



UNIVERSITÀ DI PISA

General Agent SNMP

Vigiani Igor

A.A. 2019-20

Indice

1	Introduzione	2
2	Requisiti	2
3	Architettura del sistema	2
3.1	Agent	2
3.2	ProcessData	3
3.3	DataBase	3
3.4	Interazione Moduli	4
4	Eseguire il Programma	5
5	Esempio	5
5.1	Host-Resource-MIB	5
5.2	InfluxDB	6
5.3	Esecuzione	10

1. Introduzione

Il progetto consiste nell'implementazione di un software che reperisce i dati a intervalli regolari da un Agent Snmp per poi processarli e salvarli in un database.

2. Requisiti

Per eseguire il software è necessario avere un agent snmp attivo nella macchina che si vuole monitorare.

Per avere un agent snmp in windows si veda la seguente guida: <https://www.raffaelechiatto.com/installazione-e-configurazione-del-servizio-snmpp-in-windows-10-version-1809-e-versioni-successive/>

Mentre per linux si veda: <http://www.net-snmp.org/>

Il progetto è stato implementato utilizzando Python3 con l'utilizzo della libreria EasySnmp.

Per maggiori informazioni: <https://easysnmp.readthedocs.io/en/latest/>

Per salvare e visualizzare i dati si è scelto l'utilizzo di influxdb.

Per maggiori informazioni: <https://www.influxdata.com/products/influxdb-overview/influxdb-2-0/>

3. Architettura del sistema

Il progetto è suddiviso in tre moduli principali:

- Agent
- ProcessData
- DataBase

3.1 Agent

Il modulo Agent reperisce ad intervalli regolari informazioni dall'agent snmp installato nella macchina e li memorizza in struttura dati la cui chiave è il nome del OID e il valore è il valore ottenuto. Al momento non è implementato il comando walk quindi se si necessita di ciò è necessario specificare ogni OID nel file AgentSnmp.json.

Il file AgentSnmp.json ha la seguente struttura:

```
{
```

```

    "version":      String  // al momento solo versione 1 e 2c
    "community":    String
    "indirizzo ip": String
    "port":         int
    "OIDS":         [String]
    "intervall":    int     // espresso in secondi
}

```

3.2 ProcessData

Il modulo ProcessData elabora i dati ottenuti dal modulo Agent restituendo i dati da memorizzare. Di default i dati ottenuti dal modulo Agent non vengono elaborati ma se necessario è sufficiente definire un modulo che implementi il metodo **process**.

Il metodo **process** deve restituire una struttura dati la cui chiave è il nome del dato e il cui valore è il dato calcolato.

Il modulo che implementa il metodo **process** deve essere nella stessa directory di ./agent.py

3.3 DataBase

Il modulo DataBase salva i dati elaborati dal modulo ProcessData. Di default i dati sono salvati in formato csv ma se necessario è sufficiente definire un modulo che implementi i metodi: **create**, **close** e **update**.

Il file csv è formato dalla prima colonna il nome degli OIDS e successivamente ogni riga è composta dal numero di riga + i valori.

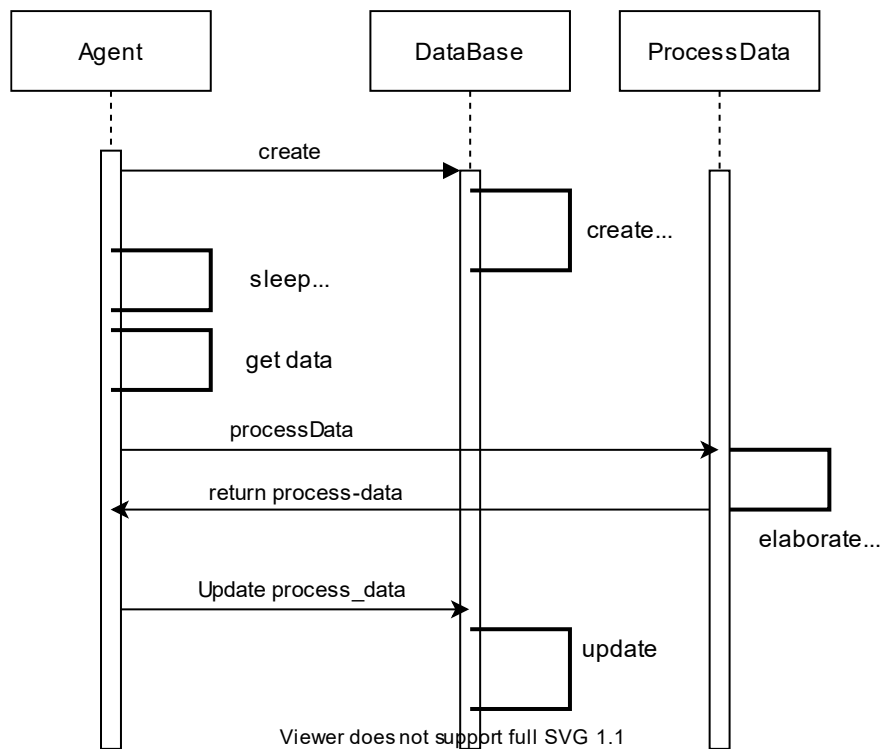
I caratteri per separare i valori sono:

- Punto e virgola(;) per i valori singoli
- Virgola(,) per le liste di valori

Il modulo che implementa i metodi: **create**, **close** e **update** deve essere nella stessa directory di ./agent.py

3.4 Interazione Moduli

Di seguito si è scelto l'utilizzo di un diagramma di flusso per mostrare come i vari moduli interagiscono tra di loro.



4. Eseguire il Programma

Per eseguire il programma è necessario il comando:

```
python3 agent module_ProcessData_name module_DataBase_name
```

Se il programma viene eseguito senza passare nessun argomento allora il suo

comportamento è quello di default(i dati non vengono elaborati e salvati in formato json) altrimenti i moduli DataBase e ProcessData si comporteranno come indicato dai moduli passati come argomenti.

Nel caso di errore nel caricamento del modulo il programma continua ma con il comportamento di default.

Se il caricamento del modulo avviene con successo ma non implementa le funzioni sopra indicate allora il programma termina.

[!] I moduli devono essere nella stessa directory del programma principale [!]

5. Esempio

In questo esempio utilizziamo il programma per analizzare i consumi della cpu e della ram durante l'esecuzione di un videogioco in ambiente windows. I dati verranno memorizzati nel database temporale influxdb.

5.1 Host-Resource-MIB

Per ottenere le informazioni a noi di interesse abbiamo utilizzato:

- HOST-RESOURCES-MIB::hrProcessorLoad per analizzare i consumi della cpu
- HOST-RESOURCES-MIB::hrStorage per analizzare i consumi della ram

Nel mib **Host-Resource-MIB** non sono presenti queste informazioni direttamente per questo motivo è necessario calcolarseli. Questo comporta che è necessario modificare il comportamento del modulo ProcessData. Il modulo che calcola queste informazioni è MyProcessData.

Per l'installazione di un agent snmp si veda la sezione 2.

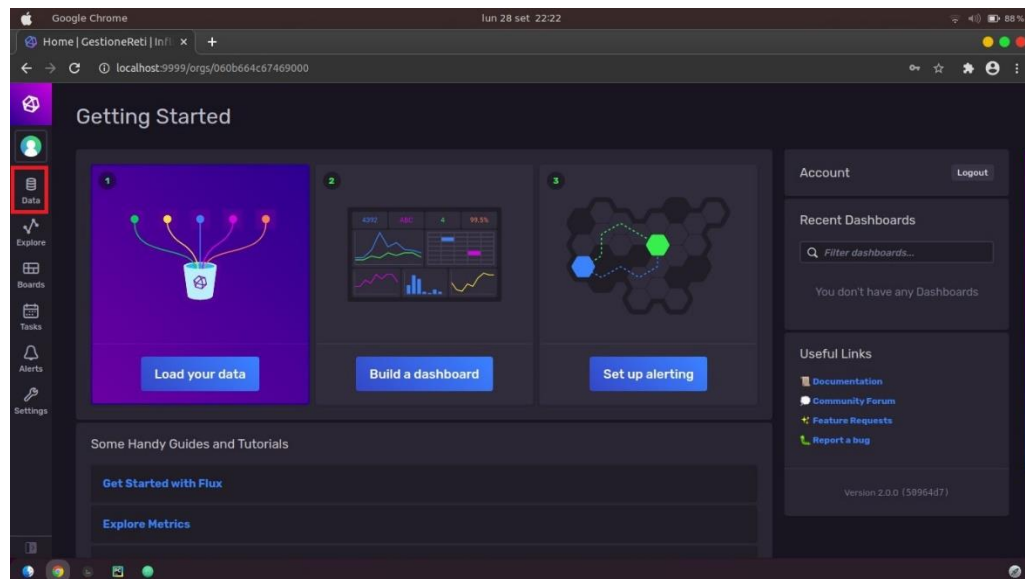
5.2 InfluxDB

Per salvare i dati si è optato l'utilizzo di influxdb quindi è necessario modificare il comportamento del modulo DataBase. Il modulo che implementa il database è InfluxDBMenager.

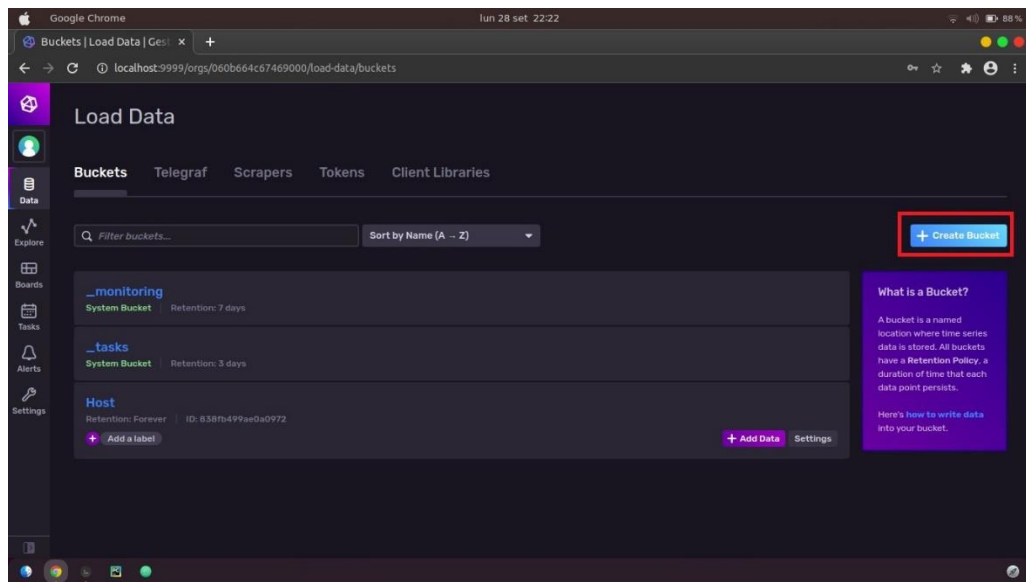
Per l'installazione di influxdb2.0 si veda la sezione 2.

Di seguito verrà illustrato come creare un database in influxdb e renderlo accessibile a terzi.

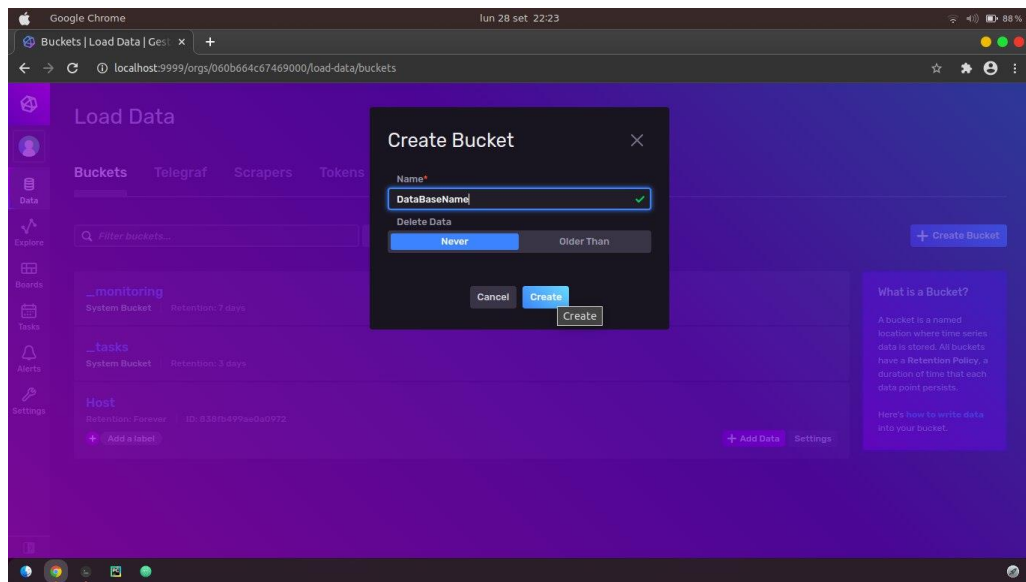
Dopo essersi registrati o collegati selezionare l'icona con il database



Cliccare in create Bucket

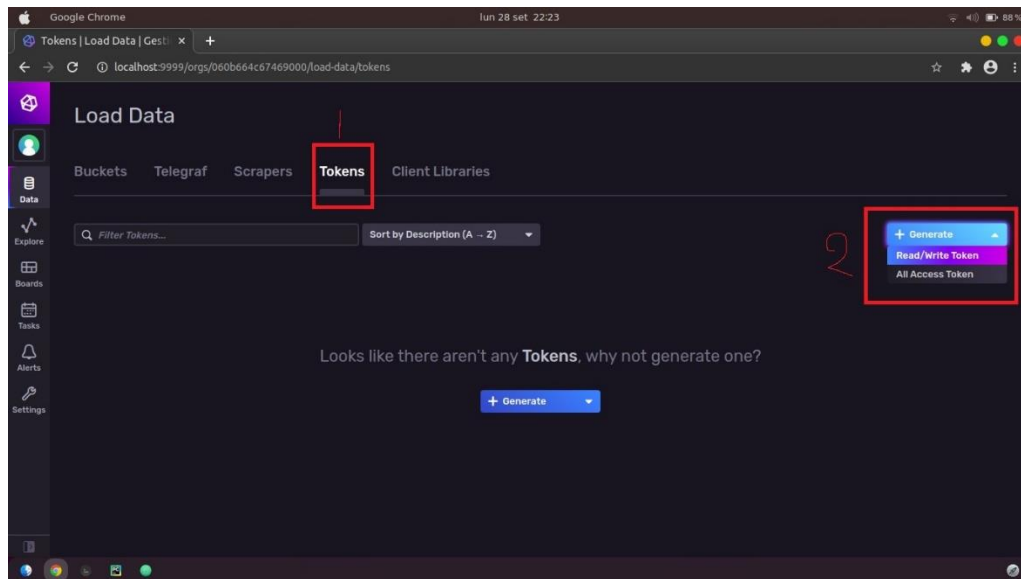


Inserire il nome del Bucket e cliccare su create.

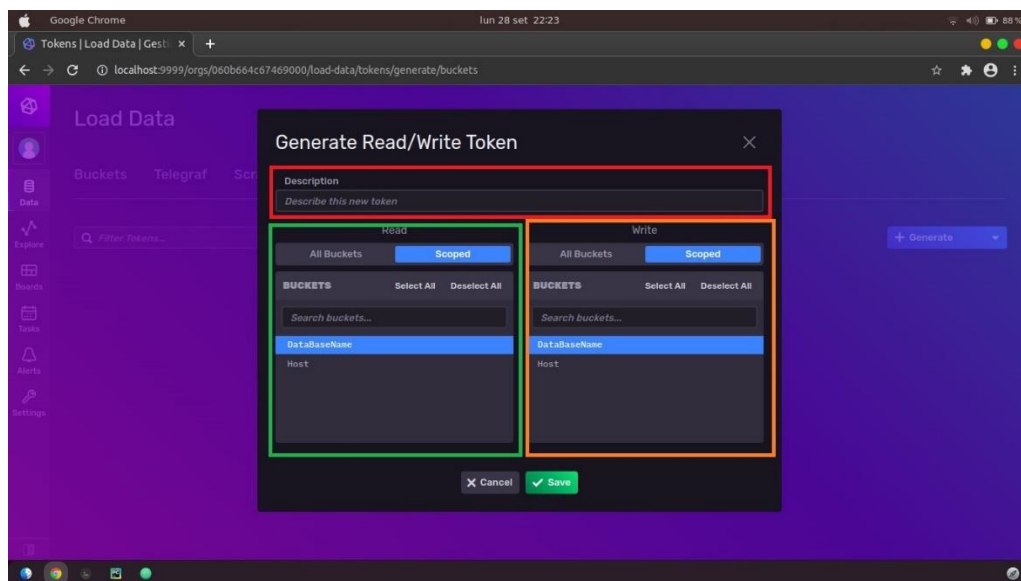


A questo punto abbiamo creato il Bucket (database) in cui verranno salvati i dati. Per rendere accessibile il Bucket a terzi è necessario fornirne un Token, vediamo come generarlo.

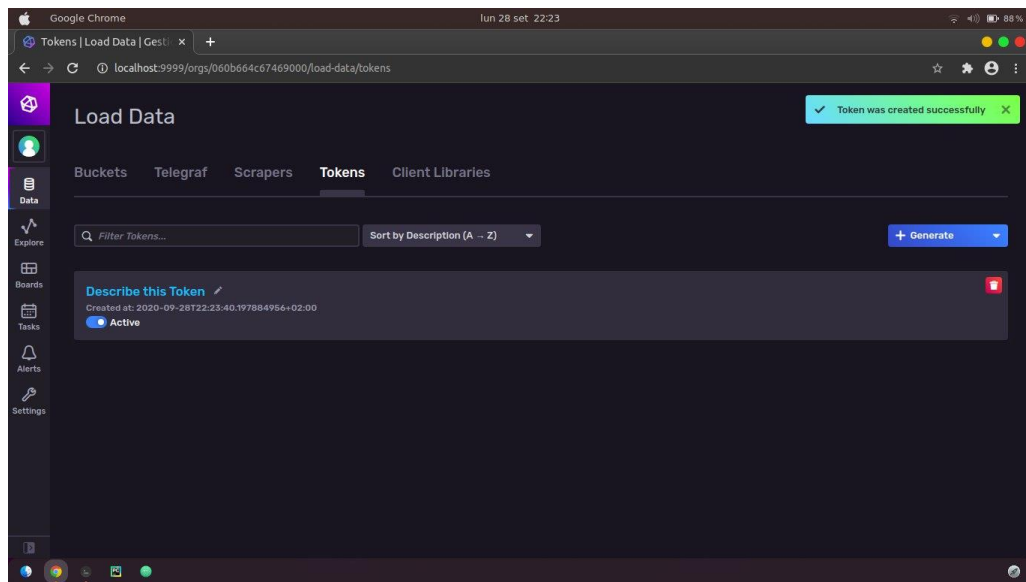
Si selezioni la voce Tokens e poi Generate.



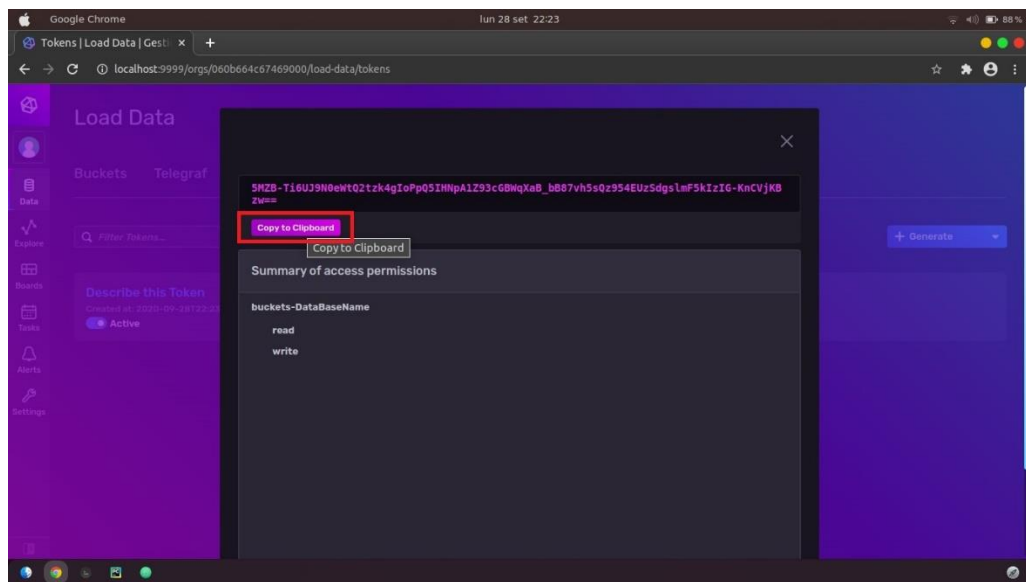
Nella sezione rossa si può aggiungere una breve descrizione. Nella sezione verde si fornisce i diritti di lettura mentre nella sezione arancione si fornisce i diritti di scrittura al Bucket selezionato.



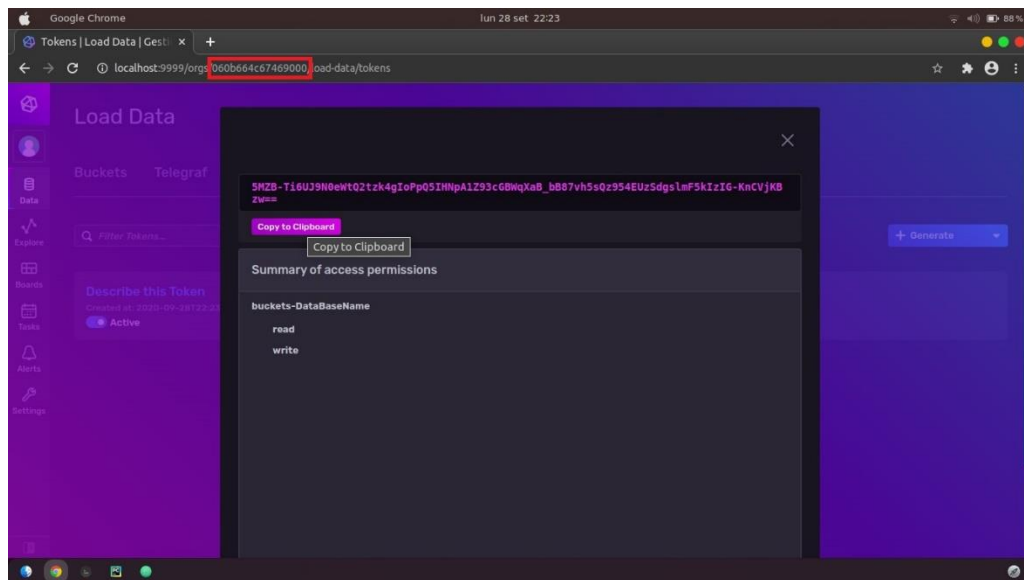
Se è andato tutto a buon fine dovresti vedere la seguente schermata con il token appena generato.



Selezionando il Token apparirà la seguente schermata. Premendo nel tasto evidenziato in rosso potrai copiare il Token necessario per accedere al Bucket.



Oltre al Token è necessario anche l'org che è disponibile nell'url. Si è evidenziato in rosso.



Copiato sia il Token che org è sufficiente inserire i valori nelle variabili token e org presenti nel modulo InfluxDBMenager.

5.3 Esecuzione

Per eseguire il programma è necessario passare come argomenti il nome dei moduli MyProcessData e InfluxDBMenager nel seguente modo:

```
python3 agent MyProcessData. InfluxDBMenager
```

Di seguito possiamo vedere i grafici ottenuti analizzando i dati con influxdb:



Il grafico è coerente con l'esecuzione del videogioco in quanto nei momenti di pausa l'andamento del grafo tende a decrescere.

Di seguito possiamo vedere il consumo di memoria ram

