Share Your Terminal Output Using this Secure Application

The authors describe in detail how to build an application that shares the terminal output on the Internet, and can also search the files that are shared.

Most educational institutes in India have mandatory projects for students as part of their courses. Working together for a project generally involves sharing the terminal outputs, which can be cumbersome for the teams collaborating with each other. The Web application described in this article shares the terminal output on the Internet and can also search the files that are shared. This application can be used by students and teachers, as well as by engineers working with IoT devices. It has been designed such that it can help share any kind of file format.

The motivation behind the project described in this article was to be able to incorporate all these tasks so that it could support many console output systems, ranging from education to IoT. Its GUI features ensure the system can be used without any technical knowledge.

The name of this application is 'Real_time_sharing'. It is Web application software, which helps to share the terminal output. By registering in the application through an email ID, the student is allowed to login and share the output files to the database using MongoDB. And by logging in, the faculty can access all the files shared by every student, evaluate the work based on its novelty, and then upload the evaluated files.

The developed application is platform-independent. It supports durability, monitoring and a secure connection with ssh enabled databases, i.e., MongoDB. We have provided the login system for the authentication of appropriate users.

The workflow of the application is depicted in Figure 1. The source code for this project is available at *https://github.com/cmouli96/real_time_sharing*. The setup is as follows.

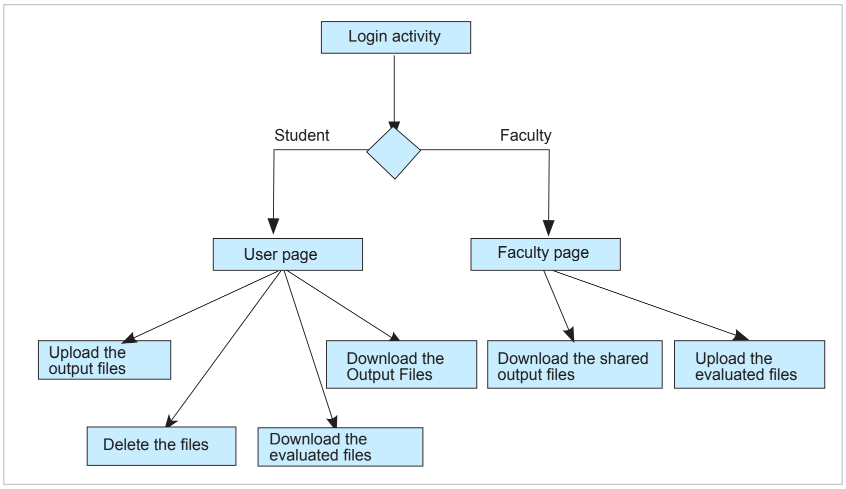

Login activity
Student
Faculty
User page
Faculty page
Upload the output files
Download the Output Files
Download the shared output files
Upload the evaluated files
Delete the files
Download the evaluated files

Figure 1: Workflow of the application

System configuration

Ubuntu version 18.04LTS, 8GB RAM and 4 core CPU

Features

- Users can upload output files from the terminal in any format up to any size.
- The guidelines for the file creation are given.
- Users can download and delete the files from the database.
- Users can search the files from the database.
- Async/await mechanism is used to create a promise chain.

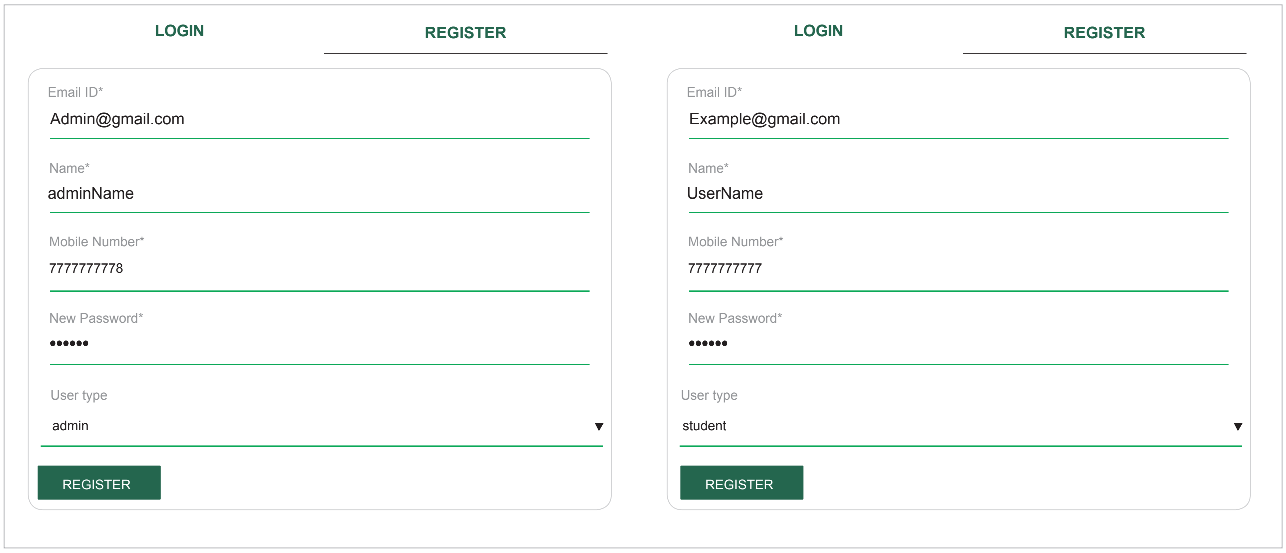


Figure 2: User registration

- To structure and handle the application, an express framework is used.
- Body-parser is used to transfer data from one *get request* body to another one.
- The template engine-EJS is used to simplify the setting up of dynamic content in html pages.
- Files are stored as chunks; this helps to store files of any size to the database.
- Mongoose is used to create schema and store the files in the database.

Functionality

Login/register: The application helps you to register by providing an email ID, user name, phone number, and the password. You can register as a student or admin. The admin can see all the users' files, and upload or download them. The details provided by you are stored in the database after checking their credentials. The user registration template is shown in Figure 2.

You can login using the email ID and password provided while registering. The application will validate the details and redirect to the user page; otherwise, it will give an error message that says 'email-id/password incorrect'. The user login page is shown in Figure 3.

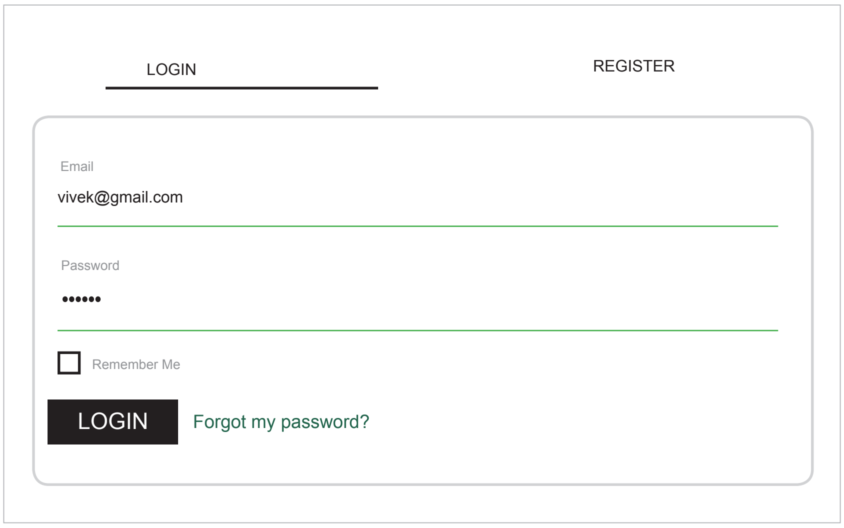


Figure 3: User login

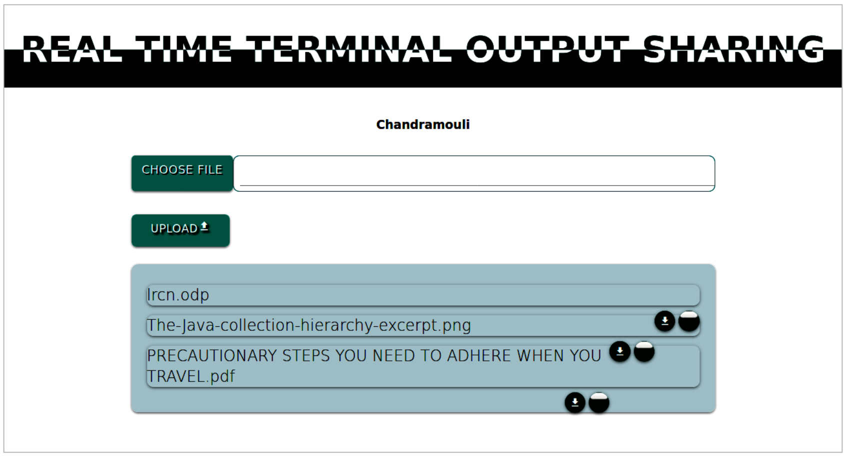

REAL TIME TERMINAL OUTPUT SHARING
Chandramouli
CHOOSE FILE
UPLOAD
lrcn.odp
The-Java-collection-hierarchy-excerpt.png
PRECAUTIONARY STEPS YOU NEED TO ADHERE WHEN YOU TRAVEL.pdf

Figure 4: User page

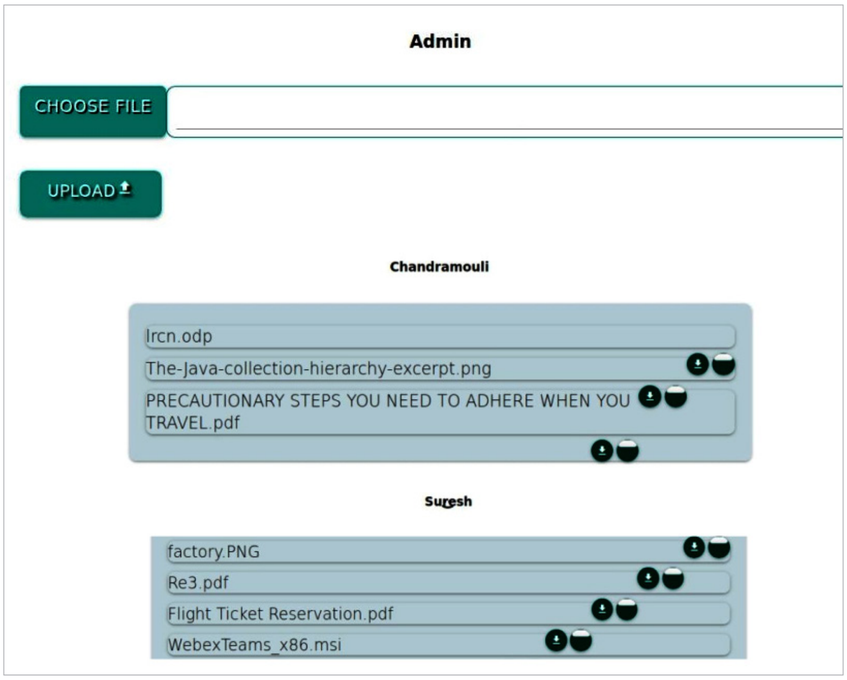
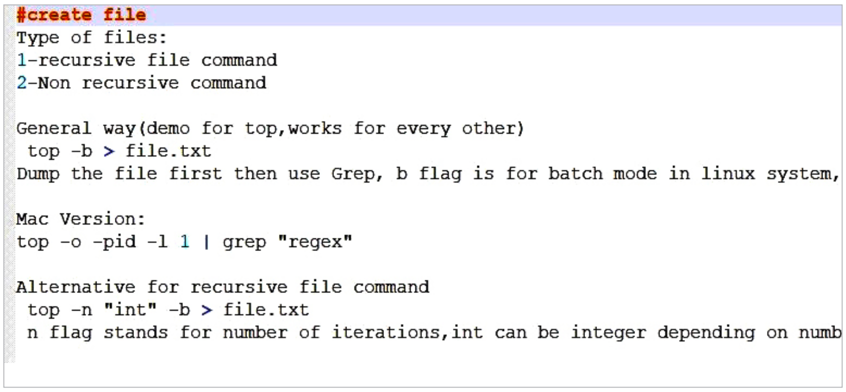

Admin
CHOOSE FILE
UPLOAD
Chandramouli
lrcn.odp
The-Java-collection-hierarchy-excerpt.png
PRECAUTIONARY STEPS YOU NEED TO ADHERE WHEN YOU TRAVEL.pdf
Suresh
factory.PNG
Re3.pdf
Flight Ticket Reservation.pdf
WebexTeams_x86.msi

Figure 5: Admin page



```
#create file
Type of files:
1-recursive file command
2-Non recursive command

General way(demo for top,works for every other)
 top -b > file.txt
Dump the file first then use Grep, b flag is for batch mode in linux system,

Mac Version:
top -o -pid -l 1 | grep "regex"

Alternative for recursive file command
 top -n "int" -b > file.txt
 n flag stands for number of iterations,int can be integer depending on numb
```

Figure 6: Creating a console file

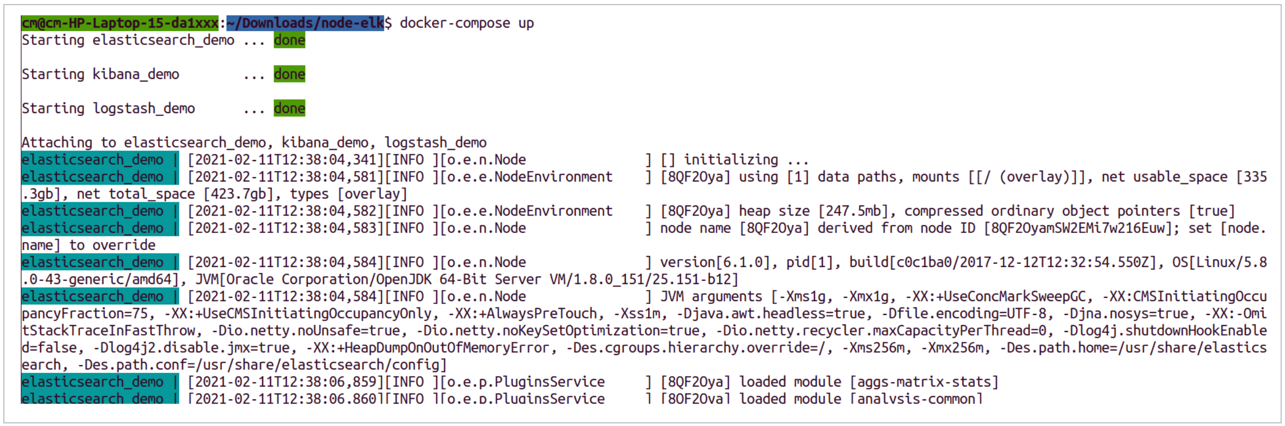
User page: You can upload files to the application, and all your files can be displayed on this page. You can also download the files or delete them anytime. The files that you choose to upload can be of any size, as all of these are stored in the form of small chunks. Figure 4 shows the user page, Figure 5 shows the admin page, and Figure 6 demonstrates how to create a console file.

Gridfs: Gridfs multi-storage is used to store all your files. It divides files into small byte chunks, which helps to store the data efficiently. Gridfs gives every chunk of a file an ID, so that the same order is followed when the files are retrieved.

Connection to MongoDB: Gridfs uses the given location to store the files in the form of chunks. Mongoose is used to help the storage in MongoDB. First, connect to MongoDB. Then, create a schema in it and use that to fix a location address as storage. That address is sent to Gridfs, with the help of Multer. Gridfs and Mongoose are able to store the user files in the database.

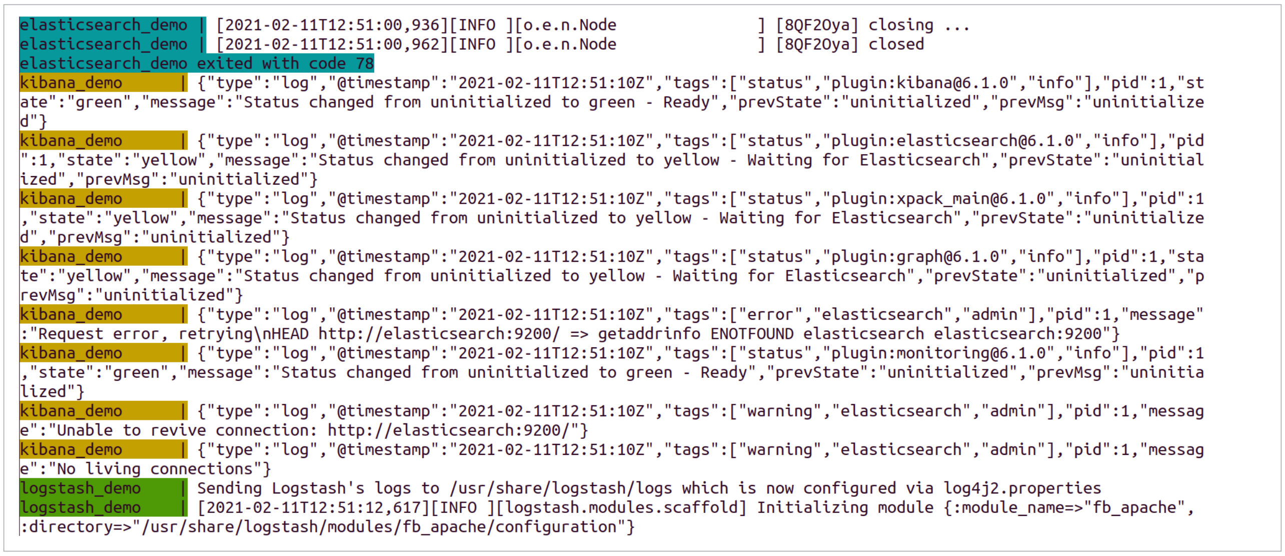
Retrieving files from storage: Students are able to view all the files in their account, while the admin is able to view all the files in the database. Files are sorted in the order of the time taken to upload them.

Uploading files to a database: The upload function is already defined. Here, it uploads your files into the user database.



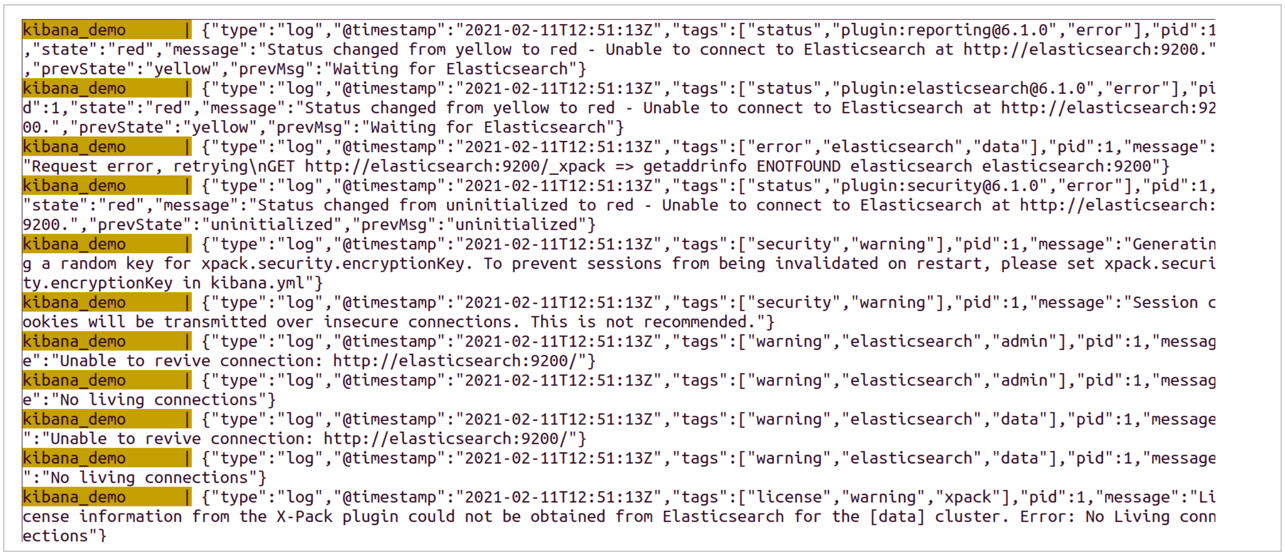
```
cm@cm-HP-Laptop-15-da1xxx:~/Downloads/node-elk$ docker-compose up
Starting elasticsearch_demo ... done

Starting kibana_demo        ... done

Starting logstash_demo      ... done

Attaching to elasticsearch_demo, kibana_demo, logstash_demo
elasticsearch_demo | [2021-02-11T12:38:04,341][INFO ][o.e.n.Node               ] [] initializing ...
elasticsearch_demo | [2021-02-11T12:38:04,581][INFO ][o.e.e.NodeEnvironment    ] [8QF2Oya] using [1] data paths, mounts [[/ (overlay)]], net usable_space [335
.3gb], net total_space [423.7gb], types [overlay]
elasticsearch_demo | [2021-02-11T12:38:04,582][INFO ][o.e.e.NodeEnvironment    ] [8QF2Oya] heap size [247.5mb], compressed ordinary object pointers [true]
elasticsearch_demo | [2021-02-11T12:38:04,583][INFO ][o.e.n.Node               ] node name [8QF2Oya] derived from node ID [8QF2OyamSW2EMi7w216Euw]; set [node.
name] to override
elasticsearch_demo | [2021-02-11T12:38:04,584][INFO ][o.e.n.Node               ] version[6.1.0], pid[1], build[c0c1ba0/2017-12-12T12:32:54.550Z], OS[Linux/5.8
.0-43-generic/amd64], JVM[Oracle Corporation/OpenJDK 64-Bit Server VM/1.8.0_151/25.151-b12]
elasticsearch_demo | [2021-02-11T12:38:04,584][INFO ][o.e.n.Node               ] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccu
pancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-Omi
tStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnable
d=false, -Dlog4j2.disable.jmx=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.cgroups.hierarchy.override=/, -Xms256m, -Xmx256m, -Des.path.home=/usr/share/elastics
earch, -Des.path.conf=/usr/share/elasticsearch/config]
elasticsearch_demo | [2021-02-11T12:38:06,859][INFO ][o.e.p.PluginsService     ] [8QF2Oya] loaded module [aggs-matrix-stats]
elasticsearch_demo | [2021-02-11T12:38:06,860][INFO ][o.e.p.PluginsService     ] [8QF2Oya] loaded module [analysis-common]
```

Figure 7: Executing the *docker-compose* command



```
elasticsearch_demo | [2021-02-11T12:51:00,936][INFO ][o.e.n.Node               ] [8QF2Oya] closing ...
elasticsearch_demo | [2021-02-11T12:51:00,962][INFO ][o.e.n.Node               ] [8QF2Oya] closed
elasticsearch_demo exited with code 78
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["status","plugin:kibana@6.1.0","info"],"pid":1,"st
ate":"green","message":"Status changed from uninitialized to green - Ready","prevState":"uninitialized","prevMsg":"uninitialize
d"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["status","plugin:elasticsearch@6.1.0","info"],"pid
":1,"state":"yellow","message":"Status changed from uninitialized to yellow - Waiting for Elasticsearch","prevState":"uninitial
ized","prevMsg":"uninitialized"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["status","plugin:xpack_main@6.1.0","info"],"pid":1
,"state":"yellow","message":"Status changed from uninitialized to yellow - Waiting for Elasticsearch","prevState":"uninitialize
d","prevMsg":"uninitialized"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["status","plugin:graph@6.1.0","info"],"pid":1,"sta
te":"yellow","message":"Status changed from uninitialized to yellow - Waiting for Elasticsearch","prevState":"uninitialized","p
revMsg":"uninitialized"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["error","elasticsearch","admin"],"pid":1,"message"
:"Request error, retrying\nHEAD http://elasticsearch:9200/ => getaddrinfo ENOTFOUND elasticsearch elasticsearch:9200"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["status","plugin:monitoring@6.1.0","info"],"pid":1
,"state":"green","message":"Status changed from uninitialized to green - Ready","prevState":"uninitialized","prevMsg":"uninitia
lized"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["warning","elasticsearch","admin"],"pid":1,"messag
e":"Unable to revive connection: http://elasticsearch:9200/"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:10Z","tags":["warning","elasticsearch","admin"],"pid":1,"messag
e":"No living connections"}
logstash_demo      | Sending Logstash's logs to /usr/share/logstash/logs which is now configured via log4j2.properties
logstash_demo      | [2021-02-11T12:51:12,617][INFO ][logstash.modules.scaffold] Initializing module {:module_name=>"fb_apache",
:directory=>"/usr/share/logstash/modules/fb_apache/configuration"}
```

Figure 8: Logstash collecting the logs and forwarding to Elasticsearch



```
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["status","plugin:reporting@6.1.0","error"],"pid":1
,"state":"red","message":"Status changed from yellow to red - Unable to connect to Elasticsearch at http://elasticsearch:9200."
,"prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["status","plugin:elasticsearch@6.1.0","error"],"pi
d":1,"state":"red","message":"Status changed from yellow to red - Unable to connect to Elasticsearch at http://elasticsearch:92
00.","prevState":"yellow","prevMsg":"Waiting for Elasticsearch"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["error","elasticsearch","data"],"pid":1,"message":
"Request error, retrying\nGET http://elasticsearch:9200/_xpack => getaddrinfo ENOTFOUND elasticsearch elasticsearch:9200"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["status","plugin:security@6.1.0","error"],"pid":1,
"state":"red","message":"Status changed from uninitialized to red - Unable to connect to Elasticsearch at http://elasticsearch:
9200.","prevState":"uninitialized","prevMsg":"uninitialized"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["security","warning"],"pid":1,"message":"Generatin
g a random key for xpack.security.encryptionKey. To prevent sessions from being invalidated on restart, please set xpack.securi
ty.encryptionKey in kibana.yml"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["security","warning"],"pid":1,"message":"Session c
ookies will be transmitted over insecure connections. This is not recommended."}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["warning","elasticsearch","admin"],"pid":1,"messag
e":"Unable to revive connection: http://elasticsearch:9200/"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["warning","elasticsearch","admin"],"pid":1,"messag
e":"No living connections"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["warning","elasticsearch","data"],"pid":1,"message
":"Unable to revive connection: http://elasticsearch:9200/"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["warning","elasticsearch","data"],"pid":1,"message
":"No living connections"}
kibana_demo        | {"type":"log","@timestamp":"2021-02-11T12:51:13Z","tags":["license","warning","xpack"],"pid":1,"message":"Li
cense information from the X-Pack plugin could not be obtained from Elasticsearch for the [data] cluster. Error: No Living conn
ections"}
```

Figure 9: Kibana output with colour coding and status

Downloading files from a database: You can download any file that is retrieved and displayed in the user page by clicking the 'Download' button.

Deleting files from a database: You can delete any file that is retrieved and displayed in the user page by using the 'Delete' button.

ELK logger

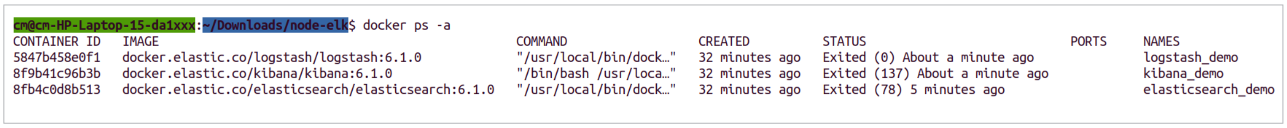
Gelf: The graylog extended format provides better functionality compared to the traditional syslog format. It is space optimised and reduces the payload. It has well-defined data structures where strings and numbers can be differentiated clearly, which is absent in traditional syslog. It provides compression features vividly. We have routed the incoming traffic from Gelf to Logstash and forwarded that to Elasticsearch at port 9200.

Elasticsearch, Logstash and Kibana (ELK): We have configured the ELK stack in a Docker image with the appropriate networks and host such that it can work anywhere, provided there is an appropriate path from Gelf and Logstash.

We have written a docker-compose file for three containers. We have to type the command *docker-compose* to start these containers. Figure 7 shows the command execution screenshot.

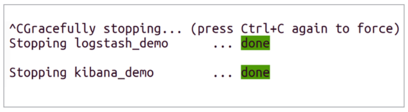
The three containers sync up, and the environments get configured according to the *docker-compose.yml* file. Then Logstash collects the logs, which is shown in Figure 8.

Elasticsearch and Kibana parse the input log file and colour code the incoming log message, with green being the most appropriate and correct, which is shown in Figure 9. The red coloured text is the most inappropriate and severe.



```
cm@cm-HP-Laptop-15-da1xxx:~/Downloads/node-elk$ docker ps -a
CONTAINER ID   IMAGE                                              COMMAND                  CREATED          STATUS                        PORTS     NAMES
5847b458e0f1   docker.elastic.co/logstash/logstash:6.1.0          "/usr/local/bin/dock…"   32 minutes ago   Exited (0) About a minute ago           logstash_demo
8f9b41c96b3b   docker.elastic.co/kibana/kibana:6.1.0              "/bin/bash /usr/loca…"   32 minutes ago   Exited (137) About a minute ago         kibana_demo
8fb4c0d8b513   docker.elastic.co/elasticsearch/elasticsearch:6.1.0 "/usr/local/bin/dock…"  32 minutes ago   Exited (78) 5 minutes ago               elasticsearch_demo
```

Figure 10: ELK containers



```
^CGracefully stopping... (press Ctrl+C again to force)
Stopping logstash_demo      ... done

Stopping kibana_demo        ... done
```

Figure 11: Killing the containers

Now, Kibana and ELK run parallelly to fetch the logs automatically at a certain interval.

You can check that all containers are in sync with the Docker *$ps - a* command, which shows the time of running for all the three containers (Figure 10).

You can terminate the *docker-compose* command with *CTRL+C/CTRL+X*, which will delete all the containers sequentially and maintain the states. The screenshot for this is shown in Figure 11.

This application can be used as a platform for online exams and for IoT based receivers. It can be used to track real-time console outputs for the evaluation of any project, including audio and video based projects apart from the routine text and image based ones. IoT sensors require a huge chunk of data to be transferred frequently in order to work properly. You can add a cron job and use the functionality of this project. Video proctoring features can also be added to this application. END

References

[1] How to share terminal output on the Web, Sorin Ionut Sbarnea, 2018, *https://medium.com/@sbarnea/how-to-share-terminal-output-on-the-web-83bf73c05c4c*
[2] Term.js, Christopher jeffrey, 2013, *https://github.com/chjj/term.js/*
[3] Rtail, Kilian Ciuffolo Tim Rio Sandaruwan-Silva, 2015, *https://github.com/kilianc/rtail*
[4] GridFSBucket, mongoDB, *https://api.mongodb.com/java/3.3/?com/mongodb/client/gridfs/GridFSBucket.html*
[5] Syslog daemon for security event monitoring using UDP protocol, C. Roja and P. N. Jayanthi, 2019, *https://ieeexplore.ieee.org/abstract/document/8821956.*
[6] Elastic Stack and product documentation, elastic.co, *https://www.elastic.co/guide/index.html*

By: K. Chandramouli, Gadde Suresh, Vivek Kumar Singh, Amitesh Anand and B. Thangaraju

The authors are associated with the open source technology lab in the International Institute of Information Technology, Bengaluru.

Statement about ownership and other particulars about
OPEN SOURCE FOR YOU
FORM IV (See Rule 8)

1.	Place of publication	:	New Delhi
2.	Periodicity of its publication	:	Monthly
3.	Printer's Name	:	Ramesh Chopra
	Nationality	:	Indian
	Address	:	OPEN SOURCE FOR YOU D-87/1, Okhla Industrial Area, Phase-1, New Delhi 110020
4.	Publisher's Name Nationality and address	:	Same as (3) above
5.	Names and addresses of individuals who own the newspaper & partners or shareholders holding more than 1% of the total capital	:	**EFY Enterprises Pvt Ltd** D-87/1, Okhla Industrial Area, Phase-1, New Delhi 110020

I, Ramesh Chopra, hereby declare that the particulars given above are true to the best of my knowledge and belief.

Date: 28-2-2021

Ramesh Chopra
Publisher