

# Uptime Kuma

## Manual de despliegue herramienta Uptime Kuma

Este manual sirve como una guía paso a paso para la instalación y configuración de Uptime Kuma, una herramienta de monitoreo de código abierto. A lo largo de este manual, exploraremos cómo implementar esta solución para supervisar la disponibilidad de servicios y sitios web, permitiéndote recibir alertas en tiempo real sobre cualquier interrupción.

Con Uptime Kuma, obtendrás una visión clara y proactiva del estado de tus sistemas, garantizando la estabilidad y el rendimiento que tus proyectos y clientes merecen.

### 1. Instalación de Docker Compose y dependencias en el servidor.

Ejecuta las siguientes órdenes para preparar el entorno:

```
sudo apt update
sudo apt upgrade -y
```

[Página oficial de instalación de Docker, caso Ubuntu.](#) En caso de instalación mediante los pasos de la página, en el manual se tiene un resumen con los comandos a utilizar para la instalación mediante el repositorio de Docker de fuentes APT.

Agregar la clave GPG oficial de Docker

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

Añadir el repositorio de Docker a las fuentes de APT

```
echo \  
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]  
https://download.docker.com/linux/ubuntu \  
  $(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \  
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update
```

Instalar Docker y complementos en su versión actual.

```
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Probar la instalación de Docker.

```
sudo docker run hello-world
```

Respuesta esperada del terminal:

```
ubuntu@ubuntu:~$ sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
17eec7bbc9d7: Pull complete  
Digest: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31  
Status: Downloaded newer image for hello-world:latest  
  
Hello from Docker!  
This message shows that your installation appears to be working correctly.  
  
To generate this message, Docker took the following steps:  
1. The Docker client contacted the Docker daemon.  
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.  
   (amd64)  
3. The Docker daemon created a new container from that image which runs the  
   executable that produces the output you are currently reading.  
4. The Docker daemon streamed that output to the Docker client, which sent it  
   to your terminal.  
  
To try something more ambitious, you can run an Ubuntu container with:  
$ docker run -it ubuntu bash  
  
Share images, automate workflows, and more with a free Docker ID:  
https://hub.docker.com/  
  
For more examples and ideas, visit:  
https://docs.docker.com/get-started/
```

## 1.1. Configuración de docker post instalación.

Agregar el usuario actual al grupo docker:

```
sudo usermod -aG docker $USER
```

Habilitar el inicio automático de Docker con systemd.

```
sudo systemctl enable docker.service
sudo systemctl enable containerd.service
```

Respuesta esperada del terminal:

```
ubuntu@ubuntu:~$ sudo systemctl enable docker.service
sudo systemctl enable containerd.service
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker
```

## 2. Instalación de uptime Kuma con servicio docker.

- Clonar el repositorio de Uptime Kuma

```
git clone https://github.com/louislam/uptime-kuma.git
```

Respuesta esperada del terminal:

```
ubuntu@ubuntu:~$ git clone https://github.com/louislam/uptime-kuma.git
Cloning into 'uptime-kuma' ...
remote: Enumerating objects: 35403, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 35403 (delta 13), reused 8 (delta 8), pack-reused 35382 (from 2)
Receiving objects: 100% (35403/35403), 29.06 MiB | 665.00 KiB/s, done.
Resolving deltas: 100% (26865/26865), done.
ubuntu@ubuntu:~$ ls
uptime-kuma
ubuntu@ubuntu:~$ cd uptime-kuma/
ubuntu@ubuntu:~/uptime-kuma$ ls
CNAME          CONTRIBUTING.md  extra          package-lock.json  server          tsconfig.json
CODE_OF_CONDUCT.md  db              index.html     public             src
compose.yaml    docker          LICENSE        README.md          test
config         ecosystem.config.js  package.json  SECURITY.md        tsconfig-backend.json
```

- Desplegar Uptime Kuma en Docker

```
docker run -d --restart=always -p 3001:3001 -v uptime-kuma:/app/data --name laboratorio-kuma louislam/uptime-kuma:1
```

Verifica que el contenedor esté en ejecución

```
docker ps
```

```
ubuntu@ubuntu:~/uptime-kuma$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                                  CREATED        STATUS
PORTS         NAMES
5bb706c91b09   louislam/uptime-kuma:1              "/usr/bin/dumb-init ..."            About a minute ago    Up About a minute (healthy)
0.0.0.0:3001->3001/tcp, [::]:3001->3001/tcp   laboratorio-kuma
```

- Acceso desde el navegador

Abre en tu navegador en el siguiente enlace y selecciona el idioma Español antes de continuar.

[http://IP\\_servidor:3001/setup](http://IP_servidor:3001/setup)

Observará la siguiente página de inicio una vez desplegado el docker:



Es fundamental crear un usuario y una contraseña robustos para proteger tu información. Elige un nombre de usuario que no revele datos personales y combina caracteres alfanuméricos, símbolos y mayúsculas/minúsculas en tu contraseña.

### 3. Primeros pasos post instalación

Verificamos la pantalla de inicio.

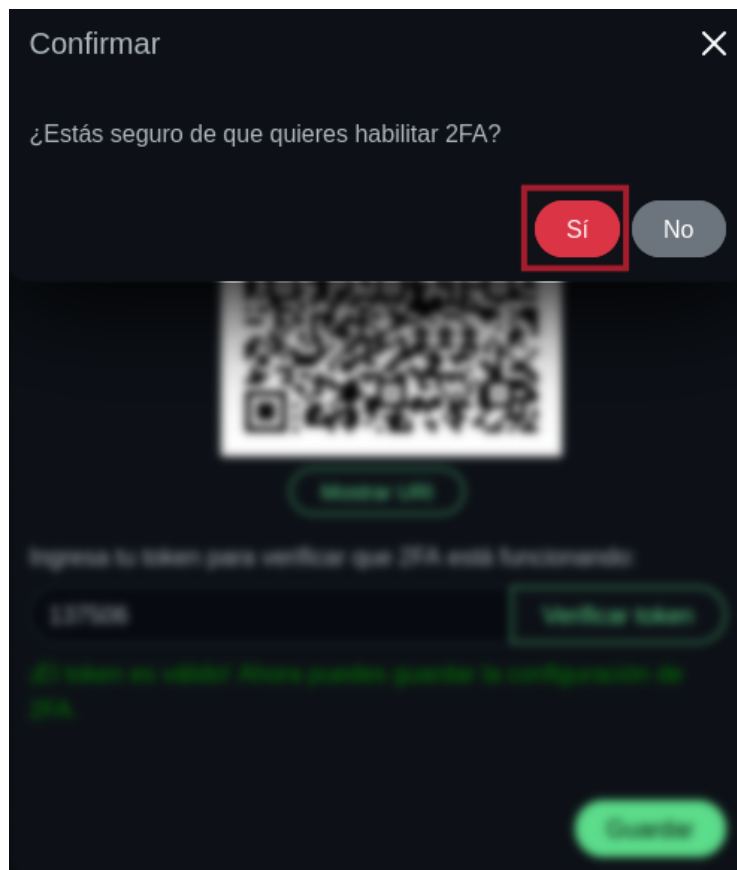


### 3.1. Doble factor de autenticación (2FA).

Para comenzar, usa la aplicación de autenticación de tu preferencia en tu dispositivo móvil y escanea el código QR que se presenta a continuación. Esta acción activará una capa de seguridad esencial que requiere un código único y temporal en cada inicio de sesión, lo cual es fundamental para proteger tu información.



Una vez escaneado el token, la aplicación proporcionará un código de 6 dígitos y se debe colocar para verificar el token y presionar el botón "Verificar token".



## 3.2. Añadir Nuevos Monitores en Uptime Kuma

Uptime Kuma permite crear distintos tipos de monitores para supervisar servicios, aplicaciones y recursos de red.

Al agregar un monitor, se despliega una lista con varias categorías y parámetros, cada una enfocada en un tipo específico de verificación.

Primeramente verificaremos que tipos de monitores existen para uptime Kuma.

### • Monitor Tipo General

Entre los más utilizados podemos destacar:

- Group: Agrupa varios monitores en una sola categoría para organizarlos y ver su estado de forma conjunta.
- HTTP(s): Comprueba si un sitio web responde correctamente mediante HTTP o HTTPS.
- DNS: Revisa la resolución de un dominio, confirmando que el servidor DNS responda correctamente.
- Contenedor de Docker: Supervisa el estado de un contenedor Docker específico.

También tenemos la opción de utilizar para monitorear lo siguiente:

- TCP Port: Verifica que un puerto TCP específico de un host esté abierto y aceptando conexiones.
- Ping: Envía paquetes ICMP para medir la latencia y disponibilidad de un host.
- HTTP(s) – Palabra clave: Valida que una palabra o frase aparezca en el contenido de una página web.
- HTTP(s) – Consulta JSON: Analiza una respuesta en formato JSON y evalúa valores clave.
- gRPC(s) – Palabra clave: Comprueba servicios gRPC y busca una cadena de texto en la respuesta.
- HTTP(s) – Browser Engine (Chrome/Chromium) (Beta): Realiza una prueba con un motor de navegador real para verificar la carga de una página (útil para sitios dinámicos con JavaScript).

#### • Monitor Tipo Pasivo

- Push: El servidor monitoreado envía (“push”) una señal a Uptime Kuma en intervalos establecidos. Ideal para sistemas que no permiten consultas externas.

#### • Monitor Tipo Específico

Entre los más utilizados podemos destacar:

- PostgreSQL: Verifica la conexión a una base de datos PostgreSQL.
- MySQL/MariaDB: Controla la disponibilidad de servidores MySQL o MariaDB.
- MongoDB: Monitorea el acceso a una base de datos MongoDB.
- Radius: Comprueba la autenticación a través del protocolo RADIUS.
- Redis: Controla el estado de un servidor de base de datos en memoria Redis.

También tenemos la opción de utilizar para monitorear lo siguiente:

- Servidor de Juegos de Steam: Monitorea la disponibilidad de servidores de juegos que utilizan el protocolo Steam.
- GameDig: Compatible con múltiples motores de juego; verifica que un servidor de juegos esté en línea.
- MQTT: Supervisa brokers del protocolo MQTT, común en IoT.
- Kafka Producer: Comprueba la conectividad y funcionamiento de productores Kafka.
- Microsoft SQL Server: Evalúa la disponibilidad de una base de datos MS SQL.

### 3.3. Procedimiento Básico

En resumen para poder añadir un nuevo monitor deben seguirse los pasos generales:

- Inicia sesión en Uptime Kuma.
- Haz clic en “Nuevo Monitor”.

- Selecciona el Tipo de Monitor de la lista.
- Completa los campos solicitados (URL, host, puerto, credenciales, etc.).
- Guarda la configuración para comenzar la supervisión.

## Ejemplo 1. Monitor HTTPs.

### General

Tipo de monitor  
HTTP(s)

Nombre sencillo  
CGII

URL  
https://cgii.gob.bo/

Intervalo de latido (Comprobar cada 300 segundos)  
300

Reintentos  
1

Número máximo de intentos antes de que el servicio se marque como CAÍDO y una notificación sea enviada.

Intervalo de reintento de latido (Reintentar cada 300 segundos)  
300

Tiempo de espera máximo de petición (Expirar después de 48 segundos)  
48

Reenviar Notificación si Caído X veces consecutivamente (Reenvío deshabilitado)  
0

### Avanzado

Notificación de Caducidad del Certificado

### Notificaciones

MonitorCGII Editar

monitoreo CGII2 Editar

Configurar notificación

### Proxy

No disponible, por favor configúralo.

Configurar Proxy

### Opciones HTTP

Método  
GET

Codificación del cuerpo  
JSON

Cuerpo

```
Ejemplo:  
{  
  "key": "value"  
}
```

Encabezados

Ejemplo:

### Avanzado

Notificación de Caducidad del Certificado

Ignorar errores TLS/SSL para sitios web HTTPS

Add the `uptime_kuma_cachebuster` parameter  
Randomly generated parameter to skip caches.

Modo invertido  
Invertir el estado. Si el servicio es alcanzable, está CAÍDO.

Redirecciones máximas  
10

Número máximo de direcciones a seguir. Establecer a 0 para deshabilitar.

Códigos de estado aceptados  
200-299

Seleccionar los códigos de estado que se consideran como respuesta exitosa.

Grupo de Monitoreo  
AGETIC

Descripción  
CGII

### Etiquetas

+ Añadir

Guardar

### Encabezados

Ejemplo:

```
{  
  "HeaderName": "HeaderValue"  
}
```

### Autenticación

Método  
Ninguno

## Identificación de parámetros importantes:

- **Intervalo de Latido** (Heartbeat Interval): Es el tiempo que Uptime Kuma espera para realizar una verificación regular del servicio (En este ejemplo, cada 300 segundos).
- **Fallo inicial**: Si la verificación falla (el servicio no responde), Uptime Kuma no lo marca inmediatamente como "caído".
- **Intervalo de reintento de latido**: Entre cada uno de estos reintentos, Uptime Kuma espera el tiempo especificado en el Heartbeat Retry Interval (en tu caso, 300 segundos). Esto es crucial para evitar falsos positivos por fallos temporales en la red o el servidor.
- **Tiempo de espera máximo de petición** (Request Timeout). Es el tiempo que Uptime Kuma esperará por una respuesta del servidor antes de considerar que el intento de conexión ha fallado.
- **Notificación de Caducidad del Certificado**. Esta opción te permite establecer un umbral, en días, para recibir una Notificación de Caducidad del Certificado antes de que el certificado SSL/TLS de un sitio web expire. Esto es útil para prevenir interrupciones en el servicio.
- **Ignorar errores TLS/SSL para sitios web HTTPS**. Es una configuración que te permite forzar a Uptime Kuma a ignorar los problemas con los certificados, como los que son autofirmados, caducados o que no coinciden con el nombre de dominio. Esto es útil para monitorear servicios en desarrollo o entornos de prueba, pero no es recomendable para sitios en producción debido a riesgos de seguridad.
- **Add the "uptime\_kuma\_cachebuster" parameter**. Uptime Kuma añade un parámetro aleatorio (?uptime\_kuma\_cachebuster=<valor>) al final de la URL que está monitoreando. Esto obliga al servidor web a servir la página directamente desde el servidor y no desde una caché, lo que te asegura que estás monitoreando el estado en tiempo real del sitio web.
- **Modo invertido** (Inverted Mode). Cambia la lógica del monitor. En lugar de notificar cuando un servicio está "caído," el monitor notificará cuando el servicio está "en línea" (UP). Esto es útil para monitorear servicios que deben estar fuera de línea, como un servidor de respaldo que solo se enciende para una tarea específica.
- **Redirecciones máximas** (Max. Redirects). Definen el número máximo de redirecciones HTTP que Uptime Kuma seguirá al monitorear una URL. Si el número de redirecciones supera este valor, el monitor marcará el estado del servicio como "caído" o con un error.
- **Códigos de estado aceptados** (Accepted Status Codes). Permiten especificar qué códigos de respuesta HTTP se consideran una respuesta exitosa. Por ejemplo, en un monitor HTTP, el código 200 OK indica éxito, pero un código 301 (redirección) o 404 (no encontrado) se consideraría un error. Puedes personalizar esta lista para que Uptime Kuma no envíe una alerta si recibe un código de estado específico.
- **Grupo de Monitoreo** (Monitor Group). Es una forma de organizar tus monitores por categorías. Puedes agrupar servicios relacionados, como todos los monitores de un proyecto específico o todos los servicios de una misma ubicación, para una mejor gestión visual y para activar o desactivar notificaciones para todo un grupo a la vez.

## Ejemplo 2. Monitor DNS.

### General

Tipo de monitor  
DNS


Nombre sencillo  
DNS AGETIC

Nombre del host  
ns.agnetic.gob.bo

Servidor de resolución  
1.1.1.1  
Cloudflare es el servidor por defecto, puedes cambiar el servidor de resolución en cualquier momento.

Puerto  
53  
Puerto servidor DNS. Por defecto al 53. Puedes cambiar el puerto en cualquier momento.

Tipo de registro  
A  
Selecciona el tipo de registro que quieres monitorizar

Conditions  
record equals   
[Add Condition](#) [Add Group](#)

Intervalo de latido (Comprobar cada 600 segundos)  
600


### Notificaciones


MonitorCGII [Editar](#)


monitoreo CGii2 [Editar](#)

[Configurar notificación](#)

### Reintentos


3   
Número máximo de intentos antes de que el servicio se marque como CAÍDO y una notificación sea enviada

Intervalo de reintento de latido (Reintentar cada 600 segundos)  
600 

Reenviar Notificación si Caído X veces consecutivamente (Reenvío deshabilitado)  
0 

### Avanzado

Modo invertido  
Invertir el estado. Si el servicio es alcanzable, está CAÍDO.

Grupo de Monitoreo  
AGETIC  [+](#)

Descripción

### Etiquetas

[Guardar](#)

## Identificación de parámetros importantes:

- **Nombre del host** (Hostname). Es el nombre de dominio que deseas monitorear. Uptime Kuma intentará resolver este nombre a través del servidor DNS que le especifiques para asegurarse de que el registro exista y sea accesible.
- **Servidor de resolución** (Resolver Server). Dirección IP del servidor DNS que Uptime Kuma utilizará para realizar la consulta. Puedes usar el servidor DNS de tu proveedor de Internet, un servidor público como el de Google (8.8.8.8) o Cloudflare (1.1.1.1), o un servidor DNS privado que estés monitoreando.
- **Puerto** (Port). Es el puerto de red que Uptime Kuma usará para conectarse al servidor de resolución. El puerto estándar para las consultas DNS es el "53". A menos que tu servidor DNS use un puerto no estándar, este valor debe ser "53".
- **Tipo de registro** (Record Type). Especifica el tipo de registro DNS que Uptime Kuma debe buscar. Los más comunes son:
  - A: Para la dirección IPv4 de un host.
  - AAAA: Para la dirección IPv6 de un host.
  - CNAME: Para un alias de un nombre de host.
  - MX: Para servidores de correo.
  - TXT: Registros de texto, a menudo usados para verificar la propiedad del dominio o para configuraciones de seguridad.
- **Condiciones** (Conditions). Reglas que definen lo que se considera una respuesta exitosa. Uptime Kuma te permite establecer condiciones para asegurar que la resolución del nombre de host no solo funcione, sino que también devuelva el valor esperado. Puedes agregar condiciones para:
  - Comparación de valor: Contiene o No contiene un valor específico.
  - Comparación de IP: Es igual a o No es igual a una dirección IP esperada.
- **Intervalo de latido** (Heartbeat Interval). Es el tiempo, en segundos, entre cada verificación de estado. Por ejemplo, si lo estableces en 600 segundos, Uptime Kuma intentará resolver el nombre de host cada 10 minutos para asegurarse de que el registro DNS esté disponible y sea correcto.

## Ejemplo 3. Grupo de monitores.

Una vez que tengas tus primeros monitores configurados, puedes organizarlos en grupos para estructurar la Uptime Kuma según la organización interna de la institución. Los parámetros de configuración para los grupos son similares a los de los monitores individuales:

Tipo de monitor MonitorCGII [Editar](#)  
monitoreo CGII2 [Editar](#)

Grupo

Nombre sencillo Configurar notificación

Intervalo de latido (Comprobar cada 300 segundos)

Reintentos   
Número máximo de intentos antes de que el servicio se marque como CAÍDO y una notificación sea enviada.

Intervalo de reintento de latido (Reintentar cada 300 segundos)

Reenviar Notificación si Caído X veces consecutivamente (Reenvío deshabilitado)

**Avanzado**

Modo invertido  
Invertir el estado. Si el servicio es alcanzable, está CAÍDO.

Grupo de Monitoreo  +

Descripción

Guardar

Una vez creado un grupo, puedes asignar un monitor a él. Simplemente, edita el monitor que deseas y selecciona el grupo correspondiente en el campo "Grupo de monitoreo"

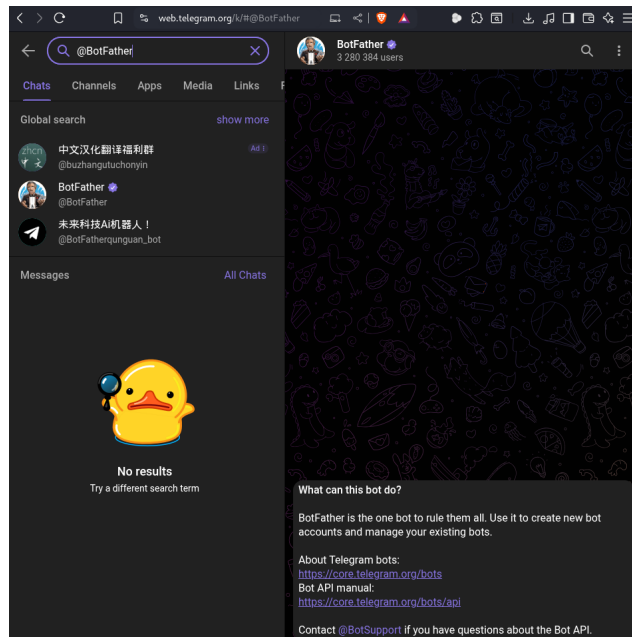
Grupo de Monitoreo

+

## 4. Configurar notificaciones por Telegram.

### 4.1. Crear el bot de Telegram

Abrir Telegram en la versión web, y buscar el usuario @BotFather. Inicia una conversación con el comando /start.



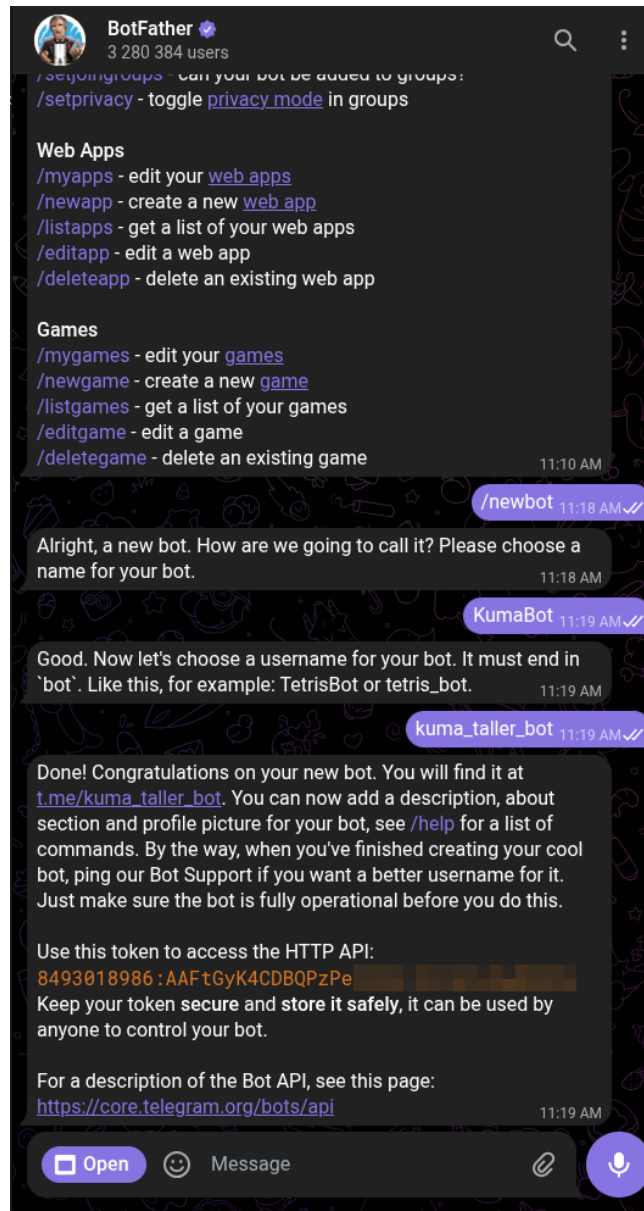
Usa el siguiente comando para crear un nuevo bot para este caso específico.

/newbot

BotFather solicitará los siguientes datos:

- **Nombre del bot:** Ej. KumaBot
- **Username del bot.** Debe terminar en "bot", por ejemplo: kuma\_taller\_bot.

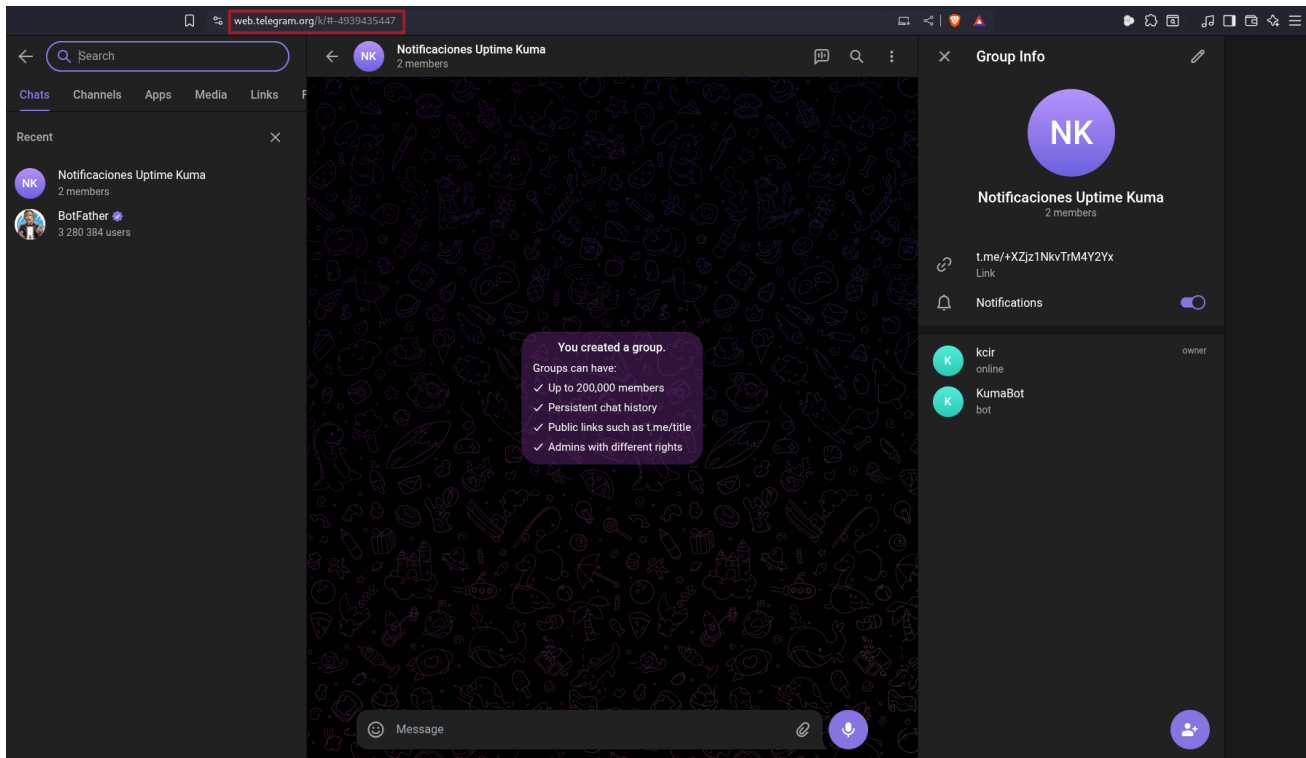
BotFather te enviará un token de la API, similar a: 1234567890:ABC-DEF1234ghIkl...



Para fines prácticos de monitoreo, se utilizará la opción de notificación a un grupo de Telegram. De esta manera, más de un usuario podrá recibir las alertas en caso de caída de servicios. Sigue estos pasos:

Crear un grupo en Telegram y agregar al nuevo bot. Busca el bot por el nombre asignado, en este caso: [@kuma\\_taller\\_bot](#).

- Enviar un mensaje de prueba al grupo.
- Verificar el ID de grupo en el navegador.



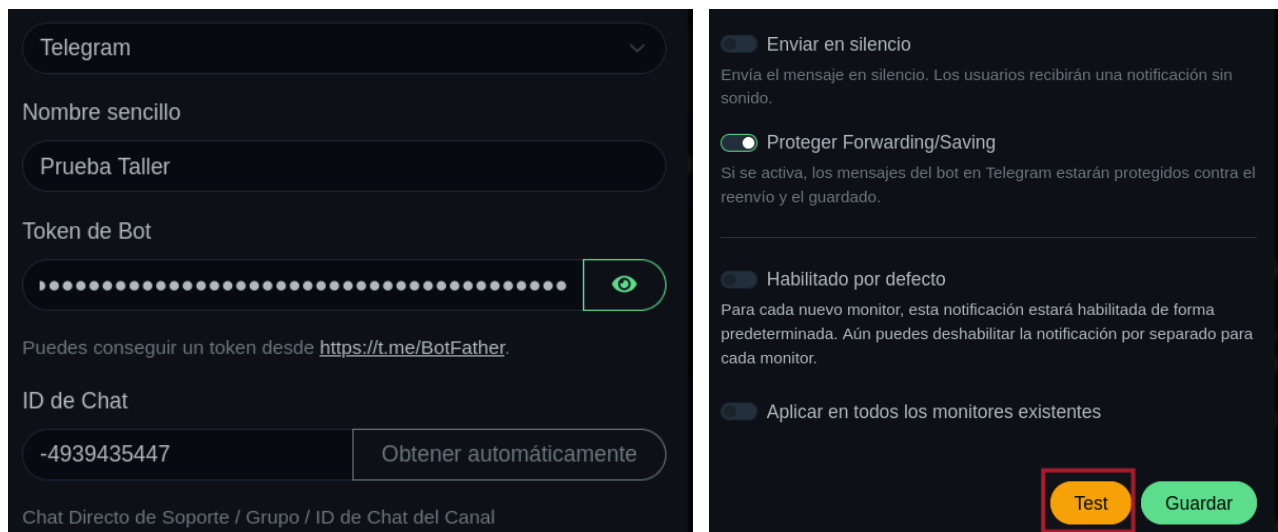
## 4.2. Integración con Uptime Kuma

Accede al panel de Uptime Kuma en el servidor desplegado: <https://localhost:3001> o el dominio configurado

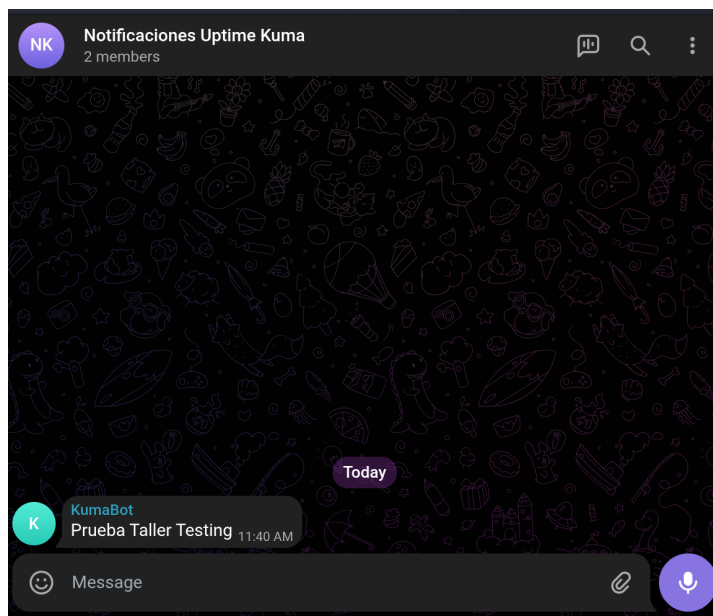
- En el menú izquierdo, ve a Configuración → Notificaciones.
- Haz clic en “Agregar una nueva notificación”.
- Selecciona Telegram en el tipo de servicio.

Completa los campos solicitados:

- Bot Token: Token obtenido de BotFather.
- Chat ID: el número obtenido del enlace de grupo en telegram web.
- Presiona “Probar” para enviar un mensaje de prueba.



Verifica que el mensaje haya sido enviado correctamente al grupo, y presiona Guardar.



Con la integración de notificaciones vía Telegram, se garantiza que múltiples miembros del equipo estén informados en tiempo real ante cualquier caída o interrupción del servicio. Esta capacidad mejora la velocidad de respuesta al contar con un canal de comunicación ágil y automatizado.

Revision #25

Created 17 septiembre 2025 16:44:45 by Ricardo Chavez

Updated 26 septiembre 2025 17:42:08 by Ricardo Chavez