

# Integracija nadzornog sustava Zabbix s web aplikacijom na Unix serveru

---

**Stančer, Adrian**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Applied Sciences in Information Technology / Veleučilište suvremenih informacijskih tehnologija**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:289:852452>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-20**

*Repository / Repozitorij:*

[VSITE Repository - Repozitorij završnih i diplomskih radova VSITE-a](#)



**VELEUČILIŠTE SUVREMENIH INFORMACIJSKIH TEHNOLOGIJA**  
**STRUČNI PRIJEDIPLOMSKI STUDIJ INFORMACIJSKIH**  
**TEHNOLOGIJA**

*Adrian Stančer*

**ZAVRŠNI RAD**

**INTEGRACIJA NADZORNOG SUSTAVA ZABBIX S WEB**  
**APLIKACIJOM NA UNIX SERVERU**

**Zagreb, studenoga 2024.**



Veleučilište suvremenih informacijskih tehnologija  
10000 Zagreb, Ulica Vjekoslava Klaića 7

Studij: Stručni prijediplomski studij informacijskih tehnologija  
smjer programiranje  
Student: **Adrian Stančer**  
Matični broj: 2022007

## Zadatak završnog rada

Predmet: Programski alati na UNIX računalima  
Naslov: **Integracija nadzornog sustava Zabbix s web aplikacijom na Unix serveru**  
Zadatak: Objasniti nadzorni sustav Zabbix koji omogućava cjeloviti nadzor IT sustava te prikazati njegovu integraciju s web aplikacijom. Zabbix sustav integriran je s Laravel web aplikacijom baziranoj na docker tehnologiji.  
Mentor: Ivan Capan, pred.  
Zadatak uručen kandidatu: 29.2.2024.  
Rok za predaju rada: 15.11.2024.  
Rad predan: \_\_\_\_\_

### Povjerenstvo:

dr. sc. Aleksandar Skendžić, v. pred.	član predsjednik	_____
Ivan Capan, pred.	mentor	_____
Jurica Đurić, v. pred.	član	_____

# SADRŽAJ

1. UVOD .....	7
2. NADZORNI SUSTAV ZABBIX .....	9
2.1. Ključne značajke .....	9
2.2. Arhitektura.....	10
2.3. Instalacija i konfiguracija Zabbix poslužitelja .....	12
2.4. Instalacija i konfiguracija Zabbix agenta .....	13
2.5. Korisničko mrežno sučelje .....	13
2.6. Kako nadzirati resurs (engl. host)?.....	14
3. LARADOCK I NJEGOVA INTEGRACIJA SA ZABBIXOM .....	16
3.1. Docker i kontejnerizacija.....	16
3.2. Laravel mrežno okruženje .....	17
3.3. Zabbix i Laradock .....	18
3.4. Zabbix i Docker.....	19
3.5. Zabbix i Laravel .....	21
4. PRAKTIČNI RAD - INTEGRACIJA ZABBIX SUSTAVA SA LARADOCK MREŽNIM POSLUŽITELJOM .....	24
4.1. Instalacija i konfiguracija Laradocka .....	24
4.2. Postavljanje Zabbix kontejnera .....	26
4.3. Konfiguracija Zabbix agenata .....	29
4.4. Postavljanje nadzornih objekata.....	31
4.5. Postavljanje pravila upozorenja.....	32
5. ZAKLJUČAK .....	33
LITERATURA .....	35
SAŽETAK .....	36
SUMMARY .....	37

## POPIS SLIKA

Slika 1. Arhitektura programske podrške Zabbix .....	10
Slika 2. Povezanost sastavnica Zabbix sustava .....	11
Slika 3. Zabbix korisničko mrežno sučelje .....	14
Slika 4. Početna stranica Zabbix mrežnog sjedišta .....	15
Slika 5. Arhitektura Dockera .....	17
Slika 6. Zabbix i Docker – kontejnerizacija .....	20
Slika 7. Laravel aplikacija (osobna izrada, <a href="http://localhost">http://localhost</a> ).....	26
Slika 8. Pokrenuti Zabbix kontejneri (osobna izrada, Ubuntu shell).....	28

## **POPIS TABLICA**

Tablica 1. Razlozi za kombinaciju Zabbixa i Laradocka.....	19
Tablica 2. Ključne značajke nadzora Laravel aplikacija s pomoću Zabbixa.....	22

## POPIS KODOVA

Kôd 1. Konfiguracija .env datoteke .....	26
Kôd 2. Konfiguracija docker-compose.yml datoteke .....	27
Kôd 3. Dodavanje zabbix agenata u docker-compose.yml datoteku .....	29

## 1. UVOD

Nadzor rada sustava IT (Informacijskih Tehnologija) obuhvaća skup svih postupaka i alata koji omogućavaju stalno praćenje i nadziranje njegovog ispravnog rada. To znači otkrivanje problema na svim sastavnicama IT sustava, kao što su servisi poslužitelja, sklopovlje poslužitelja, mrežni uređaji svih vrsta i dr.

Nadzorni sustav Zabbix je besplatni programski alat za praćenje različitih IT sastavnica, uključujući mreže i poslužitelje. To je aplikacija otvorenog koda (engl. open source) usmjerena i stvorena za praćenje milijuna mjernih podataka u stvarnom vremenu, preuzetih s tisuća poslužitelja, virtualnih ili mrežnih uređaja unutar poslovnog sustava, institucije, organizacije. Tako predstavlja odgovarajuće rješenje za male i velike poslovne korisnike.

Laradock je implementacija Docker kontejnera koji omogućavaju razvoj aplikacija u Laravel razvojnom okruženju (engl. framework). Docker je tehnologija za stvaranje spremnika ili kontejnera (engl. container), tj. paketa pojedinačnih aplikacija koji kombiniraju različite IT sastavnice i izoliraju ih od ostatka sustava.

Cilj ovog rada je objasniti nadzorni sustav Zabbix, koji omogućava cjeloviti nadzor IT sustava, te prikazati njegovu primjenu na mrežnoj aplikaciji. Zabbix sustav primijenjen je na Laravel mrežnoj aplikaciji baziranoj na Docker tehnologiji.

Tema završnog rada obrađena je na dva načina:

1. teoretski dio rada: izvršeno je prikupljanje podataka, strukturiranje, obrađena su i prezentirana saznanja iz literature vezana uz nadzorni sustav Zabbix
2. praktični dio rada: primijenjeno je znanje stečeno analizom stručne literature u traženju načina za rješavanje praktičnih problema nadzora Zabbix sustava koji je integriran s Laravel mrežnom aplikacijom; navedeno je ostvareno s pomoću analize primjera iz prakse, studijom slučaja, te samostalnom obradom zadataka.

Primarni izvori podataka i informacija korišteni pri izradi ovog rada su: knjige, stručni članci, literatura za seminare, PP (engl. Power Point) prezentacije, internetske stranice i ostalo. Korišteni su i sekundarni izvori podataka: riječnik hrvatskog jezika, enciklopedija hrvatskog



jezika i ostale referentne publikacije. Metode koje su korištene pri izradi rada su sljedeće: metoda konkretizacije, klasifikacije, analize, sinteze, indukcije, dedukcije i metoda deskripcije.

Prema strukturi, ovaj završni rad sastoji se od pet cjelina, popisa slika, tablica, kodova, literature, te na kraju sažetka i popisa ključnih riječi. Prva cjelina je *Uvod* u kojem se čitatelja uvodi u temu i cilj rada, te strukturu i tehniku izrade istog. Druga cjelina nosi naslov *Nadzorni sustav Zabbix*, sastoji se od šest potpoglavlja u kojima su opisane ključne značajke, arhitektura, instalacija i konfiguracija Zabbix sustava, mrežno sučelje i host. Treća cjelina nosi naslov *Laradock i njegova integracija sa Zabbixom*, sastoji se od pet potpoglavlja u kojima je opisan Docker i kontejnerizacija, Laravel mrežno okruženje, Zabbix i Laradock, te je dat prikaz implementacije Zabbix kontejnera u Dockeru i korištenje Zabbixa za nadzor Laravel mrežne aplikacije. Četvrta cjelina je praktični rad pod naslovom *Integracija Zabbix sustava s Laradock mrežnim poslužiteljom*. Sastoji se od pet potpoglavlja u kojima su razrađeni koraci i kodovi potrebni za implementaciju Laradock mrežne aplikacije i Zabbix nadzora nad istom. U petoj cjelini iznesen je *Zaključak*. Rad završava popisom literature, sažetkom i popisom ključnih riječi.

## 2. NADZORNI SUSTAV ZABBIX

Osnivač, vlasnik i kreator Zabbixa je Alexei Vladishev. Izradio ga je 1998. godine. Razvio ga je kao programsko rješenje otvorenog koda koji se može dijeliti sa svima. Prva verzija Zabbixa izdana je 2001. godine, a Zabbix kao tvrtka osnovana je 2005. godine s ciljem pružanja stručnih usluga tehničke podrške. Prema definiciji, Zabbix je distribuirani sustav otvorenog koda koji se koristi za nadzor računalnih resursa. Upotrebljava se za praćenje stabilnosti, učinkovitosti, stanja rada poslužitelja, mreže, servisa, aplikacija i dr. Pruža mogućnost nadzora opterećenja sklopovlja - procesora, memorije, diska. Također, nadzire dostupnost servisa, sadržaj logova, sigurnost sustava, procese sustava. Nadalje, Zabbix je i sustav za nadzor mreže, dizajniran za nadgledanje i bilježenje stanja različitih mrežnih usluga, poslužitelja i mrežnog sklopovlja. (Zabbix, The story of how Zabbix software became one of the worlds most demanded IT monitoring solutions, April 17, 2020, <https://www.zabbix.com/la/pr/pr280>) (pristupljeno 21.5.2024.)

Zabbix koristi prilagodljivi sustav obavijesti koji korisnicima omogućava konfiguriranje određenih upozorenja koja se dogode u sustavu, a mogu se slati na e-poštu. S ovom opcijom, Zabbix omogućava informiranje o svim promjenama koje se dogode na poslužitelju. Također, omogućava razne funkcije izvještavanja i vizualizacije podataka na temelju pohranjenih stavki. Zbog navedenog, koristi se i za planiranje povećanja kapaciteta, primjerice: nakon opetovanih upozorenja da je disk gotovo pun, može se planirati proširenje prostora za pohranu podataka. (Zabbix, forum, <https://www.zabbix.com/forum/>) (pristupljeno 25.5.2024.)

Svim izvješćima i statistikama koje nudi Zabbix, kao i konfiguracijskim parametrima, može se pristupiti putem mrežnog sučelja. To jamči da se stanje mreže i stanje poslužitelja mogu procijeniti s bilo kojeg mjesta, a da se pritom ne mora biti fizički na tom mjestu.

Njegov internetski prednji kraj (engl. frontend) napisan je na PHP-u, stražnji kraj (engl. backend) napisan je na jeziku C i koristi MySQL, PostgreSQL, SQLite, Oracle ili IBM DB2 za spremanje podataka koji se analiziraju.

### 2.1. Ključne značajke

Ključne značajke Zabbix-a su sljedeće:

- mogućnost prilagodbe: proširivao se kroz API-je (engl. Application Programming Interface) i prilagođene skripte
- centralizirani nadzor putem mrežnog administratora (frontend)
- SLA (engl. Service Level Agreement) omogućuje analizu kvalitete usluga koje pruža

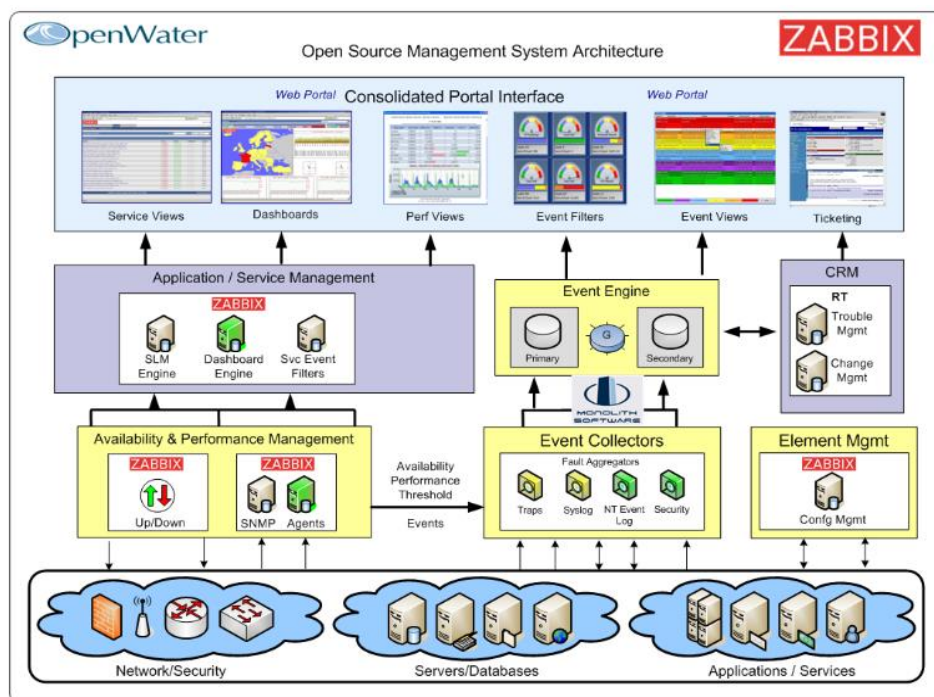
informatička organizacija u određenom poslovnom sustavu i dr.

- jednostavno mrežno sučelje za održavanje postavki koje određuju sastavnice nadziranog IT sustava, te pregled svih događaja koje je zabilježio sustav za nadzor
- svi događaji pohranjuju se u bazu podataka
- podržava veliki broj platformi: Windows, Linux, MacOS
- omogućuje razvoj dodatka (engl. plugin) od neovisnih autora
- skalabilnost: pogodan za mala i velika okruženja i dr.

Zabbix nudi praćenje za lokalne mreže (engl. LAN) i širokopojasne mreže (engl. WAN). Aplikacija se instalira na poslužitelj (Linux) i tada je usmjerena prikupljanju podataka. Na svojem mrežnom sučelju grafički prikazuje sve informacije koje prikuplja. Zabbix sadržava agente za Linux, Mac i Windows koji su instalirani na poslužiteljima ili klijentima koje prema izboru korisnika nadgledaju. Dodatno, ovo rješenje korisnicima omogućava i prikaz stanja pisača, usmjerivača, preklopnika, senzora temperature i vlage. (Administracija Portala Sustav, Instalirajte Zabbix na RedHat Centos 7 sustave <https://hr.admininfo.info/instalar-zabbix-en-sistemas-redhat-centos-7>, svibanj 2024.) (pristupljeno 20.7.2024.)

## 2.2. Arhitektura

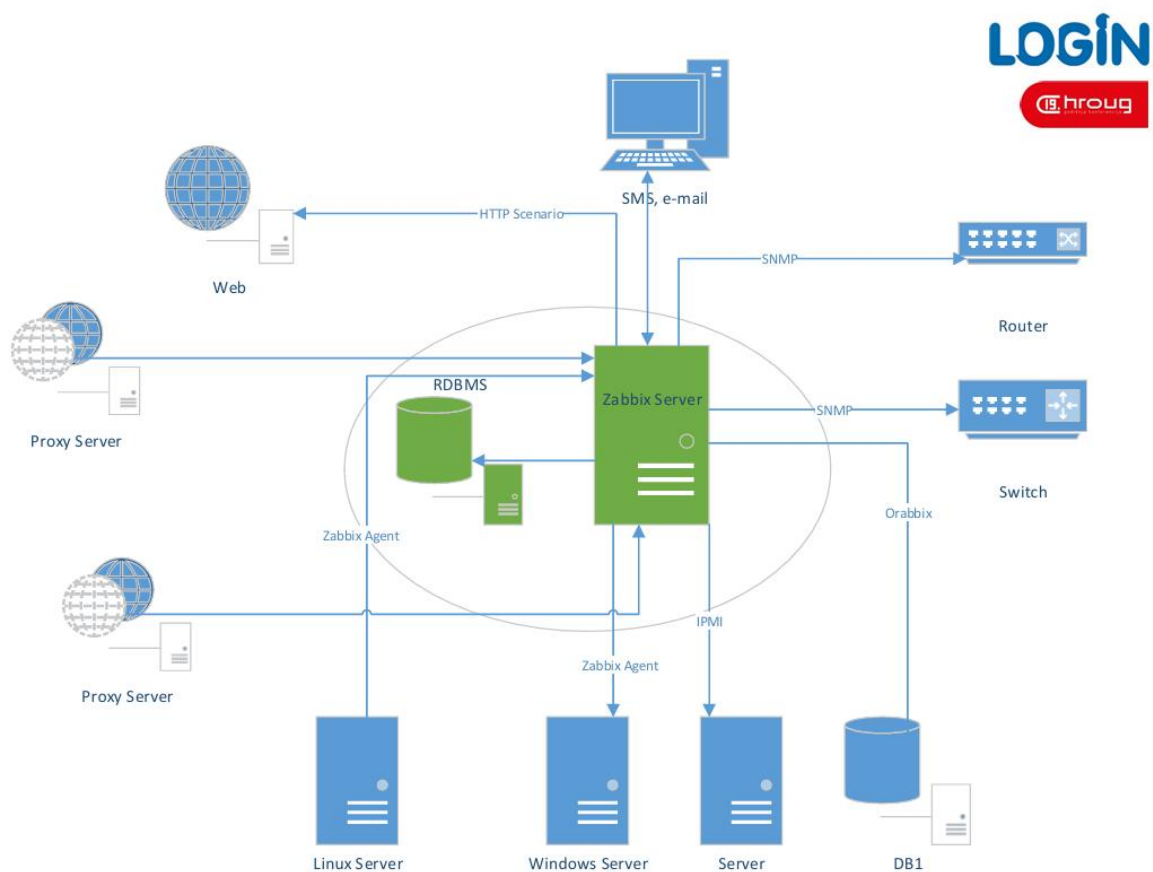
Zabbix se sastoji od nekoliko programskih sastavnica. Dvije osnovne sastavnice su: Zabbix poslužitelj i Zabbix agent. Slijedi prikaz arhitekture nadzornog sustava Zabbix (Slika 1).



Slika 1. Arhitektura programske podrške Zabbix  
(CARNet, SNMP protokol NCERT-PUBDOC-2010-09-313, Revizija, str. 26/30)

Zabbix poslužitelj sastoji se od tri dijela: Zabbix poslužitelj, Zabbix korisničko sučelje i baza podataka. Zabbix poslužitelj je centralna sastavnica, koja prikuplja podatke od agenata. Baza podataka je mjesto gdje poslužitelj pohranjuje sve podatke koje je prikupio od agenata. Zabbix Proxy je međuposlužitelj (neobavezan) koji prikuplja podatke u ime Zabbix poslužitelja, te na taj način smanjuje opterećenje na Zabbix poslužitelju (Drageljević, M. (2019), *Zabbix - ili si OK ili si problem*, Rovinj: Login Software, 19. HROUG konferencija). Zabbix agent instalira se na uređaj koji se nadzire i šalje podatke Zabbix poslužitelju. Zabbix agenti izvršavaju pasivne i aktivne provjere sustava. U pasivnoj provjeri agent odgovara na zahtjev za podacima. Zabbix poslužitelj (ili Proxy) traži podatke, npr. CPU (engl. Central Processing Unit) opterećenje, a Zabbix agent šalje natrag rezultat.

Aktivne provjere zahtijevaju složeniju obradu. Agent prvo dohvaća popis stavki sa Zabbix poslužitelja, a zatim povremeno šalje nove vrijednosti poslužitelju. Hoće li se izvršiti pasivne ili aktivne provjere postavlja se odabirom odgovarajuće vrste nadzora. Zadaci i povezanost glavnih sastavnica programske podrške Zabbix sustava prikazani su u nastavku (Slika 2).



Slika 2. Povezanost sastavnica Zabbix sustava (Drageljević, M. (2019), *Zabbix – ili si OK ili si problem*, Rovinj: Login Software, 19. HROUG konferencija)

Zabbix sustav radi i funkcionira povezujući Zabbix poslužitelja, Zabbix korisničko sučelje i Zabbix bazu podataka. Kao što je prethodno navedeno, centralna sastavnica Zabbix sustava je Zabbix poslužitelj, prema kojemu Zabbix Agenti (moguće i Zabbix Proxy) šalju podatke o stanju i integritetu sustava. Na osnovi definiranih pravila i prikupljenih podataka, Zabbix poslužitelj šalje upozorenje koje se prosljeđuje administratorima na odgovarajući način (e-mail, sms, Slack poruka i dr.). Jednostavan pristup Zabbixu s bilo kojeg mjesta i s bilo koje platforme osigurava korisničko sučelje. Nadalje, sve informacije o postavkama za Zabbix pohranjene su u bazi podataka s kojom su u interakciji i poslužitelj i korisničko sučelje. Zabbix Proxy (kao neobavezan međuposlužitelj) je sastavnica koja nadzire i prikuplja podatke s jednog ili više uređaja, svi podatci su spremljeni lokalno, te se šalju Zabbix poslužitelju kojemu Proxy pripada. Zabbix agenti instaliraju se na IT sustav kojem se želi nadzirati resurse i aplikacije, prikupljaju podatke te ih prosljeđuju poslužitelju ili Proxyju, na daljnju obradu. Zabbix agenti podržani su na istim operacijskim sustavima kao Zabbix poslužitelj i Zabbix Proxy.

Zabbix agent 2 je nova generacija Zabbix agenta, a razvijen je kako bi se: smanjio broj mrežnih veza, poboljšala istovremenost provjera (osigurati jednostavne provjere koje se sastoje od nekoliko redaka koda i omogućiti složene provjere koje se sastoje od dugotrajnih skripti i samostalnog prikupljanja podataka s povremenim slanjem) i omogućila zamjena Zabbix agenta (jer Zabbix agent 2 podržava sve prethodne funkcionalnosti).

(zabbix.com, <https://www.zabbix.com/documentation/current/en/manual/concepts/server>)

(pristupljeno 5.8.2024.)

### **2.3. Instalacija i konfiguracija Zabbix poslužitelja**

Zabbix se može instalirati na većinu UNIX sustava kao što su Linux, FreeBSD, MacOS. Instalacija i konfiguracija Zabbix poslužitelja ključna je faza u postavljanju sustava za nadzor. Ovaj proces uključuje nekoliko koraka koji omogućavaju korisnicima da brzo i jednostavno pokrenu Zabbix u svom okruženju. Potrebno je preuzeti službene Zabbix Docker slike koje sadrže sve sastavnice za rad Zabbix poslužitelja. Zabbix obično koristi bazu podataka kao što je MySQL ili PostgreSQL, stoga je važno osigurati da je i ova sastavnica dostupna. Preporučuje se korištenje Docker alata za upravljanje kontejnerima, što omogućava lako pokretanje i održavanje aplikacija u izoliranim okruženjima. Nakon preuzimanja potrebnih slika, sljedeći korak je definiranje konfiguracijskih postavki za Zabbix poslužitelj i bazu podataka. Ovo uključuje određivanje korisničkih imena, lozinki i imena baze podataka. Ove informacije bit će potrebne tijekom inicijalne konfiguracije Zabbix sučelja. Nakon što su kontejneri pokrenuti,

korisnik može pristupiti Zabbix korisničkom sučelju putem preglednika. Na početnoj stranici će se prikazati vodič za inicijalnu konfiguraciju, koji uključuje provjeru veze s bazom podataka i unos administratorskog računa. Ovaj proces pruža upute za ispravno postavljanje Zabbix sustava.

(Zabbix SIA (2023), *Zabbix Documentation*, <https://www.zabbix.com/documentation/current/manual/installation/install>) (pristupljeno 9.8.2024.)

Nakon što su svi koraci uspješno završeni, Zabbix poslužitelj će biti spreman za korištenje. Korisnici mogu dodavati hostove, definirati nadzorne objekte i postaviti pravila upozorenja kako bi optimizirali rad svojih aplikacija i infrastrukture. Instalacija Zabbix poslužitelja na ovaj način pruža prilagodljivost i skalabilnost, omogućujući korisnicima da lako upravljaju svojim sustavima za nadzor.

#### **2.4. Instalacija i konfiguracija Zabbix agenta**

Instalacija i konfiguracija Zabbix agenta sljedeća je važna faza u postavljanju sustava za nadzor, jer agent omogućuje Zabbix poslužitelju prikupljanje informacija o resursima i stanju sustava na odabranim hostovima. Zabbix agent može se instalirati na različitim operativnim sustavima, uključujući Linux, Windows i MacOS, čime se osigurava prilagodljivost u nadzoru različitih okruženja. Korisnici trebaju preuzeti odgovarajuću verziju Zabbix agenta koja odgovara njihovom operativnom sustavu. Nakon preuzimanja, agent se može instalirati s pomoću paketnog menadžera ili kao binarna datoteka, ovisno o platformi. Tijekom instalacije važno je osigurati da agent ima pristup mreži kako bi mogao komunicirati sa Zabbix poslužiteljom. Nakon što je agent instaliran, potrebno je izvršiti konfiguraciju putem datoteke `zabbix_agentd.conf`. Ova konfiguracija uključuje postavljanje adrese Zabbix poslužitelja, definiciju hosta i druge opcije koje određuju koje će informacije agent prikupljati. Nakon što su sve potrebne postavke izvršene, agent se može pokrenuti, a Zabbix poslužitelj će početi primati podatke o radnim karakteristikama i resursima s nadziranih hostova.

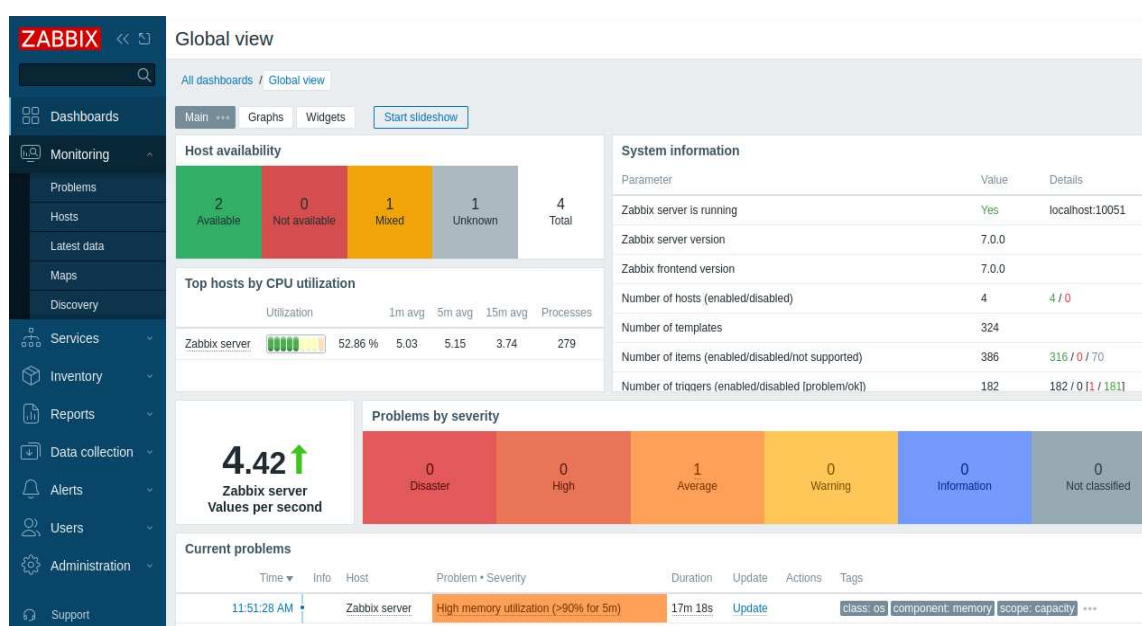
(Zabbix SIA (2023), *Zabbix Agent Configuration*, [https://www.zabbix.com/documentation/current/manual/appendix/install/agent\\_configuration](https://www.zabbix.com/documentation/current/manual/appendix/install/agent_configuration)) (pristupljeno 9.8.2024.)

#### **2.5. Korisničko mrežno sučelje**

Korisničko mrežno sučelje Zabbixa predstavlja ključno sredstvo za upravljanje i interakciju sa

sustavom za nadzor. Korisnicima omogućuje pristup svim mogućnostima Zabbixa putem preglednika, čime se olakšava konfiguracija, praćenje i analiza podataka o performansama sustava. Sučelje je dizajnirano da bude intuitivno i prilagodljivo, što omogućuje korisnicima svih razina stručnosti da se lako snalaze kroz različite opcije. Korisnici se mogu prijaviti u Zabbix mrežno sučelje koristeći svoje administratorske ili korisničke račune. Nakon prijave, dostupne su im različite funkcionalnosti, uključujući dodavanje i upravljanje hostovima, definiranje nadzornih objekata, postavljanje pravila upozorenja i pregled statističkih podataka. Sučelje također omogućuje izradu prilagođenih grafikona i izvještaja, što korisnicima pomaže u vizualizaciji podataka i praćenju radnih karakteristika u stvarnom vremenu (Slika 3). Jedna od značajnih prednosti mrežnog sučelja Zabbixa je njegova mogućnost prilagodbe. Korisnici mogu mijenjati izgled i funkcionalnost sučelja prema svojim potrebama, čime se povećava efikasnost rada. Također, Zabbix podržava različite preglednike, što ga čini dostupnim na raznim platformama.

(Zabbix SIA (2023), *Getting Started with Zabbix Web Interface*, preuzeto s [https://www.zabbix.com/documentation/current/manual/web\\_interface/getting\\_started](https://www.zabbix.com/documentation/current/manual/web_interface/getting_started)) (pristupljeno 10.8.2024.)

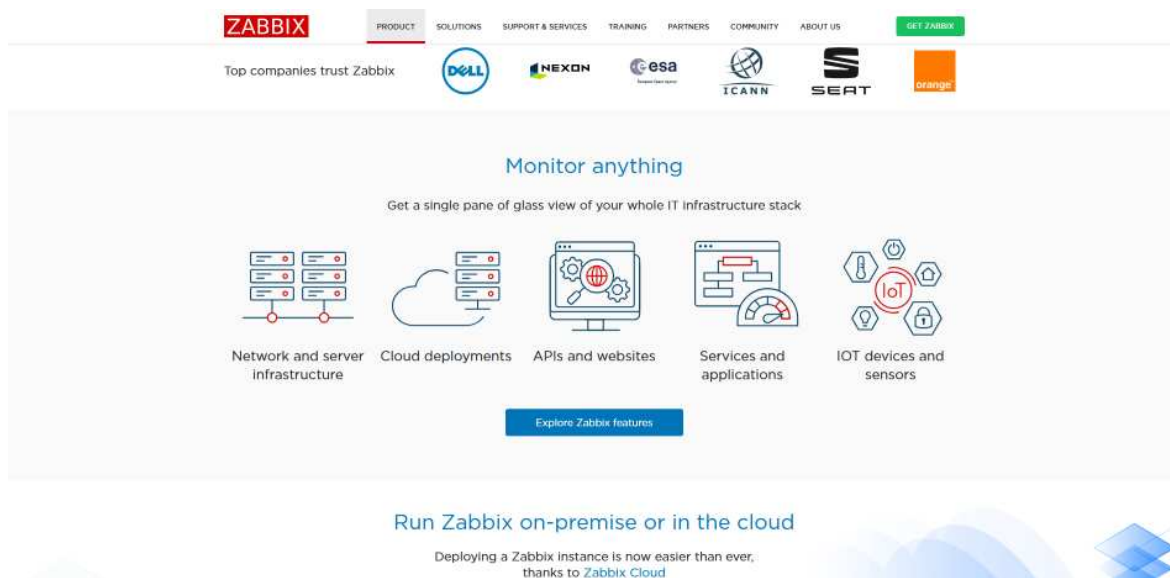


Slika 3. Zabbix korisničko mrežno sučelje (zabbix.com, documentation, <https://www.zabbix.com/documentation>)

## 2.6. Kako nadzirati resurs (engl. host)?

Nadzor resursa (engl. host) u Zabbixu predstavlja ključnu funkcionalnost koja omogućuje praćenje radnih karakteristika, dostupnosti i stanja raznih sastavnica unutar infrastrukture. Ovaj

proces uključuje nekoliko koraka kako bi postavke za sve resurse koje je potrebno nadzirati bile ispravne. Prvi korak u nadzoru resursa je dodavanje novog resursa u Zabbix sustav putem mrežnog sučelja. Tijekom ovog postupka, korisnik unosi osnovne informacije o resursu, uključujući ime, IP (engl. Internet Protocol) adresu ili DNS (engl. Domain Name System) naziv, te definira grupu resursa kojoj pripada. Nakon što je resurs dodan, potrebno je odabrati odgovarajući Zabbix agent koji će prikupljati podatke s tog resursa. Nakon dodavanja resursa, korisnik može definirati koji će se karakteristike resursa pratiti. Zabbix omogućuje praćenje različitih karakteristika, uključujući CPU opterećenje, iskorištenost memorije, dostupnost mrežnih usluga i još mnogo toga. Ove metrike se definiraju kroz nadzorne objekte (items) koji se dodaju resursu. Svaki nadzorni objekt može imati specifične postavke, poput intervala prikupljanja podataka i uvjeta za aktiviranje upozorenja. Konačno, korisnik može postaviti pravila upozorenja koja će se aktivirati kada nadzirane karakteristike pređu definirane pragove. Upozorenja omogućuju pravovremeno obavještanje administratora o potencijalnim problemima, čime se osigurava brza reakcija i održavanje optimalne razine usluga (Slika 4). (Zabbix SIA (2023), *Monitoring Hosts in Zabbix*, preuzeto s <https://www.zabbix.com/documentation/current/manual/config/host/overview>) (pristupljeno 10.8.2024.)



Slika 4. Početna stranica Zabbix mrežnog sjedišta  
(Zabbix SIA (2023), *Zabbix.com*, <https://www.zabbix.com>)



### 3. LARADOCK I NJEGOVA INTEGRACIJA SA ZABBIXOM

Laradock je razvojno okruženje otvorenog koda temeljeno na Dockeru, osmišljeno kako bi olakšalo razvoj PHP aplikacija, posebno onih temeljenih na Laravelu. Pruža Docker kontejnere s unaprijed određenim postavkama koji omogućuju brzu i jednostavnu instalaciju svih potrebnih alata i servisa kao što su Nginx, MySQL, Redis i drugi, bez potrebe za ručnim postavljanjem svakog od njih. Laradock značajno pojednostavljuje proces postavljanja okruženja za razvoj aplikacija, smanjujući vrijeme potrebno za određivanje postavki te omogućava jednostavnu prilagodbu okruženja prema specifičnim potrebama korisnika. Svojom prilagodljivošću omogućuje rad na različitim projektima koristeći iste Docker postavke, što ga čini jednim od popularnih alata u zajednici za brzi razvoj aplikacija. (Laradock (2023) *Laradock Documentation*, preuzeto s <https://laradock.io>) (pristupljeno 12.8.2024.)

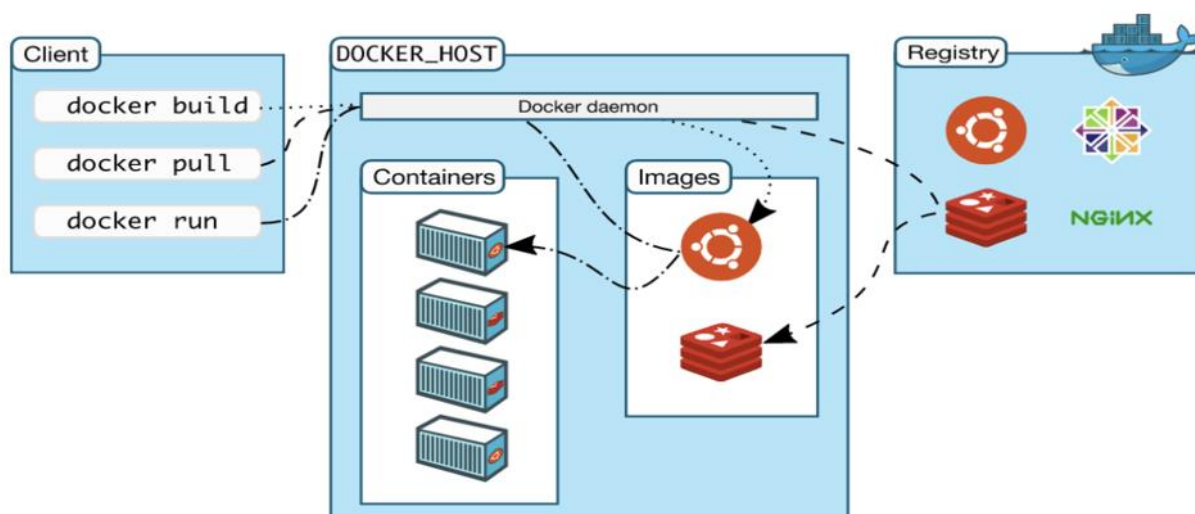
#### 3.1. Docker i kontejnerizacija

Kontejnerizacija je tehnologija koja omogućuje pokretanje aplikacija i njihovih ovisnosti unutar izoliranih okruženja poznatih kao spremnici ili kontejneri (engl. container). Kontejneri su lagani, prenosivi i ne zahtijevaju pokretanje cijelog operativnog sustava kao virtualnog stroja, već dijele jezgru domaćeg operacijskog sustava, čime se smanjuje potreban prostor i resursi. Ovaj način izolacije osigurava da aplikacija ima dosljedno ponašanje neovisno o okruženju u kojem se pokreće.

Jedna od glavnih prednosti kontejnerizacije je njena efikasnost. Budući da kontejneri koriste manje resursa nego tradicionalni virtualni strojevi, omogućuju brže pokretanje i veći broj instanci aplikacija na istom sklopovlju. Osim toga, kontejneri pružaju dosljedno okruženje za razvoj, testiranje i produkciju, što smanjuje mogućnost "radi na mom računalu, ali ne radi na poslužitelju" problema. Kontejneri također omogućuju lako skaliranje aplikacija te povećavaju sigurnost izolacijom aplikacija. (Merkel, D. (2014), *Docker: lightweight Linux containers for consistent development and deployment*, Linux Journal, 2014 (239), 2.)

Docker je najpoznatija i najraširenija platforma za kontejnerizaciju. S pomoću Dockera, razvojni timovi mogu kreirati, distribuirati i upravljati kontejnerima na jednostavan i učinkovit način. Docker koristi datoteku nazvanu "Dockerfile" koja opisuje potrebne ovisnosti i korake za izgradnju aplikacije unutar kontejnera. Docker Hub, službeni repozitorij za Docker slike, dodatno olakšava preuzimanje i dijeljenje unaprijed definiranih okruženja. Implementacija s pomoću Dockera ubrzava razvojne procese, olakšava prijelaz između različitih okruženja

(lokalno, staging, produkcija) te osigurava visoku razinu prenosivosti aplikacija (Slika 5). (Docker Inc. (2023), *Docker Documentation*, preuzeto s <https://docs.docker.com>) (pristupljeno 14.8.2024.)



Slika 5. Arhitektura Dockera  
(Docker Inc. (2023), *Docker Documentation*, <https://docs.docker.com>)

### 3.2. Laravel mrežno okruženje

Laravel je PHP razvojno okruženje otvorenog koda za mrežni razvoj, poznat po svojoj elegantnoj sintaksi i jednostavnosti upotrebe. Kreiran je kako bi olakšao razvoj robusnih mrežnih aplikacija i sustava, pružajući razne ugrađene funkcionalnosti poput upravljanja rutama, autentifikacije, sesija i cache memorije. Laravel se ističe svojim modularnim dizajnom i velikom prilagodljivošću, što ga čini idealnim izborom za razvoj širokog raspona aplikacija, od malih mrežnih stranica do kompleksnih poslovnih rješenja.

Jedna od najvećih prednosti Laravel okruženja u odnosu na konkurenciju je njegova intuitivnost i bogata dokumentacija. Laravel dolazi s ugrađenim alatima koji ubrzavaju razvoj, kao što su Eloquent ORM za rad s bazama podataka, Blade template engine za prikazivanje sadržaja te Artisan, alat za automatizaciju uobičajenih zadataka. Osim toga, njegova podrška za RESTful API-je i napredna sredstva sigurnosti čini ga atraktivnim za moderne mrežne aplikacije. U odnosu na druga PHP razvojna okruženja, Laravel nudi jednostavnije rješenje za složene zadatke te bolju organizaciju koda kroz princip MVC (engl. Model-View-Controller) arhitekture.

Kada je riječ o implementaciji Laravel okruženja s pomoću Dockera i kontejnerizacije, Docker omogućuje brzo postavljanje i upravljanje svim potrebnim sastavnicama za Laravel aplikacije.

Korištenjem Docker kontejnera, Laravel mrežne aplikacije mogu se razvijati, testirati i pokretati unutar konzistentnog i izoliranog okruženja. (Docker Inc. (2023), *Docker Documentation*, preuzeto s <https://docs.docker.com>) (pristupljeno 14.8.2024.)

S pomoću Laradocka, specijaliziranog za Laravel, moguće je postaviti sve potrebne servise, kao što su mrežni poslužitelj (npr. Nginx), baze podataka (MySQL, PostgreSQL), Redis i druge, unutar Docker kontejnera bez potrebe za ručnim postavljanjem. Ova implementacija omogućuje lakše upravljanje ovisnostima, brže pokretanje razvojnih okruženja te osigurava da aplikacija uvijek radi na isti način, bez obzira na lokalne konfiguracije računala na kojem se razvija ili produkcijski poslužitelj.

Korištenjem Docker i Laradock rješenja, Laravel aplikacije postaju prenosive, skalabilne i jednostavne za implementaciju na bilo kojoj infrastrukturi, čime se značajno smanjuje vrijeme potrebno za postavljanje okruženja te rizik od konfiguracijskih pogrešaka. (Otwell, T. (2022), *Laravel Documentation*, Laravel)

### **3.3. Zabbix i Laradock**

Kombinacija Zabbixa i Laradocka donosi značajne prednosti za razvojna i operativna okruženja koja zahtijevaju nadzor i kontrolu nad aplikacijama i resursima. Zabbix, kao moćan alat za nadzor sustava i mreža, omogućava detaljno praćenje performansi i ispravnosti svih sastavnica sustava, dok Laradock osigurava brzu implementaciju razvojnih okruženja putem Docker kontejnerizacije. Kombiniranjem ova dva alata, moguće je u potpunosti automatizirati nadzor nad Docker kontejnerima i servisima koji se koriste unutar Laradock okruženja.

Razlozi za integraciju Zabbixa s Laradockom uključuju potrebu za učinkovitim nadzorom kontejneriziranih aplikacija, skalabilnost sustava i osiguranje visokog stupnja dostupnosti. Zabbix omogućuje praćenje performansi servisa kao što su mrežni poslužitelji, baze podataka i aplikacije unutar kontejnera, te pravovremeno detektiranje problema, poput preopterećenja CPU-a, memorije ili diskovnog prostora (Tablica 1). Uz to, Zabbix može slati obavijesti o problemima u stvarnom vremenu, omogućavajući brzu reakciju i smanjenje zastoja. (Zabbix SIA (2023), *Zabbix Documentation*, preuzeto s [zabbix.com](https://www.zabbix.com)) (pristupljeno 17.8.2024.)

Osim navedenog, Laradock i Docker omogućuju jednostavno postavljanje Zabbixa unutar kontejnera, što ubrzava implementaciju nadzornog sustava bez potrebe za ručnim konfiguracijama na različitim poslužiteljima. Ova integracija povećava prilagodljivost i prenosivost, jer se sustav za nadzor može lako premjestiti između različitih razvojnih i produkcijskih okruženja. Tako, kombinacija Zabbixa i Laradocka osigurava optimiziranu,

dosljednu i stabilnu infrastrukturu, što je ključno za održavanje visokih performansi i pouzdanosti aplikacija. (Laradock (2023), *Laradock Documentation*, preuzeto s <https://laradock.io>) (pristupljeno 17.8.2024.)

Tablica 1. Razlozi za kombinaciju Zabbixa i Laradocka

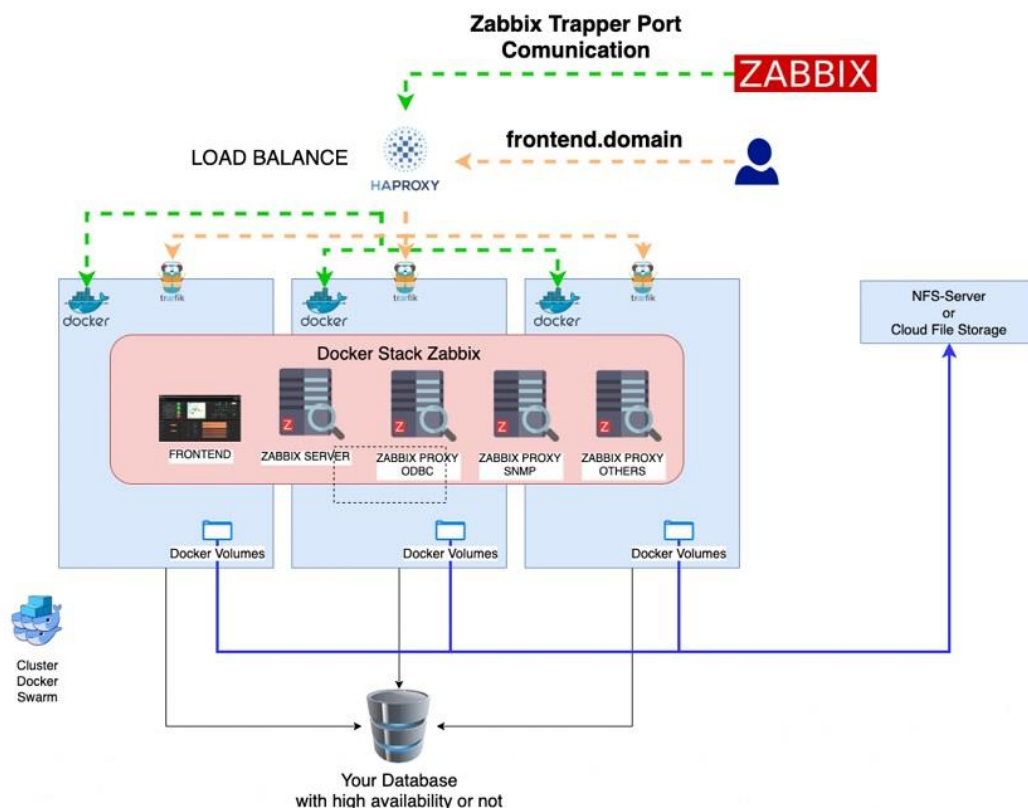
Razlog	Opis
automatizacija nadzora	automatsko otkrivanje i nadzor kontejnera
centralizirani nadzor	nadzor svih servisa unutar Laradocka
performanse i optimizacija	praćenje resursa za poboljšanje performansi
skalabilnost	jednostavno skaliranje nadzora u Docker okruženju

(Laradock (2023), *Laradock Documentation*, preuzeto s <https://laradock.io>)

### 3.4. Zabbix i Docker

Kombinacija Zabbixa i Dockera unutar Laradock okruženja predstavlja kvalitetno rješenje za nadzor i upravljanje aplikacijama i infrastrukturom u kontejneriziranim okruženjima. Docker, kao platforma za kontejnerizaciju, omogućuje jednostavno pokretanje i upravljanje aplikacijama unutar izoliranih okruženja, što osigurava njihovu prenosivost i konzistentnost. Međutim, zbog dinamičke prirode Docker okruženja, praćenje performansi i ispravnosti kontejnera postaje složenije. Ovdje dolazi Zabbix, koji omogućuje sveobuhvatan nadzor nad svim sastavnicama unutar Docker sustava. (Zabbix SIA (2023), *Zabbix Documentation*, preuzeto s <https://www.zabbix.com>) (pristupljeno 17.8.2024.)

Zabbix, integriran s Dockerom, omogućava praćenje ključnih metrika kontejnera kao što su potrošnja CPU-a, memorije, mrežnog prometa i diskovnog prostora, te prikupljanje informacija o statusu pojedinih servisa unutar kontejnera (Slika 6). Time se osigurava potpuna vidljivost i kontrola nad cijelim sustavom. Posebno je važno pravovremeno otkrivanje problema kao što su preopterećenja resursa, neispravni procesi ili zastoji u mreži, što može ozbiljno utjecati na dostupnost i performanse aplikacija. Integracija Zabbixa omogućuje automatsko generiranje upozorenja i obavijesti, što pomaže administratorima u brzom rješavanju problema i smanjuje vrijeme zastoja.



Slika 6. Zabbix i Docker – kontejnerizacija  
(Zabbix SIA (2023), Zabbix.com, <https://www.zabbix.com>)

Razlozi za kombiniranje Zabbixa s Dockerom u sklopu Laradocka uključuju:

- Automatizaciju nadzora: Docker okruženja često uključuju veliki broj kontejnera koji dinamički dolaze i odlaze. Zabbix omogućuje automatsko otkrivanje i nadzor novih kontejnera, što smanjuje potrebu za ručnim postavljanjem nadzora.
- Centralizirani nadzor: Zabbix omogućuje nadzor ne samo pojedinačnih kontejnera, već i svih servisa unutar Laradock okruženja (mrežni poslužitelji, baze podataka, cache sustavi, itd.). Ovo osigurava da svi kritični dijelovi aplikacije budu pod stalnim nadzorom.
- Performanse i optimizaciju: Zabbix prikuplja detaljne podatke o potrošnji resursa unutar kontejnera, što omogućuje optimizaciju performansi aplikacija i infrastrukture. Na primjer, uvid u potrošnju memorije i CPU-a može pomoći u pravilnom rasporedu resursa između različitih kontejnera.
- Skalabilnost: Zabbix se jednostavno skalira u Docker okruženju, omogućujući nadzor velikog broja kontejnera i servisa bez gubitka performansi. Ova prilagodljivost je posebno korisna za aplikacije koje rastu i zahtijevaju dodatne resurse.

- Prenosivost i jednostavnost: Docker omogućuje jednostavno postavljanje Zabbix poslužitelja unutar kontejnera, što znači da cijelo nadzorno okruženje može biti prenosivo i lako implementirano na različitim poslužiteljima. To olakšava upravljanje okruženjima u različitim fazama razvoja, od lokalnog razvoja do produkcije.

Korištenjem Dockera za implementaciju Zabbixa, organizacije dobivaju visoko učinkovito rješenje za nadzor koje je prilagodljivo, skalabilno i lako za implementaciju, čime se osigurava stabilnost i visok stupanj dostupnosti aplikacija unutar kontejneriziranih okruženja. (Docker Inc. (2023), *Docker Documentation*, preuzeto s <https://docs.docker.com>) (pristupljeno 19.8.2024.)

### 3.5. Zabbix i Laravel

Kombinacija Zabbixa i Laravela unutar Laradock okruženja pruža sveobuhvatno rješenje za nadzor performansi i ispravnosti aplikacija razvijenih s pomoću Laravel razvojnog okruženja. Laravel, kao jedan od najpopularnijih PHP razvojnih okruženja za razvoj mrežnih aplikacija, omogućuje brzu i efikasnu izradu složenih aplikacija, no kako bi se osigurala njihova pouzdanost i visoka dostupnost u produkciji, potreban je stalan nadzor nad resursima i funkcionalnostima aplikacija. Zabbix se ovdje pojavljuje kao idealan alat za osiguranje nadzora nad ključnim aspektima Laravel aplikacija, kao što su potrošnja resursa, performanse baze podataka, status poslužitelja i opterećenje.

Razlozi za kombiniranje Zabbixa s Laravelom unutar Laradock okruženja su (Tablica 2):

- Praćenje performansi aplikacija: Laravel aplikacije često se oslanjaju na različite sastavnice, uključujući mrežne poslužitelje, baze podataka i vanjske API-je. Zabbix omogućuje praćenje svakog od tih aspekata u stvarnom vremenu, omogućujući uvid u to kako aplikacija koristi resurse. Na primjer, može pratiti vrijeme učitavanja stranica, broj aktivnih korisnika, opterećenje poslužitelja, te stanje HTTP (engl.. Hypertext Transfer Protocol) zahtjeva, što pomaže u prepoznavanju uskih grla u aplikaciji i optimizaciji performansi.
- Nadzor baza podataka: mrežne aplikacije najčešće koriste relacijske baze podataka, u ovom slučaju MariaDB – izvedba MySQL baze. Zabbix može kontinuirano pratiti metrike baza podataka, uključujući broj upita, brzinu izvršenja upita i korištenje indeksa. Ovaj nadzor omogućuje pravovremeno otkrivanje problema s bazom podataka,

kao što su spori upiti ili visoko opterećenje baze, čime se smanjuje mogućnost pada performansi aplikacije.

- Nadzor ovisnosti i resursa: Laravel aplikacije često koriste dodatne servise kao što su Redis za caching ili mail servisi za slanje e-pošte. Zabbix omogućuje nadzor nad dostupnošću i performansama tih servisa unutar Docker kontejnera, što osigurava da svi dijelovi aplikacije rade ispravno i bez zastoja. Time se omogućuje brza reakcija na potencijalne probleme i izbjegava se negativan utjecaj na korisničko iskustvo.
- Sigurnosni nadzor: Laravel nudi različite sigurnosne mehanizme, a praćenje potencijalnih sigurnosnih prijetnji može dodatno ojačati sigurnost aplikacije. Zabbix može pratiti neovlaštene pokušaje pristupa, neobične obrasce ponašanja korisnika ili neočekivane promjene u konfiguraciji, pružajući dodatni sloj zaštite Laravel aplikaciji. Time se smanjuje rizik od sigurnosnih incidenata i omogućuje pravovremena intervencija.
- Skalabilnost i prilagodljivost: Laravel aplikacije često rastu kako se povećava broj korisnika, što zahtijeva skalabilnu infrastrukturu. Korištenjem Zabbixa unutar Laradock okruženja, moguće je nadzirati skaliranje aplikacije i osigurati da svi resursi budu optimalno iskorišteni. Zabbix omogućuje praćenje vodoravnog i okomitog skaliranja resursa unutar Docker kontejnera, čime se osigurava da aplikacija ostane stabilna i visoko dostupna, čak i pod visokim opterećenjem.
- Automatizacija nadzora i obavijesti: Jedna od ključnih prednosti integracije Zabbixa s Laravelom je mogućnost postavljanja automatskih upozorenja i obavijesti u slučaju problema. Primjerice, dođe li do preopterećenja poslužitelja ili ako neki od ključnih servisa prestane raditi, Zabbix će odmah poslati obavijest administratorima putem e-pošte ili drugog kanala, omogućujući brzu reakciju i rješavanje problema prije nego što ozbiljno utječe na korisnike. (Otwell, T. (2022), *Laravel Documentation*, Laravel)

Tablica 2. Ključne značajke nadzora Laravel aplikacija s pomoću Zabbixa

Značajka	Opis
praćenje performansi	mjerenje vremena učitavanja, opterećenja poslužitelja i brzine upita
nadzor baza podataka	praćenje metrika poput brzine izvršenja i broja upita
nadzor ovisnosti	kontrola dostupnosti vanjskih servisa poput Redisa ili API-ja
sigurnosni nadzor	praćenje sigurnosnih prijetnji i neovlaštenih pristupa

(Otwell, T. (2022), *Laravel Documentation*, Laravel)

Kombinacija Zabbixa i Laravela unutar Laradock okruženja omogućuje sveobuhvatan nadzor i optimizaciju Laravel aplikacija, osiguravajući stabilnost, sigurnost i visoke performanse. Uz mogućnost detaljnog praćenja svih aspekata aplikacije i njezinih resursa, Zabbix pomaže osigurati visoku dostupnost i pouzdanost aplikacija, smanjujući vrijeme zastoja i poboljšavajući korisničko iskustvo. (Laradock (2023), *Laradock Documentation*, preuzeto s <https://laradock.io>) (pristupljeno 20.8.2024.)



## 4. PRAKTIČNI RAD - INTEGRACIJA ZABBIX SUSTAVA SA LARADOCK MREŽNIM POSLUŽITELJOM

U praktičnom dijelu rada detaljno su opisani koraci potrebni za integraciju Zabbix sustava s Laradock mrežnim poslužiteljem. Integracija omogućava centralizirani nadzor aplikacija i infrastrukture unutar kontejneriziranog okruženja, što je ključno za održavanje visoke dostupnosti i performansi aplikacija razvijenih s pomoću Laravel razvojnog okruženja. Proces obuhvaća instalaciju i konfiguraciju Zabbix poslužitelja i agenta unutar Laradock okruženja, postavljanje potrebnih kontejnera, kao i konfiguraciju nadzornih objekata za učinkovito praćenje svih relevantnih metrika.

### 4.1. Instalacija i konfiguracija Laradocka

Instalacija i konfiguracija Laradocka ključni su koraci za uspostavljanje radnog okruženja za razvoj Laravel aplikacija uz podršku za kontejnerizaciju. Laradock nudi unaprijed definirane Docker kontejnerizirane usluge koje olakšavaju razvoj i implementaciju aplikacija. U nastavku su navedeni koraci potrebni za instalaciju i konfiguraciju Laradocka.

*Koraci za instalaciju i konfiguraciju Laradocka:*

#### 1. Preuvjeti

Potrebno je provjeriti postoji li instaliran Docker i Docker Compose na sustavu. Mogu se preuzeti i instalirati sa službenih mrežnih stranica:

- <https://www.docker.com/>
- <https://docs.docker.com/compose/>

#### 2. Kloniranje Laradock repozitorija

Putem otvorenog terminala navigira se do direktorija u kojem se želi postaviti projekt. Zatim se klonira Laradock repozitorij:

```
git clone https://github.com/laradock/laradock.git
```

#### 3. Postavljanje Laravel aplikacije

Ako ne postoji Laravel aplikacija, može se stvoriti koristeći Composer:

```
composer create-project --prefer-dist laravel/laravel my-laravel-app
```

#### 4. Konfiguracija .env datoteke

U Laradock direktoriju potrebno je pronaći .env.example datoteku i preimenovati je u .env:

```
cd laradock
cp .env.example .env
```

U .env datoteci, odrede se potrebne varijable okruženja. Na primjer, naziv baze podataka, korisničko ime i lozinka.

#### 5. Instalacija i pokretanje kontejnera

Potrebno je pokrenuti Laradock usluge koristeći Docker Compose. U terminalu, unutar Laradock direktorija, pokreće se:

```
docker-compose up -d nginx mysql
```

Ova komanda instalira i pokreće Nginx i MySQL kontejner, koji su potrebni za Laravel aplikaciju.

#### 6. Provjera instalacije

Nakon pokretanja kontejnera, potrebna je provjera jesu li kontejneri ispravno pokrenuti:

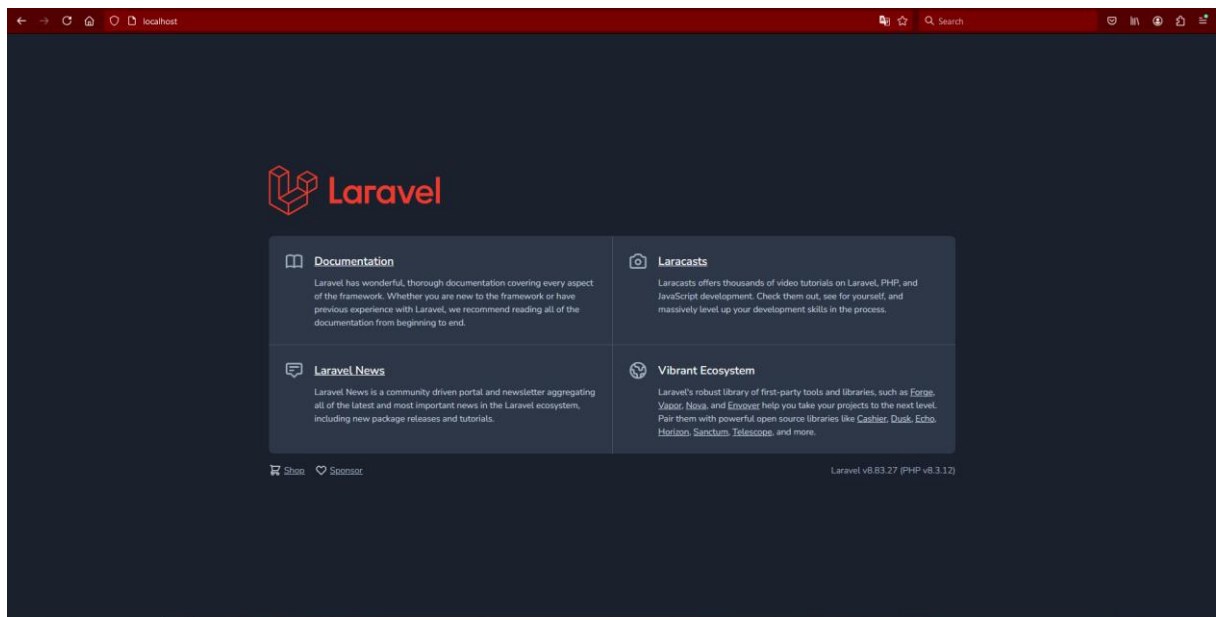
```
docker-compose ps
```

Ako su svi kontejneri u statusu Up, Laradock je uspješno instaliran i konfiguriran.

#### 7. Pristup aplikaciji

Potrebno je otvoriti mrežni preglednik i otići na <http://localhost>. Trebala bi se vidjeti početna stranica Laravel aplikacije (Slika 7).

Ovim koracima, Laradock je uspješno instaliran i konfiguriran, čime se stvara temelj za daljnju integraciju Zabbix sustava. Sljedeći korak je postavljanje Zabbix kontejnera unutar ovog okruženja. (Laradock (2023), *Laradock Documentation*, preuzeto s <https://laradock.io>); (Laravel (2022), *Laravel Documentation*, preuzeto s <https://laravel.com/docs>) (pristupljeno 28.8.2024.)



Slika 7. Laravel aplikacija (osobna izrada, <http://localhost>)

## 4.2. Postavljanje Zabbix kontejnera

Postavljanje Zabbix kontejnera unutar Laradock okruženja omogućava učinkovito praćenje resursa i performansi Laravel aplikacija. Sljedeći koraci omogućavaju konfiguraciju i pokretanje Zabbix poslužitelja, frontend sučelja i Zabbix agenta koji će komunicirati s Laravel aplikacijom, a što je potrebno za uspješno postavljanje Zabbix kontejnera.

*Koraci za postavljanje Zabbix kontejnera:*

1. Navigacija do Laradock direktorija

Prvo se otvori terminal i navigira do Laradock direktorija unutar Laravel projekta.

2. Konfiguracija .env datoteke za Zabbix

U Laradock direktoriju, otvori se .env datoteka i provjere opcije vezane za Zabbix. Ako ne postoje, dodaju se sljedeće varijable (Kôd 1):

```
ZBX_DB_HOST=mysql  
ZBX_DB_NAME=zabbix  
ZBX_DB_USER=zabbix  
ZBX_DB_PASSWORD=zabbix
```

Kôd 1. Konfiguracija .env datoteke

Ove postavke omogućavaju Zabbixu da se poveže s MySQL kontejnerom koji će se koristiti za pohranu podataka.

### 3. Uređivanje docker-compose.yml datoteke

U Laradock direktoriju, otvori se docker-compose.yml datoteka i dodaju se sljedeći servisi za Zabbix (Kôd 2):

```
zabbix-server:
  image: zabbix/zabbix-server-mysql:latest
  environment:
    DB_SERVER_HOST: mysql
    MYSQL_USER: zabbix
    MYSQL_PASSWORD: zabbix
    ZBX_JAVASCRIPT: true
  ports:
    - "10051:10051"
  depends_on:
    - mysql
zabbix-web-nginx:
  image: zabbix/zabbix-web-nginx-mysql:latest
  environment:
    DB_SERVER_HOST: mysql
    MYSQL_USER: zabbix
    MYSQL_PASSWORD: zabbix
    ZBX_SERVER_HOST: zabbix-server
  ports:
    - "8080:8080"
  depends_on:
    - zabbix-server
```

Kôd 2. Konfiguracija docker-compose.yml datoteke

Ove postavke definiraju Zabbix poslužitelj i mrežno sučelje, uključujući potrebne varijable okruženja i ovisnosti između servisa.

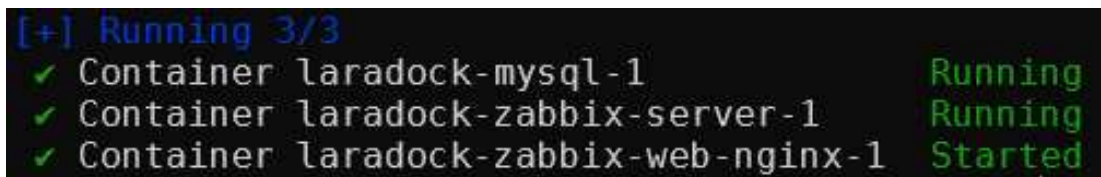
### 4. Pokretanje Zabbix kontejnera

Nakon konfiguracije docker-compose.yml, pokreću se Zabbix kontejneri koristeći

sljedeću komandu:

```
docker-compose up -d zabbix-server zabbix-web-nginx-mysql
```

Ova naredba pokrenuti će Zabbix poslužitelj i mrežno sučelje u pozadini (Slika 8).



```
[+] Running 3/3
✓ Container laradock-mysql-1      Running
✓ Container laradock-zabbix-server-1  Running
✓ Container laradock-zabbix-web-nginx-1 Started
```

Slika 8. Pokrenuti Zabbix kontejneri (osobna izrada, Ubuntu shell)

#### 5. Provjera statusa kontejnera

Potrebno je provjeriti jesu li Zabbix kontejneri ispravno pokrenuti:

```
docker-compose ps
```

Status kontejnera trebao bi biti Up za oba Zabbix servisa.

#### 6. Pristup Zabbix mrežnom sučelju

Otvori se mrežni preglednik i ide na <http://localhost:8080>. Trebala bi se vidjeti stranica za prijavu Zabbix mrežnog sučelja.

Korisničko ime i lozinka za prijavu su:

- Korisničko ime: *Admin*
- Lozinka: *zabbix*

#### 7. Prva konfiguracija Zabbix-a

Nakon prijave, potrebno je slijediti upute na ekranu kako bi se završila početna konfiguracija Zabbix sustava, uključujući odabir baze podataka.

Ovim koracima, Zabbix kontejner je uspješno postavljen unutar Laradock okruženja, što omogućava praćenje performansi i resursa Laravel aplikacije. Sljedeći korak je konfiguracija Zabbix agenata za nadzor specifičnih resursa. (Zabbix SIA (2023), *Zabbix Documentation*, preuzeto s <https://www.zabbix.com>); (Laradock (2023), *Laradock Documentation*, preuzeto s <https://laradock.io>); (Docker Hub, (2023), *Zabbix Docker Images*, preuzeto s <https://hub.docker.com/r/zabbix/zabbix-server-mysql>) (pristupljeno 29.8.2024.)

### 4.3. Konfiguracija Zabbix agenata

Konfiguracija Zabbix agenata ključna je za prikupljanje i slanje podataka o performansama iz aplikacija i infrastrukture prema Zabbix poslužitelju. U nastavku su prikazani koraci potrebni za instalaciju i konfiguraciju Zabbix agenata unutar Laradock okruženja, što omogućava efikasno praćenje raznih resursa, uključujući Laravel aplikacije.

*Koraci za konfiguraciju Zabbix agenata:*

1. Dodavanje Zabbix agenta u docker-compose.yml

Otvori se docker-compose.yml datoteka unutar Laradock direktorija i dodaje sljedeći servis za Zabbix agenta (Kôd 3):

```
zabbix-agent:  
  image: zabbix/zabbix-agent:latest  
  environment:  
    ZBX_SERVER_HOST: zabbix-server  
    ZBX_HOSTNAME: laravel-app  
  depends_on:  
    - zabbix-server
```

Kôd 3. Dodavanje zabbix agenata u docker-compose.yml datoteku

Ove postavke definiraju Zabbix agenta koji će komunicirati sa Zabbix poslužiteljom, koristeći naziv hosta Laravel aplikacije.

2. Pokretanje Zabbix agenta

Nakon što se doda Zabbix agent u docker-compose.yml, potrebno je pokrenuti agenta koristeći sljedeću komandu:

```
docker-compose up -d zabbix-agent
```

Ova komanda pokrenuti će Zabbix agenta u pozadini.

3. Provjera statusa Zabbix agenta

Provjeri se je li Zabbix agent ispravno pokrenut:

```
docker-compose ps
```

Status kontejnera za Zabbix agenta trebao bi biti Up.

#### 4. Konfiguracija Zabbix agenta

Ako je potrebno dodatno prilagoditi konfiguraciju Zabbix agenta, otvori se datoteka `zabbix_agentd.conf` unutar kontejnera. Moguće je to učiniti na sljedeći način:

```
docker exec -it <container_id> /bin/bash
```

Zatim, otvaranje datoteke:

```
vim /etc/zabbix/zabbix_agentd.conf
```

Ovdje je moguće konfigurirati razne parametre kao što su:

- **Hostname:** ime hosta koji se želi pratiti
- **Server:** IP adresa ili naziv Zabbix poslužitelja
- **LogFile:** putanja do datoteke sa zapisima.

#### 5. Dodavanje hosta u Zabbix poslužitelj

Potrebno je prijaviti se na Zabbix mrežno sučelje na <http://localhost:8080> koristeći navedene podatke za prijavu:

- ide se na *Configuration* > *Hosts* i klikne na *Create host*
- unese se ime hosta (npr. "laravel-app"), odabere se grupa (npr. "Linux servers") i unese IP adresa Zabbix agenta (obično 127.0.0.1 ili IP adresa kontejnera)
- dodaje se predložak (template) za prikupljanje podataka (npr. "Template OS Linux")
- potrebno je spremi promjene.

#### 6. Testiranje konfiguracije

Nakon što je host dodan u Zabbix poslužitelj testira se veza, tako da se vraća na stranicu s hostovima i provjerava status hosta. Status bi trebao biti Up, što znači da Zabbix agent uspješno šalje podatke na poslužitelj.

Ovim koracima, Zabbix agent je uspješno konfiguriran za praćenje Laravel aplikacije unutar Laradock okruženja. Ova konfiguracija omogućava prikupljanje ključnih metrika i informacija koje su potrebne za učinkovito upravljanje performansama i resursima. (Zabbix SIA (2023), *Zabbix Documentation*, preuzeto s <https://www.zabbix.com>); (Laradock (2023), *Laradock*

*Documentation*, preuzeto s <https://laradock.io>); (Docker Hub, (2023), *Zabbix Docker Images*, preuzeto s <https://hub.docker.com/r/zabbix/zabbix-server-mysql>) (pristupljeno 29.8.2024.)

#### 4.4. Postavljanje nadzornih objekata

Postavljanje nadzornih objekata u Zabbix sustavu ključno je za prikupljanje relevantnih metrika i informacija o performansama aplikacija i infrastrukture. Ova faza uključuje definiranje elemenata koje će Zabbix pratiti, kao što su hostovi, metrike i pravila upozorenja. U nastavku su opisani koraci potrebni za postavljanje nadzornih objekata u Zabbixu.

*Koraci za postavljanje nadzornih objekata:*

1. Prijava na Zabbix mrežno sučelje

Prijava se izvršava koristeći korisničko ime *Admin* i lozinku *zabbix* na <http://localhost:8080> adresi s pomoću mrežnog preglednika.

2. Odabir hosta za praćenje

Na meniju *Configuration > Hosts* vidjet će se popis hostova koji su već dodani. Odabere se host (npr. "laravel-app") za koji se želi postaviti nadzorne objekte.

3. Dodavanje nadzornih objekata

Nakon odabira hosta, klikne se na *Items*. Ovdje se mogu vidjeti postojeći nadzorni objekti. Klikne se na *Create item* za dodavanje novog nadzornog objekta.

4. Definiranje nadzornog objekta

U prozoru za kreiranje nadzornog objekta, potrebno je popuniti sljedeća polja:

- Name: unese se naziv nadzornog objekta (npr. "CPU Usage")
- Type: odabere se tip prikupljanja podataka (npr. "Zabbix agent")
- Key: unese se ključ koji definira metriku (npr. system.cpu.util za korištenje CPU-a)
- Type of information: odabere se tip informacija (npr. "Numeric (float)")
- Update interval: potrebno je odrediti koliko će se često će podaci prikupljati (npr. 60 sekundi)
- Applications: dodaje se aplikacija kojoj pripada ovaj nadzor (npr. "CPU").

Nakon popunjavanja svih potrebnih polja, klikne se na *Add*.



#### 4.5. Postavljanje pravila upozorenja

Nakon što su dodani nadzorni objekti, mogu se postaviti pravila upozorenja. Navigira se na *Configuration > Actions*. Klikne se na *Create action*. Ovdje se mogu definirati uvjeti pod kojima će se upozorenja slati.

Potrebno je postaviti uvjete, poput:

- *Name*: unese se naziv akcije (npr. "High CPU Usage")
- *Conditions*: odaberu se uvjeti pod kojima će se akcija pokrenuti (npr. kada CPU Usage prelazi 80%).

Odabire se radnja koja se događa kada se uvjeti ispune, kao što su slanje e-pošte ili obavijesti.

#### *Testiranje nadzornih objekata*

Nakon postavljanja nadzornih objekata i pravila upozorenja, slijedi provjera je li Zabbix počeo prikupljati podatke. Host se odabire na meniju *Monitoring > Latest data*. Provjeravaju se prikupljeni podatci za novododane nadzorne objekte.

#### *Praćenje performansi*

U ovoj fazi može se pratiti performanse i analizu podataka na Zabbix dashboardu. Prate se metrike i provjerava se je li Zabbix pravilno postavljen za nadzor svih relevantnih objekata.

Ovim koracima, postavljeni su nadzorni objekti unutar Zabbix sustava, omogućavajući prikupljanje i analizu ključnih metrika za Laravel aplikaciju. Ova konfiguracija je ključna za učinkovito praćenje performansi i identifikaciju potencijalnih problema u stvarnom vremenu. (Zabbix SIA (2023), *Zabbix Documentation*, preuzeto s <https://www.zabbix.com>) (pristupljeno 2.9.2024.)

## 5. ZAKLJUČAK

U ovom završnom radu istražen je proces integracije nadzornog sustava Zabbix s mrežnom aplikacijom pokrenutom unutar Laradock okruženja. Kroz teorijski dio rada, objašnjene su ključne sastavnice Zabbix sustava, kontejnerizacija s pomoću Dockera te Laravel mrežno okruženje. Navedena kombinacija tehnologija omogućava razvoj i održavanje robusnih i učinkovitih aplikacija, uz istodobno osiguranje visoke razine nadzora i performansi.

Zabbix je programska podrška otvorenog koda koja izvršava nadzor nad odabranim sastavnicama IT sustava. Zabbix nadzire odabrane parametre mreže te ispravnost i cjelovitost poslužitelja, virtualnih strojeva, aplikacija, usluga, baza podataka, mrežnih stranica, oblaka i dr. Koristi prilagodljiv sustav interaktivnih obavijesti koje omogućavaju slanje upozorenja putem e-pošte za gotovo svaki događaj, čime omogućava brzu reakciju na probleme poslužitelja. Zabbix nudi kvalitetno izvještavanje i vizualizaciju događaja na temelju pohranjenih podataka, što ga čini pogodnim za planiranje kapaciteta IT sustava. Prednost Zabbixa je u tome što može proizvoljno definirati svoje zadaće. On u sebi već sadrži niz predložaka s gotovim zadaćama za razne operativne sustave, ali u praksi se često nailazi na zadatak koji nije podržan u standardu, te se isti mora doraditi.

Praktični dio rada usredotočen je na korake potrebne za instalaciju i konfiguraciju Zabbix poslužitelja, agenata i nadzornih objekata. Osnovna prednost korištenja Zabbixa u kontekstu Laravela leži u njegovoj sposobnosti da prikuplja podatke o resursima u stvarnom vremenu, što omogućava pravovremenu identifikaciju potencijalnih problema. Ova funkcionalnost je od ključne važnosti za osiguranje stabilnosti i efikasnosti aplikacija, posebno u okruženjima koja zahtijevaju visoku dostupnost i performanse.

Tijekom implementacije, naglašena je jednostavnost korištenja Laradock okruženja koje omogućava brzu i jednostavnu konfiguraciju aplikacijskih kontejnera, uključujući Zabbix i Laravel. Korištenje Docker tehnologije omogućava kreiranje izoliranih okruženja koja su jednostavna za upravljanje, testiranje i distribuciju. Ova kontejnerizacija dodatno doprinosi prilagodljivosti razvoja i olakšava suradnju među timovima.

Također, kroz integraciju Zabbixa s Laravel aplikacijom, omogućena je vizualizacija ključnih metrika i performansi putem Zabbix mrežnog sučelja. To olakšava analizu podataka i donošenje informiranih odluka na temelju prikupljenih informacija. Prikazivanje upozorenja u slučaju

odstupanja od očekivanih performansi omogućuje pravovremeno djelovanje i održavanje optimalne razine usluge.

Na temelju izvedenih koraka, može se zaključiti da je integracija Zabbix sustava s Laravelom u Laradock okruženju učinkovita i korisna strategija za praćenje i optimizaciju mrežnih aplikacija.

Stoga, ovaj rad daje prikaz praktične osnove za daljnje istraživanje i razvoj nadzornih sustava, kao i za unapređenje postojećih aplikacija. U budućnosti bi se mogla istražiti i implementacija dodatnih funkcionalnosti Zabbix sustava, kao što su analitika podataka i automatizacija reakcija na određena događanja.

Integracija Zabbixa s Laraveldom i Dockerom predstavlja značajan korak prema modernizaciji pristupa razvoju i održavanju aplikacija, čime se omogućava bolja korisnička iskustva i veća učinkovitost poslovnih procesa. Ovaj rad pruža informacije i korake za implementaciju i osnove za buduće istraživanje u području nadzora i upravljanja performansama modernih aplikacija.

## LITERATURA

1. Uytterhoeven, P., Olups, R. (2019), *Zabbix 4 Network Monitoring - Third Edition: Monitor the performance of your network devices and applications using the all-new Zabbix 4.0*, Birmingham, UK: Packt Publishing
2. Dalle Vacche, A. (2015), *Mastering Zabbix 2nd Edition*, Birmingham, UK: Packt Publishing
3. Merkel, D. (2014), *Docker: lightweight Linux containers for consistent development and deployment*, *Linux Journal*, 2014 (239), 2.
4. Zabbix, The story of how Zabbix software became one of the worlds most demanded IT monitoring solutions, April 17, 2020, <https://www.zabbix.com/la/pr/pr280>
5. Zabbix, forum, <https://www.zabbix.com/forum/>
6. Administracija Portala Sustav, Instalirajte Zabbix na RedHat Centos 7 sustave <https://hr.admininfo.info/instalar-zabbix-en-sistemas-redhat-centos-7>, svibanj 2024.
7. CARNet, SNMP protokol NCERT-PUBDOC-2010-09-313, Revizija, str. 26/30
8. Drageljević, M. (2019), *Zabbix - ili si OK ili si problem*, Rovinj: Login Software, 19. HROUG konferencija
9. zabbix.com, <https://www.zabbix.com/documentation/current/en/manual/concepts/server>
10. Zabbix SIA (2023), *Zabbix Documentation*, <https://www.zabbix.com/documentation/current/manual/installation/install>
11. Zabbix SIA (2023), *Zabbix Agent Configuration*, [https://www.zabbix.com/documentation/current/manual/appendix/install/agent\\_configuration](https://www.zabbix.com/documentation/current/manual/appendix/install/agent_configuration)
12. Zabbix SIA (2023), Getting Started with Zabbix Web Interface, preuzeto s [https://www.zabbix.com/documentation/current/manual/web\\_interface/getting\\_started](https://www.zabbix.com/documentation/current/manual/web_interface/getting_started)
13. Zabbix SIA (2023), *Monitoring Hosts in Zabbix*, preuzeto s <https://www.zabbix.com/documentation/current/manual/config/host/overview>
14. Laradock (2023) *Laradock Documentation*, preuzeto s <https://laradock.io>
15. Docker Inc. (2023), *Docker Documentation*, preuzeto s <https://docs.docker.com>
16. Otwell, T. (2022), *Laravel Documentation*, Laravel
17. Docker Hub, (2023), *Zabbix Docker Images*, preuzeto s <https://hub.docker.com/r/zabbix/zabbix-server-mysql>
18. <https://www.docker.com/>
19. <https://docs.docker.com/compose/>
20. <http://localhost>

## SAŽETAK

Zabbix je sustav otvorenog koda, dinamičan mehanizam obavijesti, a pruža učinkovit način upozorenja u slučaju neželjenih događaja u okviru IT sustava. U ovom završnom radu prikazane su praktične smjernice koje će omogućiti korisnicima da uspješno implementiraju Zabbix u svojim Laravel projektima, koristeći prednosti koje kontejnerizacija nudi putem Dockera. Kombinacija Zabbixa i Laradocka donosi značajne prednosti za razvojna i operativna okruženja koja zahtijevaju nadzor i kontrolu nad aplikacijama i resursima. Zabbix omogućava detaljno praćenje performansi, zdravlja svih sastavnica sustava i mrežnih aplikacija, a Laradock osigurava brzu implementaciju razvojnih okruženja putem Docker kontejnerizacije. Takva integracija omogućava centralizirani nadzor aplikacija i infrastrukture unutar kontejneriziranog okruženja, što je ključno za održavanje visoke dostupnosti i performansi aplikacija razvijenih s pomoću Laravel radnog okruženja.

**Ključne riječi:** Zabbix, Laravel, Docker, Laradock, nadzor IT sustava, kontejnerizacija, mrežna aplikacija

## SUMMARY

Zabbix is an open source dynamic notification system that provides an effective way of warning in case of unwanted events within the IT system. This paper presents practical guidelines that will enable users to successfully implement Zabbix in their Laravel projects, using the advantages that containerization offers through Docker. The combination of Zabbix and Laradock brings significant benefits for development and operational environments that require monitoring and control over applications and resources. Zabbix enables detailed monitoring of performance, health of all system components and web applications, and Laradock ensures fast implementation of development environments through Docker containerization. Such integration enables centralized monitoring of applications and infrastructure within a containerized environment, which is crucial for maintaining high availability and performance of applications developed using the Laravel framework.

**Keywords:** Zabbix, Laravel, Docker, Laradock, IT system monitoring, containerization, web application