Politecnico di Milano

Advanced Computer Architecture - Project P10

Students: *Bressan G., de Santis S.*

Teaching Professor: *Cristina Silvano*

Professor: *Gianluca Palermo*

Academic Year 2016-2017

Project Purpose

The purpose of the project is to create a Dashboard for analyzing application performance at run-time using performance data broadcasted with three integrated tools:

- *Metricbeat*
- *Elasticsearch*
- *Kibana*

MetricBeat

MetricBeat is an application (daemon process) that permit to take all the information about the Machine on which it is running such as CPU percentage, Memory used and available, percentage of CPU and Memory used by a single process.
In our project MetricBeat is the daemon process that it is running on the “server” machine, in other words it will put on the machine in which you want to check and control the parameters. Once read the parameter, MetricBeat is able to upload the information into a database using a simple PUT request formatted in JSON and this database application is Elasticsearch.

Elasticsearch

Elastichsearch is one of the powerful application dedicated to store and read data in realt-time, it is based on JSON string. So when you want to store something you have to create a well-formed string in JSON with your information and you have to specify also

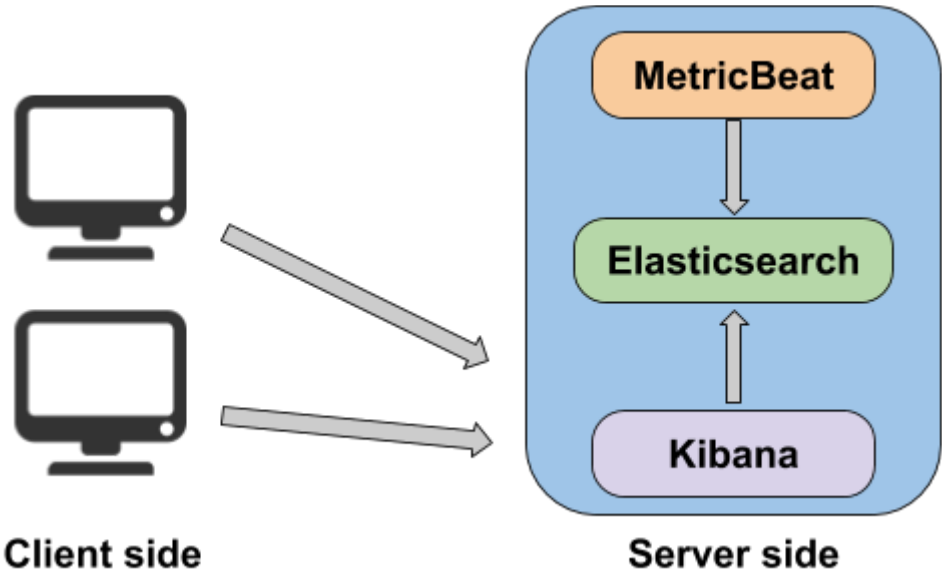
- **Index:** An index is like a ‘database’ in a relational database. It has a mapping which defines multiple types.
- **Doc_type:** Types are a convenient way to store several types of data in the same index.
- **Id:** the Identifier.

As said before Elasticsearch is provided by API that permit to do different requests using http/https protocol.

Kibana

At the end we need to visualize the data stored into Elasticsearch, and for this purpose we decided to use Kibana. Kibana is an open source data visualization application written in JavaScript. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line and scatter plots, or pie charts and maps on top of large volumes of data. In other words kibana does a request to Elasticsearch each time period defined into configuration file, but will be described better during the Setup description.

Example of how MetricBeat, Elasticsearch and Kibana cooperate



Setup

For the installation of all components, we have built a script that supply all the installation and configuration steps of Elasticsearch, Kibana and Metricbeat. Moreover the script will configure and sets itself all the custom visualization, dashboard, and sets the index pattern required by Kibana to show properly our collected data.
Following the next steps is possible to download and run the script.

On Ubuntu 16:
(Requires Java JDK)

Download the script file “p10_install.sh”
Make script executable:
chmod +x p10_install.sh
Run it:
sudo ./p10_install.sh

At the end of the script, the environment have all the required application configured well in order to show properly all the data required.
There is a recap in order to Start and Stop Elasticsearch, Kibana and Metricbeat.

Start:
sudo -i service elasticsearch start
sudo -i service kibana start
sudo /etc/init.d/metricbeat start

Stop:
sudo -i service elasticsearch stop
sudo -i service kibana stop
sudo /etc/init.d/metricbeat stop

Set configuration file

elastichsearch.yml

For setting elasticsearch.yml you have to go in this directory (Ubuntu 16):

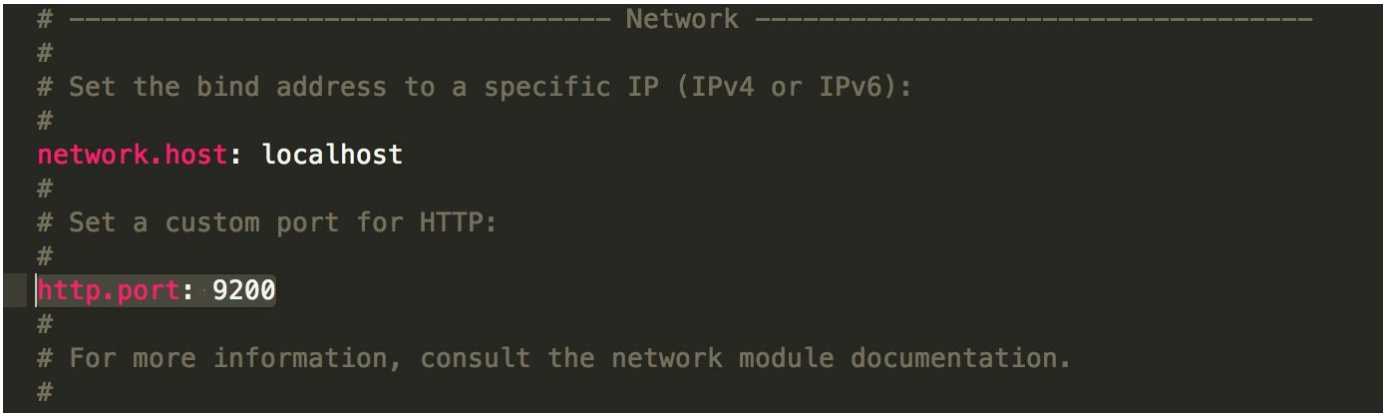
/etc/elasticsearch

after that you will see a file called **elasticsearch.yml**, so you have to open it with text editor and you have to modify these parameter in Network section of the file

From
#network.host: localhost
#http.port: 9200

To (you have only to delete '#')
network.host: localhost
http.port: 9200

netwok.host indicate the ip address of the machine in which run elasticsearch and http.port is the port of the service. Leave **network.host: localhost** and **http.port: 9200**.
See the example below



metricbeat.yml

For setting **metricbeat.yml** you have to go in this directory (Ubuntu 16):

/etc/metricbeat

after that you will see a file called **metricbeat.yml**, so you have to open it with text editor and you have to do two things. First of all go in 'System Module' section (of the **metricbeat.yml** file) and you have to modify these parameters:

From

```
- module: system
  metricsets:
    # CPU stats
    - cpu

    # System Load stats
    - load

    # Per CPU core stats
    #- core

    # IO stats
    #- diskio

    # Per filesystem stats
    - filesystem
```

```
    # File system summary stats
    - fsstat

    # Memory stats
    - memory

    # Network stats
    - network

    # Per process stats
    - process

    # Sockets and connection info (linux only)
    #- socket
```

To (you have only to delete '#')

```
- module: system
  metricsets:
    # CPU stats
    - cpu

    # System Load stats
    - load

    # Per CPU core stats
    - core

    # IO stats
    - diskio

    # Per filesystem stats
    - filesystem

    # File system summary stats
    - fsstat

    # Memory stats
    - memory

    # Network stats
    - network
```

```
  # Per process stats
  - process

  # Sockets and connection info (linux only)
  - socket
```

Deleting the '#' you can choose which parameter you can send to elasticsearch to be shown in kibana.

The example below show you the final result.

```
#------------------------------ System Module ------------------------------
- module: system
  metricsets:
    # CPU stats
    - cpu

    # System Load stats
    - load

    # Per CPU core stats
    - core

    # IO stats
    - diskio

    # Per filesystem stats
    - filesystem

    # File system summary stats
    - fsstat

    # Memory stats
    - memory

    # Network stats
    - network

    # Per process stats
    - process

    # Sockets and connection info (linux only)
    - socket
  enabled: true
  period: 10s
  processes: ['.*']
```

The second thing to do is to set the ip address and port of elasticsearch, in such a way to send the information taken from the system. So for do that you have to move in the '**Elasticsearch output**' part of the '**metricbeat.yml**' and change this parameters:

From
#hosts: ["localhost:9200"]

To (you have only to delete '#')
hosts: ["localhost:9200"]

Here you have to set the ip address and the port of Elasticsearch application.
This is an example

```
#------------------------- Elasticsearch output ------------------------------
output.elasticsearch:
  # Boolean flag to enable or disable the output module.
  #enabled: true

  # Array of hosts to connect to.
  # Scheme and port can be left out and will be set to the default (http and 9200)
  # In case you specify and additional path, the scheme is required: http://localhost:9200/path
  # IPv6 addresses should always be defined as: https://[2001:db8::1]:9200
  hosts: ["localhost:9200"]
```

kibana.yml

For setting **kibana.yml** you have to go in this directory (Ubuntu 16):

/etc/kibana

after that you will see a file called **kibana.yml**, so you have to open it with text editor and you have to modify these parameters of the file:

From
#server.port: 5601
#server.host: "192.168.1.63"
#elasticsearch.url: "http://localhost:9200"
#elasticsearch.username: "elastic"
#elasticsearch.password: "changeme"

To (you have only to delete '#')
server.port: 5601
server.host: "192.168.1.63"
elasticsearch.url: "http://localhost:9200"
elasticsearch.username: "elastic"
elasticsearch.password: "changeme"

Where server.host and server.port represent the address and the port of the machine in which kibana is running. For example in this case kibana is running on the machine with ip 192.168.1.63 on port 5601.
So for set server.host you have to know what is the internal ip address and put it on.
elasticsearch.url represent the address and the port in which is running elasticsearch and you have to leave "localhost" as address and "9200" as port.

At the end you have to set the username and password to access in kibana, so the default value are “elastic” as username and “changeme” as password.

The example below show you the final result.

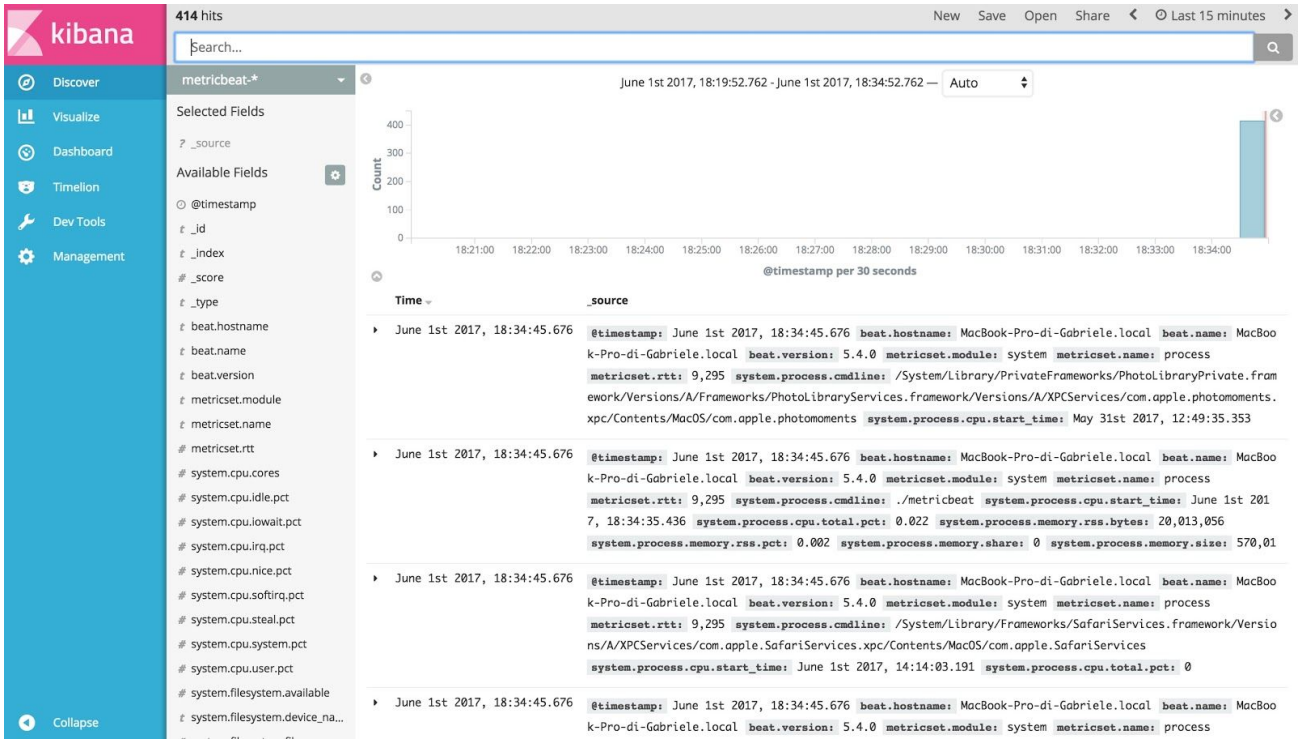
```
# Kibana is served by a back end server. This setting specifies the port to use.
server.port: 5601

# Specifies the address to which the Kibana server will bind. IP addresses and host names are both valid values.
# The default is 'localhost', which usually means remote machines will not be able to connect.
# To allow connections from remote users, set this parameter to a non-loopback address.
server.host: "192.168.1.63"

# Enables you to specify a path to mount Kibana at if you are running behind a proxy. This only affects
# the URLs generated by Kibana, your proxy is expected to remove the basePath value before forwarding requests
# to Kibana. This setting cannot end in a slash.
#server.basePath: ""

# The maximum payload size in bytes for incoming server requests.
#server.maxPayloadBytes: 1048576

# The Kibana server's name.  This is used for display purposes.
#server.name: "your-hostname"

# The URL of the Elasticsearch instance to use for all your queries.
elasticsearch.url: "http://localhost:9200"

# When this setting's value is true Kibana uses the hostname specified in the server.host
# setting. When the value of this setting is false, Kibana uses the hostname of the host
# that connects to this Kibana instance.
#elasticsearch.preserveHost: true

# Kibana uses an index in Elasticsearch to store saved searches, visualizations and
# dashboards. Kibana creates a new index if the index doesn't already exist.
#kibana.index: ".kibana"

# The default application to load.
#kibana.defaultAppId: "discover"

# If your Elasticsearch is protected with basic authentication, these settings provide
# the username and password that the Kibana server uses to perform maintenance on the Kibana
# index at startup. Your Kibana users still need to authenticate with Elasticsearch, which
# is proxied through the Kibana server.
elasticsearch.username: "elastic"
elasticsearch.password: "changeme"
```

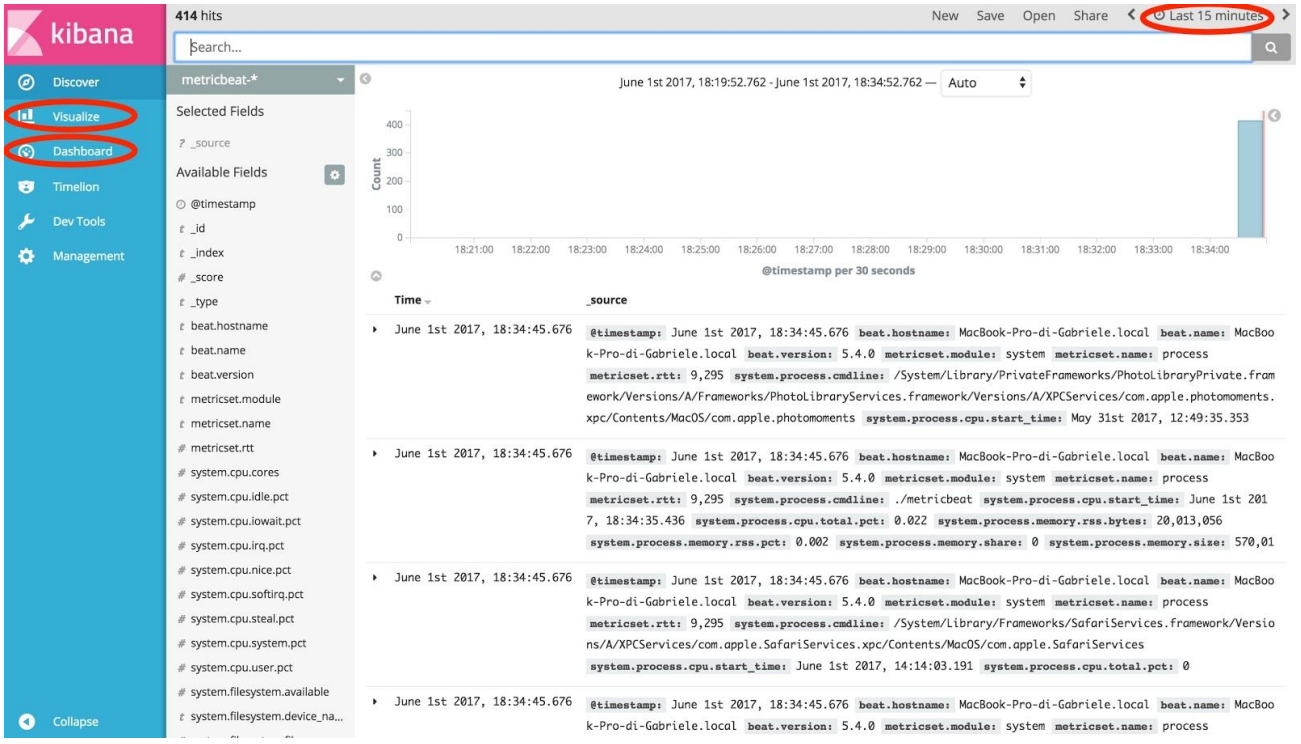
How to use kibana

One previous steps are done, from the client side you can see all the graphs from Kibana.
The right way to access into Kibana is to open your browser and go to the address composed by public ip of the server and as port you have to put the port that you have set in the config file of Kibana (**kibana.yml**). The default port is **5601**.
So the example is:

www.serverPublicIp:5601

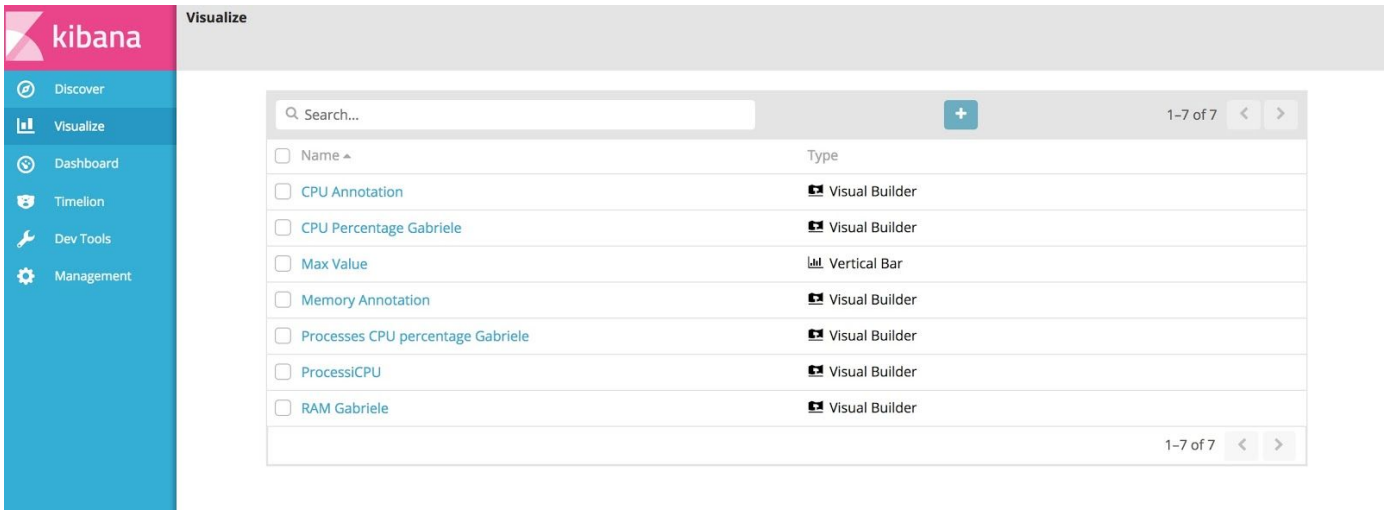
Once connected to Kibana, you will see the main page.





In kibana you can do different things.

First of all, you can create a new graph clicking on Visualize. As you can see in the image below there are different type of graph, so you can choose your best.



After created one or more graphs, you can create a Dashboard that contains graph/s. For doing that, you have to click on Dashboard, and "**add Dashoard**" (or plus button).

At the end, once a dashboard is created, you can choose to visualize data with different frequency of refresh only by clicking "**Last 15 minutes**" —> "**Auto-refresh**", and select the **refresh interval**

How insert values in Elasticsearch using Python

You can create a script in python that send the value to Elasticsearch to be visualized in Kibana. First of all you have to install the python library of elasticsearch in this way:

sudo pip install elasticsearch

After that, in your Python code you have to open the connection with elasticsearch using this command:

es = Elasticsearch(['elastic:changeme@localhost:9200'])

elastic: username of elasticsearch
changeme: is the password of elasticsearch
localhost: is the ip address of the machine in which elasticsearch is running
9200: is the port of the elasticsearch service

Once the connection is opened, to send the information/value you have to insert this command:

es.index(index="max_value", doc_type="max_value", id=uuid.uuid4(), body={"max": maxVal, "@timestamp": datetime.datetime.fromtimestamp(time.time()+time.timezone-3600).strftime('%Y-%m-%dT%H:%M:%S.%fZ')})

Index: An index is like a ‘database’ in a relational database. It has a mapping which defines multiple types.
Doc_type: Types are a convenient way to store several types of data in the same index.
Id: the Identifier, so we used uuid.uuid4() in such a way to have always a different and unique id.
body: in the body part you can insert the information that you want to send to elasticsearch and visualize in kibana. For example, in elasticsearch, under the tag “max” we will see a maxValue sent from python, and the timestamp well formed and acceptable for elasticsearch.