Sending IBM App Connect Enterprise log messages to an ELK stack

SanjayNagchowdhury
Published on April 1, 2020 / Updated on April 1, 2020

0

Introduction

ELK is an acronym for three open source projects: Elasticsearch, Logstash, and Kibana. Beats has been added to the stack and it is now referred to as the Elastic Stack.

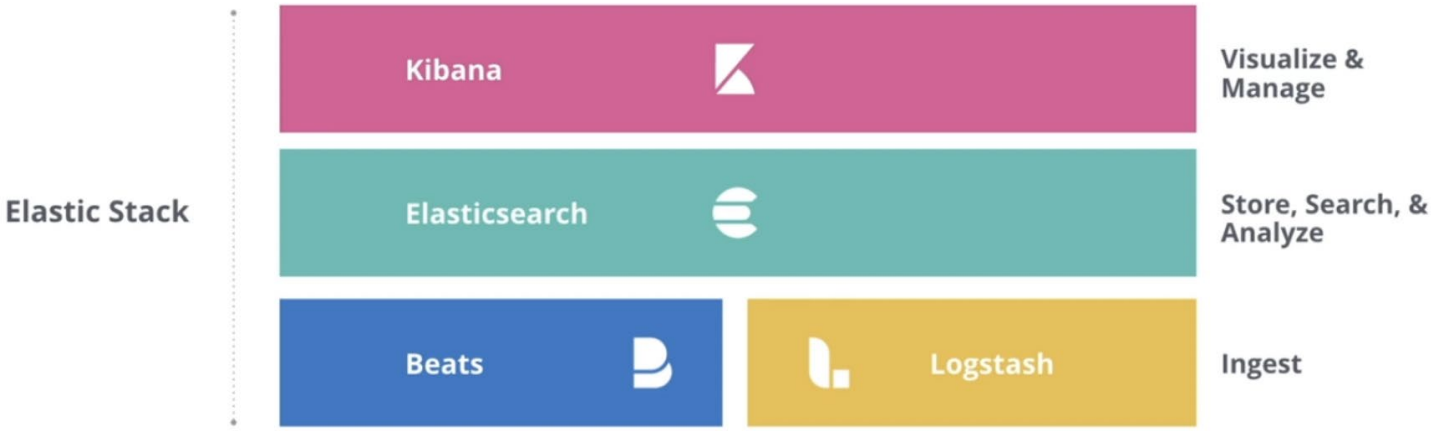
The Elastic Stack is the next evolution of the ELK Stack.

Elasticsearch: an open source search and analytics engine. It is able to achieve fast search responses because, instead of searching the text directly, it searches an index instead.

Logstash: a light-weight, open-source, server-side data processing pipeline. It can receive data from multiple sources simultaneously, transform it, and then send it to a specific destination. It is often used as a pipeline for Elasticsearch.

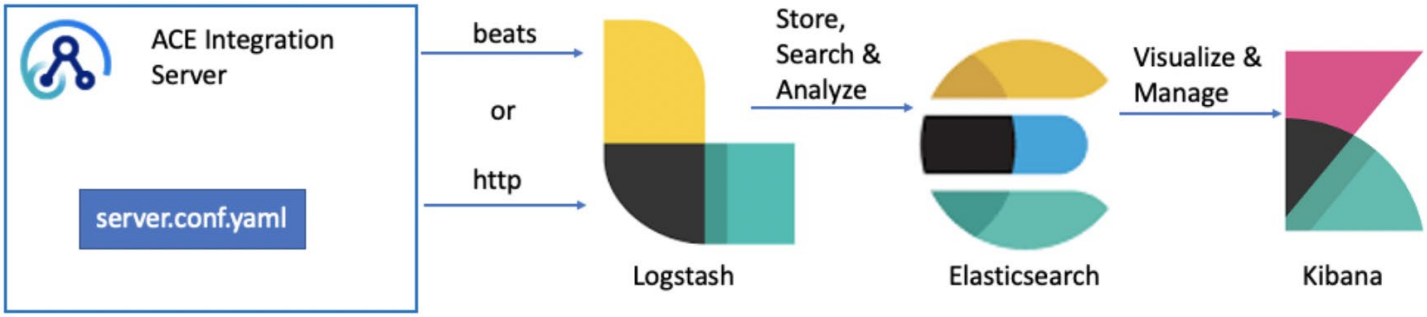
Kibana: an open-source data visualization and exploration tool. Kibana lets you visualize your Elasticsearch data. You can use it to build clear visualizations and dashboards. Kibana uses an index pattern to tell it which Elasticsearch indices to explore.

Beats: open source 'data shippers' which can be installed as agents on servers to send operational data to Elasticsearch. Beats can be used for capturing audit data, log files, cloud data, availability, metrics, network traffic and windows event logs. Beats can send data directly to Elasticsearch or via Logstash, where data can be further processed and enhanced, before it is visualized in Kibana.



Sending IBM App Connect Enterprise log messages to Elastic Stack

Capability has been added in ACE v11.0.0.8 which allows log messages to be sent to an Elastic Stack. Simple configuration can be done in server.conf.yaml (for Independent Integration Servers) or node.conf.yaml (for Integration Nodes and node-owned Integration Servers) to configure ACE to send log messages to a Logstash server in the Elastic Stack. The log messages can be sent using the beats or http protocols. Transport level security can be applied for both protocols.



Configuring Logstash

ACE can send log messages to the Logstash server using beats or http. The Logstash server must be configured to receive the log messages using http or beats. A config file is used by the Logstash server. For more information about configuring Logstash, see https://www.elastic.co/guide/en/logstash/current/configuration.html

For this article, I have used the following logstash.conf file. For simplicity I have copied the file to /tmp/logstash.conf

```
# Logstash configuration file

# Log messages can be received using http on port 5888
# or
# Log messages can be received using beats on port 5444

input {
   http {
    port => 5888
    codec => json
   }
   beats {
    port => 5444
   }
}

# Data is sent to Elasticsearch to port 9200

output {
  elasticsearch { hosts => ["elasticsearch:9200"] }
}
```

Configuring Elastic Stack using Docker Containers

The following was carried out to start the three servers in separate Docker Containers in a Docker network.

Each of the open source projects are available to download using Docker. Below are links to the official Docker images on dockerhub provided by Elastic:

Dockerhub Links
Elasticsearch: https://hub.docker.com/_/elasticsearch
Kibana: https://hub.docker.com/_/kibana?tab=description
Logstash: https://hub.docker.com/_/logstash?tab=description

At time of writing this article, the latest level of the open source projects was 7.6.1.

Pull down the docker images

First, pull down each docker image using the following commands:

```
docker pull elasticsearch:7.6.1

docker pull kibana:7.6.1

docker pull logstash:7.6.1
```

You can list the docker images using:

```
docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
logstash            7.6.1               d6d66afe6805        4 weeks ago         813MB
kibana              7.6.1               f9ca33465ce3        4 weeks ago         1.01GB
elasticsearch       7.6.1               41072cdeebc5        4 weeks ago         790MB
```

Create a Docker network and start the containers

For this article, I am using a docker network. There are other ways of configuring the Elastic Stack by using Docker Compose for example, or install the native open source projects.

To create a Docker network and run three separate containers using the same network, you can run the following commands. I am running each of these commands in separate terminal windows.

Create the Docker network.

```
docker network create elk-network
```

Start the Elasticsearch container in the docker network.

```
docker run -p 9200:9200 -p 9300:9300 -e "cluster.initial_master_nodes=elasticsearch" -h elasticsearch --name elasticsearch --net=elk-network elasticsearch:7.6.1
```

Confirm that Elasticsearch has started, by entering this curl command

```
curl http://localhost:9200
```

and check that you receive this response:

```
{
  "name" : "elasticsearch",
  "cluster_name" : "docker-cluster",
  "cluster_uuid" : "Sp9TXD0CRDyxhvPQtopmlg",
  "version" : {
    "number" : "7.6.1",
    "build_flavor" : "default",
    "build_type" : "docker",
    "build_hash" : "aa751e09be0a5072e8570670309b1f12348f023b",
    "build_date" : "2020-02-29T00:15:25.529771Z",
    "build_snapshot" : false,
    "lucene_version" : "8.4.0",
    "minimum_wire_compatibility_version" : "6.8.0",
    "minimum_index_compatibility_version" : "6.0.0-beta1"
  },
  "tagline" : "You Know, for Search"
}
```

Start the Kibana container in the docker network.

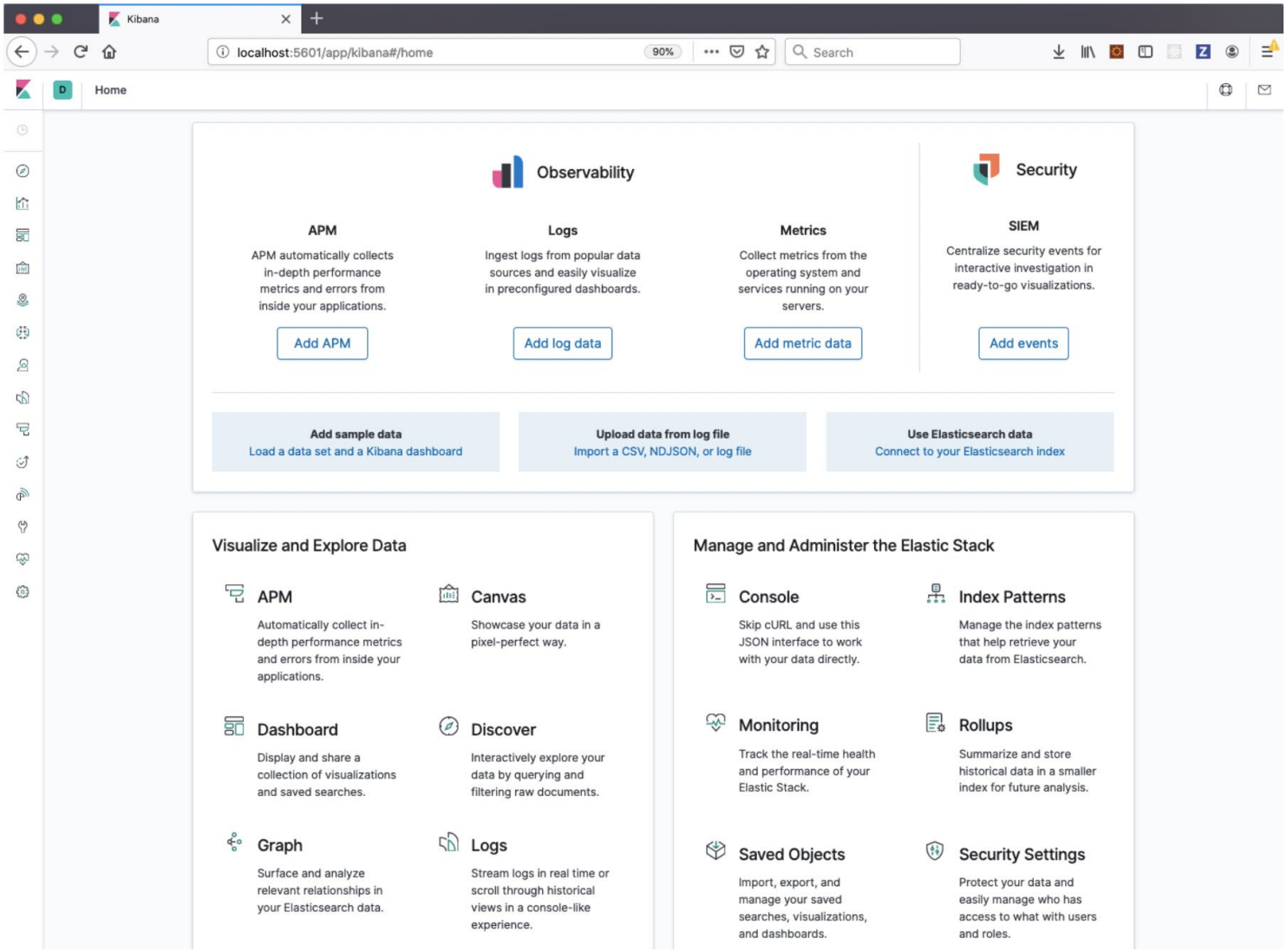
```
docker run -p 5601:5601 -h kibana --name kibana --net=elk-network kibana:7.6.1
```

When the Kibana has finished starting up, you should see this output:

```
{"type":"log","@timestamp":"2020-03-31T11:03:58Z","tags":["listening","info"],"pid":6,"message":"Server running at http://0:5601"}
{"type":"log","@timestamp":"2020-03-31T11:03:58Z","tags":["info","http","server","Kibana"],"pid":6,"message":"http server running at http://0:5601"}
```

After the Kibana server has started, you can open the WebUI for Kibana on http://localhost:5601

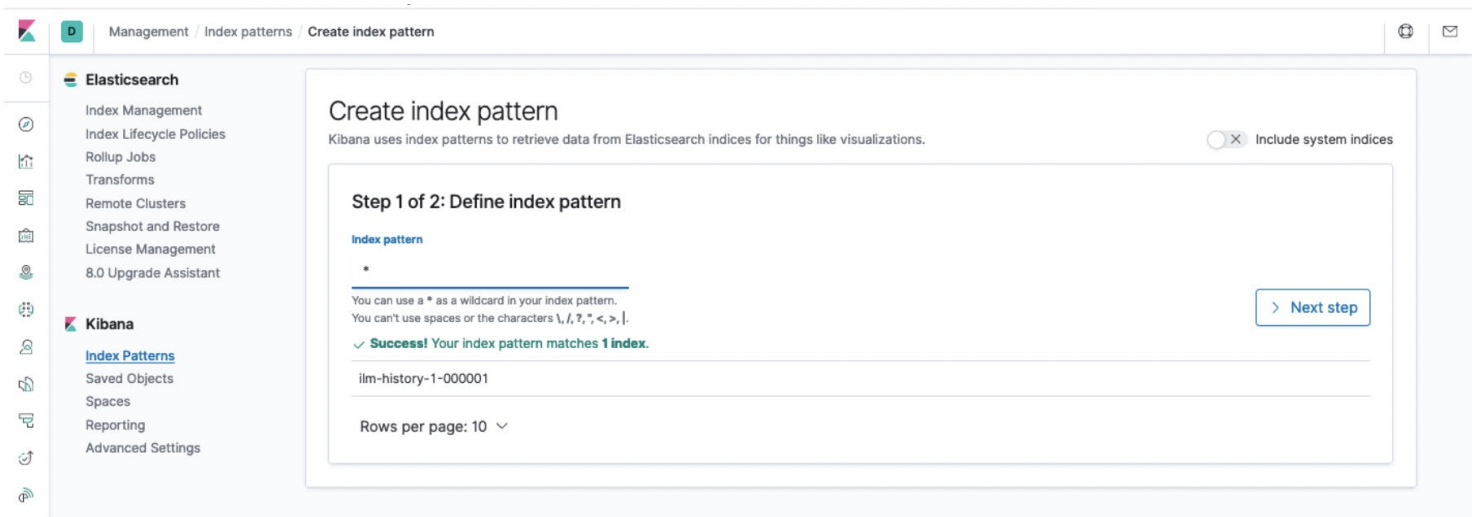
You should see the home page for Kibana:



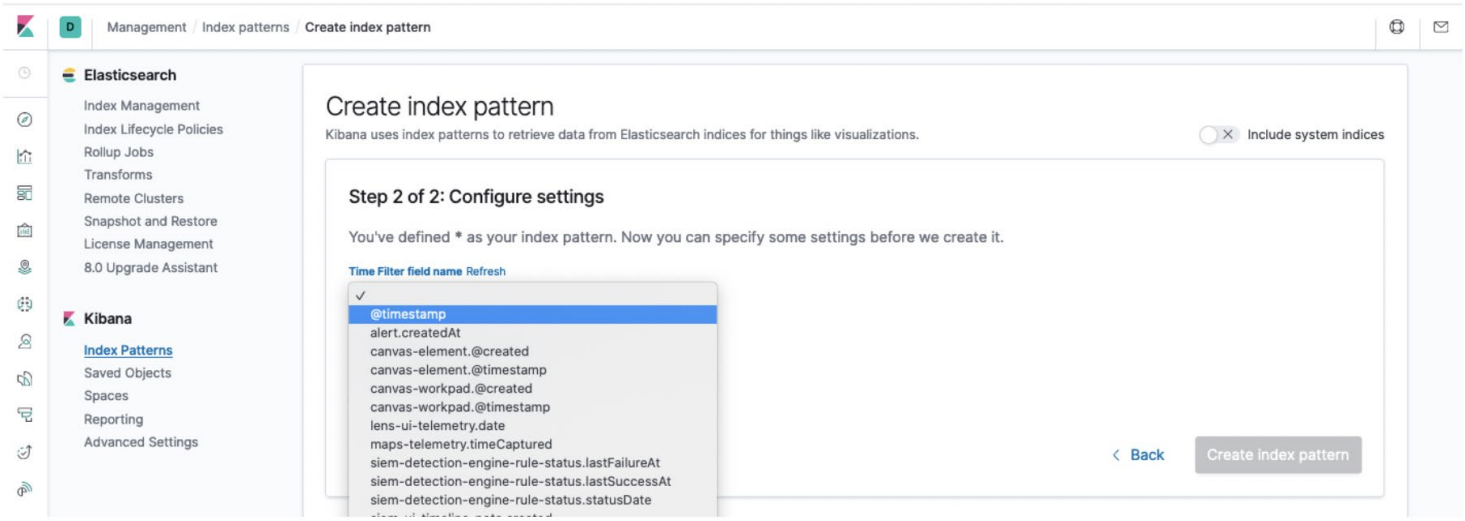
While in the WebUI, create an index pattern, so that you can see the log messages that are being sent from the ACE Integration Server.

Click on Discover

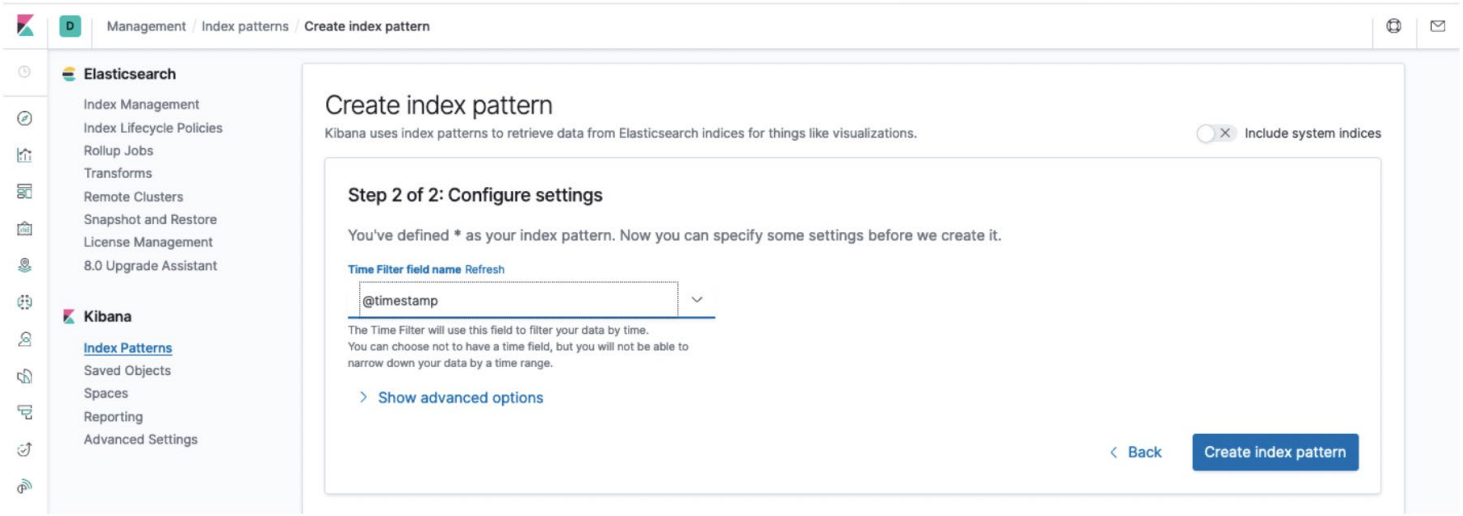
Enter '*' as the index pattern



Select @timestamp in the Time Filter field.



Click on Create index pattern



Start the Logstash container in the docker network.

Ensure that you have the configuration file /tmp/logstash.conf described earlier.

```
ls -l /tmp/logstash.conf
-rw-r--r--  1 sanjayn  wheel  348 31 Mar 11:51 /tmp/logstash.conf
```

Start the container, referencing the logstash.conf in /tmp and the docker network.

```
docker run -p 5444:5444 -p 5888:5888 -h logstash --name logstash --net=elk-network --rm -v /tmp:/config-dir logstash:7.6.1 -f /config-dir/logstash.conf
```

When it has finished starting, you should see a message like this:

```
[2020-03-31T11:37:21,069][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600}
```

You can confirm that data that is sent to logstash on http can be visualized in Kibana by doing the following.

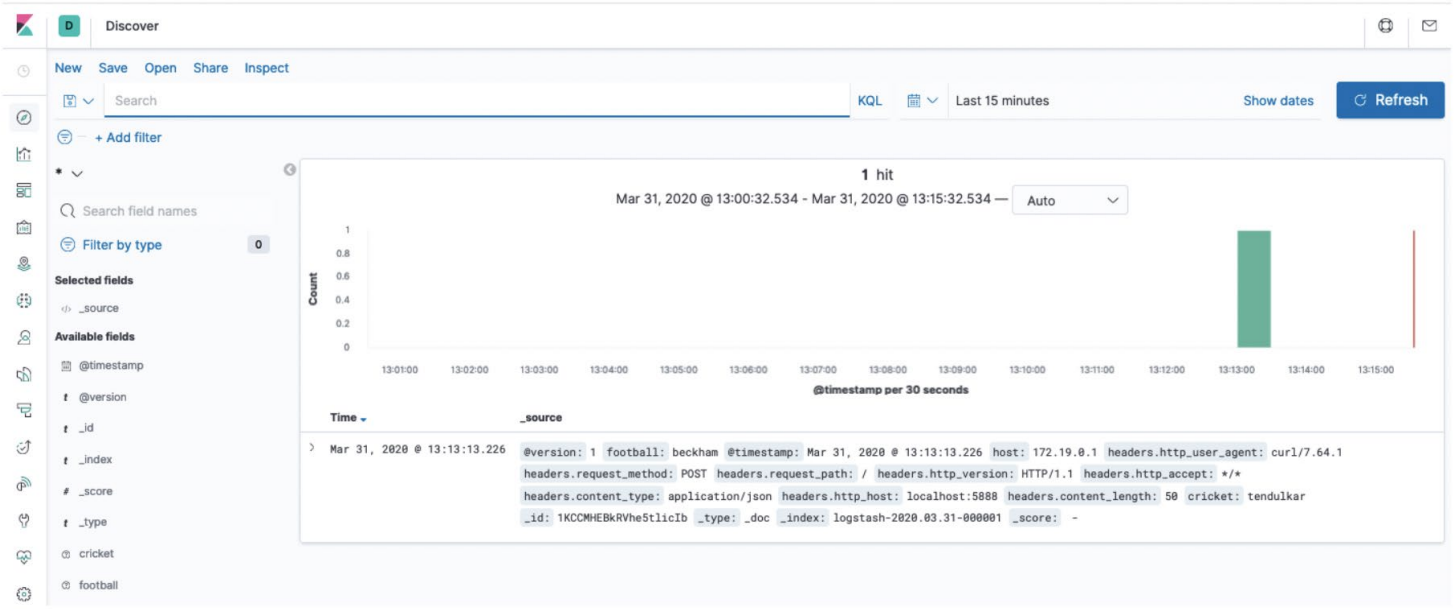
Send a curl request like this to port 5888:

```
curl -X POST --header 'Content-Type:application/json'  --data '{ "cricket" : "tendulkar", "football" : "beckham"}' http://localhost:5888 -i
```

You will receive a response like this:

HTTP/1.1 200 OK
content-length: 2
content-type: text/plain

You can check that the data has been sent to Elasticsearch and visualized by Kibana by clicking on Discover in the Kibana WebUI. You will see the page has a log entry:

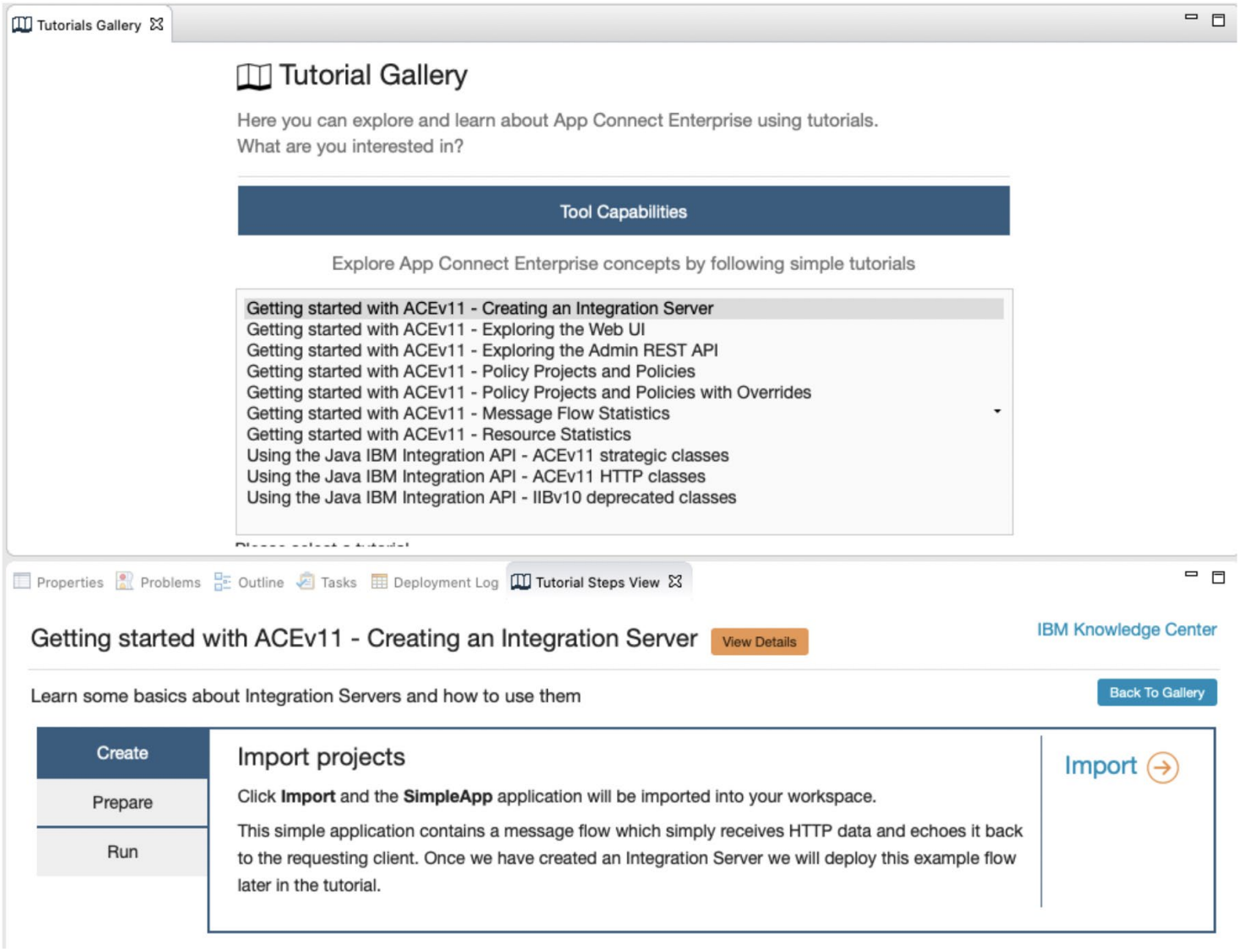


Configure ACE to send log messages using http

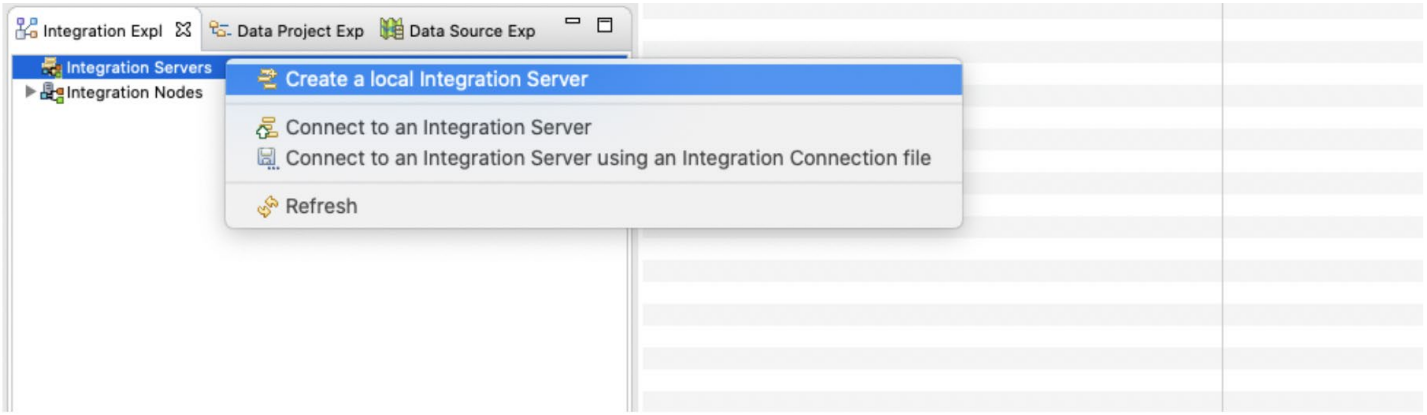
Now that you have the Elastic Stack running in Docker containers and proved that messages can be sent to Logstash, parsed by Elasticsearch and visualized in Kibana, you can configure ACE to send log messages to the Elastic Stack.

Launch the ACE toolkit.

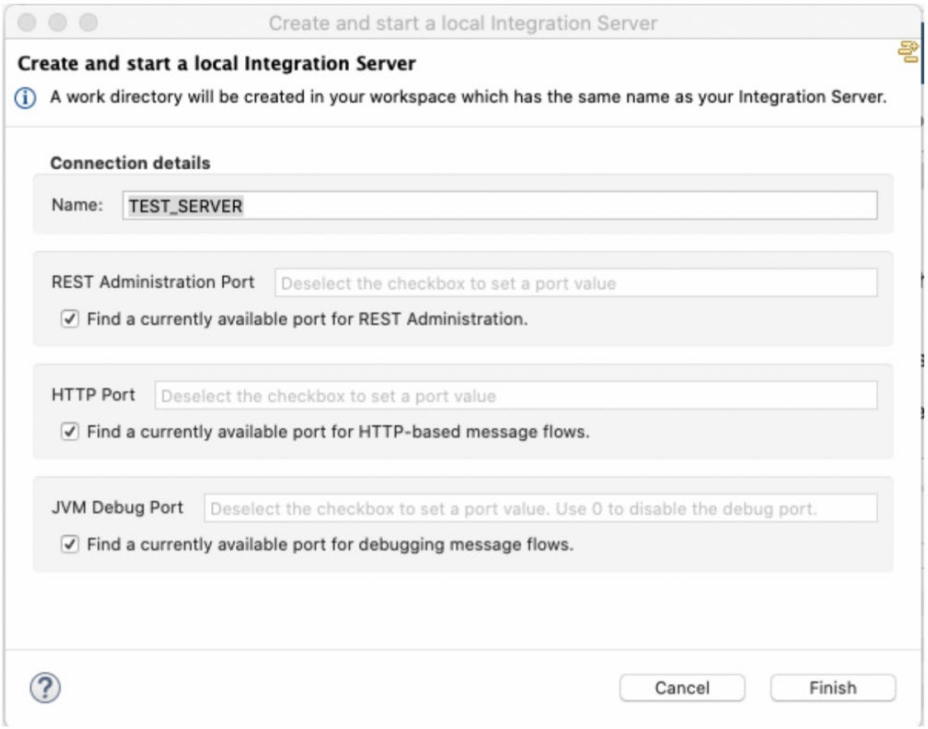
Import the ‘SimpleApp’ Application using the tutorial ‘Getting started with ACEv11 – Creating an Integration Server’ from the Tutorial Gallery.



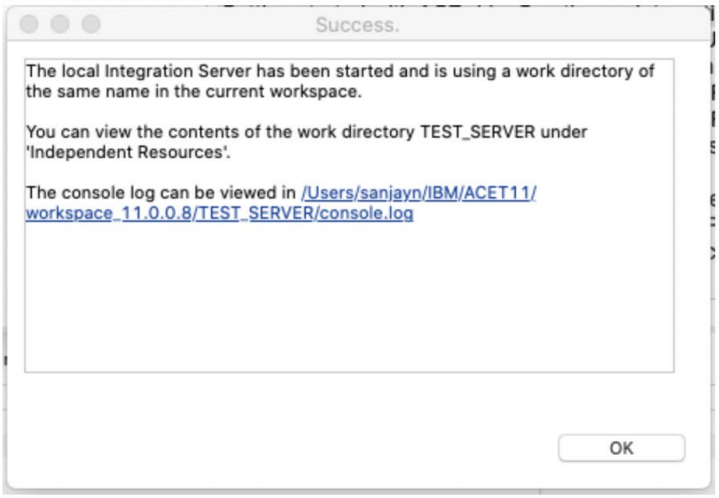
Create a local Integration Server.

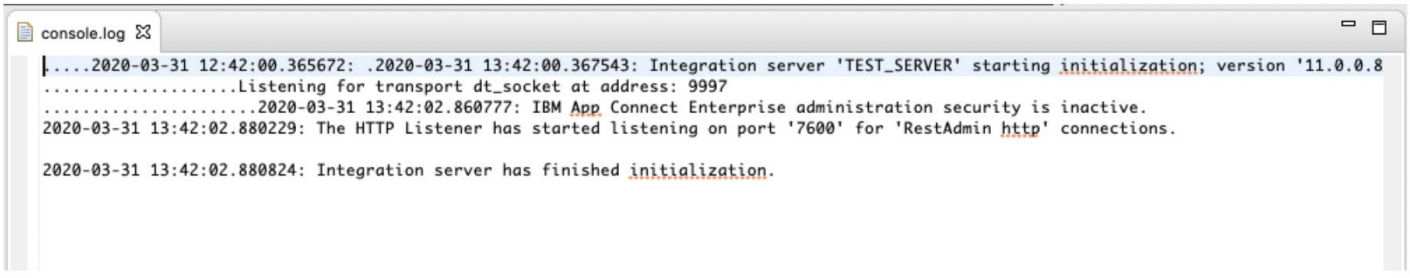


Leave the default settings and click Finish.



The server will be started and the console log can be shown by clicking on the link at the end of the wizard.





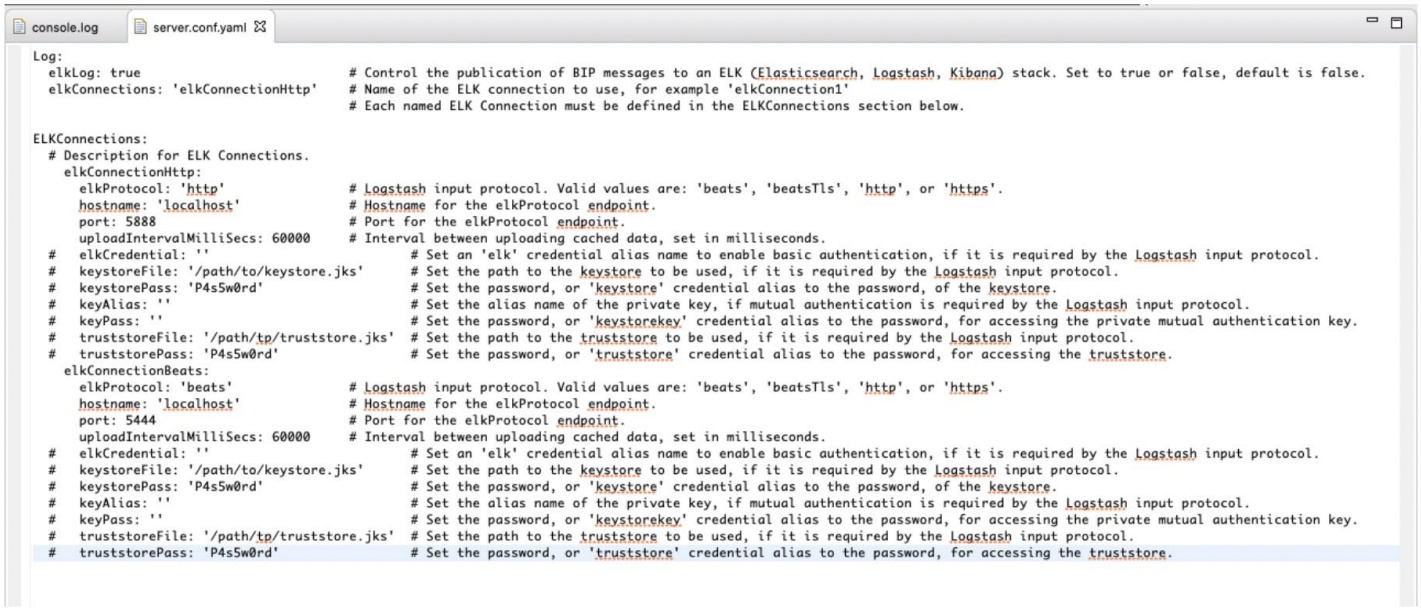
Configure overrides/server.conf.yaml to add the following lines to the bottom of the file:

```
Log:
  elkLog: true
  elkConnections: 'elkConnectionHttp'

ELKConnections:
  # Description for ELK Connections.
    elkConnectionHttp:
```

```
    elkProtocol: 'http'
    hostname: 'localhost'
    port: 5888
    uploadIntervalMilliSecs: 60000
  elkConnectionBeats:
    elkProtocol: 'beats'
    hostname: 'localhost'
    port: 5444
    uploadIntervalMilliSecs: 60000
```

After adding the lines, my overrides/server.conf.yaml looked like this:



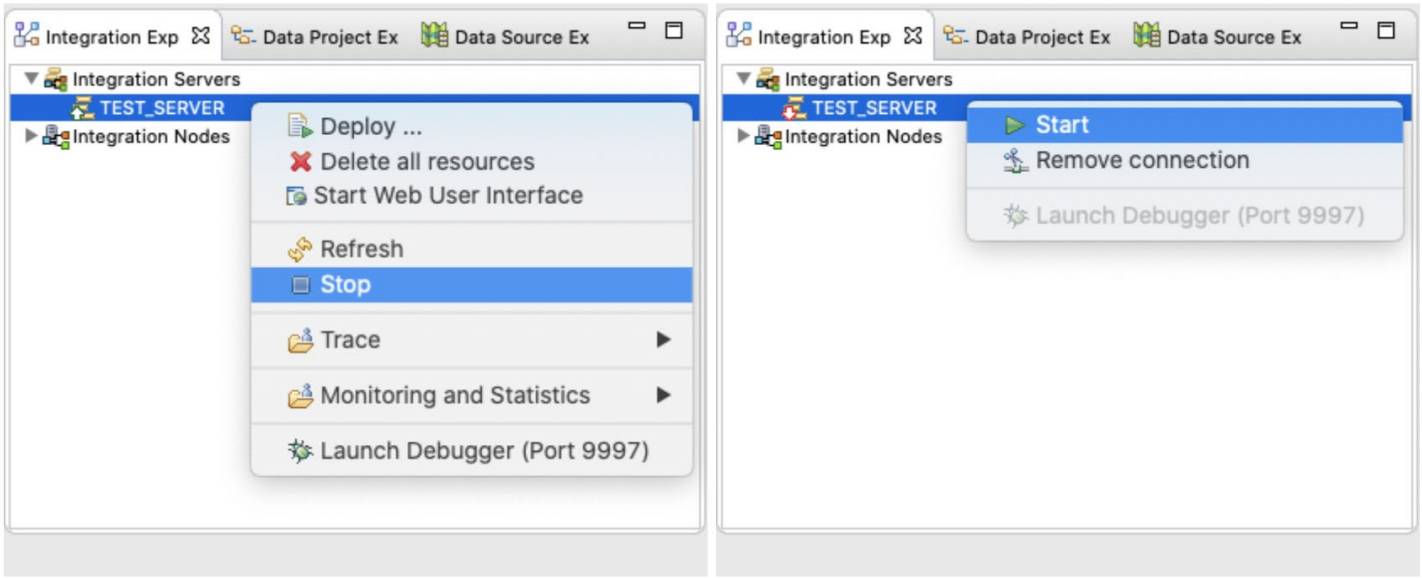
You will see that in the Log: stanza, there is an option, elkLog, whose value determines whether to send log messages to an ELK stack or not. By default it is false. When set to true, it will use the elk connection that is defined.

There is a new stanza in server.conf.yaml where you can define ELKConnections. In this article, I am not configuring TLS for http or beats. I describe how to send messages to ELK with TLS mutual authentication in this article.

I have two connections defined for sending log messages to ELK using http and beats. Only one connection can be used by an Integration Server at a time.

In this example the elkConnectionHttp is being used which is instructing ACE to send the log messages to port 5888. Logstash has already been configured to receive http data on port 5888 (check the values in /tmp/logstash.conf that was shown earlier on).

Save the file and stop and start the Integration Server, so that changes can take effect.



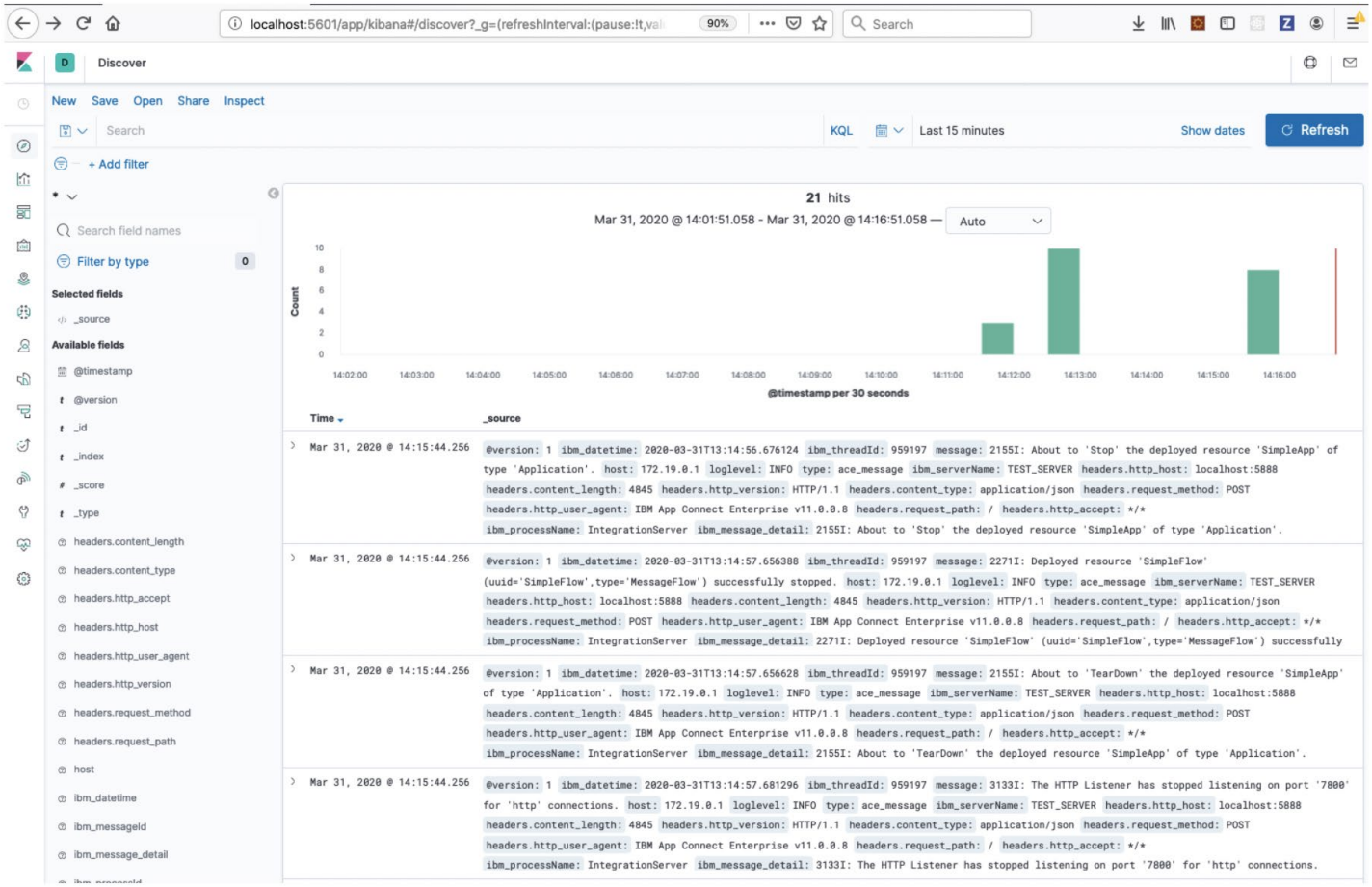
Deploy the SimpleApp to the TEST_SERVER Integration Server.

You will see this confirmation message appear in console.log, confirming that data was sent to the ELK stack.

```
The integration server successfully sent data to ELK connection 'elkConnectionHttp' using elkProtocol 'http', hostname 'loca
lhost' and port '5888'.
```

Stop the Application, and then start the application that is deployed to the TEST_SERVER Integration Server.

Refresh the Discover page in the Kibana WebUI. You will see entries in there for the ACE log messages that have been generated when stopping and starting the Application.



Congratulations – you have just set up ACE to send log messages using http to the Elastic Stack!

Configure ACE to send log messages using beats

To send the log messages to the Logstash server using beats, change the value of elkConnections to ‘elkConnectionBeats’ in overrides/server.conf.yaml.

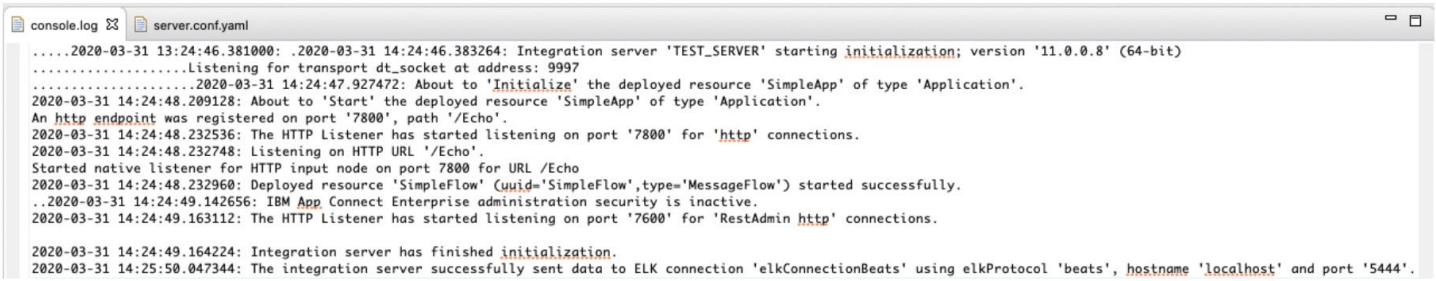
```
Log:
  elkLog: true
  elkConnections: 'elkConnectionBeats'

ELKConnections:
  # Description for ELK Connections.
    elkConnectionHttp:
      elkProtocol: 'http'
      hostname: 'localhost'
      port: 5888
      uploadIntervalMilliSecs: 60000
    elkConnectionBeats:
      elkProtocol: 'beats'
      hostname: 'localhost'
      port: 5444
      uploadIntervalMilliSecs: 60000
```

Save the file and stop and start the TEST_SERVER Integration Server.

You will see this message, confirming that data is now being sent to the Elastic Stack using the beats protocol.

```
The integration server successfully sent data to ELK connection 'elkConnectionBeats' using elkProtocol 'beats', hostname 'lo
calhost' and port '5444'.
```



Stop and start the Application that is deployed to the TEST_SERVER Integration Server.

Refresh the Discover page in the Kibana WebUI. You will see entries in there for the ACE log messages that have been generated when stopping and starting the Application.

Congratulations – you have just set up ACE to send log messages using beats to the Elastic Stack!

For further information on this capability, please see the Knowledge Center at this page https://www.ibm.com/support/knowledgecenter/SSTTDS_11.0.0/com.ibm.etools.mft.doc/bz91195_.html

Look out for the next article Sending ACE log messages to an ELK stack using Basic Auth and TLS mutual authentication which will describe how to send the log messages using TLS mutual authentication.

TAGS ACE, ELK STACK, ELASTICSEARCH, DOCKER NETWORK, DOCKER CONTAINERS

SanjayNagchowdhury