



UNIVERSITÀ DI PISA

SNMPNotifierBot

Nicolò Maio, Matteo Biscosi

A.A. 2019-20

Indice

1	Introduzione	3
2	Architettura del sistema	3
2.1	Gestione delle richieste	3
2.2	Grafici	4
2.3	Modifica dell'agent da monitorare	4
3	Sicurezza	5
4	Istruzioni e prerequisiti	5
5	Esempi di esecuzione	6

1 Introduzione

Il progetto SNMPNotifierBot consiste nell'implementazione di un bot Telegram che permette di monitorare delle informazioni di un agent SNMP (quali: percentuale di utilizzo della CPU, della RAM, del Disco, bits al secondo in Download e in Upload) e di poterle salvare in file .rrd e ottenerne quindi dei grafici da essi.

2 Architettura del sistema

La gestione dell'interazione con telegram viene gestita da 'bot_main.py'. Esso sta in ascolto delle richieste da parte dell'utente e ogni qualvolta ottiene una nuova richiesta, fa il parsing di essa, usufruendo delle funzioni 'on_callback_query', che controlla le richieste dai bottoni, e 'on_chat_message', che controlla le richieste testuali, e una volta capita la richiesta, richiamerà gli appositi metodi che la gestiranno e restituiranno il risultato alla chat, identificata dal parametro 'chat_id'.

2.1 Gestione delle richieste

Le richieste vengono gestite all'interno di 'snmp_op.py'. É possibile richiedere le seguenti informazioni sull'agent SNMP:

- Percentuale attuale di utilizzo della CPU;
- Percentuale attuale di utilizzo della RAM;
- Percentuale attuale di utilizzo del Disco;
- Ram totale (in KB);
- Numero totale di tasks attualmente in esecuzione;
- Informazioni generiche dell'agent (quali Nome, Locazione, Contatti, Descrizione, OID e System up time).

Inoltre è possibile salvare le info riguardanti CPU, RAM, Disco, InOctets, OutOctets utilizzando il database di serie temporali RRDtool. Alla richiesta di 'START_MONITORING', il bot avvierà un thread demone. Esso controllerà se esistono già i files contenenti le informazioni riguardanti l'agent (in quel caso riprenderà il monitoraggio), altrimenti richiederà all'utente per quanto tempo si vogliono mantenere i dati e dopodiché creerà i files e cartelle necessarie all'interno dei quali inserirà queste informazioni ogni 10 secondi (la scelta di tale frequenza di aggiornamenti è dovuta al fatto che SNMP, per alcune di queste info effettua caching e quindi non avremmo sempre dati consistenti con un tempo di richieste minore di 10 secondi).

In caso si voglia stoppare il monitoraggio dell'agent, una volta avviato, basta premere il bottone, nella home del bot, 'STOP MONITORING' ed in caso si

voglia rimuovere le informazioni ottenute fino ad ora sull'agent, basta scrivere in chat `/remove` (operazione disponibile solo se l'host al momento non è monitorato).

Altra funzione che rende disponibile il bot, solo se l'agent SNMP è localhost, è la classifica dei processi che al momento utilizzano più memoria sulla macchina (contenente nome, pid e memoria utilizzata dal processo in GB).

2.2 Grafici

Per quanto riguarda la creazione dei grafici, essa verrà gestita da `'graph_maker.py'`. Verranno creati grafici delle ultime 24 ore, contenenti, oltre che all'info richiesta, il 95esimo percentile, il lower e l'upper bound, il valore minimo, il valore massimo, la media dei valori (RAM, CPU, Disco sono misurati in percentuale, Download (InOctets) e Upload (OutOctets) in bits per seconds) e i fallimenti (viene generato un fallimento, con una colonna gialla all'interno del grafico, ogni qualvolta il valore registrato è per 3 volte consecutive al di fuori del lower ed upper bound).

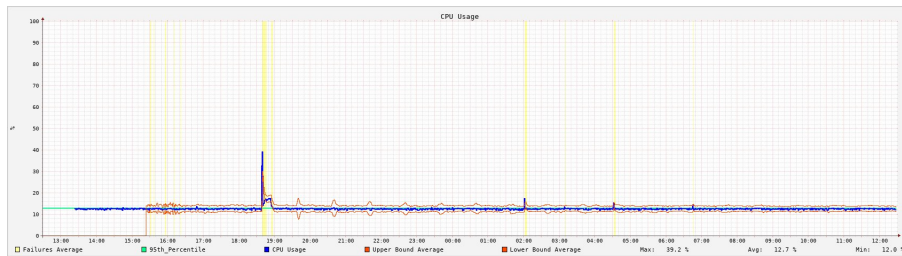


Figure 1: Esemplio grafico CPU



Figure 2: Esemplio grafico Download

2.3 Modifica dell'agent da monitorare

È inoltre possibile cambiare l'agent da monitorare, modificando hostname, community e versione (supportata solamente la n. 1 e 2c). I comandi per modificare

l'agent sono visibili sotto 'settings' nella home (Telegram) del bot. Non è possibile tuttavia modificare queste info se attualmente è in corso il monitoraggio di un agent, per cambiarle è necessario prima stoppare il monitoraggio.

3 Sicurezza

Per quanto riguarda la sicurezza dell'accesso alle info rese disponibili dal bot, è stato inserito un sistema di login ed ogni qualvolta verrà richiamato il bot verrà controllato se la chat ha effettuato o meno il login (controllo effettuato usufruendo di un metodo ed una classe situati in '/package_login/login.py'). In caso affermativo, gli verranno rese disponibili le varie funzionalità, altrimenti verrà restituito un messaggio apposito che richiederà di effettuare il login. È inoltre possibile cambiare la password tramite l'esecuzione dello script python './change_password.py old_password new_password', tuttavia per poter effettivamente vedere modificata la password è necessario riavviare il bot. Le password e gli id sono criptati, all'interno dei file json, utilizzando l'algoritmo 'SHA-256'.

4 Istruzioni e prerequisiti

Come indicato nel 'README.md' al fine di poter eseguire correttamente il bot sono necessari i seguenti moduli (ottenibili in Linux con il comando indicato dopo il nome del modulo):

- python3:

```
sudo apt-get install python3
```

- telepot:

```
pip3 install telepot
```

- emoji:

```
pip3 install emoji
```

- easysnmp:

```
pip3 install easysnmp
```

- psutil:

```
pip3 install psutil
```

- rrdtool:

```
pip3 install rrdtool
```

Dopodiché:

- Su Telegram: bisogna iniziare una conversazione con '@BotFather' per creare un nuovo bot. Va salvato il token sul pc.
- Sul Server: va scaricata la repository di github, bisogna spostarsi all'interno di '/bot' ed inserire il token, datoci da '@BotFather', all'interno del file 'bot_main.py' tra le virgolette alla linea 30 (al posto di token='11111111', diventerà token='new_token'). Dopodiché se si desidera cambiare la password di default (gestionedireti) dovrà essere eseguito lo script './change_password.py old_password new_password'. A questo punto basterà eseguire lo script './bot_main.py' e sarà possibile iniziare una conversazione col bot o aggiungerlo ad un gruppo su Telegram.

5 Esempi di esecuzione

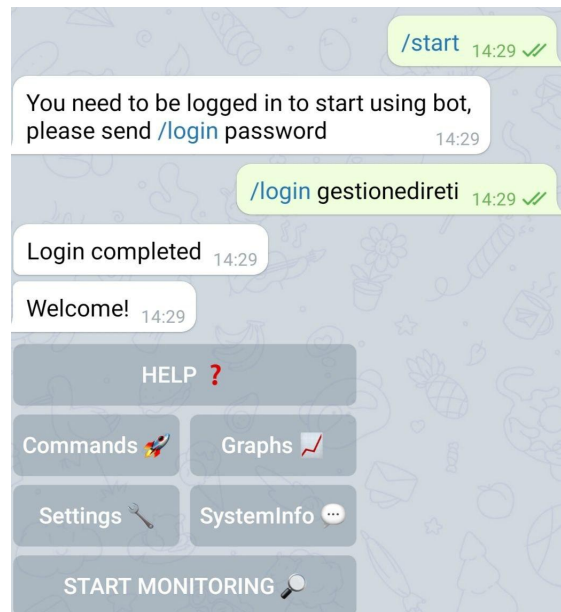


Figure 3: Esempio di login

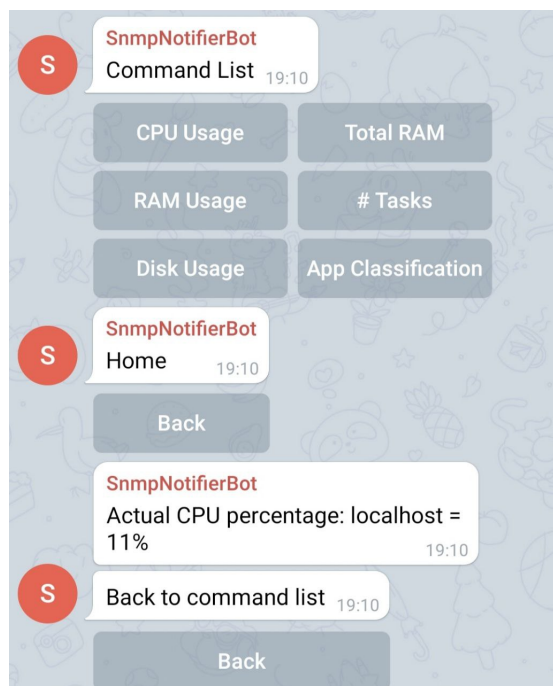


Figure 4: Esempio di info attuali riguardo l'host (Commands -> CPU Usage)

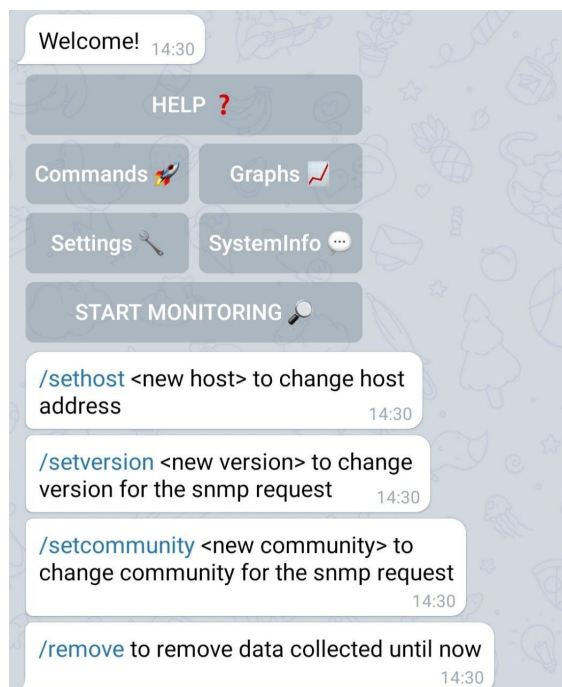


Figure 5: Esempio di impostazioni (Settings)

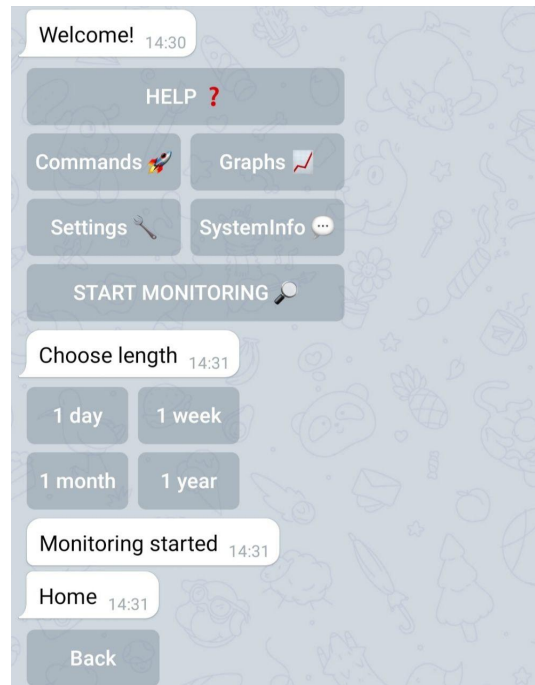


Figure 6: Esempio di avvio monitoraggio con mantenimento dei dati di 1 giorno (START MONITORING -> 1 day)

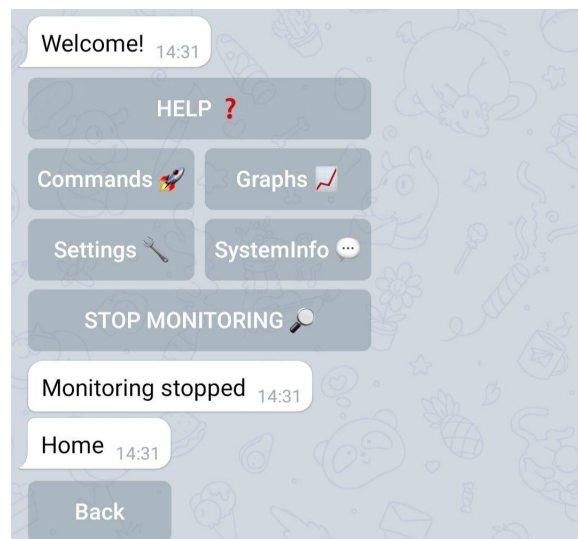


Figure 7: Esempio di stop monitoraggio

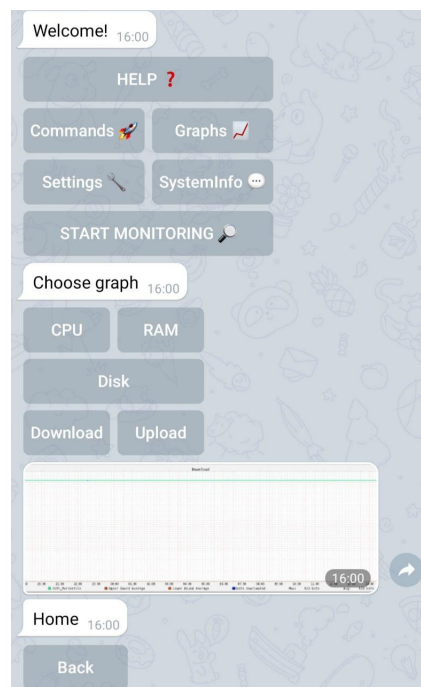


Figure 8: Esempio di grafico del disco (Graphs -> Disk)