



# Data Warehouse Monitoring mit Icinga 2.

Köln, 22. August 2015

Sebastian Henrich ([sebastian.henrich@cologne-intelligence.de](mailto:sebastian.henrich@cologne-intelligence.de))

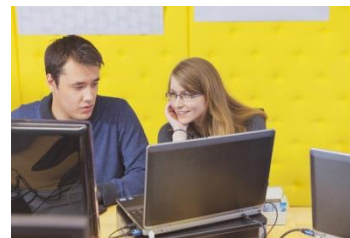
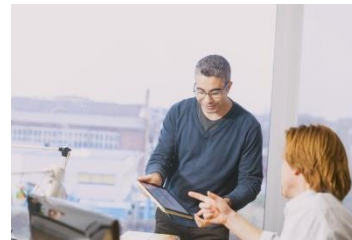
[www.cologne-intelligence.de](http://www.cologne-intelligence.de)

# Agenda



1. Über uns
1. Motivation
2. Technik
  1. Icinga 2
  2. Icinga Web 2
  3. Graphite
3. Checks
4. Livedemo
5. Fazit

# Über uns



Wir bieten hochqualifizierte Mitarbeiter um Ihr Projekt erfolgreich zu gestalten:

- Business Analysten
- Agile Projektmanager
- Technische Architekten
- Mobile Entwickler
- JAVA Entwickler
- BI Experten
- Security Experten
- Frontend Designer
- Test Manager

Wir verstehen uns als Ihr Partner und Mitdenker, der für Sie Lösungen entwickelt, die pragmatisch sind und langfristig funktionieren. Deshalb prüfen wir sie auch mit Argusaugen auf ihre Nachhaltigkeit. Der Aufwand lohnt sich.



# Fragestellungen während der Entwicklung



- Wie lange läuft ein ETL-Prozess?
- Welche CPU-Last erzeugt ein ETL-Prozess?
- Werden alle vorhandenen Ressourcen genutzt oder liegen Ressourcen brach?
- Wer ist für die langsame Verarbeitung verantwortlich?  
Der ETL-Prozess oder die Datenbank?
- ...

# Fragestellungen während dem Betrieb



- In welchem Maße steigt die zu verarbeitende Datenmenge?
- Welchen Einfluss hat die steigende Datenmenge im Data Warehouse auf die Bewirtschaftung und Auswertungsprozesse?
- Ist noch genügend Festplattenkapazität, Tablespace, usw. vorhanden?
- Sind meine ETL-Jobs gelaufen und wie viel Zeit haben Sie benötigt?
- Sind alle Quellsysteme erreichbar?
- Sind die Komponenten meines Data Warehouse Systems erreichbar?
- Wurden alle Quelldateien verarbeitet oder liegen sie noch in den Verzeichnissen zur Abholung?
- ...

# Icinga 2



## Geschichte:

- 16. Dezember 2009 Icinga Core 1.0 als Nagios Fork
- 16. Juni 2014 Icinga 2
- Aktuelle Version 2.3.8
- Webseite: <https://www.icinga.org/icinga/icinga-2/>

## Funktion:

- Führt die eigentlichen Überprüfungen (Checks) aus
- Benachrichtigt die Benutzer über Ereignisse via E-Mail, XMPP, IRC, Twitter, usw.

# Icinga Web 2



## Geschichte:

- Nachfolger von Icinga Web
- Angekündigt am 20. November 2013
- Aktuelle Version 2.0.0 RC1
- Webseite: <https://github.com/Icinga/icingaweb2>

## Funktion:

- Erhält Daten über DB IDO (Database Icinga Data Output) oder Livestatus von Icinga 2
- Wird über einen Webserver (z.B. Apache 2) bereitgestellt
- Hat eigene Datenbank für Benutzer, Einstellungen, etc. zusätzlich zur IDO Datenbank
- Benutzt Command Pipe um Kommandos an Icinga 2 zu schicken
- Einfaches Web Setup

# Graphite



## Geschichte:

- Entwickelt in 2006 von Chris Davis bei Orbitz
- 2008 unter der Apache 2.0 Lizenz veröffentlicht
- Aktuelle Version 0.9.13
- Webseite: <https://github.com/graphite-project> (Source) <http://graphite.readthedocs.org/> (Dokumentation)

## Funktion:

- Speichert Zeitreihendaten
- Rendert Diagramme aus den Zeitreihendaten
- Carbon: Daemon der die Zeitreihendaten annimmt
- Whisper: Datenbankbibliothek zum Speichern von Zeitreihendaten, ähnlich RRD
- Graphite Webapp:
  - auf Django basierende Webapp,
  - generiert Diagramme On-Demand mit Hilfe von Cairo
- Eingebaute Unterstützung in Icinga 2
- Modul für Icinga Web 2 verfügbar (<https://github.com/findmypast/icingaweb2-module-graphite>)



# Was sollte überwacht werden?



- CPU Load inkl. IOWait
- Arbeitsspeicher
- Festplatte
  - Kapazität
  - Auslastung
- Netzwerk
- Datenbank
- Erreichbarkeit (Ping)
- ETL-Prozesse

# CPU, Arbeitsspeicher, Festplattenkapazität, Netzwerk

## CPU Load:

- Lokal: `check_load`
- Remote: NRPE Plugin oder SNMP (`check_snmp_load.pl`)

## CPU IoWait:

- Lokal: `check_iostat` (<https://github.com/dnsmichi/icinga-plugins>)
- Remote: NRPE Plugin

## Arbeitsspeicher

- Lokal: `check_memory`
- Remote: NRPE Plugin oder SNMP (`check_snmp_mem.pl`)

## Festplatte:

- Lokal: `check_disk`
- Remote: NRPE Plugin oder SNMP (`check_snmp_storage.pl`)

## Netzwerk:

- Lokal: `check_traffic.sh` ([https://github.com/cloved/check\\_traffic/](https://github.com/cloved/check_traffic/))
- Remote: NRPE Plugin oder SNMP (`check_snmp_traffic`, [https://github.com/treed/check\\_snmp\\_traffic](https://github.com/treed/check_snmp_traffic))

# Datenbankserver, Quellsysteme, Dateien



## ConSol Labs (<https://labs.consol.de>)

- DB2: `check_db2_health`
- MS SQL: `check_mssql_health`
- MySQL: `check_mysql_health`
- Oracle: `check_oracle_health`
- SAP: `check_sap_health`

## Dateien

- `check_file_exists_glob` ([https://github.com/mclarkson/check\\_file\\_exists\\_glob](https://github.com/mclarkson/check_file_exists_glob))

# ETL-Prozesse



## Allzweckwerkzeuge

- `check_mysql_query`  
([https://www.monitoring-plugins.org/doc/man/check\\_mysql\\_query.html](https://www.monitoring-plugins.org/doc/man/check_mysql_query.html))
- `check_mssql_query.py`  
([https://github.com/opinkerfi/nagios-plugins/blob/master/misc/check\\_mssql\\_query.py](https://github.com/opinkerfi/nagios-plugins/blob/master/misc/check_mssql_query.py))
- `check_oracle_query.py`  
([https://github.com/opinkerfi/nagios-plugins/blob/master/misc/check\\_oracle\\_query.py](https://github.com/opinkerfi/nagios-plugins/blob/master/misc/check_oracle_query.py))

## Funktionsprinzip

1. Abfrage wird gegen die Datenbank abgefeuert
2. numerischer Rückgabewert wird ausgewertet
3. je nach Schwellwert wird eine Warnung ausgelöst

Auch Statistik- und Logtabellen der ETL-Werkzeuge lassen sich so auswerten

LIVE DEMO

**Los geht's!**

# Fazit

- Icinga 2 lässt sich einfach aus Paketquellen installieren
- Icinga Web 2 bietet ansprechende Oberfläche, ebenfalls einfach zu installieren
- direkte Graphite Unterstützung in Icinga 2 bietet schnellen Start
- Installation/Konfiguration Graphite und Modul für Icinga Web 2 etwas hakelig
- Verlauf über Graphite Diagramme gut nachvollziehbar
- zur Unterstützung der Entwicklung kurze Check Intervalle notwendig
- Checks kosten Ressourcen



**Vielen Dank für Ihre  
Aufmerksamkeit!**