



Bots de Telegram para Sysadmins

Controla tus sistemas desde Telegram

¿Que es Telegram?

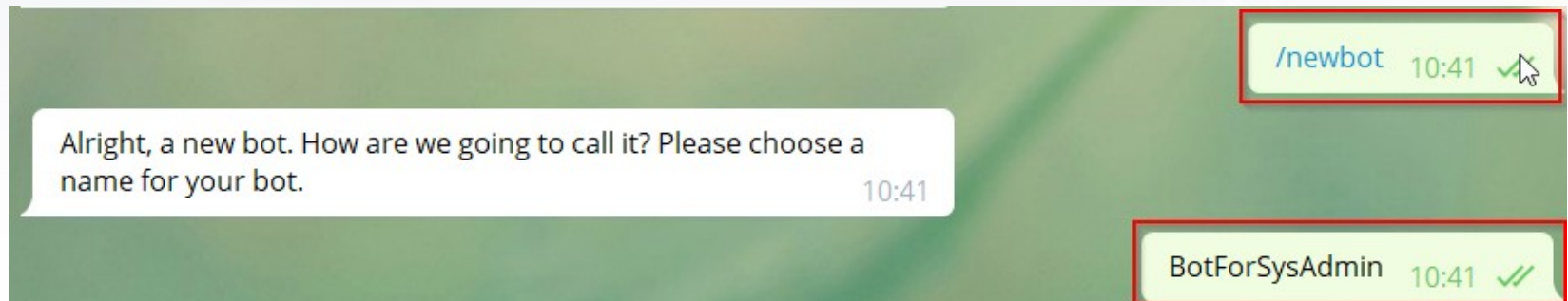
- Servicio de mensajería instantánea
- Ventajas:
 - Multiplataforma (PC, Móvil, Tablet)
 - Grupos privados/públicos
 - No se comparte número de teléfono
 - Canales unidireccionales.
 - Bots con “Botfather”

BotFather

- Bots
 - Permiten interactuar a Telegram con aplicaciones locales/remotas
- BotFather
 - Herramienta para crear Bots fácilmente
 - Documentación completa
 - <https://core.telegram.org/bots>

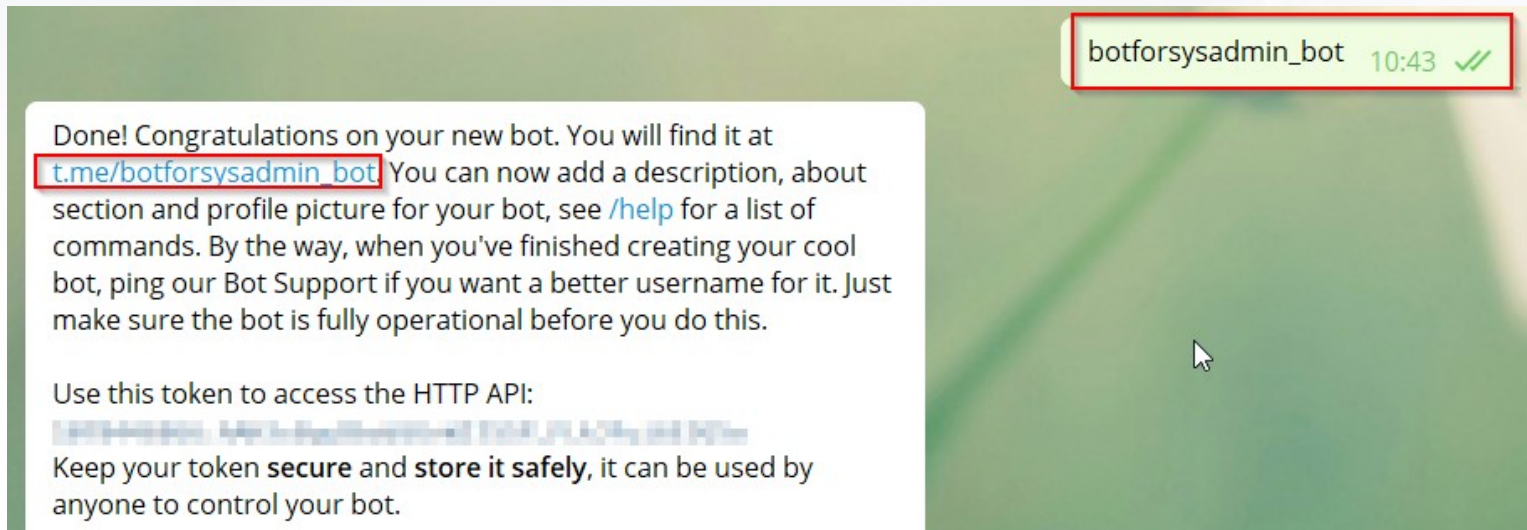
Configurando BotFather (1)

- Accedemos a BotFather mediante @BotFather o el enlace <https://t.me/BotFather>
- Escribimos la orden /newbot y damos nombre al Bot



Configurando BotFather (2)

- Ahora damos un Username al Bot. Debe ser único y acabar en `_bot`. Tras ello obtendremos:
 - URL de acceso al Bot
 - Token para acceder a la API (Debe ser secreto!)



Configurando BotFather (3)

- Para algunas operaciones necesitamos saber ID única de Usuarios y de Canales/Grupos.
- Las proporcionan los siguientes Bots
 - Usuarios: @userinfobot
 - Grupo/Canal: @ChannelIdBot

Configurando BotFather (4)

- Antes de nada, iniciar una charla con nuestro Bot y mandarle /start para que se le permita comunicarse con nosotros.
- Los ejemplos aquí presentados están disponibles en ShellScript y Python en:
 - <https://github.com/sergarb1/TelegramBots>

Nuestro primer programa (1)

- **Hola mundo:** Bot que envía el mensaje “Hola mundo” a un usuario.
- Ejemplo en ShellScript
- Necesitamos
 - Herramienta curl “sudo apt-get install curl”
 - Token
 - ID usuario receptor
 - Mensaje a enviar

Nuestro primer programa (2)

- Hola mundo (10 líneas)

```
1  #!/bin/bash
2
3  TOKEN="Escriba aqui su token" # Rellenar con el Token
4  ID_RECEPTOR="1552233" #Del estilo 1552233
5  MENSAJE="HOLA_MUNDO"
6  URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
7  # Se requiere CURL instalado ("sudo apt-get install curl")
8
9  # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
10 curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
```

Programa Status servidor (1)

- **Status servidor:** Bot que envía cada cierto tiempo status de servicio apache2.
- Orden para obtener status de Apache2
 - “sudo systemctl status apache2 2>&1”
 - 2>&1 hace que la salida de error se redirija a la salida estándar

Programa Status servidor (3)

- Servidor Status – Primera versión (11 líneas)

```
1  #!/bin/bash
2
3  TOKEN="Escriba aqui su token" # Rellenar con el Token
4  ID_RECEPTOR="1552233" #Del estilo 1552233
5  #Genera el mensaje sobre el estado de apache2
6  MENSAJE=`sudo systemctl status apache2 2>&1`
7  URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
8
9  # Se requiere CURL instalado ("sudo apt-get install curl")
10 # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
11 curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
```

Programa Status servidor (4)

- **Status servidor bucle:** modificamos el programa para que se repita en bucle.
- Usamos sleep para parar X segundos.
 - sleep 20
 - Para nuestro programa 20 segundos

Programa Status servidor (5)

- Servidor Status – Segunda versión (17 líneas)

```
1  #!/bin/bash
2
3  TOKEN="Escriba aqui su token" # Rellenar con el Token
4  ID_RECEPTOR="1552233" #Del estilo 1552233
5  URL="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web, incluyendo Token
6  # Se requiere CURL instalado ("sudo apt-get install curl")
7
8  #Bucle infinito
9  while :
10 do
11     #Duerme 20 segundos
12     sleep 20
13     #Genera cada vez el mensaje para comprobar el estado de Apache2
14     MENSAJE=`sudo systemctl status apache2 2>&1`
15     # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
16     curl -s -X POST $URL -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
17 done
```

Programa Status servidor (6)

- **Status servidor bucle activable:** modificamos el programa para que el que nos informe cada X segundos, se pueda activar o desactivar mediante Telegram.
- Para leer mensajes, Api Web de Telegram “getUpdates”
- Manejaremos el Bot escribiéndole por Telegram mensajes /empezar y /parar

Programa Status servidor (7)

- Servidor Status – Tercera versión (36 líneas)
- Parte 1 – Código inicial

```
1  #!/bin/bash
2
3  TOKEN="Escriba aqui su token" # Rellenar con el Token
4  ID_RECEPTOR="1552233" #Del estilo 1552233
5  URL_SEND="https://api.telegram.org/bot$TOKEN/sendMessage" #URL API Web para enviar Mensaje
6  URL_UPDATE="https://api.telegram.org/bot$TOKEN/getUpdates" #URL API Web para recibir Mensajes
7
8  #Inicialmente, estado desactivado, no envia notificaciones
9  estado="PARADO"
10
11 #Bucle infinito
12 while :
```

Programa Status servidor (8)

- Parte 2 – Leyendo mensajes y activando/desactivando

```
14      #Coge los updates y los procesa para obtener la ultima orden recibida por el chat
15      orden=$(curl -s -X POST $URL_UPDATE -d chat_id=$ID_RECEPTOR | sed -E 's/.*"text":'
16
17      #Si la ultima orden recibida es /empezar o /parar, cambia el estado
18      if [ $orden = "/parar" ];
19      then
20          estado="PARADO"
21      elif [ $orden = "/empezar" ];
22      then
23          estado="FUNCIONANDO"
24      fi
```

Programa Status servidor (9)

- Parte 3 – Si esta activado, mandamos mensaje

```
26      #Enviamos el mensaje si el estado es funcionando
27      if [ $estado = "FUNCIONANDO" ];
28      then
29          #Mensaje para obtener estado Apache2
30          MENSAJE=`sudo systemctl status apache2 2>&1`
31          #Enviamos el mensaje
32          curl -s -X POST $URL_SEND -d chat_id=$ID_RECEPTOR -d text="$MENSAJE"
33      fi
34      #Duerme durante 3 segundos
35      sleep 3
```

¿Otros lenguajes de programación?

- Estos ejemplos pueden realizarse en casi cualquier lenguaje de programación: Java, NodeJS, PHP
- Los ejemplos presentados estan disponibles tanto en ShellScript como en Python en:
 - <https://github.com/sergarb1/TelegramBots>

Ejemplo con Python

- Servidor Status Python– Tercera versión (43 líneas)



```
1 #!/usr/bin/python3
2 import subprocess
3 import time
4
5 TOKEN="Escriba aqui su token" # Rellenar con el Token
6 ID_RECEPTOR="1552233" #Del estilo 1552233
7 #Obtenemos el status del servidor Apache2
8 comandoStatus="sudo systemctl status apache2 2>&1"
9
10 URL_SEND="https://api.telegram.org/bot"+TOKEN+"/sendMessage" #URL API Web para enviar
11 URL_UPDATE="https://api.telegram.org/bot"+TOKEN+"/getUpdates" #URL API Web para leer
12
13 #Estado de los envios. Por defecto no se envia nada.
14 estado="PARADO"
15
16 while(True):
17
18     #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
19     p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
20     (output, err) = p.communicate()
21     ordenEstado=output.decode("utf-8") #Paso de Bytes a String UTF8
22
23     #Si la ultima orden recibida es /empezar o /parar, cambia el estado
24     if ordenEstado==" /empezar":
25         estado="FUNCIONANDO"
26     elif ordenEstado==" /parar":
27         estado="PARADO"
28
29     #Enviamos el mensaje si el estado es funcionando
30     if estado=="FUNCIONANDO":
31         #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
32         p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
33         (output, err) = p.communicate()
34         MENSAJE=output.decode("utf-8") #Paso de Bytes a String UTF8
35
36         # Envia por Post a la URL dada, al chat con el usuario ID el mensaje
37         comando="curl -s -X POST "+URL_SEND+" -d chat_id="+ID_RECEPTOR+" -d text='"+str(MENSAJE)+"'"
38         #Lanzo el comando y btengo salida normal y salida de error del comando ejecutado
39         p = subprocess.Popen(comando, stdout=subprocess.PIPE, shell=True)
40         (output, err) = p.communicate()
41
42     #Paramos 10 segundos
43     time.sleep(10)
```


Bibliotecas útiles para Bots Telegram

- Los ejemplos aquí vistos han programado el Bot usando la Web Api directamente.
- Colección de bibliotecas para facilitar la tarea
 - <https://core.telegram.org/bots/samples>
- Bibliotecas destacadas
 - **Python** <https://github.com/python-telegram-bot/python-telegram-bot>
 - **NodeJS** <https://github.com/yagop/node-telegram-bot-api>
 - **PHP** <https://github.com/tg-bot-api/bot-api-base>
 - **Java** <https://github.com/rubenlagus/TelegramBots>