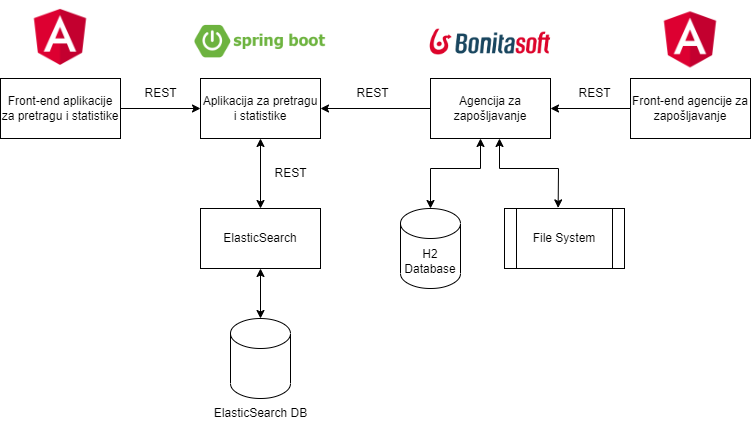
# Upravljanje digitalnim dokumentima

# - Kontrolna tačka -

## Anica Đukić E2 55/2022

### Arhitektura sistema

Arhitektura sistema prikazana je na sledećoj slici:



Kao što je prikazano sistem se sastoji od dve aplikacije:

* ***Agencija za zapošljavanje*** – aplikacija koja je implementirana pomoću Bonitasoft Bussiness Proccess Managment sistema (BPMS) koji ima ugrađenu podršku za korišćenje H2 in-memory baze podataka. Pored baze podataka koristi se File System za skladištenje dokumenata poput CV dokumenta i propratnog pisma. Bonitasoft BPMS komunicira sa front-end aplikacijom koja je implementirana pomoću Angular radnog okvira. Ova front-end aplikacija omogućava kandidatima da apliciraju za posao i da podnesu zahtev za premium zapošljavanje.
* ***Aplikacija za pretragu i statistike*** – Spring Boot aplikacija koja omogućava pretraživanje aplikacija (na osnovu kriterijuma zadatih u specifikaciji) uz pomoć ElasticSearch platforme i generiše statistike (definisane specifikacijom) na osnovu podataka dobijenih iz odgovarajućih logova. Ova aplikacija ima troslojnu arhitekturu (controller, service, repository – upotrebom ***ElasticRepository*** interfejsa šalje i dobija podatke od ElasticSearch platforme). Takođe, ova aplikacija komunicira sa front-end aplikacijom koja je implementirana pomoću Angular radnog okvira. Ova front-end aplikacija pruža formu za pretragu aplikacija kao i pregled statistika.

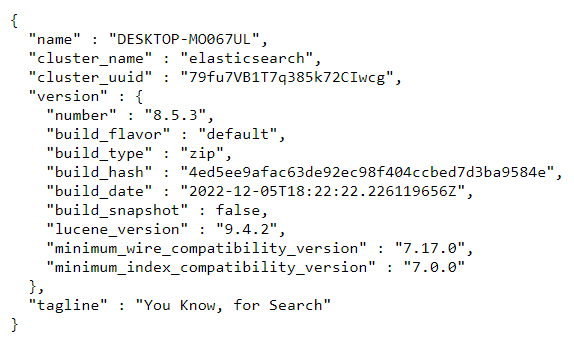
***Agencija za zapošljavanje*** svaki put kada dobije novu aplikaciju šalje njene podatke REST pozivom ***Aplikaciji za pretraživanje i statistiku***. Zatim, ***Aplikacija za pretraživanje i statistiku*** obrađuje podatke (dobijanje geo koordinata adrese, parsiranje pdf dokumenata – CV i propratno pismo) i prosleđuje ***ElasticSearch platformi***.

Što se tiče obrade logova, svaki put kada neki kandidat podnese zahtev za premium zapošljavanje ili se uspešno zaposli preko premium zapošljavanja **Agencija za zapošljavanje** javlja REST pozivom ***Aplikaciji za pretragu i statistike*** koja dobijene podatke loguje kod sebe. Log fajlove ***Aplikacije za pretragu i statistike*** prati ***Filebeat*** koji čim primeti novi log, šalje ga ***Logstash***-u na obradu, koji zatim taj log prosleđuje ***ElasticSearch***-u. ***ElasticSearch*** je povezan sa **Kibanom** koja omogućava pregled pristiglih logova.

### Konfiguracija ELK steka

#### ElasticSearch

* Preuzeta najnovija verzija ElasticSearch-a (8.5.3)
* Da bi se pokrenuo ElasticSearch potrebno je pozicionirati se u njegov ***bin*** folder i pokrenuti ***elasticsearch.bat***.
* ElasticSearch obično trči na portu 9200, pa bi za proveru da li je ElasticSearch podignut bilo dovoljno otići na <http://localhost:9200> u browseru.
* Ukoliko se dobije sledeći prikaz ElasticSearch je pokrenut.



#### Logstash

* Preuzeta najnovija verzija Logstash-a (8.5.3)
* Unutar njegovog ***bin*** foldera je kreiran konfiguracioni (.conf) fajl ***logstash.conf***koji se sastoji od tri komponente:
  + Input – definiše izvor podataka (apsolutna putanja do log fajla)
  + Filter – definiše kakvo procesiranje želimo da primenimo na podatke
  + Output – definiše gde šaljemo podatke (adresa i port ElasticSearch-a)
* Da bi se pokrenuo Logstash dovoljno je pokrenuti komandu ***logstash -f logstash.conf.***

#### Kibana

* Preuzeta najnovija verzija Kibane (8.5.3)
* Da bi se pokrenula Kibana potrebno je pozicionirati se u njen ***bin*** folder i pokrenuti ***kibana.bat***.
* Kibana obično trči na portu 5601, pa bi za proveru da li je Kibana pokrenuta bilo dovoljno otići na <http://localhost:5061> u browseru.
* Ukoliko se dobije Kibana home page, Kibana je pokrenuta.

#### Filebeat

Da bi se ELK stack mogao kombinovati sa Filebeat-om potrebno je:

* Preuzeti najnoviju verziju Filebeat-a (8.5.3)
* Upisati u ***filebeat.yml*** sledeće:

filebeat.inputs:

- type: log

enabled: true

paths:

- /ELK/logs/application.log # Absolute path of the log file

output.logstash:

hosts: ["localhost:5044"] # Output is now pointing to Logstash

* U inputu specificirati apsolutnu putanju do log fajla iz kog će Filebeat čitati podatke.
* U outputu navesti adresu i port na kom se nalazi Logstash
* Pokrenuti komandu ***filebeat.exe -c filebeat.yml***
* U logstash.yml dodati sledeće

input {

beats { # Notice the input is now being taken from 'Beats'

type => "logs"

port => "5044"

}

}

filter { # Adding the filters

if [message] =~ "\tat" {

grok {

match => ["message", "^(\tat)"]

add\_tag => ["stacktrace"]

}

}

}

output { # Output pointing to Elasticsearch

stdout {

codec => rubydebug

}

elasticsearch {

hosts => ["localhost:9200"]

}

}

* + U inputu podešeno da se sluša Filebeat
  + U outputu podešeno da se podaci šalju ElasticSearch-u
* Otići na [*http://localhost:9200/\_cat/indices*](http://localhost:9200/_cat/indices)i proveriti da li se pojavio novi indeks
* Ukoliko jeste otići na [*http://localhost:5601/app/management/kibana/indexPatterns*](http://localhost:5601/app/management/kibana/indexPatterns)i dodati novi indeks.
* Otići na [*http://localhost:5601/app/discover*](http://localhost:5601/app/discover)i izabrati kreirani indeks kako bi se videli preuzeti logovi.

\* Ispraćeno uputstvo sa sajta: <https://auth0.com/blog/spring-boot-logs-aggregation-and-monitoring-using-elk-stack/>

Serbian Analyzer

Koristi se 8.5.0 verzija ElasticSearch platforme koja bi trebalo da ima ugrađenu podršku za pretprocesiranje srpskih tekstova.

Indexing unit

## Aplikacija:

* Ime
* Prezime
* Obrazovanje - stepen stručne spreme
* CV - isparsiran pdf dokument CV-a
* Propratno pismo - isparsiran pdf dokument propratnog pisma
* Latituda
* Longituda

## Logovi:

* Čuva se tekst logova dobijenih iz Logstash-a

### Geoprostorna pretraga

***Latituda i longituda*** adrese korisnika koji aplicira za posao, se mogu dobiti uz pomoć ***Geocoding API***-ja. ***Geocoding API*** koji je dostupan na linku: <https://positionstack.com/> nudi [Forward Geocoding](https://positionstack.com/documentation#forward_geocoding) koji na osnovu poslate adrese vraća podatke među kojima su ***latituda i longituda***, koje se čuvaj u okviru ***Aplikacije*** (Indexing unit).

Kada se bude vršila ***pretraga zadavanjem imena grada i radiusa***, korišćenjem ***Geocoding API***-ja se mogu dobiti ***latituda i longituda zadatog grada*** i računati njihova ***udaljenost od latitude i longitude traženih aplikacija*** na osnovu ***Haversine*** ***metode***.

Što se tiče pravljenja statistike ***iz kojeg grada je bilo najviše zahteva za premium zapošljavanje***, može se koristiti ***Geolocation******API*** (<https://ipgeolocation.io/documentation/ip-geolocation-api.html>) koji na osnovu ***IP adrese*** vraća podatke među kojima je ***naziv grada***. Tako da se može dobiti grad na osnovu IP adrese sa koje je kandidat poslao zahtev za premium zapošljavanje i to koristiti za pravljenje pomenute statistike.