SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

ZAVRŠNI RAD br. 7189

# Primjena alata ElasticSearch, LogStash i Kibana za analizu podataka o IP adresama

Dora Bezuk

Zagreb, svibanj 2021.

# SADRŽAJ

# UVOD

Računalna sigurnost bavi se zaštitom internetskih sustava i podataka od kibernetičkih prijetnji. Uloga obavještajnog rada u kibernetičkom prostoru je sakupljanje i kategorizacija znanja, koje služi za osmišljavanje najbolje metode obrane od potencijalnih napada. Logovi su datoteke koje pružaju sigurnosne informacije o pojedinom sustavu. Zahvaljujući informacijama iz logova, sigurnosni stručnjaci koji se bave obavještajnim radom, mogu pravovremeno poduzeti odgovarajuće mjere u cilju poboljšanja sigurnosti sustava, te osmisliti najbolju metodu za zaštitu osjetljivih podataka.

Sustavno praćenje logova ključan je faktor za detaljnu analizu mogućih prijetnji, a detaljnijom analizom povećava se vjerojatnost uspjeha pri obrani od kibernetičkih napada. Zbog opsežnosti i količine logova koju generiraju sustavi, otežano je praćenje promjena i analiza podataka, stoga ručno analiziranje i sakupljanje podataka prestaje biti dostatno. Korištenjem alata koji agregiraju, automatski osvježavaju te naposljetku, vizualiziraju korisne informacije omogućava se praćenje podataka u stvarnom vremenu. U ovom radu promatra se način na koji se navedeni alati mogu koristiti za analizu informacija o IP adresama.

Rad je koncipiran u više dijelova. Prvi dio rada opisuje postojeće izvore kibernetičkih prijetnji, te ulogu analize logova i IP adresa u obavještajnom radu. Drugi dio opisuje korištene tehnologije i alate za agregaciju i vizualizaciju podataka, dok se u trećem dijelu rada naglasak prebacuje na korištenje navedenih alata u kontekstu analize Apache logova. Na kraju, donosi se zaključak o upotrebi ELK platforme i alata Grafane za potrebe analize logova, te IP adresa.

# 1. Obavještajni rad o kibernetičkim prijetnjama

## 1.1. Uloga CTI-a u borbi protiv kibernetičkog kriminala

Obavještajni rad u kibernetičkom prostoru oslanja se na prikupljanje i analizu podataka o kibernetičkim prijetnjama s ciljem boljeg razumijevanja motiva, ciljeva i ponašanja napadača u svrhu kreiranja najbolje obrane. Za uspješan obavještajni rad, izuzetno je važno razlikovati pojmove informacije o kibernetičkim prijetnjama (engl. *cyber threat information*) i inteligenciju kibernetičkih prijetnji (engl. *cyber threat intelligence*). Informacije o kibernetičkim prijetnjama predstavljaju sirovi, nefiltrirani oblik podataka koji u trenutku isporuke nije prošao niti jednu vrstu evaluacije. Podaci u informacijama su agregirani iz velikog broja izvora što može stvoriti pogrešnu ili obmanjujuću viziju o stvarnim prijetnjama. S druge strane, inteligencija o kibernetičkim prijetnjama predstavlja obrađene i sortirane informacije, koje su prošle analizu od strane obučenih obavještajnih analitičara. Također, informacije korištene u inteligenciji su agregirane iz pouzdanih izvora te međusobno korelirane kako bi bile što relevantnije. Najvažnija razlika između dva navedena pojma je upravo činjenica da se na temelju inteligencije može djelovati u cilju obrane, dok se na temelju informacija ne može.


**Slika 1.1:** Četiri vrste inteligencije vezane uz kibernetičke prijetnje [1]

Slika 1.1 prikazuje podjelu inteligencije kibernetičke prijetnje na četiri glavne vrste: strateška, taktička, tehnička i operativna.

Strateška inteligencija širi je pojam obično rezerviran za netehničku publiku. Koristi detaljne analize trendova i novih rizika kako bi se stvorila opća slika mogućih posljedica kibernetičkog napada. Neki konkretni primjeri vezani uz stratešku inteligenciju, uključuju tehničke dokumente, dokumente o pravilima i publikacije distribuirane unutar IT industrije. Taktička inteligencija nudi konkretnije detalje o taktikama, tehnikama i postupcima aktera u kontekstu prijetnje, poznatim i kao TTP (engl. *Tactics, Techniques, and Procedures*). Namijenjena je pretežno tehničkoj publici i pomaže razumijevanju na koji način mreža ili sustav može biti napadnuta pomoću najnovijih metoda koje napadači koriste za postizanje svojih ciljeva. Ova vrsta inteligencije rezervirana je za sigurnosne timove koji su izravno uključeni u zaštitu mreže od napada. Obavještajna inteligencija usredotočena je na tehničke zapise koji ukazuju na prijetnju kibernetičkoj sigurnosti. Ova vrsta inteligencije je važna jer ljudima daje ideju na što treba paziti i koji podaci su korisni za analizu napada u sklopu socijalnog inženjeringa. Operativna inteligencija pomaže IT stručnjacima koji brane sustave da razumiju prirodu specifičnih kibernetičkih napada detaljno opisujući relevantne čimbenike poput prirode, namjere, vremena i sofisticiranosti odgovorne skupine.

Svi aspekti inteligencije kibernetskih prijetnji jednako su važni za sveobuhvatnu procjenu prijetnji.

## 1.2. Izvori kibernetičkih prijetnji

Prilikom identificiranja kibernetičkih prijetnji, informacija o tome tko stoji iza prijetnje ponekad je važnija od tehnologije korištene za napad. Mehanizmi korišteni za napad neprestano se razvijaju i mijenjaju, no izvori prijetnji ostaju isti jer se oslanjaju na ljudski faktor i motiv. Prema [1] izvori kibernetičkih prijetnji mogu se podijeliti na:

1. Nacionalne organizacije
2. Terorističke skupine
3. Špijune i skupine organiziranog kriminala
4. Haktiviste
5. Hakere

Nacionalne organizacije koje koriste kibernetičke oblike prijetnji, predstavljaju prijetnju čitavom spektru informacija, čija potencijalna kompromitiranost može naštetiti interesima neke države. Prijetnje se mogu kretati od propagande i oštećenja web stranica niske razine do špijunaže i velikih poremećaja u infrastrukturi neke države. Glavna prednost organizacija na nacionalnoj razini su financijska sredstva i opseg uključenih ljudi. Terorističke skupine trenutno predstavljaju manji rizik u kibernetičkom prostoru jer se još uvijek oslanjaju na tradicionalne metode napada, no konstantnim povećanjem tehničkih kompetencija generacija pretpostavlja se da će u budućnosti biti u mogućnosti predstavljati značajniju kibernetičku prijetnju. Međunarodni špijuni i organizacije organiziranog kriminala predstavljaju prijetnju na srednjoj razini zbog sposobnosti unajmljivanja i razvijanja hakerskih talenata. Njihov motiv je isključivo monetarne prirode. Haktivisti čine manje skupine politički aktivnih hakera koji predstavljaju prijetnju srednje razine za izvršavanje izoliranih, ali štetnih napada. Cilj većine međunarodnih haktivističkih skupina jest širenje političke propagande, a ne nanošenje ozbiljne štete infrastrukturi sustava kojeg napadaju. Hakeri čine najbrojniji i najozloglašeniji izvor kibernetičke prijetnje. Mnogi kibernetički napadi amatera koji hakiraju računala, predstavljaju zanemarivu prijetnju za nanošenje ozbiljne štete infrastrukturi nekog velikog sustava upravo zbog manjka motivacije. Unatoč tome, veliki broj svjetske populacije hakera predstavlja relativno visoku prijetnju u svojim izoliranim napadima, a kako hakerska populacija raste, tako raste i vjerojatnost da izuzetno vješt i zlonamjeran haker pokuša i uspije proizvesti napad s ozbiljnim posljedicama za sustav. Uz to, veliki broj manje kvalificiranih hakera u svijetu povećava mogućnost nenamjernog narušavanja infrastrukture nekog sustava.

## 1.3. Apache logovi i IP adrese

Pravilna pohrana, analiza i praćenje logova igra ključnu ulogu u digitalnoj sigurnosti računala u mreži. Pomoću temeljite analize logova, omogućen je uvid u sve aktivnosti unutar mreže, a takva informacija pomaže u poduzimanju pravovremene i prikladne zaštite u cilju obrane od potencijalnih prijetnji.

Prema [2], manjak sigurnosne evidencije (engl. *security logging*) i analize omogućuje napadačima prikrivanje položaja, zlonamjernih softvera ili aktivnosti na računalima žrtava. Čak i ako žrtve znaju da su njihovi sustavi ugroženi, bez zaštićenih i cjelovitih evidencija logova, slijepi su za detalje napada i za naknadne radnje koje napadači mogu poduzeti. Bez praćenja logova i njihove pravilne analize, napad može neograničeno trajati, a određena šteta koja je nanesena može biti nepovratna.

Redoviti pregled logova ključan je za uspješnu obranu sustava, no s obzirom na veliku količinu logova koje generiraju sustavi, nepraktično je svakodnevno pregledavati sve logove, pa se u tu svrhu koriste alati za agregaciju i vizualizaciju podataka kao što je ELK stack platforma i Grafana.

Logovi su datoteke koje bilježe događaje unutar operacijskog sustava ili drugog softvera. Razlikujemo više vrsta logova. Dnevnici događaja (engl. *event logs*) bilježe aktivnosti koje se događaju pri izvršavanju sustava, a koriste se za razumijevanje aktivnosti sustava i dijagnosticiranje problema. Dnevnici poruka (engl. *message logs*) koriste se za automatsko prijavljivanje ili spremanje tekstualne komunikacije između korisnika. Podaci pohranjeni u zapisnicima transakcija (engl. *transaction logs*) internetskih pretraživača, pružaju uvid u razumijevanje procesa pretraživanja internetskih pretraživača. Zapisnici poslužitelja ((engl. *server logs*) su skupine logova koje automatski stvara i održava poslužitelj, a koje se sastoje od popisa aktivnosti koje poslužitelj izvodi.

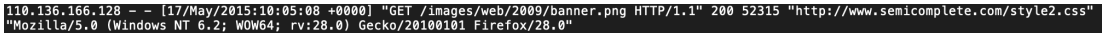
Zapisnici poslužitelja od posebne su važnosti u svijetu sigurnosti stoga se pohranjuju u standardiziranim oblicima. Uobičajeni format loga (engl. *Common Log Format*) standardizirani je format tekstualne datoteke koji web poslužitelji koriste prilikom generiranja datoteka zapisnika poslužitelja. Budući da je format standardiziran, datoteke mogu lako analizirati razni programi za web analizu.


**Slika 1.2:** Primjer loga u uobičajenom formatu *Common Log Format*

Svaki redak u datoteci pohranjenoj u uobičajenom formatu, vidljivoj na slici 1.2 sastoji se od unaprijed definiranih dijelova, te slijedi navedenu strukturu:

1. 172.33.102.37 - IP adresa klijenta koji je uputio zahtjev poslužitelju
2. "-" - korisnički identifikator u RFC 1413 obliku, koji najčešće nije dostupan
3. "-" - korisničko ime osobe koja traži dokument, najčešće nije dostupan osim ako se ne zatraži provjera autentičnosti
4. "[13 / May / 2021: 22: 41: 30 +0000]" - datum, vremenska zona i vrijeme kada je zahtjev primljen, prema zadanim postavkama u strftime formatu
5. "GET /category/games HTTP / 1.1" - redak zahtjeva klijenta, sadrži metodu GET, traženi resurs, i verziju HTTP protokola
6. "200" - HTTP statusni kôd vraćen klijentu. 2xx je uspješan odgovor, 3xx predstavlja preusmjeravanje, 4xx pogrešku klijenta i 5xx pogrešku poslužitelja
7. "88" - veličina predmeta vraćena klijentu, mjerena u bajtovima

Navedeni format može biti proširen i pretvoren u kombinirani format loga (engl. *Combined Log Format*) s dodatnim informacijama kao što su podaci o uputitelju, korisničkom agentu i slično.




**Slika 1.3:** Primjer loga u kombiniranom formatu *Combined Log Format*

Na slici 1.3 vidljiv je primjer kombiniranog formata loga koji sadrži dodatne informacije:

- podatke o uputitelju (engl. *referer*) koji sadrže informacije otkud je korisnik došao do stranice, a dostupni su samo ako klijent pošalje navedene podatke na poslužitelj
- podatke o korisničkom agentu poslane sa klijentske strane, navedeni podaci se mogu koristiti za prepoznavanje preglednika korištenog u pretrazi, ali ti podaci nisu uvijek točni

Apache logovi bilježe događaje kojima je rukovao web poslužitelj Apache, uključujući zahtjeve s drugih računala, odgovore koje je poslao Apache i unutarnje radnje na Apache poslužitelju. Apache poslužitelj generira više vrsta logova: logove pristupa (engl. *access logs*) te logove pogrešaka (engl. *error logs*).

Logovi pogrešaka pohranjuju dijagnostičke podatke i svaku pogrešku koja se dogodila tijekom obrade zahtjeva. Koriste se za operativni nadzor i rješavanje problema jer sadrže dijagnostičke podatke i pogreške evidentirane tijekom obrade zahtjeva kao što je vidljivo na slici 1.4.

```
[Sat Jan 18 16:22:00 2020] [error] [client 192.168.33.1] File does not exist: /var/www/favicon.ico, referer: http://192.168.33.72/
```

**Slika 1.4:** Primjer zapisnika pogreške [2]

Logovi pristupa pohranjuju sve zahtjeve koje obrađuje Apache HTTP poslužitelj i koriste se za nadzor sustava, te za rješavanje sigurnosnih problema. Sadrže informacije o zahtjevima usmjerenim prema Apache serveru. Apache zapisnici najčešće se pohranjuju u kombiniranom formatu loga.

Vrlo važna informacija koju u sebi nose Apache zapisnici je IP adresa klijenta. IP adresa predstavlja numerička oznaka koja je dodjeljena svakom uređaju spojenom na računalnu mrežu, a koje za potrebe komunikacije koristi internetski protokol. IP adresa ima dvije glavne funkcije: identifikacija mreže, odnosno mrežnog sučelja i adresiranje lokacije.

# 2. Tehnologije i alati za agregaciju i vizualizaciju podataka

## 2.1. ELK stack

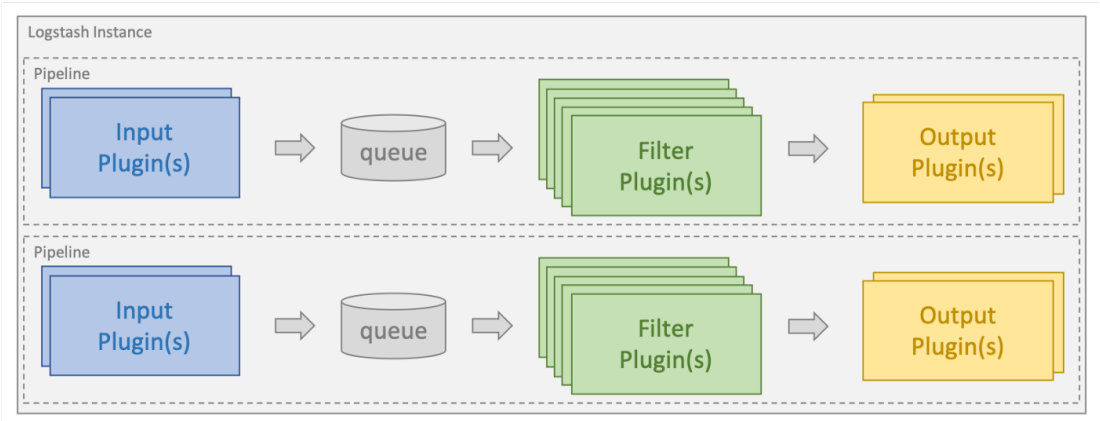
Elasticsearch, Logstash i Kibana tri su softvera otvorenog koda koji zajedno objedinjeni tvore ELK stack platformu. Slika 2.1 vizualno prikazuje na koji način su povezane komponente ELK platforme. Platforma prikuplja i obrađuje podatke iz više izvora podataka, pohranjuje podatke u jednu centraliziranu bazu podataka koja nudi mogućnost skaliranja shodno porastu broja podataka, te pruža skup alata za analizu navedenih podataka i vizualizaciju.


**Slika 2.1:** Koncept ELK platforme [3]

ELK svoju primjenu pronalazi u vođenju analitike logova, nadzor izvedbe aplikacija, nadzor i analizu sigurnosti te u analitici podataka. Porastom popularnosti ELK platforme, proširene su sigurnosne značajke koje omogućuju osiguravanje nakupina podataka, zaštitu podataka lozinkama, kao i primjenu naprednijih sigurnosnih mjera poput šifriranja komunikacije, podešavanja kontrole pristupa zasnovane na ulogama, IP filtriranja i revizije.

### 2.1.1. Logstash

Logstash je cjevovod (engl. *pipeline*) za obradu podataka na poslužitelju koji istodobno unosi podatke iz više izvora, dinamički ih transformira, a zatim prosljeđuje alatu za pretraživanje i analitiku kao što je Elasticsearch. Struktura Logstash cjevovoda, vidljiva na slici 2.2, podijeljena je u tri dijela: ulaz, filter i izlaz.




**Slika 2.2:** Struktura logstash cjevovoda [4]

Ulaz (engl. *input*) definira izvore podataka. Dostupan je veliki broj softverskih proširenja (engl. *plugin*) koji pojednostavljuju unos podataka iz raznih izvora te omogućuju Logstashu čitanje određenih izvora događaja. Neka od mogućih proširenja su: *datoteka* pomoću koje je omogućeno čitanje iz datoteke u datotečnom sustavu, *syslog* koji se koristi za osluškivanje poruka protokola sistemskih logova na dobro poznatom portu 514, *stdin* koji čita događaje sa standardnog ulaza i mnogi drugi.

Filter služi za transformaciju podataka te kao posrednik u cjevovodu između ulaza i izlaza. Primarna uloga filtera je pretvorba podataka u željeni oblik prije analize. Različite vrste filtera dostupne su biblioteci filtera, a neki od mogućih filtera su: *grok* koji omogućava raščlanjivanje i strukturiranje proizvoljnog teksta, *mutacija* koja omogućava izvođenje općih transformacija nad poljima događaja, *geoip* kojim se omogućava dodavanje informacija o zemljopisnom položaju IP adresa i mnogi drugi.

Izlaz (engl. *output*) predstavlja posljednji korak u cjevovodu i služi za određivanje jedne ili više izlaznih lokacija na koju se šalju podaci koji su prošli kroz cjevovod. Kao za ulaz i filtere, postoje mnogobrojna proširenja za izlaze dostupna u biblioteci izlaza. Neki od mogućih izlaza uključuju: *elasticsearch* koji prosljeđuje podatke o događajima Elasticsearchu, *datoteka* koja omogućava zapis podataka o događajima u datoteku na disku i ostali.

### 2.1.2. Elasticsearch

Elasticsearch je softver otvorenog koda za pretraživanje i analitiku podataka, izgrađen na biblioteci Apache Lucene, prvi put objavljen 2010. godine. Nudi mogućnost pohrane, pretraživanja i analize velike količine podataka u gotovo stvarnom vremenu jer umjesto izravnog pretraživanja pretražuje po indeksu. Arhitektura Elasticsearcha bazira se na logičkim konceptima organiziranim u obliku podatkovnih komponenti.

#### 2.1.2.1. Elasticsearch logički koncepti

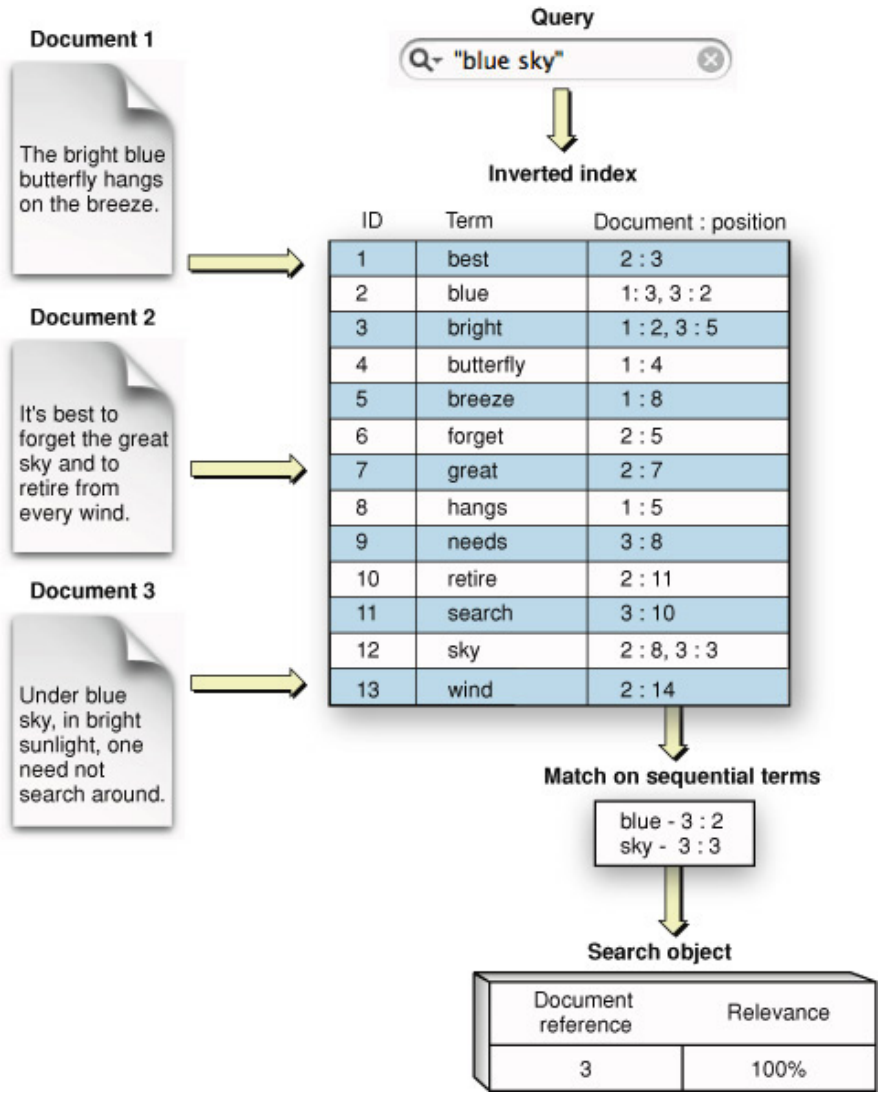
Promatrano logički, Elasticsearch je podijeljen na:

1. dokumente (engl. *documents*) - Dokumenti predstavljaju osnovne jedinice podataka, paralelni su entitetima (redu) u relacijskoj bazi podataka, a osim teksta, dokumenti mogu biti bilo koji strukturirani podaci zapisani u JSON standardnom formatu datoteke (brojevi, nizovi, datumi). Svaki dokument ima jedinstveni ID i zadanu vrstu podatka kojom se opisuje vrsta entiteta koju dokument predstavlja.

2. indekse (engl. *indices*) - Indeksi čine skup dokumenata sa sličnim svojstvima koji su logički povezani, a koji zajedno čine mehanizam organizacije podataka koji omogućava korisniku da grupira podatke po želji. Pojam indeksa sličan je pojmu baze u relacijskoj bazi podataka.

3. obrnuti indeksi (engl. *inverted index*) - Obrnuti indeksi su podatkovne strukture koja imaju mapiran sadržaj (npr. riječ, broj) i njegovu poziciju u dokumentu ili skupu dokumenata. Obrnuti indeksi su nalik hashu koji usmjerava od riječi do dokumenta, konkretno - razdvaja svaki dokument do pojedinačnih pojmova za pretraživanje (tj. riječi), a zatim preslikava svaki pojam za pretraživanje na dokumente u kojima se ti pojmovi pojavljuju. Korištenje distribuiranih obrnutih indeksa Elasticsearchu omogućava brzo pronalaženje najboljeg podudaranja za pretraživanja cijelog teksta iz čak i vrlo velikih skupova podataka.

Prethodno opisan mehanizam stvaranja obrnutih indeksa iz dokumenata kod pretraživanja prikazan je na slici 2.3.




**Slika 2.3:** Mehanizam stvaranja obrnutih indeksa iz dokumenata kod pretraživanja [5]

Mnogi logički koncepti Elasticsearcha mogu se paralelno usporediti sa onima iz relacijske baze podataka. Kao što je vidljivo na slici 2.4 indeksi su paralelni pojmu baze, tipovi pojmu tablice, a dokumenti pojmu redova.

- MySQL ⇒ Databases ⇒ Tables ⇒ Columns/Rows
- Elasticsearch ⇒ Indices ⇒ Types ⇒ Documents with Properties

**Slika 2.4:** Usporedba relacijske baze podataka i ElasticSearcha [2]

#### 2.1.2.2. Elasticsearch podatkovne komponente

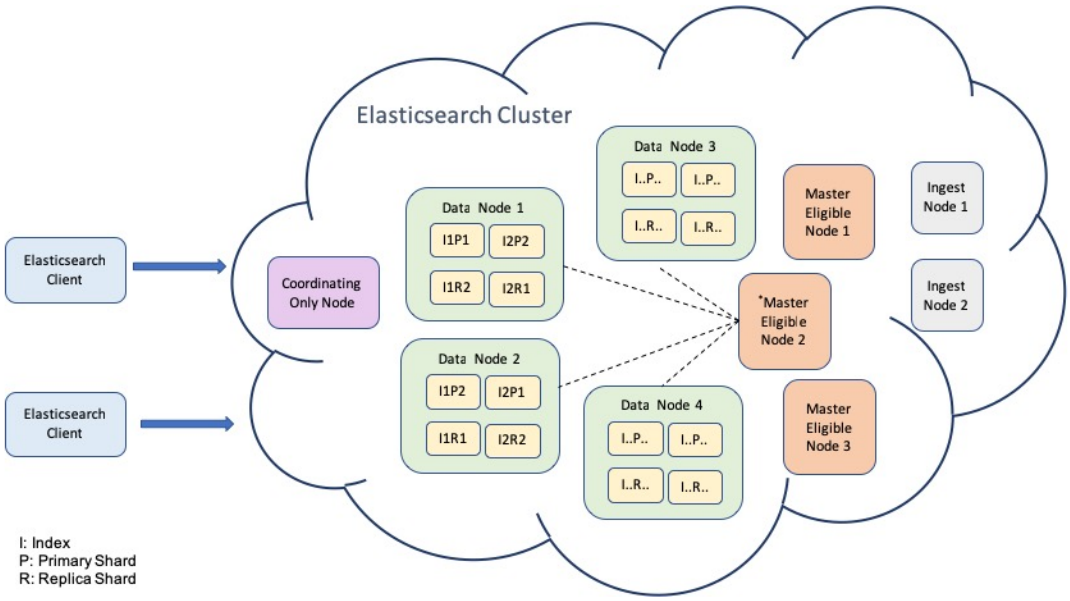
Pored logičke podjele, postoji podjela na podatkovne komponente. Podatkovne komponente koje čine Elasticsearch i njihov međusobni odnos prikazan na slici 2.5 su:

1. čvor (engl. *node*) - Poslužitelj koji je dio klastera. Uloga mu je čuvanje podataka i sudjelovanje u indeksiranju i mogućnostima pretraživanja klastera, a može se konfigurirati na različite načine, kao:
   - glavni čvor (engl. *master node*) - kontrolira klaster Elasticsearcha i odgovoran je za sve operacije na razini klastera, uključujući stvaranje i brisanje indeksa te dodavanje i uklanjanje ostalih čvorova.
   - čvor podataka (engl. *data node*)- pohranjuje podatke i izvršava operacije povezane s podacima, poput pretraživanja i agregiranja.
   - klijentski čvor (engl. *client node*) - prosljeđuje zahtjeve klastera na glavni čvor i zahtjeve povezane s podacima na čvorove podataka.

2. klasteri (engl. *cluster*) - Klasteri su skupine jedne ili više instanci čvorova koje su povezane zajedno. Moć Elasticsearch klastera leži u raspodjeli zadataka, pretraživanju i načinu indeksiranja po svim čvorovima klastera.

3. krhotine (engl. *shards*) - Krhotine su dijelovi na koje se može razdijeliti indeks. Svaka krhotina je sama po sebi potpuno funkcionalan i neovisan "indeks" koji se može smjestiti na bilo koji čvor unutar klastera.

4. replike ili replike krhotina (engl. *replicas*) - Replike su jedna ili više kopija (primarnih) krhotina indeksa. Svaki dokument u indeksu pripada jednoj primarnoj krhotini, a svrha stvaranja kopije podataka je osiguranje u slučaju otkazivanja hardvera, te povećanje kapaciteta posluživanja zahtjeva za čitanje, poput pretraživanja ili dohvaćanja dokumenta.




**Slika 2.5:** Arhitektura ElasticSearch podatkovnih komponenti [6]

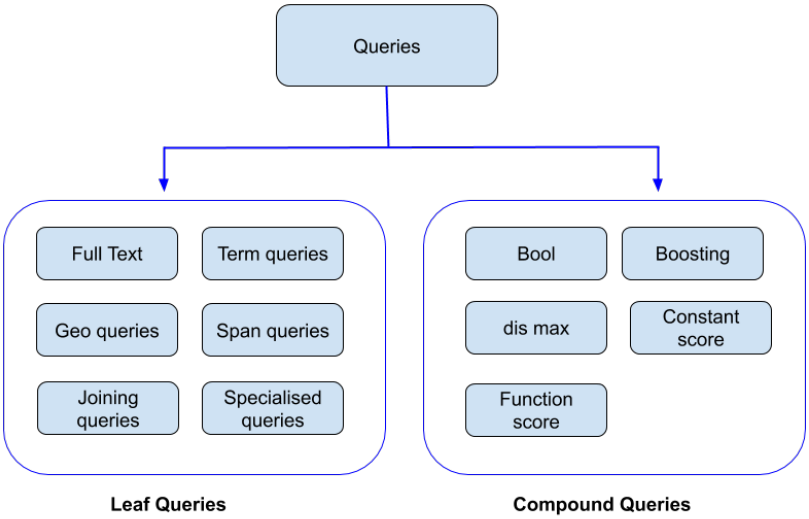
#### 2.1.2.3. Elasticsearch opcije pretraživanja

Pretraživanje u Elasticsearchu može se podijeliti na dva glavna načina pretraživanja:

1. Upiti (engl. *queries*) - vraćaju dokumente koji odgovaraju određenim kriterijima
2. Agregacije - zbirke upita (engl. *agregations*) - sažimaju podatke i prikazuju ih kao mjerne podatke, statistike i druge vrste analitika

Upiti se također mogu razdijeliti na:

- Složene upite (engl. *compound queries*)- upiti koji umotavaju druge složene upite, te kombiniraju njihove rezultate kako bi promijenili ponašanje ili kako bi se prebacili s upita na kontekst filtriranja. Neki primjeri složenih upita su: bool query, boosting, function score, constant score.
- Upite s cjelovitim tekstom (engl. *full text queries*) - omogućuju pretraživanje analiziranih tekstualnih polja, poput tijela e-mailova. Niz upita obrađuje se istom logikom koja je bila primijenjen na polje tijekom indeksiranja, a neki primjeri upita su: intervals, match, common terms, query string, multi match, it.
- Ostale upite - u ostale upite pripadaju drugi razni oblici upita koji su mješavina dviju navedenih skupina ili ne pripadaju po karakteristikama niti jednoj navedenoj. Neki primjeri takvih upita su: joining queries (nested, has child, has parent), shape queries, geo queries (geo bounding, geo distance, geo polygon, geo shape), span queries (span containing, field masking span, span multi, itd.), specialized queries (distance feature, more like this, rank feature, script, itd).




**Slika 2.6:** Vizualni prikaz raspodjele složenih upita [7]

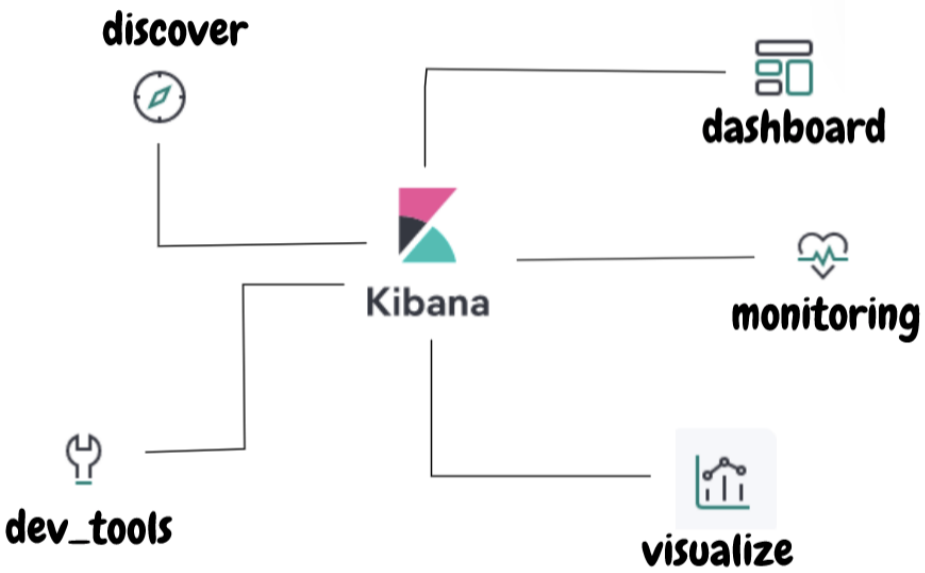
Posebno zanimljivi upiti s vrlo širokom primjenom su složeni upiti čija je raspodjela vidljiva na slici 2.6. Bool upiti najčešće su korišteni složeni upiti koji na upit odgovaraju dokumentima koji odgovaraju logičkim kombinacijama drugih upita. Bool upiti sadrže 4 klauzule: *must, must not, should, filter*. Sve navedene klauzule mogu se kombinirati, a svaka klauzula može sadržavati jedan ili više upita koji određuju njen kriterij. Klauzula *must* definira sve upite (kriterije) po kojima se dokument mora podudarati da bi se vratio kao rezultat, a kriteriji su izraženi u obliku jednog ili više upita. Svi upiti u klauzuli moraju biti zadovoljeni da bi se dokument vratio kao pogodak. Što je više upita u klauzuli povećana je preciznost upita. Klauzula *must not* definira upite (kriterije) s kojima se dokument ne smije podudarati kako bi bio uključen u rezultate pretraživanja. Klauzula *should* dodaje upite (kriterije) koje bi bilo "dobro imati", ali se dokumenti ne moraju podudarati s kriterijem da bi se smatrali pogotcima. Međutim, oni dokumenti koji se podudaraju dobit će višu ocjenu pa će se prikazati više u rezultatima pretraživanja. Klauzula *filter* sadrži upite za filtriranje koji dokumente svrstavaju u kategoriju "da" ili "ne", npr. traženje podataka u određenom vremenskom rasponu pri kojem će neki dokumenti pripadati u taj raspon (da) ili neće (ne). Rezultat pretrage uključuje samo dokumente koji pripadaju u kategoriju da.

Kao i upiti, agregacije su organizirane u tri kategorije:

- Metričke (engl. *metric*) - agregacije metričkih podataka koje izračunavaju metričke vrijednosti, poput zbroja ili prosjeka, iz vrijednosti polja
- Segmentne (engl. *bucket*) - agregacije segmenata koje grupiraju dokumente u segmente, koji se nazivaju i kante, na temelju vrijednosti polja, raspona ili drugih kriterija
- Cjevovodne (engl. *pipeline*) - agregacije cjevovoda koje uzimaju ulaz iz drugih agregacija umjesto dokumenata ili polja

### 2.1.3. Kibana

Kibana je alat koji se koristi za potrebe vizualizacije i upravljanjem podacima. Dizajnirana je u obliku nadzornih ploča koje pružaju mogućnost vizualizacije klastera iz Elasticsearcha. Korisnicima je omogućeno stvaranje histograma, linijskih grafikona, tortnih grafikona i mapa u stvarnom vremenu. Također, omogućen je izvoz datoteka u pdf, png ili CSV oblike i kontroliranje pristupa određenom sadržaju, zasnovanom na ulogama, te stvaranje upozorenja koja koriste pragove temeljene na indeksu i mjernim podacima za slanje obavijesti kada su određeni sadržaji izvan zadanih okvira.




**Slika 2.7:** Organizacija Kibane [8]

Dijelovi Kibane vidljivi na slici 2.7, predstavljaju organizacijske jedinice, odnosno ploče, od kojih se sastoji Kibana: *discover, dashboard, dev tools, monitoring, visualize*.

Na Discover ploči, omogućen je vremenski prikaz i raspodjela indeksa, kao i pojedinosti svakog dijela podatka.

Više vizualizacijskih ploča zajedno se kombinira unutar ploče Dashboard koja objedinjuje više vizualizacijskih prikaza podataka u stvarnom vremenu na jednom mjestu.

Dev tools ploča sastoji se od 3 prikaza: console, search profiler i grok debugger. Konzolni dio koristi se slanje zahtjeva elasticsearchu te prikaz rezultata zahtjeva. Search Profiler koristi se za pojednostavljeni prikaz složenih izlaza iz Elasticsearcha pretvorenih u grafikone koji nam omogućuju učinkovitije uočavanje nepravilnosti. Grok

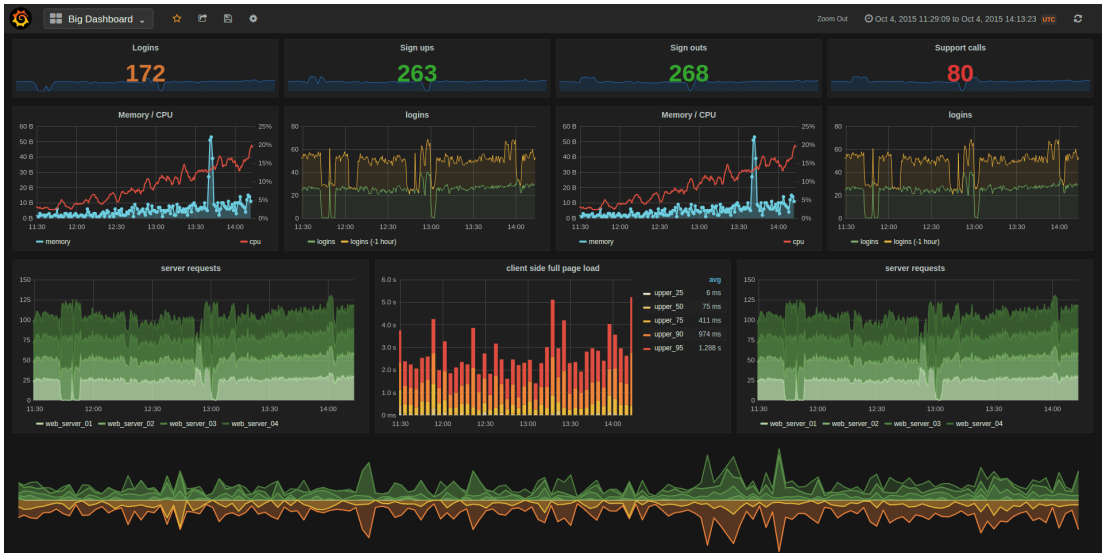
Debugger nudi mogućnost tesitranja napisane grok sintakse koja se koristi za potrebe raščlambe ulaznih log datoteka.

Monitoring ploča prikazuje informacije vezane uz Elasticsearch klastere i indekse, kao što su verzije klastera, memorija na disku koju zauzimaju, broj indeksa, broj krhotina, broj replika, te broj čvorova.

Visualize ploča čini grafički dio Kibane u kojem se nalazi mnoštvo alata za vizualizaciju podataka iz elasticsearcha.

## 2.2. Grafana

Grafana je platforma otvorenog koda za praćenje, analizu i vizualizaciju podataka. Koncipirana je u obliku nadzornih ploča, a nudi mogućnost integracije s raznim izvorima podataka kao što su Graphite, Prometheus,Elasticsearch, Influxdb, AWS Cloud Watch i mnogi drugi. Grafana omogućava postavljanje upita, vizualizaciju putem grafikona, kao što je vidljivo na slici 2.8, korištenje dinamičnih nadzornih ploča, kombiniranje različitih izvora podataka u istom grafikonu, te pretraživanje po filterima. Također, Grafana nudi ugrađeni sistem upozoravanja za promjenu vrijednosti osjetljivih podataka i integraciju sa sustavima kao što je Slack, PagerDuty i slično za slanje obavijesti pri promjeni nekog podatka. Grafanina moć počiva u sposobnosti povezivanja nekoliko izvora podataka na jednoj nadzornoj ploči.




**Slika 2.8:** Primjer prikaza Grafanine nadzorne ploče [9]

### 2.2.1. Usporedba Grafane i Kibane

Grafana i Kibana su platforme otvorenog koda koje služe za vizualizaciju podataka. Razlika između njih je njihova namjena. Grafana je primarno orijentirana i namijenjena za podatke u stvarnom vremenu (engl. time-series), dok je Kibana primarno fokusirana na prikaz logova. Također, Kibana je usko vezana uz rad Elasticsearcha što znači da je limitirana ograničenjima ELK platforme dok Grafana kao ulazne podataka koristi posebno postavljene neovisne i različite izvore. U Kibani su mogućnosti za dijeljenja i izvoz podataka limitirane u usporedbi s Grafanom, jer je dostupna podrška samo za PDF i CSV formate. Grafana također nudi veći izbor i mogućnosti kod vizualizacije od Kibane, jer sadrži veći broj predložaka i dodataka, kao i potpuno prilagodljive nadzorne ploče s ugrađenim mehanizmima za upozoravanje i obavijestima. S druge strane, Kibana ima sposobnost stvaranja vlastitih načina vizualizacije podataka, te učinkovito i jednostavno korisničko sučelje.

# 3. Korištenje alata u analizi Apache logova

## 3.1. Primjena ELK stacka za pohranu podataka o IP adresama

### 3.1.1. Korištenje Logstasha za obradu podataka na strani poslužitelja

#### 3.1.1.1. Instalacija Logstasha

Za rad s Logstashom, potrebno je dohvatiti instalacijski paket sa službene Logstash stranice, pozicionirati se unutar direktorija, te pokrenuti njegovu instancu iz komandne linije, naredbom: "bin/logstash -f logstash-simple.conf". Ako je Logstash cjevovod uspješno pokrenut, na unaprijed definiranim vratima ili pretpostavljenim vratima 9600, vidljiva je povratna informacija o uspješnosti prikazana na slici 3.1.

```
[INFO ][logstash.runner] Starting Logstash {"logstash.version"=>"7.12.1", "jruby.
    version"=>"jruby␣9.2.13.0␣(2.5.7)␣OpenJDK␣64−Bit␣Server␣VM␣[darwin−
    x86_64]"}
[WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file
    because modules or command line options are specified
[INFO ][logstash.agent] Successfully started Logstash API endpoint {:port
    =>9600}
[INFO ][org.reflections. Reflections] Reflections took 73 ms to scan 1 urls,
    producing 23 keys and 47 values
```

**Slika 3.1:** Odgovor Logstasha pri uspješnom pokretanju

#### 3.1.1.2. Podešavanje konfiguracijske datoteke

Pri pokretanju Logstasha, potrebno je izrijekom navesti konfiguracijsku datoteku koju želimo koristiti pri obradi određene vrste podataka kao izvora, kako bi se omogućio ispravan rad Logstasha kao cjevovoda. Unutar konfiguracijske datoteke potrebno je definirati obilježja izvora.

```
input {
        file {
                path => "/Users/mac/Desktop/apache_logs"
                start_position => "beginning"
        }
}

filter {
        if [path] =~ "access" {
                mutate { replace => { "type" => "apache_access" } }
                grok {
                        match => { "message" => "%{
                            COMBINEDAPACHELOG}" }
                        }
                }
                date {
                        match => [ "timestamp" , "dd/MMM/yyyy:HH:mm:ss␣Z
                            " ]
                }
        }

output {
                elasticsearch {
                        hosts => ["localhost:9200"]
                }
                stdout { codec => rubydebug }
        }
```

**Slika 3.2:** Programski kod konfiguracijske datoteke logstash-apache.conf

U konkretnom slučaju, prikazanom u kodu sa slike 3.2, za ulazne podatke koristimo dodatak za čitanje iz datoteka koji omogućuje Logstashu prikupljanje logova iz lokalnog repozitorija za kojeg je zadan relativni put, kao i pozicioniranje čitača na početak. U filter dijelu konfiguracijske datoteke, korišteni su dodaci grok i date. Filter datuma koristi se za raščlanjivanje datuma iz polja. Dobiveni datum bit će korišten kao vremenska oznaka Logstasha za pojedini događaj. Dodatak grok koristi se za parsiranje logova. Pomoću njega, olakšano je raščlanjivanje nestrukturiranih podataka logova u strukturiran oblik nad kojim se mogu vršiti upiti. Konačno, za izlaz, korišten je izlazni dodatak za Elasticsearch, te dodatak codec. Izlazni dodatak za Elsticsearch omogućava direktnu isporuku i preusmjeravanje podataka u lokalnu instancu Elasticsearcha kako bi logovi mogli biti podvrgnuti kasnijoj daljnoj obradi, dok se dodatak codec koristi za dodatni ispis i provjeru podataka.

### 3.1.2. Korištenje Elasticsearcha u pohrani i analizi podataka

#### 3.1.2.1. Instalacija i podešavanje rada Elasticsearcha

Za rad s Elasticsearchom, potrebno je dohvatiti instalacijski paket sa službene Elasticsearch stranice, pozicionirati se unutar direktorija, te pokrenuti njegovu instancu iz komandne linije, naredbom: "bin/elasticsearch". Nakon uspješno pokrenutog jednočvornog klastera, na unaprijed definiranim vratima ili pretpostavljenim vratima 9200, provjeravamo je li Elasticsearch usluga uspješno pokrenuta i poslužuje li zahtjeve.


```
"name:"macs−MacBook−Pro.local",
"cluster_name":"elasticsearch",
"cluster_uuid":"OwhT3F45RE6PGTLq7WDqMA",
"version":{
"number":"7.12.0",
"build_flavor":"default",
"build_type":"tar",
"build_hash":"78722783c38caa25a70982b5b042074cde5d3b3a",
"build_date":"2021−03−18T06:17:15.4101533057",
"build_snapshot":false,
"lucene_version":"8.8.0",
"minimum_wire_compatibility_version":"6.8.0",
"minimum_index_compatibility_version":"6.0.0−beta1"
}
"tagline":"You Know, for Search"
```


**Slika 3.3:** Odgovor Elasticsearch poslužitelja pri pokretanju

Na slici 3.3 prikazan je odgovor poslužitelja sa informacijama o stvorenom klasteru, kao što je ime podešenog elasticsearch korisnika, ime klastera, klasterov identifikacijski broj (uuid), te podaci o verziji.

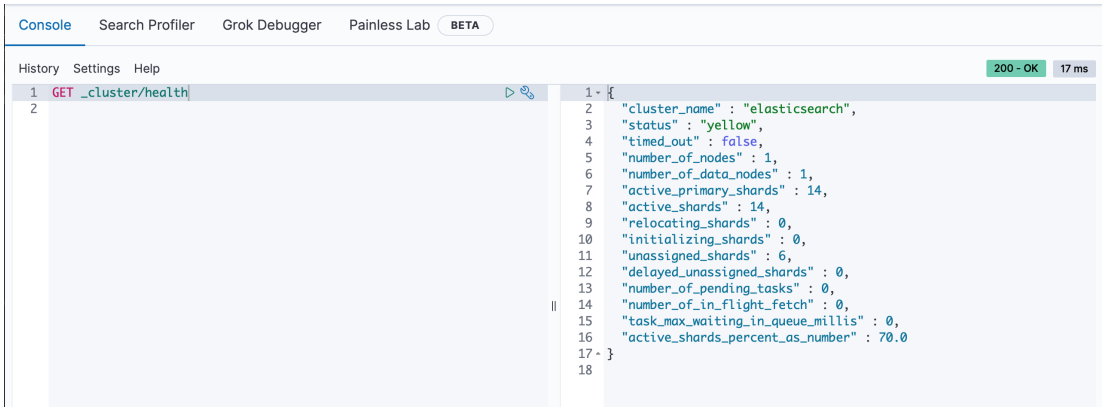
Za promjenu imena čvorova i klastera, kao i dodatna podešavanja u vezi izvora i mjesta pohrane izlaza, sigurnosnih značajki ili upotrebu memorije, potrebno je ažurirati konfiguracijsku datoteku Elasticsearcha na način prikazan na slici 3.4 , nakon čega je potrebno ponovno pokrenuti instancu usluge.

```
// file: config/elasticsearch.yml //
#
# ---------------------------------- Cluster -----------------------------------
#
# Use a descriptive name for your cluster:
#
cluster.name: apache_logs
#
# ------------------------------------ Node ------------------------------------
#
# Use a descriptive name for the node:
#
node.name: node1
#
```

**Slika 3.4:** Podešavanje imena klastera i čvora u konfiguracijskoj datoteci

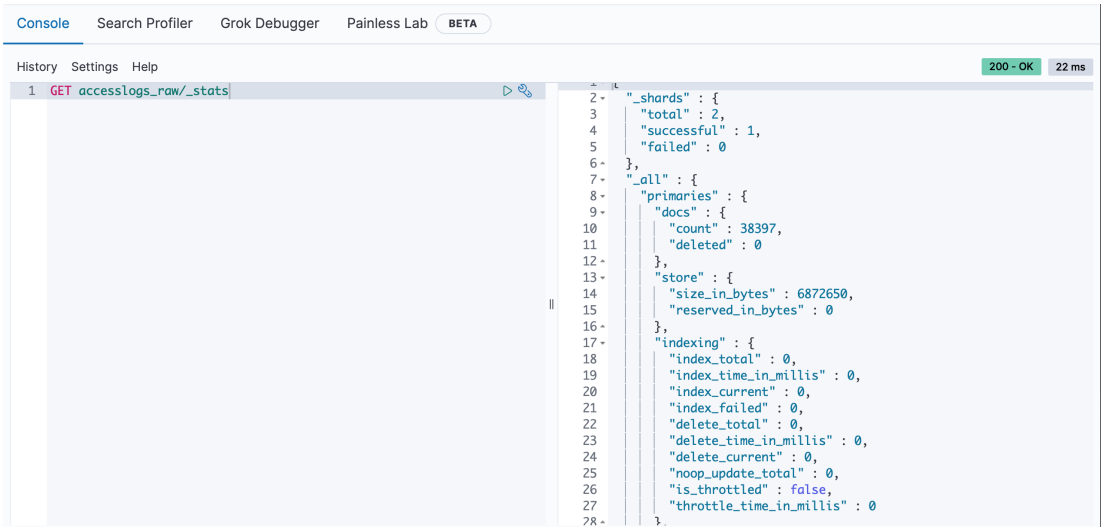
#### 3.1.2.2. Rad s Elasticsearchom

Rad s Elasticsearchom omogućen je kroz Konzolni dio Kibaninog grafičkog sučelja. Konzolni dio omogućuje interakciju s REST API-jem, te slanje zahtjeva Elasticsearchu i pregled odgovora, pregled API dokumentacije te dohvat povijesti svih upućenih zahtjeva.




**Slika 3.5:** Odgovor na zahtjev health nad elastisearch klasterom

Konzolni dio korišten je za CRUD operacije nad čvorovima, a na slici 3.5 vidljiv je primjer korištenja zahtjeva health. Za dobivanje informacija o zdravlju klastera, korišten je "GET cluster/health" zahtjev, kojim provjeravamo stanje klastera, ali i krhotina. Zdravlje klastera može biti: zeleno, žuto ili crveno. Na razini krhotina, crveni status označava da određena krhotina nije dodijeljena klasteru, žuta znači da je dodijeljena primarna krhotina, ali replike nisu, a zelena označava da su dodijeljene sve krhotine. Status razine indeksa kontrolira najlošiji status krhotine, dok status klastera kontrolira najlošiji status indeksa.



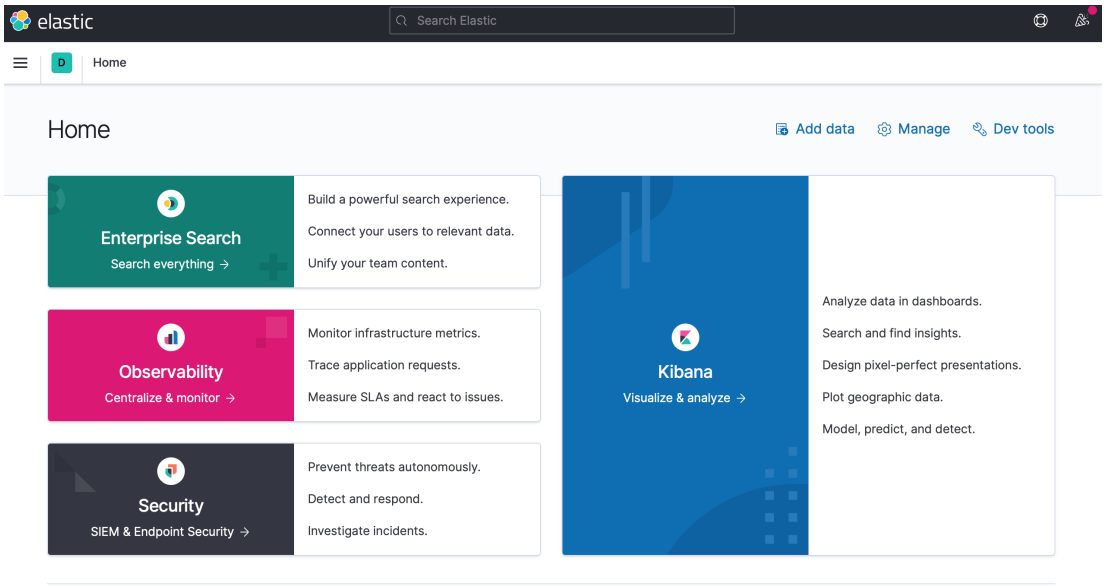

**Slika 3.6:** Odgovor na zahtjev stats nad elastisearch čvorom

Kao što je vidljivo na slici 3.6, za dobivanje informacija o čvoru unutar klastera, korišten je GET nodes/stats zahtjev, kojim provjeravamo pojedinosti o čvoru kao što su: ukupan broj čvorova, broj čvorova koji odgovaraju uvjetima zahtjeva, broj čvorova koji nisu pozitivno odgovorili na zahtjev, ime klastera, podatke o čvoru kao što je ip adresa, poslužitelj, ime, uloga, atributi, poveznice s krhotinama i slično.

### 3.1.3. Korištenje Kibane za vizualizaciju podataka

#### 3.1.3.1. Instalacija Kibane

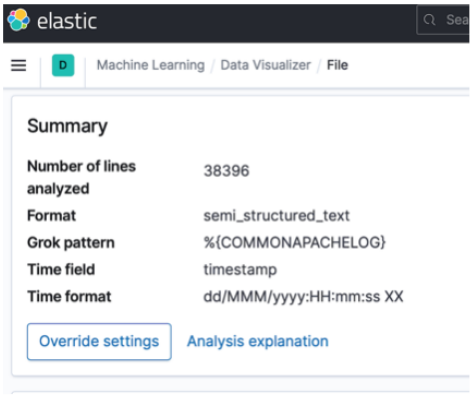
Za rad s alatom Kibana, potrebno je dohvatiti instalacijski paket sa službene Kibanine stranice, pozicionirati se unutar direktorija, te pokrenuti instancu Kibane iz komandne linije, naredbom: "bin/kibana". Nakon uspješno pokrenute usluge, na unaprijed definiranim vratima ili pretpostavljenim vratima 5600, provjeravamo je li Kibana uspješno pokrenuta. Ako je Kibana uspješno pokrenuta, u pregledniku na navedenoj adresi lokalnog poslužitelja, vidljivo je njeno grafičko sučelje kao sa slike 3.7.




**Slika 3.7:** Prikaz Kibaninog sučelja

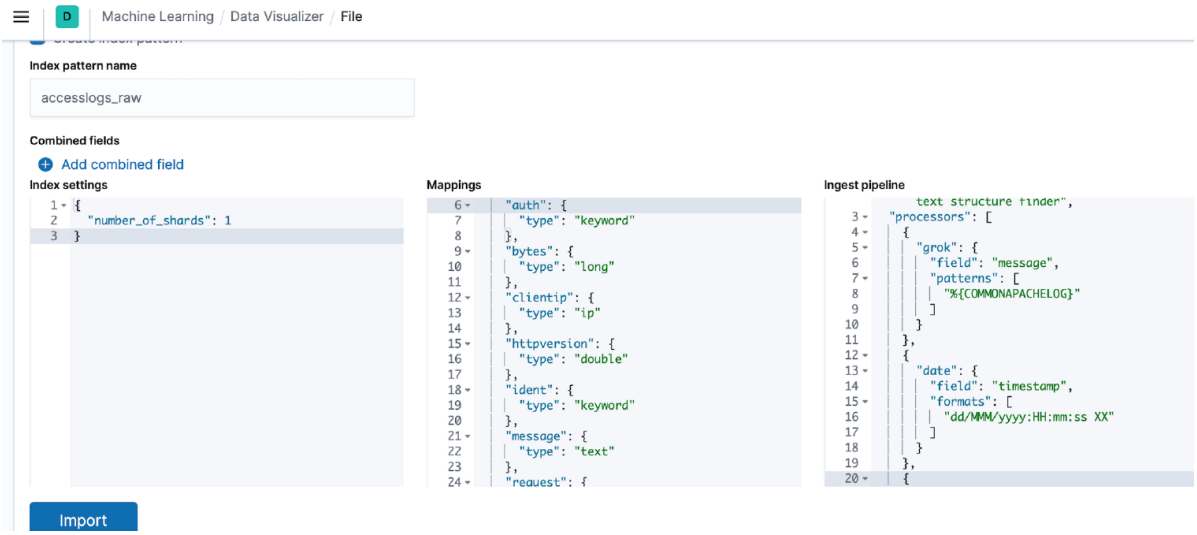
#### 3.1.3.2. Učitavanje podataka izravno u Kibanu

Za učitavanje podataka u sustav ELK, pored korištenja agregacijskog alata kao što je Logstash, podatke je moguće učitati i izravno u Kibanu putem sučelja Data Visualizer. Korištenje Data Visualizera praktično je kada se koristi fiksan skup podataka, a prednost ovakvog načina unosa je izbjegavanje komplikacija pri podešavanju i usklađivanju konfiguracijskih datoteka koje mogu uslijediti kod rada s Logstashom. Data Visualizer koristi metode strojnog učenja za automatsko podešavanje postavki pri kreiranju čvorova i klastera.




**Slika 3.8:** Podešavanje konfiguracijske datoteke u Data Visualizeru

Na slici 3.8 prikazane su automatski generirane postavke za ulazne podatke nastale nakon učitavanja datoteke. Prikazan je broj obrađenih linija, format, korišteni grok uzorak, ime polja koje se koristi kao vremenska oznaka, te format u kojem je pohranjena vremenska oznaka.

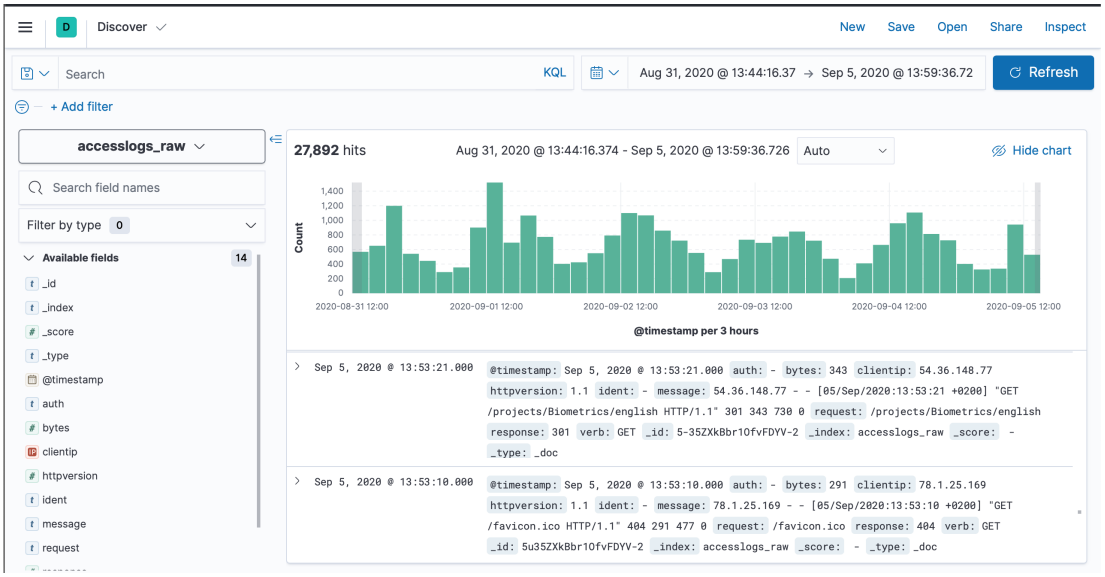



**Slika 3.9:** Podešavanje indeksa u Data Visualizeru

Kod Data Visualizera u Override settings moguće je ručno podešavanje grok patterna, time fielda, formata i ostalog. Na slici 3.9 prikazano je stvaranje indeksa iz Kibane uz podešavanje broja krhotina, mapiranja i naziva, te podešavanje samog cjevovoda.

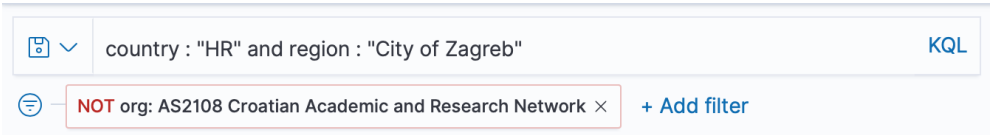
### 3.1.3.3. Analiza i vizualizacija logova

Nakon uspješno unešenih podataka i izvora logova, sav rad s podacima prebacuje se u Kibanino grafičko sučelje Discover. U Discover sučelju prikazanom na slici 3.10 omogućeno je učitavanje bilo kojeg od svih stvorenih indexa, te radni prostor u kojem postoje prostori za dodavanje filtera, pretragu prema ključnim riječima i pravilima Kibana Query Languagea, te biranje vremenskog odsjeka za koji želimo napraviti analizu.



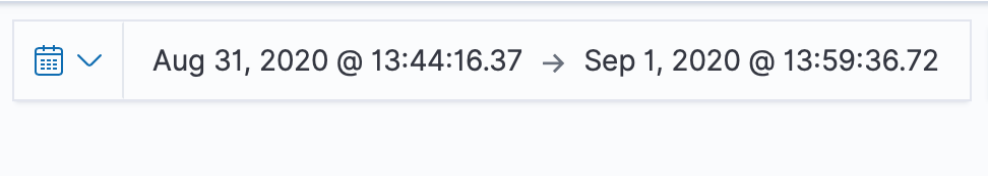

**Slika 3.10:** Prikaz logova u sučelju Discover

KQL upiti omogućavaju pisanje filtera nad podacima unutar proizvoljnog vremenskog intervala. Omogućeno je primjenjivanje više filtera odjednom. Tako je na slici 3.11 prikazana primjena filtera koji traži sve IP adrese koje su povezane sa teritorijem Republike Hrvatske, konkretno Gradom Zagrebom, a koje nisu povezane sa organizacijom CARNET.



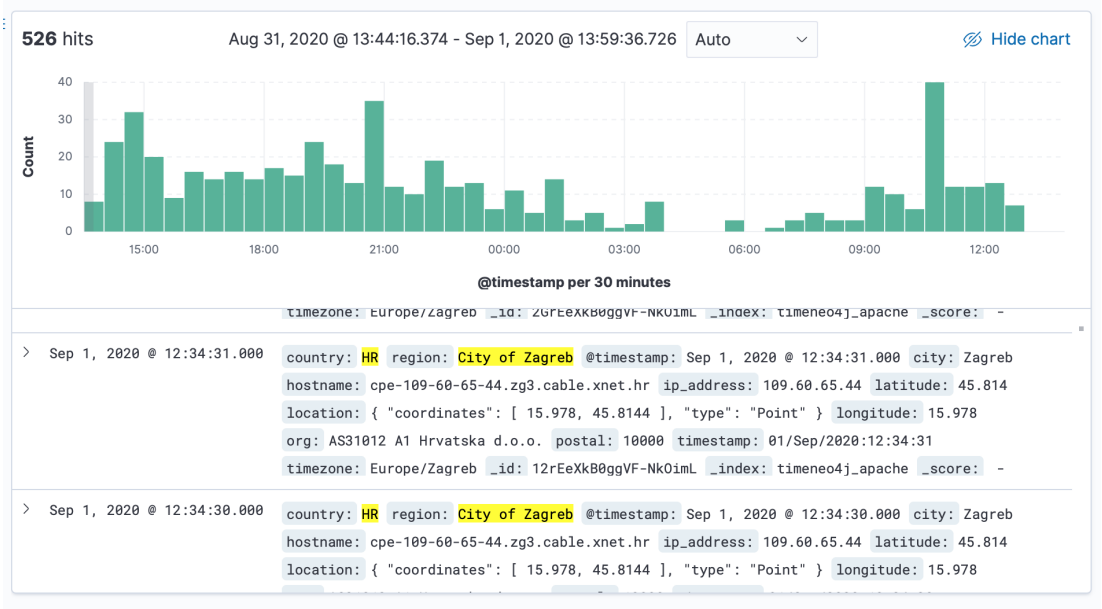

**Slika 3.11:** Primjena filtera nad podacima o IP adresama

Također, kako bi definirali vremenski raspon za navedene podatke, u za to predviđenom prostoru, odabiremo opciju prikaza logova u vremenskom intervalu koji se može detaljizirati do sekunde kao što je vidljivo na slici 3.12.



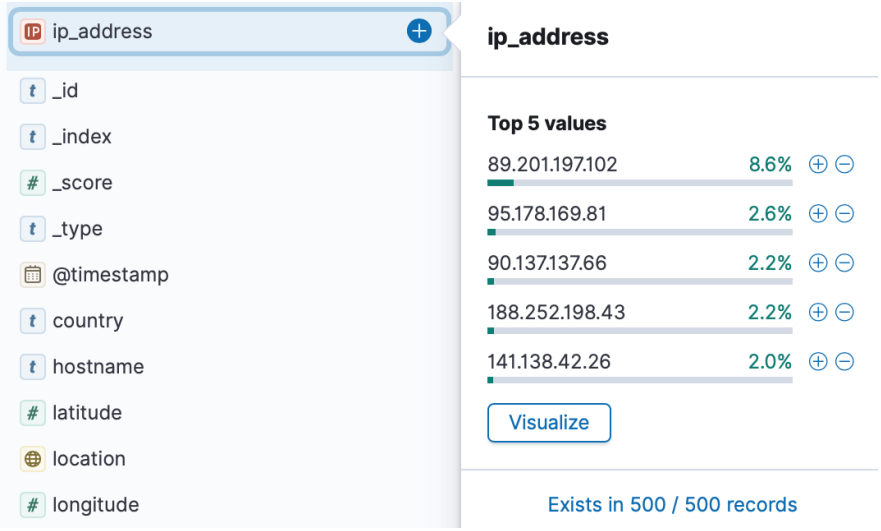

**Slika 3.12:** Prostor za odabir vremenskog raspona

Konačno, na slici 3.13 prikazan je finalni analizirani grafički oblik, nastao primjenom filtera i vremenskog raspona, broja upita raspoređenih unutar uzastopnih 30 minuta, te popis detaljnih podataka o pojedinoj IP adresi sa podacima koji odgovaraju zadanom filteru osjenčanima žutom bojom.



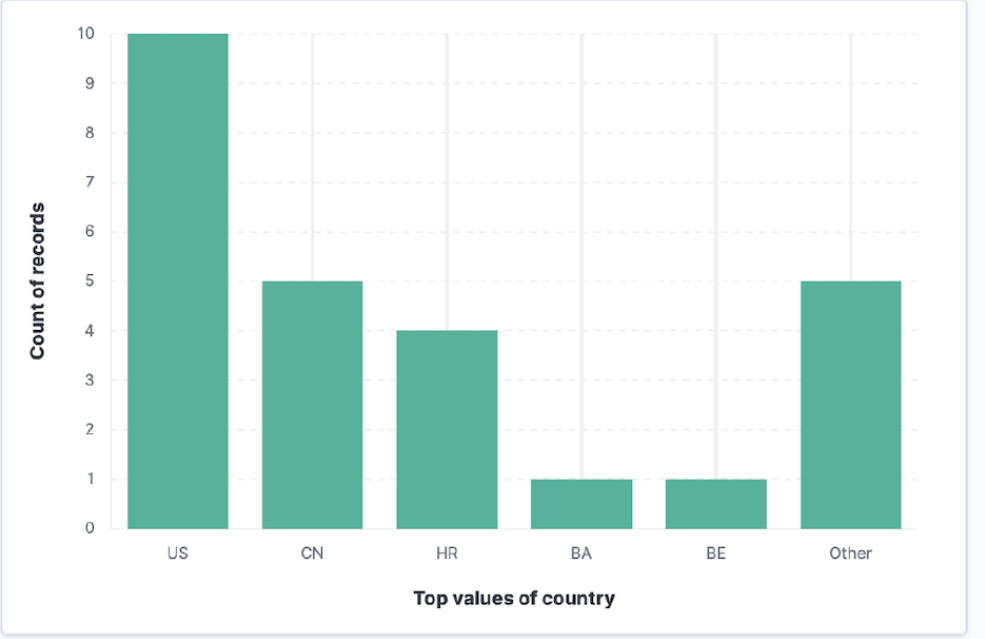

**Slika 3.13:** Prikaz podataka o IP adresama nakon primjene filtera u vremenskom rasponu

Kibana u posebnom izborniku, vidljivom na slici 3.14 , nudi opciju filtiriranja po poljima detektiranim u izvornoj datoteci. Odabirom pojedinog polja, kao što je polje IP adrese automatski je izračunato prvih pet najučestalijih vrijednosti, te je ponuđena opcija za dodatnim podešavanjem vizualizacija, ili analizom podataka na temelju jednog polja.



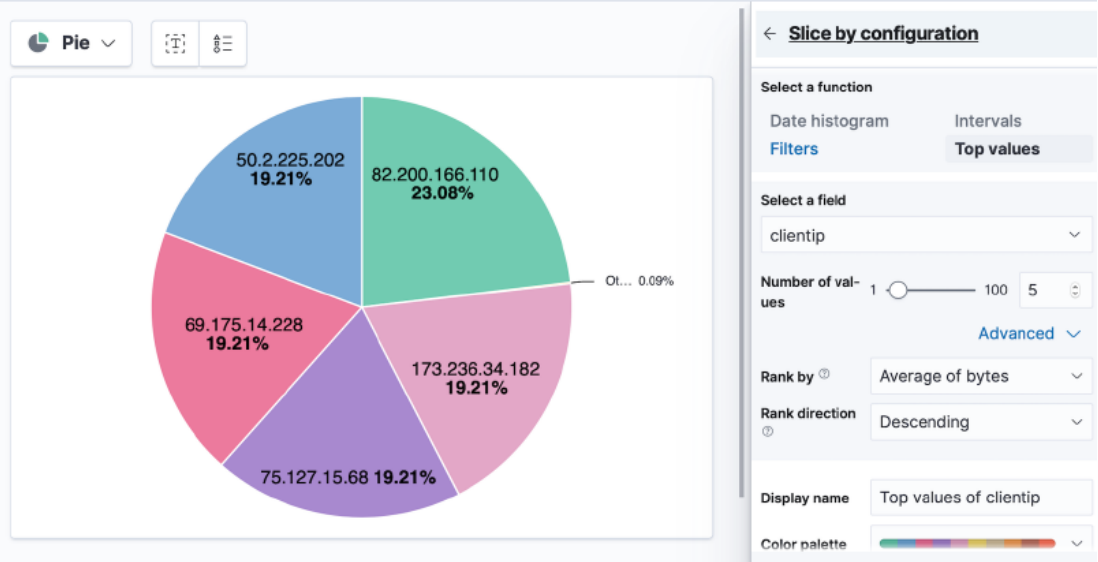

**Slika 3.14:** Opcija vizualizacije po 5 najučestalijih IP adresa

Pomoću Kibaninih vizualizacijskih alata, moguće je stvoriti razne oblike grafičkog prikaza podataka o IP adresama koji su filtrirani iz samih logova. Primjer jednog od grafičkih prikaza je prikaz država s brojem IP adresa iz podešenog izvora, poredanih silazno u stupčastom obliku, vidljivog na slici 3.15.




**Slika 3.15:** Vizualizacija podataka o IP adresama pojedine države

Na slici 3.16 vidljiv je jedan od mogućih prikaza IP adresa sa naglašenim postotkom učestalosti pojavljivanja, poredanih silazno po prosječnom broju bajtova u obliku tortnog grafikona. Nad svim vizualizacijskim grafikonima omogućen je odabir boja, stilova i oblika.




**Slika 3.16:** Prikaz IP adresa u postotku učestalosti pojavljivanja u obliku tortnog grafikona

Za sve prikazne ploče postoji mogućnost izvoza u CSV oblik tablica, a na slici 3.17 prikazana je tablica najviše spomenutih IP adresa i učestalost te je nad svakim skupom upita kroz funkciju median zapisano memorijsko zauzeće u bajtovima.

| Top values of auth | Top values of clientip | Count of records | Median of bytes |
|---|---|---|---|
| - | 66.249.73.135 | 482 | 13,299.5 |
| - | 46.105.14.53 | 364 | 14,872 |
| - | 130.237.218.86 | 357 | 26,289.333 |
| - | Other | 8,796 | 10,763.486 |

**Slika 3.17:** Prikaz IP adresa u postotku učestalosti pojavljivanja u CSV formatu

## 3.2. Korištenje Grafane za prikaz stanja sustava

### 3.2.1. Instalacija Grafane

Za rad Grafane, potrebno je dohvatiti instalacijski paket sa službene Grafanine stranice, pozicionirati se unutar direktorija, te pokrenuti instancu Grafane iz komandne linije, naredbom: "/bin/grafana-server web". Nakon uspješnog pokretanja, na unaprijed definiranim ili pretpostavljenim vratima 3000, provjeravamo radi li Grafanin server.

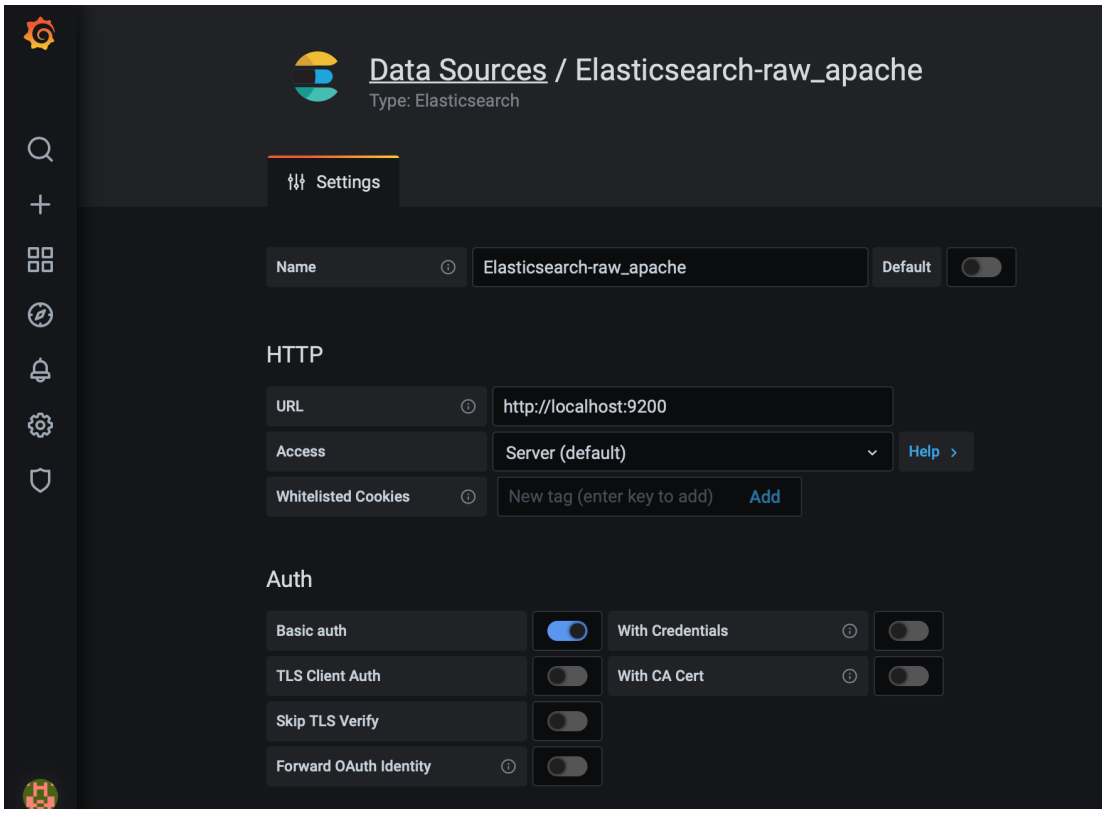
```
INFO[05−29|19:42:32] HTTP Server Listen logger=http.server address=[::]:3000
protocol=http subUrl= sockets
```

**Slika 3.18:** Obavijest o uspješno pokrenutom Grafaninom serveru

Nakon pojave poruke o uspješnom pokretanju, vidljivoj na slici 3.18, potrebno je unijeti podatke za prijavu u sustav. Nakon uspješne prijave, vidljivo je Grafanino sučelje u kojem je moguće dodavati izvore, kreirati panele i još mnogo toga.

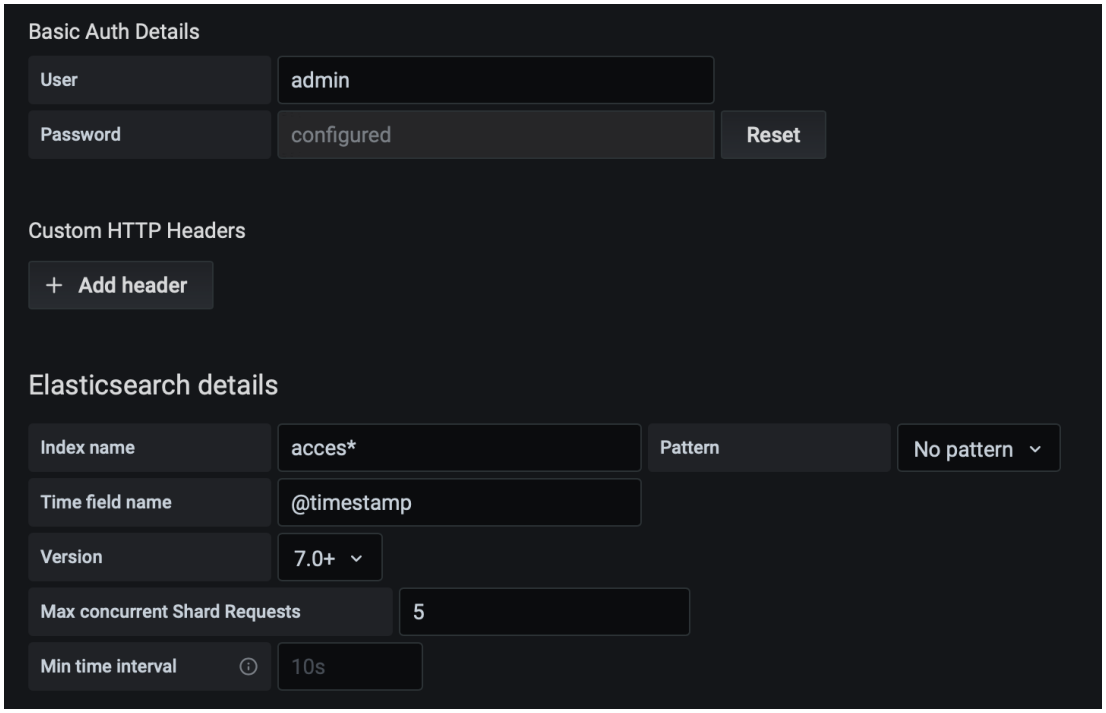
### 3.2.2. Podešavanje izvora podataka

Za potrebe analize logova, potrebno je imati pokrenutu Elasticsearch instancu. Također, potrebno je inicijalizirati izvore podataka koje šaljemo u Grafanu. U konkretnom slučaju, koristiti će se izravna integracija Grafane sa Elastisearchom, koja se može podesiti u Grafaninom sučelju Data sources. Grafana dolazi sa ugrađenom podrškom za Elasticsearch, stoga nije potrebno koristiti posebna proširenja.




**Slika 3.19:** Prikaz podešavanja izvora podataka u Grafani - prvi dio

Na slici 3.19 prikazan je postupak unošenja podataka vezanih uz izvor. Za svaki indeks iz Elasticsearcha potrebno je definirati zaseban Data source u Grafani. Kod podešavanja, zabire se proizvoljno ime izvora podatka, url lokalne instance Elasticsearcha, kao i oblik pristupa - serverski. Također, potrebno je unjeti autentifikacijske podatke o korisniku koji ima pravo pristupa Elasticsearchu.



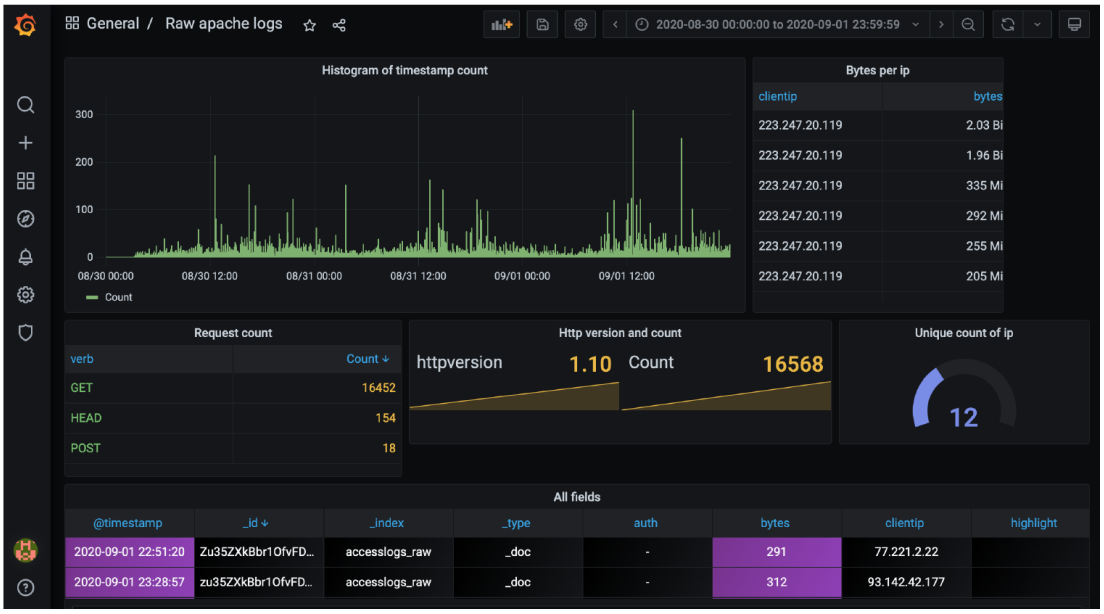

**Slika 3.20:** Prikaz podešavanja izvora podataka u Grafani - drugi dio

U drugom dijelu postavki vezanih za izvor, vidljivih na slici 3.20, popunjavaju se podaci vezani za pojedini indeks. Grafana radi isključivo s vremenskim podacima, stoga je potrebno naglasiti koje polje se koristi za vremensku referencu. Također, potrebno je unijeti ime indeksa, verziju Elasticsearcha, te opcionalno zadani uzorak, maksimalan broj krhotina i minimalni vremenski interval.

### 3.2.3. Analiza i vizualizacija logova

Nakon uspješne integracije sa Elasticsearchom, moguće je kreiranje panela unutar ploča panela.

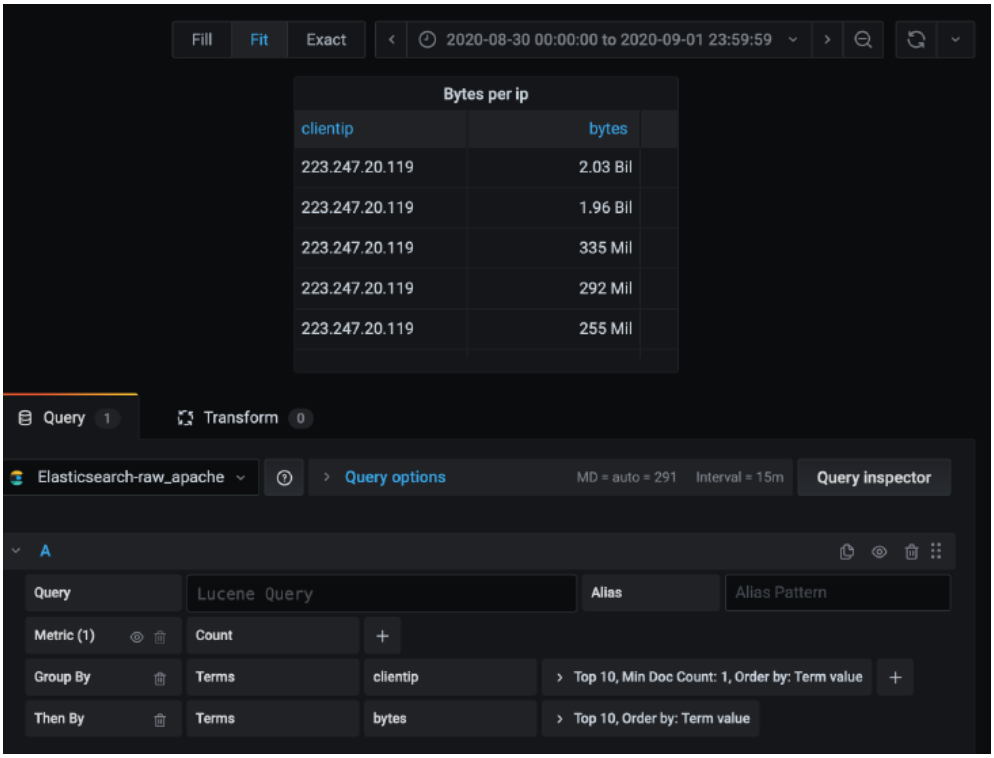
*Grafana dashboards* je zajednica kreatora Grafaninih panela prilagođenih za razne vrste podataka, a iz koje se mogu preuzeti gotovi predlošci panela. Nad pojedinim panelnim pločama, koje se sastoje od više manjih panela, postoji mogućnost podešavanja kontrole pristupa, verzije, editabilnosti, kao i vremenskog intervala svih panela zajedno.




**Slika 3.21:** Izgled ploče panela za Apache logove u Grafani

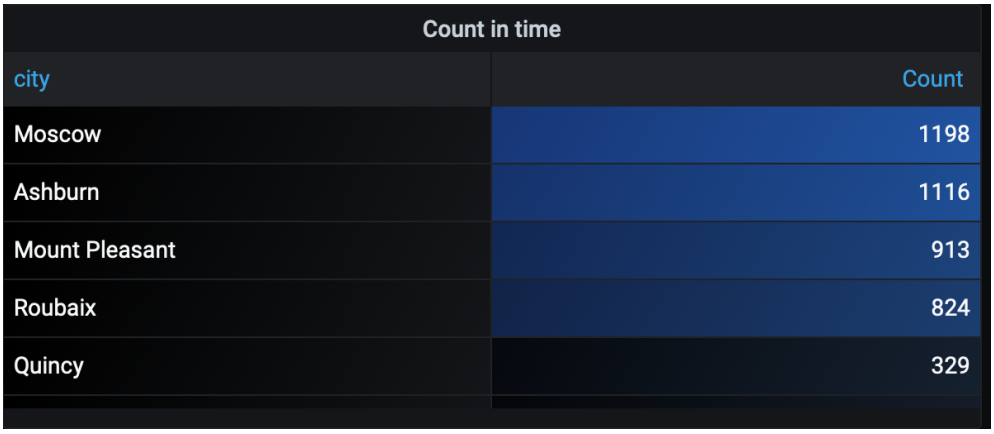
Za potrebe prikaza Apache logova, kreirani su raznovrsni oblici panela u obliku tabličnih prikaza i dijagrama vezanih uz IP adrese, vidljivih na slici 3.21.

Pojedini paneli mogu se uređivati i kreirati kroz pisanje lucene upita za filtriranje, korištenje grupacijskih funkcija, te kroz odabir različitih vrsta vizualizacija.




**Slika 3.22:** Izgled sučelja za uređivanje pojedinog panela

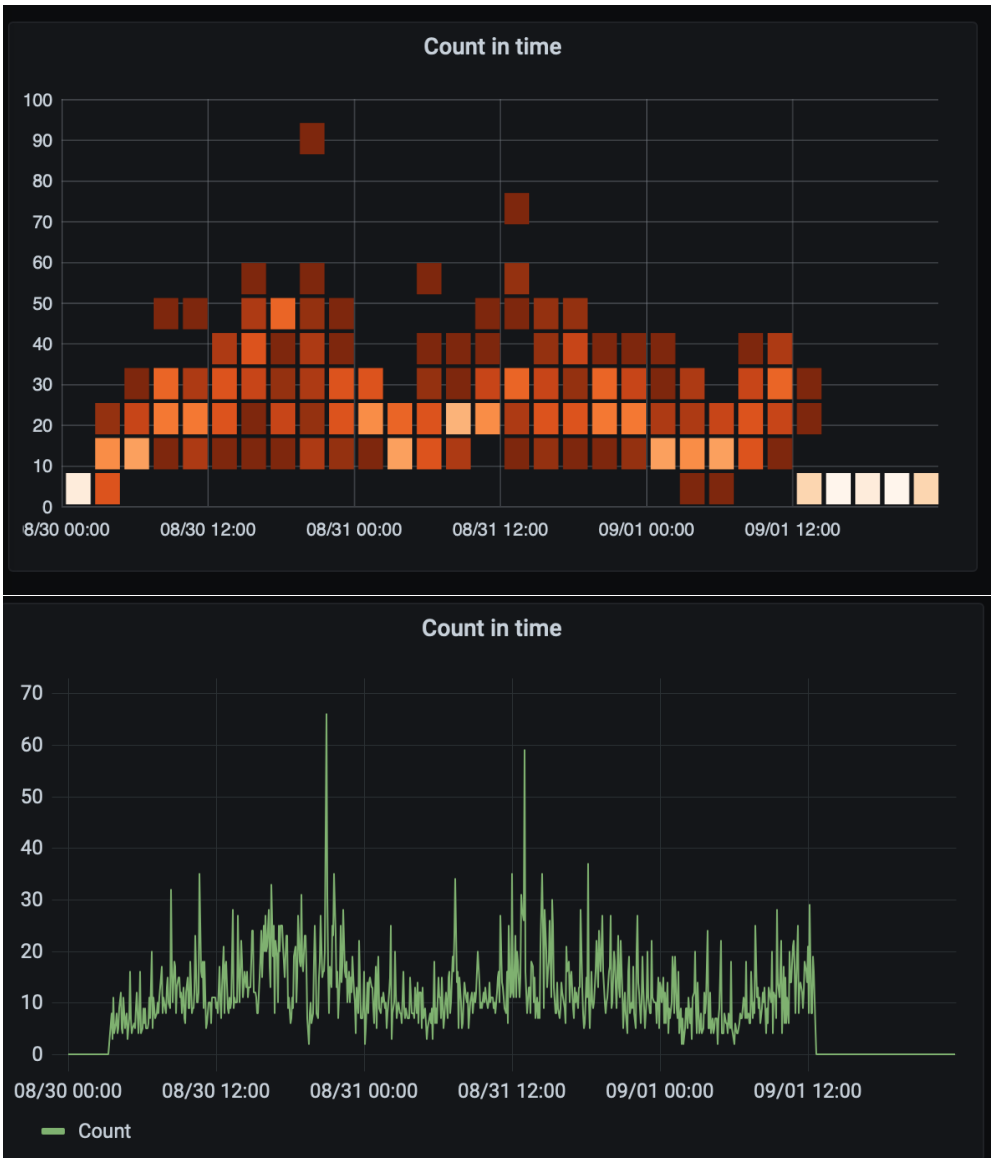
Pri kreiranju tablice za prikaz IP adresa i zbroja bajtova, vidljivoj na slici 3.22, grupiranih po IP adresama, korišteno je grupiranje po polju IP adrese, pri čemu se prikazuje deset IP adresa s najvećim brojem prenesenih bajtova, sortiranih silazno unutar vremenskog intervala od jednog dana.




**Slika 3.23:** Prikaz gradova u tablici po učestalosti pojavljivanja u upitima

Za prikaz gradova po učestalosti uz korištenje različitih vizualizacijskih podešavanja, vidljivih na slici 3.23, korištena je opcija grupacije nad kojom je izvršen upit nad svim podacima izvora. Vizualizacijski alat omogućava isticanje pojedine vrste poda-

taka, pa je na taj način u ovakvom prikazu, svijetlijom nijansom plave boje označeno polje sa većim brojem kako bi se naglasili gradovi koji su bili više spomenuti.




**Slika 3.24:** Dva različita načina prikaza istih podataka izvora u Grafani

Grafana dolazi s nekoliko ugrađenih oblika vizualizacije, kao što su grafikoni, ali veliki broj dodatnih oblika dostupan je korištenjem proširenja za pojedine vrste oblika kao što je tortni grafikon. Različiti oblici prikaza istih podataka poboljšavaju preglednost pri praćenju kretanja pojedinih podataka o IP adresama. Na slici 3.24 prikazana su dva različita načina na koje možemo prikazati broj različitih logova unutar vremenskog intervala. Također, Grafana nudi opcionalnu značajku kreiranja upozorenja kada određene vrijednosti premaše zadane iznose, što omogućava konstantno kvalitetno praćenje logova unutar vremenskog intervala.

# ZAKLJUČAK

U ovom radu dan je pregled tehnologija i alata za agregaciju i vizualizaciju podataka, te njihova primjena u rješavanju problema analize logova. Opisan je način analize Apache logova u ELK platformi, te metode vizualizacije logova u alatu Grafana.

Pregledom širokog spektra mogućnosti koje nude alati poput Logstasha, Elasticsearcha i Kibane, utvrđene su prednosti takvog sustava za pohranu i analizu IP adresa. Mogućnost agregacije više izvora podataka u jedan skup vrijednosti koji može automatski osvježavati podatke snažan je argument za korištenje Logstasha kao cjevovoda. Brzina pretraživanja velike količine podataka u sekundama i stvarnom vremenu, te skalabilnost Elasticsearcha, važne su prednosti kod analize logova. Široki raspon alata za vizualizaciju koje nudi Kibana, te integracija sa sustavom Elasticsearch, omogućava lakše praćenje i razumijevanje kretanja trendova u velikim količinama podataka. S druge strane, mogućnost integracije alata Grafane sa različitim izvorima podataka, te orijentiranost na raznovrsnost pri odabiru vizualizacije i slobodu prilagodbe, čine Grafanu oštrim konkurentom prethodno spomenutoj Kibani.

Prostora za napredak uvijek postoji pa se tako analiza Apache logova u navedenih alatima može pospješiti uporabom više vanjskih biblioteka, te pisanjem vlastitih proširenja za pojedine aspekte vizualizacije i agregacije. Posebnu pažnju poželjno je posvetiti samoj integraciji velikog i pomalo nezgrapnog ELK sustava u postojeće administrativne dijelove stranica kao što je Django Admin gdje bi njihova primjena zaživjela u punom potencijalu. Također poželjno, bilo bi omogućiti vizualizacijskim alatima kao što su Kibana i Grafana integraciju u postojeće sustave i aplikacije, u kojima ne bi postojali samo kao statički dijelovi stranice već kao dinamički prikazi podataka u stvarnom vremenu.

# LITERATURA


1. Službene stranice Sigurnosne agencije za internetsku sigurnost, dostupno: https://us-cert.cisa.gov/ics/content/cyber-threat-source-descriptions, posjećeno: 30.5.2020
2. Službena Elasticsearch dokumentacija, dostupno: https://www.elastic.co/guide/index.html, posjećeno: 30.5.2020
3. Službena Logstash dokumentacija, dostupno: https://www.elastic.co/guide/en/logstash /current/index.html, posjećeno: 30.5.2020
4. Službena Kibana dokumentacija, dostupno: https://www.elastic.co/guide/en/kibana/index.html, posjećeno: 30.5.2020
5. Službena dokumentacija Grafane, dostupno: https://grafana.com/docs/grafana/latest/ , posjećeno: 30.5.2020

IZVORI SLIKA

1. DNS stuff internetski portal, dostupno: https://www.dnsstuff.com/what-is-threat-intelligence, posjećeno: 30.5.2020
2. Logstail internetski portal, dostupno: https://logstail.com/blog/how-to-analyze-apache-logs-with-elk-stack-and-logstail-com/, posjećeno: 30.5.2020
3. My soft key internetski portal, dostupno: https:// www.mysoftkey.com/nosql/elastic-search-logstash- and-kibana-is-know-as-elk-stack/ , posjećeno: 30.5.2020
4. Članak sa Medium internetskog portala, dostupno: https://ipindersinghsuri.medium.com/noobs-guide-to-logstash-10adfdbe742c , posjećeno: 30.5.2020
5. Članak sa Hitachi internetskog portala, dostupno: https://community.hitachivantara.com/s /article/search-the-inverted-index , posjećeno: 30.5.2020

6. Članak sa Packt internetskog portala, dostupno: https://subscription.packtpub.com/book/data/9781789957 754/1/ch01lvl1sec04/elasticsearch-architectural-overview , posjećeno: 28.4.2020

7. Članak sa Medium internetskog portala, dostupno: https://medium.com/elasticsearch/introduction-to-elasticsearch-queries-b5ea254bf455 , posjećeno: 30.5.2020

8. Članak sa Medium internetskog portala, dostupno: https://medium.com/analytics-vidhya/kibana-brings-the-data-to-life-7c0e528507c0 , posjećeno: 30.5.2020

9. Članak sa 8bitmen internetskog portala, dostupno: https://www.8bitmen.com/what-is-grafana-why-use-it-everything-you-should-know-about-it/ , posjećeno: 30.5.2020

## Primjena alata ElasticSearch, LogStash i Kibana za analizu podataka o IP adresama


### Sažetak

Cilj promatranja i pomne analize alata za agregaciju i vizualizaciju podataka u kontekstu logova i podataka o IP adresama, je njihova primjena u vrsti obavještajnog rada koji se fokusira na zaštitu od kibernetičkih prijetnji. U ovom radu isprobane su mogućnosti korištenja alata Logstash i Elasticsearch za agregaciju i analizu podataka o IP adresama dobivenim iz Apache logova, a pomoću alata Kibana i Grafana promotrene su postojeće opcije i alati koji se koriste u vizualizacijske svrhe. Analizom mogućnosti i karakteristika sustava i alata, utvrđene su prednosti korištenja takve vrste sustava za pohranu i analizu IP adresa.



**Ključne riječi:** kibernetičke prijetnje, obavještajni rad, Apache logovi, IP adresa, Grafana, Kibana, Elasticsearch, Logstash


## Application of ElasticSearch, LogStash and Kibana tools for analysis of IP addresses


### Abstract

The goal of observation and careful analysis of data aggregation and visualization tools in the context of logs and IP address data is their application in a type of intelligence work that focuses on protection against cyber threats. In this paper, the possibilities of using Logstash and Elasticsearch tools for aggregation and analysis of IP address data obtained from Apache logs are tested, and with the help of Kibana and Grafana tools, the options and tools used for visualization purposes are observed. By analyzing the capabilities and characteristics of systems and tools, the advantages of using such types of storage systems and IP address analysis have been identified.



**Keywords:** intelligence, cyber threats, Apache logs, IP address, Grafana, Kibana, Elasticsearch, Logstash