Operations Analytics and Dashboards

NextGEOSS User Management
1. Using Analytics dashboards in Client applications

The figure below presents an overview of the logical architecture of the Analytics Engine. On one side there are the Data Sources, that provide information important to be collected for further use. These Data Sources can be the NextGEOSS DataHub, the user accounts registered and signed-in through the NextGEOSS User Management service, and the pilot services that are running on the NextGEOSS Platform.

The information is gathered through a Data Stream Processor, that collect the raw data through different channels keeping the important information, and subsequently stored in a Database. After the information is stored in the database it is then possible to use Data Visualisation tools that query and prepare the outputs ready to be used by the users, either through reports, alerts, dashboards that can be useful to identify problems, trends, needs and other valuable decision-making information.

2. Ingestion of data logs and metrics

Currently, the ingestion of data and metrics is performed by Terradue and Deimos teams, on behalf of partner users integrating a Pilot.
This is a critical step to be able to create the different analytics dashboards within the NextGEOSS Analytics Engine.

According to the NextGEOSS Analytics Engine architecture, the metrics are collected within the different components of the system using a lightweight software package called Beats. These Beats are Data Shippers that can collect information from logs, system, network and forward the information to a node that is able to parse and extract what really matters. This node is called the Logstash, that parse the information received by the different Beats and forwards it to a database, which is a Elasticsearch instance.

The NextGEOSS Analytics Engine has already deployed an instance of Logstash and Elasticsearch ready to collect information from the different sources. The instructions provided below show how to install a Filebeat and a Metricbeat along with the instructions to send the data to the Logstash instance of NextGEOSS. This is a pre-version of the Integration Guide which will be updated during the NExxtGEOSS project Consolidation phase (covering the period from EP2 to EP3 milestones).

2.1. **Filebeat**

Conufgure Filebeat on a remote node to collect data that can be sent to NextGEOSS Logstash instance.

Note that:

- Filebeat just ships files and logstash interprets these files.
- The format of data can be agreed. CSV format straightforward because it is easy to parse and has some formality to it's structure. However Filebeat is very flexible with the log format.
- The references below allow deployment to various target environments.
- Please use the version stated below. We have noticed in the past that sometimes mixing different versions does not work.
- This installation also includes examples of the logstash configuration to process the data collected.
References


Version

- ELK Stack 6.1.1

Installation


Note that filebeat.yml must be configured so that:

- "paths:" is configured to point at the correct file to be shipped. E.g. the log file.
- "data_provider:" is configured to use the correct provider name. E.g. TDUE. This name should be agreed with Deimos

**Example deployment using docker**

Here is an example of how to deploy using docker, but other solutions describe above can be used as long as they use the filebeat.yml provided.

**Step 1: Install Filebeat**

e.g.

docker pull docker.elastic.co/beats/filebeat:6.1.1

use docker images to find out what is there or confirm the above action

**Step 2: Configure Filebeat**

see filebeat.yml above

Modify hosts: "["213.63.139.62:5044"]" to use the agreed ip address

Modify paths: to include log files path
**Step 3: Configure Filebeat to use Logstash**

see filebeat.yml above
e.g. store in /home/gass/ELK_docker_searchguard/filebeat/config/filebeat.yml

**Step 4: Start Filebeat**
e.g. ( this sets filebeat.yml and the log-data directory containing the log file )
docker run -d --name nextgeoss_filebeat -v /home/gass/ELK_docker_searchguard/filebeat/config/filebeat.yml:/usr/share/filebeat/filebeat.yml -v /home/gass/ELK_docker_searchguard/pilot-data:/log-data
docker.elastic.co/beats/filebeat:6.1.1

To stop the service
- sudo docker ps
- sudo docker stop nextgeoss_filebeat
- sudo docker rm nextgeoss_filebeat

**other commands**
- docker logs nextgeoss_filebeat // see logs
- docker exec -it nextgeoss_filebeat /bin/bash // to run a command prompt in a docker

**Note:** the above config create and index in the format logstash-tdue-%{+YYYY.MM.dd}

### 2.2. Metricbeat

This Metricbeat deployment is intended to collect metric data from a remote VM and then display data in a NextGEOSS Analytics dashboard.

Please use the version stated below. We have noticed in the past that sometimes mixing different versions does not work.

**References**
Version

- ELK Stack 6.1.1

Installation

  Follow the steps
  Use this metricbeat.yml template.

Note that metricbeat.yml must be configured so that:

- name: is configured as agreed. E.g. "TDUE". This name should be agreed with Deimos

Example deployment using Docker

Here is an example of how to deploy using docker, but other solutions describe above can be used as long as they use metricbeat.yml


Step 1: Install Metricbeat

```
cd user_defined_target_directory
sudo mkdir metricbeat
cd metricbeat
add metricbeat.yml  // see same directory as this file
give metricbeat.yml the correct permissions and access rights
sudo chown 1000 metricbeat.yml
sudo chmod go-w metricbeat.yml
```

Step 2: Configure Metricbeat

see metricbeat.yml attached

Modify name: "SIMOcean" to use the correct unique name agreed with Deimos: E.g. TDUE or TDUE1, TDUE2, TDUE3 etc
Modify username: "******" to use the correct username provided separately by Deimos
Modify password: "******" to use the correct password provided separately by Deimos

**Step 3: Get Metricbeat**
e.g.
docker pull docker.elastic.co/beats/metricbeat:6.1.1

**Step 4: Start service**
docker (seems the best approach at the moment - see below )
cd /home/gass/metricbeat
run in background -d
sudo docker run -d --name <nodename> --volume=/proc:/hostfs/proc:ro --volume=/sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro --volume=/data:/data:ro --net=host -v /home/gass/metricbeat/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml
docker.elastic.co/beats/metricbeat:6.1.1 -system.hostfs=/hostfs
or run in foreground to test if needed
sudo docker run --name <nodename> --volume=/proc:/hostfs/proc:ro --volume=/sys/fs/cgroup:/hostfs/sys/fs/cgroup:ro --volume=/data:/data:ro --net=host -v
/home/gass/metricbeat/metricbeat.yml:/usr/share/metricbeat/metricbeat.yml
docker.elastic.co/beats/metricbeat:6.1.1 -system.hostfs=/hostfs

to stop the service
- sudo docker ps
- sudo docker stop <nodename>
- sudo docker rm <nodename>

After metrics are collected within the different components of the system using the described Beats software packages, it is possible to configure the analytics dashboards that will provide a visual rendering over these metrics. For this, the NextGEOSS Platform leverages a server instance of the 'Grafana' software.
3. **Using Grafana to create Analytics Dashboards**

The framework of the Analytics Dashboards service has been developed as a self-service, customizable service based on open-source solutions. The "Grafana" solution provides the analytics visualisation tool used in NextGEOSS to develop dashboards. It supports multi-tenancy to address a large user base, such as foreseen by NextGEOSS. It is possible to create public or private dashboards. It also supports alert mechanisms to notify owners in case of threshold breaches. It integrates smoothly with the Logstash/Elastic engine. Users build their dashboards once, and then the dashboards get updated in real time from the Beats components deployed on their application backend, and sending metrics to the Logstash/Elastic engine.

The Grafana server is currently made available to the NextGEOSS partners at this URL:

- [http://analytics-grafana.deimos.pt/login](http://analytics-grafana.deimos.pt/login)

To create an analytics dashboard, we'll login on that server with a NextGEOSS UM account, and then interact with four key concepts:

- Data Source (for us, it is the Elastic engine)
- Folders (the Dashboards are organized in Folders)
- Panels (the Visualization tools)
- Dashboard (a collection of Panels)
Click on the button "+Dashboard" to create a new dashboard:

Select a visualization tool:
Click on the edit option to start configuring the selected Panel:

This is the configuration page of the visualization tool to set all the parameters:

Choose the Data Source (e.g. the Index associated with the data in Logstash):
Configure the Query and other Parameters:

The Graph preview appears according to the configuration previously set:

Finally, update other Visualization Parameters:
After Login, navigate to the User Page where the list of registered users can be accessed.

You can assign User Roles:

- Admin
- Editor
- Viewer

From the Configuration function, click on the "+Add Team" button. A user group will be created to assign further any access privileges.
Navigate to the Dashboard Folders page, and select the folder where you wish to assign the access Privilege.

In the Permissions tab of the Folder management section, you can choose either the list of users or the User Group (Team) to whom to provide access. This way you can create a private Dashboard.
Once you have set the permission as Viewer, the dashboard is available publicly for anonymous access. This way Public Dashboards are created.

Finally, you can also configure alerts: