**Hands On Adding data to Elastic Search Using Beats**

Adding data using Beats

In the previous recipes in this chapter, we learned how to ingest timestamped data using Elastic Agent and its integrations. Sometimes, you may not find an available Elastic Agent integration for your data source, but there might be a Beats module that supports it. In such cases, using Beats becomes the natural alternative. Beats is a suite of lightweight data shippers that can efficiently gather and transport data to Elasticsearch for analysis and visualization. In this recipe, we will learn how to configure and use Beats to collect data from a data source and send it to Elasticsearch.

By following the steps in the upcoming *How to do it…* section, you will learn how to set up **Metricbeat** to monitor **Apache Tomcat** metrics and stream this data to Elasticsearch. Metricbeat, one of the many Beats shippers supported by Elasticsearch, collects metrics from your systems and services and sends them to Elasticsearch or Logstash for indexing. We have chosen Apache Tomcat metrics as an example to demonstrate how Beats can be used as an alternative when Elastic Agent integration is still missing (At the time this book was written, there was no Elastic Agent integration available for Tomcat metrics, although the Metricbeat Tomcat module was available).

Getting ready

Prior to ingesting data, ensure that you have a running Apache Tomcat instance with the supported version, Apache Tomcat 9.x. To fetch metrics, we will use Jolokia, which means that both Apache Tomcat and Jolokia need to be installed. Let’s first look into the installation instructions for Apache Tomcat and Jolokia.

Installing Apache Tomcat 9

There are multiple ways to set up an Apache Tomcat instance. For this recipe, we will use the **Google Cloud Platform** (**GCP**) Marketplace solution (<https://console.cloud.google.com/marketplace/browse(cameo:product/cloud-infrastructure-services/tomcat-ubuntu>) to provision an **Ubuntu** VM on the GCP Compute Engine, bundled with a preconfigured Apache Tomcat 9 instance.

Note

If you prefer other cloud providers or a self-managed environment for your Apache Tomcat instances, the instructions in this recipe will still be applicable. For a self-managed installation, the official Tomcat setup guide can be found here: <https://tomcat.apache.org/tomcat-9.0-doc/setup.html>.

Jolokia

Jolokia (<https://jolokia.org/>) is an open source Java library and agent that streamlines the management and monitoring of Java applications by providing a user-friendly HTTP interface to interact with JMX MBeans. The Metricbeat Tomcat module requires Jolokia to fetch JMX metrics; therefore, Jolokia must be installed in the Tomcat application. Detailed instructions are available at <https://jolokia.org/tutorial.html>.

The snippets of the recipe can be found at <https://github.com/PacktPublishing/Elastic-Stack-8.x-Cookbook/blob/main/Chapter4/snippets.md#adding-data-from-beats>.

How to do it...

In this recipe, we detail the installation and configuration of Metricbeat on Ubuntu, aimed at integrating it with Elastic Cloud and capturing metrics through the Tomcat and Jolokia modules. We will also demonstrate how to set up Metricbeat to stream data and use Kibana for visualization. Moreover, we will illustrate how to monitor system health via Metricbeat’s System module within Kibana’s Observability UI. Let’s get started:

1. Start by installing Metricbeat with the appropriate operating system and Elastic Stack version; in our example, we deploy Metricbeat with version 8.12.2 on Ubuntu (deb):

curl -L -O   https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-8.12.2-amd64.deb

sudo dpkg -i metricbeat-8.12.2-amd64.deb

1. Once installed, you need to configure Metricbeat to connect to your Elastic Cloud deployment. The Metricbeat configuration file is located at /etc/metrcibeat/metricbeat.yml.
2. In the configuration file, you can provide the cloud.id details of your Elasticsearch service and set cloud.auth to a user who has the necessary authorization to set up Metricbeat, as shown here:

cloud.id: "YOUR\_CLOUD\_ID"

cloud.auth: "YOUR\_USERNAME:YOUR\_PASSWORD"

Next, we will configure the Metricbeat Tomcat module.

1. To enable the Tomcat module, you can use the following command:

sudo metricbeat modules enable tomcat

1. The configuration file is located at /etc/metricbeat/modules.d /tomcat.yml. For the Tomcat metrics in our example, we don’t need to modify the default configuration as the configuration is set in the Jolokia module in the next steps.
2. To retrieve the Tomcat metrics, configure the Metricbeat Jolokia module:  
     
   The configuration file is located at /etc/metricbeat/modules.d/jolokia.yml.

sudo metricbeat modules enable jolokia

The most important part of the configuration is the jmx.mapping attribute, where we map the JMX MBean attributes to **Elastic Common Schema** (**ECS**) attributes. We omit the full mapping here, but the complete configuration file example can be found at <https://github.com/PacktPublishing/Elastic-Stack-8.x-Cookbook/blob/main/Chapter4/metricbeat/jolokia.yml>:

- module: jolokia

  metricsets: ["jmx"]

  period: 10s

  hosts: ["localhost"]

  namespace: "metrics"

  path: "/jolokia/?ignoreErrors=true&canonicalNaming=false"

  jmx.mappings:

    ...

    - mbean: 'java.lang:type=Memory'

      attributes:

        - attr: HeapMemoryUsage

          field: memory.heap\_usage

        - attr: NonHeapMemoryUsage

          field: memory.non\_heap\_usage

    ...

  jmx.application:

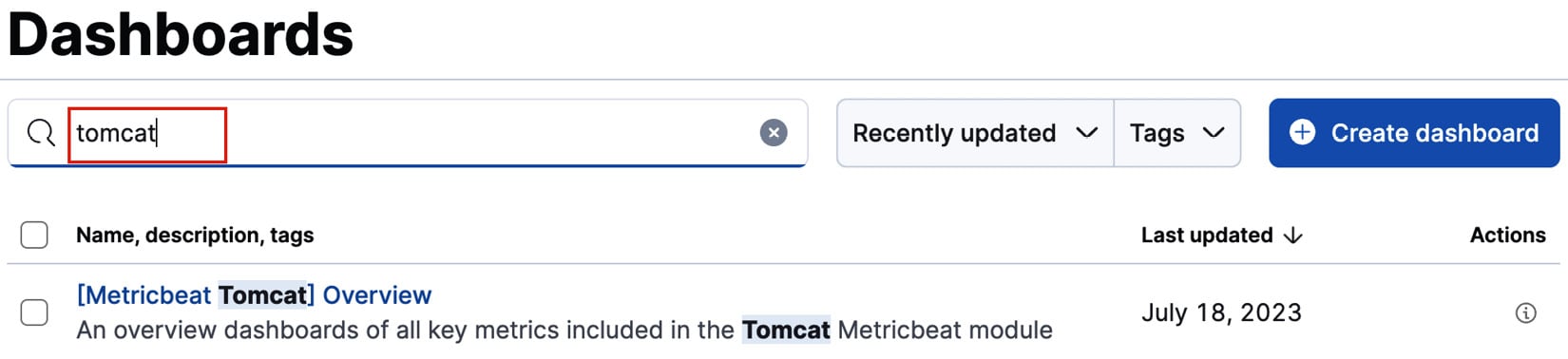
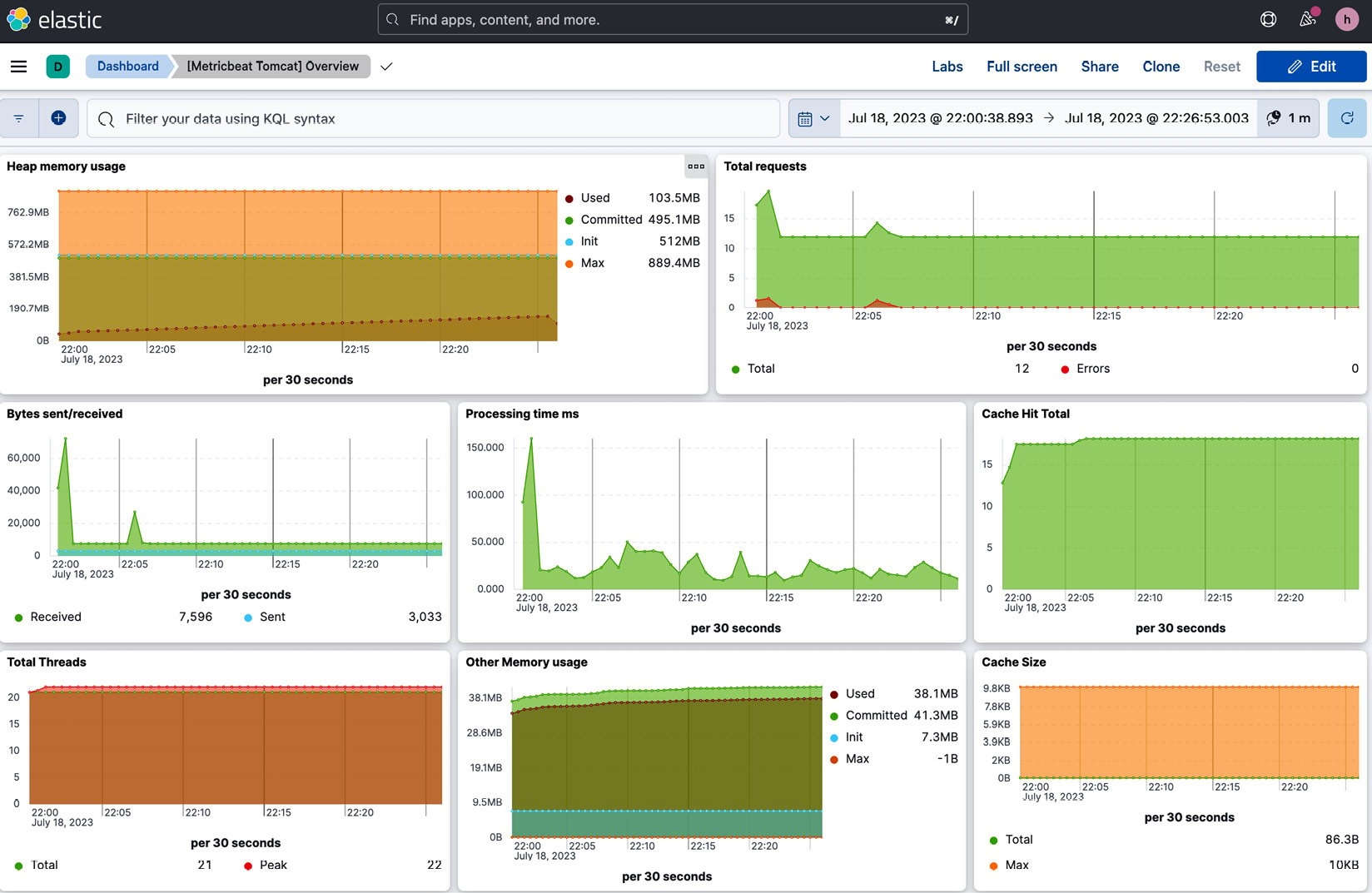
  jmx.instance:

1. Once the configuration is finalized, start Metricbeat to begin collecting Apache Tomcat metrics. Use the following command to set up the related assets (visualizations, dashboards, and index templates):

sudo metricbeat setup -e

1. Then, start Metricbeat with the following command:

sudo service metricbeat start

1. At this stage, Metricbeat will begin streaming metrics to your deployment; you can verify the metrics on the default Metricbeat Tomcat dashboard. In Kibana, go to **Analytics** | **Dashboards**, and type tomcat in the search bar to find the **[Metricbeat Tomcat] Overview** dashboard, as shown in *Figure 4.19*:  
   
2. Click on the dashboard to verify the metrics collected by the Metricbeat Tomcat and Jolokia modules, such as memory, threading, and global requests, as shown in *Figure 4.20*:  
   
3. The **System** module is enabled by default, allowing you to also verify metrics in the **Observability Infrastructure** UI. In Kibana, head to **Observability** | **Infrastructure** | **Hosts** to inspect the system metrics of the Ubuntu VM, as shown in *Figure 4.21*:

