

Production detectives

solving the mystery before the crash





Sébastien Ferrer

Software engineer @  OVHcloud

 sebferrer

 Sébastien Ferrer

Preamble



git commit
↓
git push →



Preamble

SI&A

System Integration & Automation

2 objectives

Mission 1:

“You must be able to do RUN in 3 months’ time”

Mission 2:

“You’re to join the night/weekend on-call rotation 2 months later ”



RUN

BUILD

Development

New features

Infrastructure

RUN

maintenance
operation



systems in production

Incidents

" Daytime on-call "



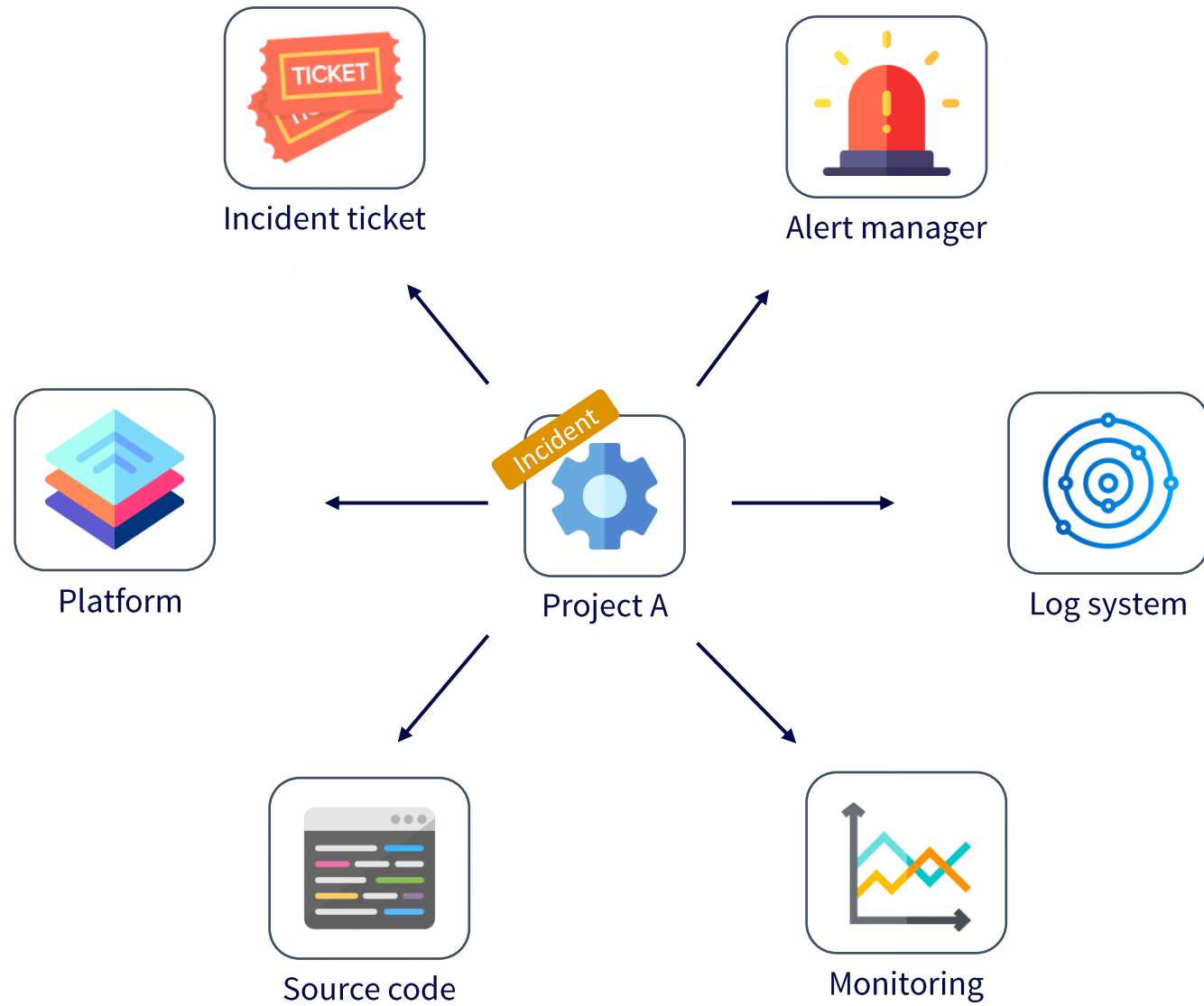
Rotation

My first incidents

 Incident							
Project A	Project B	Project C	Project D	Project E	Project F	Project G	Project H
					 Incident		
Project I	Project J	Project K	Project L	Project M	Project N	Project O	Project P
	 Incident						 Incident
Project Q	Project R	Project S	Project T	Project U	Project V	Project W	Project X
					 Incident		
Project Y	Project Z	Project AA	Project AZ	Project AE	Project AR	Project AT	Project AY



How?



Morning check

Project A

5xx errors 0	4xx errors 2
Success 58	Failures 2

Project B

5xx errors 1	4xx errors 0
Success 12	Failures 1

Project C

5xx errors 0	4xx errors 12
Success 85	Failures 12

Project D

5xx errors 0	4xx errors 0
Success 24	Failures 0

Project E

5xx errors 0	4xx errors 0
Success 42	Failures 0

Project F

5xx errors 15	4xx errors 1
Success 67	Failures 16

Monitor tickets

Affected Item	Log	Date	Priority	Time to resolve
Project A	[project-a] 4xx error	2024-02-02 05:38:01	P5	
Project B	[...] 4xx error	2024-02-02 05:39:04	P4	
Project C	[...] 4xx error	2024-02-02 05:39:06	P5	
Project A	[...] 5xx error	2024-02-02 06:40:12	P5	
Project M	[...] 5xx error	2024-02-02 10:36:02	P5	
Project B	[...] 4xx error	2024-02-02 10:42:08	P3	
Project L	[...] 4xx error	2024-02-02 11:38:06	P5	
Project W	[...] 4xx error	2024-02-02 11:40:45	P5	

Incidents

TTA

Time To Acknowledge



Inform that you are on it

TRS

Time to Restore Service



Contextualization

Troubleshooting

Resolution

Priority

P1

P2

P3

P4

P5

Monitor tickets

Affected Item	Log	Date	Priority	Time to resolve
Project A	[project-a] 4xx error	2024-02-02 05:38:01	P5	
Project B	[...] 4xx error	2024-02-02 05:39:04	P4	
Project C	[...] 4xx error	2024-02-02 05:39:06	P5	
Project A	[...] 5xx error	2024-02-02 06:40:12	P5	
Project M	[...] 5xx error	2024-02-02 10:36:02	P5	
Project B	[...] 4xx error	2024-02-02 10:42:08	P3	
Project L	[...] 4xx error	2024-02-02 11:38:06	P5	
Project W	[...] 4xx error	2024-02-02 11:40:45	P5	
Project S	[...] 5xx error	2024-02-02 12:08:29	P2	

RUN

Be the point of contact
of your team

Troubleshooting

Root Cause Analysis (RCA)



Troubleshooting

history-based method



Fast
Someone has surely already met this error
Reducing learning curve



Limited to known issues
May be misleading
Less learning

Drill down



Thorough
Helps identify the root cause
Better understanding of the project

Bottom-up



Time-consuming
Risk of following a wrong path

Troubleshooting

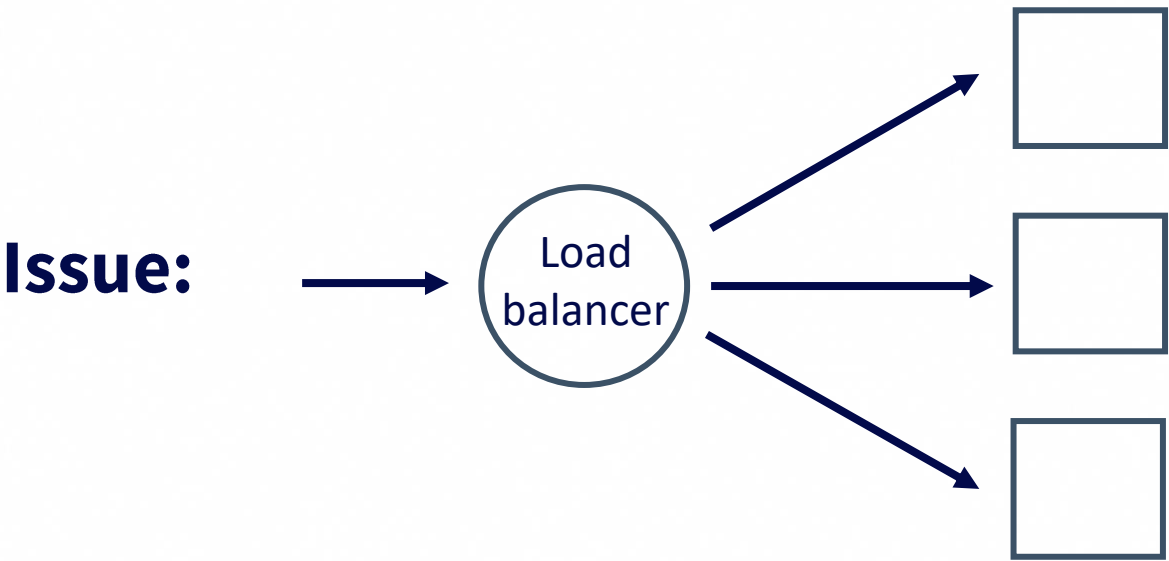
Ask yourself the right question

What problem am I trying to solve?

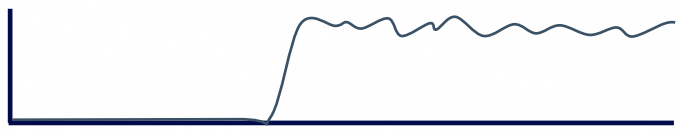


Troubleshooting – Case 1

Problem: Service very slow



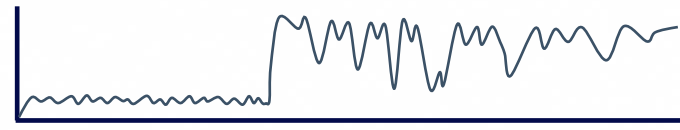
HTTP 5xx errors



HTTP 200



Latency

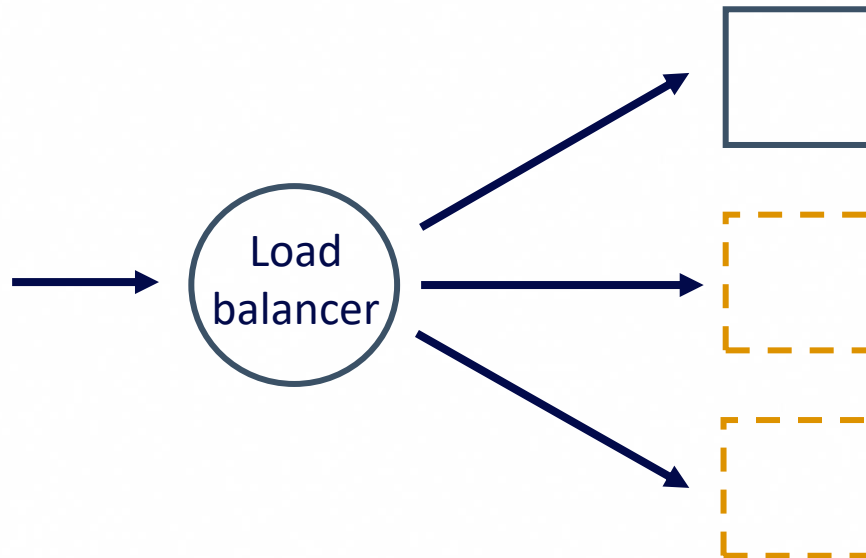


Message	Instance
Error 500: ...	instance-b
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-c

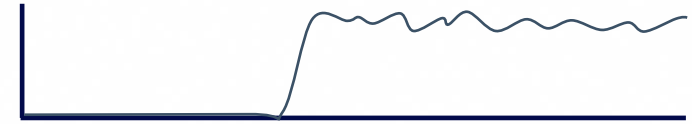
Troubleshooting – Case 1

Problem: Service very slow

Issue:



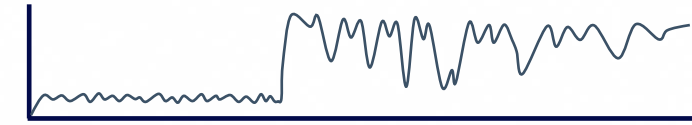
HTTP 5xx errors



HTTP 200



Latency



Message

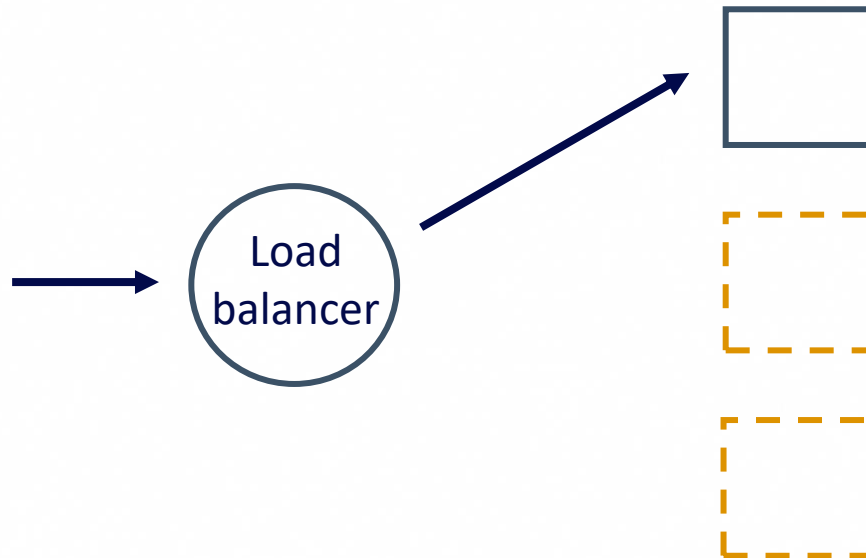
Instance

Error 500: ...	instance-b
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-c

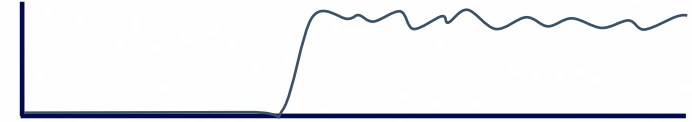
Troubleshooting – Case 1

Problem: Service very slow

Issue:



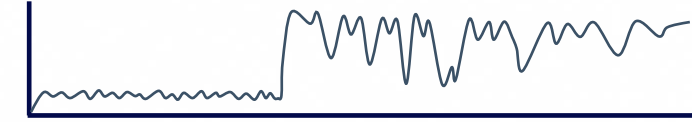
HTTP 5xx errors



HTTP 200



Latency



Message

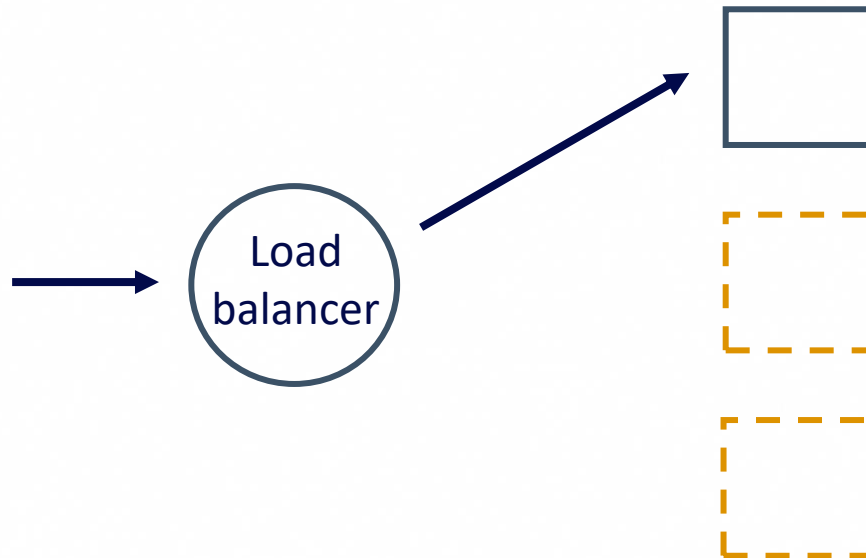
Instance

Error 500: ...	instance-b
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-b
Error 500: ...	instance-c
Error 500: ...	instance-c

Troubleshooting – Case 1

Problem: Service very slow

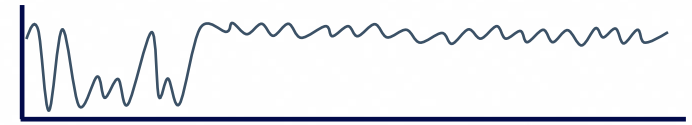
Issue:



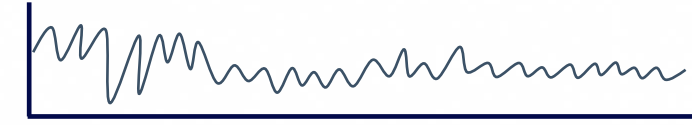
HTTP 5xx errors



HTTP 200



Latency



Message

Instance

Troubleshooting – Case 2

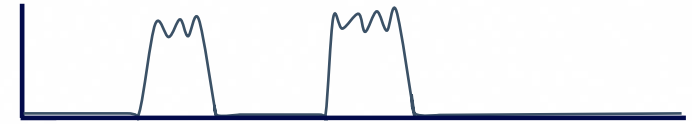
Problem: Too many 500 errors

Issue: Database connection failed

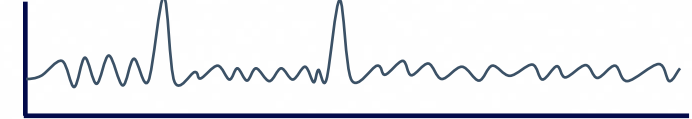
Temp. Fix: ↗ Rate limiting

Fix: Connexions optimization
Cache
Scaling

HTTP 5xx errors



Database connexions



Accounts

Nb connexions

User 42	140
User 53	18
User 65	12

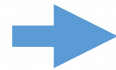
Message

Database connection failed
Database connection failed
Database connection failed
Database connection failed
Database connection failed
Database connection failed

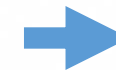
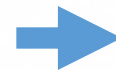
Troubleshooting – Case 3



"operation failed"



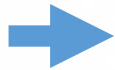
When ?



Acknowledgement

Watch alerts

Monitoring /
Assessing the extent
of the damage



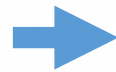
creating user: operation failed
creating user: operation failed
creating user: operation failed

Find logs

Troubleshooting – Case 3



Find logs



message

creating user: operation failed
creating user: operation failed
creating user: operation failed

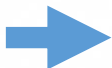
operation

CreateUser
CreateUser
CreateUser

Troubleshooting – Case 3



Find logs



message

creating user: success
creating user: operation failed
creating user: success
creating user: success
creating user: operation failed
creating user: operation failed

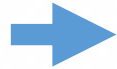
operation

CreateUser
CreateUser
CreateUser
CreateUser
CreateUser
CreateUser

Troubleshooting – Case 3



Find logs



message	pod_id
creating user: success	app-54fb9-n4vkg
creating user: operation failed	app-54fb9-n4vkg
creating user: success	app-54fb9-n4vkg
creating user: success	app-54fb9-n4vkg
creating user: operation failed	app-54fb9-n4vkg
creating user: operation failed	app-54fb9-n4vkg
	app-54fb9-n4vkg

Troubleshooting

Scope the issue

Who?

What?

When?

Reproduce

Gather information

Document

Troubleshooting – Five whys

Problem: Users can no longer connect to the application

Why? The authentication system returns an error 500

Why? The authentication service is unable to query the user database

Why? The authn service cannot establish a connection to the database

Why? The database server is unreachable

Why? A recent firewall configuration change



*Develop your applications as if
you were going to maintain them.*

Because you are.

~ a cool colleague

Logs

To avoid



Lack of
data



Noise

Log level

INFO

WARN

ERROR

DEBUG

FATAL

TRACE

Write relevant logs

Request ID

URL Path

HTTP Method

HTTP Error Code

User IDs

Stack trace

App Version

Operation

Logs

```
var user User
err := json.NewDecoder(r.Body).Decode(&user)
if err != nil {
    logrus.Errorf("decoding json: %v", err)
    http.Error(w, "invalid data", http.StatusBadRequest)
    return
}

// ...

logrus.Infof("user %s (id: %d) successfully processed", user.Username, user.ID)
```

Logs

```
logrus.Infof("user %s (id: %d) successfully processed", user.Username, user.ID)
```

INFO[0000] user Alice (id: 42) successfully processed

INFO[0000] user Bob (id: 44) successfully processed

Logs

```
logrus.Infof("user %s (id: %d) successfully processed", user.Username, user.ID)
```

```
logrus.WithFields(logrus.Fields{  
    "user_id": user.ID,  
    "username": user.Username,  
    "operation": "userProcessing",  
}).Info("user processing success")
```

Logs

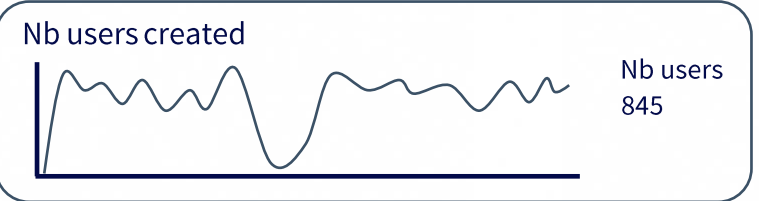
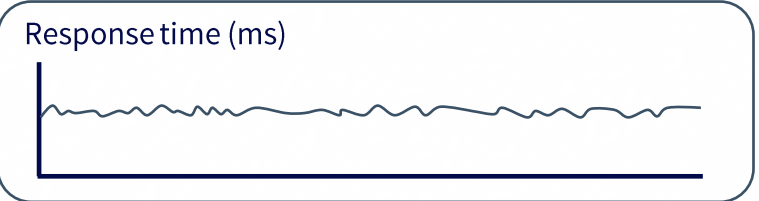
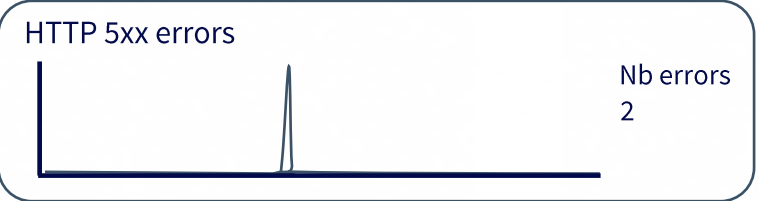
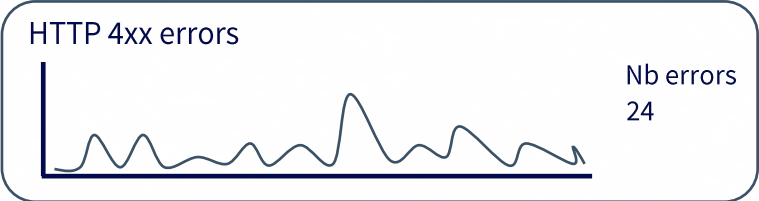
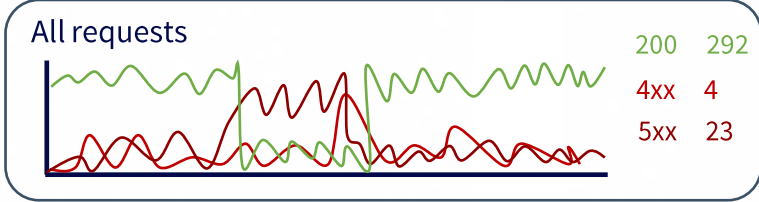
```
logrus.WithFields(logrus.Fields{  
    "user_id": user.ID,  
    "username": user.Username,  
    "operation": "userProcessing",  
}).Info("user processing success")
```

user processing success
user_id: 42
username: Alice
operation: userProcessing

user processing success
user_id: 44
username: Bob
operation: userProcessing

message: "user processing success" AND user_id:42

Monitoring / dashboards



Errors

Date	User ID	Error
2024-02-02 08:34:12.548951 +02:00	158686	creating user: POST /api/users: 403..
2024-02-02 08:34:01.548951 +02:00	694245	creating user: POST /api/users: 500..
2024-02-02 08:33:36.548951 +02:00	524215	creating user: POST /api/users: 403..
2024-02-02 08:33:24.548951 +02:00	218556	creating user: POST /api/users: 403..

Monitoring / dashboards

Define the right color

-  Everything's OK
-  Be careful
-  It's broken

The rest in white

Communicate information quickly

Clear information

Chose the right type of charts

Organize the information

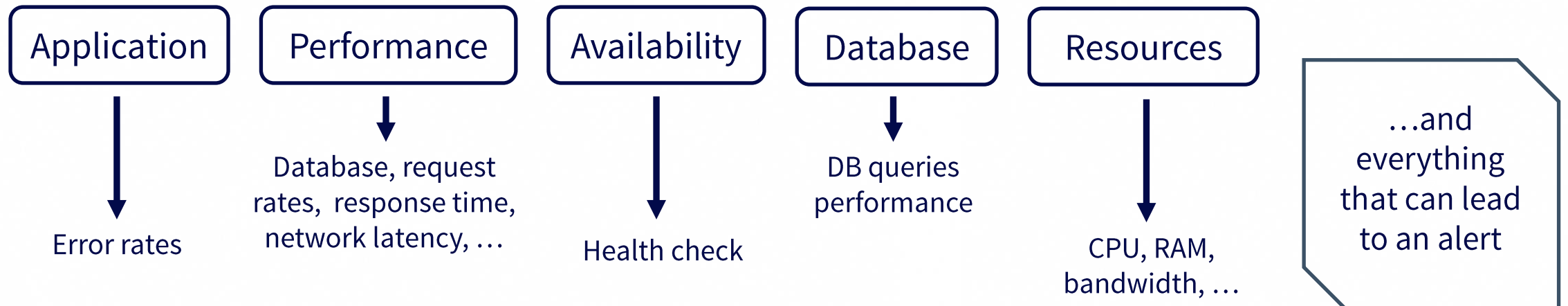
Make it customizable

Monitoring / dashboards

Why?



What?



Alerts



Logs

Metrics

Application-Level Errors

Alerts



Logs

Metrics

Application-Level Errors

Alerts



Logs

Metrics

Application-Level Errors

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts

```
groups:  
- name: api_alerts  
  rules:  
  - alert: HighHTTP500Rate  
    expr: rate(http_requests_total{status="500"}[5m]) > 0.05  
    for: 2m  
    labels:  
      severity: critical  
    annotations:  
      summary: "High rate of HTTP 500 errors"  
      description: "More than 5% of requests are failing with status 500 over the last 5 minutes."
```

Prometheus alerting rules

Alertmanager

Alerts



Logs

Metrics

Application-Level Errors

Errors handling: error wrapping

```
var ErrUserNotFound = errors.New("user not found")
```

```
func getUser(id int) (User, error) {  
    user, err := db.FindUserByID(id)  
    if err != nil {  
        if errors.Is(err, db.ErrNotFound) {  
            return User{}, ErrUserNotFound  
        }  
        return User{}, err  
    }  
    return user, nil  
}
```

Errors handling: error wrapping

```
var ErrUserNotFound = errors.New("user not found")

func getUser(id int) (User, error) {
    user, err := db.FindUserByID(id)
    if err != nil {
        if errors.Is(err, db.ErrNotFound) {
            return User{}, ErrUserNotFound
        }
        return User{}, err
    }
    return user, nil
}
```

Errors handling: error wrapping

```
var ErrUserNotFound = errors.New("user not found")

func getUser(id int) (User, error) {
    user, err := db.FindUserByID(id)
    if err != nil {
        if errors.Is(err, db.ErrNotFound) {
            return User{}, ErrUserNotFound
        }
        return User{}, err
    }
    return user, nil
}
```

`return User{}, fmt.Errorf("fetching user: %w", err)`

"record not found"

"fetching user: record not found"

Continuous improvement / LEAN

Use an archetype

Unified codebase

Unified starter

Microservice gateway communications

Packages / dependencies

Artifact repository / manager

CI/CD setup

Use common packages

logger

alerter

profiling

http requests

Error handling

Health-check

cmd

metrics

crypto

reflect

Continuous improvement / LEAN

Use a linter!

Gabriel Augendre

Sébastien Ferrer



<https://www.youtube.com/watch?v=CYxyezY4k4Q>

Sources

RUN / Troubleshooting

<https://www.ibm.com/think/topics/it-operations>

<https://www.maddyness.com/2019/12/23/startupers-build-et-run/>

<https://www.oracle.com/fr/security/troubleshooting-depannage-resolution-problemes/>

<https://www.ciscopress.com/articles/article.asp?p=102211&seqNum=3>

https://en.wikipedia.org/wiki/Five_whys

<https://medium.com/@dzanna.molly/devops-troubleshooting-strategy-2b5b38a5f3b7>

Monitoring efficace : comment sauver votre production rapidement – Antoine Beyet

<https://www.youtube.com/watch?v=6pRBWM-J-c8>

Special thanks to the
OVHcloud SI&A team

Character drawings
by goupil.poil (instagram)

Icons from Flaticon

Freepik
Smashicons
WinWwin Winaldi
Good Ware

Vectors Market
Prosymbols Premium
dmitri13
Amethyst prime

Sources

Code

Good practices

- <https://preslav.me/2023/04/14/golang-error-handling-is-a-form-of-storytelling/>
- <https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/logging-levels/>
- <https://aws.amazon.com/what-is/log-files/>
- <https://play.grafana.org/>
- https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/
- <https://betterstack.com/community/guides/logging/logging-best-practices/>
- <https://www.dataset.com/blog/the-10-commandments-of-logging/>
- <https://www.digitalocean.com/community/tutorials/handling-errors-in-go>
- https://www.jetbrains.com/guide/go/tutorials/handle_errors_in_go/best_practices/
- <https://www.ibm.com/think/topics/application-monitoring-best-practices>
- <https://www.keyprod.com/en/knowledge/production-monitoring>

Icons from Flaticon

Special thanks to the
OVHcloud SI&A team

Character drawings
by goupil.poil (instagram)

Freepik
Smashicons
WinWwin Winaldi
Good Ware

Vectors Market
Prosymbols Premium
dmitri13
Amethyst prime



Kimi, my cat



sebferrer



Sébastien Ferrer



Feedback ♥



Blog here!

blog.kimi.ovh

