

# Cloud Computing Applications and Services

## University of Minho

### Guide 4

## Monitoring

2024

Now, you have the tools to automatically deploy and configure *Swap* across multiple universities in Portugal using Docker and Kubernetes. However, maintaining a reliable service at such a scale requires continuous monitoring of both application performance and infrastructure health. This is where monitoring tools become crucial, enabling insights into resource usage, identifying bottlenecks, and allowing for proactive issue responses. Therefore, the main goal of this guide is to deploy and use a modular system monitoring tool to analyze the execution of the *Swap* application. Along this guide, you will go through the steps of configuring and deploying Elastic Stack as the Cloud Provider (☁️). Then, you will use its services to monitor the *Swap* application, taking the role of the Application Developer (👩🏫).

*Recall these symbols from previous guides:*

- ☁️ Tasks to be performed by the *cloud provider*;
- 👩🏫 Tasks to be performed by the *application developer*;

The symbol 🖥️ signals tasks that need to be performed in a specific machine (*myvm*).

#### Elastic Stack

Elastic Stack is a suite of open-source products from Elastic designed to ingest, search, analyze, and visualize data from any source and format in real-time. In this guide, we will deploy the following Elastic Stack components:

- Elasticsearch: <https://www.elastic.co/elasticsearch>
- Kibana: <https://www.elastic.co/kibana>
- Metricbeat: <https://www.elastic.co/beats/metricbeat>

## 1 Setup

You can either choose to do the tasks below with the K8s cluster (from Guide 3) or the Docker setup (from Guide 2). Choose the setup to run according to the resources (*i.e.*, CPU, RAM, Disk) available at your computer.

### 1. For the K8s setup (requires at least 7 GiB of RAM):

- **Warning:** If you have destroyed your VMs after finishing Guide 3, probably it is better to follow the *Docker setup* (it is simpler and faster to deploy). Otherwise, you can reuse your k8s cluster.
- Boot the VMs from Guide 3 (*myvm*, *controlplane*, *node1*, *node2*) and the *monitor* VM:

```
vagrant up myvm controlplane node1 node2 monitor
```
- Make sure *Swap* is running correctly.

### 2. For the Docker setup (requires at least 4 GiB of memory):

- Boot the VMs from Guide 2 (*myvm*, *node1*) and the *monitor* VM:

```
vagrant up myvm node1 monitor
```

- Install Docker and deploy Swap on *node1* using the playbooks provided with the codebase accompanying this guide. **Note:** These playbooks are based on the ones done for Guide 2:

```
// from CloudProvider project
ansible-playbook install_docker.yml

// from AppDeveloper project
ansible-playbook docker-swap-install.yml
```

- Make sure Swap is running correctly.

## 2 Tasks

### 2.1 Install Elastic Stack

Let us install and configure Elastic stack (i.e., Elasticsearch, Kibana, and Metricbeat services). Inspect the new playbook (*install\_mon.yml*) and roles provided along this guide for the *CloudProvider* project.

1. The inventory (*hosts* file) now has one more group: the *elk* group, composed of the *monitor* VM.
  - Make sure the IPs defined on the *hosts* file are correct according to your setup.
2. Inspect the role *elk\_es\_ki\_services* that installs the Elasticsearch and Kibana services at the *monitor* VM.
3. Inspect the role *elk\_metricbeat\_services* that installs the Metricbeat service at the other VMs of your cluster. Note that besides installing the service, this role is responsible for setting up default indexes at ElasticSearch and dashboards at Kibana.
4. Finally, did you noticed a *meta* subfolder at the two roles? This folder contains meta tasks that are called before running the role (e.g., dependencies). In this case, our playbook is calling another role (*elk\_repo*) that is responsible for setting up the repository for Elastic Stack components. Make sure you also inspect this role.
5. Connect to *myvm* and use this playbook to automatically install Elasticsearch and Kibana on the *monitor* VM, and Metricbeat on the other VMs:

```
ansible-playbook install_mon.yml
```

**Note:** If your playbook fails with an *apt cache error*, run the following command at the affected cluster VMs (this command updates the key for the k8s repository):

```
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.31/deb/Release.key | sudo gpg --dearmor
-o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
```

### 2.2 Monitor Swap's execution

Now that we have Elastic Stack installed and running, let us monitor Swap's execution and check the usage of system resources.

#### 2.2.1 Explore Metricbeat Dashboards

Access Kibana through the web application and explore Metricbeat's predefined dashboards.

1. Open Kibana at `http://<MONITOR_IP>:5601`
2. Observe summarized data in the Analytics => Dashboard page (e.g., "[Metricbeat System] Host overview ECS").
3. Explore the other menus from Kibana.

### 2.2.2 Create a custom Dashboard

Create your own dashboard and include predefined and custom visualizations.

1. Go to “Dashboards” and click on the “Create dashboard” button.
2. Add a predefined visualization from the library by clicking on the “Add from library” button and selecting one visualization (e.g., “CPU Usage [Metricbeat System] ECS”).
3. Create a custom visualization that shows the percentage of CPU used over time (per process):
  - (a) Click on the “Create visualization” button.
  - (b) Fill in the right side as follows:
    - Change “Bar Vertical stacked” to “Line”.
    - On the “Horizontal axis”, select “Date histogram” as the “Function” and “@timestamp” as the “Field”.
    - On the “Vertical axis”, select “Average” as the “Function” and “system.process.cpu.total.pct” as the “Field”.
    - On the “Breakdown”, select “Top values” as the “Function” and “process.name” as the “Field”.
4. Create another custom visualization that shows the bytes of memory (field `system.process.memory.rss.bytes`) used over time (per process).
5. Save the dashboard by click on the button “Save and return”.

### 2.2.3 Observe / analyze monitoring data

Through the previous dashboards and visualizations try answering the following questions:

1. Which resources (e.g., CPU, memory, network, disk) are being used the most?
  - (a) Is resource utilization similar across cluster nodes?
2. Which process/processes are consuming the most resources?
  - (a) What optimizations or deployment configurations would you apply to enhance overall resource usage?

## Learning Outcomes

Recognize different roles in a modular monitoring pipeline. Apply the Elastic Stack to monitor and visualize server and application resources.