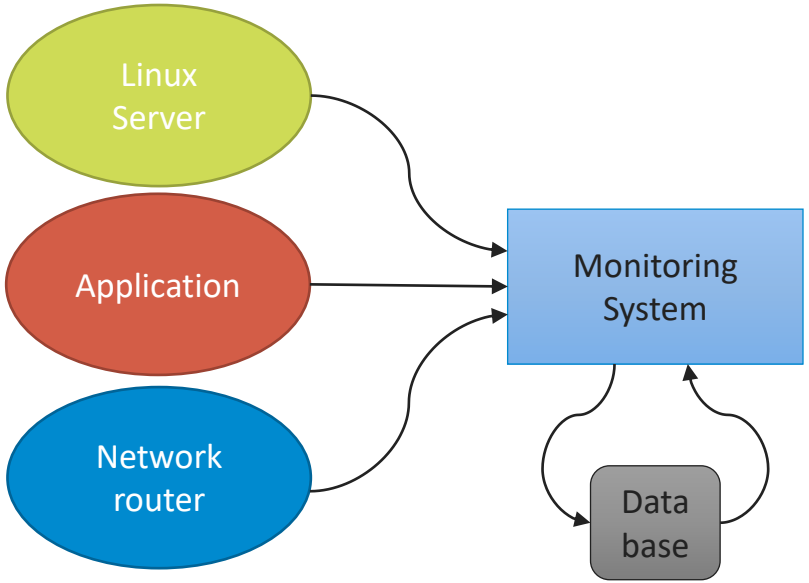
Monitoring

Monitoring theory

What monitoring is

Monitoring is a continuous process of collecting information (metrics) about monitored system. Collected information is being stored and analyzed for triggering alarms, building graphs and event correlation.



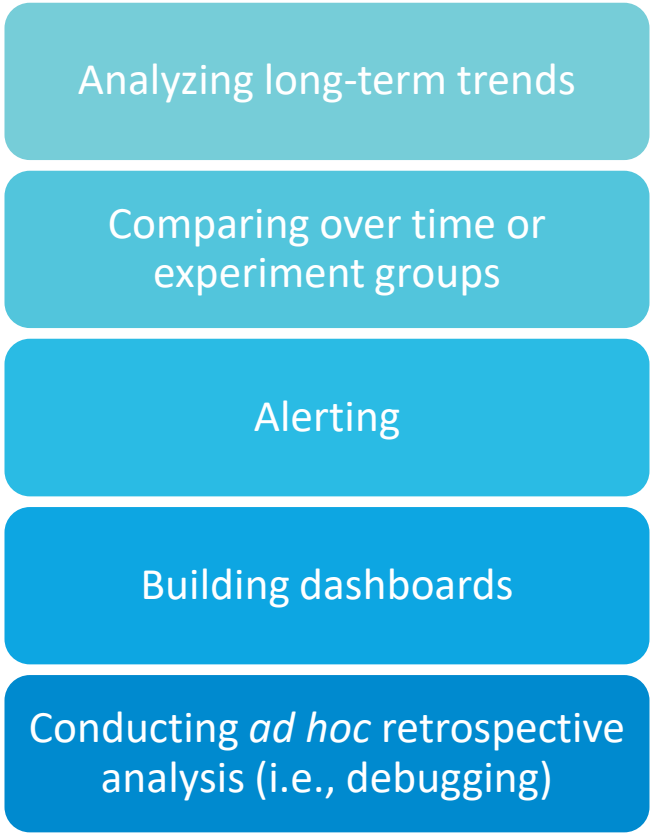
Underlying terms

Observable system – System from where we're collecting metrics. It can be *application*, *service*, *server* and etc

Metric – any collected information

Alert – A notification intended to be read by a human and that is pushed to a system such as a bug or ticket queue, an email alias, or a pager. Respectively, these alerts are classified as *tickets*, *email alerts*, and *pages*

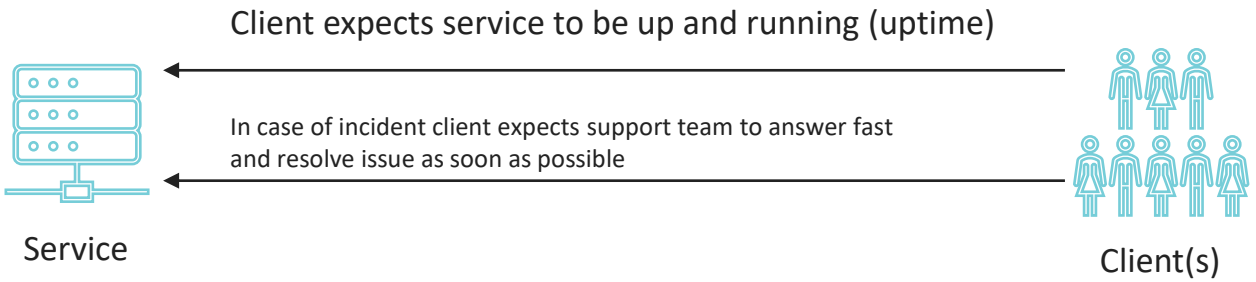
Why monitor?


Analyzing long-term trends
Comparing over time or experiment groups
Alerting
Building dashboards
Conducting *ad hoc* retrospective analysis (i.e., debugging)

Service Level - SLA

SLA is a **S**ervice **L**evel **A**greement. This is an agreement between service provider and service consumer. This agreement establishes expectations on such measurable metrics as:

- Uptime
- Response time
- Responsibilities



Service Level - SLO

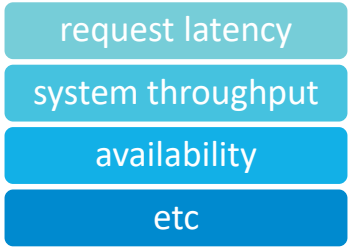
SLO is a **S**ervice **L**evel **O**bjective. SLO is an agreement about a specific metric defined under SLA. In other words, SLA is more general agreement while SLO is a targeted agreement to specify measurable metric through measurable time period. Metrics which can be included (but not limited) to SLI are following:

Object	SLO	Measurement Period
Uptime	Service must be available 99.98% of the period	Year
Service Desk Response	95% of requests must be taken to work not later than 30 minutes after submit	Month
Resolution Time	99% priority 1 tickets must be resolved in 4 hours	Month

Service Level - SLI

SLI is a **S**ervice **L**evel **I**ndicator—a carefully defined quantitative measure of some aspect of the level of service that is provided according to SLO. Example: under SLA defined that service availability will be 99.96%, this means that in SLI will be included guaranteed 99.96 % uptime for a time period. SLI here will be an actual uptime of the service for the time period.

Examples of SLIs:

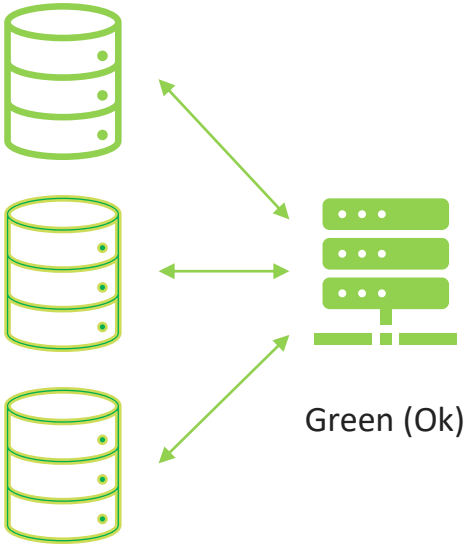


Monitoring

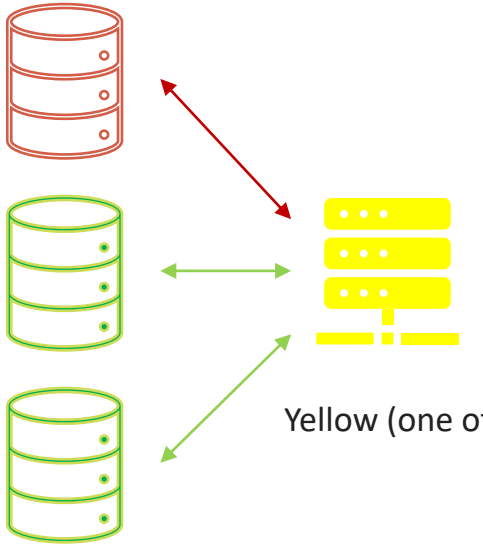
Monitoring types

System health

System health is set of metrics representing current state of a system. It includes but not limited to such metrics as state of hardware, ability to perform operations and etc. Illustrations below show difference between abstract cluster with one server on three databases. On left image all components are operational and overall cluster health is green, while on right image one of DBs is down and overall cluster health isn't ok.



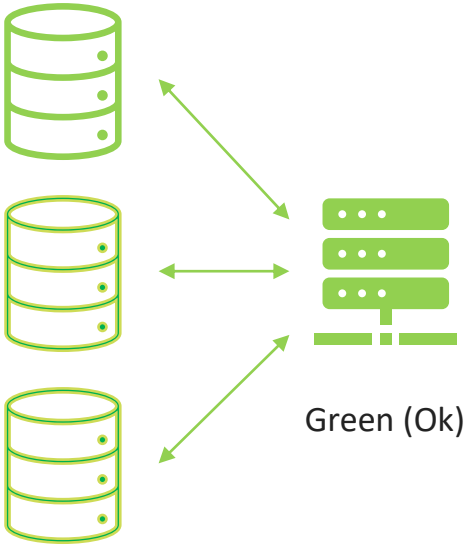
Green (Ok)



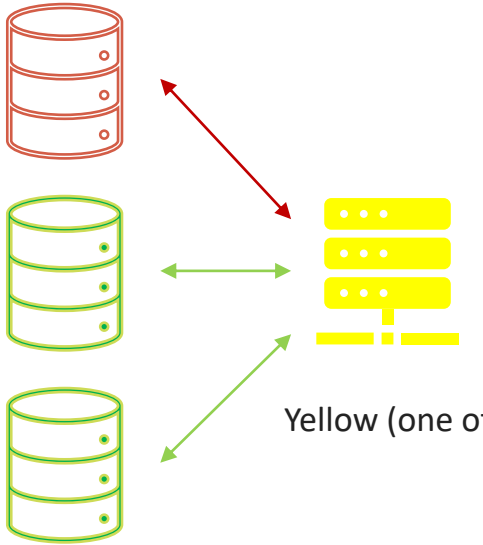
Yellow (one of components is not operational)

System performance

System performance is ability of a system to satisfy expected level of system operations (read/write/computations), time spent on these tasks and etc. On previous slide we had an example with tow clusters with one failed DB instance. In case of performance it means that cluster with failed DB may perform read/write operations slower.



Green (Ok)



Yellow (one of components is not operational)

Tool set

It’s important to have system health and performance under control to understand current state of a system, resolve occurred issues as soon as possible and have debug information to it. Likely we have a lot of systems and applications to deal with it. A short overview:

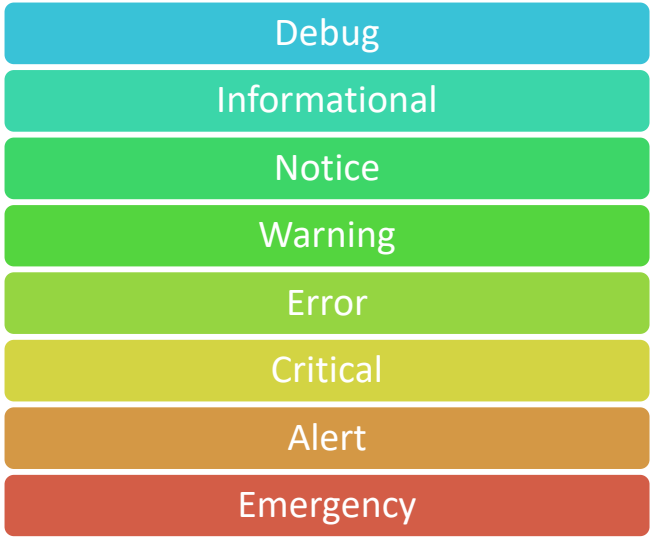
Monitoring systems:	• Prometheus • Grafana • Zabbix
Applications (UNIX specific):	• top/htop • free • iostat • iftop

It’s short and incomplete list with most popular apps and systems just to give you general understanding.

Logging

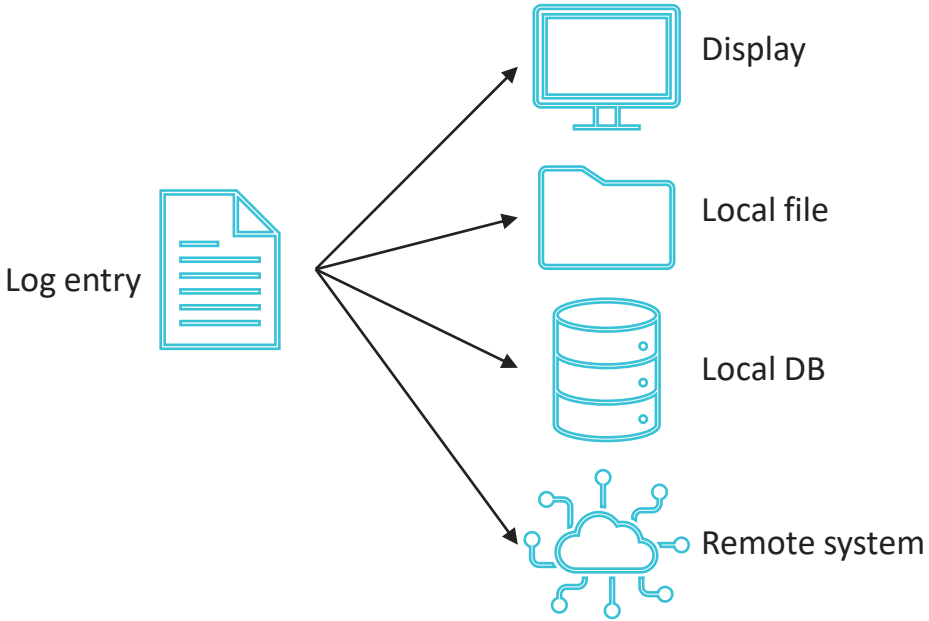
What is logging

Logging is a process of storing records to a file, data base or remote system. Each record represents an event in observable system. Log entries in UNIX have following severities:



Storage for log entries

Display – log entries can be redirected to user's display
File – log entries are written to a file on local disk
Database – log entries are written to local database such as systemd journal
Remote system – log entries are sent to remote system such as Graylog or Elasticsearch



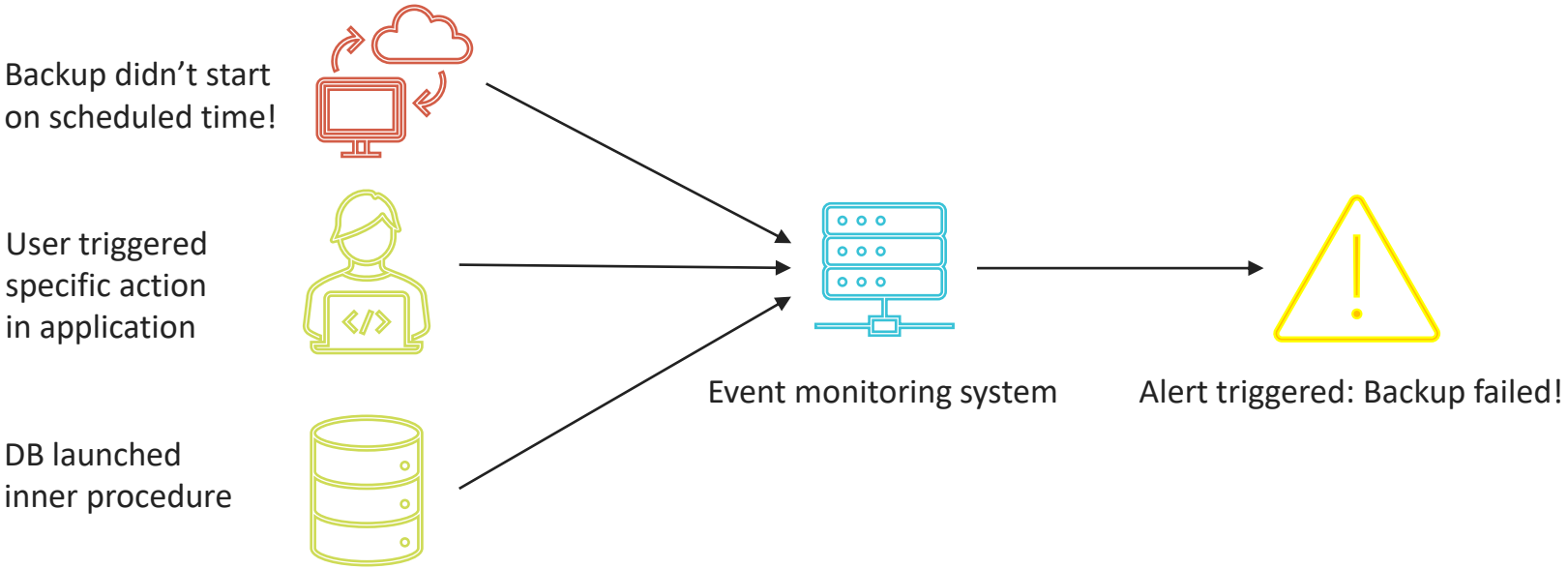
Why we need logging

Logging is extremely powerful feature for debugging, investigating issues or retrospective analysis. You can’t say you have well designed system with predictable behavior unless you don’t have configured logging subsystem.

Event monitoring

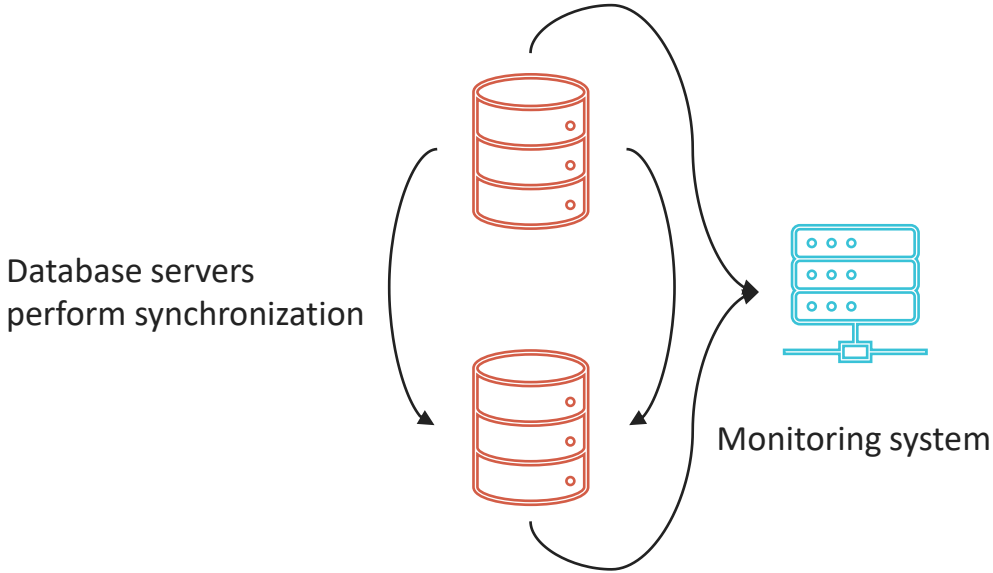
Definition

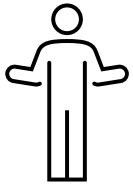
Event monitoring aims to catch specific event on observable system. Event can be a launched process in operating system, database backup produce or any action made by user in application and etc.



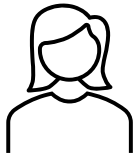
Why we need event monitoring

Event monitoring shows deeper picture than just a general monitoring of system resources. It allows to understand correlation between current resources consumption and ongoing processes. It also brings more understandable picture of user's actions in case of event monitoring of application and etc.





Oh no, we have a huge CPU and disk load on our DB servers!



Hold on, this load caused by routine data synchronization and will gone soon.

Business metrics and security

Short introduction

Monitoring solutions can be related not to IT metrics only. Other departments probably will request to monitor metrics they are interested in. Business and security metrics is not covered fully in this course as it requires deep understanding of business domain and depends on observable system. Only short overview provided for general understanding.

Business metrics are quantifiable measure of business side of the project. Example of business metrics:

- KPIs
- Sales
- Stock balance
- Customers (leads/losts/etc)

Security metrics allow us to protect project's data and prevent data loss. Example of security metrics:

- Operating system version
- Version of installed applications
- Login attempts
- Login locations
- Log of user's actions

THANK YOU