Sistemski nadzor s ELK stack alatima

Šaler, Matija

VELEUČILIŠTE SUVREMENIH INFORMACIJSKIH TEHNOLOGIJA

STRUČNI PRIJEDIPLOMSKI STUDIJ INFORMACIJSKIH TEHNOLOGIJA

Matija Šaler

ZAVRŠNI RAD

SISTEMSKI NADZOR S ELK STACK ALATIMA

Zagreb, listopada 2024.

Veleučilište suvremenih informacijskih tehnologija
10000 Zagreb, Ulica Vjekoslava Klaića 7

Studij:	Stručni prijediplomski studij informacijskih tehnologija smjer računalni sustavi i mreže
Student:	**Matija Šaler**
Matični broj:	2021072

Zadatak završnog rada

Predmet:	Programski alati na UNIX računalima
Naslov:	**Sistemski nadzor s ELK stack alatima**
Zadatak:	Cilj rada je objasniti način rada i mogućnosti 3 softverska projekta (Elasticsearch, Logstash i Kibana) koji zajedno čine tzv. "ELK stack". On se koristi za nadzor i analizu sustava u stvarnom vremenu, dugoročnu pohranu zapisa i vizualizaciju prikupljenih podataka.
Mentor:	Ivan Capan, pred.
Zadatak uručen kandidatu:	9.7.2024.
Rok za predaju rada:	29.10.2024.
Rad predan:	_____

Povjerenstvo:

Dalibor Bužić, v. pred.	član predsjednik	_____
Ivan Capan, pred.	mentor	_____
Edmond Krusha, v. pred.	član	_____

SADRŽAJ

POPIS SLIKA

POPIS KODOVA

1. UVOD

Pouzdanost i performanse IT infrastrukture su već dugi niz godina kritične za poslovanje. Većina digitalnih usluga danas jednostavno ne trpi prekid u radu što je jedan od razloga zašto je učinkovit sistemski nadzor tako važan za održavanje infrastrukture. Dobro uspostavljen nadzor može omogućiti sistemskim administratorima i inženjerima da uoče i riješe probleme prije nego prerastu u probleme koji mogu izazvati štetu raznih oblika.

Upravljanje sistemskim dnevnicima je gotovo uvijek nezanimljiv i vremenski zahtjevan posao za sve koji se bave upravljanjem računalnim sustavima. U pravilu svaka aplikacija, servis, uređaj ili poslužitelj ima svoje dnevnike koje zapisuje na određenu lokaciju i u određenom formatu. Ručno pretraživanje dnevnika koristeći alate kao što su tail, grep ili čak automatizirano pretraživanje uz pomoć bash ili PowerShell skripti može oduzeti veliku količinu vremena i ne mora nužno pružiti uvid u sve potrebne informacije. Potrebno je nešto što može brzo i učinkovito parsirati dnevnike. Isto tako često postoji potreba za centraliziranim nadzorom infrastrukture koja može biti raspršena na više prostorija na lokaciji, više fizičkih lokacija ili čak kontinenata. Postoji više izravnih i neizravnih načina za rješavanje ovog problema.

Jedno od najčešće korištenih rješenja su tzv. *Elastic Stack* alati*,* poznatiji pod nazivom *ELK stack* alati, koji se sastoje od sljedećih komponenti: *Elasticsearch, Logstash, Kibana i Beats.* Elasticsearch daje mogućnost indeksiranja i pretraživanja podataka, Logstash služi za obavljanje kompleksnih radnji nad podacima kao što je integracija i transformacija podataka iz više izvora, a Kibana služi za analizu prikupljenih podatka. Osim Logstasha često se koristi i Beats koji je također alat za prikupljanje i slanje podataka, ali ima znatno manje hardverske zahtjeve što ga čini pogodnijim za prikupljanje podataka iz krajnjih točaka sustava.

Zajedno oni čine opsežnu platformu za prikupljanje, indeksiranje, analizu i vizualizaciju sistemskih dnevnika. ELK alati omogućavaju visoku razinu fleksibilnosti i skalabilnosti zbog čega su veoma popularan izbor među organizacijama koje traže rješenje za ovaj problem.

Na aktualnost ove teme ukazuje i činjenica da većina ljudi korištenjem raznih usluga dostupnih na internetu najčešće i ne znajući dolazi u dodir s ELK alatima. Razne Samsungove usluge, Uber, Shopify i GitHub samo su neki od široko rasprostranjenih

servisa koji za pretraživanje i prikazivanje podataka krajnjem korisniku koriste ELK alate.

Cilj ovog rada je ući dublje u pojedine komponente ELK alata i istražiti njihove funkcionalnosti. Prikazan je postupak instalacije i konfiguracije ELK alata i njihova praktična primjena unutar WSL okoline, odnosno *Windows Subsystem for Linux.* Ovaj alat omogućava korisniku da na istom hardveru na kojem se pokreće Windows operacijski sustav instalira i samostalnu Linux distribuciju. Korištenje WSL-a daje neke jedinstvene prednosti u odnosu na klasičnu virtualizaciju kakva se često koristi za implementaciju ovakvih sustava. Jedna od prednosti su manji sistemski resursi potrebni za pokretanje oba operacijska sustava, što omogućava lakše pokretanje ELK alata koji su u pravilu predviđeni za pokretanje na poslužiteljskim računalima koja imaju puno više procesorske snage, memorije kao i mjesta za pohranu u odnosu na komercijalno dostupna prijenosna računala.

2. ELASTIC STACK

Elastic stack, poznatiji po starijem nazivu ELK stack sastoji se od tri, odnosno četiri glavne komponente. Kibana, Elasticsearch, Beats i Logstash zajedno mogu preuzeti podatke iz bilo kojeg izvora u bilo kojem formatu, analizirati ih, izdvojiti relevantne informacije i vizualizirati ih u stvarnom vremenu. Radi se o alatu otvorenog koda, što ga čini veoma privlačnim svima od sistemskih inženjera do programera koji rade s podacima. Najpopularnija područja primjene su dnevnički zapisi, metrike, sigurnosna i poslovna analitika.

Jedan od najčešćih slučajeva primjene je praćenje događaja na poslužiteljskim računalima. Ukoliko dođe do usporavanja usluge do te mjere da to primijete krajnji korisnici, Elastic stack se može koristiti kako bi se otkrio razlog i otklonile moguće greške u radu sustava.

2.1 Elasticsearch

Elasticsearch je distribuirani program za pretraživanje i analitiku u srcu ELK stacka. Predviđen je za rad s velikom količinom podataka uz visoke performanse i mogućnosti skaliranja. Razvijen je još 2010. godine i postao je jedan od najpopularnijih rješenja za pretraživanje zbog velikog broja mogućnosti koje nudi i jednostavnosti upotrebe.

Temelji se na Apache Lucene biblioteci otvorenog koda koja pruža mogućnosti indeksiranja i pretraživanja. Napravljen je da radi kao distribuirani sustav što znači da može skalirati horizontalno i vertikalno i osigurati neometani rad bez obzira na logičku i fizičku konfiguraciju. Ovakva arhitektura je zapravo ono što omogućava Elasticsearchu da učinkovito upravlja i pretražuje goleme količine podataka.

Za interakciju Elasticsearch koristi *RESTful* API u obliku JSON dokumenta. To omogućava korisnicima da indeksiraju, pretražuju i upravljaju podacima putem HTTP poziva. Podaci u Elasticsearchu su strukturirani u indekse koji se sastoje od većeg broja dokumenata sličnih karakteristika. Svaki od tih dokumenata je zapravo JSON objekt koji sadrži podatke za indeksiranje i pretraživanje.

2.1.1 Aplikacijska arhitektura

Indeks unutar Elasticsearcha se ponaša slično kao tablica u klasičnoj relacijskoj bazi podataka. Sadrži skup zapisa, odnosno dokumenata koji imaju slične karakteristike. Primjerice, indeks može sadržavati dnevničke zapise raznih aplikacija ili podatke o temperaturi procesora.

Kako bi učinkovitije upravljao velikim količinama podataka, Elasticsearch indekse na manje segmente (engl. *shard*). Svaki *shard* je zaseban indeks koji se može pohraniti na bilo kojem „čvoru“ unutar „skupine“. *Shard* služi za distribuiranje hardverskog opterećenja na način da paralelizira procesuiranje podataka i pretraživanje.

Čvor (engl. *node*) je pojedina instanca Elasticsearcha unutar skupine. Svaki čvor sadrži podatke i sudjeluje u indeksiranju i pretraživanju koje se odvija unutar skupine. Čvorovi međusobno komuniciraju kako bi osigurali konzistentnost podataka i koordinirali pretraživanje.

Skupina (engl. *cluster*) se sastoji od jednog ili više čvorova koji rade zajedno kako bi pružili jedinstven pregled podataka. Skupina je mehanizam preko kojeg Elasticsearch može skalirati horizontalno dodajući još čvorova kako raste količina podataka ili potreba za pretraživanjem. Ujedno pružaju i otpornost na pogreške jer se podaci repliciraju po čvorovima.

2.1.2 Način rada Elasticsearcha

Elasticsearch koristi nekoliko veoma složenih mehanizama za pretragu i indeksiranje u kojima se skriva tajna njegovih visokih performansi i fleksibilnosti. Izuzetno je pogodan za analitiku u stvarnom vremenu zbog čega je jako vrijedan alat za sistemski nadzor. Zbog brzog indeksiranja i pretraživanja pruža administratorima mogućnost da detektiraju anomalije i otklone probleme u stvarnom vremenu.

2.1.3 Indeksiranje

Indeksirani podaci u Elasticsearchu se pohranjuju u dokumentu u JSON formatu. Svaki dokument ima polja koja predstavljaju različite atribute ili dijelove podataka. Invertirani indeks je podatkovna struktura unutar Elasticsearcha koja mapira termine na dokumente u kojima se pojavljuju. To omogućava brzo pretraživanje cjelokupnog teksta lociranjem dokumenata u kojima se nalaze. Analizatori obrađuju tekstualna polja tijekom indeksiranja i pretrage. Tekst se lomi na *tokene*, normalizira i filtrira. Neki od primjera su uklanjanje čestih riječi kao što su članovi u engleskom jeziku ili neki veznici ili pretvorba svih velikih slova u mala.

2.1.4 Pretraživanje

Elasticsearch koristi poseban jezik za pretraživanje, tzv. *Query DSL (*engl. *Domain Specific Language)* koji omogućava korisniku da kreira veoma kompleksne upite. Ti upiti mogu direktno tražiti termine, raspone podataka ili čitav tekst. Osim pretraživanja moguće je i agregirati podatke što otvara mogućnosti kompleksnih analiza i izračuna

raznih metrika kao što su prosjeci, zbrajanja i prebrojavanja, a moguće je i grupirati podatke s obzirom na razne kriterije. Jedna od ključnih mogućnost Elasticsearcha je pružanje rezultata gotovo u stvarnom vremenu što je od kritične važnosti za aplikacije koje zahtijevaju najnovije informacije.

2.1.5 Upotreba Elasticsearcha u sistemskom nadzoru

Elasticsearch može agregirati dnevničke zapise iz različitih izvora u centralni repozitorij. To ubrzava upravljanje podacima bez obzira na njihov izvor. Elasticsearch se može koristiti i s alatima za strojno učenje za detekciju svakog odstupanja od očekivanog ponašanja sustava. To omogućava identifikaciju problema prije nego počnu utjecati na performanse sustav. Dojava problema u stvarnom vremenu može obavijestiti administratore o kritičnim događajima. Primjerice, notifikacije se mogu početi javljati u trenutku kad sistemske pogreške prijeđu određeni prag ili kada se pojave određeni obrasci u dnevnicima. Analiza historijskih podataka je također jedan od važnih alata za sistemski nadzor, ali i za planiranje kapaciteta i za razumijevanje ponašanja sustava.

2.2 Logstash

Logstash je alat za protočnu obradu podataka koji ima ključnu ulogu u prikupljanju, transformaciji i prijenosu podataka u ELK alatima. Logstash je osmišljen tako da može primiti podatke iz velikog broja izvora, napraviti nad njima potrebne transformacije i proslijediti ih Elasticsearchu na indeksiranje i analizu.

Logstash je alat koji ima veoma široke mogućnosti primjene. Kao i Elasticsearch može upravljati velikom količinom podataka iz raznih izvora, kao što su dnevnički zapisi, metrike ili aplikacije. Njegova osnovna zadaća je da služi kao centralizirani mehanizam za procesuiranje podataka koji može pripremiti i normalizirati podatke za pohranu i analizu. To se ostvaruje kroz njegovu arhitekturu za protočnu obradu podataka koja omogućuje korisnicima da sami definiraju protok podataka kroz različite faze obrade.

Njegova najosnovnija funkcionalnost je prikupljanje podataka iz raznih izvora, transformacija prema određenim pravilima i dostava tih podataka na drugu lokaciju koja može biti Elasticsearch, neka baza podataka ili čak druga instanca Logstasha. Taj proces omogućava velikim organizacijama da na unificiran način agregiraju i upravljaju raznovrsnim podacima.

Logstash je dakle veoma važna komponenta koja služi kako bi prikupljeni podaci bili ispravno formatirani prije nego stignu u Elasticsearch. Upravljanje podacima već u fazi

prikupljanja je bitno jer se mogu filtrirati nepotrebne informacije, primijeniti transformacije i obogatiti prikupljene podatke vremenskim ili geografskim oznakama.

2.2.1 Protočna obrada podataka

Princip protočne obrade podataka (engl. *data pipeline*) znači strogo definirana pravila za procesuiranje podataka od ulaza u sustav do izlaza. Logstash se može zamisliti kao niz cijevi s postajama koje rade zajedno kako bi prikupile i usmjerile podatke. Ove komponente se mogu podijeliti u 3 primarne kategorije: podatkovni ulaz, izlaz i filteri.

2.2.2 Ulaz i izlaz podataka

Izvori iz kojih Logstash može primati podatke uključuju sve od dnevnika, mrežnih priključaka, serijskih podataka iz senzora, baza podataka ili API-eva. Postoji i velik broj dodataka koji služe kao sučelje prema gotovo svakom podatkovnom izvoru. Primjeri dodataka su Kafka, Syslog i File, koji omogućavaju redom ulaz podataka iz Apache Kafka servera, prikupljanje dnevnika iz poslužitelja i čitanje podataka iz datoteka.

Izlaz podataka može biti bilo koja komponenta koja može trajno ili privremeno pohraniti podatke nakon što su prošli kroz filtere. U kontekstu ELK alata se u pravilu radi o Elasticsearchu.

2.2.3 Filteri

Jedan od najčešće korištenih filtera je Grok, koji omogućava parsiranje nestrukturiranih podataka u strukturirana polja koristeći regularne izraze (engl. *regular expression* ili *regex*). Posebno je koristan u slučajevima kad je potrebno izolirati korisne informacije iz dnevničkih zapisa. *Date* filter se koristi za parsiranje vremenskih oznaka i standardizaciju njihova formata, kako bi zapisi i poruke prikupljene iz raznih izvora bile sinkronizirane.

2.2.4 Integracija s podatkovnim izvorima

Direktna integracija s raznim izvorima je jedna od najvažnijih karakteristika Logstasha. Konfiguracijom ulaza podataka moguće je pratiti dnevničke zapise u stvarnom vremenu, onako kako se zapisuju u datoteke. To je iznimno korisno u poslužiteljskom okruženju gdje se dnevnici generiraju i pohranjuju lokalno ili u mrežne podatkovne sustave. Druga važna mogućnost integracije je ona s redovima poruka (engl. *message queue)* poput Apache Kafke. Logstash je moguće integrirati tako da prihvaća podatke u stvarnom vremenu što omogućava praćenje informacija o događajima, metrike i zapise koji mogu pružiti uvid u ponašanje sustava. S obzirom na današnju raširenost računalnih sustava u oblaku važna je i integracija s mrežnim protokolima. Važno je spomenuti da su podržani

protokoli HTTP i TCP/UDP, što omogućava prikupljanje podataka izravno iz mrežnih uređaja, poslužitelja ili aplikacija koristeći *syslog* standard.

2.3. Kibana

Kibana je alat otvorenog koda za vizualizaciju podataka, primarno usmjeren na interakciju s Elasticsearchem. Ova komponenta je proizašla iz tvrtke Elastic N.V., koja razvija sve komponente ELK alata. Kibana omogućava korisniku da pretražuje i analizira podatke kroz grafičko sučelje. Od posebnog je značaja mogućnost izrade složenih nadzornih odnosno kontrolnih ploča (engl. *dashboard*) koje mogu pružiti uvid u kompleksne podatkovne strukture.

2.3.1 Mogućnosti Kibana alata

Kibana postaje dostupna 2013. godine kao korisničko sučelje za Elasticsearch. Od onda do danas Kibana je prerasla u složenu platformu za vizualizaciju svih vrsta podataka. Korisniku je na raspolaganju široka paleta prikaza, od jednostavnih linijskih i stupčastih grafikona do interaktivnih geografskih karti i modela za strojno učenje. Najveća vrijednost koju daje Kibana je brza vizualizacija velike količine podataka, što daje uvid u obrasce, trendove i anomalije.

Sve te mogućnosti prikaza se mogu kombinirati u nadzorne ploče koje se mogu prilagoditi ovisno u potrebama korisnika. Integracija s Elasticsearchem omogućava korisniku trenutačnu vizualizaciju te daljnje filtriranje, agregacije i dublje istraživanje prikupljenih podataka. Podržane su i napredne značajke kao što je strojno učenje koje se može koristiti za detektiranje anomalija i trendova u podacima. Od posebnog je značaja i podrška za geografske podatke koji omogućavaju korisnicima pregled podataka po regijama.

Iako nudi izuzetno napredne mogućnosti obrade podataka Kibana je i dovoljno jednostavna za upotrebu da neke mogućnosti mogu iskoristiti i ne-tehnički korisnici bez dubokog poznavanja podatkovne analize i programiranja.

2.3.2. Derivacije podataka u Kibani

Izrada vizualizacije ovisi o vrsti podataka u koje se želi steći uvid. Jedna od najzanimljivijih mogućnosti su razne agregacije koje omogućavaju grupiranje, zbrajanje i izračun raznih statistika. Podatke je moguće prikazati kroz obične tablice, linijske, stupčaste ili tortne grafikone, ali i toplinske karte. Sve grafikone je moguće detaljno uređivati s vlastitim oznakama, bojama, određivanjem vremenskog raspona itd. Svaki prikaz se može spremiti u biblioteku prikaza te se koristiti na više različitih mjesta.

Pojedini prikazi se mogu klonirati, prilagoditi i kombinirati s drugim prikazima kako bi se dobilo točno ono što korisnik traži.

Postoji i mehanizam za suradnju i dijeljenje ploča i pregleda među korisnicima ili čak ugrađivanje u druge aplikacije, također je moguće generirati izvještaje u PDF ili PNG formatu. Upravljanje različitim pregledima daje korisnicima i mogućnost verzioniranja te brzo vraćanje na stariju inačicu neke kontrolne ploče. Moguće je podesiti i ovlaštenja koja će različitim korisnicima dati različite mogućnosti s obzirom na dostupne ploče.

2.3.3 Upotreba Kibane za nadzor i dojavu

Osim mogućnosti vizualizacije podataka uz pomoć Kibane je moguće u stvarnom vremenu pratiti sistemske performanse i primati dojave o promjenama u sustavu, događajima vezanima uz sigurnost ili neke poslovne metrike. Upotreba Kibane je veoma raširena u računalnim sustavima u oblaku. Česta primjena je praćenje neuspjelih pokušaja prijave na sustav ili visok stupanj iskorištenosti procesorskih resursa koji se zatim mogu uspoređivati s drugim činjenicama. Zbog distribuirane prirode ELK alata često se koristi za centralizirani nadzor nad više regija ili usluga.

Sustav za dojavu unutar Kibane daje priliku korisnicima da definiraju uvjete u kojima će sustav reagirati na određeni način. To može biti slanje elektroničke poruke, zapisivanje događaja u poseban dnevnik ili izvršavanje HTTP poziva. Upravljanje dojavama nudi mogućnosti grupiranja dojava s obzirom na pravila, što pojednostavljuje rad s dojavama koje prate slične uvjete. Kibana se može i integrirati s vanjskim sustavima za upravljanje incidentima (npr. Opsgenie) kako bi točno određeni korisnici dobili informacije u stvarnom vremenu i na temelju njih reagirali. Uz slanje direktnih poruka česta je i upotreba generiranja duljih izvještaja koji se određenom periodikom pohranjuju ili šalju putem e-pošte.

2.4 Beats

Beats je zapravo skupina sličnih alata koji služe kao jednostavni pošiljatelji podataka (engl. *data shipper)* i koji čine najnoviji dodatak ELK alatima. Beats je kao i Kibana plod razvoja unutar tvrtke Elastic N.V. i služi za slanje podataka od izvora prema Elasticsearchu i Logstashu. Svaki „Beat“ je specijaliziran za prikupljanje određene vrste podataka, što ga čini veoma učinkovitim rješenjem za nadzor široke palete podataka u raznim okruženjima. Često se koriste na poslužiteljima, virtualnim strojevima, kontejnerima, čak i IoT uređajima zbog minimalnih sistemskih zahtjeva. Moguće ih je implementirati zasebno ili odvojeno, ovisno o potrebama.

2.4.1 Pošiljatelji podataka

Za razliku od Logstasha koji nudi širok broj mogućnosti i u pravilu zahtijeva puno hardverskih resursa za normalan rad, Beats ima jednu svrhu što osigurava minimalan utjecaj na sustav. Pojedini Beat prikuplja određenu vrstu podataka i prosljeđuje je Elasticsearchu ili Logstashu. Ova komponenta je razvijena zbog velike potrebe da se prikupljanje podataka proširi do samog ruba računalne mreže (engl. *network perimeter).* Ova specijalizirana funkcionalnost je od velike važnosti u okruženjima gdje su izvori podataka rasprostranjeni na više fizičkih lokacija, platformi ili čak okruženja u oblaku.

Beats radi neometano u kombinaciji s ostalim ELK alatima, u mnogim slučajevima čak u potpunosti zaobilazeći Logstash, ukoliko nema potrebe za složenim procesuiranjem podataka. Broj Beats alata se redovito povećava što omogućava prikupljanje podataka iz sve većeg broja izvora.

2.4.2 Filebeat i Metricbeat

Filebeat je osmišljen za prosljeđivanje zapisa iz raznih izvora i često se koristi za prikupljanje zapisa koje stvaraju mrežni poslužitelji kao što su Apache ili Nginx, aplikacijskih ili sistemskih dnevnika. Osobito je koristan za centralizirano praćenje dnevnika u slučajevima gdje je potrebno agregirati zapise s više poslužitelja na jednu lokaciju. Podržava očitavanje zapisa u raznim formatima, od otvorenog teksta do JSON datoteka. Podržano je i višelinijsko očitavanje (engl. *multiline support*), što znači da može obrađivati zapise koje se protežu u više redova teksta.

Metricbeat je predviđen za prikupljanje metrika iz operacijskog sustava i raznih servisa koji su pokrenuti na njemu. Najčešće se prikupljaju metrike o procesu, memoriji, radu diskovnih pogona i mrežnom prometu, a služe administratorima sustava za identifikaciju sporih dijelova sustava i optimizaciju resursa. Metricbeat radi u Linux, Windows i macOS okruženju, a metrike može skupljati od baza podataka (MySQL, PostgreSQL), mrežnih poslužitelja i platformi za kontejnerizaciju (Docker, Kubernetes). Posebno su korisne mogućnosti mapiranja očitanih metrika na određenu shemu i izmjena naziva polja, prije nego što ih Elasticsearch indeksira.

3. UPOTREBA I ZAHTJEVI

Instalacija ELK alata u poslužiteljskom okruženju zahtijeva pažljivo planiranje nabave hardvera kako bi se osigurao optimalan rad. Najveći zahtjevi stavljaju se redom pred procesor, radnu memoriju i diskove. Ovise o nekoliko faktora kao što su očekivana količina nadolazećih podataka, kompleksnost upita prema bazi i broj korisnika koji pristupaju sustavu. Za implementaciju na relativno malom sustavu može biti dovoljan i jedan poslužitelj, a zaista veliki sustavi mogu zahtijevati i veliku distribuiranu arhitekturu s više servera i čvorišta. U konačnici sve ovisi o konkretnom slučaju za koji se alati koriste. Najčešće se ovakav sustav može očekivati ondje gdje postoji potreba za centralizacijom dnevnika, nadzor u stvarnom vremenu, koji može raditi zajedno sa sustavima za praćenje aplikacija ili za praćenje sigurnosnih događaja. Moguće je pronaći ovakve sustave i u okolinama koje imaju potrebu za održavanjem dnevnika ili za planiranje kapaciteta i izradu poslovnih prognoza. S obzirom na relativno veliku kompleksnost konfiguracije i održavanja koje ELK alati postavljaju pred administratore takvi slučajevi su češći u veoma velikim organizacijama u strogo reguliranim djelatnostima.

3.1 Hardverski zahtjevi

Kao što je već spomenuto Elasticsearch je srce ELK alata i zahtijeva znatnu količinu procesorskih resursa, pogotovo u slučaju velikih setova podataka i kompleksnih upita. Preporuča se korištenje višejezgrenih procesora koji podržavaju višedretveni način rada. U slučaju da je potrebna velika protočnost podataka biti će potrebno koristiti više čvorova koji će biti balansirani između većeg broja procesora.

Performanse Elasticsearcha uvelike ovise i o dostupnoj memoriji. Prema uputama proizvođača softvera, potrebno je alocirati minimalno 4 GB radne memorije da se pokriju zahtjevi Elastisearcha i još 2 GB za Logstash i Kibanu. Najčešće se koristi oko 16 GB kako bi vrijeme odgovora na upit bilo prihvatljivo krajnjim korisnicima, pogotovo kada je potrebno upravljati većim indeksima.

Vrsta i količina prostora za pohranu podataka će odrediti veličinu historijskog pregleda koji će biti dostupan, ali i brzinu kojom su podaci dostupni. Potrebno je osim samih podataka pohraniti i indekse se koji se koriste za njihovo pretraživanje. Čvrsti diskovi (engl. *solid state drive*) u pravilu brže mogu očitavati i zapisivati podatke pa se češće i koriste. Potreban kapacitet ovisi o tome koliko podataka se očekuje unutar određene jedinice vremena i o vremenu retencije podataka. Ovisno o slučaju za koji se koriste ELK

alati raspon kapaciteta može biti od 100 GB do nekoliko terabajta. Također je preporučljivo složiti diskove u neku od RAID konfiguracija (engl. *redundant array of independent disks*) kako bi se osigurala zalihost i otpornost na greške.

Mrežni zahtjevi mogu biti veoma šaroliki ovisno o tome za što se koristi ELK alate. Pouzdana i brza mrežna veza je iznimno važna, osobito u distribuiranim sustavima u kojima čvorovi moraju često komunicirati. Proizvođač preporuča korištenje sučelja brzine 1 Gbps, a za veće količine podataka ti zahtjevi mogu biti i veći. Potrebno je voditi računa i o smanjivanju mrežne latencije kako bi podaci što brže kolali između čvorova.

3.2 Softverski zahtjevi

Instalacija je moguća na gotovo sve operacijske sustave, uključujući macOS i Windows. U najvećem broju slučajeva koriste se Linux sustavi za koje su ELK alati i predviđeni. Ubuntu, CentOS i Debian su distribucije koje se najčešće koriste za implementaciju ELK alata zbog stabilnosti i već raširene upotrebe raznim poslovnim okruženjima.

Elasticsearch i Logstash zahtijevaju i Java Runtime Environment (JRE). Relativno nedavno je proizvođač odlučio uz Elasticsearch pakirati i svoju verziju OpenJDK, tako da nije potrebno dodatno instalirati Javu. U slučaju zasebne instalacije Jave potrebno je koristiti verziju 11 ili noviju, a preporuča se korištenje OpenJDK ili Oracle JDK verzije.

Elasticsearch je do 2021. godine bio otvorenog koda nakon čega je proizvođač odlučio promijeniti model licenciranja. Trenutno su dostupne dvije licence, prva je besplatna, ali ima znatno manji broj značajki u odnosu na modele koji dolaze u obliku pretplate na uslugu. Sam Elasticsearch se može preuzeti sa službene stranice proizvođača i instalirati koristeći alate kao što su „apt“ za Debian i Ubuntu, ili „yum“ za CentOS.

Logstash i Kibana se preuzimaju i instaliraju na isti način. Potrebno je obratiti pozornost na verziju koja se instalira, jer sva 3 alata moraju imati istu verziju. Beats je neobavezna komponenta, ali zbog jednostavnosti i male instalacije sve češće korištena. Ovisno o vrsti zapisa koji se prikupljaju, odabire se verzija koja će najbolje poslužiti u određenom slučaju.

3.3 Ostali zahtjevi

Fino podešavanje operacijskog sustava na kojemu se postavljaju ELK alati može bitno utjecati na performanse. Osim podešavanja sustava na kojem su postavljeni ELK alati, treba obratiti pozornost i na sigurnost. Za pristup Kibani se u pravilu koristi HTTPS protokol, uz autentikaciju i autorizaciju za Elasticsearch i korištenje vatrozida kako bi se

ograničio pristup ELK komponentama. Moguće je koristiti i razne metode enkripcije, kao i kontrolu pristupa s obzirom na ulogu u sustavu. Unutar Linux sustava postoji nekoliko značajki koje je moguće podesiti kako bi se osigurale optimalne performanse.

3.3.1 Podešavanje operacijskog sustava

Prva mogućnost se odnosi na parametar unutar Linux jezgre (engl. *kernel)*, „vm.max_map_count“, koji definira najveći broj mapiranih adresnih područja koja proces može imati. Elasticsearch se uvelike oslanja na mapiranje memorije kako bi upravljao s velikim brojem indeksa. Svaki *shard* može stvoriti nekoliko memorijskih mapiranja, ako je ovaj parametar podešen na prenisku razinu, može doći do ozbiljnih problema u radu, čak i rušenja sustava, ako ne može kreirati nova mapiranja. Preporučena vrijednost je 262144 i moguće ju je promijeniti dodavanjem izraza: `vm.max_map_count=262144` u datoteku na lokaciji /etc/sysctl.conf.

Druga bitna stavka je ograničenje otvorenih datoteka. U Unix operacijskim sustavima svaki proces ima ograničen broj datoteka koje može otvoriti u jedinici vremena. Tim ograničenjem se može upravljati koristeći „ulimit“, što je posebno važno ukoliko postoji velik broj krhotina i indeksa. U slučaju premalog broja „ulimita“ može doći do smanjivanja performansi i grešaka u radu. Za promjenu ovog parametra potrebno je dodati izraze: `elasticsearch soft nofile 65535` i `elasticsearch hard nofile 65535`. Prvi izraz će postaviti gornju granicu do koje korisnik može mijenjati postavke, a drugi izraz označava najveću dopuštenu vrijednost u sustavu.

Virtualna memorija (engl. *swap space*) je dio prostora na disku koji se koristi kao virtualna radna memorija u slučaju da fizička memorija nije dostupna. Elasticsearch može optimalno raditi samo u radnoj memoriji, a u slučaju korištenja virtualne memorije može doći do znatnog usporavanja zbog puno sporijeg očitavanja i zapisivanja podataka na disk. To je moguće riješiti na dva načina; potpunim gašenjem virtualne memorije ili podešavanjem parametara koji upravljaju vjerojatnosti da će se koristiti virtualna memorija. Mogućnost ovako finog podešavanja sustava je jedan od razloga zašto je Linux najčešće korišten operacijski sustav za implementaciju ELK alata.

3.4 Logstash kanali

Logstash kanali (engl. *Logstash pipelines*) su konfiguracije koje definiraju kako se podaci kreću kroz sustav. Ulaz (engl. *input*), izlaz (engl. *output*) i filteri su tri glavne komponente koje usmjeravaju i obrađuju podatke. Ovaj dio teksta će se fokusirati na konfiguraciju kanala za upravljanje NGINX dnevnicima, uključujući definicije ulaza, filtera i izlaza za ovu vrstu dnevnika. Ulazni dodatak *file* (Kod 1) se koristi za čitanje dnevnika i omogućava Logstashu da ih bilježi odmah nakon što nastanu.

```
input {
  file {
    path => "/var/log/nginx/pristup.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
    type => "nginx-access"
  }

  file {
    path => "/var/log/nginx/greske.log"
    start_position => "beginning"
    sincedb_path => "/dev/null"
    type => "nginx-error"
  }
}
```

Kod 1. Primjer konfiguracije ulaza za NGINX dnevnike

Definirana su dva zasebna „file“ ulaza, jedan za pristupne zapise, drugi za zapise o pogreškama. U gornjem primjeru „path“ označava lokaciju NGINX zapisa, „start_position“ određuje da Logstash počinje čitati podatke od početka datoteke, a „sincedb_path“ koji je usmjeren u /dev/null određuje da Logstash ne prati lokaciju do koje je pročitao datoteku, što može biti korisno tijekom testiranja. Ovaj parametar se u produkcijskim okolinama najčešće ne mijenja.

3.4.1 Filteri

U slučaju NGNIX-a najčešće se koristi „grok“ filter koji parsira nestrukturirane zapise u polja. Često se dodaju vremenski i geografski podaci za obogaćivanje zapisa. U primjeru (Kod 2.); postavljena su tri filtera, čiji su nazivi otisnuti debljim slovima, koja parsiraju,

```
filter {
  grok {
    match => { "message" => "%{COMBINEDAPACHELOG}" }
  }

  date {
    match => [ "timestamp", "yyyy/MMM/dd:HH:mm:ss Z" ]
  }

  geoip {
    source => "clientip"
    target => "geoip"
    add_tag => [ "nginx-geoip" ]
  }
}
```

Kod 2. Primjer konfiguracije filtera za NGINX dnevnike

obrađuju i obogaćuju NGINX zapise. Prvi filter, „grok“, se primjenjuje na polje „message“, a koristi predefinirani obrazac koji izdvaja podatke kao što su IP adresa klijenta, vremenski zapisa, HTTP zahtjev itd. Drugi filter „date“ parsira vremenske zapise i pretvara ih u standardni format. Sljedeći dio:

match => ["timestamp", "yyyy/MMM/dd:HH:mm:ss Z"] naređuje Logstashu da polje „timestamp“ parsira prema navedenom formatu koji će biti pohranjen kao vremenska oznaka događaja uz koji je vezan. U konačnici će taj zapis izgledati kao ovaj primjer: 01/Jan/2024:12:34:56 +0000

Treći filter, „geoip“ treba zapisu dodati lokacijske podatke na temelju IP adrese, ali i oznaku da je taj proces obavljen nakon što su neobrađeni podaci izuzeti iz dnevnika, kako bi ih kasnije bilo lakše identificirati.

Izlaz definira gdje se podaci šalju. U slučaju NGINX-a je to u pravilu Elasticsearch gdje se podaci pripremaju za vizualizaciju u Kibani.

4. PRAKTIČNI RAD - ELK STACK

Ovo poglavlje prikazuje postupke instalacije i konfiguracije ELK alata i upotrebu prikupljenih podataka. Budući da su ELK alati optimizirani za rad na Linux operacijskim sustavima postoje dvije realne mogućnosti za njihovo testiranje. Prva je virtualizacija Linux sustava uz pomoć alata za virtualizaciju kao što su VMWare ili VirtualBox, a druga je korištenje *Windows Subsystem for Linux* (WSL) alata. Budući da virtualizacija sama po sebi zahtijeva određene sistemske resurse i da ELK alati imaju visoke hardverske zahtjeve odabrana je druga opcija. WSL omogućava pokretanje Linux distribucije izravno unutar Windows sustava bez podizanja virtualnog stroja ili zasebne instalacije Linuxa na isto računalo.

4.1 Instalacija i konfiguracija WSL-a

To je zapravo međuprogram koji daje sučelje za korištenje Linux jezgre koja je u potpunosti integrirana u Windowse, što daje korisniku mogućnost da instalira i pokrene Linux distribuciju kao na zasebnom računalu. Zbog visokog stupnja integracije moguće je neometano dijeljenje datoteka između Windows i Linux okruženja. Za razliku od tradicionalnih virtualnih strojeva, WSL ima minimalne sistemske zahtjeve što znači da je moguće pokretati Linux aplikacije paralelno s Windows aplikacijama bez značajnijeg usporavanja sustava. Kako bi instalirali ili ažurirali WSL, potrebno je kroz PowerShell pokrenuti naredbu: „wsl –install“. Zadana Linux distribucija na WSL-u je Ubuntu koja je i korištena kao podloga za ELK alate. Za sav daljnji rad s Linuxom korištena je Windows Terminal aplikacija koja pruža dodatne mogućnosti u odnosu na PowerShell ili Command Prompt. Gotovo sve naredbe u daljnjem tekstu potrebno je zadati s najvišim ovlaštenjima, što znači da se unutar svake podrazumijeva izraz „sudo“, ali nije napisan zbog veće preglednosti. Nakon instalacije Linuxa dobra je praksa ažurirati sustav što se radi naredbama: „apt-get update“ i „apt-get upgrade“. Također je potrebno imati instaliranu Javu bez koje Elasticsearch i Logstash ne mogu raditi. Instalacija Jave se radi naredbom: „`apt-get install openjdk-11-jdk -y`“. Budući da se ELK alati ne nalaze među alatima koji su automatski dostupni potrebno je:

1) instalirati osnovne pakete koji će omogućiti sigurno upravljanje instalacijom

```
apt-get install ca-certificates apt-transport-https curl gnupg -
y
```

2) dodati enkripcijski ključ kojim se provjerava autentičnost datoteka preuzetih iz repozitorija

```
curl -fsSL https://artifacts.elastic.co/GPG-KEY-elasticsearch |
gpg --dearmor -o /usr/dijeljenja/kljucevi/elastic-archive-
keyring.gpg
```

3) a zatim dodati i APT repozitorij iz kojega će se instalirati ELK alati

```
echo "deb [signed-by=/usr/dijeljenja/kljucevi/elastic-archive-
keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt
stable main" | tee /etc/apt/sources.list.d/elastic-8.x.list
```

4.2 Instalacija Elastic stack alata

Svaka komponenta ELK alata zahtijeva zasebnu instalaciju, jer u teoriji svaka komponenta može raditi neovisno o ostalima. Elasticsearch ne mora nužno preuzimati podatke od Logstasha, niti se oni moraju vizualizirati uz pomoć Kibane. S obzirom na Linux okruženje sam proces instalacije je nešto složeniji od onog na Windows ili macOS okruženju.

4.2.1 Elasticsearch - instalacija

Prije svake instalacije ažurira se popis dostupnog softvera u repozitorijima koji su navedeni u sustavu kao izvori: `apt-get update,` a sama instalacija se provodi jednostavnom naredbom: `apt-get install elasticsearch -y`

Za pokretanje, a zatim i postavljanje automatskog pokretanja Elasticsearcha zajedno sa sustavom koriste se naredbe: `systemctl enable elasticsearch` i `sudo systemctl start elasticsearch`

Nakon ovog koraka moguće je odmah i provjeriti uspješnost instalacije zadavanjem HTTP zahtjeva kroz preglednik ili terminal. Postavljanjem GET upita prema localhost:9200 (Slika 1.) dobiva se JSON odgovor s nazivom skupine i verzije:

Slika 1. Odgovor Elasticsearcha na GET zahtjev

4.2.2 Logstash – instalacija

Kao i za instalaciju Elasticsearcha koriste se iste naredbe, s tom razlikom da im se šalje argument „logstash“. Testiranje Logstasha može se napraviti stvaranjem konfiguracijske datoteke na lokaciji: /etc/logstash/conf.d/proba.conf sa sadržajem:

```
{ stdin { } } output { stdout { } }
```

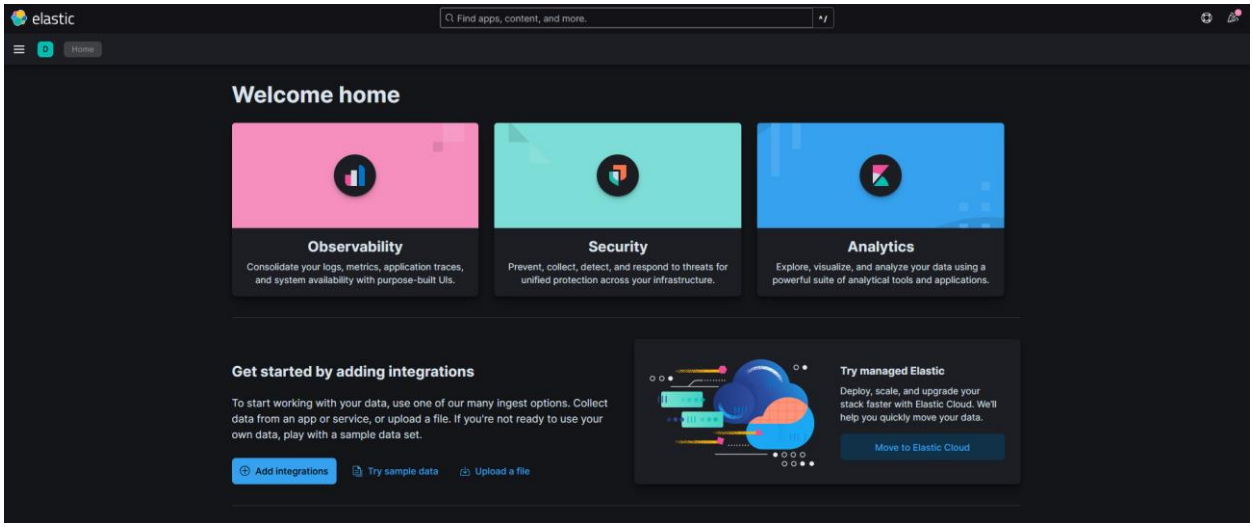
Ručnim pokretanjem Logstasha s niže navedenom naredbom mogu se unijeti probni podaci u terminal i provjeriti kako ih Logstash procesuira i ispisuje. (Slika 2.)

```
sudo /usr/share/logstash/bin/logstash -f
/etc/logstash/conf.d/proba.conf
```

Slika 2. Odgovor Logstasha na proizvoljni tekst

4.2.3 Kibana - instalacija

Nakon naredbi za instalaciju moguće je odmah i pristupiti Kibani kroz preglednik upitom prema localhost:5601, nakon čega se prikazuje osnovno Kibana sučelje: (Slika 3.)



Slika 3. Početni ekran u Kibani

4.2.4 Beats – instalacija

Kao i kod ostalih komponenti za instalaciju je potrebno najprije u datoteku s ključevima dodati javni kriptografski ključ, a zatim i repozitorij iz kojega će se preuzimati instalacija. Posljednji korak je konfiguracija automatskog pokretanja Beats servisa zajedno s operacijskim sustavom.

Za instalaciju i preuzimanje ključa koristi se „wget“ u kombinaciji s „gpg“ alatom koji pretvara ključ u binarni format.

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | gpg --dearmor | tee /usr/dijeljenja/kljucevi/elastic-archive-keyring.gpg > /dev/null
```

Repozitorij se dodaje na isti način kao i repozitorij za preuzimanje ELK alata.

```
echo "deb [signed-by=/usr/dijeljenja/kljucevi/elastic-archive-keyring.gpg] https://artifacts.elastic.co/packages/8.x/apt stable main" | tee /etc/apt/sources.list.d/elastic-8.x.list
```

Treba istaknuti da se za oba repozitorija koristi „echo“ metoda, jer se korištenjem tradicionalnog načina (add-apt-repository) dodaje zapis o nepostojećem repozitoriju u kojem se nalazi izvorni kod. Proizvođači Beatsa ne daju mogućnost takve instalacije.

4.3 Konfiguracije ELK alata

```
#!/bin/bash
# funkcija za restart
restart_service() {
    local service=$1
    echo "Restarting $service..."
    sudo systemctl restart $service
    if [ $? -eq 0 ]; then
        echo "$service restarted successfully."
    else
        echo "Failed to restart $service."
    fi
}

# odabir
echo "Make a selection:"
echo "1. Elasticsearch"
echo "2. Logstash"
echo "3. Kibana"
echo "4. All services"
echo "5. Exit"
read -p "Enter selection (1-5): " selection

case $selection in
    1)
        restart_service "elasticsearch"
        ;;
    2)
        restart_service "logstash"
        ;;
    3)
        restart_service "kibana"
        ;;
    4)
        restart_service "elasticsearch"
        restart_service "logstash"
        restart_service "kibana"
        ;;
    5)
        echo "Exiting..."
        exit 0
        ;;
    *)
        echo "Invalid selection. Exiting..."
        exit 1
        ;;
esac
```

Kod 3. Skripta za ponovo pokretanje ELK alata

Za instalaciju i testiranje bilo je potrebno relativno često ponovo pokretati servise pa je izrađena pomoćna skripta (Kod 3) koja znatno ubrzava taj proces.

Gotovo sve konfiguracijske datoteke su zapisane u YAML formatu. Radi se o formatu koji postoji od 2001. godine i dosad je doživio veoma široku primjenu. Slično kao programski jezik Python, koristi uvlačenje redova kako bi signalizirao ugnježđivanje blokova koda i veoma je osjetljiv u tom pogledu na pogreške.

Prema zadanim postavkama Elasticsearch je dostupan na *localhostu* na *portu* 9200, što je u YAML formatu zapisano na ovaj način: network.host: 0.0.0.0 i http.port: 9200. Ukoliko postoji potreba da Elasticsearch bude dostupan izvana potrebno je konfigurirati te 2 opcije. U slučaju korištenja više skupina i čvorova potrebno je definirati im nazive.

```
cluster.name: moj-cluster
node.name: cvor-1
discovery.seed_hosts: ["node-1", "node-2"]
```

Podaci koje šalju alati kao što je Metricbeat načelno dolaze u strukturiranom JSON formatu, što znači da nije potrebno puno toga parsirati kao u čistim dnevničkim zapisima, ali je zato moguće optimizirati podatke za indeksiranje.

```
input {
  beats {
    port => 5044
  }
}
```

Kod 4. Definicija Logstash ulaza za Metricbeat

Logstash se priprema za prihvat podataka iz na portu broj 5044. (Kod 4)

```
if [@metadata][beat] == "metricbeat" {
  mutate {
    add_tag => ["metricbeat"]
  }
}
```

Kod 5. Dodavanje oznaka na temelju uvjeta

Dodaje se oznaka „metricbeat“ svim događajima koje je poslao Metricbeat kako bi se podaci mogli kasnije odvajati prema izvoru. (Kod 5)

```
mutate {
  rename => { "[host][name]" => "hostname" }
}
```

Kod 7. Promjena naziva polja

Polje s nazivom [host][name] se preimenuje u „hostname“ što ga čini jednostavnijim za pretraživanje i vizualizaciju u Kibani. (Kod 7)

```
mutate {
  convert => {
    "[system][cpu][total][pct]" => "float"
  }
}
```

Kod 6. Promjena vrste podataka

Podataka u polju koje prikazuje iskorištenost procesorskih resursa u obliku postotka se mijenja u numerički oblik, kako bi se moglo lakše matematički obrađivati. (Kod 6)

```
mutate {
  add_field => { "environment" => "test" }
}
```

Kod 8. Dodavanje proizvoljnog polja

Dodavanje polja koja podacima dodaju dodatne karakteristike, u ovom slučaju naziv okoline mogu biti korisni za uspoređivanje ponašanja različitih okolina. (Kod 8)

```
mutate {
  remove_field => [ "metricset.module", "metricset.name" ]
}
```

Kod 9. Uklanjanje nepotrebnog polja

Uklanjanje polja (Kod 9) može smanjiti zahtjeve za pohranom podataka u Elasticsearchu.

```
output {
  elasticsearch {
    hosts => ["http://localhost:9200"]
    index => "metricbeat-%{+YYYY.MM.dd}"
    manage_template => false
  }

  stdout {
    codec => rubydebug
  }
}
```

Kod 10. Konfiguracija izlaza za Logstash

Definira se instanca Elasticsearcha kojoj se šalju podaci, zatim obrazac indeksiranja, a izlazni podaci se ispisuju na konzolu u čitljivom formatu za potrebe otkrivanja pogrešaka. (Kod 10)

Izrađene su dvije skripte koje služe za automatiziranje postupaka zaustavljanja (Kod 11) i provjeravanja statusa servisa (Kod 12).

```
#!/bin/bash
# stop
stop() {
    local service_name=$1
    echo "Stopping $service_name..."
    sudo systemctl stop $service_name

    # provjera
    local status=$(sudo systemctl is-active $service_name)
    if [ "$status" = "inactive" ]; then
        echo "$service_name stopped successfully"
    else
        echo "Failed to stop $service_name."
    fi
}
stop "elasticsearch"
stop "logstash"
stop "kibana"
echo "Shutting down..."
sudo shutdown -h now
```

Kod 11 Skripta za gašenje servisa

```
#!/bin/bash
# boje
GREEN='\033[0;32m'
RED='\033[0;31m'
NC='\033[0m'  # gašenje boje
# funkcija
check() {
    local service_name=$1
    local service_label=$2

    # status
    status_output=$(systemctl status $service_name --no-pager)

    # izoliranje teksta
    active_line=$(echo "$status_output" | grep -E "Active:")

    # određivanje boje
    if echo "$active_line" | grep -q "active (running)"; then
        echo -e "${GREEN}$active_line${NC}"
    else
        echo -e "${RED}$active_line${NC}"
    fi
    # print
    echo "$status_output" | grep -E "$service_label|Loaded|Main
PID|Tasks|Memory|CGroup|Docs" | grep -v "├─\|└─"
    echo # newline
}
# Check ELKs
echo "Checking Elasticsearch Service....."
check "elasticsearch" "elasticsearch.service - Elasticsearch"
echo "Checking Logstash Service....."
check "logstash" "logstash.service - Logstash"
echo "Checking Kibana Service....."
check "kibana" "kibana.service - Kibana"
```

Kod 12 Skripta za provjeru statusa servisa

4.3.1 Konfiguracija sigurnosnih postavki

Za prevenciju neovlaštenog pristupa Elasticsearchu potrebno je omogućiti sigurnosne značajke u YAML datoteci. Uključivanje sigurnosnih opcija radi se dodavanjem naredbi:

`xpack.security.enabled: true` i
`xpack.security.transport.ssl.enabled: true`,

a za autentikaciju korisnika koristi se naredba:

`sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwords interactive`, čijim se izvođenjem mogu postaviti lozinke za ugrađene korisničke uloge za Kibanu i Elasticsearch.

Za siguran pristup Kibani potrebno je omogućiti SSL/TLS uređivanjem datoteke kibana.yml, odnosno dodavanjem uputa:

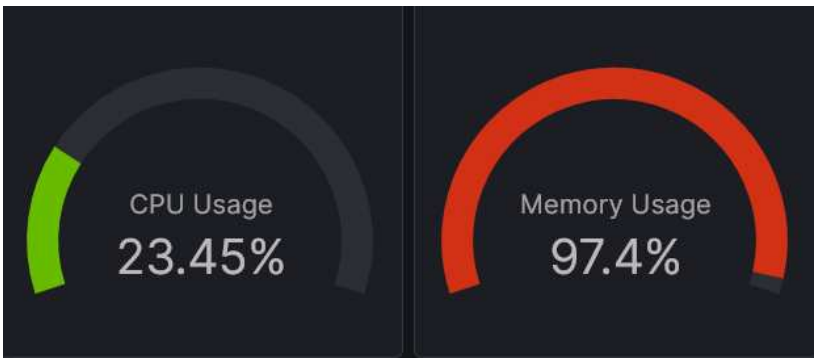
```
server.ssl.enabled: true
server.ssl.certificate: /etc/kibana/certs/kibana.crt
server.ssl.key: /etc/kibana/certs/kibana.key
```

Autentikacijom korisnika se može upravljati kroz Kibana korisničko sučelje ili kroz Elasticsearch API, što je osobito korisno u slučajevima s velikim brojem korisnika kojima je potrebna automatizacija u upravljanju s korisnicima i njihovim ulogama.

4.3.2 Vizualizacija podataka u Kibani

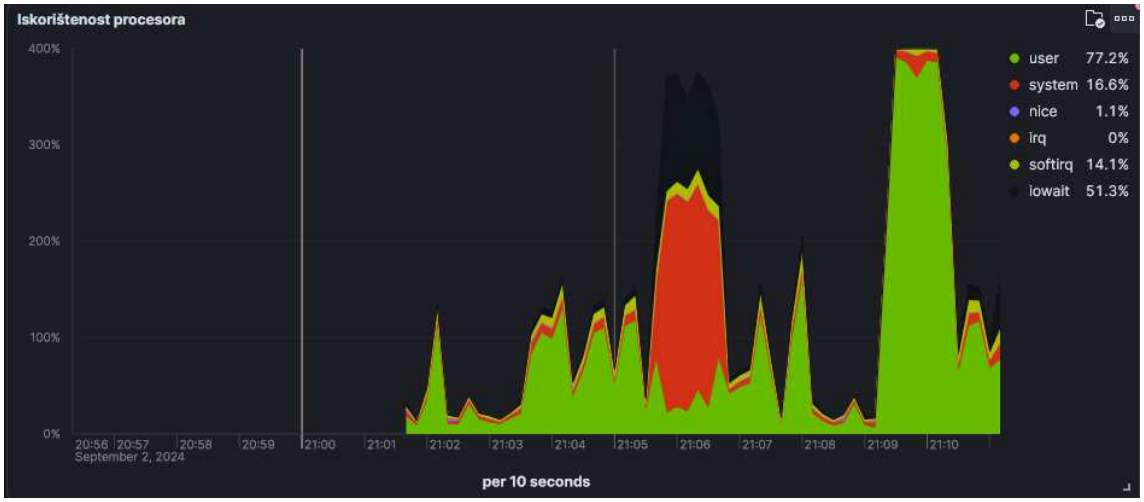
Prije vizualiziranja podataka potrebno je u Kibani definirati obrasce indeksiranja (engl. *index pattern*). U pravilu se obrasci definiraju po sistemu: **ime-servisa-logs-*,** gdje zvjezdica služi kao zamjenski znak (engl. *wildcard*) i umjesto nje ovdje može stajati bilo koji broj alfanumeričkih znakova. Ukoliko spremljeni podaci imaju vremensku dimenziju potrebno je definirati polje za nju. Nakon toga Kibana daje pregled svih polja koja su dostupna u indeksu. Potrebno ih je prepoznati i mapirati kako bi se podaci mogli koristiti.

Za sistemski nadzor najbolje je koristiti nekoliko kombinacija vizualizacija, ovisno o potrebnoj metrici. Za praćenje opterećenja procesora i memorije, najpraktičniji su pregledi u stilu manometra. (Slika 4)



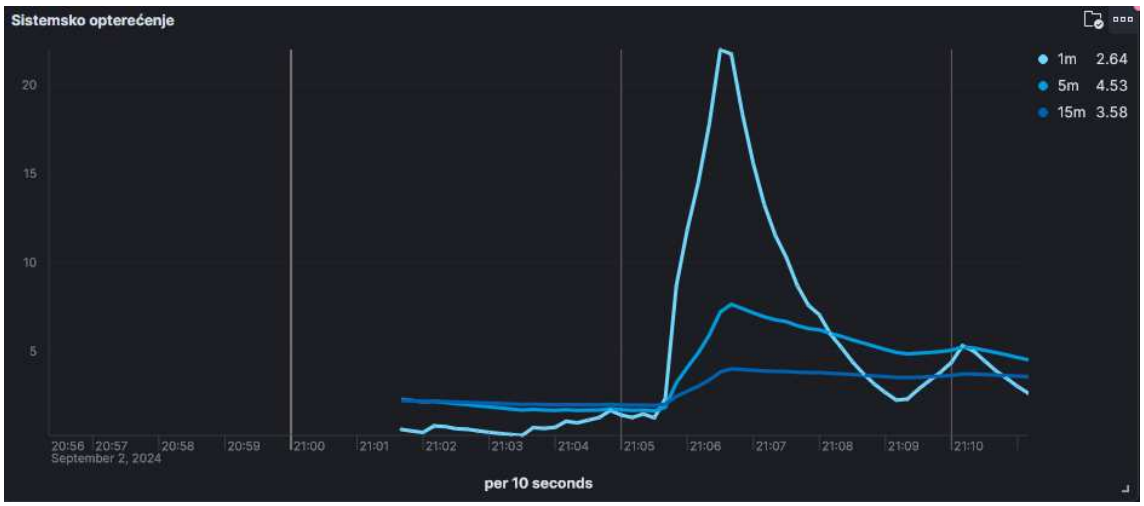
Slika 4. Opterećenje procesora i radne memorije

Za historijski pregled opterećenja procesora po korisniku upotrijebljen je višestruki linijski dijagram s ispunom prostora ispod krivulje. (Slika 5) Izračunom površina ispod krivulje dobiva se pregled opterećenosti procesora ukupno i zasebno po korisniku u stvarnom vremenu.



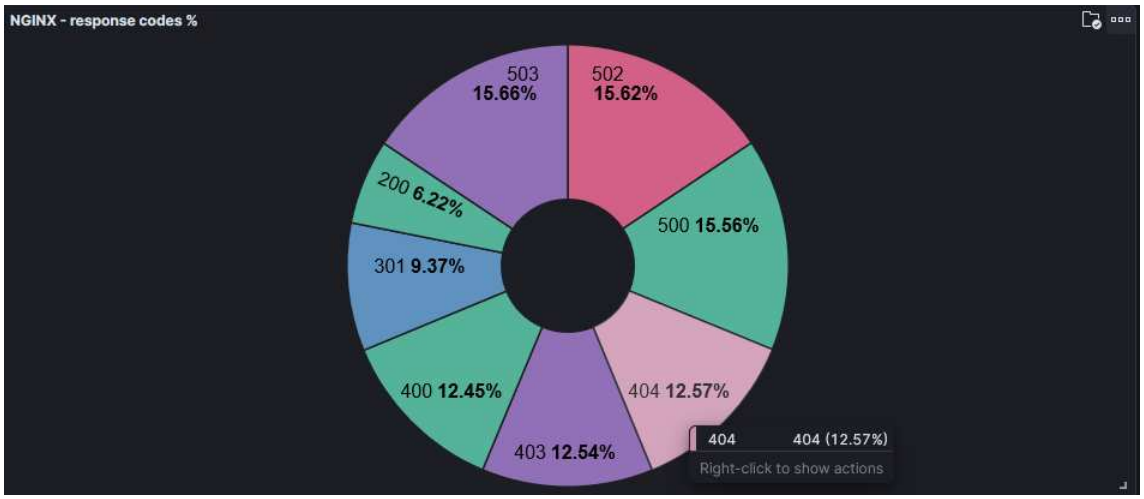
Slika 5. Histogram upotrebe procesora po korisniku

Iako takva informacija može biti varljiva, zanimljiv je i prikaz sistemskog opterećenja, koji prikazuje koliko procesa čeka neku od procesorskih jezgri da odgovori na zahtjev i vrijeme odaziva u milisekundama. (Slika 6) Ukoliko postoji velik broj procesora koji rade operacije tipa ulaz/izlaz, opterećenje sistema može biti veliko, iako je iskorištenost jezgri mala i vrijeme odaziva dobro.



Slika 6. Sistemsko opterećenje s vremenom odaziva

Za nedostupnost mrežnih stranica ponekad mogu postojati sistemski razlozi, zato primjerice nije loše bilježiti vrstu i količinu odgovora koje NGINX šalje na razne upite. (Slika 7) U kombinaciji s pregledom ponašanja sustava mogu se u kratkom vremenu izvući korisni zaključci.



Slika 7. Odgovori NGINX poslužitelja u postotcima

5. ZAKLJUČAK

Prikazana je praktična primjena ELK alata u WSL okruženju, što daje mogućnost korištenja Linux alata, uz neprimjetnu integraciju s Windows okruženjem. ELK alati već dugi niz godina dokazuju svoju učinkovitost za sistemski nadzor u raznim okruženjima. Za učinkovitost je zaslužno nekoliko faktora:

- zbog svoje distribuirane arhitekture Elasticsearch može skalirati horizontalno što ga čini pogodnim za nadzor velikih sustava. Logstash s druge strane daje fleksibilnost u korištenju podataka veoma različitih formata i izvora, a Kibana nudi mogućnosti vizualizacije koje odgovaraju i naprednim i jednostavnim zahtjevima.
- analitika u stvarnom vremenu koju pružaju Elasticsearch i Kibana su ključni faktori koji omogućavaju trenutačnu detekciju problema, pada u performansama i anomalija.
- vizualizacija i izvještavanje daju mogućnosti učinkovite interpretacije i prezentacije podataka bez koje je korisnik prisiljen snalaziti se u tabličnim prikazima podataka koji nisu uvijek pogodni za donošenje informiranih odluka.

Iako su ELK alati dostupni za instalaciju i u Windows okruženju, primarno su razvijeni za upotrebu na Linux platformama, što znači da se većina dokumentacije, podrške i najboljih praksi odnosi na Linux. S obzirom da WSL pruža Linux okruženje unutar Windows sustava bez ograničenja koja u pravilu donosi virtualizacija jasna je opravdanost ovog pristupa. Koristeći WSL za podizanje ELK alata daje bolju mogućnost kompatibilnosti, učinkovito korištenje hardverskih resursa, ali daje i konzistentnost s produkcijskim okolinama u kakvima se najčešće koriste ELK alati.

ELK alati nisu samo učinkovito rješenje za nadzor u stvarnom vremenu nego mogu poslužiti i kao dobar izvor za revizijski trag i dugoročnu pohranu dnevničkih zapisa. Revizijski tragovi su ključni za praćenje promjena u svakoj organizaciji. Potrebni su kao dokazi o događajima koji mogu biti od velike važnosti za kontinuitet poslovanja, pogotovo u pitanjima sigurnost i usklađenosti s regulatornim zahtjevima.

Kronološki prikaz sistemskih i korisničkih aktivnosti je ključna komponenta u revizijama i istragama vezanima za GDPR, razne ISO standarde (ISO 27001 ili ISO 9001), koji zahtijevaju praćenje pristupa osjetljivim podacima na određeni period. U slučaju sigurnosnih incidenata služe za identifikaciju kompromitiranih korisničkih računa i pomažu razumjeti vektore za napad.

Elasticsearch kao centralna komponenta je i dobro rješenje za dugoročnu pohranu velike količine dnevnika. Skalabilnost znatno olakšava posao organizacijama koje moraju čuvati dnevnike mjesecima ili godinama, a indeksiranje opet omogućava brzo pronalaženje i pristup relevantnim podacima. Elasticsearch ima i mogućnosti upravljanja životnim ciklusom indeksiranih podataka što daje mogućnost kreiranja politike za upravljanje podacima na način da se stariji podaci automatski premještaju na jeftinije medije za pohranu i nakon perioda retencije brišu ili uništavaju.

Implementacija kontrole pristupa s obzirom na ulogu u sustavu osigurava da je pristup dnevnicima ograničen samo na ovlašteno osoblje. Kibana i Elasticsearch podržavaju tu mogućnost što znači da organizacije mogu strogo kontrolirati dozvole za pregled i upravljanje dnevnicima.

Područja koja su osobito pogodna za daljnje istraživanje su integracija s novim tehnologijama; primjerice strojno učenje i platforme za kontejnerizaciju, poput Kubernetesa. Upotreba strojnog učenja za detekciju anomalija unutar Elasticsearcha može bitno pridonijeti prediktivnoj analizi sustava i dojavljivanju.

S obzirom na nedavne promjene u licenciranju ELK alata, tj. prelaska s licencu otvorenog koda na ograničenu licencu na neka pitanja će biti teže odgovoriti. Ipak postoji mogućnost istraživanja OpenSearch alata koji predstavljaju račvanje u razvoju ELK alata koje predvodi Amazon. Izgradnja snažne zajednice oko projekata otvorenog koda je često bila preduvjet uspjeha.

LITERATURA

1. Barnes, H. (2021) Pro windows subsystem for linux (WSL) powerful tools and practices for cross-platform development and collaboration. Berkeley, CA: Apress : Imprint: Apress.
2. Gormley, C. and Tong, Z. (2015) Elasticsearch the Definitive Guide: A distributed real-time search and analytics engine. Sebastopol, CA: O'Reilly.
3. HTML RJEČNIK, http://wiki.open.hr/l10n/rjecnik.html (Pristupljeno 31.8.2024.)
4. Konda, M. and Banon, S. (2023) Elasticsearch in action. Shelter Island, NY: Manning Publications.
5. Repositories for apt and yum Elastic. https://www.elastic.co/guide/en/beats/metricbeat/8.15/setup-repositories.html (Pristupljeno 31.8.2024.)
6. RJEČNICI Stranica GNU tima za hrvatske prijevode. https://www.gnu.org/server/standards/translations/hr/ (Pristupljeno 31.8.2024.)
7. Robbins, A. (2016) Bash pocket reference. Sebastopol, CA: O'Reilly Media.
8. Set up Elastic. https://www.elastic.co/guide/en/kibana/current/setup.html (Pristupljeno 31.8.2024.)
9. Set up Elasticsearch https://www.elastic.co/guide/en/elasticsearch/reference/current/setup.html (Pristupljeno 31.8.2024.)
10. Setting up and running Logstash. https://www.elastic.co/guide/en/logstash/current/setup-logstash.html (Pristupljeno 31.8.2024.)
11. Shah, K. (2024) Kibana 8.x - A quick start guide to data analysis: Learn about data exploration, visualization, and Dashboard Building with Kibana. Birmingham, UK: Packt Publishing Ltd.
12. Sharma, V. (2016) Beginning elastic stack. United States: Apress.
13. Shukla, P. and Sharath Kumar M. N. (2019) Learning elastic stack 7.0: Distributed search, analytics, and visualization using Elasticsearch, Logstash, beats, and Kibana. Birmingham, UK: Packt Publishing.

SAŽETAK

Ovaj završni rad istražuje upotrebu ELK alata za sistemski nadzor, dugoročnu pohranu podataka za reviziju i vizualizaciju. Kompleksnost modernih IT infrastruktura zahtijeva nadzor i učinkovito upravljanje sistemskim dnevnicima, što je s obzirom na količinu podataka koji se proizvode zadatak koji čak i u malim organizacijama nadilazi ljudske mogućnosti.

Na početku se daje relativno dubok uvid u način rada pojedinih komponenti. Opisan je način na koji Elasticsearch indeksira i pretražuje podatke, zatim kako ih Logstash prikuplja i transformira i koje su mogućnosti Kibane kao alata za vizualizaciju.

Prikazana je instalacija i priprema WSL okoline za instalaciju svih komponenti te instalacija samih komponenti. Prikazane su mogućnosti vizualizacije prikupljenih sistemskih dnevnika kao i obrada NGINX dnevnika te izvlačenje relevantnih informacija iz prikupljenih podataka.

Ključne riječi: ELK alati, Elasticsearch, Logstash, Kibana, Beats, indeksiranje, vizualizacija, obrada podataka

SUMMARY

This paper explores usage of ELK stack for system monitoring, long term data storage for auditing and visualisation. Complexity of modern IT infrastructures requires monitoring and efficient management of system logs, which considering the amount of created data is a task that even in small organisations exceeds human abilities.

In the beginning there is a deep dive in the way individual components work. There is an explanation of the way Elasticsearch indexes and searches for data, followed by a description of the way Logstash ingests and transforms data, as well as Kibana's possibilities as a tool for data visualisation.

Furthermore, the thesis explores the installation and preparation of WSL environment for installing individual components. There is a demonstration of options for visualising collected system logs as well as NGINX logs and extraction of relevant data from all collected datapoints.

Key words: ELK stack, Elasticsearch, Logstash, Kibana, Beats, indexing, visualisation, data processing