Docker Image for

Alfresco Site Activities Reporting

on ELK

Contents

[1. Project Objective 3](#_Toc478651556)

[2. Install Docker on Host machine 3](#_Toc478651557)

[3. Virtual Memory 3](#_Toc478651558)

[4. Download “Docker-Alfresco-Site-Activities” image 3](#_Toc478651559)

[5. Download “Docker-Alfresco-Site-Activities” container software 3](#_Toc478651560)

[6. Creating the Docker Container 4](#_Toc478651561)

[7. Starting the Docker Container 4](#_Toc478651562)

[8. Configuring Kibana UI 5](#_Toc478651563)

[9. Accessing the Dashboard 5](#_Toc478651564)

[10. Accessing the container 6](#_Toc478651565)

[11. Stopping the container 6](#_Toc478651566)

[12. Removing the container 6](#_Toc478651567)

[13. How we collect Site Activities data 6](#_Toc478651568)

# Project Objective

The aim of this Docker image is to provide a full ELK (Elasticsearch, Logstash and Kibana) environment with the necessary configuration and scripts to collect and present data from Alfresco Site Activities.

# Install Docker on Host machine

Install Docker on your host machine i.e. laptop as per Docker website. Please note the Docker Community Edition is sufficient to run this project (<https://www.docker.com/community-edition>)

# Virtual Memory

Elasticsearch uses a hybrid mmapfs / niofs directory by default to store its indices. The default operating system limits on mmap counts is likely to be too low, which may result in out of memory exceptions. On Linux, you can increase the limits by running the following command as root on the host machine:

# sysctl -w vm.max\_map\_count=262144

To set this value permanently, update the vm.max\_map\_count setting in /etc/sysctl.conf. To verify the value has been applied run sysctl vm.max\_map\_count.

# Download “Docker-Alfresco-Site-Activities” image

The “Docker-Alfresco-Site-Activities” image can be downloaded to the host machine with the following command:

# docker pull miguel220369/docker-elk-alfresco-site-activities

Verify the image has been downloaded:

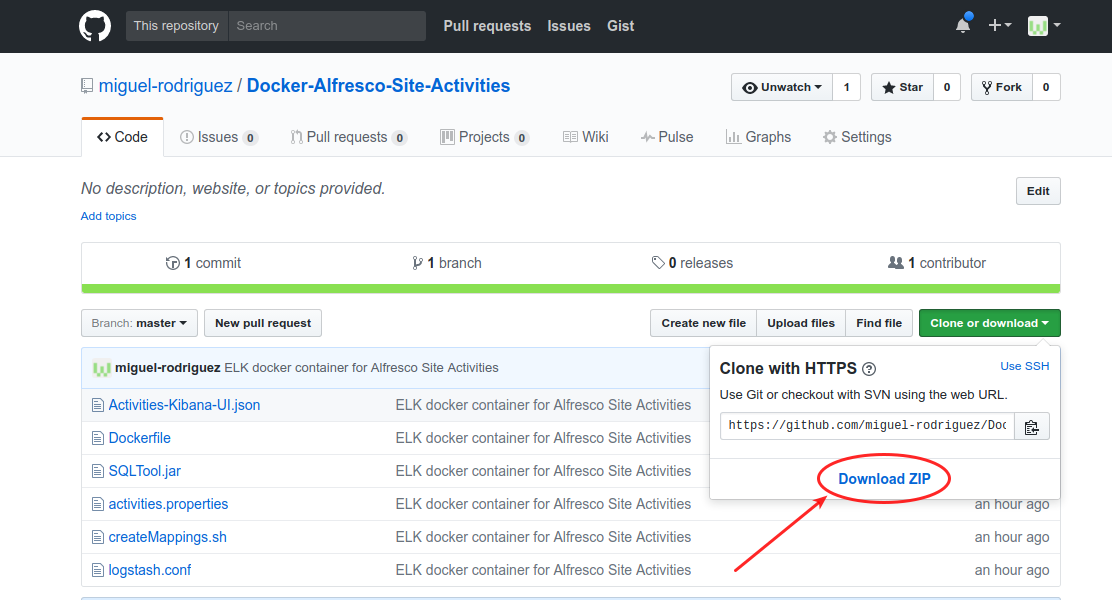
# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

miguel220369/docker-elk-alfresco-site-activities latest 8cbbc0a53592 2 hours ago 981 MB

# Download “Docker-Alfresco-Site-Activities” container software

Download the software to create the Docker container from GitHub: <https://github.com/miguel-rodriguez/Docker-Alfresco-Site-Activities> and extract the files to the file system.



# Creating the Docker Container

Before creating the container we need to configure access to Alfresco database from the Docker container. Assuming the files have been extracted to /opt/docker-projects/Docker-Alfresco-Site-Activities-master, edit file activities.properties and set the access to the DB server as appropriate, for example:

#postgresql settings

db\_type=postgresql

db\_url=jdbc:postgresql://172.17.0.1:5432/alfresco

db\_user=alfresco

db\_password=admin

**Note:** make sure the DB server allows for remote access to “alfresco” database.

We also need to set the timezone in the container, this can be done by editing the following entry in the startELK.sh script

export TZ=GB

From the command line execute the following command to create the Docker container:

# docker create -p 5601:5601 -v /opt/docker-projects/Docker-Alfresco-Site-Activities-master/startELK.sh:/usr/local/bin/startELK.sh/:rw -v /opt/docker-projects/Docker-Alfresco-Site-Activities-master/activities.properties:/tmp/activities.properties/:rw -it --name elk miguel220369/docker-elk-alfresco-site-activities

Unable to find image 'miguel220369/docker-elk-alfresco-site-activities:latest' locally

latest: Pulling from miguel220369/docker-elk-alfresco-site-activities

…..

# Starting the Docker Container

Once the Docker container has been created it can be started with the following command:

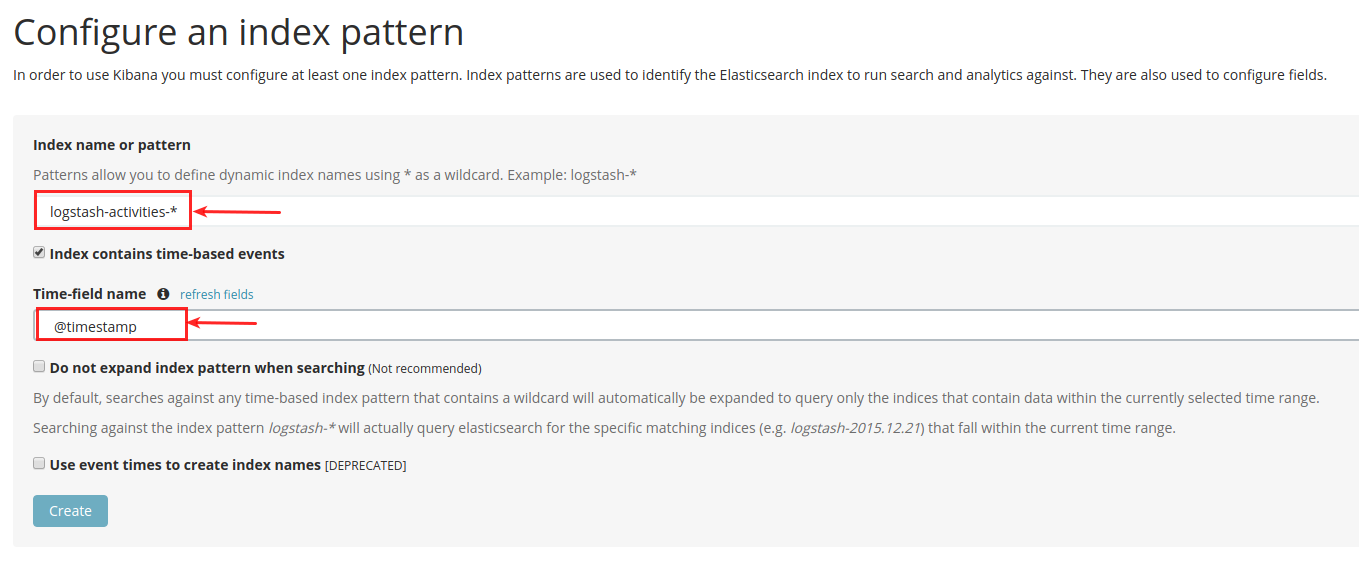
# docker start elk

Verify the ELK stack is running by accessing Kibana on <http://localhost:5601> on the host machine.

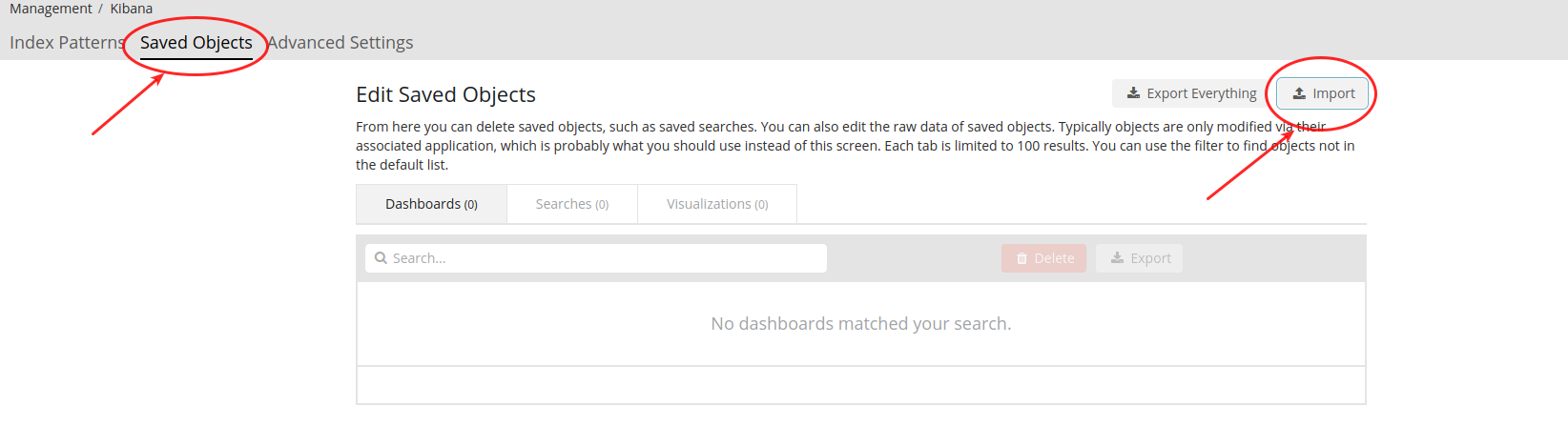
**Note:** At this point Elasticsearch and Kibana do not have any data…so we need to get Alfresco up and running to produce some content before creating the dashboards in Kibana.

# Configuring Kibana UI

Configure the index pattern for activities entries by entering “logstash-activities-\*” in the index name or pattern box and selecting “@timestamp” as the time-field name as shown below, then click on “Create” to finish.

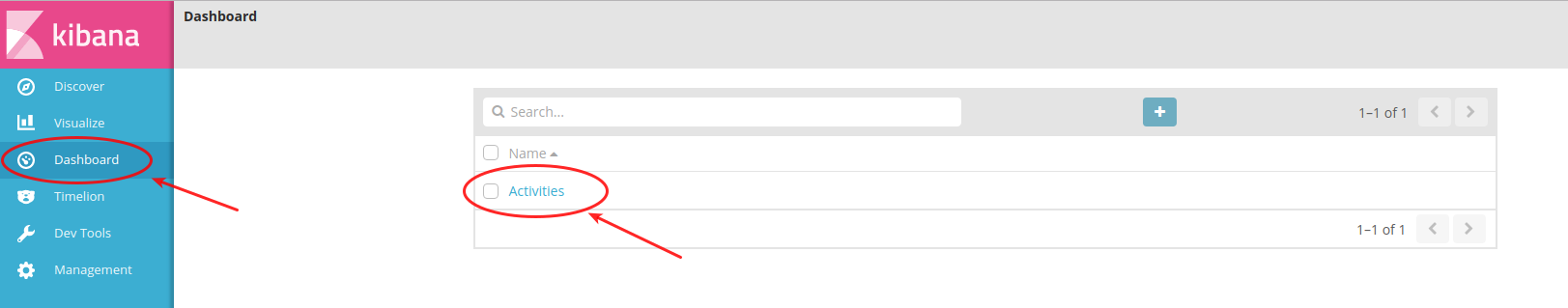


Next add the Dashboard to Kibana by clicking on “Saved Objects” link and then in “Import” link and select the file “Activities-Kibana-UI.json” file from the set of files downloaded from GitHub.

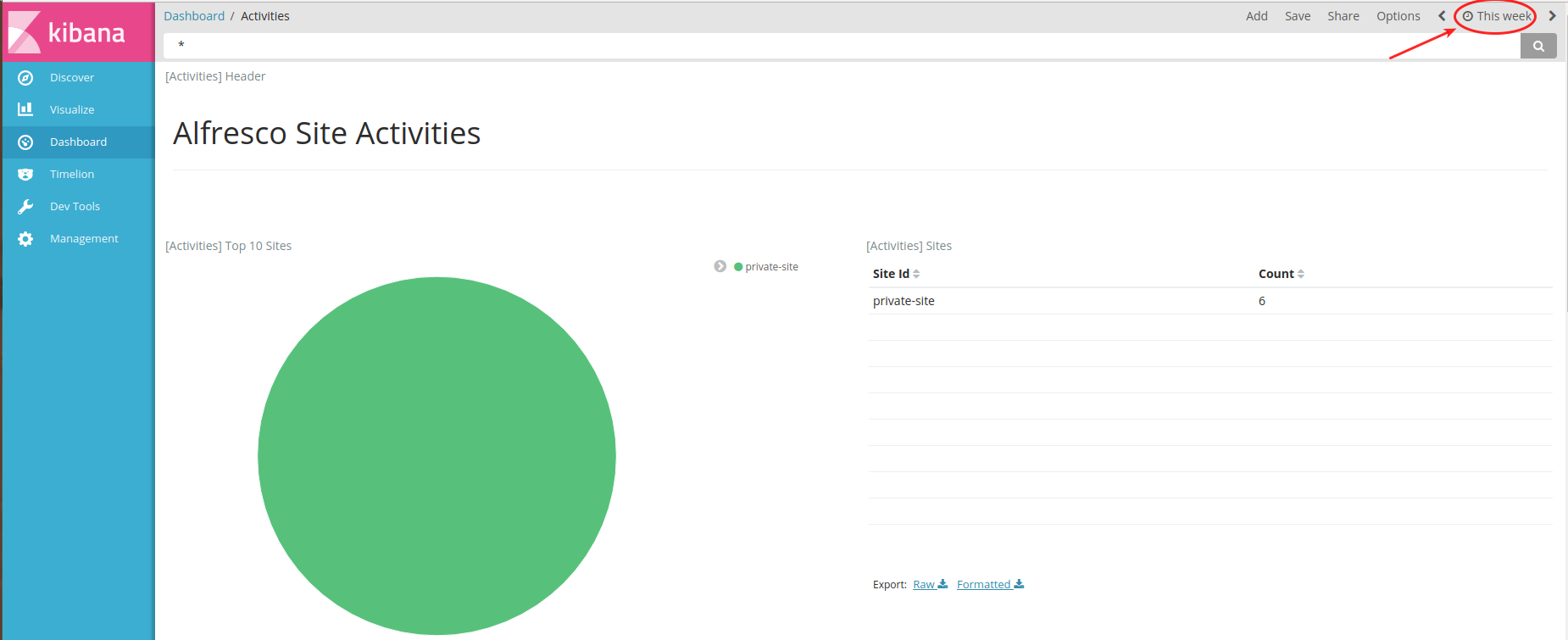


# Accessing the Dashboard

Finally access the dashboard by clicking on the “Dashboard” link on the left panel and then click on the “Activities” link.



The data should be available for the selected time period.



# Accessing the container

To enter the container itself (running image) use the following command:

# docker exec -i -t elk bash

And to exit the container just type “exit” and you will find yourself back on the host machine.

# Stopping the container

To stop the container from running type the following command on the host machine:

# docker stop elk

# Removing the container

To delete the container you first need to stop the container and then run the following command:

# docker rm elk

# How we collect Site Activities data

Site Activities are collected directly from the database so we need direct access from the Docker container to the database.

Please make sure the database server allows for remote connections to Alfresco’s database. A couple of examples how to configure the database are shown here:

1. For MySQL

Access your database server as an administrator and grant the correct permissions i.e.

# mysql -u root -p

* grant all privileges on alfresco.\* to alfresco@'%' identified by 'admin';

The grant command is granting access to all tables in ‘alfresco’ database to ‘alfresco’ user from any host using ‘admin’ password.

1. For PostgreSQL

Change the file ‘postgresql.conf’ to listen on all interfaces

listen\_addresses = '\*'

then add an entry in file ‘pg\_hba.conf’ to allow connections from any host

host all all 0.0.0.0/0 trust

Restart PostgreSQL database server to pick up the changes.

We have installed a small java application inside the container in /opt/activities folder that executes calls against the database configured in /opt/activities/activities.properties file.

For example to connect to PostgreSQL we have the following settings:

db\_type=postgresql

db\_url=jdbc:postgresql://172.17.0.1:5432/alfresco

db\_user=alfresco

db\_password=admin

You can adjust these settings if necessary. The ’SQLTool.jar’ application is called every 60 seconds by logstash as defined in /etc/logstash/conf.d/logstash.conf file.

#################

# Collect Activities #

#################

input {

exec {

command => 'cd /opt/activities&&java -jar /opt/activities/SQLTool.jar activi

ties'

interval => 30

}

exec {

command => 'find /data/logs/activities/activities-\*log -maxdepth 1 -type f -

mtime +3 -exec rm {} \;'

interval => 3600

}

}