# Practice M7: Elastic Stack

During this practice we will assume that we are working in Linux environment. It could be a physical machine or a virtual one. The distribution of choice is not that important, but it will be better to stick to some of the well supported distributions.

Most of the steps can be executed in Windows and/or macOS environment as well either directly or in a VM.

The lab infrastructure will vary during different parts of the module. For each part there is one Vagrantfile file.

## Part 1: Elastic Stack

The only prerequisite is Java 8 or later and it can be either the one provided by Oracle, or the OpenJDK.

There are two valid ways to install Elastic Stack components – first is to add the repository and then install the packages, and the second is to download the package directly and then install it.

#### Install Elasticsearch (on CentOS)

##### Repository

In order to follow this path, we must do:

* First, import the Elasticsearch GPG key:

**rpm --import https://artifacts.elastic.co/GPG-KEY-elasticsearch**

* Then, create a repository file **/etc/yum.repos.d/elasticsearch.repo** with the following content:

**[elasticsearch-6.x]**

**name=Elasticsearch repository for 6.x packages**

**baseurl=https://artifacts.elastic.co/packages/6.x/yum**

**gpgcheck=1**

**gpgkey=https://artifacts.elastic.co/GPG-KEY-elasticsearch**

**enabled=1**

**autorefresh=1**

**type=rpm-md**

* Alternative option is to use another **baseurl** which will let us use just the features available under Apache 2.0 license. For this to happen, we must use:

**baseurl=https://artifacts.elastic.co/packages/oss-6.x/yum**

* And finally, install the package:

**sudo yum install elasticsearch**

##### Direct

This is the easier and quicker installation method:

* First, download the package. It can be the regular one:

**wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.rpm**

* Or the one available under the Apache 2.0:

**wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-oss-6.5.4.rpm**

* Then, install it:

**sudo rpm -Uvh elasticsearch-\*.rpm**

#### Install Elasticsearch (on Ubuntu) \*

##### Repository

The steps are:

* Download and install the public key:

**wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add**

* Save the repository definition (for the regular package):

**echo "deb https://artifacts.elastic.co/packages/6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list**

* Or the one, for the package released under Apache 2.0 license:

**echo "deb https://artifacts.elastic.co/packages/oss-6.x/apt stable main" | sudo tee -a /etc/apt/sources.list.d/elastic-6.x.list**

* And then install the package:

**sudo apt-get update && sudo apt-get install elasticsearch**

##### Direct

Direct installation is quicker:

* Download the regular package:

**wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.4.deb**

* Or the one under Apache 2.0:

**wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-oss-6.5.4.deb**

* And install it:

**sudo dpkg -i elasticsearch\*.deb**

#### Post-installation Steps for Elasticsearch

Once that we have the software installed, we can do the following:

* We can adjust the IP address that Elasticsearch will be bound to. Open the file:

**sudo vi /etc/elasticsearch/elasticsearch.yml**

* And adjust the

**network.host: ["localhost", "192.168.99.101"]**

**http.port:9200**

* We could adjust the **cluster.name** and **node.name** parameters as well
* Reload the services

**sudo systemctl daemon-reload**

* Enable and start the service

**sudo systemctl enable elasticsearch**

**sudo systemctl start elasticsearch**

* If we want to examine the events in the system log for the service either all, or after a specific point of time:

**sudo journalctl --unit elasticsearch**

**sudo journalctl --unit elasticsearch --since "2016-10-30 18:17:16"**

* We can check if the service is responsive with:

**curl http://localhost:9200**

* If the firewall is on, either disable it, or open the appropriate port(s) if you plan to access the software from outside

#### Explore Elasticsearch

Exploration is always a good way to become familiar with something. There is REST API with which we can interact:

* Get cluster status:

**curl http://localhost:9200/\_cat/health?v**

* Get nodes information:

**curl http://localhost:9200/\_cat/nodes?v**

* List all indices:

**curl http://localhost:9200/\_cat/indices?v**

* Create an index:

**curl -X PUT "localhost:9200/customer?pretty"**

* List all indices:

**curl -X GET "localhost:9200/\_cat/indices?v"**

* Create simple document:

**curl -X PUT "localhost:9200/customer/\_doc/1?pretty" -H 'Content-Type: application/json' -d'**

**{**

**"name": "John Doe"**

**}'**

* Retrieve the document:

**curl -X GET "localhost:9200/customer/\_doc/1?"**

* Or with better formatted output:

**curl -X GET "localhost:9200/customer/\_doc/1?pretty"**

* Now delete the index:

**curl -X DELETE "localhost:9200/customer?pretty"**

* Ask again for the list of indices:

**curl -X GET "localhost:9200/\_cat/indices?v"**

#### Install Logstash (on CentOS)

Depending on which approach we followed during the Elasticsearch installation, we can do:

* If we have the repository added, then we need to execute just the following:

**sudo yum install logstash**

* If we decided to go with the package download, then again, we can repeat the procedure and choose between the standard package:

**wget https://artifacts.elastic.co/downloads/logstash/logstash-6.5.4.rpm**

* And the OSS one:

**wget https://artifacts.elastic.co/downloads/logstash/logstash-oss-6.5.4.rpm**

* Finally, we can install the package with:

**sudo rpm -Uvh logstash-\*.rpm**

#### Install Logstash (on Ubuntu) \*

Depending on which approach we followed during the Elasticsearch installation, we can do:

* If we have the repository added, then we need to execute just the following:

**sudo apt-get install logstash**

* If we decided to go with the package download, then again, we can repeat the procedure and choose between the standard package:

**wget https://artifacts.elastic.co/downloads/logstash/logstash-6.5.4.deb**

* And the OSS one:

**wget https://artifacts.elastic.co/downloads/logstash/logstash-oss-6.5.4.deb**

* Finally, we can install the package with:

**sudo dpkg -i logstash-\*.deb**

#### Explore Logstash

Go to **/usr/share/logstash/bin** and:

* First, execute:

**sudo ./logstash -e 'input { stdin { } } output { stdout {} }'**

* Press **Ctrl+C** to stop
* Okay, our service is working. Now let’s send messages to the **Elasticsearch** service:

**sudo ./logstash -e 'input { stdin {} } output { elasticsearch { hosts => ["localhost:9200"] } }'**

* The final step is to ask **Elasticsearch** for the entered data:

**curl -X GET "localhost:9200/logstash-\*/\_search"**

#### Post-installation Steps for Logstash

Now, that we have an idea how and if the **Logstash** is working, we can setup the service:

* Reload the services

**sudo systemctl daemon-reload**

* Enable and start the service

**sudo systemctl enable logstash**

**sudo systemctl start logstash**

#### Install Kibana (on CentOS)

Depending on which approach we followed during the **Elasticsearch** installation, we can do:

* If we have the repository added, then we need to execute just the following:

**sudo yum install kibana**

* If we decided to go with the package download, then again, we can repeat the procedure and choose between the standard package:

**wget https://artifacts.elastic.co/downloads/kibana/kibana-6.5.4-x86\_64.rpm**

* And the OSS one:

**wget https://artifacts.elastic.co/downloads/kibana /kibana-oss-6.5.4-x86\_64.rpm**

* Finally, we can install the package with:

**sudo rpm -Uvh kibana-\*.rpm**

#### Install Kibana (on Ubuntu) \*

Depending on which approach we followed during the **Elasticsearch** installation, we can do:

* If we have the repository added, then we need to execute just the following:

**sudo apt-get install kibana**

* If we decided to go with the package download, then again, we can repeat the procedure and choose between the standard package:

**wget https://artifacts.elastic.co/downloads/kibana/kibana-6.5.4-amd64.deb**

* And the OSS one:

**wget https://artifacts.elastic.co/downloads/kibana/kibana-oss-6.5.4-amd64.deb**

* Finally, we can install the package with:

**sudo dpkg -i kibana-\*.deb**

#### Post-Installation Steps for Kibana

Before we start our interaction with Kibana, we must take care for few additional steps:

* Open the main configuration file **/etc/kibana/kibana.yml** for editing
* Adjust the values of the following settings if needed:

**server.port: 5601**

**server.host: "localhost"**

**server.name: "your-hostname"**

**elasticsearch.url: "http://localhost:9200"**

* In our current setup – all services on one machine, we can leave them as they are – commented, or just adjust the **server.host** and set it to something like **server.host: 192.168.99.101**
* Reload the services

**sudo systemctl daemon-reload**

* Enable and start the service

**sudo systemctl enable kibana**

**sudo systemctl start kibana**

* If the firewall is on, either disable it, or open the appropriate port(s) if you plan to access the software from outside

#### Explore Kibana

The time has come. Let’s get to know Kibana:

* Open a browser tab and navigate to **http://192.168.99.101:5601**
* Let’s continue exploring Kibana, by adding a sample data and visualization set. Pick up the **Sample flight data** option
* Once installed, click on **View data**, and explore a bit
* If you want to add another sample set, click on the Kibana logo on the top left, then in the **section Add Data to Kibana** click on **Load a data set and a Kibana dashboard**, and choose one

We can also see our initial test message that we stashed during the Logstash exploration:

* Click on **Management** in the left section
* Then choose **Index Patterns** under **Kibana** section
* And then on **Create index pattern**
* Now in the **Index pattern field**, enter **logstash-\***
* Click on **Next step**
* Under **Time Filter** field name select **@timestamp**
* Click **Create index pattern**
* Now, we can examine the structure of our new index pattern, and mark it as default

Now, let’s explore the content of our index:

* Switch to the **Discover** section
* From the drop-down menu select **logstash-\***
* Examine the data collected for our event(s)

## Part 2: Beats

Extend the environment by substituting the **Vagrantfile** configuration files:

* Execute the following to add two more stations:

**cp Vagrantfile-2 Vagrantfile**

* And bring them up:

**vagrant up**

#### Heartbeat (on the server)

Depending on the way we choose to install the software, we have two options for the beats as well – repository or individual packages. We will continue following the latter approach and stick to the non OSS version:

* On **CentOS** machine we must do:

**wget https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh heartbeat-6.5.4-x86\_64.rpm**

* On **Ubuntu** machine we can install it with:

**wget https://artifacts.elastic.co/downloads/beats/heartbeat/heartbeat-6.5.4-amd64.deb**

**sudo dpkg -i heartbeat-6.5.4-amd64.deb**

Now that we have it installed, we must configure it:

* Open the main configuration file:

**sudo vi /etc/heartbeat/heartbeat.yml**

* Change the following and save:
  + Comment or delete the default monitor and create new one:

**- type: icmp**

**schedule: '\*/30 \* \* \* \* \* \*'**

**hosts: ["192.168.99.101", "192.168.99.102", "192.168.99.103"]**

* + Disable the Elasticsearch output and enable the Logstash output
* Test the configuration with:

**sudo heartbeat test config**

* Install the beat’s template in Elasticsearch:

**sudo heartbeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["localhost:9200"]'**

* Go **to /usr/share/logstash/bin** and list all installed plugins:

**logstash-plugin list**

* Our plugin is listed, so we can create a configuration file:

**sudo vi /etc/logstash/conf.d/beats.conf**

* Enter the following and save:

input {

beats {

port => 5044

}

}

output {

elasticsearch {

hosts => ["http://localhost:9200"]

index => "%{[@metadata][beat]}-%{[@metadata][version]}-%{+YYYY.MM.dd}"

}

}

* Restart Logstash service:

**sudo systemctl restart logstash**

* Start and enable the Heartbeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable heartbeat-elastic**

**sudo systemctl start heartbeat-elastic**

* Go to Kibana UI, add the new index pattern (**heartbeat-\***), and explore the messages

#### Metricbeat (on nodes)

Depending on the way we choose to install the software, we have two options for the beats as well – repository or individual packages. We will continue following the latter approach and stick to the non OSS version:

* On **CentOS** machine we must do:

**wget https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh metricbeat-6.5.4-x86\_64.rpm**

* On **Ubuntu** machine we can install it with:

**wget https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.5.4-amd64.deb**

**sudo dpkg -i metricbeat-6.5.4-amd64.deb**

Now that we have it installed, we must configure it:

* Open the main configuration file:

**sudo vi /etc/metricbeat/metricbeat.yml**

* Change the following and save:
  + Disable the Elasticsearch output and enable the Logstash output. Don’t forget to substitute the localhost with the IP of the Logstash server
* Test the configuration with:

**sudo metricbeat test config**

* To check what modules are available, you can either:

**ls -al /etc/metricbeat/modules.d**

* Or execute the following:

**sudo metricbeat modules list**

* Now, if we want to enable a module, we can execute:

**sudo metricbeat modules enable system**

* Install the beat’s template in Elasticsearch:

**sudo metricbeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["192.168.99.101:9200"]'**

* Start and enable the Metricbeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable metricbeat**

**sudo systemctl start metricbeat**

* Go to Kibana UI, add the new index pattern (**metricbeat-\***), and explore the messages

Repeat the same steps (without template installation) on node 2 and check the results in Kibana UI.

We can install NGINX for example on node 2, and enable the appropriate module:

* Go to node2
* Install NGINX:

**sudo apt-get update**

**sudo apt-get install nginx**

* Change the Metricbeat configuration

**sudo metricbeat modules enable nginx**

* Restart the service:

**sudo systemctl restart metricbeat**

* Return to Kibana UI and explore the results

#### Journalbeat (on nodes)

Depending on the way we choose to install the software, we have two options for the beats as well – repository or individual packages. We will continue following the latter approach and stick to the non OSS version:

* On **CentOS** machine we must do:

**wget https://artifacts.elastic.co/downloads/beats/journalbeat/journalbeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh journalbeat-6.5.4-x86\_64.rpm**

* On **Ubuntu** machine we can install it with:

**wget https://artifacts.elastic.co/downloads/beats/journalbeat/journalbeat-6.5.4-amd64.deb**

**sudo dpkg -i journalbeat-6.5.4-amd64.deb**

Now that we have it installed, we must configure it:

* Open the main configuration file:

**sudo vi /etc/journalbeat/journalbeat.yml**

* Change the following and save:
  + Disable the Elasticsearch output and enable the Logstash output. Don’t forget to substitute the localhost with the IP of the Logstash server
* Test the configuration with:

**sudo journalbeat test config**

* Install the beat’s template in Elasticsearch:

**sudo journalbeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["192.168.99.101:9200"]'**

* Start and enable the Journalbeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable journalbeat**

**sudo systemctl start journalbeat**

* Go to Kibana UI, add the new index pattern (**journalbeat-\***), and explore the messages

Repeat the same steps (without template installation) on node 2 and check the results in Kibana UI.

#### Filebeat (on nodes)

Depending on the way we choose to install the software, we have two options for the beats as well – repository or individual packages. We will continue following the latter approach and stick to the non OSS version:

* On **CentOS** machine we must do:

**wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh filebeat-6.5.4-x86\_64.rpm**

* On **Ubuntu** machine we can install it with:

**wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.5.4-amd64.deb**

**sudo dpkg -i filebeat-6.5.4-amd64.deb**

Now that we have it installed, we must configure it:

* Open the main configuration file:

**sudo vi /etc/filebeat/filebeat.yml**

* Change the following and save:
  + Disable the Elasticsearch output and enable the Logstash output. Don’t forget to substitute the localhost with the IP of the Logstash server
* Test the configuration with:

**sudo filebeat test config**

* To check what modules are available, you can either:

**ls -al /etc/filebeat/modules.d**

* Or execute the following:

**sudo filebeat modules list**

* We will continue with the default module configured
* Install the beat’s template in Elasticsearch:

**sudo filebeat setup --template -E output.logstash.enabled=false -E 'output.elasticsearch.hosts=["192.168.99.101:9200"]'**

* Start and enable the Filebeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable filebeat**

**sudo systemctl start filebeat**

* Go to Kibana UI, add the new index pattern (**filebeat-\***), and explore the messages

Repeat the same steps (without template installation) on node 2 and check the results in Kibana UI.

We can install Apache 2 for example on node 1, and enable the appropriate module:

* Go to node 1
* Install, enable, and start Apache 2:

**sudo yum install httpd**

**sudo systemctl enable httpd**

**sudo systemctl start httpd**

* Change the Filebeat configuration

**sudo filebeat modules enable apache2**

* Restart the service:

**sudo systemctl restart filebeat**

* Return to Kibana UI and explore the results

#### Create a Dashboard

Let’s create a dashboard. In order to do this, first we should create few Visualizations. First we will create a visualization of the average RAM usage:

* Go to Visualizations and clink on New
* Choose a Line Chart
* For Y-Axis set:
  + Aggregation to Average
  + Field to system.memory.actual.used.pct
  + Label to RAM % Used
* For X-Axis set:
  + Aggregation to Date Histogram
  + Field to @timestamp
* For Split Series
  + Sub Aggregation to Terms
  + Field to beat.hostname
* Click the Play button to see the result
* Click on Save and save it under RAM Usage

Create a second one for CPU Usage:

* Choose an Area Chart
* For Y-Axis set:
  + Aggregation to Average
  + Field to system.cpu.system.pct
  + Custom Label to CPU: System
* Add another Y-Axis with:
  + Aggregation to Average
  + Field to system.cpu.user.pct
  + Custom Label to CPU: User
* For X-Axis set:
  + Aggregation to Date Histogram
  + Field to @timestamp
* In Split Chart set:
  + Sub Aggregation to Terms
  + Field to beat.hostname
* Click the Play button to see the result
* Click on Save and save it under CPU Usage

Let’s create one more. This one will be showing the free desk space:

* Choose a Vertical Bar Chart
* For Y-Axis set:
  + Aggregation to Average
  + Field to system.fsstat.total\_size.free
  + Custom Label to Disk Free
* For X-Axis set:
  + Aggregation to Terms
  + Field to beat.hostname
  + Custom Label to Host
* Click the Play button to see the result
* Click on Save and save it under Disk Free

We are ready to create our first dashboard:

* Go to **Dashboard**
* Click on Create new dashboard
* Click on Add
* Select, position, and resize the three visualizations we created earlier
* Click Save and for name enter Resources

## Part 3: Docker and Elastic Stack

Prepare the environment by:

* Exchange the current **Vagrantfile** with the **Vagrantfile-3**

**cp Vagrantfile-3 Vagrantfile**

* And apply the changes:

**vagrant up**

#### Docker Metrics and Logs

Let’s install the appropriate beats on the docker host:

* On Docker node, download and install the package:

**wget https://artifacts.elastic.co/downloads/beats/filebeat/filebeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh filebeat-6.5.4-x86\_64.rpm**

* Open the main configuration file:

**sudo vi /etc/filebeat/filebeat.yml**

* Change the following and save:
  + Disable the Elasticsearch output and enable the Logstash output. Don’t forget to substitute the localhost with the IP of the Logstash server
  + Add the following:

**- type: docker**

**containers.ids:**

**- '\*'**

**processors:**

**- add\_docker\_metadata: ~**

* Start and enable the Filebeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable filebeat**

**sudo systemctl start filebeat**

* Start few containers
* Go to Kibana UI, add the new index pattern (**filebeat-\***), and explore the messages

Now let’s set to monitor the docker metrics:

* On Docker node, download and install:

**wget https://artifacts.elastic.co/downloads/beats/metricbeat/metricbeat-6.5.4-x86\_64.rpm**

**sudo rpm -Uvh metricbeat-6.5.4-x86\_64.rpm**

* Open the main configuration file:

**sudo vi /etc/metricbeat/metricbeat.yml**

* Change the following and save:
  + Disable the Elasticsearch output and enable the Logstash output. Don’t forget to substitute the localhost with the IP of the Logstash server
* Test the configuration with:

**sudo metricbeat test config**

* Enable the docker module:

**sudo metricbeat modules enable docker**

* Start and enable the Metricbeat service:

**sudo systemctl daemon-reload**

**sudo systemctl enable metricbeat**

**sudo systemctl start metricbeat**

* Go to Kibana UI, add the new index pattern (**metricbeat-\***), and explore the messages
* Run few containers, and check again the UI

#### Run ELK in Docker

The example is borrowed from: <https://github.com/aboullaite/docker-elk>

Either copy the **elk-on-docker.zip** to the docker host, or execute it on your host machine.

Ensure that the host where Docker is running meets the requirements:

* Check the parameter for system limits on mmap counts

**sudo sysctl vm.max\_map\_count**

* And set it to:

**sudo sysctl -w vm.max\_map\_count=262144**