a Internet.

EN EL SERVIDOR

```
sudo mkdir -p /opt/monitoring/prometheus
docker network create monitoring
docker network inspect monitoring -f '{{range .IPAM.Config}}Subred: {{.Subnet}}  Puerta de
enlace: {{.Gateway}}{{end}}'
```

SALIDA ESPERADA

```
Subred: 172.19.0.0/16  Puerta de enlace: 172.19.0.1
```

Versiones fijas, en un archivo .env

Las imágenes con `:latest` cambian de versión el día que hagas `docker compose pull`, aunque sea una versión mayor con cambios incompatibles. Fijamos cada versión en un archivo `.env` que el compose lee solo. Para elegirla, mira la última versión estable en la página de cada imagen en Docker Hub (pestaña *Tags*), o en la de lanzamientos de su proyecto en GitHub.

EN EL SERVIDOR

```
sudo tee /opt/monitoring/.env > /dev/null <<'EOF'
# Versiones fijas: cámbialas a mano cuando decidas actualizar
KUMA_VERSION=1
NODE_EXPORTER_VERSION=v1.11.1
CADVISOR_VERSION=v0.55.1
PROMETHEUS_VERSION=v3.13.0
GRAFANA_VERSION=13.1.0
# Contraseña del usuario admin de Grafana (larga, de tu gestor)
GRAFANA_PASSWORD=TU_CONTRASENA_GRAFANA
EOF
sudo chmod 600 /opt/monitoring/.env
```

Comprueba antes de seguir

Existe la red `monitoring`, tienes apuntadas su subred y su puerta de enlace, y `.env` tiene permisos `-rw-------`.

Paso 3 Episodio 27

Uptime Kuma

Por qué: empezamos el compose solo con Uptime Kuma; en la parte B le añadimos el resto.

EN EL SERVIDOR

```
sudo nano /opt/monitoring/docker-compose.yml
```

CONTENIDO DEL ARCHIVO · /OPT/MONITORING/DOCKER-COMPOSE.YML

```
services:
  uptime-kuma:
    image: louislam/uptime-kuma:${KUMA_VERSION}
    container_name: uptime-kuma
    restart: unless-stopped
    volumes:
      - uptime-kuma-data:/app/data
    ports:
      - "127.0.0.1:3001:3001"
    networks:
      - monitoring

volumes:
  uptime-kuma-data:

networks:
  monitoring:
    external: true
```

EN EL SERVIDOR

```
cd /opt/monitoring
sudo docker compose up -d
sudo docker compose ps
```

Comprueba antes de seguir

`uptime-kuma` aparece `Up`, y `sudo ss -tlnp | grep 3001` muestra `127.0.0.1:3001`.

Paso 4 Episodio 27

Publicarlo solo para la VPN

Por qué: con Nginx, como el panel de la VPN de la Guía 06: certificado propio, y el snippet `solo-vpn.conf` para que solo se abra con la VPN conectada.

En el DNS de IONOS, un registro **A** `uptime` → `TU_IP`. Después:

EN EL SERVIDOR

```
sudo tee /etc/nginx/sites-available/uptime.TU_DOMINIO > /dev/null <<'EOF'
# HTTP: Let's Encrypt y redirección a HTTPS
server {
    listen 80;
    listen [::]:80;
    server_name uptime.TU_DOMINIO;

    location ^~ /.well-known/acme-challenge/ {
        root /var/www/letsencrypt;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

# HTTPS: Uptime Kuma (contenedor en 127.0.0.1:3001), solo por la VPN
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name uptime.TU_DOMINIO;

    ssl_certificate     /etc/letsencrypt/live/uptime.TU_DOMINIO/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/uptime.TU_DOMINIO/privkey.pem;
    include snippets/tls.conf;
    include snippets/solo-vpn.conf;

    add_header Strict-Transport-Security "max-age=86400" always;
    add_header X-Content-Type-Options "nosniff" always;

    location / {
        proxy_pass http://127.0.0.1:3001;
        include snippets/proxy.conf;
    }
}
EOF
```

EN EL SERVIDOR

```
sudo ln -s /etc/nginx/sites-available/uptime.TU_DOMINIO /etc/nginx/sites-enabled/
sudo certbot certonly --webroot -w /var/www/letsencrypt -d uptime.TU_DOMINIO --cert-name
uptime.TU_DOMINIO
sudo nginx -t && sudo systemctl reload nginx
```

Con la VPN conectada, abre `https://uptime.TU_DOMINIO` y crea el usuario administrador, con una contraseña larga de tu gestor.

Comprueba antes de seguir
Con la VPN, el panel abre; sin ella, da `403 Forbidden`.

Paso 5 Episodio 27

Los monitores

Por qué: un monitor por cada cosa que, si falla, quieres saberlo.

Monitor	Tipo	Dirección	Detalles
Web	HTTP(s)	`https://www.TU_DOMINIO`	Marca **Notificar sobre la caducidad del certificado**
Correo entrante	Puerto TCP	`mail.TU_DOMINIO` · 25	Postfix
Envío de correo	Puerto TCP	`mail.TU_DOMINIO` · 587	*Submission*
Buzones	Puerto TCP	`mail.TU_DOMINIO` · 993	Dovecot
Chat	Puerto TCP	`TU_DOMINIO` · 5222	ejabberd (Guía 10)
VPN	**Push**	—	Ver 5.2: WireGuard es UDP y su panel solo abre por la VPN
Cada web o app más	HTTP(s)	Su dirección	WordPress, Chatwoot… con la caducidad del certificado

Kuma llega desde la red monitoring
Cuando Kuma pide una web de tu propio servidor, Nginx no ve la IP pública sino la de Kuma en la red `monitoring` (TU_SUBRED). Si esa web está cerrada con `allow` / `deny`, el monitor dará **403**: añade `allow TU_SUBRED;` antes del `deny` (o en `solo-vpn.conf`, si quieres vigilar todos los paneles). Para ver quién lo rechaza: `sudo grep 'Uptime-Kuma' /var/log/nginx/*access*.log | grep ' 403 '`.

5.2 · La VPN, con un monitor Push

WireGuard usa UDP, que Kuma no sabe comprobar, y el panel de wg-easy solo abre por la VPN. Se vigila al revés: el servidor le dice a Kuma «estoy bien» cada minuto, y si deja de oírlo, avisa. En Kuma: tipo

Push, nombre `VPN WireGuard`, intervalo `120`. Copia el código que va después de `/api/push/` en la URL que te muestra, y en el servidor:

EN EL SERVIDOR

```
read -rs -p "Código del push: " T; echo
sudo tee /usr/local/bin/kuma-vpn.sh > /dev/null <<EOF
#!/bin/sh
# Avisa a Uptime Kuma de que WireGuard funciona (monitor Push "VPN WireGuard")
if docker exec wg-easy wg show wg0 > /dev/null 2>&1; then
    curl -fsS -m 10 "http://127.0.0.1:3001/api/push/$T?status=up&msg=OK" > /dev/null
fi
EOF
unset T
sudo chmod 700 /usr/local/bin/kuma-vpn.sh
echo '* * * * * root /usr/local/bin/kuma-vpn.sh' | sudo tee /etc/cron.d/kuma-vpn
```

Comprueba antes de seguir

Todos los monitores en verde, también `VPN WireGuard`.

Paso 6 Episodio 27

Los avisos

Por qué: Kuma avisa por muchos canales.

6.1 · Por correo

Crea en PostfixAdmin un buzón solo para esto, por ejemplo `avisos@TU_DOMINIO`, con su propia contraseña. Después, en Kuma: **Configuración** → **Notificaciones** → **Configurar notificación**:

Campo	Valor
Tipo	SMTP (correo)
Servidor / Puerto	`mail.TU_DOMINIO` · `587`, seguridad **STARTTLS**
Usuario y contraseña	`avisos@TU_DOMINIO` y su contraseña
Remitente	`avisos@TU_DOMINIO`: el mismo buzón, o Postfix lo rechazará (Guía 09, remitentes)
Destinatario	TU_BUZON
Por defecto / Aplicar a todos	Marcadas: se usa en todos los monitores, también en los nuevos

Pulsa **Probar**: debe llegarte un correo de prueba.

6.2 · Un segundo canal

Cualquiera que llegue a tu móvil sin pasar por tu servidor. Con **Telegram**:

1. En Telegram, escribe `/newbot` a **@BotFather**, ponle nombre y un usuario acabado en `bot`. Te da un **token**: es una contraseña, cópialo solo en Kuma.
2. Abre tu bot y mándale cualquier mensaje (un bot no puede escribir a quien no le ha escrito antes).
3. En Kuma, notificación de tipo **Telegram**: el token, y en **ID del chat** pulsa **Obtener automáticamente**. Marca **Activado por defecto** y **Aplicar en todos los monitores existentes**.

Comprueba antes de seguir

Los dos avisos de prueba llegan.

Paso 7 Episodio 27

Vigilar desde fuera

Por qué: todo lo anterior corre en el servidor.

Opción	Cómo
Otro servidor tuyo	Un Uptime Kuma en otro VPS (o en un equipo de casa siempre encendido) con monitores hacia este. Si tienes dos servidores, que se vigilen el uno al otro
Un servicio externo gratuito	Hay servicios de monitorización con un plan gratuito que comprueban unas cuantas direcciones cada pocos minutos y avisan por correo o al móvil

Basta con dos monitores externos: la web (`https://www.TU_DOMINIO`) y el correo (puerto 25 de `mail.TU_DOMINIO`). Si alguno de los dos falla desde fuera, el servidor tiene un problema serio.

Comprueba antes de seguir

Al menos un monitor que no corre en este servidor vigila la web.

PARTE B

Prometheus y Grafana: ¿cómo va?

Dos medidores (uno para el servidor, otro para los contenedores), Prometheus que guarda lo que miden y Grafana que lo dibuja. Corresponde al **episodio 28**.

Paso 8 Episodio 28

Los medidores: node-exporter y cAdvisor

Por qué: node-exporter mide el servidor (CPU, memoria, disco, red) y **cAdvisor** mide cada contenedor.

8.1 · node-exporter con la red del servidor

Un contenedor tiene su propia red aislada. Si node-exporter no usa la del servidor, mide **el tráfico de su contenedor**, no el de tu VPS, y las gráficas de red engañan. La solución es `network_mode: host`; pero entonces escucharía el 9100 en todas las interfaces, también la de Internet. Lo limitamos a `TU_GATEWAY`, la dirección del servidor en la red `monitoring`, que es por donde llega Prometheus.

Añade al compose, debajo de `uptime-kuma`, los bloques `node-exporter` y `cadvisor` del Anexo B. Y en el cortafuegos:

EN EL SERVIDOR

```
sudo ufw allow from TU_SUBRED to TU_GATEWAY port 9100 proto tcp comment 'Prometheus → node-exporter'
```

8.2 · cAdvisor, a dieta

Comprueba antes de seguir

Los bloques están en el compose y la regla de UFW añadida.

Paso 9 Episodio 28

Prometheus

Por qué: pregunta a los medidores cada 15 segundos y guarda 30 días de datos.

EN EL SERVIDOR

```
sudo nano /opt/monitoring/prometheus/prometheus.yml
```

CONTENIDO DEL ARCHIVO · /OPT/MONITORING/PROMETHEUS/PROMETHEUS.YML

```
global:
  scrape_interval: 15s
  evaluation_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'node'
    static_configs:
      - targets: ['TU_GATEWAY:9100']

  - job_name: 'cadvisor'
    static_configs:
      - targets: ['cadvisor:8080']
```

Añade al compose el bloque `prometheus` del Anexo B (y su volumen), y arranca:

EN EL SERVIDOR

```
cd /opt/monitoring
sudo docker compose up -d
sleep 40
docker exec prometheus wget -qO- 'http://localhost:9090/api/v1/targets' | grep -o
'"scrapeUrl":"[^"]*"\|"health":"[^"]*"'
```

Comprueba antes de seguir

Los tres destinos (`prometheus` , `node` y `cadvisor`) salen con `"health":"up"` .

Paso 10 Episodio 28

Grafana

Por qué: el panel que dibuja las gráficas.

Añade al compose el bloque `grafana` del Anexo B (y su volumen). La contraseña la lee del `.env` .

EN EL SERVIDOR

```
cd /opt/monitoring
sudo docker compose up -d
```

En el DNS, un registro **A** `monitor` → TU_IP, y el sitio de Nginx igual que el de Uptime Kuma, cambiando el nombre y el puerto:

EN EL SERVIDOR

```
sudo tee /etc/nginx/sites-available/monitor.TU_DOMINIO > /dev/null <<'EOF'
# HTTP: Let's Encrypt y redirección a HTTPS
server {
    listen 80;
    listen [::]:80;
    server_name monitor.TU_DOMINIO;

    location ^~ /.well-known/acme-challenge/ {
        root /var/www/letsencrypt;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}

# HTTPS: Grafana (contenedor en 127.0.0.1:3000), solo por la VPN
server {
    listen 443 ssl http2;
    listen [::]:443 ssl http2;
    server_name monitor.TU_DOMINIO;

    ssl_certificate     /etc/letsencrypt/live/monitor.TU_DOMINIO/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/monitor.TU_DOMINIO/privkey.pem;
    include snippets/tls.conf;
    include snippets/solo-vpn.conf;

    add_header Strict-Transport-Security "max-age=86400" always;
    add_header X-Content-Type-Options "nosniff" always;

    location / {
        proxy_pass http://127.0.0.1:3000;
        include snippets/proxy.conf;
    }
}
EOF
```

EN EL SERVIDOR

```
sudo ln -s /etc/nginx/sites-available/monitor.TU_DOMINIO /etc/nginx/sites-enabled/
sudo certbot certonly --webroot -w /var/www/letsencrypt -d monitor.TU_DOMINIO --cert-name
monitor.TU_DOMINIO
sudo nginx -t && sudo systemctl reload nginx
```

Comprueba antes de seguir

Con la VPN, `https://monitor.TU_DOMINIO` muestra la entrada de Grafana y entras con `admin` y la contraseña del `.env`.

Paso 11 Episodio 28

Los paneles

Por qué: Grafana no trae gráficas: hay que decirle de dónde salen los datos y qué dibujar.

11.1 · La fuente de datos

1. **Conexiones** → **Fuentes de datos** → **Añadir** → **Prometheus**.
2. URL: `http://prometheus:9090` (por nombre, dentro de la red `monitoring`).
3. **Guardar y probar**: debe decir que la consulta funciona.

11.2 · El panel del servidor

1. **Paneles de control** → **Nuevo** → **Importar**.
2. Escribe el número **1860** (*Node Exporter Full*, el panel más usado para node-exporter) y pulsa **Cargar**.
3. Elige como fuente la de Prometheus e importa.

11.3 · Un panel de contenedores

1. **Nuevo** → **Nuevo panel de control** → **Añadir visualización**, con la fuente de Prometheus.
2. En el editor, modo **Código**, la consulta de abajo; en **Opciones** → **Leyenda** → **Personalizado**, `{{name}}`.
3. Visualización: **Serie temporal**; unidad: **megabytes**. Guárdalo como **Contenedores**.

CONTENIDO DEL ARCHIVO · CONSULTA DE GRAFANA

```
container_memory_working_set_bytes{name=~".+"} / 1024 / 1024
```

Comprueba antes de seguir

El panel del servidor muestra CPU, memoria, disco y red, y el de contenedores una línea por contenedor.

Paso 12 Episodio 28

Comprobar que mide el servidor

Por qué: un error frecuente es que node-exporter mida su contenedor y no el servidor: las gráficas salen, pero con datos falsos.

EN EL SERVIDOR

```
docker exec prometheus wget -qO- 'http://localhost:9090/api/v1/query?
query=node_network_receive_bytes_total' | grep -o '"device":"[^"]*"' | sort -u
ip -br link | awk '{print $1}'
```

Deben coincidir: la de Internet (`ens6` o parecida), `docker0`, los puentes `br-...`. Si solo sale `eth0` y `lo`, node-exporter no tiene `network_mode: host` (paso 8).

Comprueba antes de seguir

Prometheus mide la interfaz de Internet del servidor.

PARTE C

El chequeo diario

Un script que repasa el servidor cada mañana (servicios, contenedores, puertos, disco, memoria, certificados, actualizaciones) y te escribe solo si algo necesita atención. Corresponde al **episodio 29**.

Paso 13 Episodio 29

El script de chequeo

Por qué: lo descargamos de la web, comprobamos que es el original con su huella SHA-256, y lo adaptamos a tu servidor.

EN EL SERVIDOR

```
cd /tmp
curl -fsSO https://www.ebenezer.org.es/descargas/chequeo-servidor.sh
curl -fsSO https://www.ebenezer.org.es/descargas/SHA256SUMS.txt
sha256sum -c --ignore-missing SHA256SUMS.txt
sudo install -m 750 chequeo-servidor.sh /usr/local/bin/chequeo-servidor.sh
rm chequeo-servidor.sh SHA256SUMS.txt
sudo nano /usr/local/bin/chequeo-servidor.sh
```

Arriba del todo está la configuración. Ajusta:

Variable	Qué poner
`DOMAIN`	`TU_DOMINIO`
`SERVICES`	Tus servicios: `ssh nginx fail2ban docker` y los que tengas (`postfix dovecot rspamd ejabberd crowdsec` ...)
`DOCKER_CONTAINERS`	Tus contenedores: `wg-easy uptime-kuma prometheus grafana node-exporter cadvisor` ...
`TCP_PORTS`	`"ssh:22" "https:443"` y, si tienes correo, `"smtp:25" "submission:587" "imaps:993"`
Umbrales	Los de serie están bien; `SWAP_WARN` súbelo a 80 si tu servidor siempre tiene algo de swap usada

EN EL SERVIDOR

```
sudo /usr/local/bin/chequeo-servidor.sh
```

Comprueba antes de seguir

El chequeo termina con `TODO EN ORDEN`, o con avisos que entiendes y has decidido dejar.

Paso 14 Episodio 29

Que te escriba solo si hay algo

Por qué: una tarea programada lo ejecuta cada mañana.

EN EL SERVIDOR

```
sudo tee /etc/cron.d/chequeo-servidor > /dev/null <<'EOF'
# Chequeo diario del servidor a las 07:15; correo solo si hay incidencias
15 7 * * * root /usr/local/bin/chequeo-servidor.sh --plain > /var/log/chequeo-servidor.log
2>&1; [ $? -ne 0 ] && (echo "Subject: [Chequeo] Incidencias en $(hostname)"; echo; cat
/var/log/chequeo-servidor.log) | /usr/sbin/sendmail TU_BUZON@TU_DOMINIO
EOF
```

Prueba que el correo llega, sin esperar a mañana:

EN EL SERVIDOR

```
(echo "Subject: [Chequeo] Prueba"; echo; echo "Si lees esto, los avisos funcionan.") |
/usr/sbin/sendmail TU_BUZON@TU_DOMINIO
```

Comprueba antes de seguir

El correo de prueba llega a tu bandeja.

Paso 15 Episodio 29

Comprobación final e instantánea

Por qué: un repaso de las tres herramientas.

EN EL SERVIDOR

```
cd /opt/monitoring && sudo docker compose ps
sudo ss -tlnp | grep -E ':(3000|3001|9100)\b'
docker exec prometheus wget -qO- 'http://localhost:9090/api/v1/targets' | grep -o '"health":"
[^"]*"' | sort | uniq -c
sudo /usr/local/bin/chequeo-servidor.sh --quiet
```

Comprobación	Resultado correcto
Contenedores	Los cinco `Up`
Puertos	3000 y 3001 en `127.0.0.1`; 9100 en TU_GATEWAY. Ninguno en `0.0.0.0`
Prometheus	`3 "health":"up"`
Paneles	Solo con la VPN
Avisos	Probados por correo y por el segundo canal
Desde fuera	Al menos un monitor externo

Después, en IONOS, la instantánea `12-vigilancia`.

Comprueba antes de seguir

Si algo se cae, te enteras en un minuto; si algo va mal poco a poco, lo ves en las gráficas o te llega un correo por la mañana.

ANEXOS

Para cuando algo no sale

`sudo docker compose logs --tail 50 NOMBRE` desde `/opt/monitoring` cuenta casi todo de cada contenedor.

Anexo A · Problemas frecuentes

Lo que ves	Qué pasa	Qué hacer
`open /opt/monitoring/.env: permission denied`	El `.env` es solo de root	`sudo docker compose ...`
Grafana: *No data* en todo	La fuente de datos apunta mal	URL `http://prometheus:9090`, y Prometheus en la red `monitoring`
El destino `node` en `down`	Falta la regla de UFW o la dirección no coincide	Paso 8: la regla y `--web.listen-address` con la misma TU_GATEWAY que `prometheus.yml`
`network_mode: host` da error al arrancar	Quedó un `networks:` en el bloque de node-exporter	Quítalo: con la red del servidor no puede estar en otra
Las gráficas de red no cuadran con lo que esperas	node-exporter mide su contenedor	Paso 12
Uptime Kuma se queda cargando	Faltan las cabeceras de WebSocket en Nginx	Que el sitio incluya `snippets/proxy.conf` (Guía 03), que trae `Upgrade` y `Connection`
El aviso por correo de Kuma falla	Usuario, puerto o remitente	587 con STARTTLS, y el remitente igual al usuario
cAdvisor gasta cientos de MB	Faltan las opciones del paso 8.2	Añádelas y `sudo docker compose up -d cadvisor`

Anexo B · El docker-compose.yml completo

Así queda `/opt/monitoring/docker-compose.yml` al terminar la parte B. Las versiones y la contraseña salen del `.env` del paso 2.

CONTENIDO DEL ARCHIVO · /OPT/MONITORING/DOCKER-COMPOSE.YML

```
services:
  uptime-kuma:
    image: louislam/uptime-kuma:${KUMA_VERSION}
    container_name: uptime-kuma
    restart: unless-stopped
    volumes:
      - uptime-kuma-data:/app/data
    ports:
      - "127.0.0.1:3001:3001"
    networks:
      - monitoring

  node-exporter:
    image: prom/node-exporter:${NODE_EXPORTER_VERSION}
    container_name: node-exporter
    restart: unless-stopped
    command:
      - '--path.rootfs=/host'
      - '--web.listen-address=TU_GATEWAY:9100'
    network_mode: host
    pid: host
    volumes:
      - '/:/host:ro,rslave'

  cadvisor:
    image: gcr.io/cadvisor/cadvisor:${CADVISOR_VERSION}
    container_name: cadvisor
    restart: unless-stopped
    command:
      - --docker_only=true
      - --housekeeping_interval=30s
      - --store_container_labels=false
      - --
disable_metrics=advtcp,cpu_topology,cpuset,hugetlb,memory_numa,percpu,process,referenced_memory
,resctrl,sched,tcp,udp
    mem_limit: 256m
    privileged: true
    devices:
      - /dev/kmsg
    volumes:
      - /:/rootfs:ro
      - /var/run:/var/run:ro
      - /sys:/sys:ro
      - /var/lib/docker/:/var/lib/docker:ro
      - /dev/disk/:/dev/disk:ro
    networks:
      - monitoring

  prometheus:
```

```
    image: prom/prometheus:${PROMETHEUS_VERSION}
    container_name: prometheus
    restart: unless-stopped
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--storage.tsdb.path=/prometheus'
      - '--storage.tsdb.retention.time=30d'
    volumes:
      - ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml:ro
      - prometheus-data:/prometheus
    networks:
      - monitoring

  grafana:
    image: grafana/grafana:${GRAFANA_VERSION}
    container_name: grafana
    restart: unless-stopped
    depends_on:
      - prometheus
    environment:
      - GF_SECURITY_ADMIN_USER=admin
      - GF_SECURITY_ADMIN_PASSWORD=${GRAFANA_PASSWORD}
      - GF_SERVER_ROOT_URL=https://monitor.TU_DOMINIO
      - GF_USERS_ALLOW_SIGN_UP=false
    volumes:
      - grafana-data:/var/lib/grafana
    ports:
      - "127.0.0.1:3000:3000"
    networks:
      - monitoring

volumes:
  uptime-kuma-data:
  prometheus-data:
  grafana-data:

networks:
  monitoring:
    external: true
```

Para actualizar una imagen

Cambia su versión en el `.env`, y después `sudo docker compose pull NOMBRE && sudo docker compose up -d NOMBRE`. Lee antes las notas de la versión si es un salto de versión mayor.

Anexo C · Chuleta de PromQL

PromQL es el lenguaje de consultas de Prometheus. Se escribe en Grafana (modo **Código**) o en **Explorar**.

Para ver...	Consulta
Memoria por contenedor (MB)	`container_memory_working_set_bytes{name=~".+"} / 1024 / 1024`
CPU por contenedor (%)	`rate(container_cpu_usage_seconds_total{name=~".+"}[5m]) * 100`

Para ver...	Consulta
Disco usado en / (%)	`100 - node_filesystem_avail_bytes{mountpoint="/"} / node_filesystem_size_bytes{mountpoint="/"} * 100`
Tráfico de entrada de Internet (Mbit/s)	`rate(node_network_receive_bytes_total{device="ens6"}[5m]) * 8 / 1000000`
¿Responden los medidores?	`up` (1 = sí, 0 = no)