

Lehen eszenategia

Observium notifikazioak Telegram kanal batean jaso (Ordubete)

Eszenategi honetako helburua Observium-ek bidalitako alertak Telegram kanal batean jasotzea izango da. Horretarako lehenik eta behin Telegrameko bot-a sortu beharko dugu, ondoren alertak jasotzeko kanala eta azkenik Observium konfiguratu beharko dugu alertak kanal horretara bidaltzeko.

1- Sortu Telegram bot-a

@botfather-ekin elkarrizketa bat hasi behar da Telegrameko bot-a sortzeko. Bot hau sortzea beharrezkoa da Observium-en notifikazioak konfiguratu ahal izateko. Azken finean, Observium-ek bot horren izenean idatziko du kanalean. Behin robota sortutakoan gorde emandako tokena, aurrerago beharko dugu eta.

<https://core.telegram.org/bots#6-botfather>

2- Sortu notifikazioak jasotzeko taldea eta gehitu aurrez sortutako bot-a bertara

3- Lortu bot eta kanalaren identifikadoreak

Observium-eko notifikazioak konfiguratu ahal izateko kanalaren identifikadorea lortu behar dugu. Robotaren tokena lehen pausuan lortu dugu, kanalaren identifikadorea lortzeko aukera desberdinak dauzkagu:

1. @getidsbot robota gehitu gure kanalera. Behin gehituta, kanalaren datuak erakutsiko dizkigu JSON batean
2. Robota kanalean sartu bezain pronto sartu helbide honetan:
 - https://api.telegram.org/botBOT_TOKEN/getUpdates
 - JSON bat erantzungo du, bilatu chat->id atributua

4- Konfiguratu Observium

Observium-ek alerta mezuak bideratzeko aukera desberdinak eskaintzen ditu, eta kontaktu-en bidez kudeatzen ditu. Telegramera bidalketak egin ahal izateko Telegram motako kontaktu berri bat gehitu eta konfiguratu da.

5- Froga egin

Behin kontaktua konfiguratuta dagoenean "Alert Checker" bat gehitu behar da. Nahi dena monitorizatu daiteke, baina simulazioak egiteko errexena localhost-en disko erabilpena aztertzen duen checker bat gehitzea da. Alerta egoera sortu nahi baldin badugu check-aren balioa aldatuko dugu, adibidez alerta sortzeko disko erabilpena %10 baino handiagoa denean.

Checker-a sortu ondoren bueltatu sortu berri dugun kontaktuaren edizio orrira eta lotu ('Associate') alert checker-arekin.

Kontuan izan observium-ek bost bat minutuko check zikloa duela checkerrak exekutatzeke, hortaz baliteke pixka batean zain egon behar izatea alarma notifikazioa jasotzeko.

Bigarren eszenategia

Telegram bot baten garapena (2 ordu)

Badaukagu gure Telegram bota, baina momentu honetan Observium-etik mezuak bidaltzeko baino ez da erabiltzen. Robotek elkarreragiketa aukera gehiago eskaintzen dituzte, eta eszenategi honetan horietako batzuk ikusiko ditugu.

Botekin ekintza desberdinak burutzeko aukera ematen duen APIa eskaintzen du Telegramek (<https://core.telegram.org/bots/api>), baina zuzenean erabili beharrean Python eta python-telegram-bot liburutegia erabiliko ditugu (<https://python-telegram-bot.readthedocs.io>).

Python Telegram bot liburutegia Telegrameko bot apiaren python inplementazio bat da. APIaren inplementazio 'puruaz' (APIak eskaintzen duen eragiketa bakoitzaren baliokidea den python kodea) gain, bot garapena errazten duten goi mailako klaseak eskaintzen ditu. Azken hauek erabiliko ditugu gure robotaren eragiketa desberdinak inplementatzeko.

Lau eragiketa desberdin landuko ditugu:

1. Oinarrizko komando bat
2. Mezuei erantzutea
3. Parametroak jasotzen dituzten komandoak (adb. Ping bat egiten duen komandoa, python ping3 liburutegia erabili daiteke <https://pypi.org/project/ping3/>)
4. Aurredefinitutako teklatu bidezko erantzunak

Garapena errazteko kodearen oinarrizko txantiloia bat daukazue aukeran

<https://github.com/aitzol/telegram-bot-txantiloia>

Txantiloia hau erabiltzeko gomendatutako pausuak hauek dira:

1. Klonatu repoa
2. Sortu eta aktibatu virtualenv ingurunea
3. Instalatu dependentsiak
4. Ezarri .env fitxategian beharrezko informazioa
5. Kodetu eragiketa bakoitza src/bot.py fitxategian
6. Exekutatu robot python src/bot.py exekutatuaz

Burutu beharreko eragiketa bakoitza inplementatzeko, Python Telegram Botak klase lagungarriak eskaintzen dizkigu. Laguntza gisa hemen daukazue gutxienez erabili beharko dituzuen klaseak:

- Oinarrizko komando bat -> CommandHandler
- Mezuei erantzutea -> MessageHandler + Filter (TIP: bot batek mezuei erantzun ahal izateko administradorea izan behar da taldean)
- Parametrodun komandoak -> CommandHandler (TIP: context.arg barruan datoz parametroak)
- Teklatuak -> CallbackQueryHandler, InlineKeyboardButton, InlineKeyboardMarkup

Proposatutako erronka

- Oinarrizko komandoa -> Mezu sinple batekin erantzun komandoa jasotzean
- Mezuei erantzun -> Jasotako mezuak hitz gakoren bat duenean (adb. Errorrea) erantzun
- Parametrodun komandoa -> Ping bat egin parametro gisa jasotako ip helbidera
- Teklatuak -> “Mezuei erantzun” eragiketan garatutakoari teklatu bat gehitu informazio gehigarria eskatzeko (adb, zein motako errorrea izan den)

Hirugarren eszenategia

Ebentu kudeaketarako web API-a (2 ordu)

Orain artean beti beste plataforma batek (Telegram) ematen dituen aukera eta murriztapenen menpe egon gara. Eszenategi honetan ebentuen kudeaketa guztiz gure esku hartuko dugu.

Honek noski erantzukizun guztia (autentikazioak, baimenak, segurtasuna...) gure ardurapera pasatzen du.

Ebentuen kudeaketarako Flask frameworka (<https://flask.palletsprojects.com/en/2.1.x/>) erabiliko dugu, eta aurreko kasuan bezala oinarritzko kodea repositorio honetatik hartu dezakezue <https://github.com/aitzol/observium-api-txantiloia> (jarraitu gidalerro berak martxan jartzeko)

Gure web aplikazioa martxan jartzeko garapena Digital Oceaneko zerbitzarian egin behar da (bertan klonatu repositorioa) eta observium-eko kontaktu berri bat konfiguratu (Webhook JSON motakoa) zerbitzariaren IP:5000 portua adierazi notifikazioak bidaltzeko helbide gisa.

Proposatutako erronka

Observium-ek bidalitako alertak Telegrameko kanalera bidali baldin eta soilik baldin lan ordutegiaren barruan baldin bagaude (demagun astegunak 9:00 – 17:00) edo alertaren maila 'Critical' den.