## Metricbeat Documentation

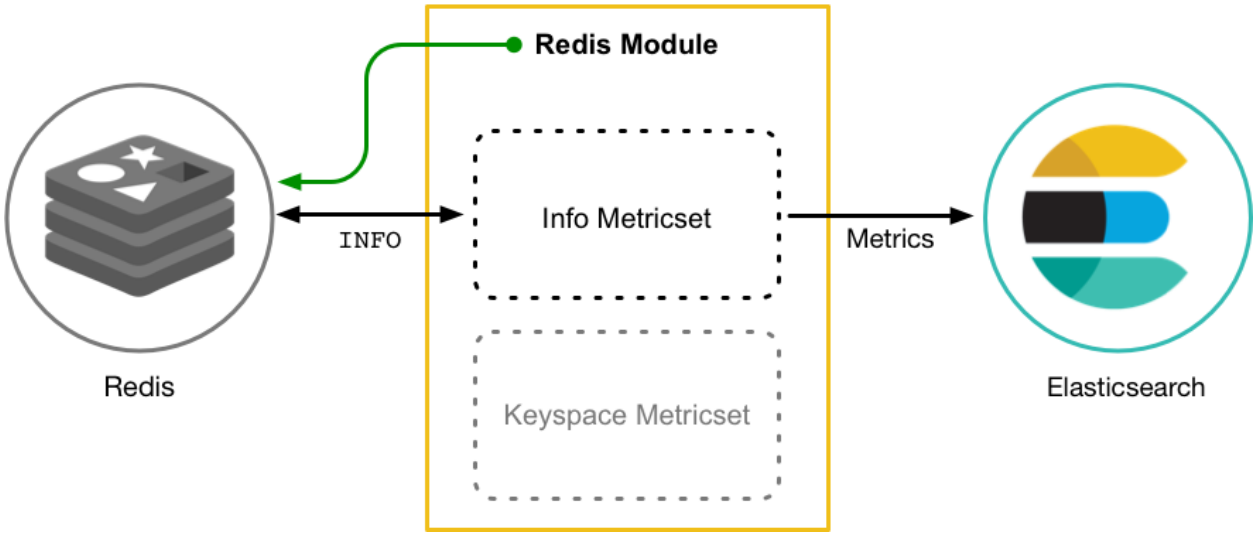
- Metricbeat is a lightweight shipper that you can install on your servers to periodically collect metrics from the operating system and from services running on the server.

- Metricbeat helps you to monitor servers by collecting metrics from the system and services running on the server.

How Metricbeat Works :

- Metricbeat collects data from servers or systems and puts it into logstash or elasticsearch.Metricbeat module defines how to connect with server or system and how to fetch the metrics from that and also which metrics to collect.

- Metricbeat from time to time interrogates the host system based on the time interval value specified in configuration.If it’s not connected to the system within the time it returns the error.



# Metricbeat Documentation

### Features of Metricbeat

- Metricbeat sends more than just metrics. When it cannot retrieve metrics, it sends error events. The error is not simply a flag, but a full error string that is created during fetching from the host systems. This enables you to monitor not only the metrics, but also any errors that occur during metrics monitoring.

- Metricbeat sends the raw data retrieved from the host to the output for processing. When using Elasticsearch, this has the advantage that all raw data is available on the Elasticsearch host for drilling down into the details, and the data can be reprocessed at any time.

- Metricbeat sends more than just numbers. The metrics that Metricbeat sends can also contain strings to report status information. This is useful when you're using Elasticsearch to store the metrics data. Because each metricset has a predefined structure, Elasticsearch knows in advance which types will be stored in Elasticsearch, and it can optimize storage.

- Rather than containing a single metric, each event created by Metricbeat contains a list of metrics. This means that you can retrieve all the metrics in a single request to the host system, resulting in less load on the host system. If you are sending the metrics to Elasticsearch as the output, Elasticsearch can directly store and query the metrics as a nested JSON document, making it very efficient for sending metrics data to Elasticsearch.

# Metricbeat Documentation

**Configure**:

- In the **modules**.**d** directory there are many different configuration files available which are related to metricbeat configuration.we can enable it from **metricbeat.yml** file.
- In metribeat.yml file we can set **metricbeat.config.module:**
    - ➔ **Path**: we can set a path of any configuration file from module.d directory
    - ➔ **Reload**: we can set reloading of file with period of time
    - ➔ **Reload.period**: To set a period of reloading

- In metricbeat.yml we can also choose template settings :
    - ➔ **Setup.template**:this setting tells elasticsearch how to config index when it is created.
    - ➔ We can set number of shards inside this config setting
    - ➔ **Index**.**codec**: The `default` value compresses stored data with LZ4 compression, but this can be set to `best_compression` which uses higher compression ratio.

- **General Settings**:
    - ➔ In general settings we define the name of the shipper that publishes the network data. It can be used to group all the transactions sent by a single shipper in the web interface.
    - ➔ **tags:** The tags of the shipper are included in their own field with each transaction published.

- **Kibana**:
    - ➔ **Setup.kibana: In kibana setup we are provide host into it , in our case it is “localhost:5601”.**

- **Elastic Cloud**:
    - ➔ By using this settings we can setup metricbeat with the elastic cloud (https://cloud.elastic.co/)
    - ➔ Here we just have to provide cloud Id and authentication
    - ➔ **cloud.id** and **cloud.auth**

# Metricbeat Documentation

- **OUTPUT**:

- **Output into Elasticsearch** :
  - ➔ In ELasticsearch output we need to provide path of the host on which elastic search is up
    Ex: **hosts:[“localhost:9200”]**

  - ➔ We also provide a username and password of a elasticsearch to authenticate a user
    **api_key: "id:api_key"**
    **username: "elastic"**
    **password: "changeme"**
  - ➔ Here we need either API or either authentication Details

- **Output to Logstash**:
  - ➔ We can also send our metrics output to logstash ,same as elasticsearch we have to provide host here
    Ex: **hosts:[“localhost:5044”]**
  - ➔ We can also provide SSL certificate for authentication
    **ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]**
  - ➔ **ssl.certificate: "/etc/pki/client/cert.pem"**

- **Processors**:
  - ➔ We can define processors in configuration file to process events before they are sent to the configured output.
  - ➔ It enhance events with additional metadata
  - ➔ Performing additional processing and decoding .

# Metricbeat Documentation

- **Logging:**
  - ➔ The logging system can write logs to the syslog or rotate log files. If logging is not explicitly configured the file output is used.

    ```
    logging.level: info
    logging.to_files: true
    logging.files:
      path: /var/log/metricbeat
      name: metricbeat
      keepfiles: 7
      permissions: 0640
    ```

  - ➔ Logging step helps us to log informational messages, warnings, errors, and critical errors. When the log level is `debug`, you can specify a list to display debug messages for specific components.

- **X-Pack monitoring**:
  - ➔ Metricbeat can export internal metrics to a central Elasticsearch monitoring cluster. This requires xpack monitoring to be enabled in Elasticsearch.
    **monitoring.enabled: True**

  - ➔ We can set cluster uuid to monitor Elasticsearch cluster under which monitoring data for this Metricbeat instance will appear in the Stack Monitoring UI. If output.elasticsearch is enabled, the UUID is derived from the Elasticsearch cluster referenced by output.elasticsearch.
    **monitoring.cluster_uuid:**

**FOR DOCKER:**

- TO run Metricbeat on docker ,configuration is almost the same as a local system, the additional thing is we create one docker file where we give the path of our metricbeat file which we have defined with all above configuration. After that we run that separate docker image or docker-compose file to run that container of metricbeat service .

- We can use **`add_docker_metadata: ~`** into the processor of metribeat.yml .

# Metricbeat Documentation

**How To Setup Metricbeat:**

1. **For metricbeat setup you have to first install Elasticsearch and Kibana in your system and run it.**

2. **After that you have to install metricbeat in your sys**

```
-curl -L -O
https://artifacts.elastic.co/downloads/beats/metricbeat/met
ricbeat-8.3.3-amd64.deb
-sudo dpkg -i metricbeat-8.3.3-amd64.deb
```

3. **Set the connection information in `metricbeat.yml`. To locate this**

```
output.elasticsearch:
hosts: ["https://myEShost:9200"]
username: "metricbeat_internal"
password: "YOUR_PASSWORD"
ssl:
  enabled: true
  ca_trusted_fingerprint:
"b9a10bbe64ee9826abeda6546fc988c8bf798b41957c33d05db736716513dc9
```

4. **Enable and configure metrics collection modules**

   1. Identify the modules you need to enable.

```
-metricbeat modules list
```

   2. `-sudo metricbeat modules enable nginx`
   3. `-metricbeat setup -e`

5. **Start Metricbeat**

`-sudo service metricbeat start`

6. **View Your logs in Kibana**

Point your browser to http://localhost:5601

In the side navigation, click Discover.