

## **Zabbix + Telegram Entegrasyonu**

Bu proje, Zabbix sistem izleme aracının Telegram bot ile entegre edilerek olay (trigger) bazlı anlık bildirim gönderebilmesini sağlamaktadır. Amaç, sistem yöneticisinin kritik olayları doğrudan mesaj yoluyla öğrenmesini sağlayarak müdahale süresini kısaltmak ve sistem güvenliğini artırmaktır.

### **Kullanılan Teknolojiler**

- Zabbix Server 6.0+
- Ubuntu 22.04
- MySQL
- Apache + PHP (Zabbix frontend)
- Zabbix Agent
- Telegram Bot API
- Python 3 (requests modülü)

### **Kurulum Adımları**

#### **Zabbix Paket Kurulumu**

```
wget https://repo.zabbix.com/zabbix/6.0/ubuntu/pool/main/z/zabbix-release/zabbix-release_6.0-1+ubuntu22.04_all.deb
```

```
sudo dpkg -i zabbix-release_6.0-1+ubuntu22.04_all.deb
```

```
sudo apt update
```

```
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent mysql-server
```

#### **MySQL Veritabanı Kurulumu**

```
sudo mysql
```

```
create database zabbix character set utf8mb4 collate utf8mb4_bin;
```

```
create user zabbix@localhost identified by 'zabbixpass';
```

```
grant all privileges on zabbix.* to zabbix@localhost;
```

```
quit;
```

#### **Zabbix Veritabanı Şemasını Yükleme**

```
zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql -uzabbix -p zabbix
```

#### **Zabbix Server Yapılandırması**

```
sudo nano /etc/zabbix/zabbix_server.conf
```

```
DBPassword=zabbixpass
```

### **Servisleri Başlatma**

```
sudo systemctl restart zabbix-server zabbix-agent apache2
```

```
sudo systemctl enable zabbix-server zabbix-agent apache2
```

### **Web Arayüz Kurulumu**

http://localhost/zabbix adresine gidilerek sihirbaz tamamlanır.

Varsayılan kullanıcı: Admin

Varsayılan şifre: zabbix

### **Telegram Bot Kurulumu**

- @BotFather üzerinden yeni bot oluşturuldu ve bot token alındı.
- @userinfobot ile kullanıcıya ait chat\_id öğrenildi.

### **Python Script**

/scripts/telegram.py dosyası içeriği:

```
#!/usr/bin/env python3
```

```
import requests
```

```
import sys
```

```
TOKEN = "BOT_TOKEN"
```

```
CHAT_ID = sys.argv[1]
```

```
SUBJECT = sys.argv[2]
```

```
MESSAGE = sys.argv[3]
```

```
url = f"https://api.telegram.org/bot{TOKEN}/sendMessage"
```

```
data = {"chat_id": CHAT_ID, "text": f"{SUBJECT}\n{MESSAGE}"}
```

```
requests.post(url, data=data)
```

### **Test Komutu**

```
python3 scripts/telegram.py 123456789 "Test" "Zabbix Telegram testi başarılı"
```

### **Zabbix Arayüz Ayarları**

## Media Type Tanımı

- Name: TelegramScript
- Type: Script
- Script name: telegram.py
- Parameters:
  - {ALERT.SENDTO}
  - {ALERT.SUBJECT}
  - {ALERT.MESSAGE}

## Kullanıcıya Medya Ekleme

- Send to: chat\_id
- Active time: 1-7,00:00-24:00
- Severity: All

## Action Tanımı

- Name: TelegramScript Test Action
- Event source: Triggers
- Send to: Admin
- Send only to: TelegramScript
- Custom message:
  - Subject: {TRIGGER.NAME}
  - Message: {TRIGGER.STATUS}: {TRIGGER.NAME} on {HOST.NAME}

## Test Trigger

- Item key: system.cpu.load[percpu,avg1]
- Update interval: 30s
- Trigger expression:

{Zabbix server:system.cpu.load[percpu,avg1].last()}>0.001

## Sorun Giderme

- chat\_id is empty → Parametre sırası düzenlendi
- import not found → Script başına #!/usr/bin/env python3 eklendi

- git push hatası → Personal Access Token (PAT) ile çözüldü

## Sonuç

Zabbix sisteminde oluşan olaylar başarıyla Telegram'a aktarılmıştır. Telegram bot entegrasyonu ile kullanıcıya anında mesaj gönderimi sağlanarak sistem izleme süreci daha güvenilir ve hızlı hale getirilmiştir.

## Dosya Yapısı

zabbix-telegram-integration/

|

| └─ README.md

└─ scripts/

└─ telegram.py