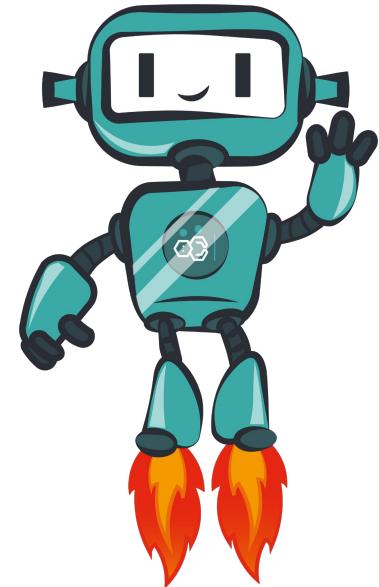
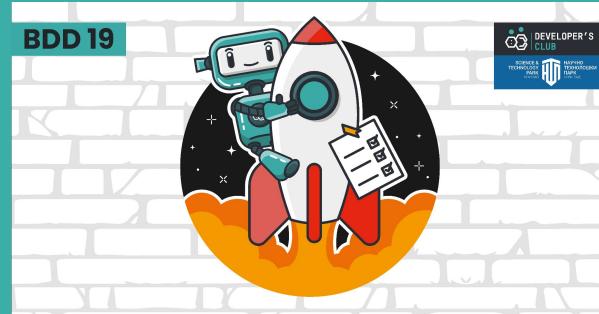




BDD#19 <https://bit.ly/bdd-19>

Production Ready: Launch & Release Technical Must-Haves



Cilj predavanja



- Šta nas sve čeka kada lansiramo app u PROD:
 - Arhitektura (dugovečnost projekta!)
 - Performanse (baze, load testing, security)
 - Docker
 - Testovi, QA
 - Logovi & Monitoring, Disaster Recovery
 - 3rd Party servisi (BANKROT \$\$\$\$ + iznenađenja)
 - Specifičnosti runtime, framework-a u prod env

Arhitektura (1)



```
my_project/
|
├── main.py
├── app.py
├── routes.py
├── helper_functions.py
├── utils.py
├── db.py
├── config.py
└── logger.py
|
└── data/
    ├── backup.sql
    └── temp_data.txt
|
└── static/
    ├── styles.css
    ├── scripts.js
    └── image.png
```

```
|
└── db_queries/
    ├── query1.sql
    └── query2.sql
```

Arhitektura (2)



```
my-spring-boot-app/
|
+-- src/
|   +-- main/
|   |   +-- java/
|   |   |   +-- MySpringBootApp.java
|   |   |   +-- Controller.java
|   |   |   +-- UserController.java
|   |   |   +-- DataService.java
|   |   |   +-- DatabaseConfig.java
|   |   |   +-- Utility.java
|   |   |   +-- EmailService.java
|   |   |   +-- UserRepository.java
|   |   |   +-- UserModel.java
|   |   |   +-- AppProperties.java
|   |   |   +-- SecurityConfig.java
|   |   \-- HelperMethods.java
```

```
db-scripts/
|
+-- schema.sql
|
+-- data.sql
```

Arhitektura (3)



DEVELOPER'S CLUB

SCREAMING ARCHITECTURE

TECH DRIVEN



```
App
└── Controllers
└── Models
└── Views
└── DTOs
└── Database
└── ...
```

FEATURE DRIVEN



```
CarRentalService
└── AutoPicker
└── UserManager
└── PaymentGateway
└── InvoiceGenerator
└── SupportCenter
└── ...
```



Daniel Moka

Arhitektura (4)



DEVELOPER'S
CLUB



Dmitry R. • 2nd

Software developer | Fractional CTO @ CatWatchDog

10h ...

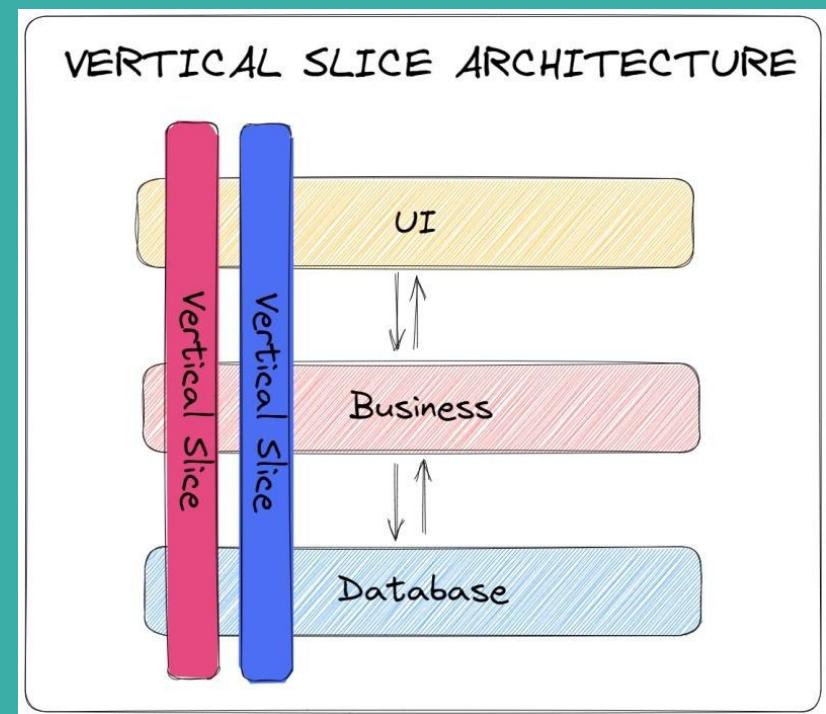
In reality it will be like this

```
📁 CarRentalService
|__ 📁 AutoPicker
|  |__ 📁 API
|  |__ 📁 Models
|  |__ 📁 Views
|  |__ 📁 Controllers
|  |__ 📁 DTOs
|  |__ 📁 Database
|  |__ 📁 UserManager
|  |__ ...
|__ 📁 PaymentGateway
|  |__ ...
|__ 📁 InvoiceGenerator
|  |__ ...
|__ 📁 SupportCenter
|  |__ ...
```

Arhitektura (4)



- > pwt-app
- > pwt-app-impl
- > pwt-app-spec
- > pwt-boot
- > pwt-domain
- > pwt-eventbus
- > pwt-eventbus-kafka
- > pwt-repo-api
- > pwt-repo-jpa
- > pwt-repo-spec
- > pwt-web-api



Arhitektura (5)



```
java/
  com/
    myapp/
      application/          # Application Layer (Use Cases)
        CreateUserUseCase.java
        GetUserUseCase.java
        UpdateUserUseCase.java

    domain/                  # Domain Layer (Entities & Interfaces)
      User.java
      UserRepository.java   # Domain-level interface
      UserService.java      # Business logic interface

    infrastructure/          # Infrastructure Layer (External Systems)
      repository/           # Repos can be swapped out
        JpaUserRepository.java # Implementation with JPA
        MongoUserRepository.java # Implementation with MongoDB
        RedisUserCache.java   # Caching with Redis
        InMemoryUserRepository.java # Testing implementation

    web/                     # Web Layer (Controllers)
      UserController.java   # REST Controller for HTTP requests
      ApiErrorHandler.java  # Central error handling

  config/                  # Configuration Layer
    AppConfig.java          # Application-level configuration
    WebSecurityConfig.java  # Security settings (Spring Security)

  Copy code
```

Arhitektura (6)



DEVELOPER'S CLUB

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
NOVI FEAT	DEV ZNA GDE ŠTA IDE	POMOZ BOŽE, negde???
ZAMENA DELA INFRA	SAMO IMPL, LAKA IZMENA	CELA APP SE MORA REFACTOR I PROVERITI
VIŠE DEVS RADI NA PROJEKTU	MOGU NEZAVISNO	TRKA KO ĆE PRVI COMMIT A KO DA REŠAVA CONFLICTS
NOVI FEAT	DODA SE LAKO, TESTIRA SE LAKO, OGRANIČEM UTICAJ NA OSTATAK APP	CELA APP MORA DA SE TESTIRA I PROVERI DA LI DALJE RADI
RELEASE	MOŽE ČEŠĆE, SVEDENE IZMENE, STABILNO JE!	JEDNOM MESEČNO ILI U 6 MESECI

Arhitektura (6)



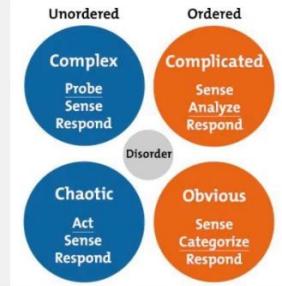
ODGLEDATI BDD 12 !!!
<https://youtu.be/oFJdplApwhA>
@Dejan



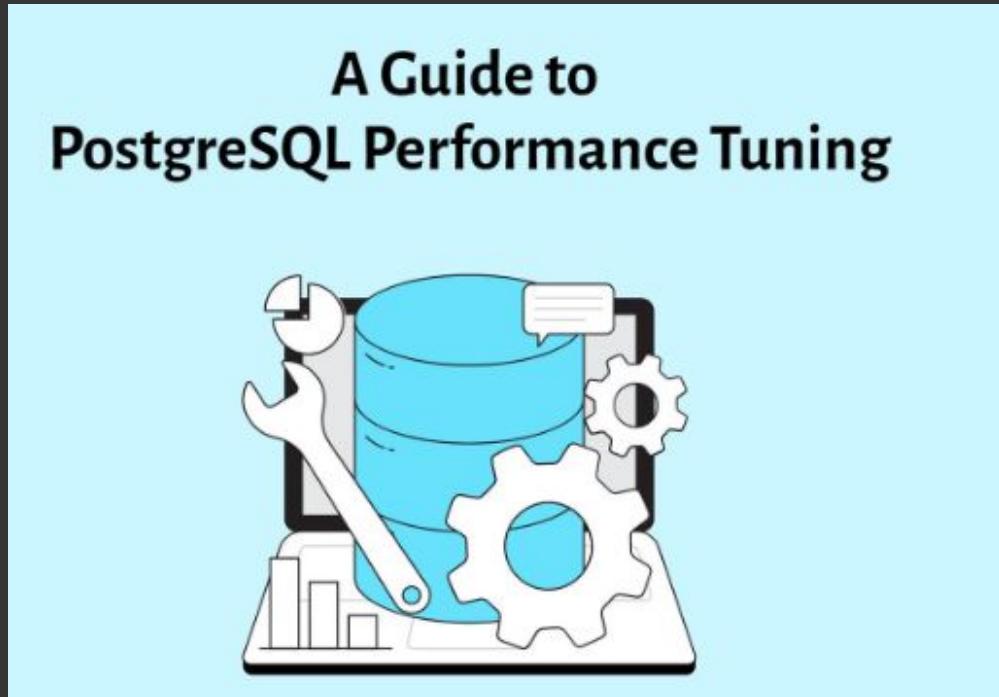
Dejan Milicic · 1st
Software Craftsman

FRED BROOKS: NO SILVER BULLET, 1986

- **Essential** complexity
 - complexity of **domain**
 - cannot be avoided
 - cannot be changed
- **Accidental** complexity
 - complexity of **implementation**
 - problems engineers create and fix



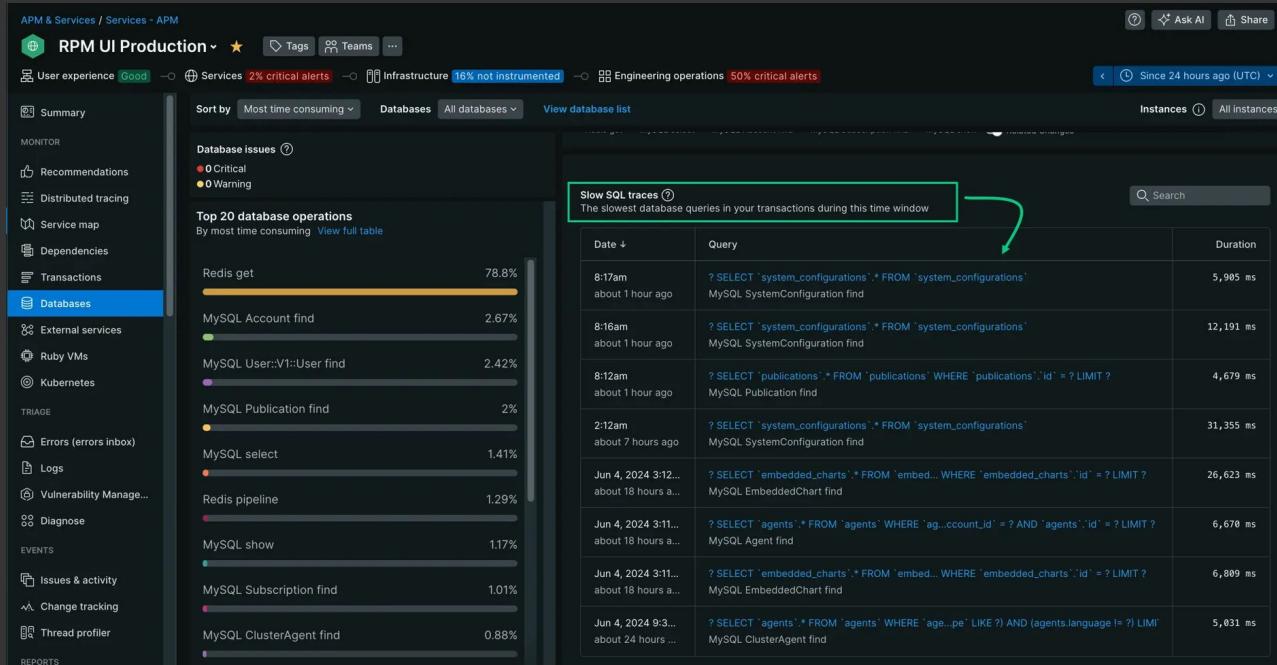
– PERFORMANCE - BAZA - tune



– PERFORMANSE - BAZA - LOAD TEST

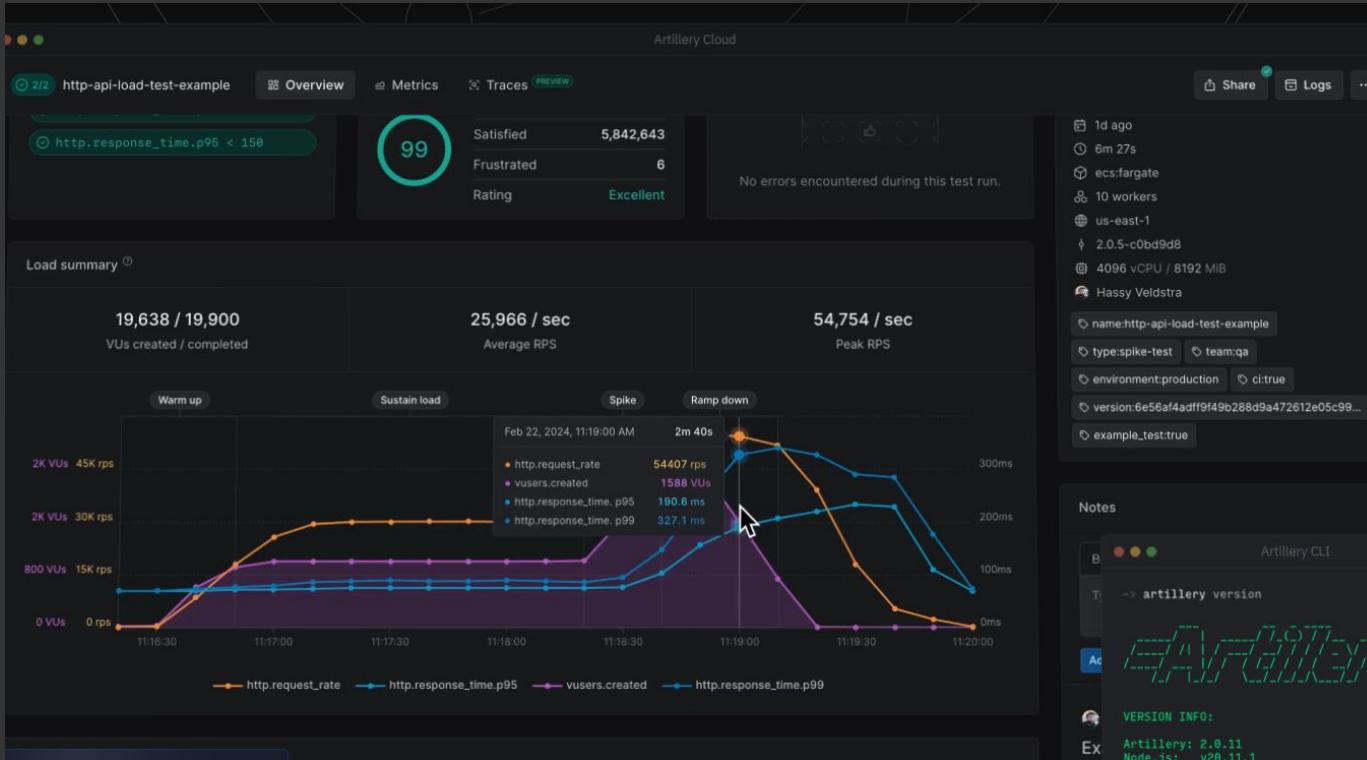
- NEW RELIC (APM, APPLICATION PERFORMANCE MONITORING)
- ODLIČAN ZA INTRO U APMs

- INSTRUMENTACIJA
- NAPASTI SERVER SA 2x BROJEM KORISNIKA



– PERFORMANSE - BAZA - LOAD TEST

- <https://www.artillery.io/> ili samo lansirati ON-DEMAND EC2 instances, FLOTU TAKVIH (plaća se po minutu)



– PERFORMANCE - BAZA - explain

- EXPLAIN (ANALYZE, BUFFERS) SELECT count(*) FROM c WHERE pid = 1 AND cid > 200;

QUERY PLAN

Aggregate (cost=219.50..219.51 rows=1 width=8) (actual time=2.808..2.809 rows=1 loops=1)

 Buffers: shared read=45

 I/O Timings: read=0.380

 -> Seq Scan on c (cost=0.00..195.00 rows=9800 width=0) (actual time=0.083..1.950 rows=9800 loops=1)

 Filter: ((cid > 200) AND (pid = 1))

 Rows Removed by Filter: 200

 Buffers: shared read=45

 I/O Timings: read=0.380

Planning:

 Buffers: shared hit=48 read=29

 I/O Timings: read=0.713

Planning Time: 1.673 ms

Execution Time: 3.096 ms

(13 rows)

– PERFORMANCE - BAZA - explain

Very Bad Query Stats:

Metric	Value
Total Execution Time	16,939.915 ms
Major Time Consumers	
- Merge Join Time	8,279.225 ms
- Sort Operations Time	3,019.779 ms
- Nested Loop Time	1,983.626 ms
- Index Only Scan Time	3,573.075 ms
Percentage of Total Time	
- Merge Join	48.9%
- Sort Operations	17.8%
- Nested Loops	11.7%
- Index Only Scan	21.1%

Comment: This query performs poorly due to heavy reliance on expensive operations like Merge Joins, Sorts, and Nested Loops, leading to high execution times.

– PERFORMANCE - BAZA - explain

- EXPLAIN SELECT count(*) FROM c WHERE pid = 1 AND cid > 200;

Good Query Stats:

Metric	Value
Total Execution Time	500 ms
Major Time Consumers	
- Hash Join Time	200 ms
- Index Scan Time	150 ms
- Aggregate Functions Time	50 ms
Percentage of Total Time	
- Hash Join	40%
- Index Scans	30%
- Aggregate Functions	10%

Per node type stats

node type	count	sum of times	% of query
Append	1	0.096 ms	0.0 %
GroupAggregate	1	60.778 ms	0.4 %
Hash	1	0.138 ms	0.0 %
Hash Join	1	0.773 ms	0.0 %
Index Only Scan	1	3,573.075 ms	21.1 %
Index Scan	3	18.375 ms	0.1 %
Merge Join	1	8,279.225 ms	48.9 %
Nested Loop	5	1,983.626 ms	11.7 %
Nested Loop Left Join	2	4.274 ms	0.0 %
Seq Scan	37	7.070 ms	0.0 %
Sort	4	3,019.779 ms	17.8 %
Unique	1	0.075 ms	0.0 %

Per table stats

Table name	Scan count	Total time	% of query
scan type	count	sum of times	% of table
component_summary	1	0.121 ms	0.0 %
Seq Scan	1	0.121 ms	100.0 %
lts_component_vulnerability_remediation	1	7.410 ms	0.0 %
Index Scan	1	7.410 ms	100.0 %
lw_component_mapping_0000	1	0.003 ms	0.0 %
Seq Scan	1	0.003 ms	100.0 %
lw_component_mapping_0001	1	0.002 ms	0.0 %
Seq Scan	1	0.002 ms	100.0 %
lw_component_mapping_0002	1	0.001 ms	0.0 %
Seq Scan	1	0.001 ms	100.0 %
lw_component_mapping_0003	1	0.001 ms	0.0 %

– PERFORMANCE - BAZA

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
QUERY PERF	FAST, INDEXED	SLOW, FULL TABLE SCAN
RESOURCE UTILIZATION	ODRAĐEN TUNING MINIMAL I/O!!!! CPU i MEM usage < 80%	DEFAULT CONFIG DISK THRASHING! 100% CPU/MEM
TRANSACTIONS	MINIMAL LOCKING/BLOCKING CORRECT ISOLATION LEVELS PER SCENARIO	DEADLOCKS ANOMALIES
MIGRACIJE	BACKUP SPREMAN ROLLBACK SPREMAN VERSION CONTROLLED	DDL AND PRAY OPASNO ZA CEO BIZNIS

PERFORMANCE - FRONT



Core Web Vitals



(Loading)



(Interactivity)



(Visual Stability)

LCP

Largest Contentful Paint



FID

First Input Delay

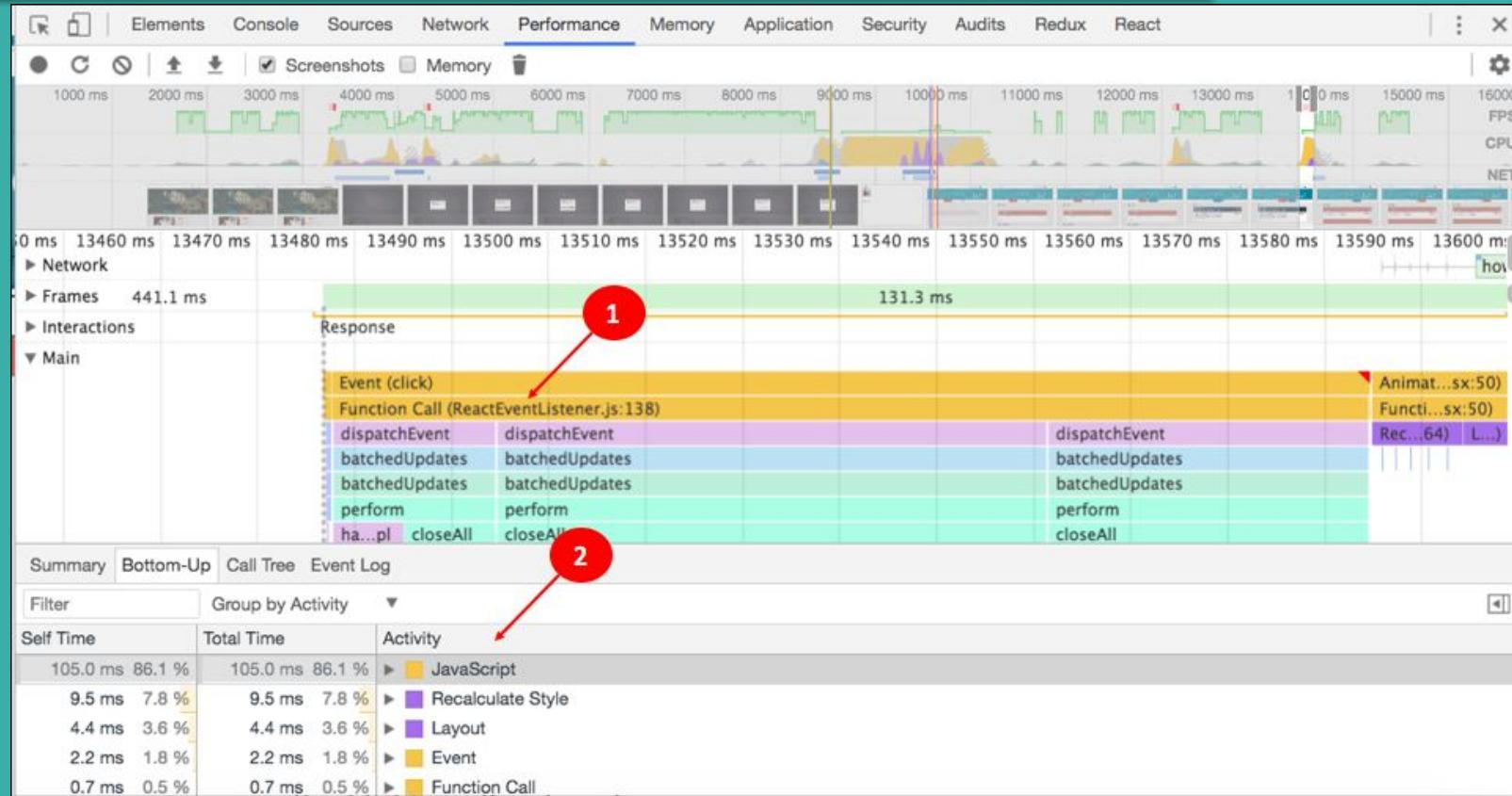


CLS

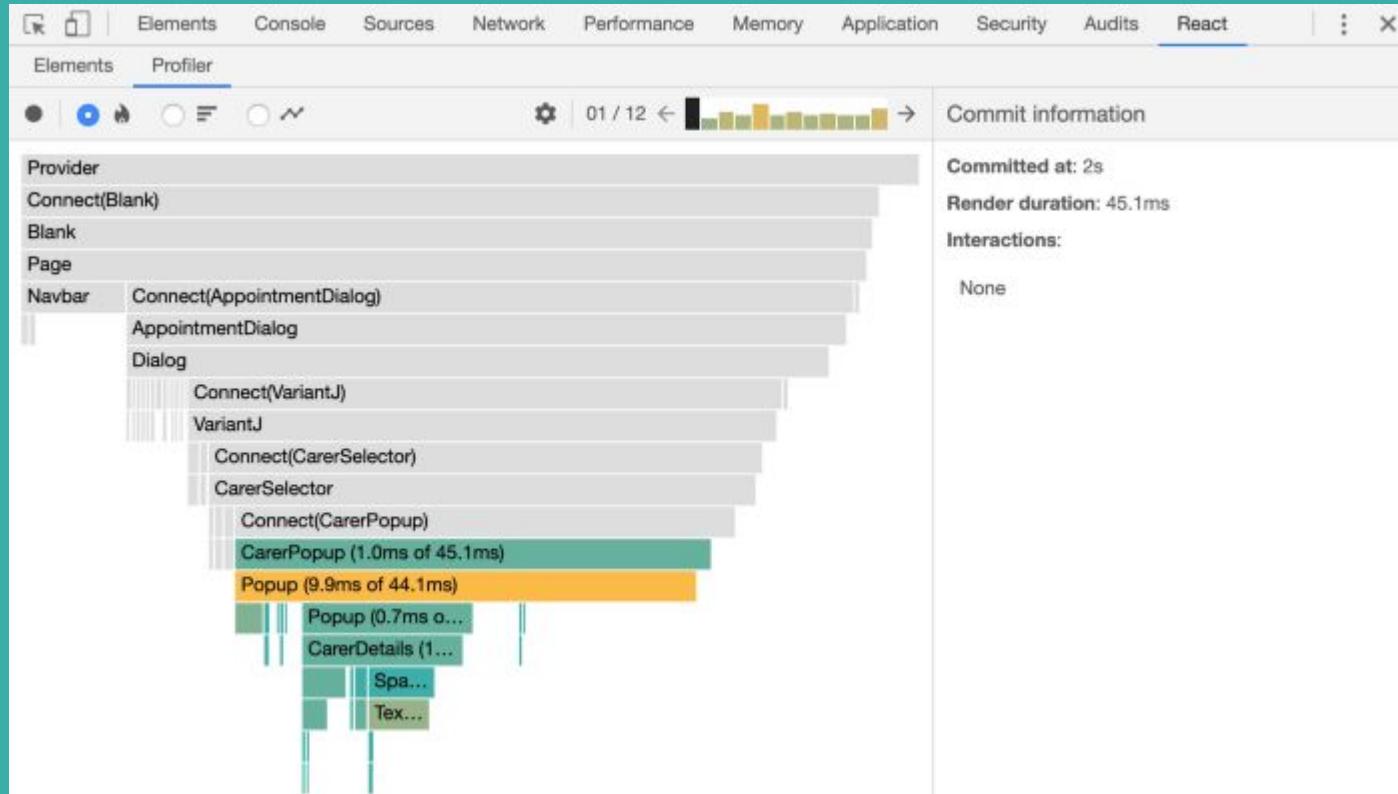
Cumulative Layout Shift



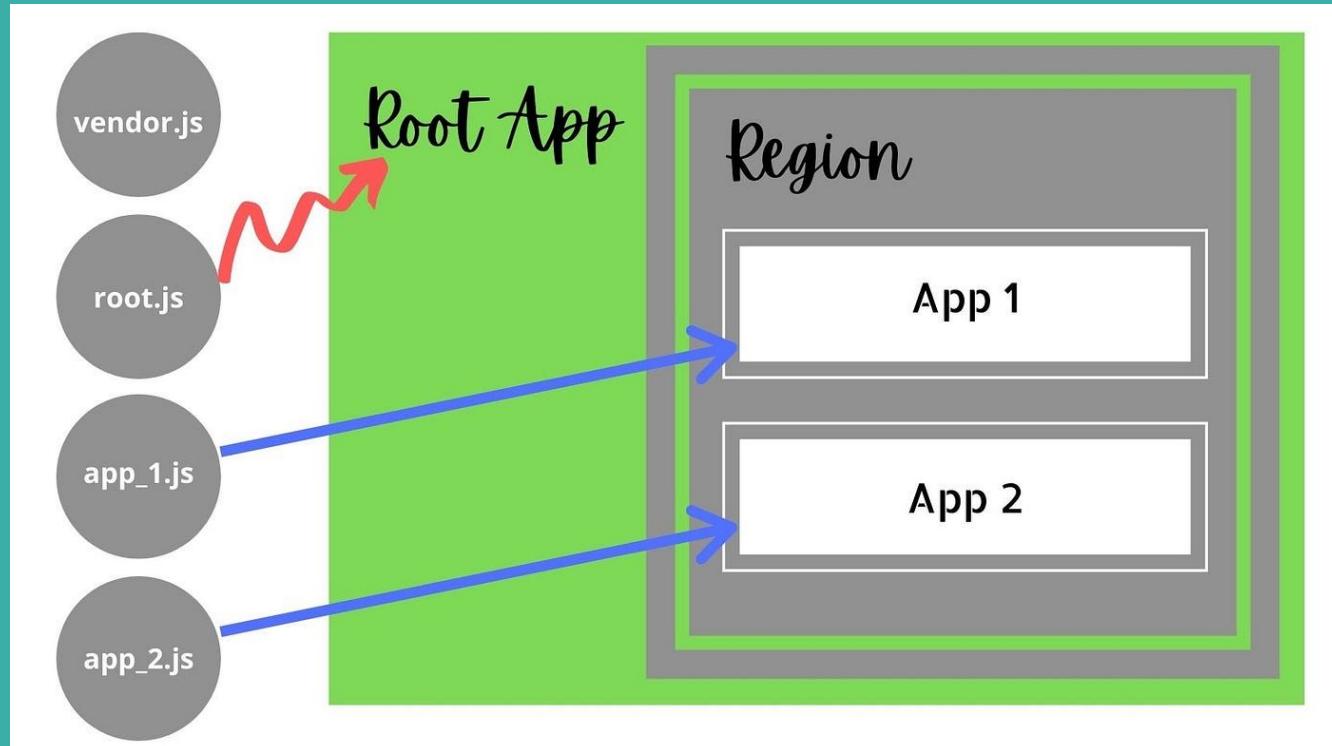
PERFORMANCE - FRONT



PERFORMANCE - FRONT



PERFORMANCE - FRONT

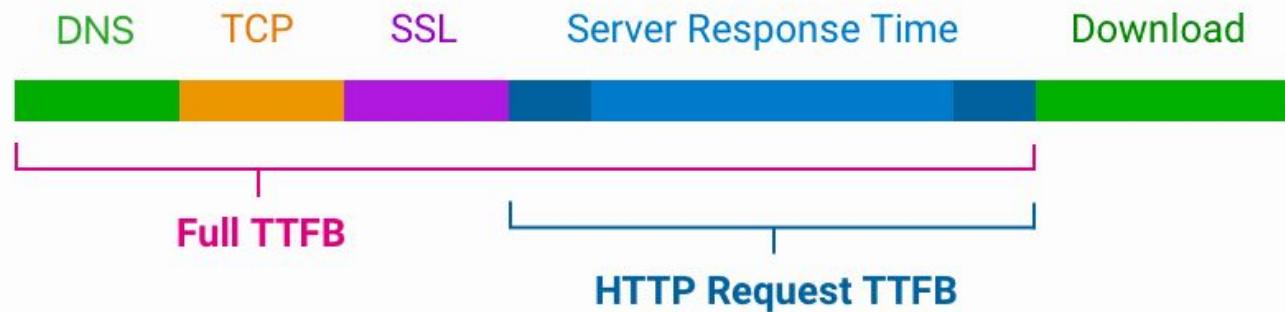


PERFORMANCE - FRONT



What is time to First Byte?

And how can I optimize it?



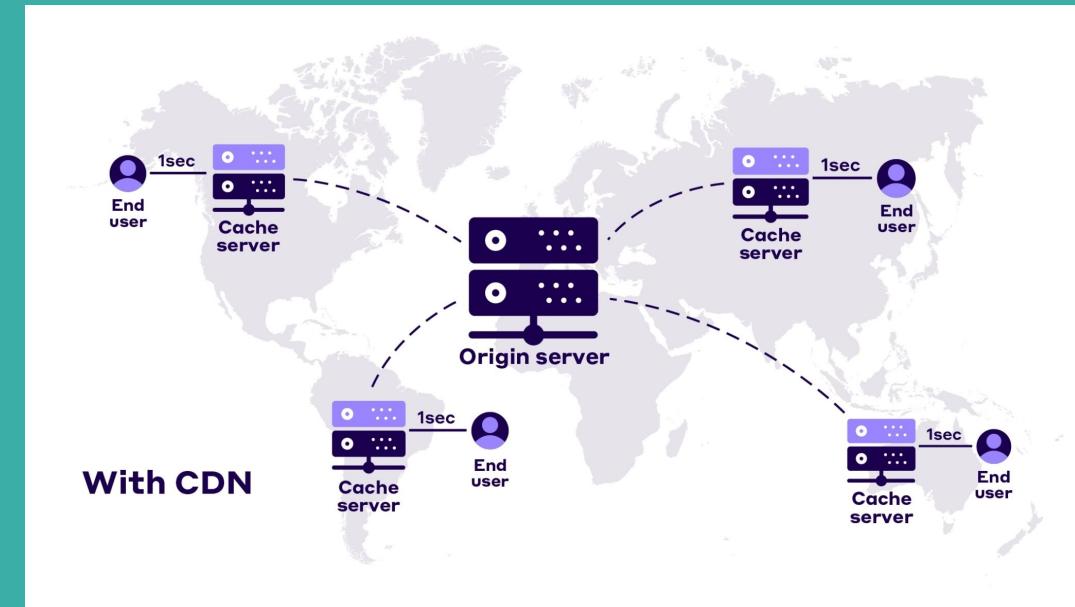
PERFORMANCE - FRONT



- Code splitting
- CDN
- Speculation Rules
- HTTP3

html

```
<script type="speculationrules">
{
  "prerender": [
    {
      "source": "list",
      "urls": ["/about", "/contact"]
    }
  ]
}
</script>
```



PERFORMANCE - FRONT



https://developer.chrome.com/docs/web-platform/prerender-pages?utm_source=devtools

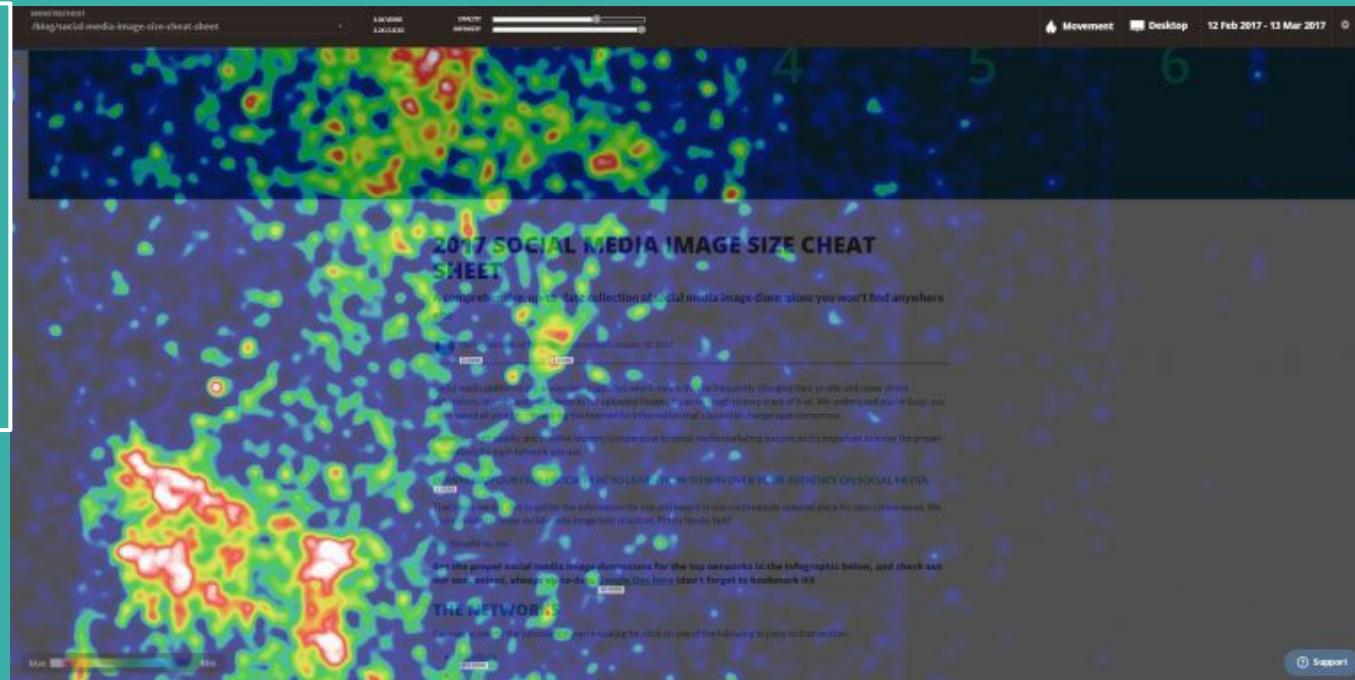
Application				
Preloading Attempt				
All preloads ▾	URL	Action	Rule set	Status
▶ Local storage	/cdns/cdnnetworks/	prefetch	www.cdnplanet.com/	Ready
▶ Session storage	/cdns/imperva/	prefetch	www.cdnplanet.com/	Not triggered
IndexedDB	/cdns/lumen/	prefetch	www.cdnplanet.com/	Not triggered
Web SQL	/guides/origin-shield/	prefetch	www.cdnplanet.com/	Not triggered
▶ Cookies	/cdns/cachefly/	prefetch	www.cdnplanet.com/	Not triggered
Private state tokens	/changelog/	prefetch	www.cdnplanet.com/	Not triggered
Interest groups	/cdns/bunnycdn/	prefetch	www.cdnplanet.com/	Not triggered
▶ Shared storage	Background services			
Cache storage	/cdns/fastly/	prefetch	www.cdnplanet.com/	Not triggered
	/cdns/cdn77/	prefetch	www.cdnplanet.com/	Not triggered
	/tools/cdnfinder/	prefetch	www.cdnplanet.com/	Not triggered
	/cdns/akamai/	prefetch	www.cdnplanet.com/	Not triggered
	/cdns/cloudflare/	prefetch	www.cdnplanet.com/	Not triggered
	/guides/websockets/	prefetch	www.cdnplanet.com/	Not triggered
	/tools/cdn-performance-check/	prefetch	www.cdnplanet.com/	Not triggered
	/cdns/edgio/	prefetch	www.cdnplanet.com/	Not triggered
Detailed information				
URL https://www.cdnplanet.com/cdns/cdnnetworks/				
Action prefetch				
Status Preloading finished and the result is ready for the next navigation.				
Frames				
▶ top				
Rule set www.cdnplanet.com/				

PERFORMANCE - FRONT



- MouseFlow
- Hotjar
- Amplitude

RAGE CLICKing
VISIBILITY



PERFORMANSE - FRONT



DEVELOPER'S
CLUB

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
CORE WEB VITALS	ZELENO, app LETI	NE KORISTIMO, app se RASPADA
TTFB Time to First Byte	DOVOLJNO BRZO DA SE VIDI DA CHROME NESTO UCITAVA I DA KRECE DA CRTA	NISTA SE NE DESAVA, KORISNIK ZATVORI APP pre nego sto stigne ista sa servera
ANALITIKA	LJUDI KORISTE APP VIDE APP/FEATURE KLIKCU NORMALNO	FRUSTRACIJA, ODUSTAJANJE, GASENJE POZARA

— DOCKER

- MULTI STAGE IMAGE
- CLEAN, MINIMAL FINAL
- ENV I SECRETS



The image shows the official website for the Slim toolkit. It features a dark background with a large, stylized blue and teal geometric logo resembling a diamond or a four-pointed star. To the right of the logo, the word "Slim" is written in a large, bold, lowercase sans-serif font, with "toolkit" in a smaller, lighter font to its right. Below the logo is a horizontal navigation bar with several buttons: "CHAT" (grey), "ON GITTER" (green), "CHAT" (grey), "ON DISCORD" (blue), "FOLLOW" (grey), "ON TWITTER" (blue), and "YOUTUBE" (red). Underneath this bar are two rows of smaller buttons: "GITPOD" (grey) and "READY-TO-CODE" (grey) in the first row; and "INSTALL" (grey), "SLIM" (blue), "SLIM" (grey), and "APP EXAMPLES" (green) in the second row. At the bottom of the page, there is a large, bold, white text message: "Optimize Your Experience with Containers. Make Your Containers Better, Smaller, More Secure and Do Less to Get There (free and open source!)."

Optimize Your Experience with Containers. Make Your
Containers Better, Smaller, More Secure and Do Less to
Get There (free and open source!)

– DOCKER

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
VELICINA IMAGE	MALI IMAGE, BRZ BUILD, DEPLOY, PULL	IMAGE OD 50 GB, obara server kada se uradi pull/deploy Obara i DEV masinu OBARA SVE + opasno
DAL GA UOPŠTE IMA?	AKO GA IMA, LAKŠE JE: - Dev setup - Deploy	AKO GA NEMA, svaki frontend dev mora da instalira Kafka, Redis, Elastic, networking za broadcast brokera
PROMENA RUNTIME-a	KONTROLISANA	POMOZ BOŽE

REFACTORING ENABLED DEVELOPMENT AKA RED

20. septembar 2020.

To je takav razvoj u kome u svakom trenutku možemo da primenimo kakav god potreban refactoring.

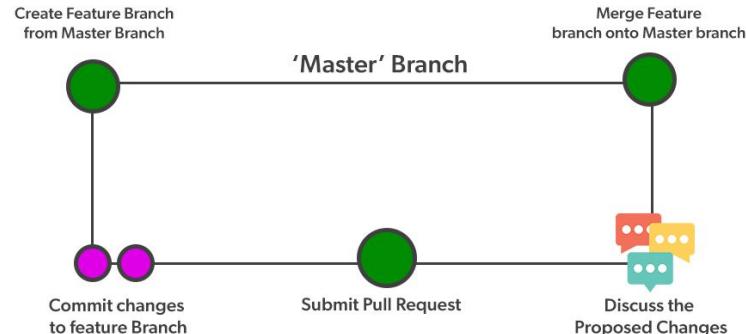
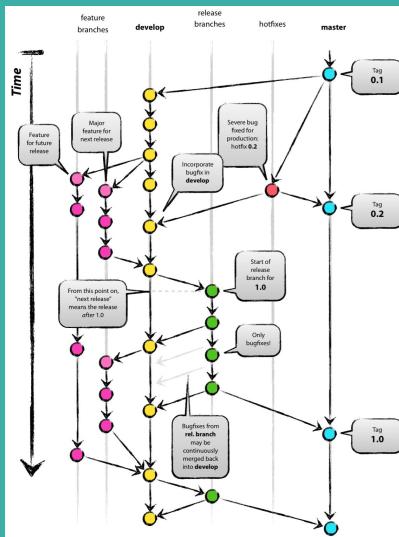
oblac.rs

<https://oblac.rs/refactoring-enabled-development-aka-red/>

TESTOVI , QA - RELEASE



- PROCES KAKO DOĆI OD 1.0 DO 1.1?
- BLOCKER
- KAKO DA TO BUDE POUZDANO?
- GIT FLOW, GITHUB FLOW



TESTOVI , QA



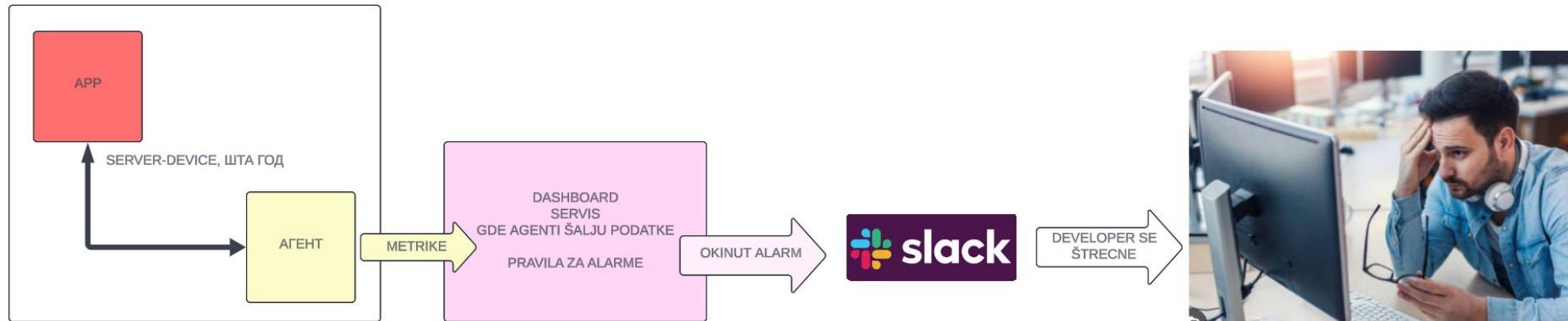
DEVELOPER'S CLUB

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
UPDATE LIBA OD KOG ZAVISI X FEATURE-A	TESTOVI UTVRDE DA LI JE PONAŠANJE ISPRAVNO	BAGOVI NA PRODUKCIJI NEUHVACENI ILI JAKO TEŽAK REGRESSION SUITE
IZMENA FEAT-A	TESTOVI UTVRDE DA LI JE PONAŠANJE ISPRAVNO	BAGOVI NA PRODUKCIJI NEUHVACENI ILI JAKO TEŽAK REGRESSION SUITE
NOVI DEVS ULAZE U CODEBASE	TESTOVI KAO LIVING SPEC PONAŠANJA (AAA)	DEBUG KODA I MENTALNA GIMNASTIKA
RELEASE CADENCE	VISOK, IZMENE KODA SE LAKO VERIFIKUJU	NIZAK, SVE MORA DA STANE

– LOGOVI I MONITORING

- NE BI TREBAO KORISNIK DA SE JAVI DA NAS OBAVESTI DA NAM NE RADI APP
- KADA NEŠTO PUKNE, TREBA DA REKONSTRUIŠEMO FOREZNIČKI CEO DOGAĐAJ

– LOGOVI I MONITORING



LOGOVI I MONITORING sentry

Issues ⓘ

Unresolved 82 For Review 2 Regressed Escalating Archived

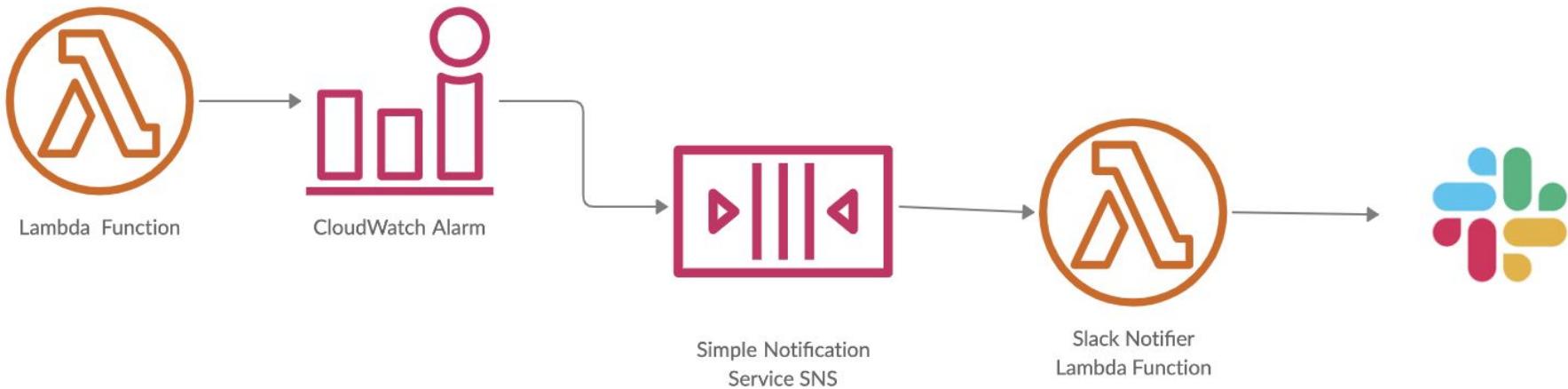
frontend-javascript All Envs 14D Custom Search is:unresolved

Archive Resolve ... Last Seen GRAPH: 24h 14d EVENTS USERS ASSIGNEE

Issue Type	Description	Created	Last Seen	Graph (Events)	Events	Users	Assignee
SyntaxError	syntaxError(utils/errors) unexpected token: identifier	New FRONTEND-JAVASCRIPT-3H3	36s ago 18hr old	11	95	95	
Error	n(utils/errors) unhandled error	New FRONTEND-JAVASCRIPT-3W	40s ago 18hr old	6	55	55	
N+1 API Call	/plususone GET https://application-monitoring-flask-dot-sales-engineering-sf.appspot.com/pr...	Ongoing FRONTEND-JAVASCRIPT-2V	7min ago 6mo old	110	28k	14k	YD
SyntaxError	https://application-monitoring-react-dot-sales-engineering-sf.app... The string did not match the expected pattern.	New FRONTEND-JAVASCRIPT-3H2	7min ago 18hr old	303	3.2k	1.1k	
SyntaxError	json([native code]) The string did not match the expected pattern.	New FRONTEND-JAVASCRIPT-3H1	7min ago 18hr old	235	2.5k	1k	
SyntaxError	https://application-monitoring-react-dot-sales-engineering-sf.app... JSON.parse: unexpected keyword at line 1 column 1 of the JSON data	New FRONTEND-JAVASCRIPT-3GZ	7min ago 18hr old	90	1.1k	1k	
TypeError	https://application-monitoring-react-dot-sales-engineering-sf.appspot.com/ NetworkError when attempting to fetch resource.	New FRONTEND-JAVASCRIPT-3HO	7min ago 18hr old	83	949	924	
ReferenceError	referenceError(utils/errors) undefinedVariable is not defined			10	100	100	

- NE BI TREBAO KORISNIK DA SE JAVI DA NAS OBAVESTI DA NAM NE RADI APP

– LOGOVI I MONITORING aws



– LOGOVI I MONITORING - INCIDENTI



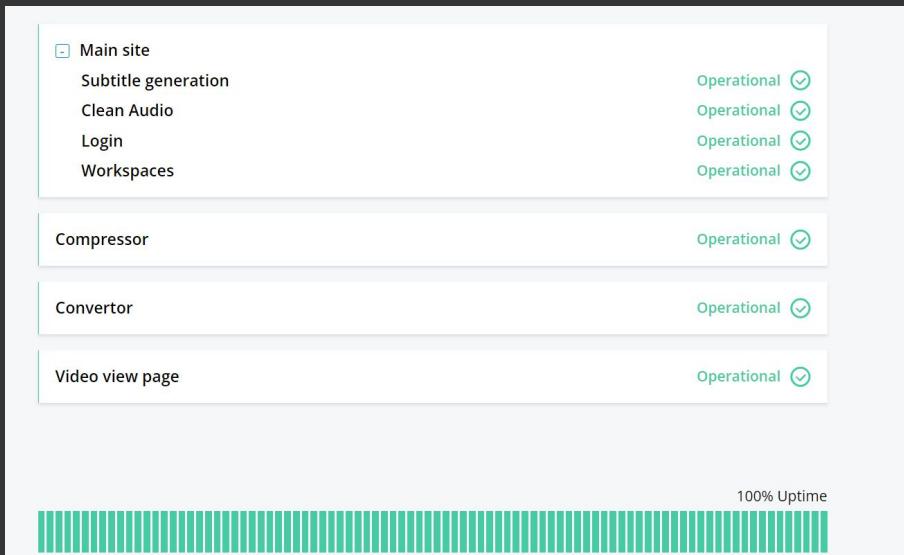
DEVELOPER'S CLUB

BDD#8 Incident Response



- BDD #8
- <https://bit.ly/bdd-8>

– LOGOVI I MONITORING - INCIDENTI



- BDD #8
- <https://bit.ly/bdd-8>

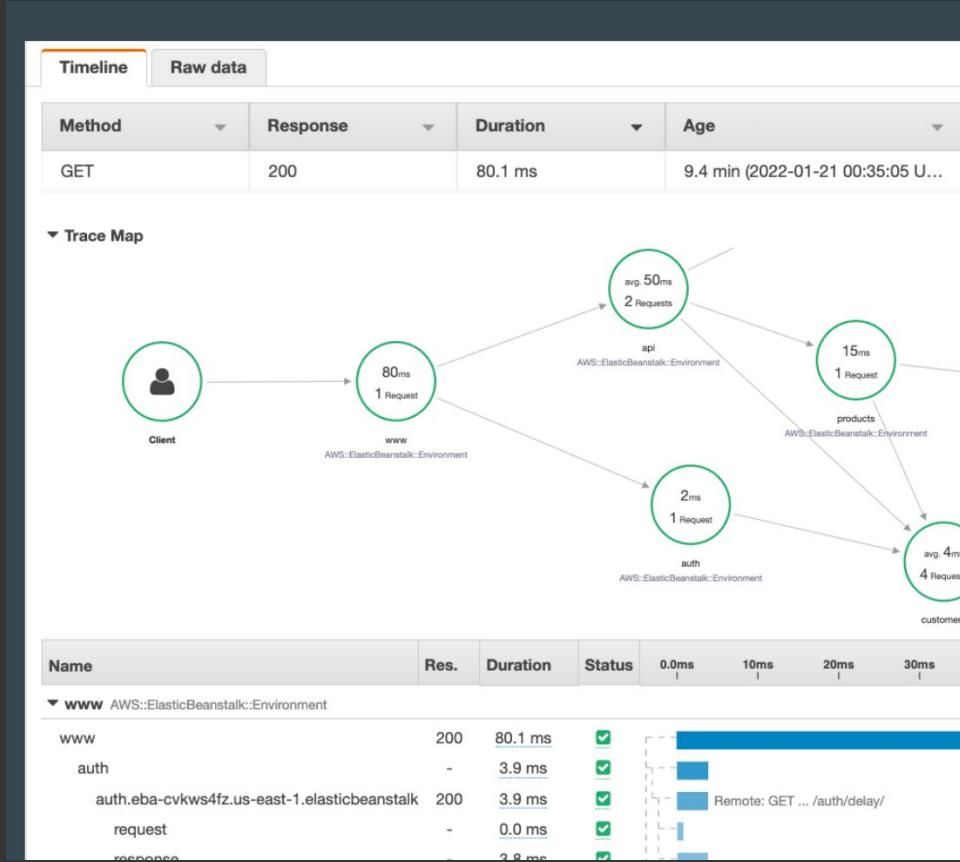
status.veed.io
Status page
+ Incident updates

– LOGOVI I MONITORING - INCIDENTI

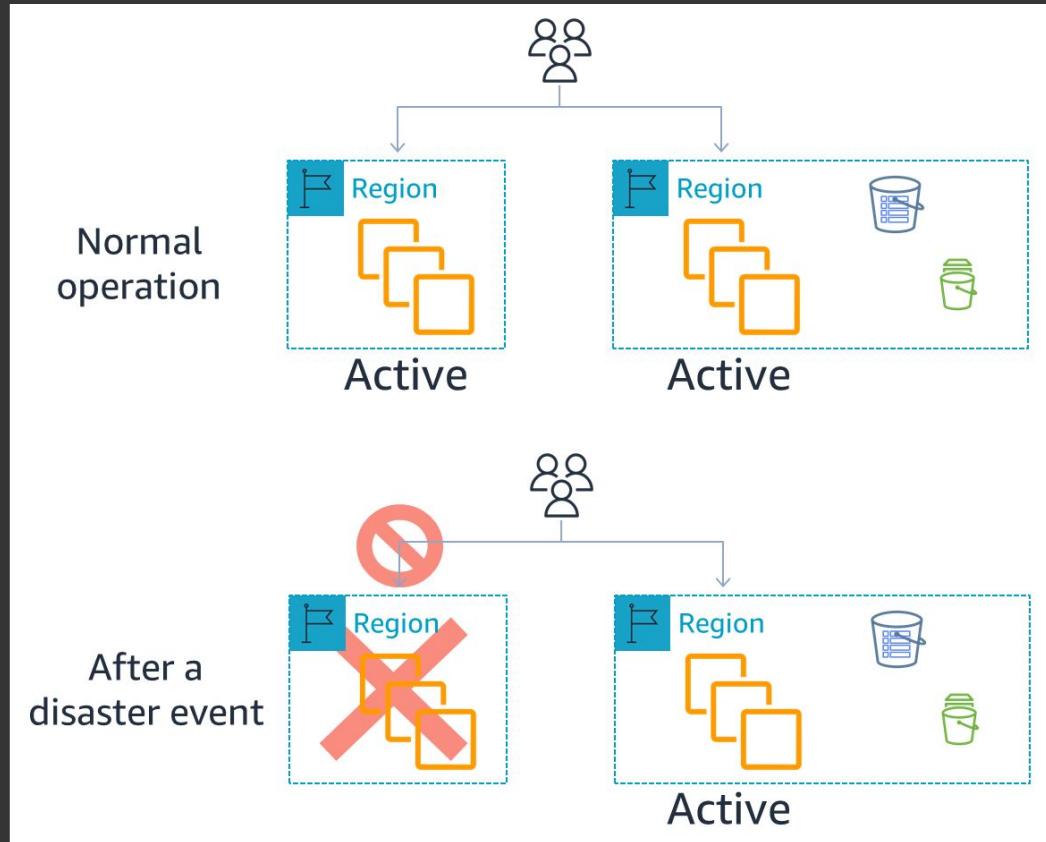
Service	RSS	<	Today	25 Sep	24 Sep	23 Sep	22 Sep	21 Sep	20 Sep
Amazon API Gateway (Calgary)	RSS		🕒	🕒	🕒	🕒	🕒	🕒	🕒
Amazon API Gateway (Canada-Central)	RSS		🕒	🕒	🕒	🕒	🕒	🕒	🕒
Amazon API Gateway (N. California)	RSS		🕒	🕒	🕒	🕒	🕒	🕒	🕒
Amazon API Gateway	RSS		🕒	🕒	🕒	🕒	🕒	🕒	🕒

- AWS status page
Health svih AWS servisa
<https://health.aws.amazon.com/health/status>

– LOGOVI I MONITORING - INCIDENTI



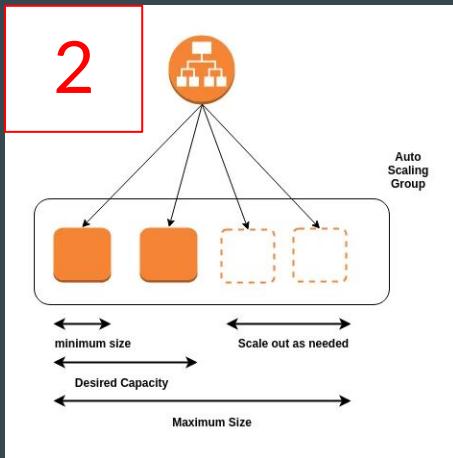
– LOGOVI I MONITORING - INCIDENTI



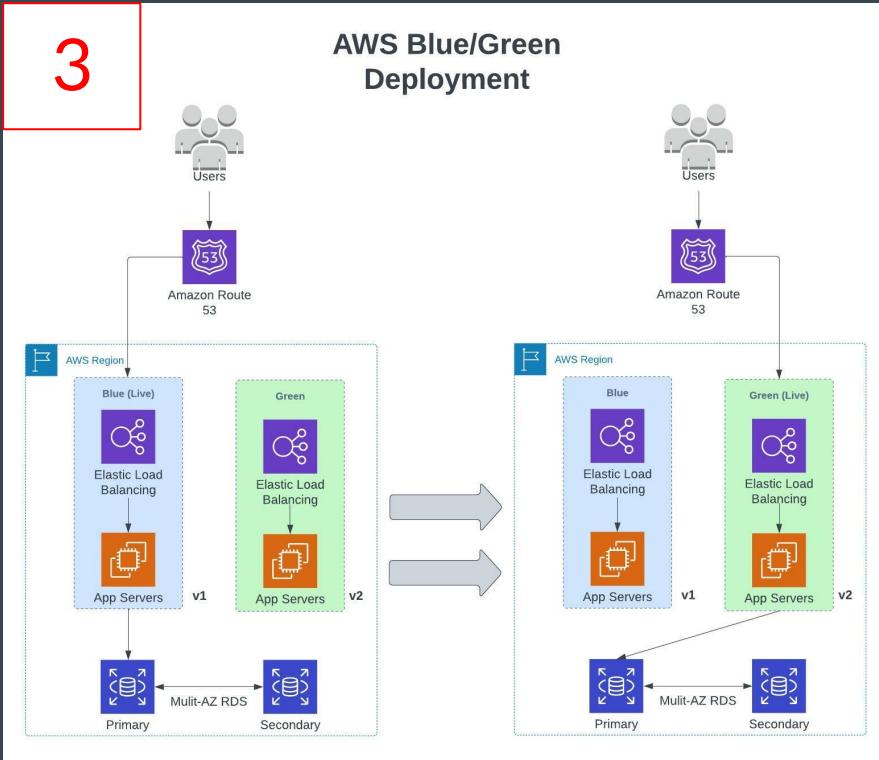
Recovery



Recovery



Recovery



– LOGOVI I MONITORING - INCIDENTI

SCENARIO	DOBRO ✓	OPASNO! LOŠE! XXX
IMA PROBLEM U SISTEMU	DETektujemo ga, rano, pre korisnika, rešimo pošto imamo sve podatke za istragu	Korisnici nam javljaču i besne, nemamo info pokušavamo da reprodukujuemo gatanjem u pasulj
INCIDENT, PAO CEO SISTEM	Znamo kako da se oporavimo, pišemo lekcije za ubuduće, post mortem	????????? GAŠENJE POŽARA?
PROMENA RUNTIME-a	KONTROLISANA	POMOZ BOŽE

3rd PARTY SERVISI (\$\$\$)



DEVELOPER'S
CLUB

- NE SMEMO ČEKATI LANSIRANJE APP DA PROVERIMO DA SVE RADI SA 3RD PARTY PROVAJDERIMA (I PRAVNO DAL JE SVE OK)
- BANKROT JE REALAN, RATE LIMITOVATI
- PROBIJANJE RATE LIMITA PROVIDERA; PAD NAŠEG SISTEMA
- PADNE PROVAJDER, PADAMO LI I MI?

SPECIFIČNOSTI RUNTIME-A



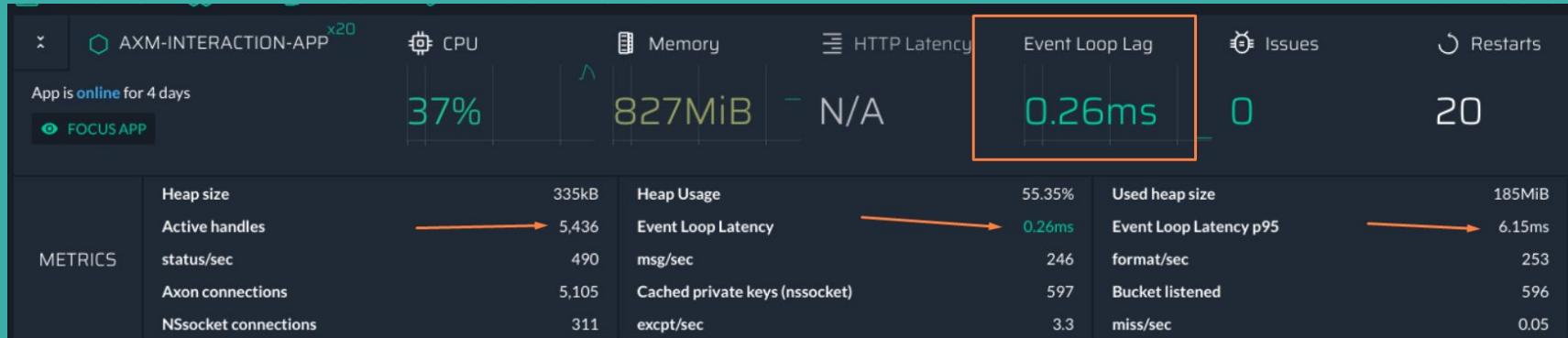
DEVELOPER'S
CLUB

- NODE
- JAVA
- .NET
- PYTHON

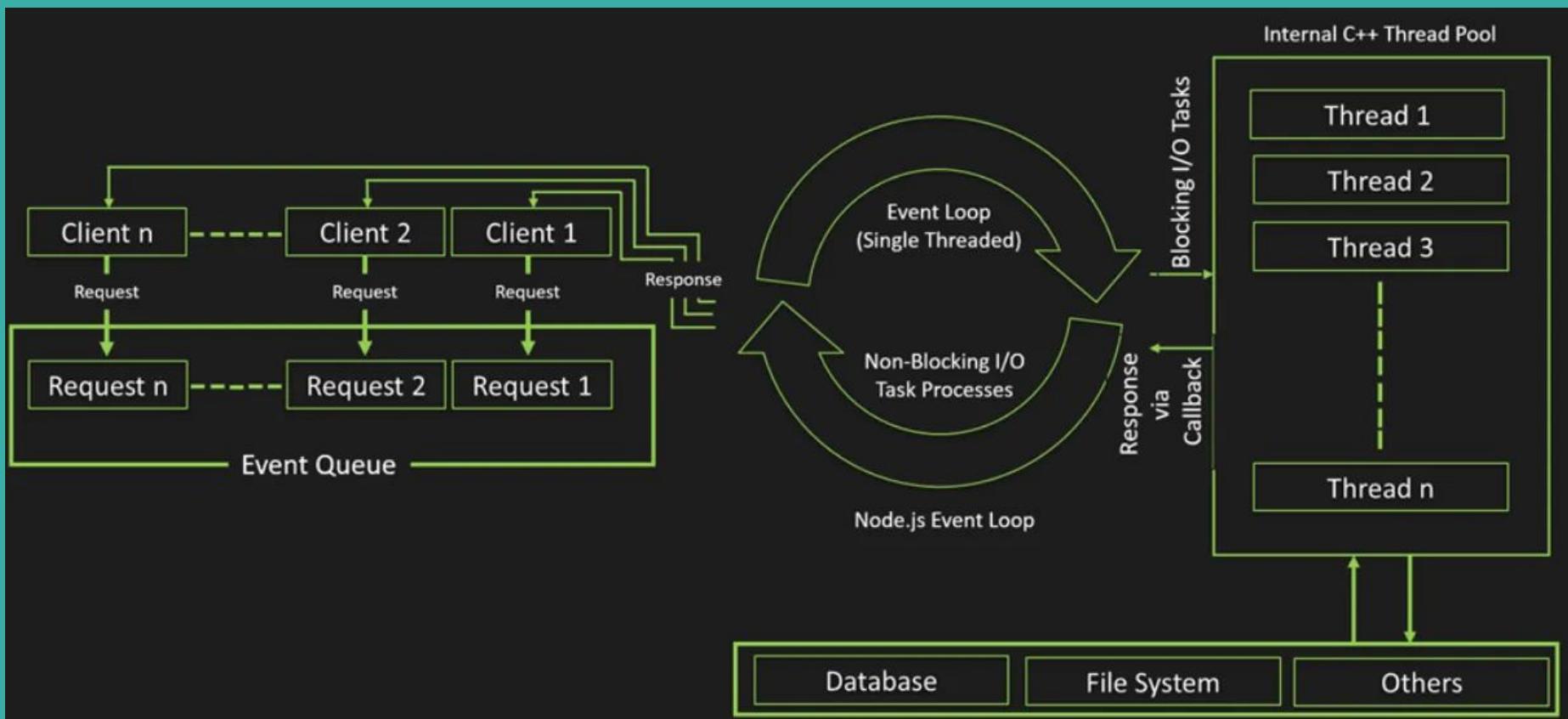
SPECIFIČNOSTI RUNTIME-a => NodeJS



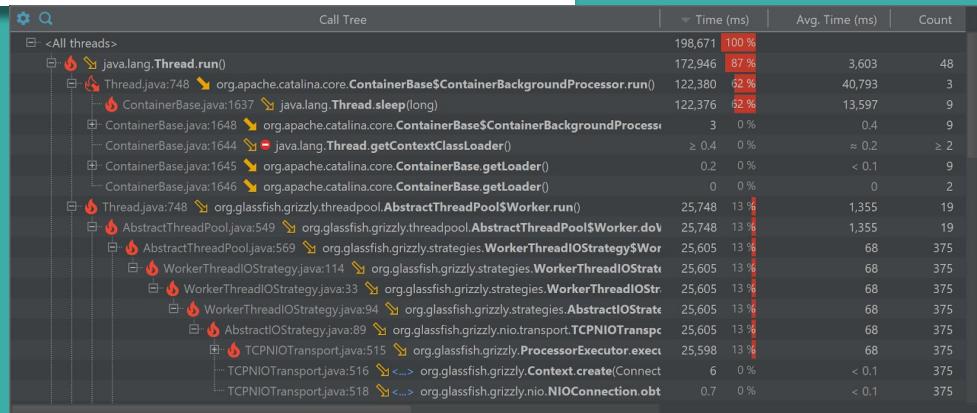
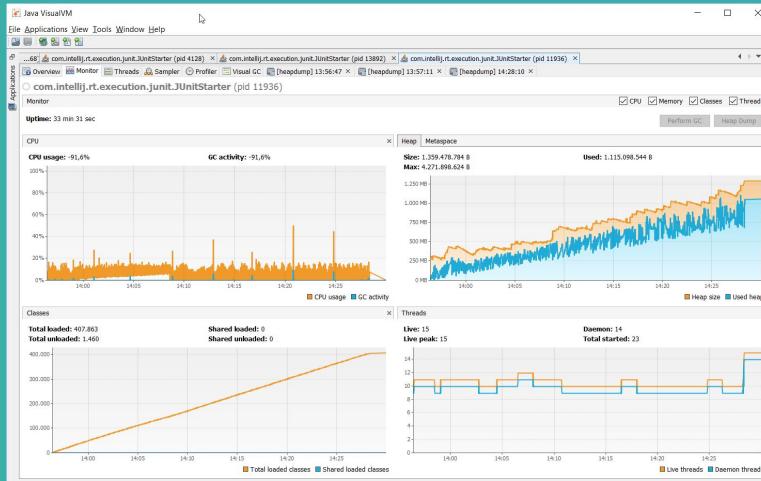
- Pratiti Event Loop lag ! !
- LOCK-IN DEPENDENCIES ! ! ! ! !
- process.on('uncaughtException')
- process.on('unhandledRejection')
- node --trace-gc app.js
const v8 = require('v8');



SPECIFIČNOSTI RUNTIME-a => NodeJS



SPECIFIČNOSTI RUNTIME-a => Java



Java's garbage collector can impact performance if not tuned correctly.

To optimize garbage collection, you can adjust JVM options, such as `-Xmx`, `-Xms`, and `-XX:MaxGCPauseMillis`,

- **VISUAL VM (memory leak primer)** ! ! !

- **LOCK-IN DEPENDENCIES (gradle, maven)**

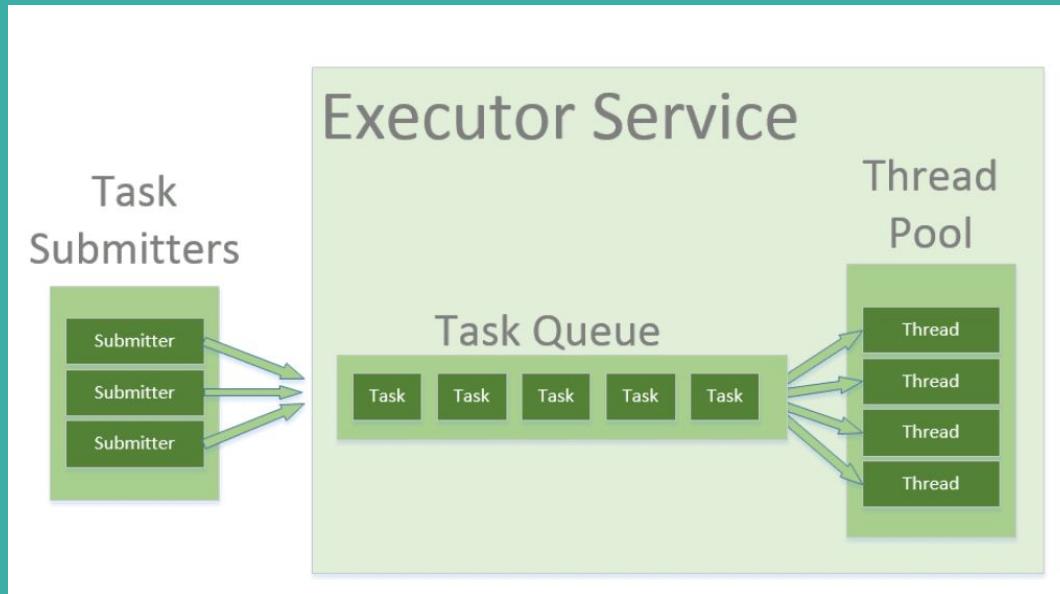
```
<exclusions>  
  <exclusion>
```

https://docs.gradle.org/current/userguide/dependency_resolution.html

SPECIFIČNOSTI RUNTIME-a => Java



- **Thread Pool Exhaustion** ! ! !
corePoolSize, maxPoolSize, queueCapacity



- **LOCK-IN DEPENDENCIES** !
- **async await** ! ! !

4. Overusing `Task.Run` for I/O-Bound Operations

Bad Example:

```
csharp

public async Task<string> ReadFileAsync(string path)
{
    return await Task.Run(() => File.ReadAllText(path));
}
```

```
public void ProcessData()
{
    var data = GetDataAsync().Result;
    // Process data synchronously
}

public async Task<string> GetDataAsync()
{
    await Task.Delay(1000); // Simulate asynchronous operation
    return "Data";
}
```

```
public async Task ProcessDataAsync()
{
    var data = await GetDataAsync();
    // Process data asynchronously
}
```

SPECIFIČNOSTI RUNTIME-a => .NET



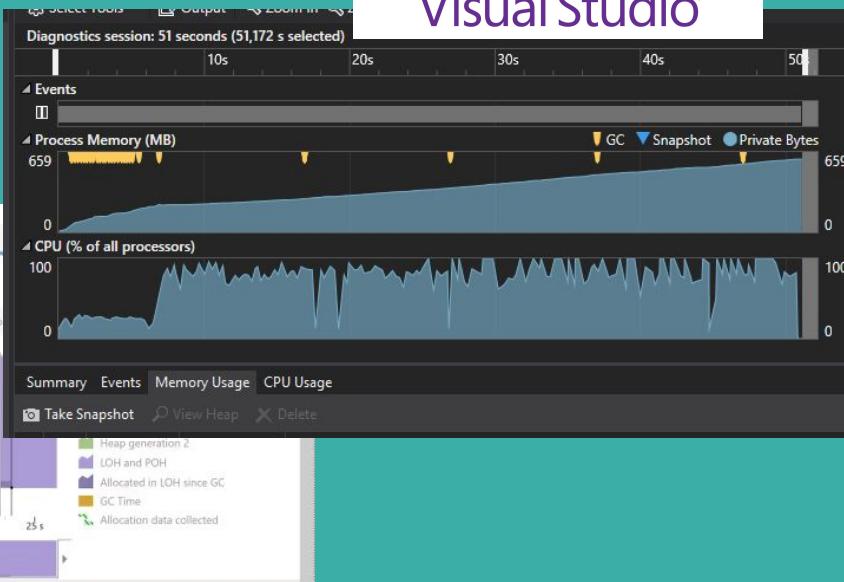
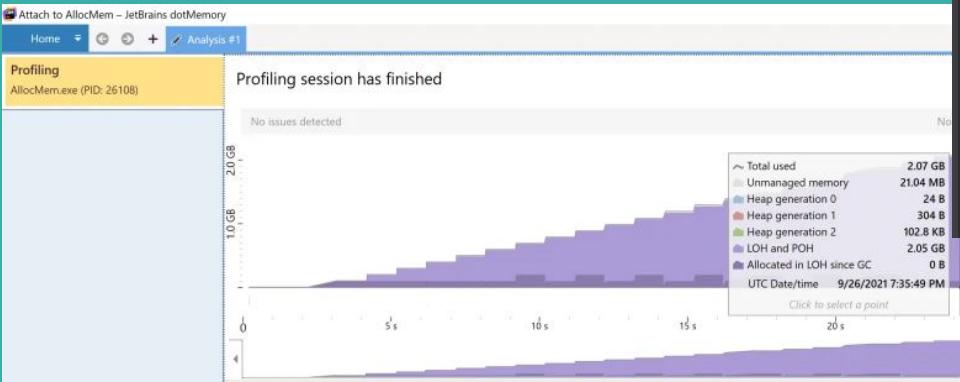
part of dotUltimate
dotMemory

The .NET Memory Profiler

[Download](#)

[Check out our performance profiler](#)

Free 30-day trial

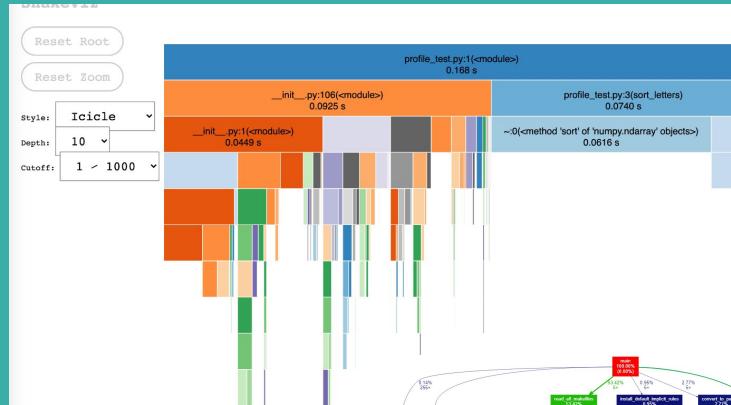
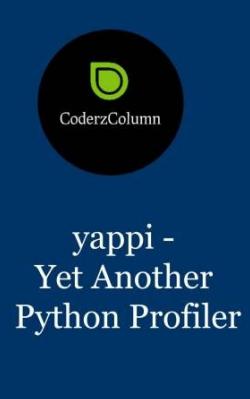
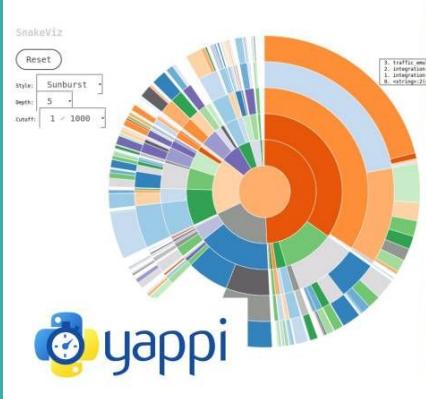


Visual Studio

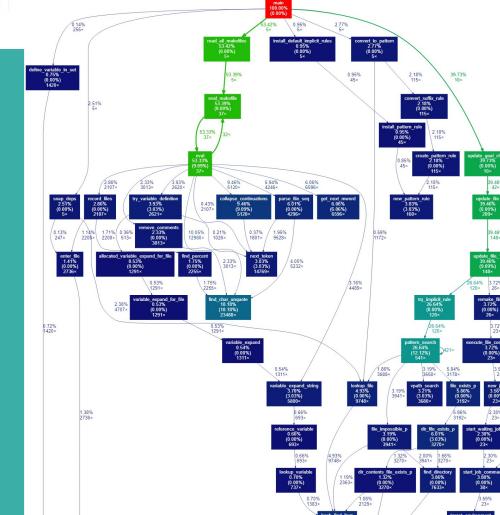
SPECIFIČNOSTI RUNTIME-a => PYTHON



- **LOCK-IN DEPENDENCIES** ! ! !
- **venv, virtualenv** !
- **GIL** ! ! ! **asyncio**
- **WSGI**



cProfiler +
SnakeViz
+ py-spy
+ Yappi



SPECIFIČNOSTI RUNTIME-a => PYTHON



```
gunicorn myapp:wsgi --workers=4 --worker-class=sync --threads=1  
# Each worker is a separate process with its own GIL
```

```
uwsgi --module myapp:wsgi --processes=4 --threads=1  
# Each process is a separate process with its own GIL
```

SPECIFIČNOSTI RUNTIME-a => PYTHON



asyncio event loop je **SINGLE THREADED**, welcome back V8

CPU intensive -> worker pool

```
import asyncio
from concurrent.futures import ProcessPoolExecutor

def cpu_bound_function():
    total = sum(i * i for i in range(10**8))
    return total

async def main():
    loop = asyncio.get_running_loop()
    with ProcessPoolExecutor() as executor:
        result = await loop.run_in_executor(executor, cpu_bound_function)
        print(result)

asyncio.run(main())
```



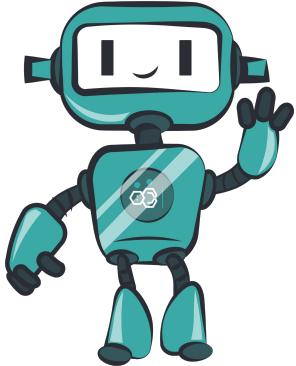
SPECIFIČNOSTI RUNTIME-A



- MORAMO DUBOKO RAZUMETI RUNTIME U KOM NAM SE IZVRŠAVA APLIKATIVNI KOD
- DEPENDENCIES
- MEMORY/PERFORMANCE PROFILING
- CORRECT USE OF WORKER THREADS/PROCESSES



Hvala!
Postani Član:



developersclub.rs