



UNIVERSITÀ DI PISA

*Master Universitario di I Livello in Cybersecurity*



# CyberIntelligence: esercitazione su NiFi

Tiziano Fagni

IIT,CNR

[tiziano.fagni@iit.cnr.it](mailto:tiziano.fagni@iit.cnr.it)

Anno accademico 2022/2023

# Soluzione docker Nifi + ES

<https://github.com/tizfa/cyber-intelligence-2022>

Per l'installazione, seguite le istruzioni disponibili sul file README.md

La soluzione tramite Docker Desktop \*dovrebbe\* funzionare senza problemi su qualsiasi piattaforma supportata (Apple Intel, Apple Silicon, Windows, Linux)

# Prima di cominciare...

- La documentazione di NiFi è disponibile su <https://nifi.apache.org/docs.html>
- La documentazione di NiFi Expression Language è disponibile su <https://nifi.apache.org/docs/nifi-docs/html/expression-language-guide.html>
- Google è vostro amico!
- ....e ovviamente anche i vostri docenti sono sempre disponibili!

Dopo che il software Nifi è stato lanciato in esecuzione tramite docker potete accedere al pannello di controllo di Nifi andando su <https://localhost:8443/nifi/>

Username: **user**

Password: **cyberintelligence**

# Prima di cominciare (continua)...

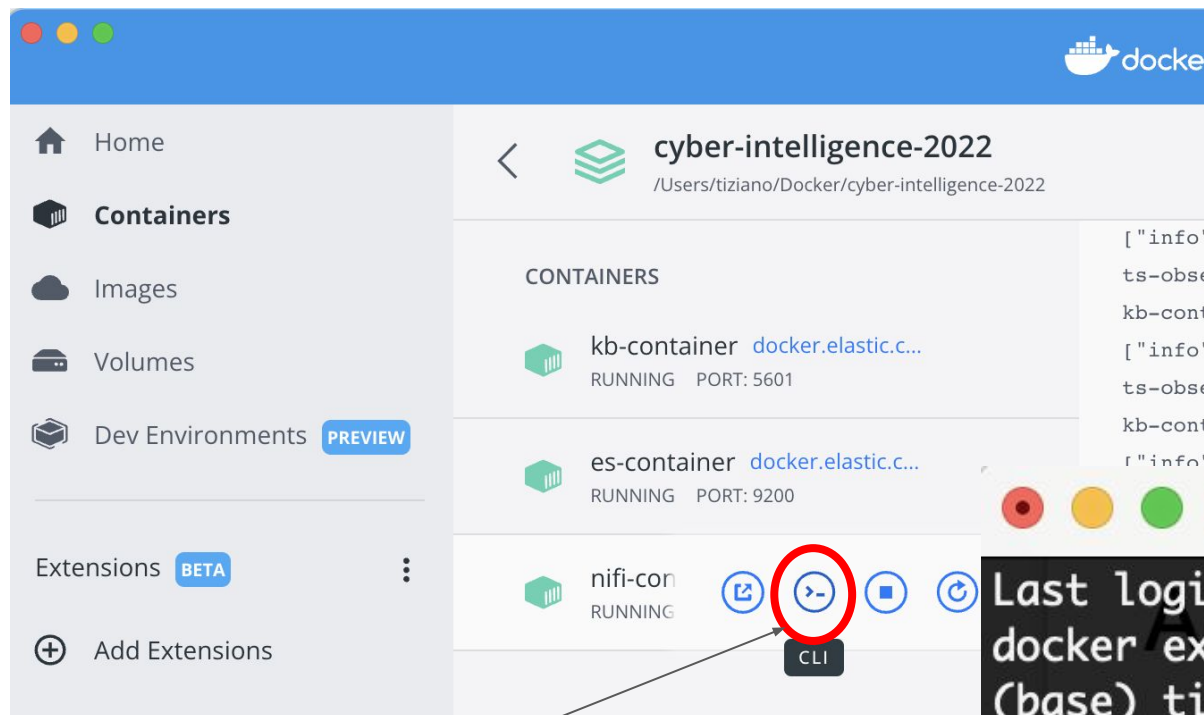
Soluzione docker composta da 3 container:

- **es-container**: VM per server Elasticsearch
- **kb-container**: VM per Kibana
- **nifi-container**: VM per Nifi

La cartella **cint** che si trova nella directory root della soluzione docker viene montata ed è condivisa con il path **/home/cint** del container “nifi-container”.

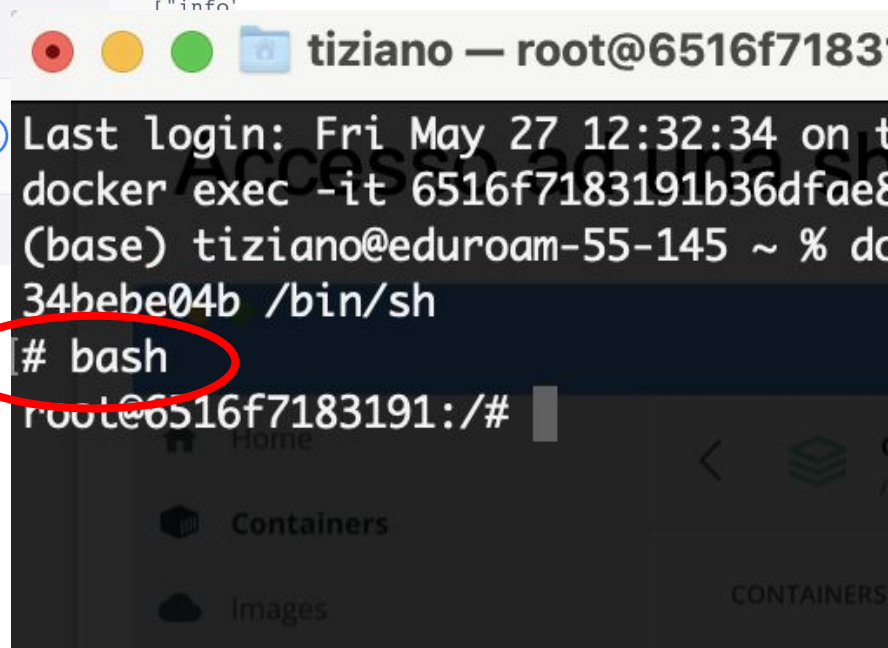
Dentro **cint** trovate le slide di documentazione (cartella *documentazione*), tutti gli script necessari al funzionamento degli esercizi proposti (cartella *esercitazione*) e l'output che viene generato da Nifi tutte le volte che eseguite un esercizio (cartella *test*).

# Accesso ad una shell di un container



Premere qui per aprire un terminale sul container desiderato


Eeguire il comando **bash** sulla shell




# Nifi: uso dei template dataflow

Su Nifi potete importare/esportare template di dataflow da/verso altre istanze di Nifi oppure per replicare un macro-componente all'interno della stessa istanza Nifi

## Creazione di un template dataflow

1. Selezionate il processore o il ProcessGroup di interesse
2. Premete il pulsante 
3. Impostate il nome del template e salvate
4. I template dataflow sono visibili e scaricabili da Menu (alto a destra) -> Templates

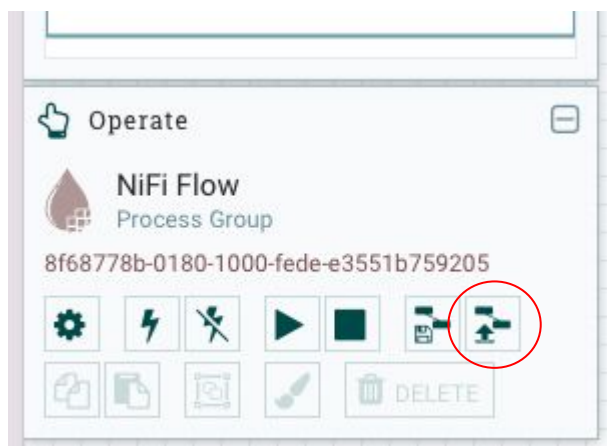
## Inserimento di un template dataflow

1. Trascinare  sull'area di lavoro del dataflow
2. Selezionare il template di interesse e confermare la scelta

# Importare dataflow su Nifi

Su Nifi potete importare/esportare dataflow da/verso altre istanze di Nifi

Per importare un dataflow esistente, usare



e selezionare il template XML da importare

Una volta che un template è stato importato, potete utilizzarlo nel vostro dataflow nello stesso modo un cui abbiamo visto prima (Inserimento di un template dataflow)

# Esercizio 1

In questo esercizio impareremo a usare e configurare i seguenti processori:

1. GetHTTP per accedere senza autenticazione ai dati forniti da un web service.
2. EvaluateJSONPath per catturare alcune informazioni dal JSON di input.
3. ReplaceText per generare un FlowFile con contenuto customizzato.

**Obiettivo:** Periodicamente ogni 10 secondi, ottenere e salvare su due cartelle distinte il prezzo attuale in dollari e euro delle criptovalute Bitcoin e Ethereum.

Per ottenere il prezzo corrente, si utilizzerà il Web service di CryptoCompare:

<https://www.cryptocompare.com/api/>



# Esempio di richiesta/risposta del Web service

## Esempio di richiesta per risolvere l'esercizio:

<https://min-api.cryptocompare.com/data/pricemultifull?fsyms=BTC,ETH&tsyms=USD,EUR>

## Estratto JSON di risposta:

```
"RAW": {  
  "BTC": {  
    "USD": {  
      "TYPE": "5",  
      "MARKET": "CCCAGG",  
      "FROMSYMBOL": "BTC",  
      "TOSYMBOL": "USD",  
      "FLAGS": "4",  
      "PRICE": 6758.39,  
      "LASTVOLUME": 1.82501163,  
      ...  
    },  
    "EUR": {  
      "TYPE": "5",  
      "MARKET": "CCCAGG",
```

# NiFi: il processore GetHTTP

**Scopo:** Permette di interrogare un server HTTP, utile per l'accesso a Web service di tipo REST.

Configure Processor

SETTINGS

SCHEDULING

PROPERTIES

COMMENTS

Required field

Property	Value
URL	<a href="https://min-api.cryptocompare.com/data/pricemultifull?fsyms=B">https://min-api.cryptocompare.com/data/pricemultifull?fsyms=B</a>
Filename	btce\${now():toNumber()}
SSL Context Service	StandardSSLContextService
Username	No value set
Password	No value set
Connection Timeout	30 sec
Data Timeout	30 sec
User Agent	No value set
Accept Content-Type	No value set
Follow Redirects	false
Redirect Cookie Policy	default
Proxy Host	No value set
Proxy Port	No value set

Importante configurare il contesto SSL per accesso a servizi in https

# NiFi: il processore GetHTTP, configurazione SSL

1. Creare un controller di tipo StandardSSLContextService.
2. Configurarli correttamente.
3. Abilitarlo.

GENERAL

CONTROLLER SERVICES

Name	Type	Bundle	State	Scope
JettyWebSocketServer	JettyWebSocketServer 1.5.0	org.apache.nifi - nifi-websocket-s...	Disabled	Cyberintelligence
JettyWebSocketServer	JettyWebSocketServer 1.5.0	org.apache.nifi - nifi-websocket-s...	Disabled	Cyberintelligence
StandardHttpContextMap	StandardHttpContextMap 1.5.0	org.apache.nifi - nifi-http-context-...	Enabled	Cyberintelligence
StandardHttpContextMap	StandardHttpContextMap 1.5.0	org.apache.nifi - nifi-http-context-...	Disabled	Cyberintelligence
StandardRestrictedSSLContextS...	StandardRestrictedSSLContextS...	org.apache.nifi - nifi-ssl-context-s...	Enabled	Cyberintelligence
StandardSSLContextService	StandardSSLContextService 1.5.0	org.apache.nifi - nifi-ssl-context-s...	Enabled	BitcoinPrice

Abilita o disabilita il controller

Per configurare correttamente il controller fare riferimento a questa guida:

<https://community.hortonworks.com/questions/9509/connecting-to-datasift-https-api-using-nifi.html>

Controller Service Details

SETTINGS

PROPERTIES

COMMENTS

Required field

Property	Value
Keystore Filename	No value set
Keystore Password	No value set
Key Password	No value set
Keystore Type	No value set
Truststore Filename	/Library/Java/JavaVirtualMachines/jdk1.8.0_121.jdk/Co...
Truststore Password	Sensitive value set
Truststore Type	JKS
TLS Protocol	TLS

# Configurazione contesto SSL

Configure Controller Service | StandardSSLContextService 1.16.0

SETTINGS
PROPERTIES
COMMENTS

Required field
☒
☐

Property	Value
Keystore Filename	No value set
Keystore Password	No value set
Key Password	No value set
Keystore Type	No value set
Truststore Filename	/usr/lib/jvm/default-java/lib/security/cacerts
Truststore Password	Sensitive value set
Truststore Type	JKS
TLS Protocol	TLS

CANCEL
APPLY

Usare questa configurazione quando usate la VM Linux. La password di default della JVM da usare nella proprietà Truststore Password è “changeit”.

# Esercizio 1: traccia su come realizzare il dataflow

1. Configurate il processore GetHttp per accedere al WebService tramite l'URL indicato.
2. Impostate il processore GetHttp affinché sia schedulato ogni 10 secondi.
3. Processate i JSON provenienti dal Web service tramite il processore EvaluateJSONPath ed inserire dei nuovi attributi con il prezzo di BTC e ETH.
4. Splittate il flusso in due sottoflussi, ognuno corrispondente a ciascuna criptovaluta.
  - a. Usate il processore ReplaceText per riscrivere i dati nel formato voluto, ad esempio CSV.
  - b. Scrivete i dati su una cartella di output tramite il processore PutFile.

## Esercizio 2

Problemi con soluzione Esercizio 1:

- Ogni flusso ha il suo processore PutFile
- Ogni file su disco contiene una sola misurazione.

**Obiettivo:** modificare il flusso di Esercizio 1 affinché da ciascun sottoflusso escano FlowFile contenenti 5 misurazioni ciascuno e ognuno di questo sia salvato su una opportuna cartella sulla base del valore dell'attributo "CryptoCurrency".

Nuovi processori:

- UpdateAttribute per aggiungere un nuovo attributo.
- MergeContent per unire i contenuti

## Esercizio 2: traccia su come realizzare il dataflow

1. Partire da una copia dell'Esercizio 1.
2. In ciascun sottoflusso, dopo avere generato il FlowFile con formato custom, andare a inserire mediante il processore UpdateAttribute un nuovo attributo "CryptoCurrency" contenente il valore "BTC" o "ETH".
3. Nello stesso sottoflusso, utilizzare il processore MergeContent per unire 5 FlowFile differenti in un nuovo FlowFile. Il processore MergeContent deve essere impostato a:
  - "Merge Strategy" = "Bin-Packing algorithm"
  - "Merge Format" = "Binary concatenation"
  - "Delimiter strategy" = "Text"
  - "Demarcator" = newline (premere Shift + Invio)
4. Utilizzare una unica istanza del processore PutFile per ricevere i FlowFile aggregati dai 2 sottoflussi e sfruttare l'attributo "CryptoCurrency" per scrivere i dati in cartelle diverse.

[Qui](#) trovate maggiori dettagli sul processore MergeContent

## Esercizio 3

Nell'esercizio impareremo a sfruttare i ProcessGroup per realizzare componenti riutilizzabili in contesti differenti.

**Obiettivo:** Scrivere un componente riutilizzabile che dato in input uno stream di FlowFile ritorni un uscita uno stream di FlowFile il cui contenuto è stato compresso solo se la dimensione del contenuto in input era superiore a 1.000.000 bytes.

**Possibile caso di uso reale:** Prendo i file da una directory di input, li passo a questo componente e i file risultanti (compressi o meno) vengono scritti in una directory di output.



## Esercizio 3: traccia su come realizzare il dataflow


1. Definire un nuovo componente chiamato “ZipBigFile”
  - a. Creare un nuovo ProcessGroup contenente una porta di Input per prendere i dati in ingresso ed una porta di Output per restituire i dati in uscita.
  - b. Sui FlowFile in ingresso sfruttare il processore RouteOnAttribute sul valore dell'attributo “fileSize” per creare due possibili sottoflussi (“BigFile” e “SmallFile”) su cui distribuire i FlowFile.
  - c. Nel sottoflusso “BigFile” sfruttare il processore CompressContent per comprimere il contenuto del FlowFile mediante il compressore “gzip”.
  - d. Far confluire nella porta di Output definita i FlowFile dei due sottoflussi.
2. Provare ad utilizzare il componente definito collegando un processore GetFile per prendere i dati da una cartella di input e un processore PutFile per scrivere i file risultanti su disco.

Dati di esempio sono disponibili sulla cartella

“/home/cint/esercitazione/nifi/Esercizio3/” della macchina virtuale.

# Reddit: how to register a new app

You need valid credentials to use Reddit API!

- After logged in, go to <https://www.reddit.com/prefs/apps>
- Click 

# Reddit: how to register a new app (2)

## create application

Please [read the API usage guidelines](#) before creating your application. After creating, you will be required to [register](#) for production API use.

1)

**name**

☐ web app A web based application  
☐ installed app An app intended for installation, such as on a mobile phone  
☒ script Script for personal use. Will only have access to the developers accounts

**description**

**about url**

**redirect uri**

Fill with your data

2)

 **ScraperBOT**  
personal use script  
GKJvrD6XyEbfCXUYgGmwQw

change icon

**secret** yQpULg8Z8gk1nEjnw6JxejGvcU5MhQ

**name**

**description**

**about url**

**redirect uri**

[delete app](#)

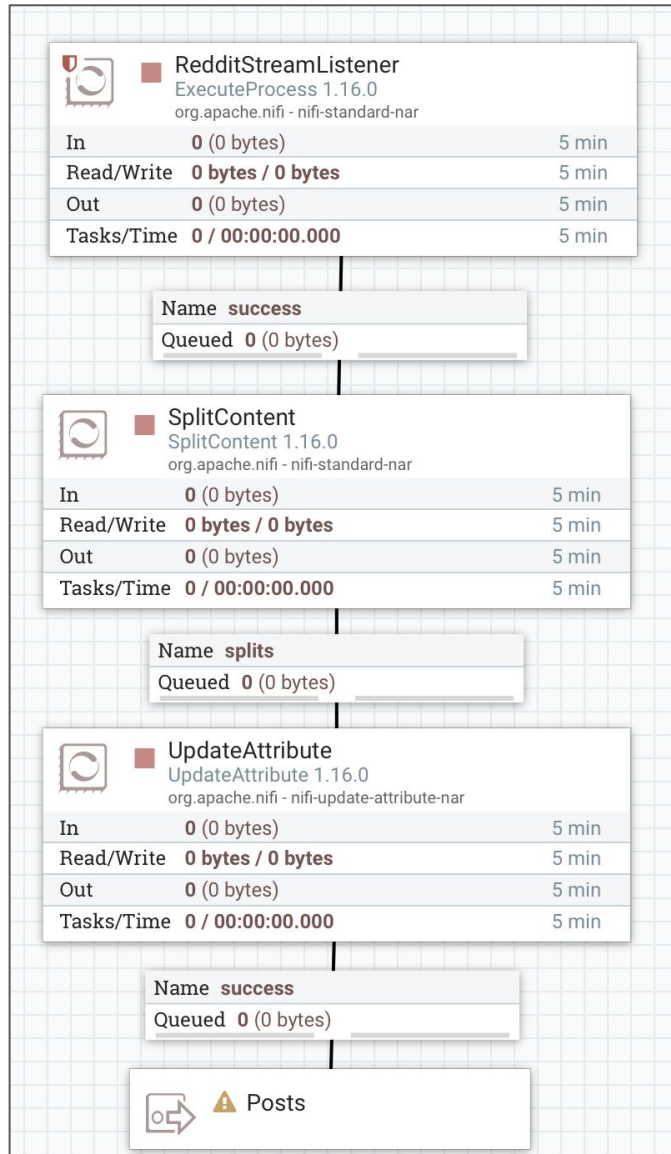
**developers** tf-iit (that's you!) [remove](#)

add developer:

Client ID

Secret key

# RedditListenerProcessor



- Based on script `RedditListener.py`
  - Specify “client ID” and “secret key”
  - Specify content type: “submission” or “comment”
  - Optionally specify the subreddit of interest
- The script produces JSON output
  - Customize `processComment()` using info here: [https://praw.readthedocs.io/en/stable/code\\_overview/models/comment.html](https://praw.readthedocs.io/en/stable/code_overview/models/comment.html)
  - Customize `processSubmission()` using info here: [https://praw.readthedocs.io/en/stable/code\\_overview/models/submission.html](https://praw.readthedocs.io/en/stable/code_overview/models/submission.html)
- The processor produce one FlowFile for each post retrieved

# RedditListenerProcessor: formato dati uscita

```
{
  "comment": {
    "author": {
      "id": "uf4ex",
      "name": "kboody22",
      "comment_karma": 7776,
      "is_mod": false,
      "is_employee": false
    },
    "body_html": "<div class=\"md\"><p>I don't know what my problem is, but for over 2 months, I pulled over 500lbs sumo with hook grip every day and no problem. I sign up for a meet that's in 5 weeks and I can't even hold on to 465lbs hook grip and now doubt if I should even compete. I don't understand why I get so into my own head . Training was going f PR's, but once I put this meet into my shifted in such a negative way. My last PL got DQ because I couldn't successfully get .</p>\n</div>",
    "created_utc": 1684847406,
    "parent_id": "t3_13pdcvr",
    "score": 1,
    "stickied": false,
    "submission_id": "13pdcvr",
    "submission_title": "May 23 Daily Thread",
    "subreddit_name": "r/weightroom",
    "subreddit_id": "t5_2ssmu"
  }
}
```

“Comment”  
content type

```
{
  "submission": {
    "author": {
      "id": "nchag8tx",
      "name": "M1LT0NK3YN3S",
      "comment_karma": 166,
      "is_mod": false,
      "is_employee": false
    },
    "id": "13poatn",
    "created_utc": 1684847909,
    "subreddit_name": "r/FreeKarma4All",
    "subreddit_id": "t5_luco6",
    "title": "UGOTMYVOTE TRAIN COMING",
    "url": "https://www.reddit.com/r/FreeKarma4All/comments/13poatn/ugotmyvote_train_coming/"
  }
}
```

“Submission”  
content type

## Esercizio 4

Nell'esercizio sfrutteremo i processori:

- `RedditListenerProcessor` per ottenere un flusso di commenti da tutti i subreddit tramite l'API di Reddit.
- `ExecuteStreamCommand` per eseguire uno script Python custom.

**Obiettivo:** Mettersi in ascolto sullo stream di Reddit che dà accesso ai nuovi commenti postati sul social e processarli tramite lo script `EnrichSentiment.py` (disponibile nella cartella `Esercizio4`). Lo script arricchisce i `FlowFile` con la polarità del sentiment di ciascun commento. Sulla base del valore di questo sentiment, si scrivono i `FlowFile` finali in tre cartelle distinte (sentiment negativo, sentiment neutro e sentiment positivo).

## Esercizio 4: traccia su come realizzare il dataflow.

**Stavolta provate a pensarci un attimo da soli!! :-)**

- Per il processore custom `RedditListenerProcessor` usate le credenziali app che vi siete creati tramite il sito di Reddit.
- Nel componente `ExecuteStreamCommand` ricordate di utilizzare come comando di Python `"python3"`.
- A parte questi due nuovi componenti, potete usare componenti che conoscete già per risolvere l'esercizio.

Dopo guardiamo insieme la soluzione, compreso il codice dello script.

## Esercizio 5

### Scaricamento delle pagine linkate nelle submission di Reddit

Estrarre il contenuto delle pagine Web linkate nei tweet (utilizzare lo script `WebPageContentExtractor_reddit.py` presente nella cartella `Esercizio5`). Lo script estrae, se disponibile, l'URL attaccato alla submission, scarica la pagina Web, estrae titolo e contenuto dell'articolo, e arricchisce il JSON originale con queste informazioni. Es:

```
"webpage" :  
{  
  "url": "https://t.co/6wixF7WHyq",  
  "title": "Notizia su Twitter...",  
  "content": "Ora sei..."  
}
```



## Esercizio 5 (2)

### Obiettivo

Mettersi in ascolto sulle nuove submission di Reddit e arricchirli con i contenuti Web tramite lo script `WebPageContentExtractor_reddit.py`. Sui FlowFile arricchiti, identificare quelli che hanno associato un contenuto Web da quelli che non hanno contenuti Web associati, scrivendo i JSON finali in cartelle separate.

In alcuni casi, utilizzando stream di tipo diverso (ad esempio ascoltando i nuovi tweet su Twitter), la coda di messaggi da processare potrebbe riempirsi molto velocemente. In tali casi, ammesso si disponga di hardware adeguato, è possibile velocizzarne il processamento agendo sul numero di istanze di un processore e lavorando su latenza/throughput.

Maggiori informazioni su come Nifi ottimizza latenza e throughput:

<https://community.cloudera.com/t5/Community-Articles/Understanding-NiFi-processor-s-quot-Run-Duration-quot/ta-p/248921>

# Esercizio 6

## Obiettivo

Riadattare i dataflow degli Esercizi 4 e 5 in modo tale da incapsularli in 2 componenti riutilizzabili. Sfruttando questi 2 nuovi componenti, scrivere un dataflow facente le seguenti cose:

1. Si mette in ascolto sulle nuove submission di Reddit provenienti da /r/all .
2. Le submission recuperate vengono processate e arricchite dal componente “Esercizio 4”.
3. Le submission arricchite dal componente “Esercizio 4” (solo quelli con polarità positiva o negativa) sono quindi processati dal componente “Esercizio 5”.
4. Le submission provenienti da componente “Esercizio 5” (sia con contenuto Web sia senza) sono memorizzate su Elasticsearch nell’indice “esercizio6”.
5. Le submission aventi polarità neutra oppure senza link a contenuto Web sono raccolte e memorizzate in una cartella di output.

# Esercizio 6: traccia visuale

- Sfruttare i ProcessGroup, le Input e le Output Port per riorganizzare gli esercizi già svolti
- Per processore PutElasticsearchHT TP è necessario ricordarsi di impostare la proprietà type a “\_doc”

