

Esempio di log

2018-01-26 00:00:02.102 ERROR cc.acme.net.TCP.ObjectTCPWorker - Error in TCP connection handler creation [Socket[addr=/10.18.122.24,port=57182,localport=4000]]

dove

2018-01-26 00:00:02.102 **timestamp**

ERROR **tipo**

cc.acme.net.TCP.ObjectTCPWorker **oggetto che genera il log**

Error in TCP connection handler creation

[Socket[addr=/10.18.122.24,port=57182,localport=4000]] **messaggio**

Primo livello (Tra () le occorrenze)

ERROR (10367)

WARN (20948)

Secondo livello:

cc.acme.dns.netProxy.listeners.TcpListener (5183)

cc.acme.sec.SSL.ObjectSSLServer (5183)

cc.acme.net.TCP.ObjectTCPWorker (20734)

Terzo livello:

Error receiving object from (WARN)

Network proxy error on port (WARN)

Error in TCP connection handler creation (ERROR)

TCP connection error on port (WARN)

Quarto livello:

10.18.122.24 (2592)

10.254.114.1(7772)

10.15.193.168 (2615)

10.2.20.247 (2592)

10.18.122.23 (2592)

10.18.122.30 (2592)

Home Work 1

Ogni volta che c'è un entry WARN,
l'intera riga di log viene pubblicata nella coda.
Il MSGHandler salva il log sul DB sul db solo se l'oggetto è

- Error receiving object from
- Network proxy error on port .

Sul DB viene salvato:
Timestamp – IdMacchina - Messaggio

Home Work 2

Ogni volta che c'è un entry WARN,
l'intera riga di log viene pubblicata nella coda.
Il MSGHandler salva il log sul DB sul db solo se l'oggetto è "Error receiving object from
" e l'IP è 10.18.122.24 o 10.18.122.30

Sul DB viene salvato:
Timestamp – IdMacchina - Messaggio

Home Work 3

Ogni volta che c'è un entry contenente un IP differente dalla classe 10.18.122.X,
l'intera riga di log viene pubblicata nella relativa coda con TOPIC
10.18.122.23 o
10.18.122.24 o
10.18.122.30.

Il relativo MSGHandler salva sul DB:
Timestamp – IdMacchina – Messaggio

Home Work 4

Ogni volta che c'è un entry contenente un IP di classe 10.18.122.X,
l'intera riga di log viene pubblicata nella relativa coda con TOPIC
10.254.114.1 o
10.15.193.168 o
10.2.20.247

Il relativo MSGHandler salva sul DB:
Timestamp – IdMacchina - Messaggio

Home Work 5

Ogni volta che c'è un entry WARN,

l'intera riga di log viene pubblicata nella coda.

Un MSGHandler (topic based) salva ogni messaggio ricevuto sul db.

Un MSGHandler (content based) salva sul DB solo i log il cui oggetto è

- Error receiving object from

Sul DB viene salvato:

Timestamp – IdMacchina - Messaggio

Home Work 13

Ogni volta che c'è un entry WARN,

l'intera riga di log viene pubblicata nella coda.

Il MSGHandler salva il log sul DB solo se l'oggetto è "Error receiving object from" e l'IP è 10.18.122.24 o 10.18.122.30

Sul DB viene salvato:

Timestamp – IdMacchina - Messaggio

Home work 8

Scrivere uno script che esegua un pidstat (pidstat -urd -h -p PID-processo) su un processo in esecuzione.

Esempio:

#	Time	UID	PID	%usr	%system	%guest	%CPU	CPU	minflt/s	majflt/s	VSZ	RSS	%MEM	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
1516977470	1000	1491	102.86	0.27	0.00	103.13	0	13.53	0.00	3316564	362952	17.80	0.02	0.74	0.00	java	

Ogni volta che %CPU è maggiore del 1%, manda l'intera sequenza di valori nella relativa coda con TOPIC "Performance"

Tutti gli Handler (sia topic-based che content-based) ricevono la sequenza.

Il primo MSG Handler (topic-based puro), prende le info e le salva sul db.

Il secondo MSG Handler (content-based), salva sul db solo se %MEM > 10

Home work 11

Scrivere uno script che esegua 2 pidstat (pidstat -urd -h -p PID-processo) su 2 processi in esecuzione.

Esempio:

#	Time	UID	PID	%usr	%system	%guest	%CPU	CPU	minflt/s	majflt/s	VSZ	RSS	%MEM	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
1516977470	1000	1491	102.86	0.27	0.00	103.13	0	13.53	0.00	3316564	362952	17.80	0.02	0.74	0.00	java	

Per il primo, ogni volta che %CPU è maggiore del 1%, manda l'intera sequenza di valori nella relativa coda con TOPIC "Performance".

Per il secondo, ogni volta che %MEM è maggiore del 1%, manda l'intera sequenza di valori nella relativa coda con TOPIC "Performance".

Ogni MSG Handler che riceve il messaggio, salva sul db solo se %CPU > 10% && %MEM > 10

Home work 12

Scrivere uno script che esegua un pidstad (pidstat -urd -h -p PID-processo) su un processo in esecuzione.
Esempio:

#	Time	UID	PID	%usr	%system	%guest	%CPU	CPU	minflt/s	majflt/s	VSZ	RSS	%MEM	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
1516977470	1000	1491	102.86	0.27	0.00	103.13	0	13.53	0.00	3316564	362952	17.80	0.02	0.74	0.00	java	

Ogni vota che %MEM è maggiore del 1%, manda l'intera sequenza di valori nella relativa coda con TOPIC "Performance"

Tutti gli Handler (sia topic-based che content-based) ricevono la sequenza.

Il primo MSG Handler (topic-based puro), prende le info e le salva sul db.

Il secondo MSG Handler (content-based), salva sul db solo se %CPU > 1

Home work 6

Scrivere uno script che esegua 2 pidstad (pidstat -urd -h -p PID-processo) su 2 processi in esecuzione.
Esempio:

#	Time	UID	PID	%usr	%system	%guest	%CPU	CPU	minflt/s	majflt/s	VSZ	RSS	%MEM	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
1516977470	1000	1491	102.86	0.27	0.00	103.13	0	13.53	0.00	3316564	362952	17.80	0.02	0.74	0.00	java	

Sia il primo che il secondo, ogni vota che %CPU è maggiore del 5%, mandano l'intera sequenza di valori nella relativa coda con TOPIC "Performance".

Ogni MSG Handler che riceve il messaggio, salva sul db solo se %CPU > 10% && %MEM > 10

Home work Studente singolo

Scrivere uno script che esegua 1 pidstad (pidstat -urd -h -p PID-processo) su un processo in esecuzione.
Esempio:

#	Time	UID	PID	%usr	%system	%guest	%CPU	CPU	minflt/s	majflt/s	VSZ	RSS	%MEM	kB_rd/s	kB_wr/s	kB_ccwr/s	Command
1516977470	1000	1491	102.86	0.27	0.00	103.13	0	13.53	0.00	3316564	362952	17.80	0.02	0.74	0.00	java	

Ogni vota che %CPU è maggiore del 5%, lo scanner manda l'intera sequenza di valori nella relativa coda con TOPIC "Performance".

Ogni MSG Handler che riceve il messaggio, salva sul db solo se %CPU > 10% && %MEM > 10